Received: 5 July 2018 | Revised: 21 October 2018 | Accepted: 23 November 2018
DOI: 10.1002/ett.3554

RESEARCH ARTICLE

WILEY

Information-centric routing in MSN based on community detection

Junling Shi[1] | Xingwei Wang[1] | Min Huang[2] | Jianhui Lv[3] | Keqin Li[4] | Jie Li[1]

[1]College of Computer Science and Engineering, Northeastern University, Shenyang, China
[2]College of Information Science and Engineering, Northeastern University, Shenyang, China
[3]Central Research Institute, Huawei Technologies Co. Ltd, Shenzhen, China
[4]Department of Computer Science, State University of New York at New Paltz, New Paltz, New York

Correspondence
Xingwei Wang, College of Computer Science and Engineering, Northeastern University, Shenyang 110819, China.
Email: wangxw@mail.neu.edu.cn

Min Huang, College of Information Science and Engineering, Northeastern University, Shenyang 110004, China.
Email: mhuang@mail.neu.edu.cn

Funding information
Major International (Regional) Joint Research Project of NSFC, Grant/Award Number: 71620107003; National Natural Science Foundation of China, Grant/Award Number: 61572123 and 61502092; National Science Foundation for Distinguished Young Scholars of China, Grant/Award Number: 71325002; MoE and ChinaMobile Joint Research Fund, Grant/Award Number: MCM20160201; Program for Liaoning Innovative Research Term in University, Grant/Award Number: LT2016007

Abstract

Mobile social network (MSN) offers a new perspective on mobile ad hoc communication since its routing principle is based on the human social relations. Although social-based routing can improve routing efficiency considerably, obtaining such social information is difficult to be achieved. In information-centric networking (ICN), content names reveal useful social information among users. In addition, each node stores and caches the received content to satisfy the forthcoming content requests in ICN due to in-network caching. In this work, the proposed MSN routing relies on named data networking, which is a well-known ICN paradigm. By the communities, which are detected based on users' interest preferences, an interest packet is delivered to the content provider based on the interest similarities among mobile users. Then, by communities, which are detected based on the nodes' encounter regularities, a data packet is returned to the interest requester according to the social relationships among mobile users. The content is cached at nodes according to both social and interest communities. Experiments and performance evaluations show that the proposed scheme has better message delivery ratio and lower network overhead than the other existing ones.

1 | INTRODUCTION

Mobile social network (MSN) attempts to exploit the social behaviors of users to forward time-insensitive data via the wireless communication established among the mobile nodes. In such network, nodes seek opportunistic device-to-device (D2D) data dissemination instead of uploading and downloading directly from the Internet to reduce the communication cost, especially for the large volume data (eg, video).[1] To this end, effective routing schemes in MSN have become an active research area and attracted intensive attention up to now. However, in case of high mobility, connections may break and the communication is disrupted until the new one is established.[2] Although utilizing mobile users' social relationships can help improve routing efficiency in MSN, exploiting their implied social regularities is a big challenge.[3] Furthermore, nodes with strong social positions (eg, high centrality) are frequently asked to forward packets by other nodes. However, such nodes may easily become the bottleneck nodes, and thus, the efficiency of routing mechanisms is harmed. Moreover, in traditional distributed MSN architectures, the destination node identifications (addresses) are always assumed to be known when routing. In fact, mobile users often do not know which nodes can satisfy their requests when requesting data, that is, their generated interest packets cannot be embedded with the destination node IDs, which is really a different circumstance from the traditional architectures. Such issues motivate us to investigate a new method to explore the social behaviors of users, eliminate the bottleneck nodes, and adapt to the scenario of unknown destination node identifications.

Information-centric networking (ICN) shifts the current host-oriented communication model toward the information-centric one, and interest requester does not need to know which node can offer the requested contents to them.[4] It relies on location-independent naming, in-network caching, and name-based routing to effectively distribute content over the network. In named data networking (NDN), names are human friendly.[4] Once the name-based routing is implemented in MSN, the implied social information, for example, users' interests and preferences, can be easily exploited by analyzing such human-friendly content names. Furthermore, since MSN works on the principle of store-carry-forward mechanism,[5] the requested data (content) can be responded by the nodes, which carry the corresponding data packet rather than just by the original content provider. Therefore, by caching content that will be frequently requested in the future at nodes, the network traffic can be relieved, and thus, the bottleneck nodes can be eliminated. For instance, one socially popular node holds some popular content, leading to its frequently being asked to provide content for other nodes, which may make bottleneck nodes appeared and traffic congestion emerged. Instead, with in-network caching, a node, which is closer to the interest requester than the original content provider, has one copy of the popular content and can provide it conveniently. Such a mechanism can eliminate the bottleneck nodes and further balance the network traffic load.

In sociology, a community is a clustering of entities that are closely linked to each other, by either direct linkage or easily accessible entities that can act as intermediates. We bring community structures into MSN routing since it can benefit routing efficiency significantly.[6] At present, some existing algorithms[7,8] based on the encounter history among nodes have been employed to detect communities. With such communities, packets are delivered by being forwarded to nodes, which have regular encounter histories to destination nodes. In ICN architecture, interest packets are delivered to nodes, which can provide the requested contents, rather than to some specific nodes. Therefore, the previous community detection algorithms cannot work well for interest packets routing because the frequently encountered nodes do not certainly represent that they hold the requested content. To this end, community detection that can work for the interest packet routing should be considered in ICN-based MSN. Furthermore, even if in the same communities, different community members also have relationships with different levels of closeness among them. Such different relationships may have a significant influence on routing efficiency and thus should be taken into account carefully.

Based on the previous discussion, we design an information-centric routing in MSN based on community detection (IRCD), and the main contributions are summarized as follows.

- One of the popular ICN paradigms, NDN,[9] is leveraged for MSN routing. By analyzing the name-based content requests, the user interest preferences are exploited, and we implement in-network caching in nodes so as to respond to the forthcoming interest requesters.
- We propose an interest and social based community detection algorithm to provide suitable communities for interest packets and data packets. The proposed encounter density metric is used to detect social communities. We classify social community members into strong social friends and weak social friends, based on which, the data packet routing is devised.
- Rather than the simple content categories, the specific content categories of the requested contents are used to define the proposed content interest similarities for interest community detection. We classify interest community members into direct interest friends and indirect interest friends, based on which, the interest packets routing is devised.
- In data packet routing, nodes cache contents in different caching time periods, which are defined based on the proposed community member interests. The content is only cached at one social community member rather than redundantly cached at every node.

The rest of this paper is organized as follows. Section 2 reviews related work and compares our work with them. Section 3 presents the system framework of the proposed routing scheme. Section 4 presents the proposed community detection algorithm. Section 5 describes the devised interest packet routing and data packet routing. Section 6 reports simulations and performance evaluations. Section 7 draws conclusions.

2 | RELATED WORK

For MSN routing, in the work of Bulut and Szymanski,[10] a friendship-based routing scheme was presented, in which the periodically differentiated friendship relations were used in forwarding messages. In the work of Chen et al,[11] aiming to develop a systematic understanding of MSN, social ties in human social networks were exploited to enhance cooperative D2D communications. In the work of Xiao et al,[12] a zero-knowledge multicopy routing algorithm was proposed for homogeneous MSN, in which all mobile nodes shared all community homes. In the work of Pholpabu and Yang,[13] by considering the problem of dynamic routing in MSN and exploiting the social characteristics of mobile nodes, a social contact probability assisted routing protocol was proposed. In the work of Pholpabu and Yang,[14] a routing protocol for MSN was proposed by exploiting the fact that people's daily lives had community preferences. In the work of Xiao et al,[15] a distributed optimal community-aware opportunistic routing algorithm was proposed and a home-aware community model was built, whereby the MSN was turned into a network, which only included communities. In the work of Gupta et al,[16] the planned mobilities of the nodes were partially known a priori. The optimal routes were computed using the most reliable path principle, in which the negative logarithm of a node pair's adjacency probability was used as a link weight metric.

However, the previous MSN routing mechanisms have two drawbacks. Firstly, they arbitrarily forwarded packets to nodes with strong social positions, bringing hotspots to the network easily and thus degrading the network performance. Secondly, all the users' social profiles were exploited from their encounter histories, which cannot accurately represent users' interest properties and guarantee efficient packet delivery. In contrast, we devise the MSN routing scheme based on NDN paradigm, resolving the hotspot problem by in-network caching and exploiting social profiles of the content preference from the user-defined content names.

In the work of Pu et al,[17] a social-aware NDN (sNDN) framework to achieve efficiently cooperative content retrieval was proposed. The cooperative content retrieval was advocated, which was a novel and content-centric service paradigm via D2D communications to reduce cellular traffic volume and to facilitate user wireless connectivity in mobile networks. In the work of Nazir et al,[18] people's predictable social patterns were exploited to improve the content delivery performance and lower end-to-end delay in time-critical applications. In the work of Anastasiades et al,[19] an information-centric routing protocol for mobile ad hoc networks was proposed. It was based on the implicit content discovery via broadcast and dynamically created unicast links for efficient content retrieval. In the work of Kuang and Yu,[20] an information-centric architecture called content-scent–based architecture (CSAR) was proposed for mobile ad hoc networks. In CSAR, content had its special content scent and could be found by tracing the scent, which is spread over the network.

For all the previous researches, which use information-centric network architecture, they neglect the inherent community structures that exist among mobile social users.

In the work of Lu et al,[21] the community-based content retrieval architecture was proposed, which was highly scalable in disruption-tolerant mobile information-centric networks. In the work of Xu et al,[22] the routing scheme delivered the mobile social content by using the selective agent and relay nodes. However, they both considered that a node just forwarded packet based on one kind of community whenever forwarding interest packets or data packets. Such consideration brings unsuitable routing guidance and makes low routing efficiency because the two kinds of packets should be based on different communities with different clustering factors.

3 | SYSTEM FRAMEWORK

The framework of IRCD is shown in Figure 1. The IRCD has the following two modules: social-interest community detection (SICD) and routing decision (RD). In SICD, there are social community detection and interest community detection. In RD, there are interest packet routing (IR) and data packet routing (DR). Specifically, in IR, there are intracommunity forwarding and intercommunity forwarding. When an interest packet is received by a node, based on the interest communities detected by SICD, the node checks whether it belongs to the destination interest community according to the requested content category. If yes, the interest packet is forwarded by the intracommunity forwarding in IR; otherwise, it is forwarded by the intercommunity forwarding in IR. In DR, there are in-network caching, intracommunity forwarding and intercommunity forwarding. When a data packet is received by a node, the carried content is cached based on the interest communities and social communities to fulfill the forthcoming interest packets. Then, based on the social communities detected by SICD, if the interest requester is in the same social community as the node, the data packet is forwarded by the intracommunity forwarding in DR; otherwise, it is forwarded by the intercommunity forwarding in DR.

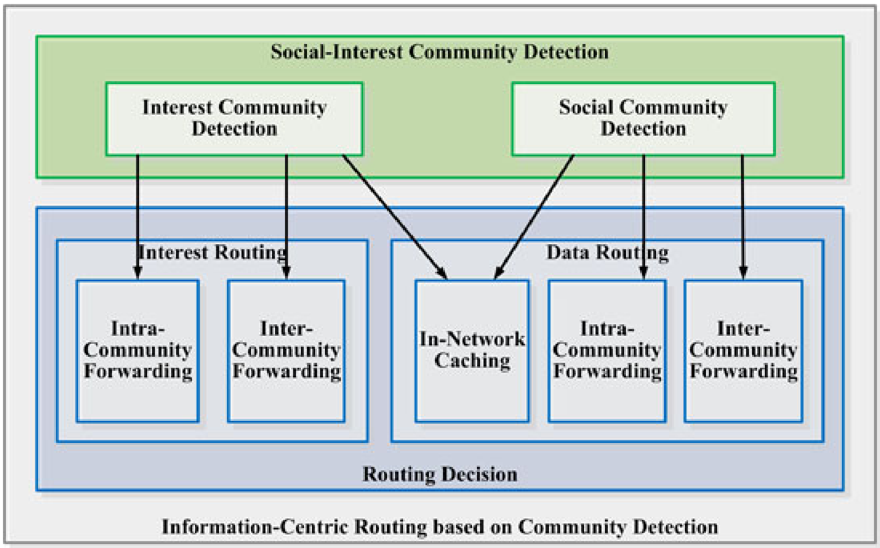

FIGURE 1 Information-centric routing based on community detection system framework

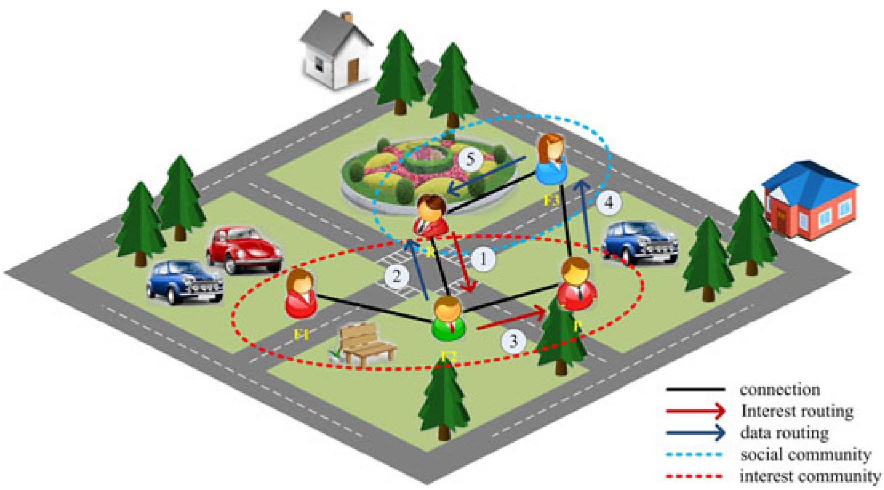

FIGURE 2 Information-centric routing based on community detection scenario and workflow

The workflow of IRCD is shown in Figure 2. Firstly, based on SICD, mobile devices held by users (called nodes) are arranged into interest communities (nodes $F1$, $F2$, and P) and social communities (nodes $F3$ and R). When the interest requester R generates an interest packet, R forwards it to $F2$ by the intercommunity forwarding in IR since R does not belong to the interest community of the requested content category. If $F2$ caches the requested content, then it generates a data packet and returns it back to R; otherwise, $F2$ forwards the request to P by the intracommunity forwarding in IR since $F2$ belongs to the interest community of the requested content category. Assume that P is the content provider, it generates the data packet to be returned to R. Next, P forwards this data packet to $F3$ by the intercommunity forwarding in DR, and then $F3$ does the in-network caching locally. $F3$ forwards the data packet to R by intracommunity forwarding in DR finally.

4 | COMMUNITY DETECTION ALGORITHM

We build communities based on the local social information that is maintained by each node. Social relationships among nodes influence their encounters. Furthermore, if two nodes have a strong social relationship, they can encounter each other with higher probability than the others, which is determined by human beings' social nature. Meanwhile, content preference of a node represents his interest. If two nodes have high-interest similarity, they prefer to share content resources with each other.

In the proposed IRCD, interest packets are forwarded to nodes, which have a high-interest similarity with the content provider, whereas data packets are forwarded to nodes, which have high social relationship with the interest requester. We first detect social communities based on the proposed encounter densities among nodes and then detect interest communities based on interest similarities among nodes.

4.1 | Encounter density

Two nodes encountering each other regularly in history indicate that they keep a good social relationship. The regularity can be reflected in several ways, for example, encounter times and encounter duration.[1] In history encounter time, if two nodes encounter each other with many encounter times and long encounter duration, they can encounter each other with a high possibility in the future. We use the encounter density to measure such property.

Suppose that there are m mobile nodes in the network topology and R_i is an arbitrary node, $1 \leq i \leq m$.

Definition 1 (**Unit period and encounter element**).
The history time duration h is divided into some unit periods. The encounter element represents whether two nodes have encountered each other in the corresponding unit period. If nodes R_i and R_j encounter each other in one unit period ξ, the corresponding encounter element is set to be 1; otherwise, it is set to be 0. Let ge_{ij}^{o} represent the encounter element between R_i and R_j within the oth unit period, and we have

$$ge_{ij}^{o} = \begin{cases} 1, R_i \text{ encoutners } R_j \text{ in time } [(o-1)\xi, o\xi) \\ 0, \text{otherwise} \end{cases} \tag{1}$$

Then, in h, the number of a unit periods np *is*

$$np = \frac{h}{\xi}. \tag{2}$$

Definition 2 (**Encounter density**).
Encounter density of two nodes is defined as the number of their nonzero encounter elements divided by np. Let ne represent the number of nonzero encounter elements and ed_{ij} represent the encounter density between R_i and R_j, and we have

$$ed_{ij} = \frac{ne}{np}. \tag{3}$$

4.2 | Content interest

Content names in NDN are composed of multiple components arranged in a hierarchy, and an interest request includes several components. Therefore, a user's historical requests can reflect his interest preferences of content.[23] Users have different preferences on different kinds of contents, for example, some of them are interested in sports, whereas some of them are interested in movies, that is, they are interested in different content categories. Furthermore, although two users are interested in the same content category, for example, they are both interested in sports, they may be interested in different specific subcategories, for example, one is interested in basketball and another is in dancing.

We classify the content names into several categories, and the naming scheme is based on such categories. For example, in Figure 3, content names are classified into sport, music, travel, game, reading, and movie, and each classification is further classified into several classifications. For example, sport is classified into dancing, basketball, football, swimming, and ping pong. If a user interests in one content category, his generated interest request includes the components of this category with high probability. For example, Bob interests in football, then his interest request includes components “football” and “sports” with high probability. Inspired by this, we calculate the interest preferences of users based on the subcategories (name components) in their historical interest requests.

Let CC represent the set of all content categories that users are interested in, and CC has z elements. Furthermore, for each content category, there are n specific subcategories.

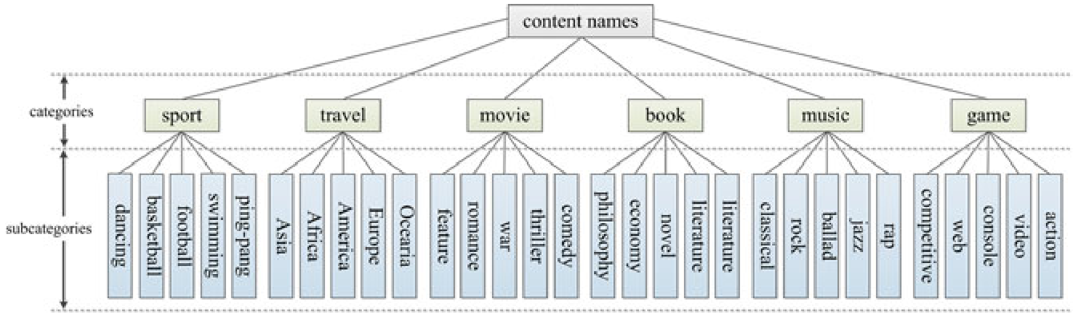

FIGURE 3 Naming scheme

Definition 3 (**Content interest matrix**).
Content interest matrix is a $z \times n$ matrix, in which the element on the tth row and the uth column is the number of the history requested times for the uth specific subcategory in the tth content category of node. Let M_i represent the content interest matrix of R_i, m^i_{tu} represent its element on tth row and the uth column, and we have

$$m^i_{tu} = rt^i_{tu}, \tag{4}$$

where rt_{tu} is the requested times for the uth subcategory in the tth content category of R_i. When R_i requests a content that belongs to the uth subcategory in the tth content category, we have

$$rt^i_{tu} \leftarrow rt^i_{tu} + 1. \tag{5}$$

Let M^g_i represent the gth row in M_i, and we have

$$M^g_i = A \times M_i, \tag{6}$$

where A is $1 \times z$ matrix, of which each element $a_l (1 \leq a_l \leq z)$ is defined as follows:

$$a_l = \begin{cases} 1, l = g \\ 0, l \neq g. \end{cases} \tag{7}$$

Definition 4 (**Content interest**).
Content interest on content category g of node R_i is the sum of its requested times of all subcategories of g. Mathematically, it is the sum of all elements in gth row in M^g_i. Let ci^g_i represent the content interest of R_i on content category g, and we have

$$ci^g_i = \sum_{k=1}^{n} m^{gk}_i, \tag{8}$$

where m^{gk}_i is the element of kth column in M^g_i.

Manhattan distance is a metric in which the distance between two points is the sum of the absolute differences of their coordinates.[24] We define interest similarity based on Manhattan distance as follows. Let is^g_{ij} represent the interest similarity of R_i and R_j on category g, and we have

$$is^g_{ij} = \frac{2}{\pi} \arctan \frac{10}{\sum_{k=1}^{n} \left| m^{gk}_i - m^{gk}_j \right|}, \tag{9}$$

where we use $2/\pi \cdot \arctan(x)$ to normalize the interest similarity.

4.3 | Social-interest community detection

When R_i encounters R_j, it calculates ed_{ij} according to (3). We use a preset threshold ω to decide whether R_i and R_j have a regular encounter history.

Definition 5 (**Strong-tie friend**).
If ed_{ij} is higher than ω, R_j is the strong-tie friend of R_i. Let ST_i represent the set of strong-tie friends of R_i, and we have

$$ST_i = \{R_j | ed_{ij} > \omega\}. \tag{10}$$

Definition 6 (**Common friends**).
If R_c is both strong-tie friend of R_i and R_j, R_c is the common friend of R_i and R_j. Let CF_{ij} represent the set of common friends of R_i and R_j, and we have

$$CF_{ij} = ST_j \bigcap ST_i. \tag{11}$$

Let nc_{ij} represent the number of common friends between R_i *and* R_j, and we have

$$nc_{ij} = |CF_{ij}|. \tag{12}$$

We use a preset threshold δ to decide whether R_i and R_j have many common friends.

Definition 7 (**Weak-tie friend**).
If nc_{ij} is higher than δ, R_j is the weak-tie friend of R_i. Let WT_i represent the set of weak-tie friends of R_i, and we have

$$WT_i = \{R_j | nc_{ij} > \delta\}. \tag{13}$$

Definition 8 (**Social community**).
If R_j is a strong-tie friend or a weak-tie friend of R_i, then R_j is in the social community of R_i. Let SoC_i represent the social community of R_i, and we have

$$SoC_i = ST_i \bigcup WT_i. \tag{14}$$

When R_i encounters R_j, if $R_j \in SoC_i$, then for each content category g $(g \in CC)$, it calculates each is_{ij}^g according to (9). We use a preset threshold θ to decide whether R_i and R_j have similar interest preference.

Definition 9 (**Direct interest friend**).
If is_{ij}^g is higher than θ and $R_j \in ST_i$, R_j is the direct interest friend of R_i on content category g. Let DT_i^g represent the set of direct interest friends of R_i on g, and we have

$$DT_i^g = \left\{ R_j \middle| is_{ij}^g > \theta \right\}. \tag{15}$$

If R_j is added into DT_i^g, the specific content subcategories on g held by R_j are also recorded by R_i.

Definition 10 (**Indirect interest friend and bridge node**).
When $R_i \in ST_j$, and $\forall\, R_q \in ST_j$, $q \neq i$, if $is_{iq}^g > \theta$, then R_q is the indirect interest friend of R_i on content category g, and R_j is the corresponding bridge node for R_i and R_q on g. Let IT_i^g represent the set of indirect interest friends of R_i on g, and we have

$$IT_i^g = \left\{ R_q \middle| R_q, R_i \in ST_j \bigwedge is_{iq}^g > \theta \right\}. \tag{16}$$

If R_q is added into IT_i^g, the content subcategories on g held by R_q are also recorded by R_i.

Definition 11 (**Category community**).
If R_j is a direct interest friend or an indirect interest friend of R_i on content category g, then R_j is in the category community of R_i on g. Let InC_i^g represent the category community of R_i on g, and we have

$$InC_i^g = DT_i^g \bigcup IT_i^g. \tag{17}$$

Definition 12 (**Interest community**).
The interest community of R_i consists of all the category communities of R_i. Let InC_i represent the interest community of R_i and ϕ represent null set, respectively, and we have

$$InC_i = \left\{ InC_i^g \middle| g \in CC \bigwedge InC_i^g \neq \phi \right\}. \tag{18}$$

In summary, the proposed SICD is shown in Algorithm 1. Lines 1-16 decide whether the encounter node belongs to the interest community. Lines 17-26 decide whether the encounter node belongs to the social community.

Algorithm 1 SICD algorithm

Input: the encounter node R_j
Output: SoC_i, InC_i or null
Begin
1: **if** $R_j \in SoC_i$
2: $flag$=false;
3: **for** g=1 to z
4: Calculate is_{ij}^g;
5: **if** the condition in (15) is satisfied
6: Add R_j into DT_i^g;
7: **if** the condition in (16) is satisfied
8: Add R_q into IT_i^g;
9: **if** $R_j \in DT_i^g$ or $R_j \in IT_i^g$
10: Add R_j into InC_i^g;
11: Add InC_i^g into InC_i
12: Set $flag$=true;
13: **end for**
14: **if** $flag$==true
15: Return InC_i;
16: **end if**
17: Calculate ed_{ij};
18: **if** the condition in (10) is satisfied
19: Add R_j into ST_i;
20: **if** the condition in (13) is satisfied
21: Add R_j into WT_i;
22: **if** $R_j \in ST_i$ or $R_j \in WT_i$
23: Add R_j into SoC_i;
24: Return SoC_i;
25: **end if**
26: Return null;
End

5 | ROUTING DECISION

5.1 | Interest packet routing

5.1.1 | Intracommunity forwarding in IR

When R_i receives an interest packet iP, of which the requested content c belongs to the specific subcategory s of the content category g, that is, $c \in s$, R_i first checks whether it has c. If it has, iP is delivered; otherwise, R_i checks whether its interest community members have c. If at least one member R_j has, R_i does the intracommunity forwarding in IR. Let CA_j represent the set of the cached content of R_j. Specifically, if at least one direct interest friend of R_i on g has c, that is,

$$R_j \in DT_i^g \bigwedge \exists c \in CA_j, \tag{19}$$

then all direct interest friends satisfying (19) are stored in set D. Suppose that R_e is an encounter node of R_i. If

$$R_e \in D, \tag{20}$$

then R_i forwards iP to R_e.

If there are no direct interest friends of R_i that has c but at least one of its indirect interest friend has it, that is,

$$R_q \in IT_i^g \bigwedge \exists s \in CA_q, \tag{21}$$

then all the bridge nodes of the indirect interest friends of R_i satisfying (21) are stored in set I. If

$$R_e \in I, \tag{22}$$

then R_i forwards iP to R_e. Specially, if there are more than one bridge nodes, the one that has the highest ed with R_i will be selected.

5.1.2 | Intercommunity forwarding in IR

If no interest community member of R_i has content belonging to s, R_i does intercommunity forwarding in IR. When R_i encounters R_j, it compares its content interest on content category g with R_j. If

$$ci_j^g > ci_i^g, \tag{23}$$

then R_i forwards iP to R_j. It is because that R_j has the higher content interest on category g and has higher probability to encounter the node that has the similar interest, that is, R_j is more likely to find content provider for iP than R_i.

In summary, the proposed IR is shown in Algorithm 2.

Algorithm 2 IR algorithm

Input: the interest packet iP
Output: a forwarding node or null
Begin
if R_i has content belonging to s
  Break;
end if
if iP is an intra-community packet of R_i
  if (20) or (22) are satisfied
    Return R_e;
  end if
end if
if iP is an inter-community packet of R_i
  if (23) is satisfied
    Return R_j;
end if
Return null;
End

5.2 | Data packet routing

5.2.1 | In-network caching

Due to the limited storage space, it is hard for the nodes to cache all the received contents. When R_i receives a data packet dP, of which carried content belongs to the specific subcategory s of content category g, R_i checks whether its social community members have content belonging to s. If at least one social community member has, in-network caching is no more needed because the future requests for the content belonging to s can be satisfied by the social community member; otherwise, if no social community member has, R_i does the in-network caching for the received content.

Definition 13 (**Community interest**).
Community interest on content category g of R_i is the sum of content interests on g of its interest community members on g. Suppose there are w community members in InC_i. Let mi_i^g represent the community interest on content category g of R_i, and we have

$$mi_i^g = \sum_{k=1}^{w} ci_k^g. \tag{24}$$

For each cached content c, R_i calculates the caching time period for c. Suppose T_i^c is the caching time period of c by R_i, and we have

$$T_i^c = T_0 \cdot mi_i^g, \tag{25}$$

where T_0 is the unit caching time period.

If the caching time period of one content has expired, the content is deleted. When caching a new content, if the cache space is full, the content with the least history requested times by others is deleted.
To verify the efficiency of the proposed caching policy (PCP), we propose the following theorem.

Theorem 1. *The PCP can achieve the highest response probability for content requests.*
The related proof is shown in Appendix.

5.2.2 | Intracommunity forwarding in DR

In ICN architecture, data packets are routed along the reverse path of the corresponding interest packets. However, it does not work in MSN because nodes move frequently and their connections are not always maintained. Therefore, we equip the interest requesters with interest packets to guarantee the data packet delivery even though the topology is changing due to node movement.

When the content provider receives an iP, it generates a data packet dP and equips dP with the corresponding interest requester. When R_i receives a data packet dP, of which destination node is R_r, it first checks whether it is R_r. If yes, then dP is delivered; otherwise, it checks whether R_r exists in the social community of R_i. If so, R_i does the intracommunity forwarding in IR. Specifically, if R_r is a strong-tie friend of R_i, that is,

$$R_r \in ST_i, \tag{26}$$

then R_i waits to forward dP to R_r until they encounter because they already have a very strong relationship to make them encounter.

If R_r is a weak-tie friend of R_i, that is,

$$R_r \in WT_i, \tag{27}$$

then the common friend that has the highest social relationship with R_r, which is denoted as R_h, is selected as the forwarding node, and we have

$$\forall R_c \in CF_{ir}, ed_{cr} \leq ed_{hr}. \tag{28}$$

Then, R_i waits to forward dP to R_h until they encounter, and R_h delivers dP to R_r finally.

5.2.3 | Intercommunity forwarding in DR

If R_r does not exist in the social community of R_i, R_i does intercommunity forwarding in DR. When R_i encounters R_j, R_i compares ed_{jr} with ed_{ir}. If

$$ed_{jr} > ed_{ir}, \tag{29}$$

then R_i forwards dP to R_j because R_j has higher social relationship with R_r and is more likely to encounter R_r than R_i.
In summary, the proposed DR is shown in Algorithm 3.

Algorithm 3 DR algorithm

Input: the data packet dP
Output: a forwarding node or null
Begin
if R_i is R_r
  Break;
end if
Delete the expired cached content;
if the cache space is full
  Delete the content which has the least history requested times;
if R_i doesn't have c
  Cache c and calculate T_c^i according to (25);
end if
if dP is an intra-community packet of R_i
  if (26) is satisfied
    Return R_r;
end if
if (28) is satisfied
  Return R_h;
 end if
end if
if dP is an inter-community packet of R_i
  if (29) is satisfied
  Return R_j;
end if
Return null;
End

6 | SIMULATIONS AND PERFORMANCE EVALUATION

6.1 | Simulation setup

6.1.1 | Simulator and traces

In this paper, since IRCD is basically an MSN routing scheme, we use opportunistic network environment (ONE)[25] to do performance evaluations over two traces. One is the Cambridge trace, which reflects good community structures,[26] and the other is the synthetic trace.[21] In Cambridge trace, the experimental data were gathered by the Haggle project, in which the iMotes were distributed to 36 students from the University of Cambridge Computer Laboratory and the data set covers 11 days. The synthetic trace consisted of 120 nodes, which were partitioned into two communities. One community consisted of 50 nodes, and the other community consisted of 70 nodes, of which 20 nodes frequently move between the two communities.

Since IRCD is an NDN-based routing scheme, we leverage NDN in the proposed framework as follows. A content request is encapsulated into an interest packet, which is represented by the requested content name. By leveraging the history requested content names of nodes, their same interest preferences can be analyzed and be used to detect communities. The returned contents are cached by the relaying nodes and used to respond to the forthcoming requests.

To implement and integrate the proposed framework with ONE, we mainly do the following works. Firstly, in the original ONE, there is only one type of packet with the determined destination (the node ID). However, there are two types of packets in the proposed framework, and the interest packet is addressed by the requested content name. Secondly, in the original ONE, the node has no caching ability and also has no cache space. Therefore, we enhance each node with cache and leverage the PCP to manage the cache. Finally, we add the community detection module to ONE.

6.1.2 | Benchmarks

We select two routing mechanisms as benchmarks and compare them with IRCD. One benchmark is from the work of Lu et al[21] called social-tie–based content retrieval among communities (STCRC). It allows users to request the content toward the higher social level nodes, which are more popular in the network. If the interest cannot be resolved in the interest requester's community, it will be forwarded to other communities for the content query. After the interest is matched by one of the content digests, the search process forwards the request to the next hop that has a higher possibility to reach the content provider based on the encounter history of the nodes. Finally, the requested content will be forwarded back from the content provider to the original interest requester. The other benchmark is from the work of Pu et al[17] called sNDN. The sNDN introduces friendship circle by grouping a user with his close friends of high physical proximity and content similarity. It constructs NDN routing tables based on friendship circle encounter frequency to navigate content requests among friendship circles. It leverages social properties in friendship circle to search for the final target as intrafriendship circle routing.

We compare the proposed IRCD with sNDN[17] and STCRC[21] for the following reasons. Among all the schemes mentioned in the related works,[10-22] some works[17-22] are routing schemes based on ICN paradigm, whereas others are not. We cannot compare IRCD with other works[10-16] because they have totally different packet formats from the ICN based routing schemes (eg, the interest packet in ICN-based routing scheme does not have destination address). Therefore, we do not select any scheme from the other works[10-16] as the benchmark. Among related works,[17-22] Lu et al[21] and Xu et al[22] consider community structures among mobile users, whereas other works[17-20] do not. Thus, we divide these related works[17-22] into two groups, that is, other works[17-20] and the works of Lu et al[21] and Xu et al[22] and select benchmarks from these two groups, respectively. In the former group, the work of Nazir et al[18] has no data routing policy, then it cannot become our benchmark. Besides, the works of Anastasiades et al[19] and Kuang and Yu[20] are flood-based routing schemes without social metrics considered, whereas IRCD is socially inspired. Therefore, they are also not suitable to become our benchmark. The work of Pu et al[17] is a social relationship-based routing scheme with interest routing and data routing devised, making it the most suitable benchmark in this group. In the latter group, since there are content servers in the network model of Xu et al,[22] which is different from the fully distributed network model in our proposed network model, we select the work of Lu et al[21] with nodes fully distributed as the benchmark.

Furthermore, in IRCD, we equip the in-network caching for responding to the forthcoming interest packets. However, in STCRC and sNDN, such in-network caching scheme is not embedded. In order to fairly compare the forwarding decisions of the previous three schemes, we equip in-network caching for the two chosen benchmarks. Since the proposed in-network caching that IRCD equipped is based on the detected social communities and interest communities, we cannot directly equip the same in-network caching method to STCRC and sNDN because they do not detect such communities. Hence, we add first-in–first-out (FIFO) as the caching replacement policy for STCRC (and get the so-called STCRC-F) and sNDN (and get the so-called sNDN-F), and equip FIFO for IRCD to replace the proposed in-network caching scheme (and get the so-called IRCD-F). Here, the principle of FIFO is as follows. When the cache space is full, the earliest cached content is deleted. As a consequence, we totally have six mechanisms simulated, ie, IRCD, STCRC, sNDN, IRCD-F, STCRC-F, and sNDN-F.

6.1.3 | Comparison metrics and parameter settings

When comparing performance among the previous six routing mechanisms, we use packet delivery ratio (PDR), Average HoP (AHP), Average DeLay (ADL), and Network OverHead (NOH) as the comparison metrics. Specifically, PDR is the ratio of the number of all delivered packets to the number of all issued packets; AHP is the ratio of the traversed hops of all delivered packets to the number of these delivered packets; ADL is the ratio of the experienced delay of all delivered packets to the number of these delivered packets; NOH is the ratio of the forwarded times of all issued packets to the number of all delivered packets. Their major meanings and the main simulation parameters are summarized in Table 1.

In our experiment, every node in the network acts both as the consumer and the provider. Therefore, in Cambridge trace, there are 36 consumers as well as 36 providers, and in the synthetic trace, there are 120 consumers as well as 120 providers. Moreover, according to the real request generation pattern of users,[27] in our simulation, all the users' request arrival numbers in a unite time are subject to the Poisson distribution. Specifically, the probability of request generation in a minute is 0.08. Therefore, the parameter of the Poisson distribution in Cambridge λ is set to be 3, and λ is set to be 10 in the synthetic. Besides, the user's request generation pattern for different contents follows the power-law distribution, where the exponent α is set to be 2.5. Simulation results are averaged over 30 independent runs and reported with the 95% confidence intervals.

TABLE 1 Comparison metrics and simulation parameter settings

Parameter	Setting
PDR	Packet successful delivery ratio of all issued packets
AHP	Average traversed hops of all delivered packets
ADL	Average experienced delay of all delivered packets
NOH	Average forwarded times that all the delivered packets spend
ξ: Unit period for encounter density	10 minutes
ω: Threshold for strong-tie friend	0.5
δ: Threshold for weak-tie friend	4(Cambridge); 10 (synthetic)
θ: Threshold for similar interest preference	0.6 (Cambridge); 0.8 (synthetic)
λ: Parameter in the Poisson distribution	3 (Cambridge); 10 (synthetic)
α: Exponent in the power-law distribution	2.5
Transmission range (synthetic)	10 m
Node speed (synthetic)	[7,8]km/h
The size of a content block	5 M
Simulation times	30
Transmission speed	2 Mbps
Content cache space	50 M
Number of nodes/subcategories	36 (Cambridge); 120 (synthetic)
Number of providers/consumers	36 (Cambridge); 120 (synthetic)
Parameter in the Poisson distribution	3 (Cambridge); 10 (synthetic)

Abbreviations: ADL, Average DeLay; AHP, Average HoP; NOH, Network OverHead; PDR, packet delivery ratio.

In MSN, each node is limited within its wireless propagation range. When two nodes are within the propagating range of each other, they can exchange packets. Besides, the content is cached along the path. The specific setting method of ξ, θ, ω, and δ are as follows.

- When configuring ξ, we set θ=1, ω=1, and $\delta = 10\,000$. Next, we test PDR under different values of ξ and observe that PDR increases when ξ gets shorter. This is because that the small value of ξ brings high division granularity for the historical encounter events and makes the accurate prediction for the future encounters. However, when ξ is shorter than 10 minutes, PDR keeps almost the same performance. Such phenomenon indicates that it is unnecessary to consider extremely high division granularity. Therefore, we set ξ to be 10 minutes. For ξ, when it is set to be 60 minutes, PDR has the worst performance with 0.0989 in Cambridge trace (72.3% worse than the best result) and 0.1827 (62.2% worse than the best result) in the synthetic trace, respectively.
- With the determined ξ, when configuring θ, we set ξ=10 minutes, ω=1, and $\delta = 1000$. Next, we test PDR under different values of θ, and observe that when θ=0.6 in Cambridge trace and θ=0.8 in the synthetic trace, respectively, PDR has the best performance. Therefore, we set θ=0.6 in Cambridge trace and θ=0.8 in the synthetic trace, respectively. For θ, when it is set to be 0, PDR has the worst performance with 0.2931 (60.3% worse than the best result) in Cambridge trace and 0.3619 (56.6% worse than the best result) in the synthetic trace, respectively.
- Similarly, when configuring ω, we set ξ=10 minutes, $\delta = 1000$, θ=0.6, and θ=0.8 for Cambridge trace and the synthetic trace, respectively. We test PDR under different values of ω, and observe that when ω=0.5, both two traces have the best PDR. Therefore, we set ω=0.5. For ω, when it is set to be 0, PDR has the worst performance with 0.5682 (29.5% worse than the best result) in Cambridge trace and 0.6954 (24.4% worse than the best result) in the synthetic trace, respectively.
- When configuring δ, we set ξ=10 minutes, ω=0.5, θ=0.6, and θ=0.8 for Cambridge trace and the synthetic trace, respectively. We test PDR under different values of δ, and observe that when δ=4 and 10 in the two traces, respectively, PDR shows the best performance. Therefore, we set δ=4 in the Cambridge trace and δ=10 in the synthetic trace. For δ, when it is set to be 0, PDR has the worst performance with 0.6154 (28.1% worse than the best result) in Cambridge trace and 0.7468 (23.6% worse than the best result) in the synthetic trace, respectively.

Then, we test the influence of some main parameters as follows. Figure 4 shows the PDR and ADL of different speeds of users in the synthetic trace, from which we can observe that when the speed increases, the performances get better at

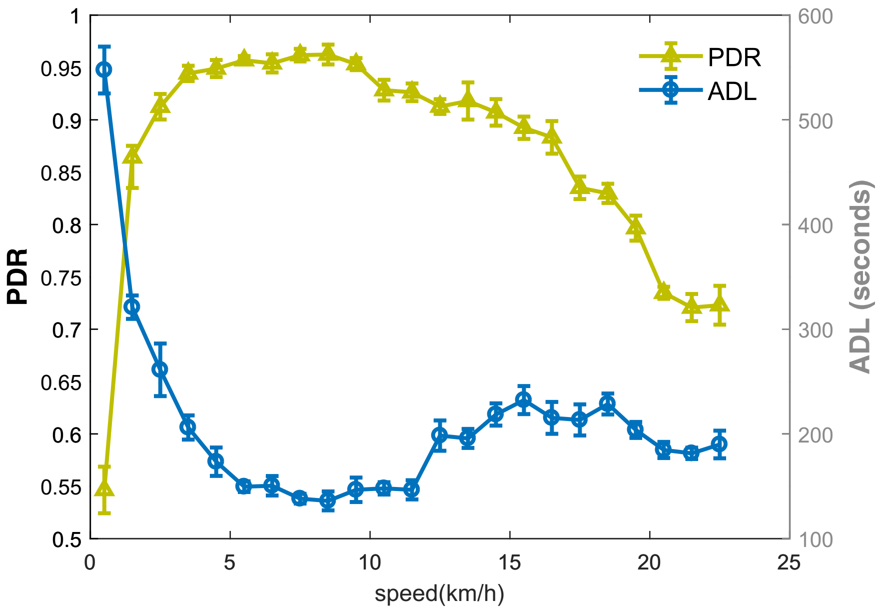

FIGURE 4 Impacts of different speeds of users. ADL, Average DeLay; PDR, packet delivery ratio

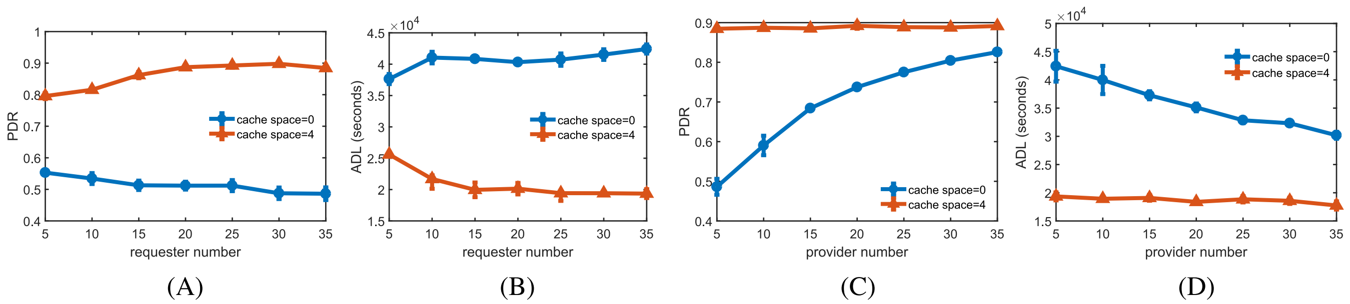

FIGURE 5 Impact of different numbers of providers and requesters A, PDR vs. provider number; B, ADL vs. provider number; C, PDR vs. requester number; D, ADL vs. requester number. ADL, Average DeLay; PDR, packet delivery ratio

first but then get worse. It is because that the slow speed (slower than 5) leads to the inadequate encounters among users, which reduce the packet forwarding and thus influence the performance. However, when the speed continues to increase (faster than 12), the probability of interruption of the link between two neighbors increases and the packet transmission cannot be finished on time, which also influence the performance.

Figure 5 shows the performances of different number of content providers and requesters in Cambridge trace, in which five content subcategories and two cache situations of each node, ie, no cache space and four cache spaces, are considered. We implement five providers when varying the number of requesters and 35 requesters when varying the number of providers. From Figures 5A and B, we can observe that when the number of requesters increases, the performances get worse if nodes have no cache space. It is because that more requesters generate more requests, which result in more requests that cannot be responded and decrease the response ratio. However, the performances get better if nodes have cache spaces. It is because that benefiting from the in-network caching, more requests bring more content categories cached by the nodes, which can improve the response efficiency for the requests. For the varying number of providers, from Figures 5C and D, we can observe that when the number of producers increases, the performances get better obviously if nodes have no cache space. It is because that more providers bring more abundant cached content in the network and thus improve the interest request response efficiency. However, when nodes have cache spaces, the performances get better slightly since the requesters can already guarantee the sufficient content categories cached by the nodes.

Figure 6 shows the performances of different number of cache spaces and content subcategories in Cambridge trace. We can observe that when the cache space increases, the performances get better since more cache space can improve the response ratio and speed for the requests. However, if the cache space continues to increase, the performances are not improved continually because the number of cached content of each node reaches the upper limit. It is worth mentioning

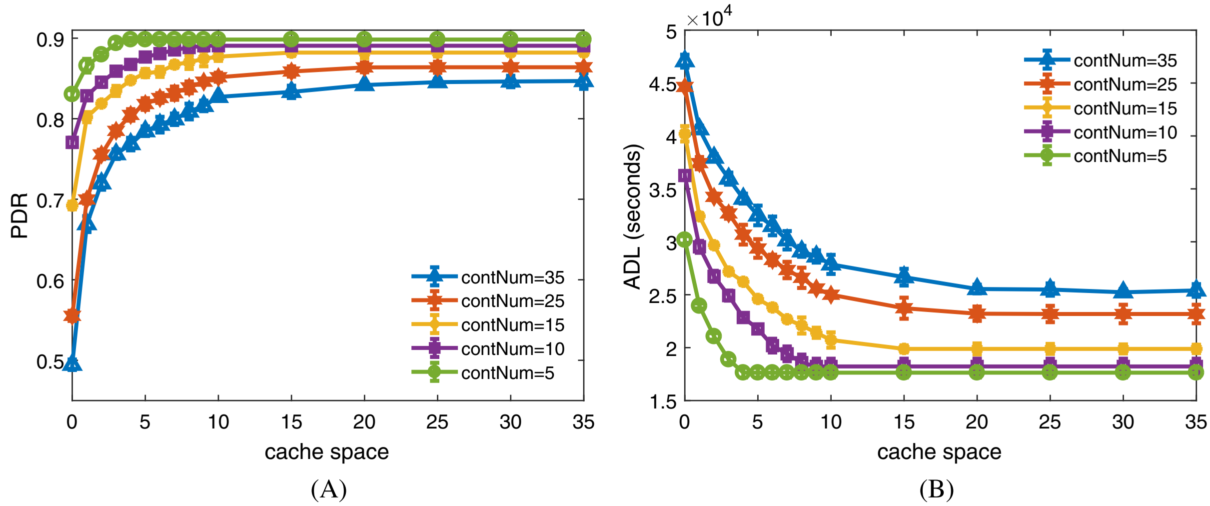

FIGURE 6 Impact of different numbers of content subcategories and cache size A, # PDR; B, # ADL. ADL, Average DeLay; PDR, packet delivery ratio

that the upper limit is less than the cache space since the PCP does not cache the redundant content in one social community. Moreover, when the number of content subcategories decreases, the performances get better and the steady state happens with less cache space. It is because that the requests can be responded more and faster with fewer content subcategories, and the number of cached content of each node reaches the upper limit earlier with the increasing cache spaces.

6.2 | Comparisons with benchmarks

The PDR for IRCD, STCRC, sNDN, IRCD-F, STCRC-F, and sNDN-F under different packet time to live (TTL) over the Cambridge trace and the synthetic trace are reported in Figure 7. For IRCD, STCRC, and sNDN, we observe that IRCD has the highest PDR. The reason is as follows. IRCD caches contents at nodes and thus can respond to the content requests with high satisfied ratio within limited TTL of packets, whereas STCRC and sNDN do not consider in-network caching for contents. Furthermore, IRCD detects communities for interest packets routing based on the interest similarities among mobile users, while detects communities for data packets routing based on the historical encounter information among

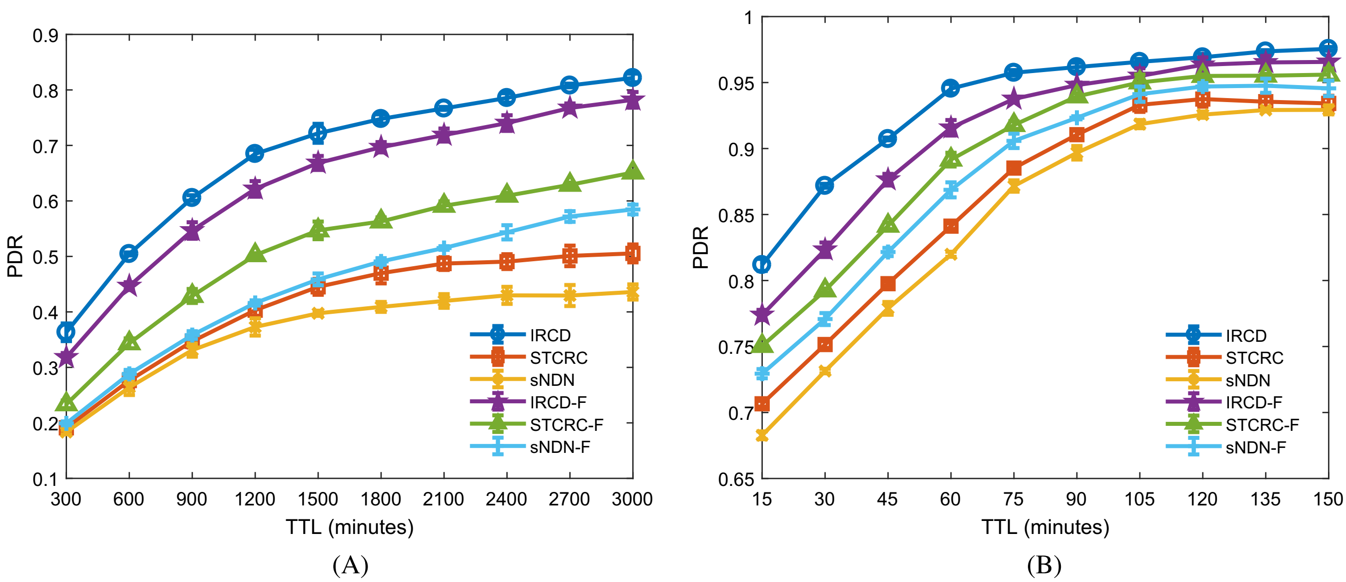

FIGURE 7 Packet delivery ratio (PDR) for the six schemes over Cambridge trace and synthetic trace A, # Cambridge trace; B, # Synthetic trace. IRCD, information-centric routing based on community detection; sNDN, social-aware named data networking; STCRC, social-tie–based content retrieval among communities; TTL, time to live

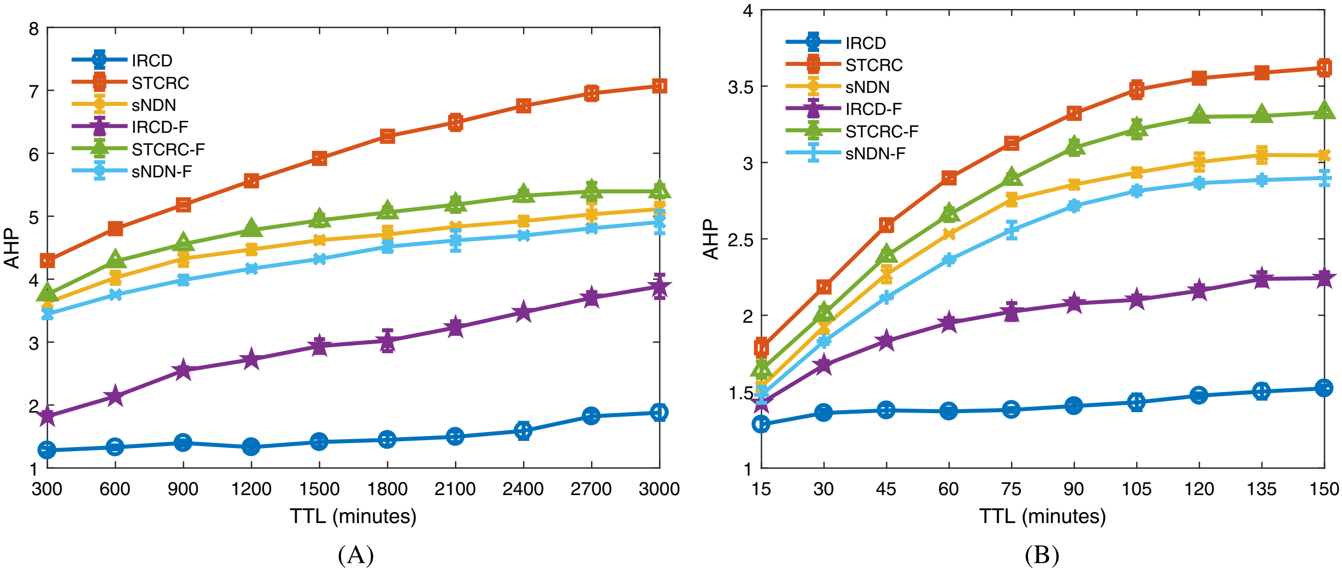

FIGURE 8 (AHP) for the six schemes over Cambridge trace and Synthetic trace A, # Cambridge trace; B, # Synthetic trace. IRCD, information-centric routing based on community detection; sNDN, social-aware named data networking; STCRC, social-tie–based content retrieval among communities; TTL, time to live

nodes. These communities accurate the routing process for different kinds of packets and thus improve PDR. In contrast, STCRC just uses one single kind of community to route both interest and data packets, and thus may make packets be routed by the misleading communities, while sNDN routes data packets based on the centralities. In addition, STCRC uses communities among nodes to forward packets, whereas sNDN only leverages friend circles. Since communities in STCRC are detected among the network-wide nodes, STCRC has the higher PDR than sNDN, which forwards packets based on the friend circles that are only formed among the encountered nodes.

When comparing IRCD-F, STCRC-F, and sNDN-F, we observe that the relationships among them are similar to those among IRCD, STCRC, and sNDN. Hence, the packets forwarding without in-network caching are proved to be effective for IRCD on PDR, and the same conclusions can be observed in AHP, ADL, and NOH. Moreover, when comparing IRCD and IRCD-F, we observe that IRCD has higher PDR than IRCD-F. The proposed in-network caching scheme keeps the contents that are requested by interest community members frequently and deletes the redundant contents in the same social community. Therefore, the cached contents are distributed in the network reasonable to respond to the forthcoming interest requests with high PDR (similar to AHP, ADL, and NOH). When comparing STCRC and STCRC-F, we observe that STCRC-F has higher PDR than STCRC because the in-network caching helps improve the routing efficiency of STCRC-F (similar to AHP, ADL, and NOH). It is also the same for sNDN.

The AHP for IRCD, STCRC, sNDN, IRCD-F, STCRC-F, and sNDN-F under different TTL over Cambridge trace and the synthetic trace are shown in Figure 8. For IRCD, STCRC, and sNDN, we observe that IRCD has the lowest AHP, followed by sNDN and STCRC. Benefiting from the in-network caching, IRCD keeps AHP between one to two. For STCRC, it has a drawback that it needs to forward interest packets to the nodes with high centralities in the community to know which node has the content. If no one in the community has the requested content, the interest packet should be forwarded to the border node and further to be forwarded to other community. Such principles make STCRC has worse AHP than sNDN.

The ADL for IRCD, STCRC, sNDN, IRCD-F, STCRC-F, and sNDN-F under different TTL over the Cambridge trace and the synthetic trace are shown in Figure 9. For IRCD, STCRC, and sNDN, we observe that IRCD has the lowest ADL, followed by sNDN and STCRC. For IRCD, its routing mechanism works based on the communities, thus can accelerate packet delivery. Moreover, IRCD detects interest communities firstly when routing interest packets and detects social communities firstly when routing data packets. The appropriate communities for the different packets also quicken the deliveries of interest packets and data packets. However, sNDN only forms friend circles by the socially close encounter nodes. The friend circles do not effectively utilize clustering properties among nodes, and thus slow down the packet delivery. Although STCRC is also based on communities among nodes, for the same reason in AHP, the unreasonable forwarding of STCRC makes packets delay worse than IRCD and sNDN.

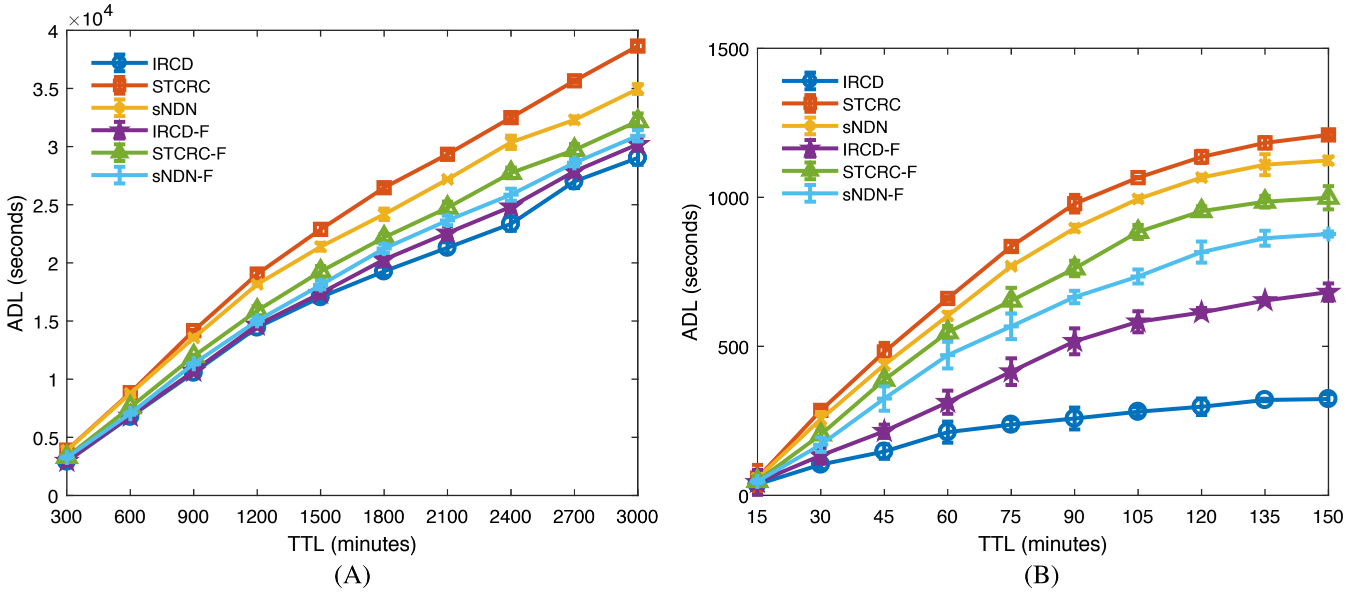

FIGURE 9 Average DeLay (ADL) for the six schemes over Cambridge trace and synthetic trace A, # Cambridge trace; B, # Synthetic trace. IRCD, information-centric routing based on community detection; sNDN, social-aware named data networking; STCRC, social-tie–based content retrieval among communities; TTL, time to live

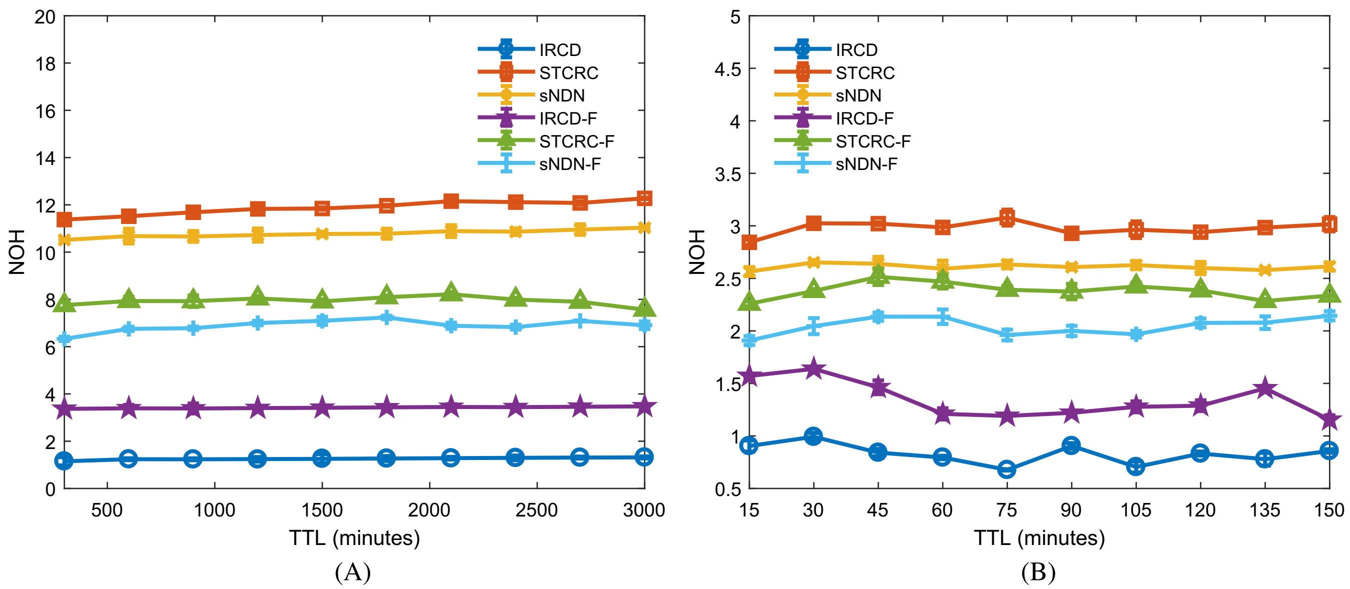

FIGURE 10 Network OverHead (NOH) for the six schemes over Cambridge trace and synthetic trace A, # Cambridge trace; B, # Synthetic trace. IRCD, information-centric routing based on community detection; sNDN, social-aware named data networking; STCRC, social-tie–based content retrieval among communities; TTL, time to live

The NOH for IRCD, STCRC, sNDN, IRCD-F, STCRC-F, and sNDN-F under different TTL over the Cambridge trace and the synthetic trace are shown in Figure 10. For IRCD, STCRC, and sNDN, we observe that IRCD has the lowest NOH, followed by sNDN and STCRC. Since the in-network caching and accurate community-based forwarding reduce the useless forwarding of packets, IRCD has the lowest NOH. For STCRC, packets have experienced forwarding process with a big number of hops, that is, firstly, the cluster head in the local community; then the border node and the cluster head in another community; and finally, the content provider. Such a process makes the packet cannot be easily delivered when its TTL expires. Therefore, such many times of packets forwarding not contributed to the final delivery. Although STCRC has higher PDR than sNDN, such many forwarding times lead STCRC to have higher NOH than sNDN.

7 | CONCLUSIONS

We have proposed an information-centric routing scheme in MSN. Although social-based routing is beneficial to effectively deliver packets to the destinations in the mobile environment, it is not easy to exploit such social information. We have described an interest community aware forwarding scheme to forward interest packets, in which the interest information is mined from the content requests attached to mobile users' interest packets. When returning data packets to the interest requesters, a social community aware forwarding scheme is devised to forward data packets, and the content that is requested with a high possibility in the future is cached to reduce response time.

In the future, we will equip IRCD with mobile users' cell phones to verify its efficiency in satisfying daily content requests of users. Moreover, to avoid the privacy disclosure of users, the routing mechanism should prevent user privacy not to be obtained by malicious users. It needs to maintain the good performance of routing and, at the same time, make the user's private information not disclosed to others.

ACKNOWLEDGMENTS

This work is supported by the Major International (Regional) Joint Research Project of NSFC under grant 71620107003, by the National Natural Science Foundation of China under grants 61572123 and 61502092, by the National Science Foundation for Distinguished Young Scholars of China under grant 71325002, by the MoE and ChinaMobile Joint Research Fund under grant MCM20160201, and by the Program for Liaoning Innovative Research Term in University under grant LT2016007.

ORCID

Junling Shi https://orcid.org/0000-0002-7275-9783
Jianhui Lv https://orcid.org/0000-0003-0884-6601

REFERENCES

1. Hu X, Chu THS, Leung VCM, Ngai EC, Kruchten P, Chan HC. A survey on mobile social networks: applications, platforms, system architectures, and future research directions. *IEEE Commun Surv Tutor*. 2015;17(3):1557-1581.
2. Conti M, Boldrini C, Kanhere SS, et al. From MANET to people-centric networking: milestones and open research challenges. *Comput Commun*. 2015;71:1-21.
3. Vastardis N, Yang K. Mobile social networks: architectures, social properties, and key research challenges. *IEEE Commun Surv Tutor*. 2013;15(3):1355-1371.
4. Bari MF, Chowdhury SR, Ahmed R, Boutaba R, Mathieu B. A survey of naming and routing in information-centric networks. *IEEE Commun Mag*. 2012;50(12):44-53.
5. Zhang S, Wu J, Qian Z, Lu S. MobiCache: cellular traffic offloading leveraging cooperative caching in mobile social networks. *Comput Netw*. 2015;83:184-198.
6. Newman MEJ, Girvan M. Finding and evaluating community structure in networks. *Phys Rev E*. 2004;69(2):1-16.
7. Hui P, Crowcroft J, Yoneki E. BUBBLE Rap: social-based forwarding in delay-tolerant networks. *IEEE Trans Mob Comput*. 2011;10(11):1576-1589.
8. Li Z, Wang C, Yang S, Jiang C, Li X. LASS: local-activity and social-similarity based data forwarding in mobile social networks. *IEEE Trans Parallel Distributed Syst*. 2014;26(1):174-184.
9. Jacobson V, Smetters DK, Thornton JD, Plass MF, Briggs NH, Braynard RL. Networking named content. *Commun ACM*. 2012;55(1):117-124.
10. Bulut E, Szymanski BK. Exploiting friendship relations for efficient routing in mobile social networks. *IEEE Trans Parallel Distributed Syst*. 2012;23(12):2254-2265.
11. Chen X, Proulx B, Gong XW, Zhang J. Exploiting social ties for cooperative D2D communications: a mobile social networking case. *IEEE/ACM Trans Netw*. 2015;23(5):1471-1484.
12. Xiao MJ, Wu J, Huang LS. Home-based zero-knowledge multi-copy routing in mobile social networks. *IEEE Trans Parallel Distributed Syst*. 2015;26(5):1238-1250.
13. Pholpabu P, Yang LL. Social contact probability assisted routing protocol for mobile social networks. In: Proceedings of IEEE Vehicular Technology Conference; 2015; Seoul, South Korea.
14. Pholpabu P, Yang LL. A mutual-community-aware routing protocol for mobile social networks. In: Proceedings of IEEE Global Communications Conference; 2014; Austin, TX.
15. Xiao M, Wu J, Huang L. Community-aware opportunistic routing in mobile social networks. *IEEE Trans Comput*. 2014;63(7):1682-1695.

16. Gupta R, Krishnamurthi N, Wang U, Tamminedi T, Gerla M. Routing in mobile ad-hoc networks using social tie strengths and mobility plans. In: Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC); 2017; San Francisco, CA.
17. Pu L, Chen X, Xu J, Fu X. sNDN: a social-aware named data framework for cooperative content retrieval via D2D communications. In: Proceedings of the 10th International Workshop on Mobility in the Evolving Internet Architecture; 2015; Paris, France.
18. Nazir F, Ma JA, Seneviratne A. Time critical content delivery using predictable patterns in mobile social networks. In: Proceedings of International Conference on Computational Science and Engineering; 2009; Vancouver, Canada.
19. Anastasiades C, Weber J, Braun T. Dynamic unicast: information-centric multi-hop routing for mobile ad-hoc networks. *Comput Netw*. 2016;107:208-219.
20. Kuang J, Yu SZ. CSAR: a content-scent based architecture for information-centric mobile ad hoc networks. *Comput Commun*. 2015;71:84-96.
21. Lu Y, Gerla M, Le T, Rabsatt V, Kalantarian H. Community aware content retrieval in disruption-tolerant networks. In: Proceedings of the 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET); 2014; Piran, Slovenia.
22. Xu Z, Su Z, Xu Q, et al. Delivering mobile social content with selective agent and relay nodes in content centric networks. *Peer-to-Peer Netw Appl*. 2017;10(2):296-304.
23. Lv JH, Wang XW, Huang M, Shi J, Li K, Li J. RISC: ICN routing mechanism incorporating SDN and community division. *Comput Netw*. 2017;123:88-103.
24. Taxicab geometry. https://en.wikipedia.org/wiki/Taxicab_geometry
25. Keranen A, Ott J, Karkkainen T. The ONE simulator for DTN protocol evaluation. In: Proceedings of the 2nd International Conference on Simulation Tools and Techniques; 2009; Rome, Italy.
26. Haggleproject.org. http://www.haggleproject.org
27. Yu H, Zheng D, Zhao BY, Zheng W. Understanding user behavior in large-scale video-on-demand systems. *ACM SIGOPS Oper Syst Rev*. 2006;40(4):333-344.

How to cite this article: Shi J, Wang X, Huang M, Lv J, Li K, Li J. Information-centric routing in MSN based on community detection. *Trans Emerging Tel Tech*. 2019;30:e3554. https://doi.org/10.1002/ett.3554

APPENDIX

THEOREM 1 PROOF

Proof of Theorem 1. Before verifying the theorem, we do the following assumptions at first:

- Each node has the limited cache space and can only cache y contents at most.
- Each node receives different contents with different constant probability.
- If a node has higher received requested probability for a content, it receives more requests for this content.

Suppose that the average time interval of received requests by R_i is Iv, and that R_i receives a content c with probability P_i^c, $P_i^c \neq 0$. Let Iv_i^c represent the average time interval of the received requests by R_i for c, and we have

$$Iv_i^c = \frac{Iv}{P_i^c}. \tag{A1}$$

Let $Rt_{i,c}^T$ represent the received requested times for c by R_i in time T, and we have

$$Rt_{i,c}^T = \frac{T \cdot P_i^c}{Iv}. \tag{A2}$$

Suppose that R_i generates requests with a uniform time internal iv, and requests c with probability p_i^c. Let $rt_{i,c}^T$ represent the generated request times by R_i for c in T, and we have

$$rt_{i,c}^T = \frac{T \cdot p_i^c}{iv}. \tag{A3}$$

When R_i receives c at T, according to Equations (25), (24), (8), and (4), we can calculate the caching time period for c by R_i as follows.

$$\begin{aligned} T_i^c &= T_0 \cdot \sum_{k=1}^{w} ci_k^g \\ &= T_0 \cdot \sum_{k=1}^{w}\sum_{j=1}^{n} rt_{k,c_{gj}}^T \\ &= T_0 \cdot \sum_{k=1}^{w}\sum_{j=1}^{n} \frac{T \cdot p_k^{c_{gj}}}{iv}. \end{aligned} \tag{A4}$$

To respond the same request for c, T_i^c should be longer than Iv_i^c. Let x represent the value of T_i^c divided by Iv_i^c, and we have

$$\begin{aligned} x &= \frac{T_0 \cdot \sum_{k=1}^{w} \sum_{j=1}^{n} \frac{T \cdot p_k^{c_{gj}}}{iv}}{\frac{Iv}{P_i^c}} \\ &= \frac{T \cdot P_i^c}{Iv} \cdot \sum_{k=1}^{w}\sum_{j=1}^{n} \frac{T_0 \cdot p_k^{c_{gj}}}{iv} \\ &= Rt_{i,c}^T \cdot \sum_{k=1}^{w}\sum_{j=1}^{n} rt_{k,c_{gj}}^{T_0}. \end{aligned} \tag{A5}$$

Suppose that R_i at least has received a request for c once in T, and at least one of its interest community members have requested for the content, which belongs to the category of c once in T_0, that is,

$$Rt_{i,c}^T \geq 1 \tag{A6}$$

$$\sum_{k=1}^{w}\sum_{j=1}^{r} rt_{k,c_{gj}}^{T_0} \geq 1. \tag{A7}$$

Thus, $x \geq 1$, that is,

$$T_i^c \geq Iv_i^c. \tag{A8}$$

As a result, T_i^c is long enough to respond to the same request for c.

However, according to the replacement policy of PCP, when the cache space of R_i is full, the content which has the least historical requested times by others is deleted. Therefore, to guarantee c can be cached for the full T_i^c, we should analyze the condition for its full time being cached.

We sort the received requested probabilities for different contents of R_i in descending order. Let P_i^{y-th} represent the yth probability. If P_i^c is not less than P_i^{y-th}, then there are at most $y-1$ contents that have higher requested probabilities than c. According to the third assumption, we can infer that there are at most $y - 1$ contents have higher historical requested times than c, that is, at most $y - 1$ contents will be cached by R_i after c. Therefore, c can be cached for the full T_i^c, and R_i can respond to the request for c. As a result, the effective cache condition of PCP is

$$P_i^c \geq P_i^{y-th}, \tag{A9}$$

which means that R_i can respond to the content requests with received requested probabilities and is not less than P_i^{y-th}. Let rp_{PCP} represent the response probability for a request of PCP, and we have

$$rp_{\text{PCP}} = \sum_{q=1}^{y} P_i^{q-th}. \tag{A10}$$

Since the highest response probability for any request by y contents are the sum of the top y requested probabilities of all the contents, let rp_{MAX} represent such highest response probability, and we have

$$rp_{\text{MAX}} = \sum_{q=1}^{y} P_i^{q-th}. \tag{A11}$$

According to (A10) and (A11), we can observe that $rp_{\text{PCP}} = rp_{\text{MAX}}$, PCP can achieve the highest response probability for content requests.

This proof is completed. □