Received March 14, 2019, accepted April 3, 2019, date of publication April 11, 2019, date of current version April 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2910117

Query Learning-Based Scheme for Pertinent Resource Lookup in Mobile P2P Networks

TAOUFIK YEFERNY[1], SOFIAN HAMAD[1], AND SADOK BEN YAHIA[2]
[1]College of Science, Northern Border University, Arar 73222, Saudi Arabia
[2]Department of Software Science, Tallinn University of Technology, 12618 Tallinn, Estonia

Corresponding author: Taoufik Yeferny (yeferny.taoufik@gmail.com)

This work was supported by the Deanship of Scientific Research at Northern Border University, Arar, Saudi Arabia, under Grant SCI-2018-3-9-F-7693.

ABSTRACT P2P networking has grasped an increasing interest worldwide among both researchers and computer networking professionals. As a witness, several P2P applications mainly used for file sharing over the Internet have been proposed. Considering the great success of mobile devices in recent years, P2P applications have also been deployed over mobile networks such as mobile ad-hoc networks (MANETs). However, the mismatch between the P2P overlay and the MANET underlay topologies makes the resources lookup mechanism in mobile P2P applications very difficult. Therefore, this downside is the main hindrance to the deployment of such applications over MANETs. To overcome the mismatch issue, we propose in this paper RLSM-P2P a cross-layer resource lookup scheme for Mobile P2P applications. The main thrust of RLSM-P2P consists of building an efficient unstructured P2P overlay that closely matches the underlay physical network and swiftly adapts to its volatility and dynamicity by considering different MANET constraints. Furthermore, RLSM-P2P relies on a query learning resource lookup mechanism for locating pertinent resources to user queries. The performed experiments show that RLSM-P2P outperforms its competitors in terms of effectiveness and efficiency.

INDEX TERMS P2P, MANET, query learning, resource lookup.

I. INTRODUCTION

P2P networks are virtual networks, in which peers are connected through logical links for building an overlay network on top of an existing physical network [1]. In pure P2P networks, every peer could join and leave the overlay at any time and directly exchanges resources, such as documents, video, images and services with each other without the coordination of any central entity. P2P networking has grasped an increasing interest worldwide among both researchers and computer networking professionals. Therefore, several P2P applications have been proposed. They are mainly used for file sharing (e.g., Gnutella, BitTorrent, to cite a few) and firstly deployed on top of an existing network infrastructure such as Internet.

Considering the great success of mobile devices in recent years on one hand and the need of users to exchange resources "anywhere, anytime" on the other hand, P2P applications have been deployed over mobile networks such as MANETs. They are commonly named mobile P2P applications.

In MANET peers are physically connected through wireless links without an existing network infrastructure or any centralized entity. They have limited battery energy and are free to move anywhere at any time, leading to a high dynamic network topology. It is worth mentioning that the P2P overlay consists on logical links between peers that don't match the physical underlay network. The mismatch between the P2P overlay and the physical underlay network does not have a significant impact on the performance of P2P applications deployed over Internet. However, it makes the resource lookup mechanism in mobile P2P applications very difficult. Therefore, it is a key hindrance to the deployment of such applications over MANETs. Indeed, in a pure unstructured P2P network, every overlay hop required by the resource lookup protocol at the application layer could result in a costly flooding-based route discovery in the MANET underlay network. Doing so undoubtedly leads to unnecessary network traffic, network congestion and delay issues. To overcome these drawbacks, we introduce, in this

The associate editor coordinating the review of this manuscript and approving it for publication was Mahmoud Barhamgi.

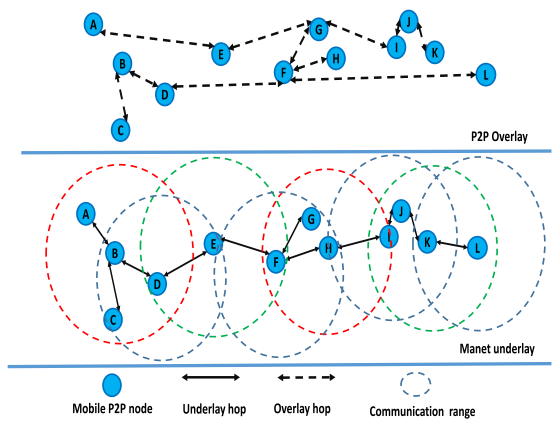

FIGURE 1. P2P overlay over MANET underlay.

paper, a cross-layer Resource Lookup Scheme for Mobile P2P applications, called RLSM-P2P. The latter has two main components, namely "P2P overlay construction component" and "resource lookup component". The first one builds an efficient unstructured P2P overlay that closely matches the underlay physical network and rapidly adapts to its volatility and dynamicity by considering different MANET constraints. The goal is to make virtual connections between peers which have stable links, less load and enough battery energy. This optimization likely-matches between overlay and underlay paths which consequently reduces the search delay, communication redundancy and unnecessary network traffic. The second component enables resource lookup to user search queries. We introduce here a new mechanism as alternative to random walk [2], which is widely used to locate resources in mobile P2P applications. To this end, we rely on the historical information of user queries to select the relevant neighbors that could provide pertinent resources for forthcoming user queries [3].

The remainder of this paper is organized as follows. In Section II, we present the background of our work. Section III details the related work. In Section IV, we thoroughly describe the main idea of our scheme, before presenting the simulation settings and the evaluation of the proposed scheme in Section V. The last section concludes this work and pins down future directions.

II. BACKGROUND

P2P applications can be divided according to their overlay topologies and the related resource lookup mechanisms into two broad categories: structured and unstructured applications [4].

A. STRUCTURED P2P APPLICATIONS

Structured P2P networks maintain a defined structure (i.e., ring) among participating peers and place objects based on logical peer identifiers, which are computed by a specific hashing function. Figure 2 (D) illustrates an example of a structured P2P network. The search mechanism for an object in this type of applications is simple and efficient. However, it is very difficult to dynamically maintain the overlay structure, since peers could join and leave the network at any time.

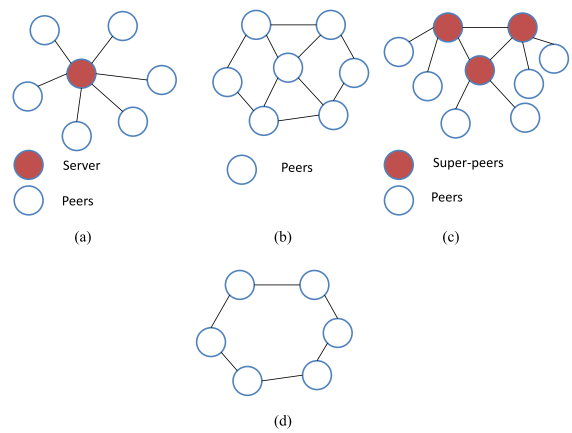

FIGURE 2. P2P architectures at a glance. (a) Centralized architecture. (b) Pure P2P architecture. (c) Hybrid P2P architecture. (d) Structured P2P.

B. UNSTRUCTURED P2P APPLICATIONS

In unstructured P2P networks, there is no fixed topology structure. Hence, peers randomly organize themselves and locally manage their own content. Although, unstructured P2P networks are supposed to work in a completely decentralized manner, in reality there are different degrees of centralization. In fact, as shown in figures 2 (A), 2 (B) and 2 (C), three architectures of unstructured P2P networks can be identified depending on the degree of centralization [5].

1) CENTRALIZED P2P ARCHITECTURE

In this architecture, all peers are connected to a central server which is responsible for indexing all shared resources on the network (c.f. figure 2 (A)). In general, the server updates its index whenever a peer joins or leaves the network. The search mechanism is very simple since any search request goes through this server. Indeed, user queries are first sent to the central server. In turn, the server will search, in its index, for relevant peers holding pertinent resources for the query and then returns a query reply to the sender. The latter can establish direct connections with the relevant peers. The advantage of centralized indexing lies in the effectiveness and efficiency of the search process. However, this architecture also has two main drawbacks. On the one hand, the system may crash when the server goes down or stops functioning properly. On the other hand, the indexation of an overwhelming number of resources makes the management of the central index extremely difficult leading to a raise of scalability issue.

2) PURE P2P ARCHITECTURE

In fully decentralized P2P networks, also called "pure P2P networks" (c.f. figure 2 (B)), there is a complete symmetry at the role of all peers, all peers are equal and play the same role (e.g., client and server at the same time). Indeed, the absence of a central server allows fully decentralized applications to be more fault-tolerant. In the latter, each peer locally manages

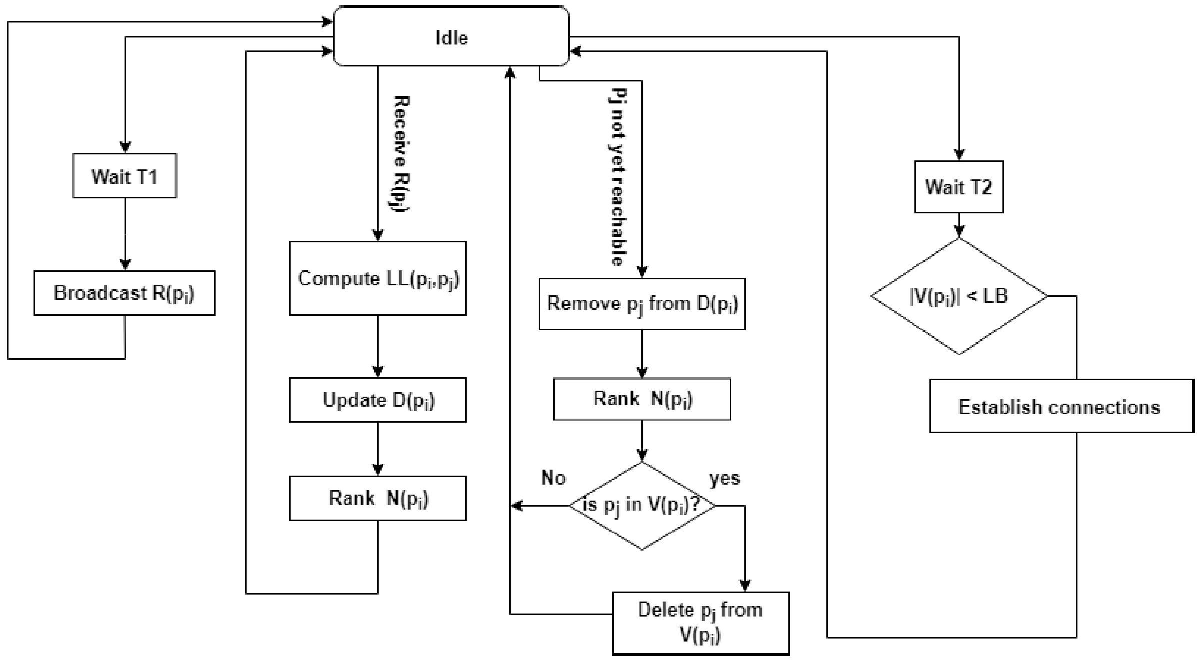

FIGURE 3. **Flowchart of the P2P overlay building algorithm.**

its shared resources and it does not have any knowledge of the content of other peers. Unlike centralized systems, the search mechanism in pure P2P is based on flooding techniques. Although this architecture fully supports the autonomy of peers, the search mechanism still remains a challenging issue.

3) HYBRID P2P ARCHITECTURE

In partially decentralized P2P networks (also called hybrid P2P networks) there are two types of peers: super-peers and peers (c.f. figure 2 (C)). The former are connected together to build a pure P2P network and ordinary peers are connected to super-peers. Indeed, a super-peer could be considered as a server that indexes all resources shared by its associated ordinary peers. User queries are first sent to the super-peer, which in turn will reply to the query whenever it finds pertinent resources in its index. Otherwise, the query will be flooded in the super-peers network.

C. DISCUSSION

Existing research studies have focused on the deployment of unstructured P2P networks over MANETs, since they share the same properties (i.e., self-organizing, dynamicity, to cite but a few). This matching facilitates the interaction between the P2P overlay and the MANET underlay. However, unstructured P2P networks are not suitable for MANET because they rely on a well-defined overlay structure, which is different from that of MANET underlay network. Furthermore, this structure must be systemically maintained through a very costly protocol.

Although the deployment of unstructured P2P applications over MANETs is a promising idea, the resource lookup mechanism in this kind of applications still remains a challenging issue. To this end, three approaches have been proposed:

1) **Layered design**: it applies existing search mechanisms proposed for unstructured P2P applications over fixed networks (e.g, Internet) on a top of MANET underlay [6], [7]. However, these latter achieve poor performances over MANET underlay, since every overlay hop required by the resource lookup protocol at the application layer could result in a costly flooding-based route discovery in the MANET underlay network leading to unnecessary network traffic, network congestion as well as delay issues.
2) **Cross-layered design**: it encourages communication between the P2P overlay and the MANET underlay. Indeed, in this design, the P2P overlay is optimized with respect to MANET constraints and it closely matches the MANET underlay [6], [7]. Furthermore, to avoid issues related to flooding-based route discovery in the MANET underlay network, the resource lookup mechanism, at the application layer, communicates to the underlay protocol a stable path between the source and destination peers.
3) **Integrated design**: it combines the resource lookup mechanism with the MANET network routing protocol at the network layer [6], [7].

In [6] it was pointed out that the cross-layer design is the most suitable one because it establishes communication between the overlay and the MANET underlay without changing the conventional system architecture. In the following section, we present some existing cross-layer resource lookup mechanisms over MANETs.

III. RELATED WORK

Integrated resource lookup mechanisms over unstructured p2p network can be roughly categorized as:

A. NON OVERLAY-BASED APPROACHES

Theses approaches do not construct a specific optimized p2p overlay topology. They simply consider the physical neighbors at MANET underlay as virtual neighbors at the p2p overlay. Hence, the MANET underlay topology is maintained at the p2p overlay. Interestingly enough, existing approaches rely on existing MANET underlay protocols (e.g., DSR [8], AODV [9], DSDV [10], to name a few.) to route search queries and to exchange resources.

Different search mechanisms have been proposed for locating pertinent resources. The simplest one is pure flooding, in which the requesting peer sets a time-to-live counter (TTL) to a certain value then broadcasts the search query to an all its neighbors that continue to forward it in the same way to their neighbors until the TTL counter is decremented to 0. Whenever, a peer sharing pertinent resources, is located, then it replies by a query-hit message. This message is routed back to the requesting peer through the reverse path of the query message. Pure flooding produces heavy network traffic leading to 'broadcast storm' issue. Therefore, several mechanisms, such as controlled flooding, random walk [2], k-walker [11] and gossiping [12] were proposed as improvement of pure flooding. All of these latter rely on the query flooding. Nevertheless, each approach proposes a specific mechanism for choosing the neighbors to forward the query to. Controlled flooding forwards the query to k randomly chosen neighbors. The random walk approach forwards the query to one randomly selected neighbor. The query goes on until the target resource is located. This approach reduces the number of duplicated queries, however as a down side it results in long search delay. To overcome this shortage the k-walker approach was proposed. Within k-walker, the requesting peer forwards the query to k randomly selected neighbors instead of one neighbor. Each of the selected k neighbors (k walkers) continues to forward the query to a one random neighbor and so on until the target resource is located. Gossiping is an interesting mechanism, which forwards the query to k neighbors chosen with respect to the likelihood that they hold pertinent resources for the query [12].

In addition to flooding based approaches, other approaches based on content dissemination have also been proposed [7]. In these latter, each peer periodically broadcasts meta-data about its shared resources to its neighborhood. Whenever a peer is interested in a shared resource, then it sends the query to the source peer.

B. OVERLAY-BASED APPROACHES

Overlay-based approaches have attempted to avoid or reduce the effect of the mismatch between the P2P overlay and the underlay topologies. Their main objective was to build an efficient and optimized P2P overlay that closely matches the MANET underlay. Furthermore, the existing approaches rely on controlled flooding, random walk, k-walker or gossiping techniques for locating resources in the P2P overlay. E-UnP2P [13] built an efficient overlay avoiding redundant links and transmissions while ensuring connectivity among peers. It introduces a root-peer in the P2P network connecting all the other peers. Each peer maintains connection with other closer peers such that it can reach the root-peer. Seddiki and Benchaïba [6] proposed a P2P lookup mechanism over MANETs called 2P-Lookup. To reduce the mismatch between the P2P overlay and the MANET underly topologies, 2P-Lookup built the P2P overlay based on the physical proximity of peers. Indeed, each peer was connected with the k nearest neighbors. This means that overlay neighbors are the physically closest peers. In addition, a resource popularity-biased random walk was suggested to efficiently guide the search query and control its propagation. Mawji *et al.* [14] proposed an adaptive topology control algorithm designed specifically for P2P overlays running on top of mobile ad hoc networks. Their goal was to reduce the stretch factor between the overlay and the underlay networks and eventually minimizing energy consumption. Leitao *et al.* [15] introduced the X-BOT protocol. The latter relies on some features, such as node degree and connectivity, to build an unstructured P2P overlay.

IV. PROPOSED SCHEME

In this section, we thoroughly describe our RLSM-P2P scheme. It is worth noting that our goals are twofold:

1) To introduce a fully distributed scheme for building an efficient P2P overlay closely that matches the MANET underlay;
2) To propose a resource lookup mechanism that swiftly locates "relevant" peers sharing pertinent resources to user queries.

In the following, we first define the main concepts used in our scheme before going through a detailed description of the p2p overlay construction (subsection IV-A) and the resource lookup (subsection IV-C) mechanisms.

Definition 1 (Peer): In our approach, a mobile peer p_i is defined as 6-tuple:
$p = \prec Id(p_i), R(p_i), Q(p_i), D(p_i), N(p_i), V(p_i) \succ$ where

- $Id(p_i)$: is the p_i's identifier;
- $R(p_i)$: is the p_i's representative vector;
- $Q(p_i)$: is the set of queries sent or received by p_i;
- $D(p_i)$: is the p_i's decision matrix;
- $N(p_i)$: is the MANET physical neighbors set of p_i (e.g., set of peers inside the communication range of p_i);
- $V(p_i)$: is a subset of $N(p_i)$ representing the virtual neighbors of p_i in the P2P overlay.

Definition 2 (Representative Vector): Each peer p_i has a three-dimensional representative vector
$R(p_i) = (CPU, BL, Load)$, where:

- CPU: is the processing power of peer p_i;

- BL: is the estimated remaining battery lifetime of p_i at a time t. The peer p_i computes its BL using the formula introduced in [16];
- Load: is the queue utilization of p_i at a time t [12]. It stands within the]0, 1] interval. Hence, a high value indicates that the queue is full then the peer is overloaded.

Definition 3 (Decision Matrix): Each peer p_i holds a decision matrix $D(p_i) = (x_{jk})_{N\times C}$ ($N = |N(p_i)|$, $C = 4$), where rows represent the p_i's physical neighbors set $N(p_i)$, columns represent p_i's ranking criteria. Hence, x_{jk} is the score of the neighbor p_j with respect to the p_i's criterion c_k. In our scheme, we consider four ranking criteria:

- LL: is lifetime of the link between p_i and p_j;
- CPU: is the processing power of p_j;
- BL: is the estimated remaining battery lifetime of p_j;
- Load: is the queue utilization of p_j;

Example 1 (Decision Matrix): The following matrix illustrates an example of a decision matrix of the peer p_i.

$$\mathbf{D(p_i)} = \begin{array}{c} \\ \mathrm{p_1} \\ \mathrm{p_2} \\ \vdots \\ \mathrm{p_j} \\ \vdots \\ \mathrm{p_N} \end{array} \begin{array}{c} \begin{array}{cccc} c_1 = \mathrm{LL} & c_2 = \mathrm{CPU} & c_3 = \mathrm{BL} & c_4 = \mathrm{Load} \end{array} \\ \left(\begin{array}{cccc} \mathrm{x_{11}} & \mathrm{x_{12}} & \mathrm{x_{13}} & \mathrm{x_{14}} \\ \mathrm{x_{21}} & \mathrm{x_{22}} & \mathrm{x_{23}} & \mathrm{x_{24}} \\ \vdots & \vdots & & \vdots \\ \mathrm{x_{j1}} & \mathrm{x_{j2}} & \mathrm{x_{j3}} & \mathrm{x_{j4}} \\ \vdots & \vdots & & \vdots \\ \mathrm{x_{N1}} & \mathrm{x_{N2}} & \mathrm{x_{N3}} & \mathrm{x_{N4}} \end{array} \right) \end{array}$$

A. P2P OVERLAY CONSTRUCTION

In the following, we first define the problem statement, and after that we introduce our algorithm for building the P2P overlay over MANET.

1) PROBLEM STATEMENT

As seen before, the P2P overlay has an important impact on the performance of P2P mobile applications. Indeed, a random overlay will undoubtedly lead to a redundant traffic, network congestion and delay issues. To avoid these drawbacks, RLSM-P2P builds an efficient P2P overlay that matches the MANET underlay and swiftly adapts to its volatility and dynamicity by considering different MANET constraints. To this end, RLSM-P2P builds a P2P overlay in which each peer p_i establishes connections with at least LB (Lower Bound) best neighbors. Each selected neighbor p_j must meet the following requirements:

- **R1 (Link Lifetime):** The link lifetime (LL) between p_i and p_j should be as high as possible to ensure the link stability. To estimate this lifetime, we rely on the affinity function defined in the RABR network protocol [17]. This function estimates at an instant t the time taken by p_j to move out of the range of p_i.
- **R2 (CPU power):** The CPU processing power of p_j should be as high as possible to quickly answers p_i's queries;
- **R4 (Battery Lifetime):** p_j should have enough battery lifetime (BL) to avoid the network partitioning issue;
- **R3 (Load):** p_j should not be fully overloaded to avoid the network congestion issue.

2) ALGORITHM FOR BUILDING THE P2P OVERLAY

The flowchart diagram Figure 3 depicts the main operations that RLSM-P2P runs to build and maintain the P2P overlay. To this end, each peer p_i, running RLSM-P2P, performs in parallel the three following processes:

1) **Broadcast** $R(p_i)$: Every $T1$ seconds (e.g., $T1$ is specified threshold), p_i broadcasts its representative vector $R(p_i)$ to inform its neighbors about its current situation (e.g., its CPU, BL and Load).
2) **Update routing indices**: p_i updates its routing indices $D(p_i)$, $N(p_i)$ and $V(p_i)$, whenever it receives a representative vector $R(p_j)$ from a neighbor p_j or it detects that a neighbor p_j is not yet reachable (i.e., move out of the range of p_i).
   - **Receive** $R(p_j)$: When p_i receives the representative vector $R(p_j)$ of the neighbor p_j, it firstly computes the link lifetime LL between p_i and p_j. Then, it updates its decision matrix $D(p_i)$ by adding p_j if it is a new neighbor or updates the associated row otherwise. After that, p_i ranks its set of neighbors $N(p_i)$, with respect to the ranking criteria LL, CPU, LB and Load, by using a multi-criteria ranking algorithm (c.f., subsection IV-B).
   - p_j **is not reachable**: Whenever a neighbor p_j is no more reachable, p_i removes it from its decision matrix $D(p_i)$ and then from its physical neighbors set $N(p_i)$. Thereafter, p_i ranks the elements of its physical neighbors set $N(p_i)$ then removes p_j from its virtual neighbors set $V(p_i)$ if p_j was considered as a virtual neighbor at the P2P overlay.
3) **Establish connections**: To build the P2P overlay, each peer p_i should establish virtual connections with the best k neighbors (e.g., $LB \leq k \leq UB$, where Lower Bound (LB) and Upper Bound (UB) are two specified thresholds that represent respectively the minimum number and the maximum number of connections per peer). As the virtual neighbors set $N(p_i)$ may change at any time, p_i runs periodically the "Establish connections" operation to maintain the P2P overlay. Indeed, p_i checks every $T2$ seconds (e.g., $T2$ is specified threshold) if the number of virtual neighbors is less than LB. In this case, it tries to make connections to the neighbors with the highest scores from $N(p_i)$. It is important to note that the set $N(p_i)$ is sorted according to the neighbors' scores.

B. NEIGHBORS RANKING ALGORITHM

To rank the neighbors set of a given peer, with respect to the ranking criteria: LL, CPU, LB and Load, we rely on the Multi-Criteria Decision Making (MCDM) theory. The latter is a branch of operation research models suitable for solving an issue featuring different decision criteria, multi-interests and perspectives, and conflicting objectives [18]. In the dedicated literature, there are dozens of methods used for solving MCDM problems. The TOPSIS method, developed by Hwang and Yoon [18], is kenned as one of the most popular methods among MCDM methods. This method attempts to choose the alternatives that simultaneously have the shortest Euclidean distance from the positive ideal solution and the furthest Euclidean distance from the negative ideal solution. The ideal solution is composed of all attainable best attribute values; the negative ideal solution is composed of all attainable worst attribute values. TOPSIS, therefore, provides a cardinal ranking for all the alternatives by taking the relative closeness to the ideal solution.

In our case, we adapt the TOPSIS [18] method to rank neighbors for each peer according to LL, CPU, LB and Load criteria as follows:

Formally, for each peer p_i, we have to rank $N = |N(p_i)|$ neighbors or alternatives evaluated on C equals to 4 criteria LL, CPU, LB and Load, noted c_k $(k = 1, \ldots, 4)$. It is worth mentioning that the input is the decision matrix $D(p_i) = (x_{jk})_{N\times C}$.

The ranking algorithm consists of five steps:

Step 1: Construction of a normalized decision matrix from the decision matrix: It transforms the decision matrix $D(p_i) = (x_{jk})_{N\times C}$ to a normalized decision matrix $D'(p_i) = (x'_{jk})_{N\times C}$ according to the following equation Eq.1.

$$x'_{jk} = \frac{x_{jk}}{\sqrt{\sum_{j=1}^{N}(x_{jk}^2)}} \quad \text{for } j = 1, \ldots, N;\ k = 1, \ldots, C \qquad (1)$$

Step 2: Construction of a weighted normalized decision matrix: It computes a weighted normalized decision matrix $D''(p_i)$ by multiplying the normalized decision matrix $D'(p_i)$ by the criteria's weight vector W as given in Eq.2.

$$x''_{jk} = w_k * x'_{jk} \quad \text{for } j = 1, \ldots, N;\ k = 1, \ldots, C \qquad (2)$$

where, w_k is the weight of the criterion c_k.

Step 3: Identification of positive ideal and negative ideal solutions from the weighted normalized decision matrix: It computes two artificial alternatives A^* and A^-:

A^* stands for the most preferable neighbor with the highest link lifetime (c_1), CPU power (c_2), battery lifetime (c_3) and with the lowest load (c_4). Formally,

$$A^* = (a_1^*, a_2^*, a_3^*, a_4^*) \qquad (3)$$

where:

$$a_k^* = \{max_{\forall j}(x''_{j,k})| \quad j = 1, \ldots, N;\ k = 1, \ldots, 3\}$$
$$a_4^* = min_{\forall j}(x''_{j,4})| \quad j = 1, \ldots, N$$

A^- stands for the least preferable neighbor with the lowest link lifetime (c_1), CPU power (c_2), battery lifetime (c_3) and with the highest load (c_4). Formally,

$$A^- = (a_1^-, a_2^-, a_3^-, a_4^-) \qquad (4)$$

where:

$$a_k^- = \{min_{\forall j}(x''_{j,k})| \quad j = 1, \ldots, N;\ k = 1, \ldots, 3\}$$
$$a_4^- = max_{\forall j}(x''_{j,4})| \quad j = 1, \ldots, N$$

a: STEP 4: COMPUTATION OF SEPARATION MEASURES FOR EACH ALTERNATIVE

The separation S_j^* between each alternative (e.g., neighbor p_j) and the positive ideal alternative A^* (e.g., most preferable neighbor) can be measured by the Euclidean distance as follows:

$$S_j^* = \sqrt{\sum_{k=1}^{C}(x''_{jk} - a_k^*)^2} \quad \text{for } j = 1, \ldots, N \qquad (5)$$

Similarly, the separation from the negative-ideal one A^- (i.e., the least preferable) can be derived using the following equation:

$$S_j^- = \sqrt{\sum_{k=1}^{C}(x''_{jk} - a_k^-)^2} \quad \text{for } j = 1, \ldots, N \qquad (6)$$

b: STEP 5: RANKING OF PREFERENCE ORDER

The neighbors set $N(p_i)$ can now be ranked by preference through the computation of the relative closeness to the ideal value solution C_i^* according the following equation:

$$C_j^* = \frac{S_j^-}{(S_j^* + S_j^-)} \quad \text{for } j = 1, \ldots, N \qquad (7)$$

C. RESOURCE LOOKUP MECHANISM

RLSM-P2P relies on gossiping-based resource lookup mechanism. Indeed, as illustrated in Algorithm 1, the requesting peer sets a time-to-live counter (TTL) to a certain value then forwards the search query q to its $k' \leq |V(p_i)|$ best neighbors. These latter continue to forward the query q in the same way to their neighbors until the TTL counter is decremented to 0. Whenever a peer sharing pertinent resources for q is located, it replies by a query-hit message. This message is routed back to p_i through the reverse path of q. Worth noting that the forwarder peer p_i computes a pertinence score for each of its neighbors $V(p_i)$ then select the best k' ones with the highest score to forward the query to (c.f. Algorithm 2). To compute the relevance of a neighbor p_j for the query q we exploit the set of queries $Q(p_i)$ sent by the forwarder peer p_i. We assume that neighbors which replied to past queries close to q could also provide answers for q. To this end, we define the following function that computes the relevance of p_j to q as follows:

$$sim_{p_i}(p_j, q) = \frac{\sum_{q_k \in Q_j(p_i)} Cos(q_k, q)}{|Q(p_i)|} \qquad (8)$$

where, $Q_j(p_i)$ is the set of query sent by p_i for which p_j provided answers and $Cos(q_k, q)$ is the cosine similarity between a query $q_k \in Q_j(p_i)$ and q. Worth noting that a query is a vector of query-terms.

Algorithm 1 Resource Lookup Algorithm Running on a Peer p_i

Input
q: *search query.*
k': *number of neighbors to forward the query to.*
TTL: *Time To Live threshold.*

$peers = \emptyset$ // list of pertinent neighbors.
if (p_i *is the query initiator*) **then**
  $q.seTTimeToLive(TTL)$
end if
if ($q.geTTimeToLive() > 0$) **then**
  $peers = getBestNeighbors(q, k')$
  $q.seTTimeToLive(TTL - 1)$
  $forwardQuery(q, peers)$
end if

Algorithm 2 Selection of the K' Best Neighbors

function GETPERTINENTNEIGHBORS(q,k')
  $peers = \emptyset$ // empty list
  for each $p_j \in V(p_i)$ **do**
    $psim = sim(p_j, q)$
    $peers.add(< p_j, psim >)$
  end for
  $peers.sort()$
  $return(peers.getElements(1, k'))$
end function

V. PERFORMANCE EVALUATION

In this section, we present the performance evaluation of our scheme RLSM-P2P versus CDP [16] and Gossiping-LB [12] protocols. We chose these latter as baseline since they rely on the gossiping approach for locating relevant peers as RLSM-P2P also does. In addition, they rely on the similar neighbors selection features as RLSM-P2P. Nevertheless, they do not construct an efficient P2P overlay like RLSM-P2P does.

A. SIMULATION SETTINGS

The network simulation is performed by the NS2 simulator [19]. The simulation settings are summarized in Table 1.

We note that we use the Music Dataset to perform different simulation scenarios. This dataset was developed under the RARE project [20] from a statistical analysis on Gnutella system data and the TREC collection. It is composed of 17000 documents and 700 queries distributed among 100 peers.

To set the weights of the features LL, CPU, BL and Load used in the ranking algorithm of the RLSM-P2P scheme, we rely on the Analytic Hierarchy Process (AHP) method [21]. In our case, we have used the pair wise comparison matrix Table 2.

TABLE 1. Simulation settings.

TTL	3
K'	3
Peers number	range from 25 to 100
MAC protocol	IEEE 802.11b
Area size	$500 \times 500\ m^2$
Node placement	Uniform distribution
Node mobility	Random way-point

TABLE 2. Pair wise comparison matrix.

	LL	CPU	BL	Load
LL	1	6	6	4
CPU	1/6	1	1	1/2
BL	1/6	1	1	1/3
Load	1/4	2	3	1

Therefore, the obtained weights are:

LL	CPU	BL	Load
0.616	0.095	0.088	0.201

B. EVALUATION METRICS

To assess the effectiveness and the efficiency of RLSM-P2P as well as that of the baseline protocols, we rely on the following metrics.

1) EFFECTIVENESS

We consider that a resource lookup scheme is effective whenever it ensures a high recall and success rate.

- **Recall**: It assesses the capacity of the system to retrieve all pertinent documents for a given query q. Formally, the recall metric is defined as follows:
$$Recall(q) = \frac{RRD(q)}{RLD(q)} \tag{9}$$
where, *RRD* stands for the number of relevant retrieved documents and *RLD* the number of relevant documents. The average recall is defined as follows:
$$Average\ Recall = \frac{\Sigma\ Recall(q)}{Number\ of\ Queries} \tag{10}$$
- **Success Rate**: Unlike the recall metric, the successes rate metric assess the capacity of the system to retrieve at least one document per query. The average success rate is defined as the ratio between the number of resolved queries to the total number of initiated resource-lookup queries.
$$Average\ Success\ Rate = \frac{RQ}{Number\ of\ Queries} \tag{11}$$
where *RQ* is the number of resolved queries.

2) EFFICIENCY

We consider that a resource lookup scheme is efficient whenever it flags out a minimum resource-lookup delay.

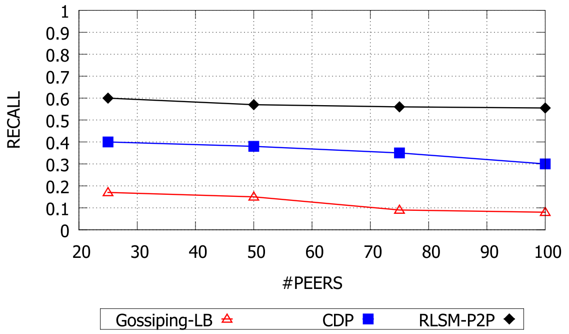

FIGURE 4. Average recall w.r.t the variation of the number of peers.

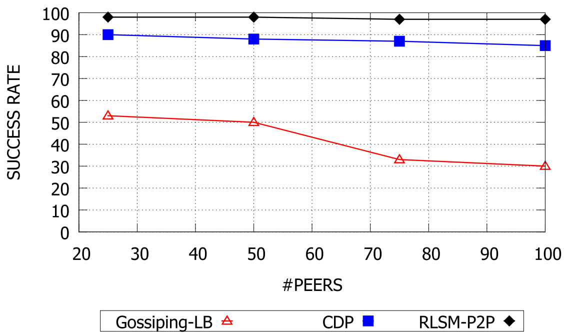

FIGURE 5. Average success rate w.r.t the variation of the number of peers.

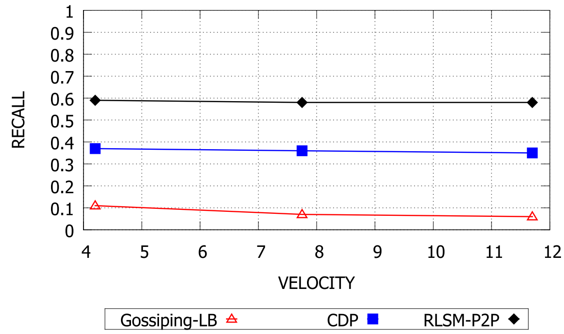

FIGURE 6. Average recall w.r.t the variation of peers velocity.

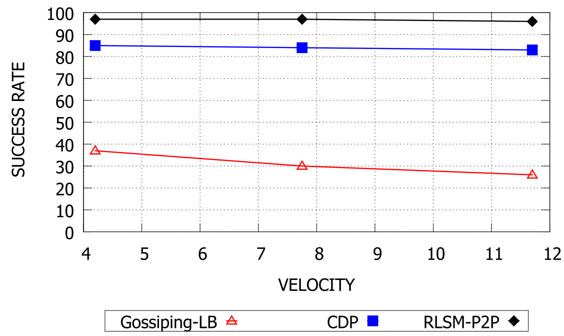

FIGURE 7. Average success rate w.r.t the variation peers velocity.

Formally, the resource-lookup delay of a query q is defined as:

$$Delay(q) = T_q - T_r \tag{12}$$

where T_q and T_r denote respectively the startup time of q and the time of receiving of the first reply for q. The average resource-lookup delay is defined as follows:

$$Average\ Delay = \frac{\Sigma\ Delay(q)}{Number\ of\ Queries} \tag{13}$$

C. RESULTS

In this section, we report the simulation results of RLSM-P2P versus CDP and Gossiping-LB.

1) RETRIEVAL EFFECTIVENESS

Figures 4 and 5 depict the evolution of the recall and the success rate of RLSM-P2P compared with CDP and Gossiping-LB w.r.t. the number of peers (i.e., we vary the number of peers from 25 to 100). We observe that the effectiveness of the three schemes goes down as far as the number of peers in the network goes up. Indeed, the higher the number of peers is, the lower the probability to locate relevant peers sharing the pertinent resources is. In addition, Figures 4 and 5 show that RLSM-P2P has the better recall and success rate according to the variation in the overlay size whenever compared with the CDP and the Gossiping-LB protocols. Indeed, it increases the overall recall of CDP and Gossiping-LB by around 57% and 346%, respectively. Furthermore, RLSM-P2P raises the overall success rate of CDP and Gossiping-LB by around 11% and 136%, respectively. These encouraging results are owed to the fact that RLSM-P2P routes the search query with respect to its content to the best neighbors regarding their answers for similar past queries, which leads to better target the adequate neighbors. In addition, it ensures a stable path between the requesting peer and the target one, since it periodically maintains the P2P overlay by considering the constraints of the MANET underlay, which increases the ratio of the query-hit messages that reach the requesting peer leading to an increase in the recall and the success rate.

Figures 6 and 7 show the evolution of the recall and the success rate of RLSM-P2P compared with CDP and Gossiping-LB w.r.t. the peers velocity (i.e., we set the number of peers to 60 and we vary the peers velocity from 4.2 to 11.7). We observe that the effectiveness of the three schemes goes down as far as the peers velocity increases. Indeed, by rising the peers velocity, links between peers become unstable leading to a decrease in the average recall and success rate. Moreover, we remark that RLSM-P2P is more effective than CDP and Gossiping-LB for all the variations of the peer velocity. In addition, RLSM-P2P is more resistant to peers velocity than the baseline protocols.

2) RETRIEVAL EFFICIENCY

Figure 8 depicts the evolution of the average resource lookup delay of RLSM-P2P, CDP and Gossiping-LB w.r.t. the number of peers (we vary the number of peers from 25 to 100). We note that the average resource lookup delay of the three schemes increases as far as the number of peers goes up.

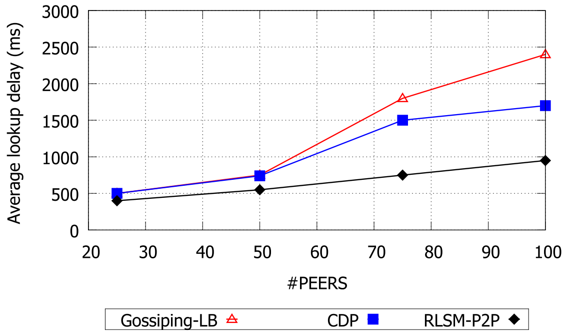

FIGURE 8. Average lookup delay w.r.t the variation of the number of peers.

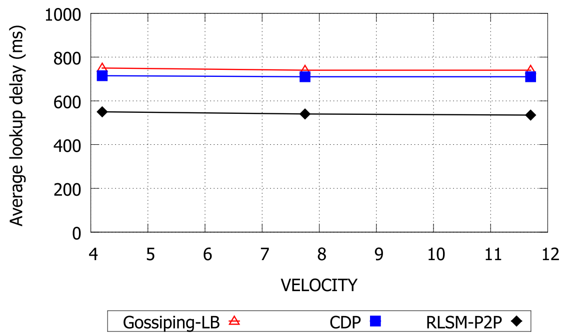

FIGURE 9. Average lookup delay w.r.t the variation of peers velocity.

This can simply be explained by the fact that by raising the number of peers in the network the overhead and the congestion rate increase, which undoubtedly leads to an increase in the resource lookup delay. In addition, Figure 8 shows that RLSM-P2P has the shortest average resource lookup delay than CDP and Gossiping-LB under different overlay sizes. Indeed, it decreases the overall delay of CDP and Gossiping-LB by around 40% and 51%, respectively. This is owed to the fact that RLSM-P2P relies on a P2P overlay in which overloaded peers are not preferred as virtual neighbors. In addition, by learning from past queries, RLSM-P2P quickly locates the target peers. Interestingly enough, by doing so, reduces the number of relays between the requesting peer and target one leading to shorter resource lookup delay.

Figure 9 shows the evolution of the average resource lookup delay of RLSM-P2P compared with CDP and Gossiping-LB w.r.t. the peers velocity (i.e., we set the number of peers to 60 and we vary the peers velocity from 4.2 to 11.7). We note that RLSM-P2P is more efficient than the baseline protocols for all the variations of the peers velocity. Indeed, it achieves around 23% and 25% less delay than CDP and Gossiping-LB, respectively.

VI. CONCLUSION AND FUTURE WORK

In this paper, we have introduced RLSM-P2P a cross-layer Resource Lookup Scheme for Mobile P2P applications. In this respect, we have introduced a new fully distributed mechanism for building an efficient unstructured P2P overlay that closely matches the underlay physical network. The proposed overlay rapidly adapts to the volatility and the dynamicity of the MANET underlay. Indeed, our scheme makes virtual connections between peers having stable links, less load and enough battery energy in order to increase the retrieval effectiveness and efficiency. In addition, we have proposed a new gossiping-based resource lookup mechanism. We have exploited the past queries to select the relevant neighbors that could provide pertinent resources for forthcoming user queries. The simulation results have showed that RLSM-P2P achieved better results than the baseline protocols. Obvious pointers for future work are as follows:

1) Highlighting certain shortcomings in the evaluation of the RLSM-P2P scheme. Thus, further evaluation is ongoing;
2) Considering other features, such as node degree and connectivity, for building the P2P overlay;
3) Adapting the RLSM-P2P scheme to file sharing applications over Vehicular Ad-hoc Networks (VANETs).

REFERENCES

[1] T. Yeferny and K. Arour, "Efficient routing method in P2P systems based upon training knowledge," in *Proc. 26th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Mar. 2012, pp. 300–305.

[2] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *Proc. 16th Int. Conf. Supercomput.*, New York, NY, USA, Jun. 2002, pp. 84–95.

[3] T. Yeferny, K. Arour, and Y. Slimani, "Routage sémantique des requêtes dans les systèmes pair-à-pair," in *Proc. CORIA*, May 2009, pp. 131–147.

[4] K. Arour and T. Yeferny, "Learning model for efficient query routing in P2P information retrieval systems," *Peer-to-Peer Netw. Appl.*, vol. 8, no. 5, pp. 741–757, 2015.

[5] T. Yeferny, K. Arour, and A. Bouzeghoub, "An efficient peer-to-peer semantic overlay network for learning query routing," in *Proc. IEEE 27th Int. Conf. Adv. Inf. Netw. Appl. (AINA)*, Mar. 2013, pp. 1025–1032.

[6] M. Seddiki and M. Benchaïba, "2P-Lookup: Popularity and proximity based P2P lookup mechanism over MANETs," *J. Netw. Comput. Appl.*, vol. 71, pp. 181–193, Aug. 2016.

[7] N. Shah, S. A. Abid, D. Qian, and W. Mehmood, "A survey of P2P content sharing in MANETs," *Comput. Elect. Eng.*, vol. 57, pp. 55–68, Jan. 2017.

[8] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The dynamic source routing protocol formulti-hop wireless ad hoc networks," in *Proc. Ad hoc Netw.*, Jan. 2001, pp. 139–172.

[9] C. Perkins, E. Belding-Royer, and S. Das, *Ad Hoc on-Demand Distance Vector (AODV) Routing*, document RFC 3561, 2003.

[10] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proc. Conf. Commun. Archit., Protocols Appl.*, Aug. 1994, pp. 234–244.

[11] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazt, "A local search mechanism for peer-to-peer networks," in *Proc. 11th Int. Conf. Inf. Knowl. Manage.*, McLean, VA, USA, Nov. 2002, pp. 300–307.

[12] D. N. da Hora *et al.*, "Enhancing peer-to-peer content discovery techniques over mobile ad hoc networks," *Comput. Commun.*, vol. 32, nos. 13–14, pp. 1445–1459, 2009.

[13] N. Shah and D. Qian, "An efficient unstructured P2P overlay over MANET using underlying proactive routing," in *Proc. 7th Int. Conf. Mobile Ad-hoc Sensor Netw.*, Dec. 2011, pp. 248–255.

[14] A. Mawji, H. S. Hassanein, and X. Zhang, "Peer-to-peer overlay topology control for mobile ad hoc networks," *Pervasive Mobile Comput.*, vol. 7, no. 4, pp. 467–478, 2011.

[15] J. C. A. Leitao, J. P. da Silva Ferreira Moura Marques, J. O. R. N. Pereira, and L. E. T. Rodr, "X-BOT: A protocol for resilient optimization of unstructured overlays," in *Proc. 28th IEEE Int. Symp. Reliable Distrib. Syst.*, Sep. 2009, pp. 236–245.

[16] T. Yeferny, S. Hamad, and S. Belhaj, "CDP: A content discovery protocol for mobile p2p systems," *Int. J. Comput. Sci. Netw. Secur.*, vol. 18, no. 5, pp. 28–35, 2018.
[17] S. Agarwal, A. Ahuja, J. P. Singh, and R. Shorey, "Route-lifetime assessment based routing (RABR) protocol for mobile ad-hoc networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2000, pp. 1697–1701.
[18] C. Hwang and K. Yoon, *Multiple Attribute Decision Making: Methods and Applications: A State-of-the-Art Survey* (Lecture Notes in Economics and Mathematical Systems). Berlin, Germany: Springer-Verlag, 1981.
[19] T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2*. New York, NY, USA: Springer, 2010.
[20] RARE. (Feb. 2018). *RARE Project (Routage Optimisé Par Apprentissage de REquêtes)*. [Online]. Available: http://www-inf.it-sudparis.eu
[21] E. H. Forman and S. I. Gass, "The analytic hierarchy process—An exposition," *Oper. Res.*, vol. 49, no. 4, pp. 469–486, 2001.

TAOUFIK YEFERNY received the M.C.S. and Ph.D. degrees in computer sciences from the University of Tunis El Manar, Tunisia, in 2009 and 2014, respectively. From 2013 to 2016, he was an Assistant Professor with the High Institute of Applied Languages and Computer Science of Beja, Tunisia. Since 2016, he has been an Assistant Professor with Northern Border University, Saudi Arabia. His current research interests include mobile P2P systems, vehicle ad hoc networks, and intelligent transportation systems.

SOFIAN HAMAD received the B.Sc. degree (Hons.) in computer science from Future University, Khartoum, Sudan, in 2003,the M.Sc. degree in management business administration (MBA) from the Sudan Academy of Science, in 2007and the Ph.D. degree in electrical engineering from Brunel University, London, U.K., in 2013. He is currently an Assistant Professor with the Department of Computer Science, Northern Border University. His current research interests include ad-hoc and mesh wireless networks, vehicular technology, and the Internet of Things.

SADOK BEN YAHIA received the habilitation degree to lead researches in computer sciences from the University of Montpellier, in 2009. His experience in teaching, in computer science and information systems is around 20 years. He was a Teaching Assistant with the Faculty of Sciences, Tunis, for two years, an Assistant Professor for seven years, and an Associate Professor for four years. Since 2013, he has been a Full Professor with the Faculty of Sciences. He has been a Professor with the Tallinn University of Technology (TalTech), since 2019. His research interests mainly focus on combinatorial aspects in big data and their applications to different fields, e.g., data mining, combinatorial analytics (e.g., maximum clique problem and minimal transversals), and smart cities (e.g., information aggregation and dissemination, and traffic prediction). He is currently a member of the steering committee of the International Conference on Concept Lattices and their Applications (CLA) and the International French Spoken Conference on Knowledge Extractions and Management.