Ingénierie des Systèmes d'Information
Vol. 26, No. 1, February, 2021, pp. 23-32
Journal homepage: http://iieta.org/journals/isi

ZRR-P2P: Zone-Based Mechanism for Data Replication and Research Optimization in Unstructured P2P Systems

Sarra Cherbal*, Imen Barouchi

Computer Science Department, University of Ferhat Abbas Setif 1, Setif 19000, Algeria

Corresponding Author Email: sarra_cherbal@univ-setif.dz

https://doi.org/10.18280/isi.260103

Received: 18 November 2020
Accepted: 25 January 2021

Keywords:
P2P, peer to peer, search mechanism, zone, area, resource popularity, replication

ABSTRACT

Peer-to-peer systems (P2P) have enjoyed great popularity in terms of sharing content in distributed environments. In mobile P2P networks, how to improve the efficiency of the resource lookup has been an important topic of research. Among the existing solutions, we find lookup mechanisms based on data replication that can increase data availability and reduce search latency. On the other side, these solutions have certain limits such as in the selection of resources to be replicated and in increasing the storage space of peers with additional data. Therefore, this work ZRR-P2P (Zone-based mechanism for data Replication and Research optimization in P2P) comes with the aim of partitioning the network zone on sub-zones and of replicating popular data shared in the network. Thus, in order to improve the research process without increasing storage space of peers.
The simulation results show that our strategy manages to increase data availability and improve the search process in terms of hops count and search latency, while avoiding increasing the storage space of peers.

1. INTRODUCTION

A Peer-To-Peer (P2P) network is a set of entities called peers that are connected to each other by virtual links in order to share resources and services. However, the distributed P2P concept has appeared on the Internet to solve the problem of client/server centralizing concept [1, 2].

In recent years, P2P networks have been widely applied in many fields, such as P2P messaging and P2P resource sharing due to their characteristics of autonomy and anonymity.

On the other side, there is the unlimited evolution of mobile devices (Smartphones, tablets, ...) which has created the need to share information anywhere and anytime, thus, a new paradigm has appeared under the name of mobile P2P [2]. Researches are interested by applying P2P systems over mobile networks like vehicular ad-hoc networks [3], wireless networks with both cellular and WiFi communication [4], cellular networks [5] and wireless sensor networks [6].

As another case of mobile networks, we can consider MANETs since they share the same characteristics as P2P networks such as decentralization and dynamic topology [7]. Researches are interested by improving P2P systems over MANET in terms of different characteristics as lookup performance by maintaining data placement on P2P topology [8], routing lookup queries to relevant peers that are more suitable to answer the query [9] and data dissemination based on clustering to reduce traffic overhead [10].

In this type of systems, lookup (search) service optimization always presents a major challenge to offer a good quality of service to users [11], in terms of criteria such as search latency, the search success rate [12], number of search hops [13, 14], etc.

Therefore, in this work, we are particularly interested in a crucial stage that is the lookup of resources (data). Technically, the search for a resource in a P2P network is an issue of routing the search request from the peer requester to the peer provider of the resource. This routing problem is particularly complicated in an unstructured decentralized architecture, because the peers are dispersed and connected to each other spontaneously and without a pre-established structure.

In this context, among the existing optimizing solutions we find replication methods. An evaluation of P2P replication systems under different scenarios is presented by Potlog et al. [15] and Xhafa et al. [16]. Moreover, several mechanisms of P2P replication are discussed and compared by Lakshmi et al. [17], Mohammadi and Navimipour [18] with a highlight of their new challenges [18]. Therefore, the objective of our work is to propose a replication mechanism based on data popularity to optimize research in a mobile P2P network.

The rest of this paper is organized as follows. In Section 2, we present an overview of P2P systems. In Section 3, we review and discuss some related works, according to which, we give a motivation of the proposed approach. Section 4 summarizes our observations on existing solutions and the report with our proposal. In Section 5, we describe the proposed approach. Section 6 presents the implementation results. Finally, Section 7 concludes the paper.

2. OVERVIEW OF P2P SYSTEMS

2.1 P2P purpose and applications

The main purpose of P2P networks is to share resources and help computers and devices work collaboratively, provide specific services, or execute specific tasks. P2P networks have been widely applied in many fields, such as P2P instant messaging, P2PTV for video streaming, VoIP applications (e.g. Skype), distributed computing, distributed currency for payment networks (e.g. Bitcoin), etc. However, the most common use case for P2P networks is the sharing of files, where peers can send and receive files simultaneously. Therefore, the lookup process is an important part of P2P, particularly, how the requester can find the searched file [2].

As example of P2P software, Bittorent if a very well-known protocol to download files from peers. Following the need for a P2P lookup protocol, Bittorent peers use Kademlia lookup protocol to find file owners. Furthermore, P2P software can be applied on a regular computer, smartphone, tablette, etc.

2.2 Advantages and disadvantages of P2P

We can summarize the different advantages and disadvantages of P2P systems as follows [19]:

2.2.1 Advantages

In comparison with client/server concept, some of the advantages of distributed P2P systems are:

- Scaling up and avoid creating bottlenecks.
- Accelerate the completion of tasks by reducing processing time through direct links between peers.
- Increase the performance of the system by sharing the workload and increasing the autonomy and aggregation of resources.
- Users can join or leave the system at any time and maintain control of their resources.

2.2.2 Disadvantages

- Security issue: because of malicious peers.
- Selfish peers: peers who exploit the resources of others and offer nothing.
- Difficulty of administration: Decentralization makes the system difficult to administer, and a global knowledge of the state of resources and the network is almost impossible.
- Resource Availability Problem: When a node leaves the network, all the resources it provides disappear as well.
- Publication and discovery of resources problem: In some types of P2P systems, effective mechanisms for publishing and discovering resources are particularly lacking when new peers join the network.

2.3 Structured and Unstructured P2P

Decentralized P2P systems are divided into two architectures [2]:

2.3.1 Unstructured P2P

It is formed when the overlay links are established arbitrarily. In which, there is no defined structure that indicates the location of peers and resources on the topology. Thus, the nodes are distributed randomly and the routing of requests is based on flooding mechanism, eg, Napster, Gnutella, etc.

2.3.2 Structured P2P

Peers are organized into a geometric topology. In this architecture, the use of identifiers offers more arrangement to the placement of peers and contents on the overlay, thus, avoiding the arbitrarily establishment of links. In addition, there are specific routing protocols as Chord, Pastry, etc.

3. RELATED WORK

Replication methods applied on P2P networks are used to achieve some purposes as enhancing data availability or reducing data searching latency [8, 19]. In the literature, we find different papers that are interested by this mechanism. In this section, we classify them into three categories according to their main interest as: the used P2P topology (specific network topology), selecting data to replicate (replicas) and selecting where to replicate (replica nodes).

3.1 Structured P2P replication

In structured P2P, we can notice some replication works like in [8, 20-22]. In which, authors replicate resources in specific placements according to a structured protocol like Chord [23].

In Chord protocol [23], each peer has a list of successor and predecessor peers. Thus, Guirat and Filali [20] propose to replicate data in the predecessor list of the holder node, in order to minimize the hops number needed to locate requested data.

The authors of each paper propose a formula to define the identifiers of nodes where to replicate data. Moreover, they improve the lookup routing to reach the closest replica node [8, 21, 22]. The formula proposed by Ghodsi et al. [21] defines identifiers of nodes that are dispersed in the network. While in [8], the defined identifiers are from the predecessor list and in ref. [22] the identifier can be any node in the network.

Consequently, these mechanisms are addressed for a particular P2P structure, unlike the works of the following two categories that can be applied on an unstructured P2P system.

3.2 Measuring popularity factor

Some works are interested by measuring popularity factor. The authors of ref. [24] review some of these works. The authors of ref. [25] measure the popularity based on history of searched videos from internal cache and neighboring cache. Wherein, an item for which the demand in the system is rapidly decreasing should get a smaller value for future demand, while an item for which the demand is rapidly increasing should get a higher value for the expected demand.

The measure is based on a decentralized approach, where, for a set of videos, each node broadcasts a popularity query to other nodes with a predefined TTL counter to get a global popularity factor [26].

However, using a global popularity calculation generates more messages between peers, thus, increases the traffic overhead that is not adaptable for a MANET network. Besides, replicating on randomly choosing nodes can lead to a poor dispersion of data through the network zone. Which is the missing point by Cherbal and Lamraoui [14], where the work interests by proposing an efficiency equation to calculate popularity factor that happens to increase data indexes on the network. However, in ref. [14], how to better disperse these indexes needs to be considered.

3.3 Selecting replica nodes

Other approaches, their main interest is selecting nodes where to replicate data, called replica nodes. The authors use a heuristic algorithm to select the high access nodes as replica nodes [27, 28]. This type of algorithms, propose measuring techniques as an exponential average technique based on the history of receiving queries to find which nodes are receiving more query traffic.

Other works like ref. [29, 30] are based on reputation technics that are trust models. In which, the nodes are considered as trustworthy according to the amount and the quality of services they provide, and then these nodes have priority to replicate data. Thereafter, the work of ref. [31] proposes a replication strategy to create multi copies of a hotspot file in the network, and a replica selection mechanism that considers the concept of non-cooperation of nodes. In other words, the authors of ref. [31] combine the two previous ideas of choosing nodes with high traffic and also high reputation to be the replica nodes.

These mentioned works choose replica nodes among all the network nodes. However, not considering a mechanism to choose replica resources, as replicating all the resources lead to increase the storage space of peers. These resources to replicate can be of high sizes, for which P2P was designed, and their contents may not be useful for these replica nodes, like an unnecessary excess storage.

Table 1 shows the common and different main characteristics between the three defined categories.

Table 1. Comparison of the three categories

Characteristics of categories	(1)	(2)	(3)
Replication method	✓	✓	✓
P2P network	✓	✓	✓
A specific P2P structure	✓	×	×
Popularity of files	×	✓	×
Random dispersion of replicas	×	✓	×
Replica dispersion according to a defined mechanism	✓	×	✓

Therefore, in this paper we propose a replication approach ZRR-P2P that considers popularity of resources, which is defined locally in each node. In addition, we avoid replicating the entire resources, but replicating their indexes. Besides, proposing a zone partition approach and a replication mechanism based on this partition to better disperse the replicas across the network. However, the main aim of this work is to improve the replication method we proposed by Cherbal and Lamraoui [14].

4. OBSERVATIONS

Resource lookup performance in mobile P2P networks can be improved in different ways like: improving the routing of search requests and replicating the most wanted resources.

In Section 3, we have exposed some related works, whose common goal is to make the search process more reliable by using different mechanisms. Like in [8], where the authors propose a new replication approach that aims to improve the performance of P2P networks, either in a static or mobile (dynamic) environment. This approach is specified for structured P2P, unlike ZRR-P2P, where we are interested in unstructured P2P. However, we consider the arrival and departure of nodes like in [8].

In addition, some works like [20, 32, 33] propose approaches while being based on the clustering of peers in groups or communities either in a static unstructured P2P like in ref. [32] or a mobile environment [33] or in a structured P2P [20]. The results presented in this related works show the efficiency of the grouping mechanism. Consequently, we integrate this partitioning concept in our proposed approach.

5. THE PROPOSED APPROACH (ZRR-P2P)

The objective of ZRR-P2P is to propose resources lookup mechanism in a mobile P2P system. First, it is necessary to choose the suitable P2P architecture for MANET. The decentralized unstructured architecture seems to be more appropriate given approximation of the characteristics of MANETs.

In the following points, we summarize the constraints we face in this system of unstructured mobile P2P:

- Node mobility can increase the churn rate (leaving/departure rate) in a dynamic environment. Such, when a node moves, its transmission range changes, and thus it is considered as a leaving node for its old neighborhood and as a new joining node to its new neighborhood.
- Data availability: when a node leaves the network, its shared resources information become unavailable.
- How to choose the resources to replicate and how to determine where to replicate them.
- The limited TTL can cause on an unreliable research. Such, the research can stop before arriving to the destination node.
- Balancing the distribution of resources in the network.

Our objective is to propose a replication mechanism that aims to improve the search quality, i.e. reduce the latency and the number of search hops, by reducing the number of peers involved in each research request. Furthermore, the replication of popular resources increases the availability of data in the network.

As shown in Figure 1, ZRR-P2P consists of four processes, which are: Devising the zone where the nodes are distributed into almost equal sub-zones (sub-areas), the calculation of resource popularity, the replication of popular resource indexes, and balancing the distribution of these indexes.

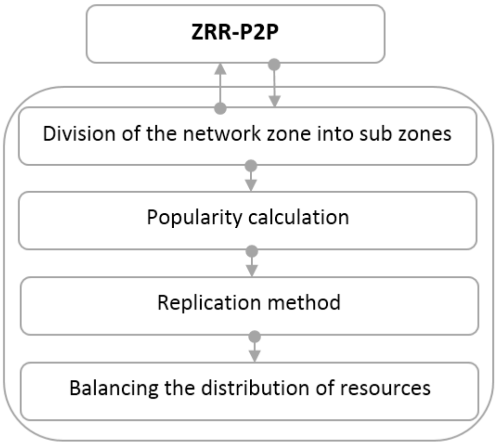
ZRR-P2P
Division of the network zone into sub zones
Popularity calculation
Replication method
Balancing the distribution of resources

Figure 1. The four processes of ZRR-P2P

5.1 Division of the network zone into sub zones

First, regarding the network area, we assume that the region where the peers are scattered is in the form of a square where the length of one side is m. We aim to divide this area to control the number of nodes in each sub-area. For this, we propose the following Equation:

$$n = \frac{1}{5} \times \sqrt{numberofNodes} \qquad (1)$$

Next, we divide each side length m of the square by the number n, which gives new squares, where $n \times n$ is the number of the resulting sub-areas.

Noting that, Eq. (1) controls the number of peers in each sub-area, such it is between 20 and 30. If we want to decrease or increase the number of peers in each sub-area we change the denominator (number 5 in Eq. (1)).

Scenarios. Assuming a network with a side of $m = 500$ and 100 nodes are distributed randomly in its area.

We apply Eq. (1) as follows:

$$n = \frac{1}{5} \times \sqrt{100} = 2$$

Thus, $2 \times 2 = 4$ is the number of sub areas in this network, i.e. dividing each side of the square on 2 results in 4 sub-areas.

In case of a dense network, e.g. with 600 peers, we apply Eq. (1) as follows:

$n = \frac{1}{5} \times \sqrt{600} \approx 4$ (As an approximation). Thus, $4 \times 4 =$ 16 subzones.

In the network, each peer has coordinates (x, y) and after the area division, each zone will have coordinates (x_z, y_z). According to these coordinates, each peer can know the sub-zone where it belongs. At the end of this phase, the network area is divided into sub-areas, each of which contains a certain number of nodes. This phase is presented in Figure 2.

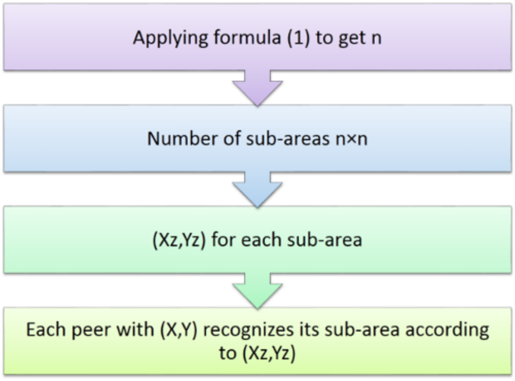
Applying formula (1) to get n
Number of sub-areas n×n
(Xz,Yz) for each sub-area
Each peer with (X,Y) recognizes its sub-area according to (Xz,Yz)

Figure 2. Division of network zone into sub-zones

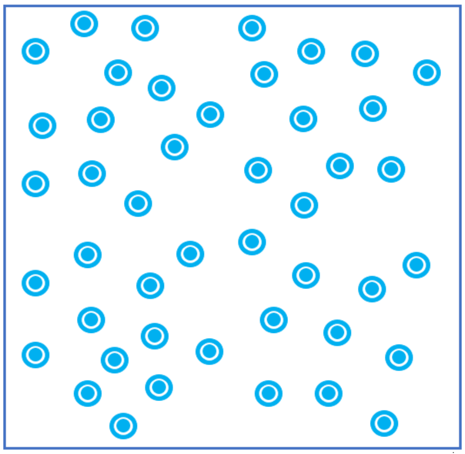

Figure 3. Network area before the division process

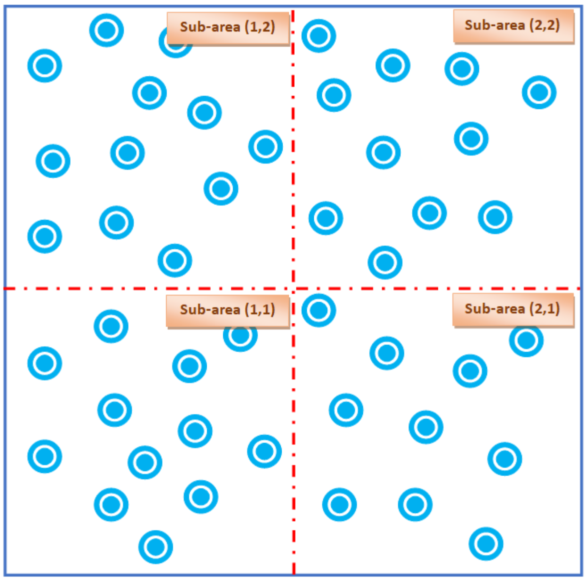

Figure 4. Network area after the division process

However, we propose to apply this network partitioning where the number of nodes exceeds 30 nodes.

Figure 3 represents a P2P network where the peers are randomly dispersed. Next, Figure 4 shows the network after the application of "Zone division" process.

5.2 Popularity calculation

Data replication aims to increase availability, reliability, and performance of data access by storing data redundantly [19].

Replicating popular resources is used to reduce the search space required to find them. The more popular the resource the less we browse the network. Consequently, the search time and network traffic are reduced.

A resource is popular if it is requested in the network more than some others are. In our approach, we define the popularity of resource A as the rate of requests sent in the network to search for A compared to the total number of requests, as follows:

$$P(A, X) = \frac{R(A)}{R} \qquad (2)$$

This measure is calculated based on the peer local knowledge in its popularity table (Table 2).

Such as:

- $P\ (A, X)$: popularity of resource A according to peer X.
- $R\ (A)$: the number of search requests for resource A received by peer X.
- R: the total number of requests received by X.

However, each peer in the network has a table like Table 2.

Table 2. Popularity table (of peer X)

Resource key	Number of requests	Popularity factor
A	7	0.25
B	3	0.10
C	15	0.53

Table 2 presents the number of requests received by peer X, which are launched to search a resource (e.g. A, B or C). The popularity factor presented in Table 2, is measured using Eq. (2) based on the corresponding number of requests.

The content of Table 2 is updated regularly, i.e. each time the peer receives a search request for a resource, it will increase the number of requests concerning this resource and thus the popularity factor will be modified according to Eq. (2).

A resource is considered popular if its popularity factor exceeds a defined threshold. According to Eq. (2), the value of popularity factor is limited in the interval [0 1], so the threshold proposed in our approach is of value 0.4. We choose this threshold value as it is approximately in the middle of the interval [0 1] of popularity values. However, when the resource popularity value measured by Eq. (2) reaches the value 0.4, this resource will be considered as popular.

This mechanism is illustrated in Figure 5 and detailed in Algorithm 1.

Algorithm 1. Popularity calculation

```
While ($X$ receives new requests) do
    $P(X,A) = Nb$ of $A$ requests / total nb of received requests
    If $P(X,A) > 0.4$ then
        $X$ considers $A$ as popular
    else
        $X$ considers $A$ as not popular
    endIf
endWhile
```

5.3 Replication method

To ensure the availability of certain resources, they must be replicated, that is, create multiple copies of these resources and distribute them. In some replication strategies, it is necessary to know whether a resource is popular or not to decide whether to replicate it and on which replica nodes [14]. There are other strategies, like [8], which takes into consideration churn rate and fault tolerance while replicating data.

Therefore, in our approach, we propose that:

First, each node must store in its cache this information $(id, key, (x, y), (x_z, y_z))$, where:

- Id: node identifier.
- key: resource identifier.
- (x, y): node coordinates
- (x_z, y_z) : its sub-area coordinates (where the node belongs).

Second, when a node detects that a resource is popular according to the calculated popularity factor, it will broadcast a resource packet in the network. This packet contains the index of the popular resource (its identifier and its owner node) and the coordinates of node's sub-area. When receiving this packet by other nodes, the receiving peer must decide:

- If the sub-area coordinates of sending node are the same as the receiver, i.e. both nodes belong to the same sub-area, then, the receiving peer will not replicate this popular resource index in its cache and it will broadcast the request to its neighbors.
- Otherwise (the two nodes do not belong to the same sub-area), the peer must replicate the index.

5.3.1 Replication priority

If two peers in the same sub-area receive the packet and they do not belong to the sender node sub-area, they will replicate the receiving index.

Therefore, we add a constraint that the priority of replication goes to the first peer that receives the request. To achieve this proposal, in each iteration, when a node meets the replication condition, it modifies the received packet by adding the replication time and then broadcasts the modified packet. Thus, when a node receives a request from its neighbor (both belong to the same sub-area), it compares the replication times. If the sender's replication time is greater, then, the receiver delete the replicated index (Figure 6).

The objective of this mechanism is that each popular resource will have an index in each sub-area of the network to reduce the search space. In other words, a node that looks for a certain resource may not exceed the limits of its sub-area to reach it. Thus, the search is reliable and with a reduced number of hops.

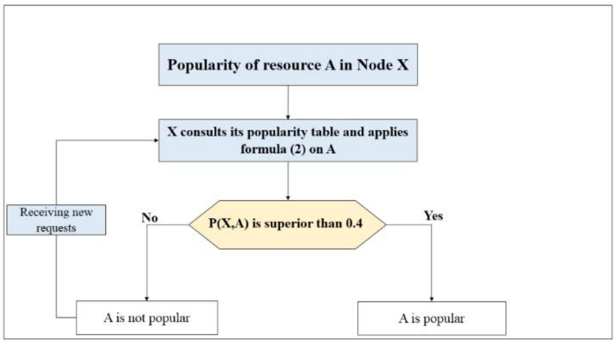
Popularity of resource A in Node X
X consults its popularity table and applies formula (2) on A
Receiving new requests
No
P(X,A) is superior than 0.4
Yes
A is not popular
A is popular

Figure 5. Mechanism of popularity calculation (Node *X* and resource *A*)

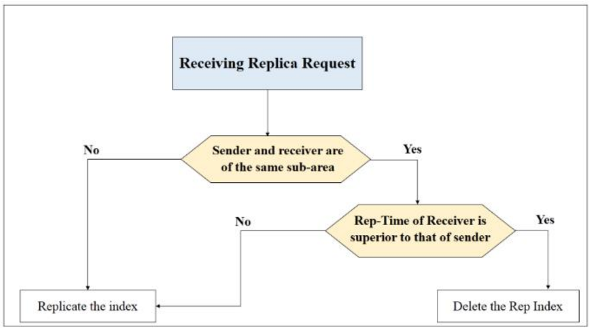
Receiving Replica Request
No
Sender and receiver are of the same sub-area
Yes
No
Rep-Time of Receiver is superior to that of sender
Yes
Replicate the index
Delete the Rep Index

Figure 6. Replication priority mechanism

The Replication process is summarized in Algorithm 2.

Algorithm 2. Replication process

```
When Resource $N$ is considered as popular by peer $W$
    $W$ forwards replica message to its neighbors
    Send ($W$:neighbors, RepMsg)
    RepMsg($N, W, (X_w, Y_w), (X_{zw}, Y_{zw})$)
//$(X_w, Y_w)$ coordinates of $W$
//$(X_{zw}, Y_{zw})$ coordinates of $W$ sub-zone
    Node $v$ receives RepMsg
    If $(X_{zw}, Y_{zw}) = (X_{zv}, Y_{zv})$
//$w$ and $v$ are of same sub-zone
        $v$ forwards RepMsg to its neighbors
    else
        $v$ replicates index $(N, W)$
        $v$ forwards RepMsg to its neighbors including RepTime
//RepTime is replication time of the sender.
        send($v$:neighbors, RepMsg, RepTime)
```

```
endIf
when receiver and sender are of the same sub-area and both receive the same RepMsg
if (receiver:RepTime > sender:RepTime)
    Delete Replica
endIf
```

Figure 7 shows the replication process execution. In which, we notice that resource N is detected as popular by peer W. Then, W will broadcast a replication request (N, W) and the replication will be done in the first three nodes that receive the request, which belong to different sub-areas. In addition, in this Figure 7, we notice the application of replication priority, where two nodes X and Y replicate the index (N, W). Then, after comparing the time replica, Y will delete this index, as X is the first node in this sub-area receiving the replica request of N.

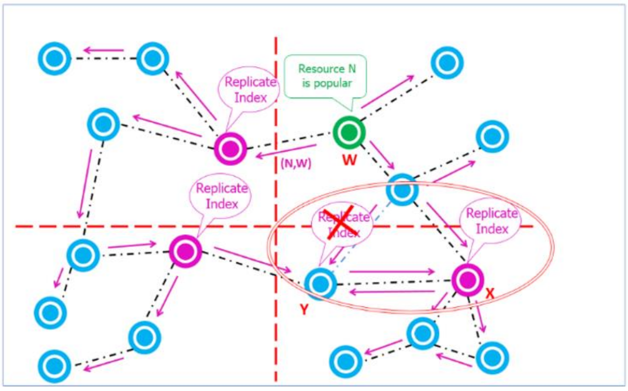

Figure 7. An example explaining the replication process

5.4 Balancing the distribution of resources

After several executions of the replication process, we notice that replicas are distributed on nodes found on the borders of each sub-region. This is due to replication in the first peer that receives replication request.

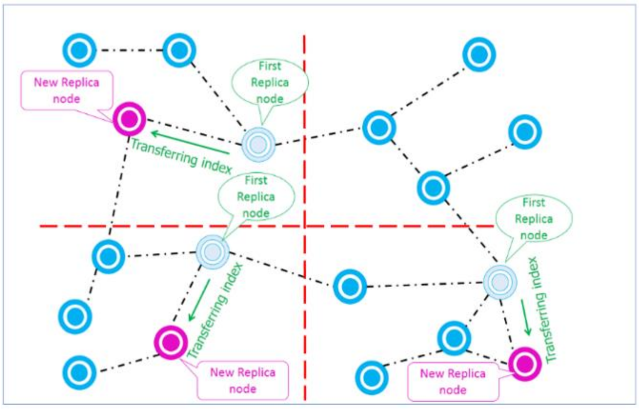

Figure 8. The balancing process of resource distribution

To solve this problem, we propose a process that allows us not to keep the replicated resources only on nodes of the borders. Therefore, after running the replication process by a certain period, the peer that has the index of a replicated resource will choose one of its neighbors (in its sub-region) to transfer this index to it.

After several iterations, we notice that there is a balance in the distribution of resource indexes (Figure 8).

5.5 The lookup process

In the following, we explain search process steps based on the proposed mechanism:

- A search request for resource N is launched in the network.
- The requester (the node that initiates this request) sends it to its neighbors.
- Each time a node receives a search request for N, it checks whether the resource is available or not.
  - If yes, the node sends a response to the sender.
  - Otherwise, the node updates the popularity of N using Eq. (1) and transmits the request to its neighbors with the exception of the sender (until TTL value expires).

If a node finds that resource N is popular, it broadcasts replication request as we explained earlier in this section. Subsequently, if this node (having replica index of N) receives a search request for N, it will respond directly to the requester by this index. Thus, the search process will be performed in this node without passing to other nodes.

When the requester finds the owner of N, it contacts him directly for download.

5.5.1 Departure and arrival of peers

- Arrival: the connection process is done according to the basic method in unstructured P2P.
- Departure of a node that owns the index: before the peer leaves the network, it sends the indexes it has to one of its neighbors (random choice) that is of the same sub-area.

5.5.2 Scenario

<u>Research process before replication.</u> Considering the network shown in Figure 9. In which, there are certain nodes that send requests to find resource N, which is in W.

When W receives the request, it sends a Reply back to the sender.

<u>Research process after replication.</u> In Figure 10, we have replication of resource N in three nodes (U, V, Z).

From now on, when nodes V and Z receive search requests for N, they will respond directly with a $Reply$ (N, W) without passing the request to their neighbors, i.e. without going through many hops or increasing the latency.

Therefore, if a node wants to search for a resource it does not exceed the limits of its sub-area and thus the search is reliable and with a reduced number of hops.

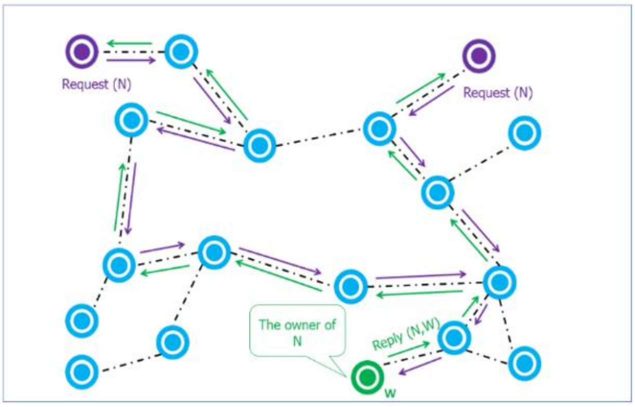

Figure 9. Research process before replication

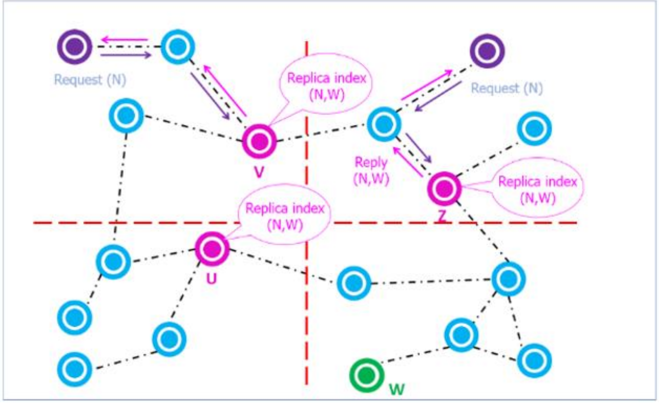

Figure 10. Research process after replication

6. SIMULATION RESULTS

Our implemented system contains a number of nodes (peers) presenting our network. Each node has a set of resources to share, which are assigned in a random manner. Each node can play the role of requester (launching search requests) and provider (providing searched resource). Thus, each peer knows the coordinates (x, y) of its location (in our simulation we give the coordinates according to their locations in the simulation window). For this simulation, we use the Java language.

Simulation area is fixed at $(700 \times 700)m^2$ (we assume that the area is square). We change the number of pairs of 50, 100, 200 and 300. Thus, the number of sub-areas changes depending on network size.

Simulation parameters are listed in Table 3.

Table 3. Simulation parameters

Simulation parameters	Values
Simulation zone size (m×m)	700 × 700
Number of nodes	50, 100, 200, 300
Simulation time (s)	100
Average number of neighbors	4

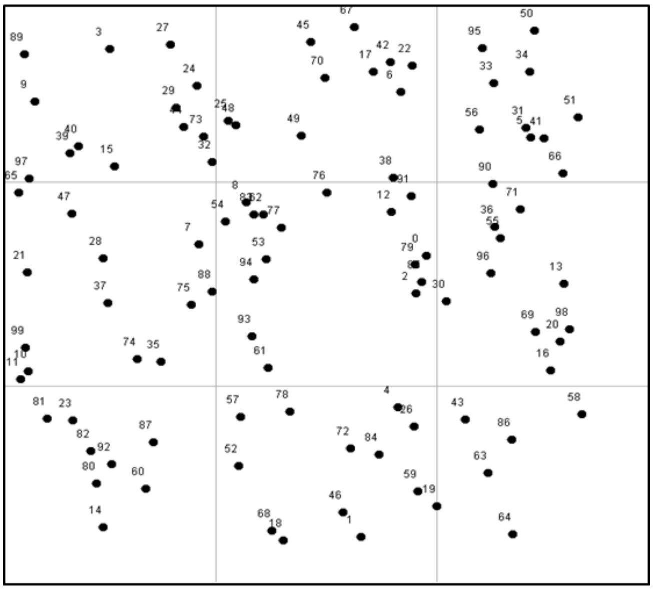

Figure 11. Network area after application of the division process

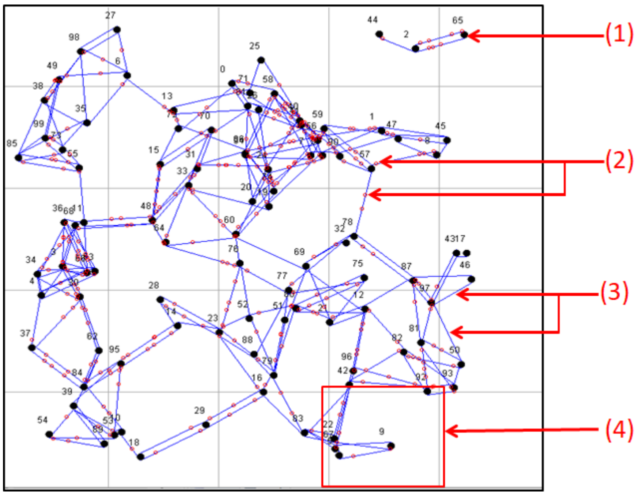

Figure 12. Exchanging messages between nodes

For mobility concept, we present the movement of nodes by disconnection/connection in the network (at some point nodes leave/join the network). Since we have worked according to ad hoc concept, we assume that each peer has a transmission range through which it can communicate with its neighbors. A peer is considered as disconnected from the network if it moves outside the reach of its neighbors.

6.1 Performance metrics

6.1.1 Availability of resources

The number of resource indexes in the network, which presents the number of originally existed resources (on peers) plus the new replicas.

6.1.2 Hop count

The average number of hops or peers traveled by each search process from the requester until the target node. It is given by:

$$H = \frac{(\sum_i^L h_i)}{L} \tag{3}$$

Such that L is the total number of search queries and h_i is the number of hops made by each search.

6.1.3 Lookup latency

The average search latency or file discovery time is given by:

$$T = \frac{(\sum_i^L t_i)}{L} \tag{4}$$

Such that L is the total number of search requests and t_i is the time elapsed by one search request from the source to the target node.

6.1.4 Storage space

The number of total stored resources as entire files on the network. Here, we do not mention the size of shared files, but we can consider large files (large sizes) according to the basic definition of P2P that is often addressed to share large files (e.g. movies).

6.2 Simulation interface

Figure 11 represents the network area after execution of the division process, and nodes are distributed randomly in sub areas where each one has its own identifier.

The network interface is shown in Figure 12, such that:

(1): A network node with its own identifier.

(2): Messages exchanged between nodes.

(3): Links that represent communication between nodes.

(4): A sub-area in the network.

6.3 Execution scenario

In this sub-section, we present a runtime scenario of some nodes connected to each other as an unstructured P2P network.

- Nodes are randomly distributed and resources are assigned to each node. It is according to the number of nodes that the zone is divided and each node knows its corresponding zone.
- Peers send search queries between them, and so the search process is initiated.
- If a node receives a search request and it does not own this searched resource, it transmits the request to its neighbors.
- Otherwise, it answers with a Reply to the requester.
- Each node builds its popularity table according to the received requests. From which it extracts resources and the number of requests for each resource.
- Each node calculates resource popularity factor based on information found in its popularity table.
- The peer perform data replication if he meets the conditions explained in the replication process. Then, the popular resource index will be stored in the storage space of the selected node (which does not belong to the sub-area of the node broadcasting the replication request).
- After a period of time, the node that has the index replica will choose one of its neighbors and transmit this index to it (the principle of balancing the distribution of resources)

6.4 Experimental results

6.4.1 Availability of resources

The main objective of replication mechanisms is to increase data availability on the network. Figure 13 shows the availability of resources before and after applying our approach. We can notice the increase of availability factor achieved by ZRR-P2P, which can lead to enhance lookup process as proved by simulation results in the following two sub-sections.

6.4.2 Hop count

The number of search hops is one of the most important metrics to prove the effectiveness of a new P2P search method over existing methods. Here, we have extracted the average hop count of some launched executions from our implementation with different number of nodes. The hop count for each search process is the number of nodes that the search request traverses from the requesting node to the target node (Eq. (3)). Besides, decreasing the number of included nodes in lookup processes implies the decrease of sending/receiving messages, traffic overhead and consumed energy.

As we have mentioned before, the main aim of this work is to improve the replication method we proposed in ref. [14]. Therefore, we compare the lookups hop count of ZRR-P2P with the work of ref. [14] (Popularity-based Rep).

In Figure 14, we present the results of the average hop count of ZRR-P2P compared to ref. [14]. Hence, we can notice the effect of ZRR-P2P approach on reducing the search path, which proves the effectiveness of our replication process to better choose the locations of replicas.

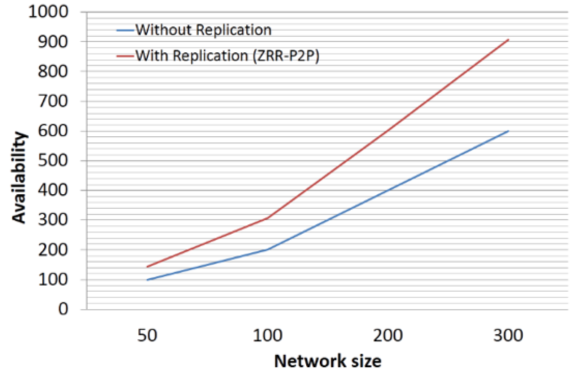

Figure 13. Data availability as function of network size

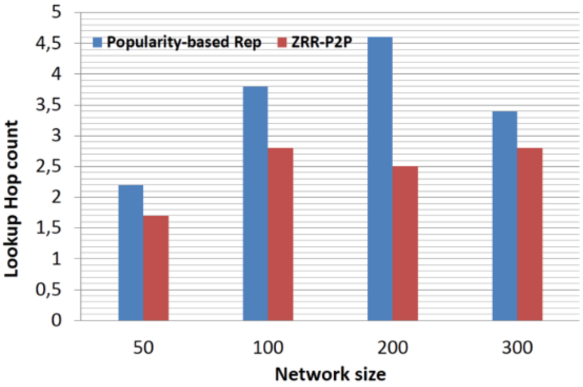

Figure 14. The average lookup hop count as function of network size

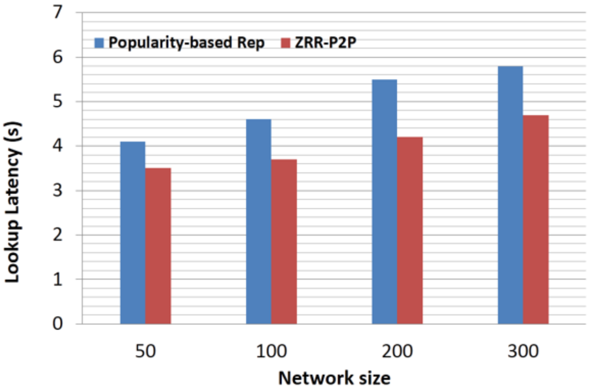

Figure 15. The average lookup latency as function of network size

6.4.3 Lookup latency

Reducing search time or latency is one of the most known requirements of P2P search. In Figure 15, we notice that the search time of our approach is shorter than that of ref. [14]. This is explained by the increase in the availability of data indexes in ZRR-P2P and how it is distributed over the sub-areas compared to ref. [14] and therefore the searched resource is found more quickly. In other words, ZRR-P2P allows finding the closest index possessor.

6.4.4 Storage space

Figure 16 shows the total stored resources (the entire file) in both mechanisms: replication of resources and replication of indexes (our approach) compared to the total number assigned in the network initialization phase. We can notice that in ZRR-P2P, the number of stored files on nodes is the same as initially stored, since in our approach we replicate indexes and not the entire file. In other words, as our approach replicates the file's index instead of the entire file, the number of stored files on nodes of ZRR-P2P does not change. The aim is to avoid storing large files that can be not useful for the replica node.

The results presented in Figure 16 shows the high storage space avoided by our proposal especially when considering large files.

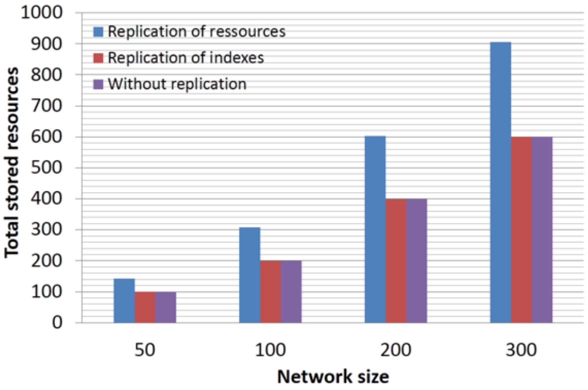

Figure 16. Total stored resources on peers as function of network size

7. CONCLUSION

In this work, we have interested in the problem of finding resources in a mobile P2P network. The question that arose was how to offer a good performance of resource search in a P2P network, while taking into account the constraints related to MANET such as frequent mobility of nodes. To answer this question, we first studied the existing solutions in the literature, which allowed us to identify their advantages and limitations. At the end of this study, we turned to a new replication mechanism based on popularity calculations. Thus, we have proposed ZRR-P2P.

In this paper, we have explained all the processes of this approach, which consist of:

- The division of network area
- The popularity calculation
- The replication method
- Balancing the distribution of resources.

Moreover, we have explained how the research process works based on the proposed approach.

We can summarize the advantages of the proposed contribution in the following points:

- Obtain a popularity factor locally without calculating the overall popularity, which allows avoiding the exchange of additional messages between nodes to calculate this factor. Therefore, avoid the inconveniences associated with excessive traffic especially on a mobile network.
- Replicating the index avoids overloading the storage space of the replica nodes with unnecessary additional resources.
- Minimize the number of hops in the search process by responding directly to the requester with the index.
- Dividing a large network into small subnets using the suggested Eq. (1) offers a better dispersion of indexes on the network area and makes the search easier.
- Each sub-area is considered as a network (the indexes of requested resources could be available in the requester's sub-area), and thus the requester can achieve his search in its sub-area with a smaller path.
- Reduce search latency (search time).
- Replication of indexes makes information about resources more available.
- Replicating popular resources instead of replicating all the resources.

Furthermore, we have presented the simulation results of our proposal compared to other very recent related work, in terms of some known metric performances.

The obtained results show the effectiveness of our contribution in the optimization of P2P search process in terms of increasing data availability, reducing the hops count and search latency while avoiding increasing the storage space of nodes.

As perspectives of this work, we aim to apply our approach with protocols like AODV or OLSR to see the adaptation to this type of protocol communication. Thus, to prove the effectiveness of our work in terms of other performance metrics such as the number of messages circulating in the network.

REFERENCES

[1] Barile, S., Simone, C., Calabrese, M. (2017). The economies (and diseconomies) of distributed technologies: The increasing tension among hierarchy and p2p. Kybernetes. 46(5): 767-785. https://doi.org/10.1108/K-11-2016-0314

[2] Cherbal, S., Boukerram, A., Boubetra, A. (2016). A survey of DHT solutions in fixed and mobile networks. International Journal of Communication Networks and Distributed Systems, 17(1): 14-42. https://doi.org/10.1504/IJCNDS.2016.077938

[3] Liu, C.L., Wang, C.Y., Wei, H.Y. (2010). Cross-layer mobile chord P2P protocol design for VANET. International Journal of Ad Hoc and Ubiquitous Computing, 6(3): 150-163. https://doi.org/10.1504/IJAHUC.2010.034968

[4] Khan, M.A., Yeh, L., Zeitouni, K., Borcea, C. (2014). MobiStore: Achieving availability and load balance in a mobile P2P data store. In 6th International Conference on Mobile Computing, Applications and Services, pp. 171-172. https://doi.org/10.4108/icst.mobicase.2014.257793

[5] Cherbal, S., Boukerram, A., Boubetra, A. (2017). An improvement of mobile Chord protocol using locality awareness on top of cellular networks. 2017 5th International Conference on Electrical Engineering - Boumerdes (ICEE-B), Boumerdes, pp. 1-6, https://doi.org/10.1109/ICEE-B.2017.8192071

[6] Cheklat, L., Amad, M., Omar, M., Boukerram, A. (2018). Energy efficient physical proximity based chord protocol

for data delivery in WSNs. 2018 International Conference on Applied Smart Systems (ICASS), pp. 1-6. https://doi.org/10.1109/ICASS.2018.8652015

[7] Shah, N., Abid, S. A., Qian, D., Mehmood, W. (2017). A survey of P2P content sharing in MANETs. Computers & Electrical Engineering, 57: 55-68. https://doi.org/10.1016/j.compeleceng.2016.12.013

[8] Cherbal, S., Boukerram, A., Boubetra, A. (2017). RepMChord: A novel replication approach for mobile chord with reduced traffic overhead. International Journal of Communication Systems, 30(14): e3285. https://doi.org/10.1002/dac.3285

[9] Yeferny, T., Hamad, S., Belhaj, S. (2020). CDP: A content discovery protocol for mobile p2p systems. arXiv preprint arXiv:2012.01373.

[10] Hu, C.C. (2020). P2P Data dissemination for real-time streaming using load-balanced clustering infrastructure in MANETs with large-scale stable hosts. IEEE Systems Journal. https://doi.org/10.1109/JSYST.2020.2992774

[11] Navimipour, N.J., Milani, F.S. (2015). A comprehensive study of the resource discovery techniques in peer-to-peer networks. Peer-to-Peer Networking and Applications, 8(3): 474-492. https://doi.org/10.1007/s12083-014-0271-5

[12] Yeferny, T., Hamad, S., Yahia, S.B. (2019). Query learning-based scheme for pertinent resource lookup in mobile P2P Networks. In IEEE Access, 7: 49059-49068, https://doi.org/10.1109/ACCESS.2019.2910117

[13] Singh, M., Kumar, C., Nath, P. (2020). Finger forwarding scheme to reduce lookup cost in structured P2P networks. Wireless Pers Commun, 114: 2263-2281. https://doi.org/10.1007/s11277-020-07475-z

[14] Cherbal, S., Lamraoui, B. (2019). Peer-to-Peer lookup process based on data popularity. In International Conference on Information, Communication and Computing Technology, pp. 177-186. https://doi.org/10.1007/978-3-030-38501-9_18

[15] Potlog, A.D., Xhafa, F., Pop, F., Cristea, V. (2011). Evaluation of optimistic replication techniques for dynamic files in P2P systems. In 2011 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 259-265. https://doi.org/10.1109/3PGCIC.2011.47

[16] Xhafa, F., Potlog, A.D., Spaho, E., Pop, F., Cristea, V., Barolli, L. (2015). Evaluation of intra-group optimistic data replication in P2P groupware systems. Concurrency and Computation: Practice and Experience, 27(4): 870-881. https://doi.org/10.1002/cpe.2836

[17] Lakshmi, C.S., Kumar, G.S., Venkatachalam, V. (2015). Survey on caching and replication algorithm for content distribution in peer to peer networks. International Journal of Computer Science and Network Security (IJCSNS), 15(10): 78. https://doi.org/10.5121/ijcseit.2013.3503

[18] Mohammadi, B., Navimipour, N.J. (2019). Data replication mechanisms in the peer-to-peer networks. International Journal of Communication Systems, 32(14): e3996. https://doi.org/10.1002/dac.3996

[19] Spaho, E., Barolli, A., Xhafa, F., Barolli, L. (2015). P2P data replication: Techniques and applications. In Modeling and processing for next-generation big-data technologies, pp. 145-166. https://doi.org/10.1007/978-3-319-09177-8_6

[20] Guirat, F.B., Filali, I. (2013). An efficient data replication approach for structured peer-to-peer systems. In ICT 2013, pp. 1-5. https://doi.org/10.1109/ICTEL.2013.6632143

[21] Ghodsi, A., Alima, L.O., Haridi, S. (2006). Symmetric replication for structured peer-to-peer systems. In Databases, Information Systems, and Peer-to-Peer Computing, pp. 74-85. https://doi.org/10.1007/978-3-540-71661-7_7

[22] Cherbal, S., Boukerram, A., Boubetra, A. (2017). Locality-Awareness and Replication for an Adaptive CHORD to MANet. International Journal of Distributed Systems and Technologies (IJDST), 8(3): 1-24. https://doi.org/10.4018/IJDST.2017070101

[23] Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M. F., Dabek, F., Balakrishnan, H. (2003). Chord: A scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Transactions on Networking, 11(1): 17-32. https://doi.org/10.1109/TNET.2002.808407

[24] Hamdeni, C., Hamrouni, T., Charrada, F.B. (2016). Data popularity measurements in distributed systems: Survey and design directions. Journal of Network and Computer Applications, 72: 150-161. https://doi.org/10.1016/j.jnca.2016.06.002

[25] Das, S.K., Naor, Z., Raj, M. (2017). Popularity-based caching for IPTV services over P2P networks. Peer-to-Peer Networking and Applications, 10(1): 156-169. https://doi.org/10.1007/s12083-015-0414-3

[26] Ganapathi, S., Varadharajan, V. (2018). Popularity based hierarchical prefetching technique for P2P video-on-demand. Multimedia Tools and Applications, 77(12): 15913-15928. https://doi.org/10.1007/s11042-017-5167-y

[27] Shen, H. (2009). IRM: Integrated file replication and consistency maintenance in P2P systems. IEEE transactions on Parallel and Distributed Systems, 21(1): 100-113. https://doi.org/10.1109/ICCCN.2008.ECP.62

[28] Shen, H. (2009). An efficient and adaptive decentralized file replication algorithm in P2P file sharing systems. IEEE Transactions on Parallel and Distributed Systems, 21(6): 827-840. https://doi.org/10.1109/TPDS.2009.127

[29] Joung, Y.J., Chiu, T.H.Y., Chen, S.M. (2012). Cooperating with free riders in unstructured P2P networks. Computer Networks, 56(1): 198-212. https://doi.org/10.1016/j.comnet.2011.09.003

[30] Selvaraj, C., Anand, S. (2012). Peer profile based trust model for P2P systems using genetic algorithm. Peer-to-peer Networking and Applications, 5(1): 92-103. https://doi.org/10.1007/s12083-011-0111-9

[31] Zhu, Y. (2019). Node reputation-based replica replication strategy in P2P networks. In IOP Conference Series: Materials Science and Engineering, 612(5): 052064. https://doi.org/10.1088/1757-899X/612/5/052064

[32] Cholvi, V., Felber, P., Biersack, E. (2004). Efficient search in unstructured peer-to-peer networks. European Transactions on Telecommunications, 15(6): 535-548. https://doi.org/10.1002/ett.1017

[33] Zhou, S., Meng, X. (2020). A location and time-aware resource searching scheme in mobile P2P ad hoc networks. The Journal of Supercomputing, pp. 1-25. https://doi.org/10.1007/s11227-019-03139-3