International Journal of Business Data Communications and Networking
Volume 19 • Issue 1

Towards a Formal MultipathP2P Protocol

Ryma Abassi, University of Carthage, Tunisia
https://orcid.org/0000-0003-2148-7965

Mohamed Amine Riahla, LMOSE Laboratory, Boumerdes University, Algeria*

Karim Tamine, Limoges University, France

Sihem Goumiri, Laboratoire Ingénierie des Systèmes et Télécommunications, Algeria

ABSTRACT

Peer-To-Peer networks are becoming increasingly common as a mean of transferring files over Internet. In this paper, we describe, first, the design and implementation of our P2P system (MultiPathP2P). This latter is based on the social networks concepts where nodes are identified through their virtual addresses, we have designed our protocol based on 1) An architecture exploiting the principle of social networks where nodes are identified by virtual addresses and are able to randomly change their neighbors 2) A process of data request and file sharing differing from those supported in other P2P networks. Second, we present a formal validation of our proposal in order to prove its optimality and completeness.

KEYWORDS

Completeness, Mobile Multi Agent Systems, Optimality, P2P Networks, Routing Protocol

1. INTRODUCTION

P2P networks have become increasingly common as a mean of transferring files over the internet (Muthusamy, 2003). In fact, more than 50 percent of the files downloaded and 80 percent of the files uploaded on the Internet are through P2P networks (Shen et al., 2014). Traditional P2P file sharing systems leverage interconnected peers and their idle resources to distribute content efficiently, albeit at the cost of requiring peers to publicly advertise their downloads (da Silva et al., 2016). However, efficiently locating desired files within these large-scale P2P systems remains a persistent challenge. This necessitates sophisticated routing algorithms.

To address this challenge, we propose a novel routing protocol for P2P networks, termed MultiPathP2P, inspired by ants' behavior. MultiPathP2P establishes multiple paths between a requesting node and its supplier. Our protocol is distinct in two key aspects: (1) it adopts an architecture akin to social networks, where nodes are identified by virtual addresses and can dynamically change their neighbors, and (2) it employs a novel approach to data request and file sharing, diverging from conventional P2P networks by constructing file transfer paths through a series of control messages rather than relying solely on the reverse path of the data request.

DOI: 10.4018/IJBDCN.344416
*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

Considering the broader landscape of networked systems, recent research has underscored the importance of robust protocols and security measures, particularly in wireless sensor networks (Bhushan & Sahoo, 2018; Bhushan & Sahoo, 2020). The authors delve into the technical challenges and vulnerabilities present in these networks, emphasizing the critical need for innovative solutions to ensure network integrity and data security.

However, deploying such a scheme carries inherent risks, underscoring the need for thorough validation prior to real-world implementation (Ben Chehida et al., 2015). Model validation involves demonstrating mathematical consistency and ensuring completeness of the specification with respect to the input space. Hence, in this work, we employ formal and automated methods for specification, utilizing inference systems based on logical rules. These systems analyze premises and derive conclusions, laying the groundwork for subsequent validation steps.

The remainder of this paper is organized as follows: Section 2 provides an overview of related work on routing in P2P protocols. In Section 3, we detail the design of our P2P system, while Section 4 presents the MultiPathP2P protocol. Section 5 introduces an inference system describing our protocol and outlines verification procedures to establish its optimality and completeness. Finally, Section 6 concludes the paper.

2. RELATED WORK

P2P internet data transfer is a longstanding goal of the research community, and our protocol uses in part some existing ideas.

The performance of P2P protocols largely depends on the characteristics of underlying physical network or query distribution for the efficient query response time and minimum resource consumption (Heo et al., 2021; Shin et al., 2020; Sim et al., 2021; Tushar et al., 2021). Unstructured P2P networks, which do not rely on a global index, have traditionally used flooding methods and random walk schemes to distribute object queries in the network. However, these approaches present disadvantages in terms of network overhead and scalability (Khatibi & Sharifi, 2021).

Meanwhile, structured P2P networks have emerged as an organized alternative to unstructured P2P networks. These networks utilize data structures such as Distributed Hash Tables (DHTs) to organize resources coherently and facilitate efficient searches (Augustine et al., 2022; Yu et al., 2020).

Regarding existing protocols, well-established systems such as GNUTELLA (Gopal et al., 2016), OneSwarm (Isdal et al., 2010), Tor (Feigenbaum et al., 2012) and Bittorrent (Torres-Cruz et al., 2017). continue to play a prominent role in file sharing on the Internet. However, these protocols face persistent challenges in terms of performance and security, such as efficient query management in highly dynamic environments and mitigating security and privacy risks.

In this dynamic of innovation and continuous improvement, new approaches are emerging to address the emerging challenges of P2P networks. Concepts such as animal behavior-based routing, the use of artificial intelligence for query optimization, and securing P2P communications through advanced encryption techniques are garnering increasing interest in the research community (Dang et al., 2021; Šešum-Čavić et al., 2016; Shoab & Alotaibi, 2022).

In most existing approaches, a source node broadcasts a request whenever it plans to retrieve any resource in the network. Unlike these methods, we do not exploit a broadcasting technique that exponentially increases the routing overhead, but we introduce a new idea that consists in setting a local request whenever a node plans to receive a data packet. It is the role of Agents, moving within the overlay network during their lifetime, to disseminate this information and to provide routes towards the supplier nodes.

The existing protocols ignored to find solutions for the disconnection due to route changes over time in response of node mobility. In order to ensure a quick response under the challenge of frequent changes in the network topology, we introduce another type of agent called AntRectifier agent which is created by a node when a modification in the value of routing table entry is detected (a link with a neighbor has failed). This allows routing tables to be updated and refreshed in every change of the network topology.

3. MULTIPATHP2P PROTOCOL: HIGH LEVEL DESIGN OVERVIEW

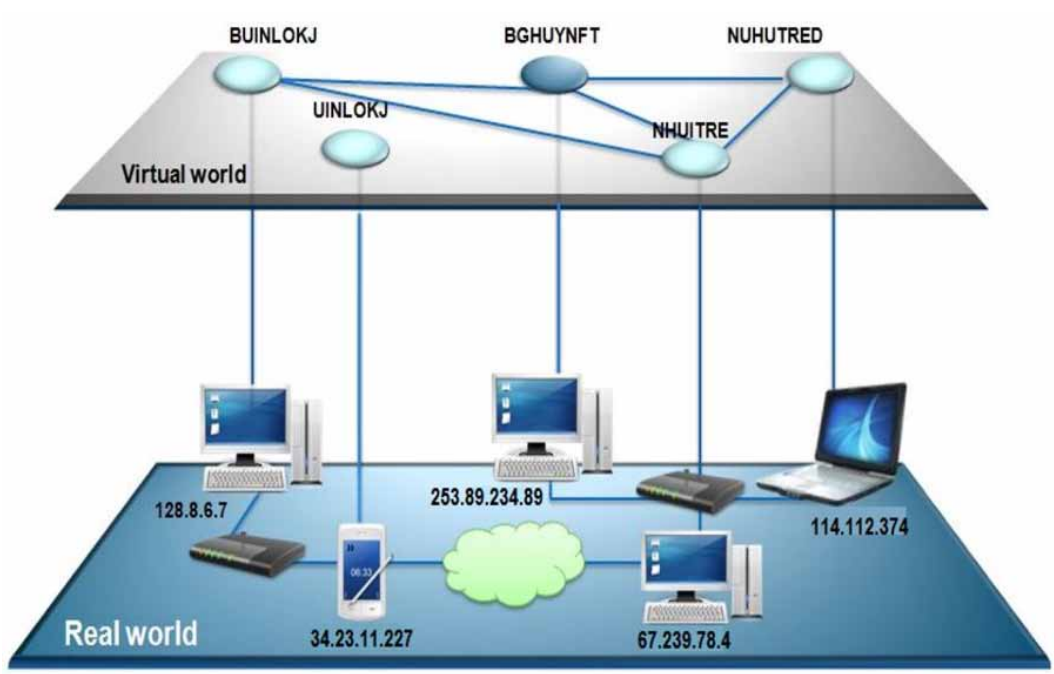
MultiPathP2P is designed to exchange resources between any nodes in the overlay network. The P2P network is based on the principle of virtual social networks such as depicted by Figure 1.

Each node is identified in the network through a connection to neighboring nodes, which are randomly modified over time. When a node plans to download a resource, it launches a request through its neighbors (the issuer of a query does not know where the resource is stored, so there is no destination in queries). This request is routed through a set of control messages operating as a Mobile Multi-Agent system. An agent is created by a given node and sent randomly through the unstructured overlay network (random walks). Let's note that there is no broadcast of request, as in the other P2P systems which exponentially increase the routing overhead. On the contrary, the collaborative Mobile Agents are responsible of the propagation of the request: each agent transports one or more resource requests and establishes paths between the nodes of the network during its life cycle.

Since several paths are built, by collaborative agents, between each pair of nodes, resource downloads are fast and multi-sources. Each network node stores its own requests as well as those of the other nodes increasing by the fact the performance of our MultiPathP2P.

Once the applicant node sends its request via Mobile Agents, it changes with a given probability some of its neighbors and initiates the computing of paths that will be used for the transfer of this resource. Hence, the "path" of the request is different from "paths" of the response. This original way has an advantage because it allows to globally exploiting much of the network for the transfer of resources.

Figure 1. Social network in MultiPathP2P



In summary, our protocol is based on an overlay social network (using virtual identities) with a dynamic topology (changing neighborhood simulating Mobility-nodes in the network).

More precisely, the protocol is built upon three main phases:

The first phase is the diffusion of queries and the establishment of "paths" using agents circulating randomly in an unstructured overlay network.

The second phase in our protocol is the transfer of the resource using several paths previously established. We proposed and implemented in this phase a new ant-based resources routing protocol based on the pheromone trail laying-following behavior of real ants and the related framework of ant colony optimization (ACO (Sama et al., 2016)). In fact, several properties belonging to ant-based routing algorithms are strongly appropriate to address the problems inherent in P2P networks: they are highly adaptive to network changes, robust to route failures, and provide multipath routing.

The third phase deals with packet and links losses with neighbors management in order to ensure a rapid response to the topology changes.

4. DESCRIPTION OF THE PROTOCOL

In this section, the main contribution of this paper is detailed, a new P2P protocol for files transfer. Used notations are defined as shown in Table 1.

4.1 Managing Identities and Connectivity

Each MultiPathP2P user is named using a virtual identity (chosen by the node) identifying the related user among its peers. Moreover, each node has a limited number of neighbors. Initially, when a node wants to join the network, its neighbors are retrieved by querying a set of well-known nodes in the network. To select the closest nodes as neighbors, the protocol establishes proximity measures for each newly known node by calculating the round-trip time (RTT) of a packet through a direct communication using TCP protocol (We use directly TCP/IP stack). Each network node decides periodically, in a probabilistic manner, to make proximity computations with other nodes (especially the newly inserted nodes in the network and which want to be its neighbors) in order to choose them as new neighbors and delete the former ones. This method allows a node to have dynamic neighbors and makes an analogy between nearness in terms of physical network and nearness of P2P overlay network.

4.2 Locating and Transferring Data

After describing how MultiPathP2P peers join and maintain overlay connections and update the connectivity information, let's present the protocol used for search and transfer data between nodes. In order to allow some network nodes to accomplish resource requests and then retrieve the corresponding resources via other nodes, the protocol uses a set of control messages working similarly to a Multi Agents Mobile System. Hence, each Mobile Agent (control message) is created by a node and sent randomly in the network, with a TTL (Time to live) value also chosen randomly. The Mobile Agent owns a set of information that allows nodes cooperating with each other in order to compute search paths resources as well as the transfer paths of these resources.

Table 1. Notations

ID_A	Virtual identity created by the node A
Applicant	The search source node.
Supplier	The data source node.

Agents are periodically sent by each node (even if it does not generate a request) in the network. Moreover, before generating a new agent by a node N, the old one is removed using TTL technique. Hence, at any moment, the number of agents equals the number of nodes in the network.

In other words, each node makes available to the other network nodes a Mobile Agent that locate the resources requested by different nodes, and also establish paths between these different nodes.

Each Mobile Agent transfers two types of information: a list of resource requests initiated by nodes, and a list containing "routing" information. This latter is used for the computation of "paths" in order to transfer resources between applicant and supplier nodes.

The information contained in these lists are dynamically exchanged and updated by the different nodes during the mobility of agents in the network.

In the remainder of this section, the search and transfer process are detailed.

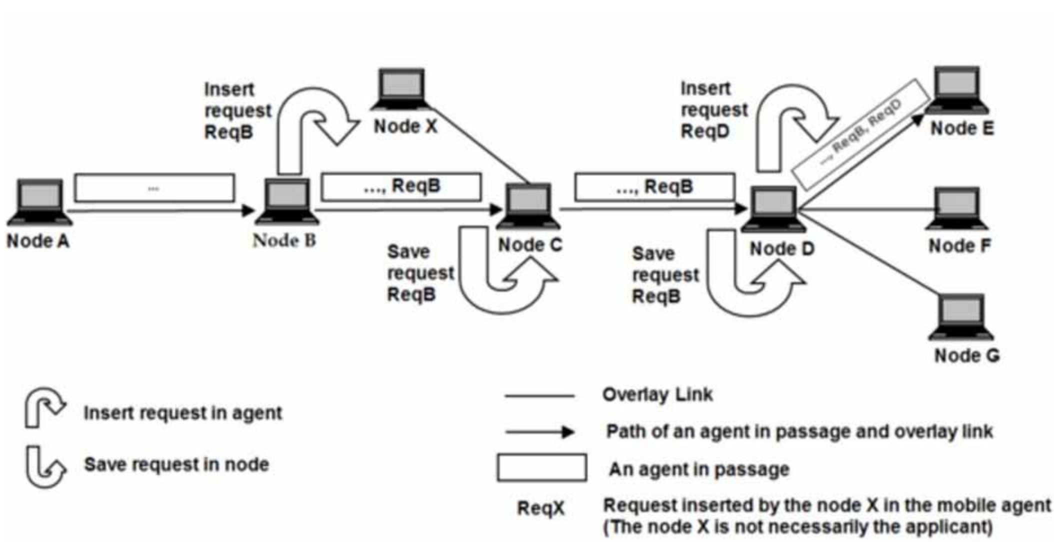
4.2.1 Search Process

As depicted by Figure 2, when a node plans to retrieve a resource in the network, it does not broadcast the corresponding request but it randomly selects a Mobile Agent moving through this node and inserts the following information:

$$\left(\left\langle IDA\right\rangle \left\langle keywords\right\rangle \left\langle TTL\right\rangle \left\langle IDA\_session\right\rangle \left\langle V\right\rangle\right)$$

where <ID_A> represents the virtual address chosen by the applicant node and distributed across the network during this request, < keywords > is a keyword list of the requested resource. <TTL> the number of times the request will be forwarded by the agent across the current node (this value will be decremented every time that the node forwards this request via an agent. When TTL=0, this request will be deleted), <IDA_session> is an integer randomly generated by the applicant node in order to identify the request and avoid loops, and finally <V> is a neighbor randomly chosen by the node to receive the Mobile Agent. Similarly, the Mobile Agents disseminate this information along the crossed nodes.

Figure 2. MultiPathP2P: the search process



4.2.2 Finding the "Paths" for Transfer

Unlike the existing protocols sending the resource along the inverse "path" of search, in our protocol, the owner of a resource will transfer its resources along other "paths" initiated by the applicant. In fact, after sending its request, an applicant node proceeds with some probability to change its neighborhood then it initiates the calculation of "paths" for transferring the resource. The applicant node will initiate this "path" by inserting in its routing table the following entry:

$$\left\langle ID_{A}\right\rangle \left\langle IDA\_session\right\rangle \left\langle neighbour\_node\right\rangle$$

where <ID$_A$> corresponds to the virtual address of the applicant node, <IDA_session> identifies the request and finally, <neighbour_node> corresponds to the next node towards a destination node <ID$_A$> (in this case ID$_A$= neighbour_node). Mobile Agents disseminate this information in order to allow to different nodes to update their routing tables with "paths" corresponding to all the resource requests of the other nodes.

The table entries of these "paths" are updated as follows: if the entry <ID> <ID_session> contained in a Mobile Agent coming from a node x corresponds to a new virtual address in the table of the current node, then the entry: <ID><ID_session><x> is inserted into the routing table of the current node. The Mobile agent applies this update process in all visited nodes.

Using this process and thanks to the collaboration with other mobile agents, multiple "paths" between the supplier and the applicant nodes will progressively be built.

This process ends when the supplier routing table contains entries with a destination towards the virtual address of the applicant node.

4.2.3 Response and Data Transfer

The supplier node sends a response via the "paths" calculated during the previous phase. The reply message includes a search identifier and a list identifying resources that correspond to the search (Each proposed resource has a unique identifier <Id_resource>). These reply messages built inverse "paths". Therefore, the applicant node sends a message through these inverse "paths" to indicate to the supplier the resource to download.

The sent message is a packet structured as follows:

$$\left(\left\langle ID_{r}\right\rangle \left\langle ID_{d}\right\rangle \left\langle ID_{f}\right\rangle \left\langle ID\_session\right\rangle\right)$$

where <ID$_r$> represents the identifier of the resource that will be downloaded, <ID_session> is the identifying number of the request, <ID$_d$> is a virtual address of the resource applicant node and <ID$_f$> is a virtual address of the resource supplier node.

When a supplier node wants to send the requested resource (a transfer message), and in order to optimize the performances of the network, it splits the resource f into a set of packets f1, f2…fn. These packets have an approximately equal size and are sent along different "paths". Each transfer message has the following structure:

$$\left(\left\langle ID_{f}\right\rangle \left\langle ID_{a}\right\rangle \left\langle \left(f,i\right)\right\rangle \left\langle ID\_session\right\rangle\right)$$

where < ID$_f$> and <ID$_a$> represent respectively the virtual addresses of the supplier and the applicant during the request phase, <*(f,i)*> is the part *i* of the resource *f* and <ID_session> the message identifier.

4.2.4 Routing Approach of Resources

In order to enhance the performances of MultiPathP2P protocol, we proposed a new routing protocol based on an optimization technique known as ant colony optimization (ACO) (Correia & Vazao, 2008; Correia et al., 2009; Di Caro et al., 2005; Laxmi et al., 2006) which is inspired from the foraging general behavior of some ant species.

The ant underlying behavior can be summarized as follows: ants deposit pheromone on the ground in order to mark some favorable paths that should be followed by other members of the colony. Other ants perceive the presence of pheromone and tend to follow paths where pheromone concentration is higher. Through this mechanism, ants are able to transport food to their nest in a significant and effective way.

Several properties belonging to ant-based routing algorithms are strongly adequate to address the problems inherent to P2P network: they are highly adaptive to network changes, robust to route failures, and provide multipath routing.

Let's recall that when a node n plans to retrieve a resource, it triggers a request process where the request is locally saved. In our new routing protocol, when an Agent during its lifetime visits a node which has made a request, the Agent transports it and deposits an amount of pheromone on each node that it visits towards this resource applicant node.

This mechanism is used to mark paths towards the resource applicant node n and to inform other nodes (particularly suppliers and other Agents) about this request. The amount of pheromone deposited by the Agent is defined by equation (1):

$$Q_{it} = Q_{i(t)1)} + q \tag{1}$$

where Q_{it} is the pheromone level in the node n_i at time t and q is a positive constant (we choose q=0.1 for our simulations).

For routing data, nodes stochastically forward the data packets. When a node has several neighbors concerned by a requesting node, it randomly selects one of them with the probability p. Each neighbor can have a quantity of pheromone related to requesting nodes.

Let's note $N(n)$ the set of n's neighbors and Q_{it} the amount of pheromone associated to a neighbor n_i stored in the routing table of the node n at time t.

The expression that gives the probability p to select a next hop n_j from node n is defined in equation (2).

$$p = {}^{Q_{jt}}\!\big/_{\sum Q_{kt}} \tag{2}$$

In order to consider requests in an equitable manner leading to a self-organizing system and a better management of frequent changes in the network topology, we propose to set up an evaporation process. This latter allows to no longer take into account the old requests already satisfied. At each time interval, the amount of pheromone corresponding to each request is decreased as defined in equation (3):

$$Q_{it} = (1-\alpha) \times Q_{i(t-1)} \tag{3}$$

where Q_{it} is the amount of pheromone related to a claimant node s, stored in the node n_i at time t and is a real (0<<1) (for the best results, we choose= 0.1 for our simulations).

4.3 Managing Packet Losses

In some cases, an applicant node may not receive the totality of the resource parts sent by a supplier node (essentially caused by a loss of packets in the network). In this case, the protocol allows the node to ask for the missing messages. The applicant node sends then the following request:

$$\left\langle index-list \right\rangle \left\langle ID_{d} \right\rangle \left\langle ID_{f} \right\rangle \left\langle ID\_session \right\rangle$$

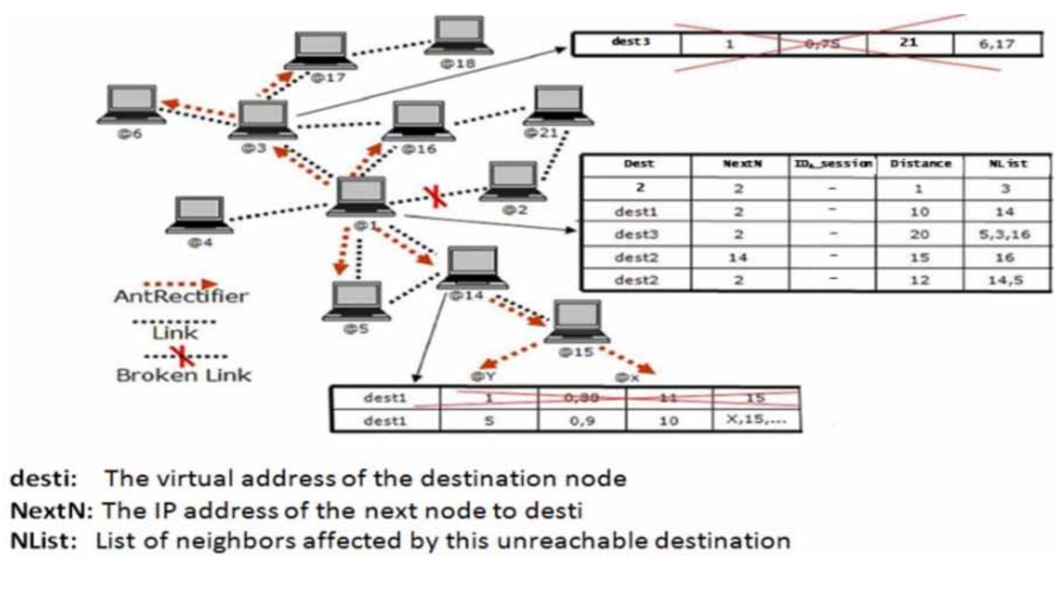
where <index-list> represents the index list of missing parts of resource *f*, <ID_session> is the number identifying the requested resource, <ID_d> is the virtual address of the resource part applicant and <ID_f> is the virtual address of the supplier node. Once the supplier node receive this message, it resends the missing parts of resource *f* by applying the same process described previously.

4.4 Managing Link Losses With Neighbors

In order to ensure a quick response under the challenge of frequent changes in the network topology, we introduce another type of agent called *AntRectifier* agent which is created by a node when a modification in the value of routing table entry is detected (a link with a neighbor has failed). In this case, the node creates the same number of *AntRectifier* agents as the number of nodes that became unreachable. Once created, the *AntRectifier* agents are sent to all the neighbors concerned by these unreachable destinations in order to inform them of this topology change and therefore to take into account this new information within their routing tables. This is depicted by Figure 4 where link between node @1 and 2 is broken. Each neighbor sends back the received *AntRectifier* agent to neighbors concerned by the unreachable destinations; and so on until all concerned source nodes are informed. This allows routing tables to be updated and refreshed in every change of the network topology.

These special agents are created by each node when a link with a neighboring node is broken. This neighbor is stored in the routing table of the node as the next hop to at least one destination in the network. Rectifiers agents are sent to all the neighbors saved in the "NList" field of the node in order to update the routing parameters of its neighbors. Each node receiving a *AntRectifier* agent, and after having updated its routing information, generates an agent with the same type to be sent

Figure 3. Managing links loss between neighbors



to neighbors affected by this broken link and so on. All nodes concerned will be informed without broadcasting information throughout the network.

5. FORMAL SPECIFICATION AND VALIDATION

In the following, we propose a formal and automated expression of the proposed routing algorithm using an inference system. This system is based on the use of logical rules consisting of a function which takes premises, analyses their applicability and returns a conclusion. The second part of this section concerns the validation task proving the optimality and the completeness of the proposal.

5.1. Formal Specification

The proposed inference system is based on the following assumptions:

- Each node has already discovered its neighbors.
- Each node has generated its corresponding mobile agent.

Used notations are depicted in Table 2.

In the following, the proposed inference system describing our system is presented.

Such as depicted in Table 3, inference rules apply to quadruplet $\left(N, REQ, \varnothing, RT\right)$ whose first component N is a set of nodes and their associated agents. The second component, REQ represents the set of requests contained in a mobile agent. The third component M is the set of messages containing the exchanged resource. The fourth component, RT, corresponds to routing table of a node. Initially M is empty.

Four inference rules are proposed. $R_{Applicant}$ handling the applicant's behavior, R_{paths} addressing paths building, $R_{supplier}$ addressing the supplier's behavior, and $Stop$ concerned with the resource download success.

In the following, each of the proposed inference rules is detailed.

Table 2. Used notations

Symbol	Signification
N	A set of couples (x,y) where: - x is the node - y is the mobile agent associated to the node
REQ	The set of requests contained in a mobile agent.
M	The message containing the exchanged resource
RT	The routing table of a node
P	The set of paths used by a supplier to respond a given applicant.
ID_x	The virtual address of x where x can be an applicant node, a resource, a session, etc.
IDA_session	The identifier of the applicant's request
Keywords	A keyword list of the requested resource
TTL	Time To Live associated to a mobile agent
v	A neighbor chosen randomly to receive the mobile agent

- $R_{Applicant}$ inference rule. $R_{Applicant}$ is triggered when a request $req \in REQ$ is sent. This latter can be a request to download a given resource or to ask for a missing packet. In both cases, the $R_{Applicant}$ rule is applied in order to initiate the calculation of paths for transferring the request's object. This is done by adding a route rt to its routing table.
- R_{paths} inference rule. R_{paths} is triggered when a route is added to the routing table but only if $\left\langle ID \right\rangle \left\langle ID_{session} \right\rangle$ contained in the mobile agent of the previous node x corresponds to a new virtual address ID_n in the table of the current node n. In such case, R_{paths} is applied in order add this route to the path P.
- $R_{supplier}$ inference rule. $R_{supplier}$ is triggered when a request arrives to a supplier. In such case, $R_{supplier}$ is applied in order to remove the request req from network $REQ \setminus \left\{ req \right\}$ and to send back a transfer or a reply message.

5.2. Validation

In this sub-section, the validation of the optimality and the completeness of the proposed MultipathP2P system is achieved.

Let us denote by ⊢ * the reflexive application of inference rules of Table 3.

Property 1 *(single agent).* Each node $n \in N$ has a unique mobile agent $a \in A$.

Theorem 1 *(Optimality).* Assuming that initially, each node has a single agent, if $N, REQ, \varnothing, RT \vdash^{*} stop$, then single agent property is preserved.

Proof. Initially, each node has a unique mobile agent. Hence, in the following, we have to check whether the application of each rule of the proposed inference system locally maintains this property. if $N, REQ, \varnothing, RT \vdash^{*} stop$, then only one inference rule among $R_{applicant}$, R_{paths} or $R_{supplier}$ applies for each element in N.

- When a new request is inserted in the network by node n, $R_{applicant}$ is applied, the mobile agent propagates the request and the routing table is updated. Therefore, $\left(n, a \right)$ remains unique.
- When paths are computed, R_{paths} is applied and only the routing table of the node is updated whereas the mobile agent remains unique.
- When a request arrives to a supplier, $R_{supplier}$ is applied, the message is handled according to the situation (transfer or reply) and the corresponding request is removed. Therefore, $\left(n, a \right)$ remains unique.

Once the optimality of the proposed inference system is proved, we proceed to the verification of its completeness. This is achieved by assessing that all potential requests are handled by the inference system.

Property 2 *(System Completeness)* The system is complete if $\forall req \in REQ, req$ is routed to the adequate supplier.

Theorem 2 *(Completeness).* Assuming that initially, the requests set is empty, if $N, REQ, \varnothing, RT \vdash^{*} stop$, then all the requests were handled.

Proof. Initially, there is no unhandled requests. By applying the inference rule $R_{applicant}$, either an applicant or a missing packets request is sent and a route rt is added by inserting the

Table 3. Proposed inference system

$init$	$\frac{}{N, REQ, \emptyset, RT}$		
$R_{Applicant}$	$\frac{(\{n\},\{a\} \sqcup A) \sqcup N, REQ, \emptyset, RT}{N, REQ \sqcup \{req\}, \emptyset, RT \sqcup \{rt\}}$		$where \begin{cases} req \equiv (< ID_A >< keywords >< TTL >< IDA_{session} >< v >)\ if\ applicant\ request \\ req \equiv < index - list >< ID_d >< ID_f < ID_{session} >\ if\ missing\ packets\ request \\ rt \equiv (< ID_A >< IDA_{session} >< neighbor - node > \end{cases}$
R_{Paths}	$\frac{(\{n\},\{a\} \sqcup A) \sqcup N, REQ, \emptyset, \{rt_x\} \sqcup RT}{N, REQ, \emptyset, P \sqcup \{rt_n\}}$	$if\ given\ \{x,a\}, < ID >< ID_{session} > \equiv IDn$	$where \begin{cases} n \in N\ is\ the\ current\ node;\ x \in N\ is\ the\ previous\ node \\ P \subseteq RT \\ \{rt_n\} \equiv < ID >< ID_{session} >< x > \end{cases}$
$R_{Supplier}$	$\frac{(\{n\},\{a\} \sqcup A) \sqcup N, \{req\} \sqcup REQ, \emptyset, P}{N, REQ \setminus \{req\}, M \sqcup \{m\}, P}$		$where \begin{cases} m \equiv \sum_{i=1}^{n} m_i \setminus m_i = < ID_f >< ID_{app} >< fi,i >< ID_{session} >\ if m\ is\ a\ transfer\ message \\ m \equiv < id_r >< ID_d >< ID_f >< ID_{session} >\ if\ m\ is\ a\ reply\ message \end{cases}$
$stop$	$\frac{N, \emptyset, \emptyset, RT}{stop}$	$if\ no\ other\ rules\ applies$	

$ID_A, IDA_{session}\ \ and\ the\ neighbour\ \ node$. This route creation triggers the inference rule R_{paths} in order to computes corresponding paths. Having these latter, the inference rule $R_{supplier}$ handles the message according to its type (transfer or reply message). Hence, the system is complete.

6. CONCLUSION AND PERSPECTIVES

In this paper, we proposed a formal validation of a new P2P system baptized MultiPathP2P. This system uses the social networks principles and builds several paths for file transfer. Hence, we proposed an inference system handling all the steps of the proposed system. Next, we built a validation process using the proposed inference systems and proving the optimality and the completeness of our proposal. Optimality was proved by showing that each node is associated to only one mobile agent. Completeness was proved by showing that all potential situations are handled by the inference system.

CONFLICTS OF INTEREST

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

FUNDING STATEMENT

No funding was received for this work.

PROCESS DATES

Received: 01/23/2021, Revision: 03/27/2024, Accepted: 04/09/2024

CORRESPONDING AUTHOR

Correspondence should be addressed to Mohamed Amine Riahla; ma.riahla@univ-boumerdes.dz

REFERENCES

Augustine, J., Chatterjee, S., & Pandurangan, G. (2022, July). A fully-distributed scalable peer-to-peer protocol for byzantine-resilient distributed hash tables. In *Proceedings of the 34th ACM Symposium on Parallelism in Algorithms and Architectures* (pp. 87-98). doi:10.1145/3490148.3538588

Ben Chehida, A., Abassi, R., Ben Youssef, N., & Guemara El Fatmi, S. (2015, December). *Cryptology and Network Security: 14th International Conference, CANS 2015, Marrakesh, Morocco, December 10-12, 2015, Proceedings*. Academic Press.

Bhushan, B., & Sahoo, G. (2018). Recent advances in attacks, technical challenges, vulnerabilities and their countermeasures in wireless sensor networks. *Wireless Personal Communications*, *98*(2), 2037–2077. doi:10.1007/s11277-017-4962-0

Bhushan, B., & Sahoo, G. (2020). Requirements, protocols, and security challenges in wireless sensor networks: An industrial perspective. Handbook of computer networks and cyber security: principles and paradigms, 683-713.

Correia, F., & Vazao, T. (2008, January). Simple ant routing algorithm. In *2008 International Conference on Information Networking* (pp. 1-8). IEEE.

Correia, F., Vazao, T., & Lobo, V. J. (2009, October). Models for pheromone evaluation in Ant Systems for Mobile Ad-hoc networks. In *2009 First International Conference on Emerging Network Intelligence* (pp. 85-90). IEEE. doi:10.1109/EMERGING.2009.16

da Silva, P. M., Dias, J., & Ricardo, M. (2016, May). Mistrustful P2P: Privacy-preserving file sharing over untrustworthy Peer-to-Peer networks. In *2016 IFIP Networking Conference (IFIP Networking) and Workshops* (pp. 395-403). IEEE.

Dang, N. T., Tran, H. M., Nguyen, S. V., Maleszka, M., & Le, H. D. (2021). Sharing secured data on peer-to-peer applications using attribute-based encryption. *Journal of Information and Telecommunication*, *5*(4), 440–459. doi:10.1080/24751839.2021.1941574

Di Caro, G., Ducatelle, F., & Gambardella, L. M. (2005). AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications*, *16*(5), 443–455. doi:10.1002/ett.1062

Feigenbaum, J., Johnson, A., & Syverson, P. (2012). Probabilistic analysis of onion routing in a black-box model. *ACM Transactions on Information and System Security*, *15*(3), 1–28. doi:10.1145/2382448.2382452

Gopal, S. V., Rao, N. S., & Naik, S. L. (2016, March). Dynamic sharing of files from disconnected nodes in peer to peer systems. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)* (pp. 767-770). IEEE. doi:10.1109/ICEEOT.2016.7754789

Heo, K., Kong, J., Oh, S., & Jung, J. (2021). Development of operator-oriented peer-to-peer energy trading model for integration into the existing distribution system. *International Journal of Electrical Power & Energy Systems*, *125*, 106488. doi:10.1016/j.ijepes.2020.106488

Isdal, T., Piatek, M., Krishnamurthy, A., & Anderson, T. (2010). Privacy-preserving p2p data sharing with oneswarm. *Computer Communication Review*, *40*(4), 111–122. doi:10.1145/1851275.1851198

Khatibi, E., & Sharifi, M. (2021). Resource discovery mechanisms in pure unstructured peer-to-peer systems: A comprehensive survey. *Peer-to-Peer Networking and Applications*, *14*(2), 729–746. doi:10.1007/s12083-020-01027-9

Laxmi, V., Jain, L., & Gaur, M. S. (2006, December). Ant colony optimization based routing on NS-2. *International Conference on Wireless Communication and Sensor Networks (WCSN).*

Muthusamy, V. (2003). *An introduction to peer-to-peer networks.* Presentation for MIE456-Information Systems Infrastructure II.

Roos, S., Schiller, B., Hacker, S., & Strufe, T. (2014, July). Measuring freenet in the wild: Censorship-resilience under observation. In *International Symposium on Privacy Enhancing Technologies Symposium* (pp. 263-282). Springer. doi:10.1007/978-3-319-08506-7_14

Sama, M., Pellegrini, P., D'Ariano, A., Rodriguez, J., & Pacciarelli, D. (2016). Ant colony optimization for the real-time train routing selection problem. *Transportation Research Part B: Methodological*, *85*, 89–108. doi:10.1016/j.trb.2016.01.005

Šešum-Čavić, V., Kuehn, E., & Kanev, D. (2016). Bio-inspired search algorithms for unstructured P2P overlay networks. *Swarm and Evolutionary Computation*, *29*, 73–93. doi:10.1016/j.swevo.2016.03.002

Shen, H., Li, Z., & Chen, K. (2014). Social-P2P: An online social network based P2P file sharing system. *IEEE Transactions on Parallel and Distributed Systems*, *26*(10), 2874–2889. doi:10.1109/TPDS.2014.2359020

Shin, J., Islam, M. R., Rahim, M. A., & Mun, H. J. (2020). Arm movement activity based user authentication in P2P systems. *Peer-to-Peer Networking and Applications*, *13*(2), 635–646. doi:10.1007/s12083-019-00775-7

Shoab, M., & Alotaibi, A. S. (2022). Deep Q-Learning Based Optimal Query Routing Approach for Unstructured P2P Network. *Computers, Materials & Continua*, *70*(3), 5765–5781. doi:10.32604/cmc.2022.021941

Sim, J., Kim, M., Kim, D., & Kim, H. (2021). Cloud Energy Storage System Operation with Capacity P2P Transaction. *Energies*, *14*(2), 339. doi:10.3390/en14020339

Torres-Cruz, N., Rivero-Angeles, M. E., Rubino, G., Menchaca-Mendez, R., & Menchaca-Mendez, R. (2017). A window-based, server-assisted P2P network for VoD services with QoE guarantees. *Mobile Information Systems*. doi:10.1155/2017/2084684

Tushar, W., Yuen, C., Saha, T. K., Morstyn, T., Chapman, A. C., Alam, M. J. E., Hanif, S., & Poor, H. V. (2021). Peer-to-peer energy systems for connected communities: A review of recent advances and emerging challenges. *Applied Energy*, *282*, 116131. doi:10.1016/j.apenergy.2020.116131

Yu, B., Li, X., & Zhao, H. (2020). Virtual block group: A scalable blockchain model with partial node storage and distributed hash table. *The Computer Journal*, *63*(10), 1524–1536. doi:10.1093/comjnl/bxaa046

Ryma Abassi received her engineering degree in Networks & Telecommunications in 2004, and her MSc and PhD degrees from the Higher Communication School, Sup'Com in 2006 and 2010, respectively. Currently, she is an Assistant Professor and the Director at ISET'Com and member of the "Digital Security" lab at SUP'Com. Dr Ryma Abassi was a Fulbright scholar at Tufts University, MA, USA where she worked on formal methods for security protocols validation. Moreover, she obtained the SSHN grant two times in 2014 and 2017 and is a visiting professor at University of Limoges. Her current researches are focusing on MANET/VANET security, trust management,, security protocols validation, IoT security etc. She has more than 30 publications in impacted journals and classified conferences and is co-supervising four PhD students.

Mohamed Amine Riahla is Computer Science Teacher at the University of Boumerdès (Algeria) since November 2008. He received the Ph.D. degree in computer sciences from University of Limoges (France) and the HDR degree from University of BOUMERDES (Algeria). His research and teaching interests are in artificial intelligence applied to security of dynamic networks (Drones, ad hoc, VANET, mesh, Iot and sensor networks). Currently, he is working on routing, security and privacy in cloud and dynamic networks.

Karim Tamine is an Associate Professor of Computer Science and Engineering at the University of Limoges (France) since September 1997. He received the Ph.D. degree and DEA in computer sciences from University Paul Sabatier (ToulouseFrance). His research and teaching interests are in artificial intelligence applied to computer graphics and security of wireless ad-hoc and sensor network. Currently, he is working on multicast routing, quality of service, intrusion detection, security and privacy in P2P networks. He has several refereed international publications (journals and conferences) in all these domains.

Goumiri Sihem is a doctoral student at the Laboratory of Systems Engineering and Telecommunications (LIST) within the Faculty of Technology at M'Hamed Bougara University of Boumerdès, where she has been enrolled since 2019. Her research interests lie in the field of networking and telecommunications, with a specific focus on routing and security in dynamic networks, including Wireless Sensor Networks (WSN), Mobile Ad hoc Networks (MANET), Vehicular Ad hoc Networks (VANET), Flying Ad hoc Networks (FANET), and Peer-to-Peer (P2P) networks.