STUDIA UNIV. BABEŞ–BOLYAI, INFORMATICA, Volume **LXIV**, Number 1, 2019
DOI: 10.24193/subbi.2019.1.08

AN ADDRESS PROPAGATION MODEL IN P2P AND F2F NETWORKS

MOHAMMED B. M. KAMEL[1,2], PÉTER LIGETI[1], AND ÁDÁM NAGY[1]

Abstract. Using identifiers to address the member nodes at DHT based peer-to-peer (p2p) networks provides structured method of addressing the nodes. The node lookup is then used to find the equivalent communication address of a given identifier. One of the main concerns is how to find the communication addresses efficiently, especially if a node has joined or rejoined the network recently. In this paper an address propagation model has been proposed which is used as a solution in friend-to-friend (f2f) overlays at p2p networks. The model keeps the required communication addresses up-to-date in order to reduce the need of any node to perform the lookup process. It allows each node to maintain the addresses in a distributed manner using bucket based broadcasting and guarantees that it has the current up-to-date addresses of its friend nodes as a necessary information to establish a direct connection without any centralized scheme. Despite adding some traffic overhead to the network, the proposed address propagation process is secure and fast.

1. Introduction

Because of sophisticated components of mobile equipment, mobile devices have become important tools to sense, communicate, and compute data. In the last few years and due to the decentralized nature of peer-to-peer (p2p) model, this model is widely used as an alternative to client-server model [14]. p2p network is a decentralized network in which each peer acts as both client and server, which makes it more applicable on emerging systems that consist mostly of mobile nodes. p2p network is also widely used in different applications such as secure chatting, distributed cash system [13], distributed data sharing [4] and distributed secret sharing [9][18]. The open nature of p2p

Received by the editors: September 21, 2018.

2010 *Mathematics Subject Classification.* 68M10, 68M14.

1998 *CR Categories and Descriptors.* C.2.1 [**Network Architecture and Design**]: Network communications – *p2p networks*; C.2.4 [**Distributed Systems**]: Distributed applications – *f2f networks*.

Key words and phrases. Private P2P network, F2F network, Network address discovery, symmetric and asymmetric cryptography.

networks and the ability of almost everyone to join the network make some systems such as [8] to use a private overlay at p2p networks called friend-to-friend (f2f) network as their underlying communication scheme to ensure secrecy and anonymity of participants beyond direct peer nodes [6]. Using f2f networks have number of advantages: first, it allows the participants utilizing the established public p2p network to communicate securely and reliably. Second, along with the reliability it provides the anonymity such that each node communicates with its trusted friend nodes without any necessary knowledge about other trusted connections beyond its direct friend nodes.

One important issue in the f2f overlays at p2p networks is the address discovery of nodes. Members of a f2f network could join and leave the network frequently and later rejoins the network with a new communication address (i.e. new logical address or port number). In distributed secret sharing systems such as Siren [8], there should be a direct connection to a number of predefined nodes in f2f network in order to recover the encrypted data stored in p2p network. This means that in addition to retrieving the data from p2p network, a set of direct connections in f2f network have to be established in order to get the required information to decrypting the retrieved data. At the same time, the node has to be able to discover its friends' addresses on the public p2p network without revealing the friendship relationship between them. Thus, keep the up-to-date address of friend nodes is an essential requirement of such networks. In case that a node failed to open a communication channel with a friend node due to a possible update in a node's logical address, a node lookup process will be started. The node lookup process is used to find an equivalent communication address of a given node identifier. Beside reliability, lookup latency is one of the main concerns of p2p systems that uses Distributed Hash Tables (DHT) [1]. In time-critical systems based on f2f overlays, such lookup process increases the required time to retrieving and deciphering the data. In this paper a model for address propagation has been proposed to guarantee that each node at the f2f network always has the fresh addresses of its friend nodes. The transmitted addresses will be kept confidential and known only by the authorized recipients.

The rest of this paper is organized as follows. The next section introduces the various issues and summarizes the efforts in current research field. Section 3 describes the proposed model of address propagation. Section 4 shows the test results and analysis of the model. Finally, Section 5 presents our conclusions.

2. Literature Review

DHT based systems assign a seemingly unique key (ID) to each node that joins the network. These keys are generated using a specific hash algorithm. The input parameters to these hash functions varies and different methods are used such as node IP [16], randomized generated ID [3] or using identity-based cryptography [2]. Each peer in the DHT p2p network is then responsible for storing the information of a number of files depending on the distance between hash value of the file and its identifier. Metrics such as bitwise exclusive or (XOR)[12] is used to determine the closeness. In distributed secret sharing[9], while the data is stored in p2p network, the required keys to decode this data is stored at f2f network. These systems can use DHT to provide a lookup service. Because of one-way property of hash functions, regardless of used method to generate a node ID, the generated identifier does not contain any information about the communication address of the node.

The logical path of peers on underlying network could vary from the id based path on DHT network between them, thus the lookup latency of the p2p networks can be high which in this case leads to operational inefficiency in applications running over it [11]. Reducing the lookup latency is specifically pertinent to decreasing the number of hops the lookup needs to traverse, which adds the scalability constraint for several lookup mechanisms [19]. On the other hand, the frequent joining and leaving of nodes in p2p network which is known as churn [17] will increase the lookup delay by requiring to connect to different nodes due to leaving of previously available nodes. In case of change in the address of one or more friend node, the connection could be lost between them until their new communication addresses will be captured by each other. Some proposed solutions that use DHT such as [10] in order to solve this issue requires a central entity which does not follow the p2p principle and removes the decentralized nature of it by adding a centralized point. The proposed solution to prevent the lookup process for address discovery is to keep each entry at the table of addresses of each node up-to-date. This will includes the direct confirmation of newly updated communication address to f2f members. Keeping the required communication addresses of the nodes up-to-date increases the performance of decoding the retrieved data and mitigate the execution of lookup process. The bucket based broadcast [5][15] has been used at the address propagation model. At the following section the model has been described in detail.

3. Model Description

3.1. **Parameters.** The participants of the model are represented as a finite set of nodes $\mathcal{N} = \{N_1,...,N_j\}$ in the p2p network that update their addresses

on different time periods. These periods could be overlapped with each other randomly. This identifier differs from the one used by DHT p2p network and should not be confused with it. Suppose that every node $i \in \mathcal{N}$ can generate a digital signature $Sign_i(m)$ of any message m. Furthermore, an existing f2f network is supposed between some subset of participants. The set of friends of node i is $\mathcal{F}_i \subset \mathcal{N}$. Every node i has a common secret key k_{if} with each of its friends $f \in \mathcal{F}_i$. Let $H(.)$ be a collision resistant one-way hash function and $Enc_k(m)$ be an encryption of the message m using the symmetric key k.

3.2. **Security model.** The address propagation process has to be reliable, secure and should be as fast as possible. The model assumes that the set of friends in f2f network for each peer are *honest* nodes and the majority of peers in p2p network are *semi-honest* whose with some predefined probability may drop some or all of the incoming packets instead of forwarding them. The security requirements that the model has to satisfy are

- Completeness: If a packet generated and sent by an honest node, its friend nodes will verify it and later update their local corresponding communication data of the issuer based on the incoming data.
- Authentic delivery: The address that has been issued by an honest node will be received uncorrupted and the receiving friend node in f2f network is able to identify and authenticate the sender.
- Packet Confidentiality: The transmitted packets that contains address data has to be kept private within members of f2f network. In addition, no intermediate node can get any information from the forwarded packets.
- Anonymity: The real identity of the packet's issuer should be kept secret to the members of the p2p network. The friendship of two nodes should not be revealed by any other friend node.

3.3. **The address propagation protocol.** After an update in a node's communication address (e.g. the node has connected to a different network and a new communication address has been assigned to it), the node will inform its friend nodes directly of the newly updated communication data. Then, node i has to generate an Update Requesting Packet (URP) and inform each uninformed member at the f2f network $\{F_{i_1},...,F_{i_n}\}$ of the newly updated address. The size of the URP is set at the system setup phase and it will remain fixed. The fixed size of URP prevents other nodes from getting extra information via the URP's size (e.g. number of friends). After assigning a new communication address to node i, the URP has to be generated. The method that generates an URP for a node is shown in algorithm 1.

Algorithm 1: Pseudo-code for generation of the URP by node n

Input:
$Psize$: the predefined size of the URP
pID_n: private ID of node n
$\mathcal{F}$ {} : set of friend nodes
$pID\{\}$: set of private IDs of $\mathcal{F}\{\}$
$CK\{\}$: set of common keys between node n and members of $\mathcal{F}\{\}$
Output: URP

$sign(hash(pID_n)) \rightarrow URP$
$key \leftarrow$ random number generator
$encrypt_{key}(data) \rightarrow URP$
for *each* f ***in*** $\mathcal{F}$ **do**
  $send(data) \rightarrow f$
  if *no acknowledgementreceived*(f) **then**
    $hash(pID_n \oplus pID_f) \rightarrow URP$
    $encrypt_{CK_f}(key) \rightarrow URP$
  end
end
if $sizeof(URP) \neq Psize$ **then**
  $randomdatasize \leftarrow$ difference$(Psize, sizeof(URP))$
  $randomdata \leftarrow$ random number generator $(randomdatasize)$
  $randomdata \rightarrow URP$
end
return URP

The URP contains the following sections: header, address data, friends' data and padding. These sections are illustrated in figure 1 and a detailed description of them are as the follows:

- Header: At the header section, the issuer node i hashes its private ID and put the signed value of it at the header. Each intermediate node will use the header to determine whether the incoming URP belongs to one of its friend nodes. The hashed value prevents from revealing the private ID of node i to the participants of the p2p network.
- Address data: The data section includes the encrypted data of the node i (i.e. new logical address of node i, its new port and any additional information). The node i encrypts the data using a key k that has been chosen uniformly at random.

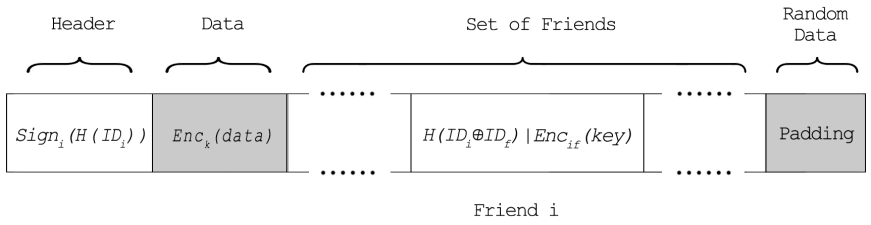

FIGURE 1. Update Requesting Packet

- Friends' data: For each uninformed friend $f \in \mathcal{F}_i$, the node i adds two parts: the common private identifier and the chosen random key. First part that is the common private identifier of nodes i and f will be used by f to indicate which part of the URP belongs to it. This part is computed by first xoring the private ID of i and f, then hashing the resulted value. The second part includes the chosen key k that will be used by f to decrypt the data. This key will be encrypted using the common secret key k_{if}.
- Padding: If the resulted URP's size is less than the predefined packet size, the node i will add some random data at the end of the packet.

After generating the URP by node i, the packet will be broadcast to the set of peer nodes $\{P_1,...,P_m\}$ at the p2p network. Any intermediate node will examine the header to check whether the incoming packet belongs to one of its friend nodes or not. At the final state, all members of the f2f network should receive the transmitted new communication address.

Every node has a set $\mathcal{T}_i$ that includes the hashed value of all friend nodes' private identifiers. This table will improve the checking time of each incoming packet. Any member of the p2p network as soon as receives a packet starts examining the incoming packet to detect whether there is any section of the packet that belongs to it. This operation will be done by first checking the header of the packet. The header will be checked by the public keys of all $i \in \mathcal{F}_i$ in order to find a similar value in $\mathcal{T}_i$. If a value has been found, it means that the sender is $i \in \mathcal{F}_i$. Then, the receiver node has to extracts its part to get the new communication data of the issuer node i. In case that the incoming packet does not belong to any member of $\mathcal{F}_i$, the node will just forward the packet. In case that the issuer node i after a predefined period of time does not receive acknowledgments from all members of $i \in \mathcal{F}_i$, it will regenerate an URP including all uninformed nodes and restart the address propagation process again.

4. Model Analysis

Each node has a unique private ID that is known only by its friend nodes. This ID differs from the node's identifier that is used at the p2p network. The hashed and signed version of this private ID will be part of the header. This field is examined by each receiving node r to check whether the incoming URP belongs to one of its friend node or not. If the incoming URP belongs to one of r's friend nodes, then node r will start checking the first part of each friend section to find its own part. Because node r stores each friend's private ID locally, this field could be computed in advance for each friend at $\mathcal{T}_r$ set. After finding a matching section, the next step is to extract the sender's chosen key k by decrypting it using k_{rf}. The final step is to decrypt the communication address data using the key k. The key k should decrypt the data correctly which indicates that the extracted key has been generated by the pretended sender. The proposed model has to satisfy four security requirements which can be found in table 1.

Table 1. Security Parameters to Meet the Requirements

Security Requirements	Completeness	Authentic Delivery	Packet Confidentiality	Anonymity
header	✓	✓	✓	✓
random key	✓	✓	-	-
encrypted data	✓	✓	✓	✓
padding	-	-	✓	✓

During the test of the proposed model, a p2p network of 300 active nodes has been simulated. All the connections including direct p2p connection and f2f network have been chosen uniformly at random. For the sake of simplicity, it is considered that the offline nodes rejoin the network quickly. During the test of the model existence of partial selfish nodes has been taken into consideration, thus there is a possibility that a node drops part of the incoming packets that do not belong to it instead of forwarding them. Re-transmission rate has been defined as a parameter that indicates the probability of forwarding the incoming URPs at overall nodes in the system. Table 2 shows the details of the parameters that used during the test of the model.

Figure 2 shows the number of URPs that have been transmitted on different test parameters. The overhead increases linearly as the numbers of issued URPs (i.e. nodes with new addresses) increase.

Figure 3 shows the number of issued URPs and the percentage of update rate at the network using different test parameters.

TABLE 2. Test Parameters of the Model

Parameters	Value
number of active nodes	300
Maximum direct peers in p2p network	9
Maximum nodes in f2f network	5
Re-transmission rate	0.2 to 1
Nodes with issued URP	15 to 150

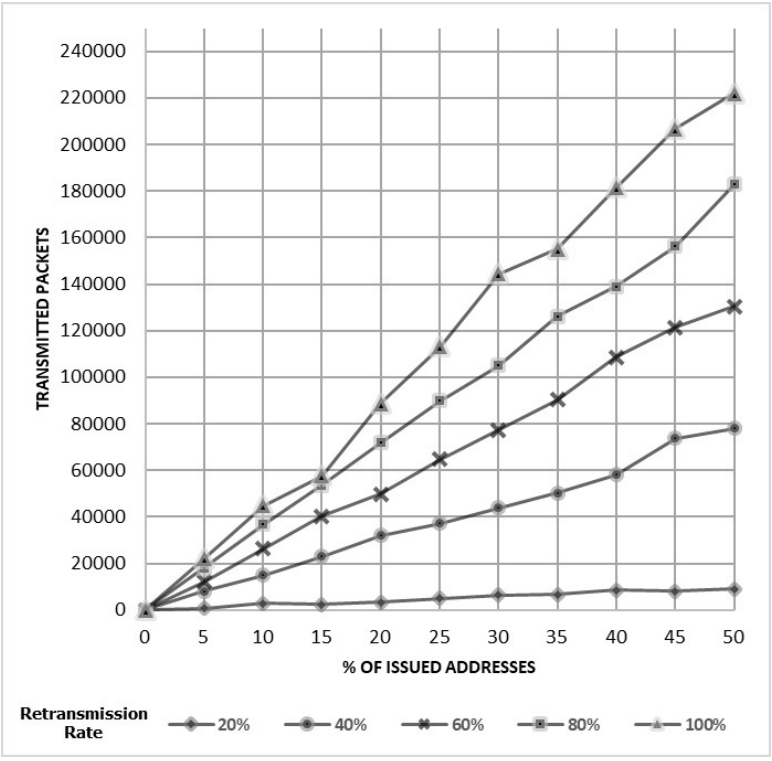

FIGURE 2. Model Overhead

Analysis of the test results indicates that during the test of the model, number of issued URPs (nodes with new addresses) does not affect the update rate of the system. This means that increasing the number of nodes that generate new URPs will not affect the final number of successfully updated addresses. On the other hand, re-transmission rate of intermediate nodes has been found to affect the update rate of the system. Increasing number

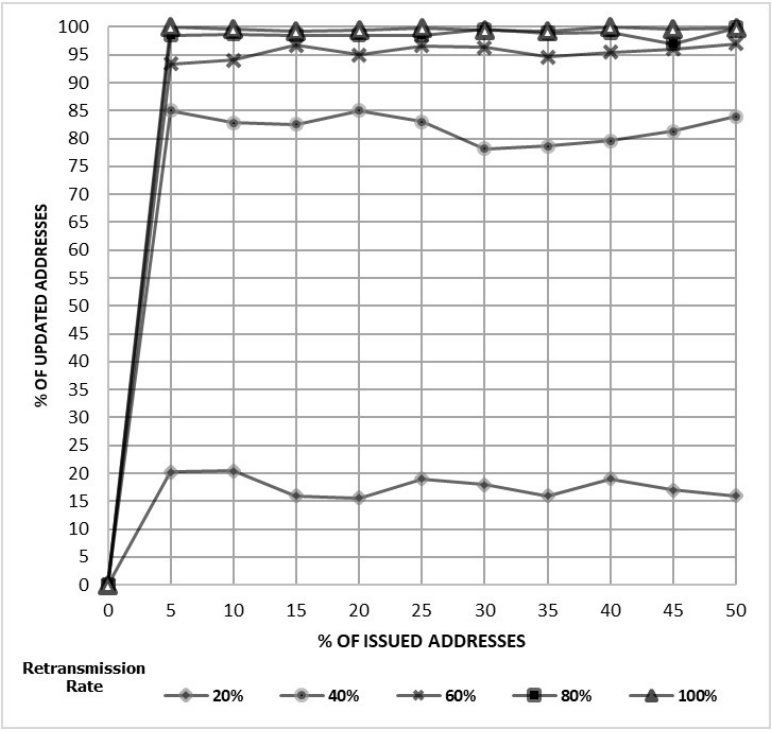

FIGURE 3. Model Update Ratio

of intermediate nodes that do not participate at the system and drops the incoming packets instead of forwarding them will lead to low update rate.

5. CONCLUSION

In this paper an address propagation model has been proposed. This model aims to keep the addresses of the nodes at the f2f network up-to-date. The model assumes honest behaviour from the participants of the f2f network, and semi-honest behaviour of the p2p nodes. Analysis of the test results of the model indicates that re-transmission rate of intermediate nodes directly affects the update rate of the system and, therefore there should be some incentives to ensure that the intermediate nodes will forward the incoming packets and prevent selfish behaviour of the peers at p2p network. The current design requires a flood for each node that has got a new logical address. Beside keeping the friendship relationship of the nodes private, it also adds a significant overhead to the network and the communication overhead increases linearly with the number of nodes. Note that, the extension of the proposed method [7] will

cover different aspects including improvements to the structure of the model to reduce the overall overhead, taking into consideration different issues including packet transmission termination, offline nodes and additional security parameters to mitigate and prevent malicious behaviour of the participants.

Acknowledgement

This research has been partially supported by the UNKP-17-4 New National Excellence Program of the Ministry of Human Capacities, Stipendium Hungaricum Programme and by the European Union, co-financed by the European Social Fund. (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).

References

[1] Balakrishnan, H., Kaashoek, M.F., Karger, D., Morris, R. and Stoica, I., 2003. Looking up data in p2p systems. Communications of the ACM, **46** (2), pp. 43-48.

[2] Butler, K.R., Ryu, S., Traynor, P. and McDaniel, P.D., 2009. Leveraging identity-based cryptography for node ID assignment in structured p2p systems. IEEE Transactions on Parallel and Distributed Systems, **20**(12), pp.1803-1815.

[3] Cai, X.S. and Devroye, L., 2015. The analysis of kademlia for random IDs. Internet Mathematics, **11**(6), pp.572-587.

[4] Cohen, B., 2008. The BitTorrent protocol specification.

[5] Czirkos, Z. and Hosszú, G., 2013. Solution for the broadcasting in the Kademlia peer-to-peer overlay. Computer Networks, 57(8), pp.1853-1862.

[6] Isdal, T., Piatek, M., Krishnamurthy, A. and Anderson, T., 2010, August. Privacy-preserving p2p data sharing with oneswarm. In ACM SIGCOMM Computer Communication Review, **40** (4) pp. 111-122.

[7] Kamel, M., Ligeti, P. and Nagy, A., 2018. Improved Approach of Address Propagation for F2F Networks. IEEE 2018 2nd European Conference on Electrical Engineering and Computer Science (EECS).

[8] Kasza, P., Ligeti, P. and Nagy, A., 2015. Siren: Secure data sharing over p2p and f2f networks. Studia Scientiarum Mathematicarum Hungarica, **52** (2), pp. 257-264.

[9] Kasza, P., Ligeti, P. and Nagy, A., 2015. On a secure distributed data sharing system and its implementation. In ANNALES MATHEMATICAE ET INFORMATICAE **44**, pp. 111-120.

[10] Kohnen, M., Gerbecks, J., and Rathgeb, E.P., 2011. Applying certificate-based routing to a kademlia-based distributed hash table. Proceedings of the Third international Conference on Advances in p2p Systems IARIA, pp. 85-89.

[11] Lua, E.K., Crowcroft, J., Pias, M., Sharma, R. and Lim, S., 2005. A survey and comparison of peer-to-peer overlay network schemes. IEEE Communications Surveys and Tutorials, **7** (2), pp.72-93.

[12] Maymounkov, P. and Mazieres, D., 2002. Kademlia: A peer-to-peer information system based on the xor metric. In International Workshop on Peer-to-Peer Systems, pp. 53-65.

[13] Nakamoto, S., 2008. Bitcoin: A peer-to-peer electronic cash system.

[14] Parameswaran, M., Susarla, A., and Whinston, A.B., 2001. p2p networking: an information sharing alternative. Computer IEEE, **34** (7), pp.31-38.

[15] Peris, A.D., Hernández, J.M. and Huedo, E., 2016. Evaluation of alternatives for the broadcast operation in Kademlia under churn. Peer-to-Peer Networking and Applications, **9** (2), pp.313-327.

[16] Stoica, I., Morris, R., Liben-Nowell, D., Karger, D.R., Kaashoek, M.F., Dabek, F. and Balakrishnan, H., 2003. Chord: a scalable peer-to-peer lookup protocol for internet applications. IEEE/ACM Transactions on Networking (TON), **11** (1), pp.17-32.

[17] Trifa, Z. and Khemakhem, M., 2016. A novel replication technique to attenuate churn effects. Peer-to-Peer Networking and Applications, **9** (2), pp.344-355.

[18] Yong-Jun, G., Li-Zheng, G., and Ming-Hui, Z. 2014. Improved Multi-secret Sharing Scheme Based on One-Way Function. Indonesian Journal of Electrical Engineering and Computer Science, **12** (6), pp. 4463-4467.

[19] Zghaibeh, M. and Hassan, N.U., 2018. d-SHAM: A Constant Degree-Scalable Homogeneous Addressing Mechanism for Structured p2p Networks. IEEE Access, 6, pp.12483-12492.

[1] Eötvös Loránd University, Budapest, Hungary, Faculty of Informatics, 3in Research Group, Martonvásár, Hungary

Email address: mkamel@inf.elte.hu, turul@cs.elte.hu, spigy88@inf.elte.hu

[2] Department of Computer Science, University of Kufa, Iraq