Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 19 (2013) 356 – 363

Procedia
Computer Science

The 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013)

RobP2P: A Robust Architecture for Resource Sharing in Mobile Peer-to-Peer Networks

Khalid Elgazzar, Walid Ibrahim, Sharief Oteafy, Hossam S. Hassanein

School of Computing, Queen's University, Canada

{*elgazzar, walid, oteafy, hossam*}*@cs.queensu.ca*

Abstract

Peer-to-peer (P2P) systems are constructed to provide resource sharing among interested participants (peers) in a distributed and self-organized fashion. The way P2P networks are formed is critical to the overall system performance due to communications and network maintenance overhead. Mobile environments pose additional challenges on P2P networks due to heterogeneity of nodes, inherent limited resources, dynamic context and wireless network characteristics. This paper presents *RobP2P*, a robust architecture to construct mobile P2P networks and efficiently maintain the network state. RobP2P introduces a novel super-peer selection protocol based on an aggregate utility function that takes into account peers' capability and context. It also presents an agile scheme through which super-peers can delegate their responsibilities to more powerful and stable joining or existing peers. Our simulation results show that the RobP2P is efficient, less prone to failure, and generates lower overhead traffic, while reliably maintaining the consistency of network state .

Selection and peer-review under responsibility of Elhadi M. Shakshuki

Keywords: Resource sharing, peer-to-peer, P2P, super-peer selection, super-peer, mobile computing

1. Introduction

Peer-to-peer (P2P) systems have evolved from simple file sharing to advanced distributed paradigms, such as collaborative activities and social applications. Nowadays, the P2P horizon extends to mobile data management [1], heterogeneous resource sharing [2, 3], and mobile service provisioning [4]. P2P systems are self-organizing, capable of maintaining a consistent state despite the fact that the network topology and the number of peers are dynamically changing. Generally, P2P systems share common characteristics such as distributed architecture, collaborative communications, aggregate shared pool of resources/services, and decentralized control over shared resources [5].

The choice of the underlying network architecture has a great impact on the overall system performance. Super-peer networks take advantage of centralized schemes, while benefiting from the robustness of distributed architectures [6]. In super-peer overlay infrastructures, the network topology is constructed in two layers. One layer contains nodes called *super-peers* (or *super-nodes*) that have relatively higher capability and assume special responsibilities. The second layer contains all other peers (called *ordinary peers*). Super-peers handle the communications inside their respective groups as well as exchange information with

Selection and peer-review under responsibility of Elhadi M. Shakshuki
doi:10.1016/j.procs.2013.06.049

other super-peers. Query resolving (resource discovery) in a super-peer architecture is much faster than any other P2P topology. However, super-peer selection is challenging due to the many factors that govern the selection decision which have direct impact on the super-peer performance. Mobile environments pose additional challenges to super-peer selection due to the constantly changing topology and the various constraints stemming from either the limited capability of mobile nodes or the intrinsic characteristics of wireless networks.

Much of the research which has investigated the challenge of super-peer selection and contributed with many algorithms and techniques were designed either for static networks [7, 8, 9, 10] or to address a specific constraint of dynamic mobile environments [11, 12, 13]. Most of these approaches either suffer from high failure rate due to the lack of aggregate consideration to the various mobile environment constrains or trade off reliability for cost of maintaining the overlay topology and incur longer query latency.

This paper proposes RobP2P, a robust mobile P2P super-node architecture with a two-fold vision: 1) lay the foundation for efficient and scalable P2P overlay networks; 2)develop a reliable P2P infrastructure that enables efficient resource sharing and service provisioning in mobile environments with no-infrastructure support. RobP2P presents a novel super-peer selection mechanism that improves the stability of super-peer P2P architectures. It also introduces a reliability improvement scheme that reduces the network maintenance overhead, while improving the overall network reliability and stability. In addition, it reduces the overdue burden on resource-constrained nodes by distributing loads evenly across the network.

The remainder of this paper is organized as follows. Section 2 gives a brief background and outlines related work. Section 3 provides an abstract description of RobP2P. The RobP2P architecture, super-peer selection criteria, selection algorithm, and role changing scheme are presented in Section 4. Section 5 discusses the findings and simulation results. Section 6 concludes the paper and draw future directions.

2. Background and Related Work

P2P systems emerged as an alternative to the well-understood client/server paradigm. The major goal of P2P architecture is distributing the load among all participants instead of relying on a central powerful node. Resources in P2P systems are shared in a collaborative manner and peers that request or offer access to resources have the option to join, leave, and voluntarily participate in an ad-hoc fashion.

Since the introduction of the super-peer P2P network architecture, many research efforts contributed with different variations of super-peer selection algorithms and overlay topology maintenance schemes. Chawathe et al. [14] introduce several modifications to the original design of Gnutella in order to accommodate node heterogeneity and efficiently handle the load when high aggregate query rates occur. Other studies address the inherent constraints of mobile environments, such as limited resources and mobility, in selecting super-nodes. For example, Kim et al. [13] propose a double-layered P2P system, in which super-nodes are selected based on their mobility pattern in order to enhance the system stability and reliability. Kim et al. [11] share the same concern, but they believe that the node energy level should be taken into consideration along with the mobility factor. Merz et al. [15] propose a super-peer topology construction and maintenance scheme based on network coordinates.

In contrast to these previous research efforts, RobP2P integrates many factors to efficiently select super-peers, including the node's current mobility, immediate energy level, network capability, mean uptime, and connectivity degree. Network capability aims to assign a higher priority to nodes with higher bandwidth availability and multiple network interfaces. Whereas the connectivity degree aims to balance the node load (number of served peers) with its capacity and uniformly distribute the peer load across the network topology.

3. RobP2P Overview Description

In this section we describe RobP2P, focusing on selecting super-peers and maintaining the consistency of the P2P overly topology. The design of RobP2P boils down to a three-fold objective: 1) develop a robust and efficient super-peer selection protocol; 2) reduce the overhead traffic of network topology maintenance; 3)

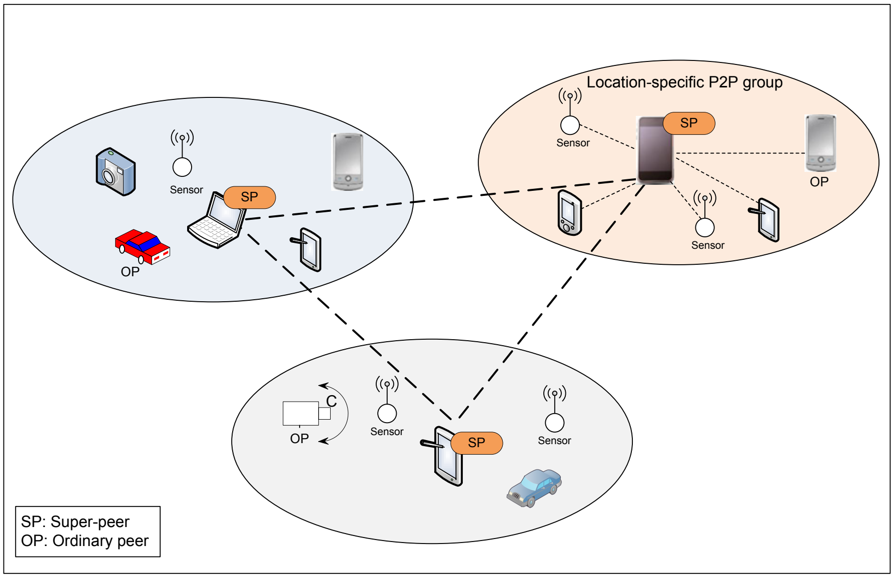

Fig. 1: An abstract overview of RobP2P

increase the reliability and stability of the network infrastructure through enabling peers to flexibly change their role.

Figure 1 shows an abstract overview of RobP2P. We assume that peers (nodes) span a wide range of mobile device form factors and wireless sensors with different features and a variety hardware and software stacks. Some of these peers are high-end mobile devices such as laptops, tablets, smartphones, and vehicles equipped with processing power and network connectivity. Other peers have limited processing power, battery life, storage, and network capabilities, such as featured phones, low-end wireless sensors, cameras, etc. We assume that peers form P2P groups based on location proximity, within a few hops from one another, regardless of their interests and resources they may share. Peers are on the move and their context changes dynamically.

RobP2P is structured in two virtual layers, one layer contains super-peers and the other layer contains ordinary peers. Super-peers are relatively powerful, trust-worthy, and reachable by other peers. The selection of super-peers is based on their current profile, as detailed in section 4.2. Ordinary peers communicate with the rest of the network through their designated super-peer.

Distinguishing features of RobP2P:
-Introducing a robust join/leave procedure, in which newly joining peers may assume super-peer responsibilities based on their profile.
-Presetting a novel *Role Changing Scheme* that enables peers to call for changing their role based on a significant change in their profiles.
-Providing an efficient super-peer selection utility function that accommodates the dynamics of mobile networks and constraints of mobile nodes.

4. RobP2P Architecture

The P2P network is divided into multiple regions. Each region represents a location-based group that contains peers that are physically located with the region boundaries. Algorithm 1 shows the group initialization procedure. Each group selects a super-peer that represents the group head, while the rest of the peers become ordinary peers. All peers calculate their profile index using the utility function in Equation (1) and participate to the super-peer selection following Algorithm 2. Once super-peers are selected, all advertisements and queries within groups are sent to respective super-peers. Super-peers collect and index the group information including active peers, advertised resources, and offered services in order to manage the group communications and resolve queries addressed to the group. The super-peer is also responsible of

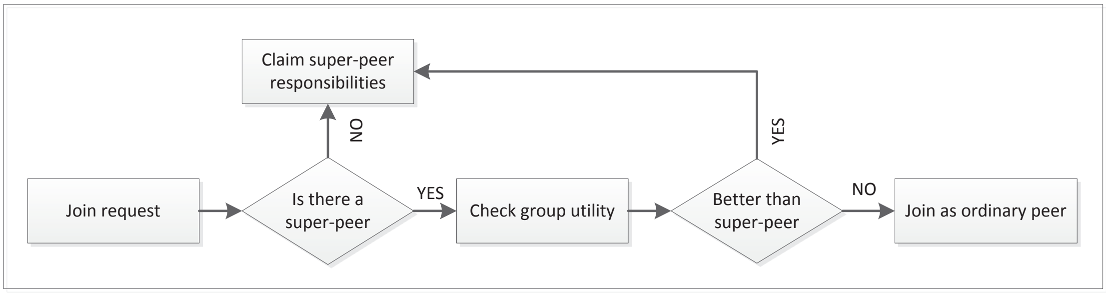

Fig. 2: Join process

maintaining the group state including selecting new super-peers, in case its context changes or moves away of the group's centroid (i.e. where the majority of the group peers can be reached). Ordinary peers generate three type of messages: *hello*, *advertise*, and *query*. Each of these messages contains the peer ID (source), super-peer ID (destination), message type, and a payload. The *hello* message maintains the peer existence in the network and is sent at specific time intervals (TTL). The *hello* message has an empty payload. Peers send the *advertise* and *query* messages whenever they announce or request access to a resource, respectively. Their payload contains information about the offered or requested resource. They also reset the TTL counter of the sender peer. Each advertisement is renewed every TTL in order to keep the associated resource valid. Otherwise, the advertisement is removed from the super-peer index. Each group maintains its state individually and independently of other groups. Super-peers coordinate between each other to communicate their indexed information.

4.1. Join/Leave

The newly joining peer calculates its profile index based on the network *utility function* and compares it with the profile of the current super-peer. If the profile of the new peer outperforms the profile of the super-peer, the new peer takes over the super-peer responsibility. Then, the current super-peer downgrades itself and sends an update message to the group declaring the new super-peer. This message updates the role of the current super-peer and provides the ID of the new super-peer. Figure 2 illustrates the join process and Algorithm 3 shows the join procedure.

Due to the dynamicity of mobile P2P networks and the ever changing context of mobile nodes, super-peers may leave gracefully or die suddenly. Super-peer reselection starts when the network detects super-peer failure or the super-peer itself expects a service disruption or a significant reduction of its connectivity degree (the number connected ordinary peers) due to mobility. In addition, dynamic context changes may result in changing the peers capability, and hence their profile index. Ordinary peers may become more capable to assume super-peer responsibilities, or super-peers might encounter performance depredation. Therefore, in addition to recovering the mobile P2P network state from super-peer failure/disruption, we introduce the *Role Changing Scheme*, aiming at enhancing the overall system reliability.

4.2. Super-peer Selection

The efficiency of a P2P network is highly dependant on the performance of its super-peers and their communication. Selecting super-peers in P2P systems is always challenging. A super-peer must be capable to improve the overall performance of P2P networks, otherwise it might become a bottleneck. The goal of our super-peer selection protocol is to satisfy the following criteria:

- Accessibility - super-peers must be accessible with minimal cost(delay) by all ordinary peers. Peers with multihoming capability (i.e. run multiple network interfaces), must have preferences in super-peer selection, as they can employ their different interfaces to communicate with a wide range of other devices [16].

Algorithm 1: Initialize group (G_i)

```
Input: N_i : Set of nodes in G_i
Output: null
for j ← 1 to |N_i| do
    Broadcast(n_j.ID)
    n_j.NC ← 0
end
while setup_duration and msg_recvd do
    add_neighbor(n_j.ID)
    n_j.NC ← n_j.NC + 1
end
choose_leader(N_i)
```

Algorithm 2: choose_leader (N_i)

```
Input: N_i : Set of nodes in G_i,
MAX_BW // maximal BW in G_i over all
heterogeneous devices
Output: n_current.leader
n_j.profile = calculate_Equ(1)
n_current.leader ← n_current
best_profile ← 0
while selection_duration and msg_recvd do
    if msg_rcvd[n_j.profile] > best_profile then
        n_current.leader ← n_j
        best_profile ← n_j.profile
    end
end
```

Algorithm 3: Join(G_i)

```
Input: n_new : new node to join G_i,
MAX_BW
Output: null
n_new.profile = calculate_Equ(1)
Broadcast("new_join",n_new.ID, n_new.profile)
while wait_duration do
    if msg_rcvd[n_j.profile] > n_new.profile then
        n_current.leader ← n_new
        break //i.e. end search
    end
end
best_profile ← n_new.profile
Broadcast(n_current.leader, best_profile)
```

Algorithm 4: Rcv_join_REQ(n_{new})

```
Input: n_new : new node to join G_i
Output: Msg_response
if Rcv_join_REQ then
    Unicast(n_new, n_current.leader)
end
```

Algorithm 5: Leave(G_i)

```
Input: n_curent.leader: Super-peer
Output: n_new.leader
n_current.leader ← delegate(n_next_best)
best_profile ← n_next_best.profile
Broadcast(n_current.leader, best_profile))
```

- Powerfulness - super-peers must possess sufficient resources to handle the group communications and resolve queries with reasonable delay.
- Distribution - super-peers must be selected so that every node in the P2P network has at least one reachable super-peer and each super-peer must serve a reasonable number of ordinary nodes according its current capacity.
- Mobility - peers with low mobility profiles must be given higher preferences to avoid frequent super-peer selections.
- Context-awareness - super-peers should be capable of detecting their status and re-select new super-peers or delegate their responsibilities to the next best candidate node if they expect service disruption or experience degradation in their performance.

4.3. Super-peer Selection Algorithm

To measure whether a peer n_j is a candidate to assume super-peer responsibilities in a group G_i, we define the peer profile using the utility function in Equation (1). In this equation, b is the current battery power level on n_j, E_{max} denotes the maximum energy level that any peer belongs to G_i might have, m is the current mobility pattern of n_j, M_{max} is the maximum mobility n_j can reach, BW is the current available bandwidth of n_j, BW_{max} represents the maximum bandwidth across G_i, ut is the normalized mean uptime of n_j, which denotes how stable the peer is, NC represents the network connectivity, i.e. how many peers in G_i can reach n_j, $w_1 - w_5$ are weights that represent the factor importance, where $\sum_{k=1}^{5} w_k = 1$. In this utility function, we reverse the peer mobility, since peers with low mobility pattern are of higher preferences. The peer profile ranges from 0 to 1. The higher the profile value, the more possibility a peer could be selected as a super-peer. Each peer in G_i calculates and shares its profile with other peers. The peer with the highest profile declares itself the super-peer serving G_i.

$$n_j.profile = \frac{1}{5}x\left(w_1 x\frac{n_j.b}{E_{max}(G_i)} + w_2 x\frac{M_{max} - n_j.m}{M_{max}} + w_3 x\frac{n_j.BW}{BW_{max}(G_i)} + w_4 x n_j.ut + w_5 x\frac{n_j.NC}{|N_i|}\right) \quad (1)$$

The super-peer selection procedure is shown in Algorithm 2. When establishing connectivity in a group, we assume symmetric communication between all nodes (however no mandate for direct 1-1 messaging to hold the diversity constraint). We assume that BW is aggregated bandwidth over all interfaces of a node. This gives preferences to nodes with multiple interfaces in super-peer selection.

4.4. Role Changing Scheme

The role changing scheme aims to accommodate the dynamic context change of mobile P2P networks, while maintaining the system reliability. A peer initiates the procedure to call for changing its current role, either promoting or demoting itself according to its current situation. Super-peers invoke the procedure shown in Algorithm 5 when they detect degradation in their performance with more than 10% of their original calculated *profile*. This enables peers with more capability to assume super-peer responsibilities. An ordinary peer may also initiate the procedure if it experiences a significant improvement in its capability (such as battery life, bandwidth, or connectivity degree). This improvement in capability must exceed the last reported profile value by the current super-peer. These threshold values are chosen to maintain the network stability and avoid undue overhead that might occur due to false calls.

5. Experimental Results and Discussions

We conducted several experiments to evaluate the performance of RobP2P, focusing on its distinguishing features. We limit the scope of our evaluation setup to investigate the following aspects: 1) the overhead of super-peer selection, 2) how RobP2P maintains a stable state while reducing the number of unnecessary super-peer selection, which will test the quality of our utility functions, 3) how RobP2P handles the churn of mobile networks (i.e. frequent join and leave of nodes), while maintaining the system reliability.

Additionally, since enabling efficient resource sharing and query handling is one of the design objectives of RobP2P, we conducted specific experiments to investigate the query failure rate and the associated overall generated network traffic. The *failure rate* is defined as the number of unsuccessful queries to the total number of submitted queries. We compare the performance of RobP2P with the approach presented by Kim et al. [11], since the authors claim that their approach is superior over MOB [13], while the authors of MOB claim that their approach outperforms Greedy and MIS [17].

The performance evaluation is carried out using the network simulator NS3 [18]. Table 1 summarizes the experimental parameters we used in our simulation.

Parameter	# of peers	Com. range	Topology area	TTL	Exper. Time	Node Energy	Mobility (m/s)
Value	200	100m	1km x 1km	10(sec)	100 TTL	100-2500J	random 0-2.5

Table 1: Summary of the experimental parameters

Figure 3 reveals that the query failure rate is much lower in our system, which reflects the system reliability and stability. This improvement is attributed to our super-peer utility function and the role changing scheme, both of which contribute the most to our system stability through efficient selection of super-peers. However, this stability comes at the expense of generating a little extra traffic as Figure 4 shows; where newly joining peers exchange messages to check whether they are more capable to assume super-peer functionalities or not. Although this little extra traffic is negligible compared to other approaches, the performance benefits to the system reliability are remarkable.

Figure 5 shows that RobP2P is less-prone to query failure in mobile environments, where most of the nodes are always on the move. The result proves that our super-peer selection is efficient, giving preference

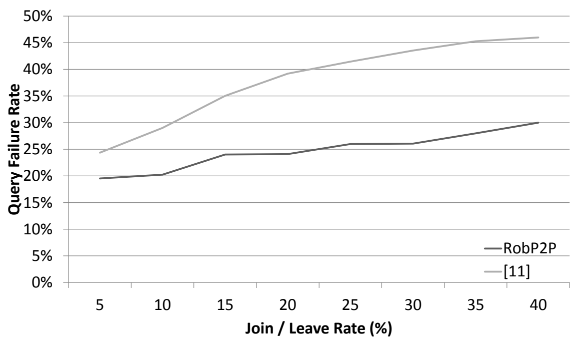

Fig. 3: Query failure versus Join/Leave rate

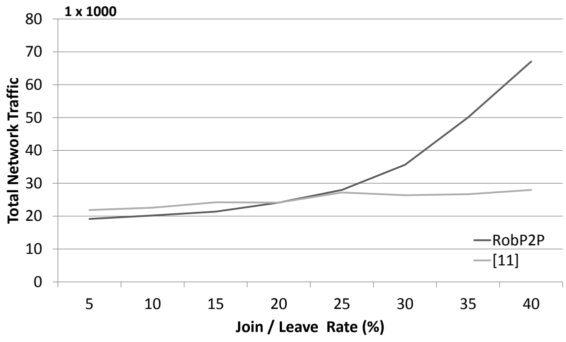

Fig. 4: Total network traffic versus Join/Leave rate

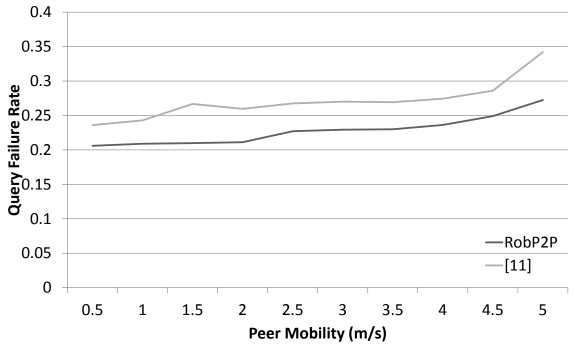

Fig. 5: Query failure versus node mobility

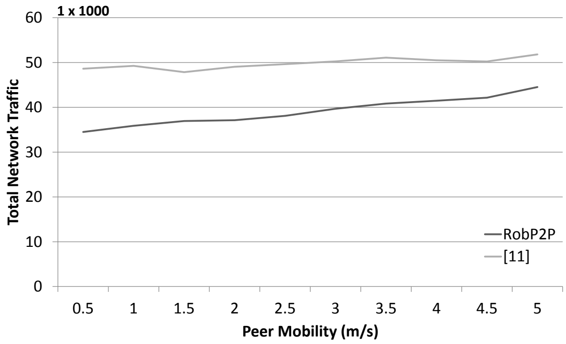

Fig. 6: Total network traffic versus node mobility

to peers with lower a mobility profile and higher connectivity factor. Taking into account the ability of super-peers to communicate over multiple interfaces significantly reduces the query failure rate, since super-peers and ordinary peers are reached through different wireless technologies. Figure 6 shows that RobP2P accommodates peer mobility while maintaining the network state, with lower cost than other architectures.

Figure 7 illustrates the impact of changing the TTL on the query failure rate. The longer the TTL is, the more query failures occur. However, RobP2P outperforms Kim's approach [11]. There are two reasons that explain this observation. First, our super-peer selection function takes into account the various factors that accommodate the inherent dynamics of P2P mobile networks, which by itself makes the super-peer selection efficient. Secondly, the role-changing scheme, that we introduced to handle the network churn and the dynamic change in the node context, enables peers to request changing their role regardless of the TTL. This reduces the number of unnecessary invocations to the super-peer selection algorithm, while maintaining an overall high reliability. On the other hand, Figure 8 shows that a significant reduction of the network maintenance traffic occurs as a natural result of extending the TTL period. However, RobP2P is capable of maintaining a stable network state, while others fail when the TTL period becomes longer.

6. Conclusion

This paper presents *RobP2P*, a robust mobile P2P architecture that enables efficient resource sharing. RobP2P introduces an aggregate utility function that determines whether a peer is a candidate to assume super-peer responsibility. This utility function takes into account both the mobile node constraints and mobile network dynamicity. RobP2P also introduces a novel scheme that enables peers to call for changing their role based on a significant change in their current profile. This scheme renders the mobile P2P network topology, constructed with RobP2P, more stable. It also significantly reduces the network maintenance overhead while maintaining a high level of reliability. Simulation results show that RobP2P outperforms other

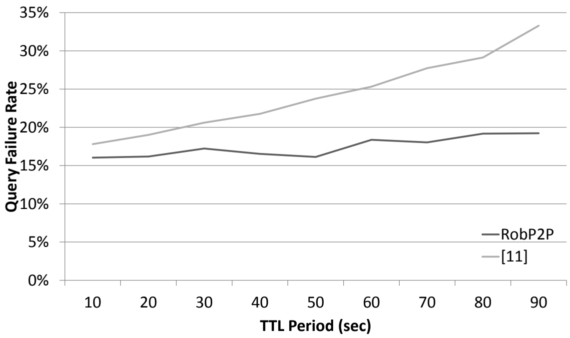

Fig. 7: Query failure versus varying TTL periods

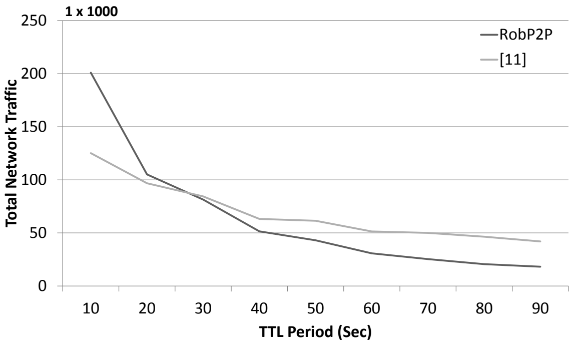

Fig. 8: Total network traffic versus TTL periods

P2P architectures. We plan to continue improving RobP2P by introducing a middle layer that contains *relay peers*, whose profiles are close to the selected super-peer. Relay peers are intended to extend the structure of the overlay network, while maintaining same level of reliability. We also plan to further investigate related performance issues.

Acknowledgment

This research is funded by a grant from the Ontario Ministry of Economic Development and Innovation under the Ontario Research Fund-Research Excellence (ORF-RE) program.

References

[1] M. Ylianttila, E. Harjula, T. Koskela, J. Sauvola, Analytical model for mobile p2p data management systems, in: Proceedings of the 5th IEEE Consumer Communications and Networking Conference, 2008, pp. 1186–1190.
[2] L. Barolli, F. Xhafa, Jxta-overlay: A p2p platform for distributed, collaborative, and ubiquitous computing, IEEE Transactions on Industrial Electronics 58 (6) (2011) 2163 –2172.
[3] C. Yu, D. Yao, X. Li, Y. Zhang, L. T. Yang, N. Xiong, H. Jin, Location-aware private service discovery in pervasive computing environment, Information Sciences (0) (2012) 1–16.
[4] K. Elgazzar, P. Martin, H. Hassanein, A framework for efficient web services provisioning in mobile environments, in: Mobile Computing, Applications, and Services, Vol. 95 of Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer Berlin Heidelberg, 2012, pp. 246–262.
[5] M. Hefeeda, Peer-to-peer systems, school of Computing Science, Simon Fraser University, Surrey, Canada (2004).
[6] B. Yang, H. Garcia-Molina, Designing a super-peer network, in: Proceedings of the Intl Conference on Data Engineering, 2003.
[7] Gnutella website. `http://www.gnutella.com`. [Accessed: Jan. 28, 2013].
[8] Freenet website. `http://freenet.sourceforge.net`. [Accessed: Jan. 28, 2013].
[9] Kazaa website. `http://www.kazaa.com`. [Accessed: Jan. 28, 2013].
[10] V. Lo, D. Zhou, Y. Liu, C. GauthierDickey, J. Li, Scalable supernode selection in peer-to-peer overlay networks, in: Proceedings of the Second International Workshop on Hot Topics in Peer-to-Peer Systems, 2005, pp. 18 – 25.
[11] S.-K. Kim, K.-J. Lee, S.-B. Yang, An enhanced super-peer system considering mobility and energy in mobile environments, in: Proceedings of the 6th International Symposium on Wireless and Pervasive Computing (ISWPC), 2011, pp. 1–5.
[12] A. M. Mahdy, J. S. Deogun, J. Wang, A dynamic approach for the selection of super peers in ad hoc networks, in: Proceedings of the Sixth International Conference on Networking, 2007.
[13] J.-H. Kim, J.-W. Song, T.-H. Kim, S.-B. Yang, An enhanced double-layered p2p system for the reliability in dynamic mobile environments, Computing and Informatics 30 (1) (2011) 1001–1023.
[14] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, S. Shenker, Making gnutella-like p2p systems scalable, in: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM '03, 2003, pp. 407–418.
[15] P. Merz, M. Priebe, S. Wolf, Super-peer selection in peer-to-peer networks using network coordinates, in: Proceedings of the 3rd International Conference on Internet and Web Applications and Services, 2008, pp. 385 –390.
[16] A. Meads, I. Warren, Extending mobile service middleware with support for context-aware service processing, in: Proceedings of the IEEE Region 10 Conference, 2010, pp. 2418 –2423.
[17] J.-S. Han, K.-J. Lee, J.-W. Song, S.-B. Yang, Mobile pee-to-peer systems using super peers for mobile environments, in: Proceedings of the International Conference on Information Networking, 2008, pp. 1 – 4.
[18] The Network Simulator website. `http://www.nsnam.org`. [Accessed: Jan. 28, 2013].