

A CROSS-LAYERED PEER-TO-PEER ARCHITECTURE FOR WIRELESS MOBILE NETWORKS

Mohammad Mursalin Akon, Sagar Naik, Ajit Singh and Xuemin (Sherman) Shen

Department of ECE, University of Waterloo, Waterloo, Ontario, Canada N2L 3G1
Email: {mmakon, k.naik, asingh}@ece.uwaterloo.ca, xshen@bbcr.uwaterloo.ca

ABSTRACT

In this paper, we propose a novel peer-to-peer architecture for wireless mobile networks where a cross-layered gossip-like protocol is the heart of the architecture. The goal of this architecture is to reduce the bandwidth consumption and at the same time, to provide more user participation flexibility. Simulation results are given to demonstrate the performance of the proposed peer-to-peer architecture.

1. INTRODUCTION

Peer-to-peer (P2P) networks have received considerable attention due to their broad applications. Unlike traditional client-server based applications, peers¹ in a P2P network collaborate to achieve certain task such as file sharing, video streaming, etc. Design of a good P2P network requires a well thought logical architecture among the participating peers. With the deployment of high speed 3G (and expected deployment of 3.5G and 4G) cellular networks and wireless LANs, there is an increasing interest in P2P networks for mobile devices. However, most of the existing research works focus on wired networks, where different resources are in abundance and hence performance metrics are abstract. So, results obtained for wired networks can not be directly ported to wireless networks due to the well-known limitations of the wireless medium and mobile devices.

P2P over wireless mobile networks can provide a wide range of services such as sharing files. Where accessing a commercial network is expensive, members of a P2P network can share downloaded objects with each other or even can collaborate to download a large object. This not only provides a cheaper way of sharing resources, but also enables low latency access to remote objects. Dissemination of rescue or strategic information in a disaster or war zone can be accomplished using a wireless mobile P2P network. Some other uses of these networks are short message broadcast, multimedia broadcast, text, audio and/or video based conference.

This research is supported by NSERC grants.

¹In this paper, we use the terms peer, user, member of a P2P network interchangeably

In this paper, we propose an unstructured P2P architecture for dense wireless mobile networks. Unlike structured architecture (for example, Past [1]), unstructured architecture does not have any precise control over resource distribution [2]. Our objective is to reduce the number of wireless link-level message flows to operate the network. This helps in reducing the bandwidth requirement, consuming less energy, and leaving more resources for the application. We employ a location aware cross-layered optimization to achieve this objective. Furthermore, the architecture protocol is computationally and memory requirement-wise cheap. At the same time, we make the architecture flexible for the users and tolerant to the dynamics of a distributed environment.

The rest of the paper is organized as follows. The proposed cross-layered P2P architecture is presented in section 2. The performance measurement from our simulations is discussed in section 3. Finally, the paper is concluded in section 4 along with future directions.

2. THE CROSS-LAYERED P2P ARCHITECTURE

In this section, we propose the peer-to-peer architecture for wireless mobile networks. We first discuss the considered network model in sub-section 2.1. In sub-section 2.2 to sub-section 2.5, we elaborate different components of the peer-to-peer architecture. Finally, in sub-section 2.6, we present the computational and space complexities of the proposed architecture.

2.1. System model

Our system model consists of a set of collaborative computing nodes, each equipped with a wireless interface. Those nodes can form a network on-the-fly using an ad-hoc networking technology [3]. For each node, participation in a P2P network is optional. However, irrespective of its membership in the P2P network, each node will participate in routing messages from one node to another node as a low level service. We also assume that each of the mobile devices has access to some form of location service [4]. Through this location service, a peer can obtain the physical location of itself or other peers.

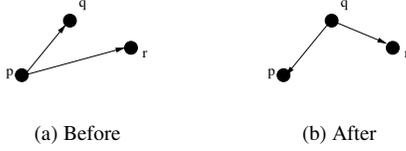


Fig. 1. Reformation (the reform-set from p to q contains r)

2.2. Maintaining the topology

Recently, a randomized P2P network, named *PROOFS*, is proposed to deal with Internet flash crowds [5]. It employs a gossip-like protocol to construct the P2P network. Each peer p of the P2P network maintains a list of neighboring peers, denoted as \mathcal{N}^p . Periodically, p engages in a *shuffle* operation with one of its neighbors, and the choice of the participating neighbor is random. During a shuffle operation, two peers exchange l number of their randomly chosen neighbors with each other, where $|\mathcal{N}^p| > l > 0$.

We introduce a *reformation* operation as the maintenance protocol in our system model. Like shuffle, reformation is also a gossip-like periodic protocol. However, reformation neither chooses a participating peer randomly nor exchanges a random neighbor set with the participating peer. Rather, in the reformation procedure, we choose to consider proximity as a biasing factor. During the operation, p chooses another peer q among its neighbors with the intention of reducing the total distance between peers. Distance between two peers convey the idea of physical distance between them. Here, we expect that hop count among two peers is proportional to the distance between them. In fact, for our system model, the expected number of hops between any two peers is an increasing function of the distance between them [3]. In a P2P network, a peer usually forwards control messages, such as query messages, to its P2P neighbors only. As not all communication nodes participate in the P2P network, a P2P level hop may consist of several link level hops. In a random P2P network such as *PROOFS*, on an average one P2P hop consists of average link level path length of the network. In the worst case, a single P2P hop has a link level hop count of the network diameter. Having a neighbor located at a nearby location results in reduction in number of hops between the peers. This helps in reducing the number of link level messages which helps in reducing the total bandwidth consumption to forward control messages. At the same time, it also reduces message latency.

To have peers located in a close geographic area, we introduce the concept of *distance gain*. During a reformation procedure between peers p and q , if the initiating peer p forwards another P2P neighbor r to q (as shown in Fig. 1, where a directed edge from x to y means that y is a neighbor of x), the distance gain is the reduction of the distances between the pairs p and r , and the second pair q and r . Formally, it is given by: $d_{q,r}^p = |\vec{p}\vec{r}| - |\vec{q}\vec{r}|$. To engage in a reformation process, for

each $q \in \mathcal{N}^p$, at first, p computes the *preliminary reform-set* $\mathcal{N}\mathcal{R}_q^p$ such that $|\mathcal{N}\mathcal{R}_q^p| = l - 1$ and $\mathcal{N}\mathcal{R}_q^p \subset \mathcal{N}^p - \{q\}$. The preliminary reform-set must satisfy that $d_{q,u}^p \geq d_{q,v}^p$, where $u \in \mathcal{N}\mathcal{R}_q^p$ and $v \in \mathcal{N}^p - \mathcal{N}\mathcal{R}_q^p - \{q\}$. Then it computes the net gain for the preliminary reform-set as, $d_q^p = \sum_{r \in \mathcal{N}\mathcal{R}_q^p} d_{q,r}^p$. Finally, p chooses $t \in \mathcal{N}^p$ as the participator of the reformation process where $d_t^p = \max_{q \in \mathcal{N}^p} \{d_q^p\}$. During the reformation, p sends over a reformation request with the reform-set $\mathcal{N}\mathcal{R}_t^p \cup \{p\}$ to t . When peer t receives the reformation request from p , it computes the reform-set for p and then sends the set back to p as a response. Unlike the reform-set from p , the set, computed by t , consists of a list of l peers from \mathcal{N}^t which maximizes the net distance gain for p . After a successful reformation operation, both p and t perform a merge operation as discussed in the next sub-section.

2.3. The merge operation

Without loss of generality, let p be a peer performing a merge operation. \mathcal{N}_{send} and \mathcal{N}_{recv} are the reform-sets that are sent and received, respectively. During the merge operation peer p computes $\mathcal{N}^{p'} = (\mathcal{N}^p - \mathcal{N}_{send}) \cup \mathcal{N}_{recv}$, where $\mathcal{N}^{p'}$ is the new P2P neighbor set of p . Note that, it is certainly possible that $(\mathcal{N}^p - \mathcal{N}_{send}) \cap \mathcal{N}_{recv} \neq \emptyset$. In such cases, $|\mathcal{N}^{p'}| < |\mathcal{N}^p|$. Measures should be taken to carefully handle such cases. This issue is further elaborated in sub-section 2.5.

2.4. Join and leave

When a wireless mobile device wants to participate in a P2P network, at first it acquires an initial neighbor set from one or more of the *known* peers by sending a request to share some of their neighbors. After receiving the request, a known peer responds in the similar way as a reformation participator. The only difference between the response as a known peer and the response as a reformation participating peer is that the known peer does not perform the merge operation while responding to a joining peer. Realization of known peers is possible in several ways. The access points of a WLAN, the Mobile Services Switching Centers (MSC) or the Base Service Centers (BSC) of a cellular network and the fixed mesh routers of a mesh network are some of the possible locations where known peers can be implemented. A returning peer, i.e., a peer that was a member of the P2P network in a near past and is going to join the network again, may decide to use its previous neighbors as known peers. In a critical case where no known peer is available, a joining node can initiate an expanded ring query to discover the nearby peers for the initial neighbor set.

In our architecture, explicit notification of leave is not mandatory and a peer may leave the network asynchronously at any time. A peer eventually discovers an unavailable peer when it initiates a control message exchange procedure (i.e., reformation, search query, etc.) with it.

2.5. Number of P2P neighbors

Unlike PROOFS, in our proposed architecture, we put an upper and a lower bound on the size of the P2P neighbor set. Those bounds are defined as N_{max} and N_{min} , respectively and must satisfy the condition: $N_{max} \geq N_{min} > l$. There are some situations when the neighbor list grows beyond the N_{max} threshold, for example, when a joining peer gathers peers for its initial neighbor set. In those cases, a peer will keep N_{max} number of the nearest peers and discard the rest. Similarly, there are some scenarios where a neighbor list shrinks below the N_{min} threshold, for example, when a neighboring peer fails to respond to a control message. Therefore, the peer requests a widely known repository to share the neighbor list, following the same procedure of a joining peer.

As discussed later, the upper bound N_{max} puts a limit on the worst case computational and space complexity for a peer. The lower bound N_{min} provides robustness for the proposed architecture. By tuning those parameters, the connectivity of the network can be controlled. The gap between N_{max} and N_{min} , provides the architecture different levels of fault tolerance towards reduction of the neighbor-set size, i.e., failure of neighbors. The PROOFS architecture can be mapped into a special scenario where N_{max} and N_{min} are equal. However, this makes the PROOFS architecture totally impractical to be implemented for wireless mobile devices or for systems which allow dynamic join and leave of peers.

2.6. Computational and space complexities

By computational complexity of reformation, we mean the computational complexity faced by the initiating peer. Indeed, the initiating peer incurs more computational complexity than the participating peer. The complexity to find the net distance gain for a specific neighbor is $\Theta(|\mathcal{N}| + (l - 1)) = \Theta(|\mathcal{N}| + l) = \Theta(|\mathcal{N}|)$. For all neighbors, the complexity turns out to be $\Theta(|\mathcal{N}|^2)$. By tracking properly during the previous computations, the neighbor with maximum net gain can be found in $\Theta(1)$ time. Therefore, the total complexity remains $\Theta(|\mathcal{N}|^2)$. The worst case scenario arises when $|\mathcal{N}| = N_{max}$ and then the computational complexity becomes $\Theta(N_{max}^2)$. A peer faces the worst case memory requirement when the neighbor list grows beyond N_{max} and this requirement can be formally expressed as $\Theta(N_{max} + l)$.

3. PERFORMANCE EVALUATION

We develop an event driven simulation tool to evaluate the performance of our proposed architecture. In our simulation, we consider a rectangular area of size 100×100 square unit, where the 5000 wireless mobile nodes are uniformly distributed. The nodes roam around following the random way-point mobility model with zero pause (i.e., the stressed mobility model). Some other parameters of the simulation are presented in Table 1. Note that, if the given radio range is considered to be

Parameter Name	Value
Simulation time	4000 sec.
Warm-up time	2000 sec.
Participation level	30% ~ 80%
Radio range	5 unit
Mobility speed	0.032 unit
N_{max}	30
N_{min}	20
l	5

Table 1. Simulation parameters

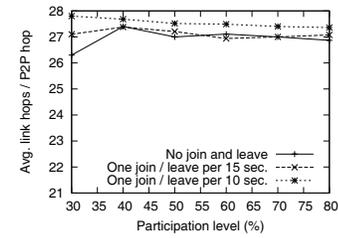


Fig. 2. Number of link level hops per P2P hop with shuffle

250 meters, the mobility speed turns out to be approximately 5.7 km/hr. After the warm-up time, we collect different status from the network at an interval of 10 seconds and finally compute the average. All the results presented here are the average of the ten readings (i.e. with minimum standard deviation) out of twenty simulation runs.

Figs. 2 and 3 show the average number of link level hops per one P2P hop using the PROOFS and the proposed architecture, respectively. In our scheme, as the percentage of wireless nodes participated (i.e., participation level) in the P2P network increases, the number of link level hops per one P2P hop decreases. In fact, as the participation level increases, the chance to find a P2P neighbor at a nearer location also increases. However, if a network uses the (random) PROOFS architecture, this metric remains approximately the same, irrespective of different participation levels. In this case, as the neighbors of a peer are uniformly distributed all over the network, the average link level hop count is not affected at all by the participation level. Indeed, in an ideal situation (which is a true random system with no network dynamics), PROOFS or similar architectures can achieve the best performance where the average length of a single P2P hop is equivalent to the average path length of the whole network. Comparing Figs. 2 and 3, link level hops per P2P level hop is significantly lower in our architecture.

Assume that the underlying routing protocol can deliver a message between two P2P neighbors using the shortest path. However, it does not guarantee that the multi-hop P2P shortest path between s and t will also be the link level shortest path, as a control message is always propagated using the P2P

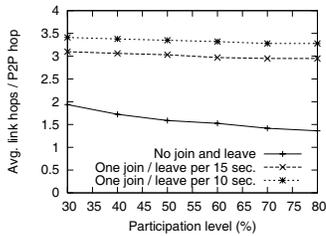


Fig. 3. Number of link level hops per P2P hop with reformation

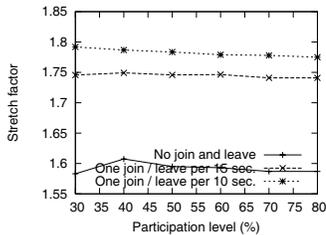


Fig. 4. Ratio of hops in link level shortest path and P2P shortest path

neighbor information, but not using the link level neighbor information. In the best case, those two measurements can be the same and in the worst case, a multi-hop P2P shortest path can be several times of the network diameter. The measurements presented in Fig. 4 is the *stretch factor* of using P2P networks. Stretch factor is defined as the ratio of shortest paths while using the P2P network and the link level network. A lower stretch factor is desired when control messages (for example, query messages) are flooded throughout or part of the P2P network.

It has already been proven that given a connected network, no shuffle operation can make the network disconnected [5]. It also holds for reformation. However, as peers join and leave the network, a P2P network may become disconnected. Mobility deteriorates the scenario, when the underlying link level network is disconnected. During the simulation, the connectivity of the P2P network is computed. If p is a neighbor of q , we consider that q knows about p and vice versa and they are connected in both way. Our simulation results fairly support the previous claim that for almost all the cases more than 90% of the peers remain connected [5]. However, we are more interested about the worst case scenario, i.e., the minimum connectivity. Though we compute the network statistics every 10 second interval, we believe that the presented results are reasonable approximation of the actual results, as they are computed from a very large number of samples. Finally, Fig. 5 shows the minimum connectivity of the network for different join/leave intervals. The results are presented after normalizing in 1. As expected, the minimum connectivity

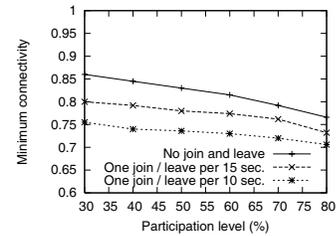


Fig. 5. Minimum connectivity among peers for different join / leave intervals

is fair enough and increases with the degree of participation as well as with the reduction of join/leave frequency.

4. CONCLUSION AND FUTURE WORKS

In this paper, we have proposed a new P2P unstructured architecture for wireless mobile networks. We have used an inexpensive gossip-like protocol to operate the network. Our architecture considers distance between neighbors as a biasing factor and is extremely economical in terms of communication, computation and memory requirement. Furthermore, the architecture helps in reducing the bandwidth consumption and latency of control messages, leaving more space for applications and other services. Currently, an architecture aware search technique is under our investigation.

5. REFERENCES

- [1] P. Druschel and A. Rowstron, "PAST: A large-scale, persistent peer-to-peer storage utility," in *HotOS VIII*, Germany, pp. 75–80, May 2001.
- [2] Q. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks," in *ICS '02: Proceedings of the 16th international conference on Supercomputing*, NY, USA, pp. 84–95, 2002.
- [3] M. Zorzi and R. R. Rao, "Geographic random forwarding (GeRaf) for ad hoc and sensor networks: Multihop performance," *IEEE Transaction on Mobile Computing*, vol. 2, no. 4, pp. 337–348, Oct. 2003.
- [4] M. Pias, J. Crowcroft, S. Wilbur, T. Harris, and S. Bhatti, "Lighthouses for scalable distributed location," in *Second International Workshop on Peer-to-Peer Systems*, pp. 278–291, 2003.
- [5] A. Stavrou, D. Rubenstein, and S. Sahu, "A lightweight, robust P2P system to handle flash crowds," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 1, pp. 6–17, Jan. 2004.