Lobby Influence: Opportunistic forwarding algorithm based on Human social relationship patterns

Sardar Kashif Ashraf Khan, Raul J Mondragon, Laurissa N Tokarchuk
School of Electronic Engineering and Computer Science
Queen Mary University of London
London, United Kingdom
{sardar.khan, raul.mondragon, laurissa.tokarchuk}@eecs.qmul.ac.uk

Abstract—In opportunistic networks, nodes have no prior knowledge of the routes to forward the information to its intended destination. Message transfer without generating excessive traffic poses a challenge in these networks. Improvements in message routing/forwarding in opportunistic networks can be achieved by exploiting nodes that exhibit a high level of popularity or influence within the network. Based on these observations, this paper presents a novel algorithm called Lobby Influence. The new algorithm is tested against two previously proposed algorithms and proves to be the better algorithm in terms of message delivery and delay. Moreover, unlike other social based algorithms, which have a tendency to concentrate traffic through their identified routing nodes, our algorithm provides a fairer distribution, thus alleviating the tendency to saturate individual nodes.

Keywords—Social Networks; forwarding algorithm; Opportunistic routing; Adhoc communication; Human relationships

I. INTRODUCTION

In opportunistic networks, it is common that there is no prior information how to deliver messages between individuals. The easiest way is to broadcast message throughout network, but this kind of communication has severe consequences in terms of resource usage. A node in a broadcast network can get irrelevant messages over and over again and this may fill the buffer of the node and force the device to shut down. An improvement to this method is to use an epidemic algorithm [6], where nodes that have received the broadcasted information are considered “infected” and they do not retransmit the message. Although this algorithm stops duplicate packets in the network but still consumes too many resources.

Social based algorithms can be used to reduce the unnecessary resource consumptions in opportunistic networks. These methods are based on the assumption that there are individuals in the network that can play a central role in the delivery of information. For example, popular individuals encounter many other individuals; therefore there is a good chance that they will meet the destination individual or someone that knows the destination individual. In order to identify these popular nodes that can be used as reliable message forwarders in the network, centrality measurements are used. Over the years three measures have become the standard for centrality measure in networks: betweenness [1], degree [2] and closeness [3]. Based on these centrality measures, many researchers [4, 5, 7] have proposed new interesting techniques to deliver information based on social interaction. Bubble Rap [4] is an algorithm that overcomes some of the overutilization of resources when using an Epidemic algorithm. Bubble Rap is based on the social relationship between the network individuals. It exploits popular individuals by using them as efficient distributors of information. Bubble Rap has the disadvantage that the resources of the popular individuals can be depleted very quickly.

In this work we present an opportunistic routing algorithm, Lobby Influence (LI), which utilises both the popularity of a node and the popularity of the node’s neighbours. This is based on the observation that seemingly un-popular nodes could have a high degree of popular neighbours and are thus, good candidates for information transmission. By targeting such nodes, high access/reach or knowledge of other relevant nodes in the network can be achieved. These nodes can ultimately help to deliver the messages to intended destinations with significant amount of reduction in communication overhead. This new algorithm is tested against two well-known algorithms Epidemic [6] and Bubble Rap (BR) in working day model (WDM) scenario [11]. Simulation results show that LI outperforms BR and Epidemic routing algorithms in terms of delivering messages at destinations with justifiable communication costs.

The paper organised as follows: Section II provides the related work. Section III provides more details the concept and operation of the Lobby Influence algorithm. Section IV provides the experimental results and discussion. Section V gives our conclusions, along with a discussion of future work.

II. RELATED WORK

Epidemic routing [6] has significantly reduced the number of duplicate packets, but it still relies on broadcasting and sends irrelevant packets everywhere other than intended destination. To address this problem, Hui et al presented a social based forwarding algorithm known as Bubble Rap in [4]. Bubble Rap (BR) forwarding is based on the theory of community and node centrality: “forward
messages to those nodes that are more popular than the current node" [4]. Taken from the analogy of person popularity in his/her social circle such that if a person is considered more popular in a given community, is more likely that information can be transmitted to other members through that person. This message forwarding method tends to direct the traffic towards its intended destinations and also reduces redundant traffic in the network. However, BR has its own concerns; it increases the load of popular nodes in the network and the buffer on these nodes can overflow very quickly.

The BR proposed that targeting key nodes in the opportunistic networks improves the routing efficiency. However, there are other algorithms that define key nodes in different prospective. For instance, Korn et al [5] presented metric known as Lobby Index. A node has high lobby index if its neighbours have at least equal or more neighbours than the node itself. Our LI algorithm is based on the lobby index which can also be formulated in terms of the diplomat’s dilemma [7]; i.e. a person has a strong influence in a society if he has relations with people. A diplomat has a high influence in a society because his connections are primarily with influential members of society. As a result, he has high reach in society with minimum effort of making personal relations (more power less connections/low cost). In real world analogy, a company boss does not need to know every person in his large company. If he has some query for a specific employee, he can pass the message to the manager of employee’s department who will eventually deliver the message to the intended employee.

The rest of the paper presents our algorithm Lobby Influence that combines the characteristics of key nodes as proposed in [4] and [5].

III. OUR WORK

A. Lobby Influence Forwarding Decisions

The forwarding decision of Lobby Influence not only depends on the node popularity (Bubble Rap) but also on the strength of the neighbourhood relationships (Lobby Index). In order to illustrate how this new algorithm increases the probability of message delivery while also keeping the cost of popular nodes low, consider the situation ‘A’ in figure 1. If only Bubble Rap communication is present, the most popular nodes will be used for message transfer. In this situation, node ‘1’ is more popular than node ‘6’ and thus node ‘1’ will not transfer message to node ‘6’ because node ‘6’ has low popularity level. Node ‘1’ will keep that message until it finds suitable node or time to live (TTL) of message expires. However, as depicted, node ‘6’ has high neighbourhood relationship in current network and thus transfer to this node would increase the probability of message transmission to its destination. Lobby Influence addresses this issue and allows more popular nodes to transmit message to less popular nodes provided that node has a strong neighbourhood relationship (high lobby index) in the current network. Apart from increasing the probability of message delivery, by directing transmission towards highly influential nodes (rather than just popular nodes) LI also helps alleviate pressure on the most popular nodes.

Now consider situation ‘B’, if only Lobby Index based network communication is present. Node ‘26’ has high lobby index and it is well-connected with its neighbours, however node ‘26’ is not able to forward messages because it could not able to locate a suitable node in its network that can take information to the destination, and will therefore wait for a suitable node to connect to the network or the TTL of the

Figure 1. Sketch of Lobby Influence
message expires. Lobby Influence addresses this issue by allowing high lobby index nodes to transfer messages to those nodes which are more popular, thus allowing messages to keep on forwarding until the destination is reached. Therefore, using LI node ‘26’ will transfer the message to node ‘29’, as although it has low lobby index, it has a high popularity level. This significantly improves the probability of message delivery over the use of Lobby Index alone.

In summary, LI addresses the shortcomings of both Bubble Rap and Lobby Index algorithms. In the case of Bubble Rap, by allowing a message to be transferred to a node with high lobby index irrespective of popularity and in the case of Lobby Index, allowing a well-connected node to transfer to a less well-connected, but popular node. This new strategy has not only increased the probability of the message delivery but also decreased the overall delay and reduced load on important nodes present in the network.

B. The Algorithm

The Lobby Influence is the extension of Bubble Rap algorithm, therefore the Label scheme [9] used in BR has been adopted. Each node is given a label which describes the node’s associativity with a community (a node may have associativity with different communities thus can have multiple labels). Those nodes will be targeted for message delivery whose community association is similar to the destination node or keep on forwarding the message until member of targeted community is encountered.

For this algorithm, three assumptions are considered:

- Each node must have a label to show its association with at least one community.
- Each node has one global rank and one local rank to define its global centrality (popularity) across the whole system and local centrality within its local community, respectively. A node may have multiple local ranks due to its association with different local communities, respectively. They may have multiple labels.
- Each node has its lobby index to indicate how well a node is connected to its neighbours in the current network.

The algorithm shown in Figure 2 works in such a way that a node may come across two situations, a) Node within a local community b) Node within Global system.

a) Node within a local community: The first part of algorithm deals with the situation, where a node is present inside a local community. This means that the label of the current node is equal to the label of the destination node, the local rank (popularity) and lobby index will be used to make the message forwarding decision. If the encountered node has high lobby index irrespective of its rank (either high or low) the message will be transferred. However, if the lobby index of the encounter node is similar to the current node then the message will be transferred based on high rank. If none of these conditions are met, the node will keep the message until it finds some suitable node or the TTL of message expires.

b) Node within Global system: The second part of algorithm deals with the situation, where a node is looking for a destination at the global level. At this level the source node or a rely node is looking at the message until it finds a suitable node which belongs to the same community as the destination node. Here we assume that whenever a node finds a member of the destination community, the current node will transfer the message to the encountered node and remove the original message from its database, this last step is to reduce the resource utilisation, with the expectation that, inside the community, relay node will use the local strategy as described in a). In order to forward the message to a suitable node at the global system level, when a node meets another node, the message will transfer to the encountered node if the lobby index of the encountered node is high (irrespective of global rank). However, if the lobby index of the encountered node is similar to the current node the message transfer will occur on the basis of global rank (popularity). If none of the above conditions are met then the source node will keep the message into its database until if finds suitable node or remove it if TTL of the message expires.

```java
Begin
Foreach (encounteredNode_i) do
  if (getLabel(EncounteredNo_i) == getLabel(Destination)) &&
    getLobbyInfluenceLocalStatus(EncounteredNode_i) == true)
    EncounteredNode_i.addMessageToBuffer(message);
  else
    if (getLabel(EncounteredNode_i) == getLabel(Destination)) ||
      getLobbyInfluenceGlobalStatus(EncounteredNode_i) == true
    EncounteredNode_i.addMessageToBuffer(message);

End
getLobbyInfluenceLocalStatus(EncounteredNode_i) == true)
  if (getPopularityLocal(EncounteredNode_i) == high)
    getLobbyIndex(EncounteredNode_i) == Equal)
      return true;
  else
    return false;
getLobbyInfluenceGlobalStatus(EncounteredNode_i)
  if (getPopularityLocal(EncounteredNode_i) == high)
    getLobbyIndex(EncounteredNode_i) == Equal)
      return true;
  else
    return false;

Figure 2. Lobby Influence Algorithm
```

IV. RESULTS AND DISCUSSION

A. Simulation Setup

To evaluate the algorithm, the Opportunistic Networking Environment (ONE) [10] simulator was used, which is specifically designed for delay tolerant networks. For this
simulation, Bluetooth-enabled mobile phones or similar devices are considered. Bluetooth is the communication channel between mobile users. Most of the devices are operating at 2 Mbit/s data rate with 10m range. The different algorithms (Epidemic, BR and LI) are evaluated by varying the queue size of the nodes. Queue size affects the overall performance in network communication; as queue size increases, the number of messages delivered at destination increases as well. This is natural; with large queue size more messages can be stored and thus there is less risk of dropping due to queue over-flow.

Table I: District Settings

<table>
<thead>
<tr>
<th>District</th>
<th>Nodes</th>
<th>Offices</th>
<th>Meeting Spots</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>30</td>
<td>30</td>
<td>4</td>
</tr>
<tr>
<td>B</td>
<td>30</td>
<td>10</td>
<td>1</td>
</tr>
<tr>
<td>C</td>
<td>30</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>E(A &amp; B)</td>
<td>30</td>
<td>20</td>
<td>2</td>
</tr>
<tr>
<td>F(A &amp; C)</td>
<td>30</td>
<td>30</td>
<td>4</td>
</tr>
</tbody>
</table>

Table II: Algorithm Settings for BR and LI

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Community detection algorithm</td>
<td>K Clique</td>
</tr>
<tr>
<td>K</td>
<td>3</td>
</tr>
<tr>
<td>Familiar threshold</td>
<td>700</td>
</tr>
<tr>
<td>Centrality algorithm</td>
<td>C window</td>
</tr>
<tr>
<td>Centrality time window</td>
<td>3600 (s)</td>
</tr>
<tr>
<td>Computation interval waiting time</td>
<td>300 (s)</td>
</tr>
<tr>
<td>Number of time intervals to average</td>
<td>3</td>
</tr>
</tbody>
</table>

In order to fully justify the algorithms performance, it is very important that the mobility of nodes should be very close to the real world. This simulation uses a scenario known as working day movement model [11]. In this model a map of Helsinki city is presented where its central areas are divided into 4 artificial districts A,B,C,D and two overlay districts E and F, please see [11] for more details. For this simulation we consider districts A, B and C. District A is the busiest district which is connected with districts B and C through overlay districts E and F respectively. Different offices, shopping areas and meeting spot are present in those districts. Nodes go to meetings points using either their own car or by taking buses. This model approximates real world scenarios in which different nodes have some chance of meeting with each other and similarly, it allows nodes which are far apart from each other to be connected through intermediate nodes. Table I shows the assignment of nodes, offices and meeting spots to the respective districts.

Table III: Parameters Used in WDM Scenario

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>World's size for Movement Model</td>
<td>10000 X 8000m</td>
</tr>
<tr>
<td>Total simulation time</td>
<td>57000s</td>
</tr>
<tr>
<td>No. of Hosts [pedestrians, buses]</td>
<td>[150, 10] = 160</td>
</tr>
<tr>
<td>Message TTL (time to live)</td>
<td>960 mins</td>
</tr>
<tr>
<td>Time to move nodes in the world</td>
<td>7200s</td>
</tr>
<tr>
<td>before real simulation commence</td>
<td></td>
</tr>
<tr>
<td>Nodes speed [pedestrians, buses]</td>
<td>[0.5-1.5, 7-10] m/s</td>
</tr>
<tr>
<td>Nodes pause time [pedestrians, cars, trans]</td>
<td>[0-0, 10-30] s</td>
</tr>
<tr>
<td>Message sizes</td>
<td>500KB-1MB</td>
</tr>
<tr>
<td>Message creation interval</td>
<td>15-25s</td>
</tr>
<tr>
<td>Air data transmit speed</td>
<td>250kbps</td>
</tr>
<tr>
<td>Transmit range</td>
<td>10m</td>
</tr>
<tr>
<td>Working day length</td>
<td>28800s</td>
</tr>
<tr>
<td>Probability to go shopping after work</td>
<td>0.5</td>
</tr>
<tr>
<td>Own car probability</td>
<td>0.5</td>
</tr>
<tr>
<td>Range of message source/destination addresses</td>
<td>0-159 nodes</td>
</tr>
<tr>
<td>Queue sizes</td>
<td>0, 20, 40, 60, 80, 100, 120, 140, 160, 180 in MB</td>
</tr>
<tr>
<td>No. of each experiment runs</td>
<td>10</td>
</tr>
</tbody>
</table>

This simulation considers activities of nodes for one day. The length of the day is approximately 16 hours because after that a node assumes to be at home. The working day length is 8 hours and probability to go out for even activity is 0.5. Every district has its own route with 2 buses each and half of the nodes can use cars. Table III summarize the parameters used in this experiment.

B. Results

Ten experiments were performed for each queue size in the WDM scenario; each one using different seeds. For these experiments, a total 24 dual core linux operating system computers are used with 8 GB RAM each. The algorithms performances are measured against three metrics: 1)
Message delivery: how many packets are received at destinations, 2) Delays: how quickly packets arrived at their destinations 3) Forward messages: the cost in terms of number of exchanged messages between nodes, which ultimately effect the utilisation of system resources (bandwidth and energy) utilisation. Each of the graphs showed here contains 3 curves which represent the behaviour of BR, epidemic and LI algorithms, respectively. The X-axis in each graph represents different queue sizes, against which the performance of each algorithm is measured.

Figure 3. Average delivery rates in WDM mobility scenario

Figure 3 shows the normalized delivery rates in WDM scenario. The graph shows that LI outperforms BR in terms of message delivery. Whereas, Epidemic has poor efficiency because it uses flooding and drops many packets when the queues are full or the TTL of the messages expires. BR relies on popularity of nodes and therefore the current node will not forward messages until it finds a more popular node than the node itself. As LI exploits both popular nodes and highly influential nodes has much better delivery ratio.

Figure 4 shows average latency experienced by the messages when using different algorithms. Epidemic has the highest latency because most of the nodes exhausted their queue size by accepting unnecessary messages due to lack of message forwarding criteria as a result more messages dropped. LI again proves to be the quickest one in this scenario; again this is because it keeps on forwarding message towards destination by exploiting more popular nodes or influential nodes with high connections.

Figure 4. Average latency in WDM mobility scenario

Figure 5. Average no. of forwarded messages in WDM mobility scenario

Figure 5 shows the number of messages forwarded during the simulation in normalized form. The graph shows that Epidemic proves to be very costly compare to BR and LI, simply because epidemic floods the whole network by passing messages to any new node it encounters. However, in case of BR and LI both use specific criteria to select a node before forwarding the message. LI has more overhead than BR because it uses influential nodes as well as popular ones. Whereas, BR only relies on most popular node, so naturally LI experiences more communication cost.

Figure 6. Load on top five popular nodes in WDM scenario

Figure 6 shows the message load on the queues of top five popular nodes and their mean by the end of simulation time. The graph shows that LI has significantly reduced the loads on queues of popular nodes compare to the BR algorithm. Because in LI, nodes can forward messages to either more popular nodes or more influential nodes, their queues are available to accept more messages that needs to be forward. In case of BR, nodes forward messages only to the more popular nodes; this will creates pressure on popular nodes and their queues. Queues fill more quickly as a result message loss could increase due to less space in the queue.

Figure 7 shows the message delivery probability i.e. the ratio of number of messages delivered at the destinations to the number of messages created at sources. WDM resembles to real life, so nodes tend to move in specific patterns, frequently meeting with other nodes as they go into offices or spending time in some evening spot. In such kind of scenario, LI again proves to provide best results compare to the BR and epidemic routing protocols. The figure shows that LI has 10% improvement in terms of message delivery compare to the BR algorithm.
V. CONCLUSION AND FUTURE WORK

Simulation results have shown that the observations we presented in this work based on which Lobby Influence takes forwarding decisions proved to be true. Lobby Influence not only targets popular nodes but also un-popular nodes with popular neighbours. This makes un-popular nodes take responsibility and forward messages to neighbouring nodes with high connections. As a result, high reach/access in the network is achieved. Lobby Influence outperformed Bubble Rap and Epidemic routing algorithms in terms of message delivery and speed. Although LI has a little higher communication cost compare to BR (which is natural outcome of this algorithm), but at the same time it significantly reduces load on most popular nodes in the network.

Until now we have implemented this algorithm in working day model, which is synthetically developed by ONE simulator team. In future, we will test this algorithm against other synthetic models to further test its robustness in different scenarios. It would also be very interesting to see how this algorithm will behave when tested against real world mobility traces such as MIT reality mining or Haggle project.

REFERENCES