American Journal of Engineering Research (AJER)
e-ISSN : 2320-0847 p-ISSN : 2320-0936
Volume-03, Issue-02, pp-08-21
www.ajer.org

Research Paper Open Access

Mobile Wireless Peer-To-Peer Network with Higher End-To-End Throughput And Lower End-To-End Delay

Md. Sajjatul Islam
Computer Science and Engineering (CSE)
East Delta University (EDU), Bangladesh

Md. Zainal Abedin
Computer Science and Engineering (CSE)
University of Science and Technology Chittagong (USTC)
Bangladesh

Sudam Das
Information Technology Basic Bank Ltd.
Bangladesh

***Abstract:* -** A mobile wireless Peer-to-Peer network (or MANET) is a collection of mobile, wireless nodes which together form a network without any fixed infrastructure or centralized administration. A MANET operates without base stations, where a node communicates directly with nodes within wireless coverage and indirectly with all other nodes using a multi-hop route through the other nodes of the MANET. In this research, a higher through-put but low-cost unstructured platform for wireless mobile Peer-to-Peer network has been proposed. The platform addresses the constraints of expensive bandwidth of wireless medium, and limited memory and computing power of mobile devices. It exploits an inexpensive gossip protocol as the main maintenance operation and uses (re)configuration algorithms which use the limited resources of the network in an efficient way, and thus improving the performance of the network. Simulation results show that this proposed platform reduces the average delay and increases average throughput. As a result the platform provides a higher through-put but low cost framework for different Peer-to-Peer applications.

Keywords: *- Delay, Peer-to-Peer, Performance, MANET, Throughput*

I. INTRODUCTION

A distributed computer system or architecture is called peer to peer that is basically intended to enable computer for sharing resources in such a way that they can exchange data directly without the help of any centralized system or server. This architecture is very reliable as it does not depend on central database rather every nodes are capable of providing necessary information to keep running the whole network by maintaining connectivity with each other and holding performance. P2P network or architecture has become popular because of its enormous applications and ability to share multifarious information or data. P2P network or system works extensively on some factors that prevent scalability and performance of the network. It maintains every disk spaces for efficient storage of information and utilizes bandwidth for transforming or exchanging them across the nodes within the system. These enable an efficient network without a central and high cost server where if the server is down in any case the whole network is useless. Though the Peer to Peer system has high prospects, it is not fully perfect and only networking system or architecture because of its some limitations. Those limitations need to be addressed to get wide acceptance as a reliable and high performance networking system. This paper is dedicated to propose a mobile wireless Peer to Peer network with negligible cost by considering the constriction of expensive bandwidth of wireless medium, less energy consumption in mobile wireless systems or networks as well as limited memory and computing power of mobile devices. To demonstrate the performance of the proposed network, several simulations have been carried out with the help of well known simulators such as NS-2 as well as some software such as MATLAB and Microsoft Excel.

II. BACKGROUND

Peer to Peer network has excellent future prospects as it is scalable and has numerous applications. Peer to Peer network operates within distributed application layer. In this network, peers work together in a goal oriented which is totally absent in client server based network. To achieve excellent performance from Peer to

Peer network, a cohesive logical structure among the participating peers has been established. Peer-to-Peer (P2P) networks have received attention because of their broad applications. P2P network is a distributed application layer network. Unlike traditional client-server based applications, peers in P2P network collaborate to achieve a certain goal. Design of a good P2P network requires a well thought logical structure among the participating peers. Structure-wise, P2P networks can be divided into three categories: (1) centralized, (2) decentralized but structured and (3) decentralized and unstructured [1]. In a centralized network, peers rely on a central host for a few services. As a result, the central host is subject to a single point of failure. The decentralized but structured network has been well studied and is still a very active area of research. The topology of the members in such a network is ruled by several constraints. Contents are distributed among the members using either some hints [2] or the topology of the members [3, 4]. The decentralized and unstructured network has neither any central host to provide some crucial services nor any precise control over resource distribution. The topology can be constructed using some knowledge about the physical or logical properties of the underlying physical network but, unlike the structured network, it puts no constraint on content distribution. A well known example of such a network is Gnutella.

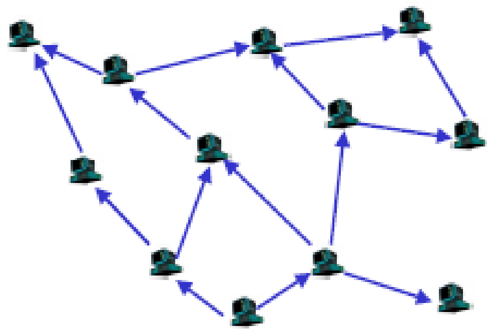


Figure 1. Content location in Gnutella.

Fig. 1 illustrates how content is located in Gnutella. A query initiated by the peer at the bottom is flooded to all peers in the system.With the exploitation of high bandwidth 3G (and expected deployment of 3.5G and 4G) cellular networks and wireless LANs, there is an increasing interest in wireless P2P networks. However, results obtained for the wired networks cannot be directly deployed in the wireless P2P networks due to the limitations of the wireless medium, expensive bandwidth, and the limitations of the mobile devices due to small memory and limited computation power. Significant research efforts have been put recently to use P2P networks in wireless environment to tackle management problems such as, routing [5], clustering[6], service discovery [7], and applications such as, multimedia distribution [8],file sharing [9] etc.

Inexpensive Peer-to-Peer Subsystem (IPPS) [10] is a location aware P2P network platform for dense wireless mobile networks. IPPS is unstructured in nature. It reduces the number of link level message flows in the network, and consumes less energy. IPPS employs a gossip protocol to maintain the P2P network and uses information about the locations of the neighbors to minimize the number of hops between peers. At each gossip iteration, a peer builds up neighborhood relation with nearer peers and discards the peers at distant. Furthermore, the IPPS platform is cheap in connection with computation and memory requirement.

In mobile wireless network, the entire available channel capacity may not be available to a wireless application, and the actual throughput is also determined by the forwarding load generated by other wireless nodes. Besides, mobile devices are battery operated. Unlike electronics, advances in battery technology still lag behind. Minimizing the number of link-level wireless hops helps in increasing the capacity available to the applications. Reduced number of link-level hops also means less number of transmissions and less power consumption for a mobile node. Along with being thrifty about bandwidth consumption, a suitable application for mobile devices is required to be computationally inexpensive to ensure prolonged battery life, and memory requirement-wise economical to confirm accommodation in the small system memory.

In spite of the limitations of wireless mobile networks, P2P over high capacity cellular networks and wireless LANs can provide a wide range of services such as sharing files [9]. In scenarios 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 popular 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 mobile wireless P2P network. Short message broadcast, multimedia broadcast, text, audio and / or video based conference are some other examples. Recently, large numbers of research articles on P2P networks have been appeared in the literature. At the same time, several implementations of different P2P became available for the users. Some of the highly structured P2P networks are CAN [3], Chord [11], Past [4], SCRIBE [12] and Tapestry [13].Those networks employ specific resource placement algorithms which are tightly coupled with the P2P topology. To retrieve or query for a resource, they use topology specific (in turn, resource distribution specific) routing mechanism. As a result, a search can be performed very efficiently in this type of networks. However, if the identification of a resource is partially available (i.e., not all properties of the Meta data are available), a search fails. Moreover, due to this impractical assumption about the resource distribution policies, those networks have not been widely deployed. Freenet [2] and Tarzan [14] are examples of loosely coupled distributed P2P networks. Some of those networks use a centralized directory which is not robust. Others use hint-based resource distribution which cannot support searching of objects whose information is partially available. Besides simple resource sharing, some of the loosely coupled P2P networks have considered issues like trust and security.

Some researches proposed to use existing or modified structured networks in wireless and ad-hoc networks. For example, XScribe [15] is modified from SCRIBE [12] to suite in ad-hoc networks. In general, structured P2P networks mandate that all the peers in the network fully conform to the system requirements. To satisfy that condition, all the peers must abide by the rules set by the administrative body. However, it is very difficult to achieve such a goal in a highly distributed environment. As a result, structured P2P networks are not able to gain popularity for resource sharing in an environment without any central administrative control such as, the Internet. However, success has been reported in developing large scale distributed storage system [16], scalable publish/subscribe system [12], and application level multicast or broadcast protocols [12].

On the other hand, a structured P2P network faces a high cost of maintenance of the network and the ability of this type of networks to work with extremely unreliable environments has not yet been investigated. On the contrary, an unstructured P2P network is a low cost network which can sustain any extreme environment [17]. Although such a benefit can be achieved at the expense of higher search cost, the network assumptions and the overall gain have made this kind of P2P networks so attractive that several unstructured P2P networks have been deployed and are being used by huge user communities. For instance, an unstructured P2P network, named PROOFS [17], has been proposed to share hot Web content. The heart of PROOFS is a periodic gossip protocol, called shuffle, where two random neighboring peers rearrange their P2P neighbor sets through an exchange of randomly selected neighboring peers. Though the shuffle operation is simple and inexpensive, query success rate for popular objects is excellent (more than 95%). With a strong theoretical background, PROOFS is an excellent unstructured P2P network for wired systems where computing power and network bandwidth are ample, and changes to the membership of the P2P network are rare. With the limitations of wireless medium and mobile devices, and dynamic join and leave of the mobile peers in the P2P networks, the benefit of randomness in PROOFS diminishes. In a wired network, due to the abundance of resources, performance metrics of many applications are abstract. However, P2P networks in wireless mobile environment should be very economic about the resources of the wireless medium and devices.

III. SYSTEM MODEL

The proposed system model consists of a set of collaborative computing nodes, each equipped with a wireless interface. It is believed that nodes can form a network on-the-fly using an ad-hoc networking technology.

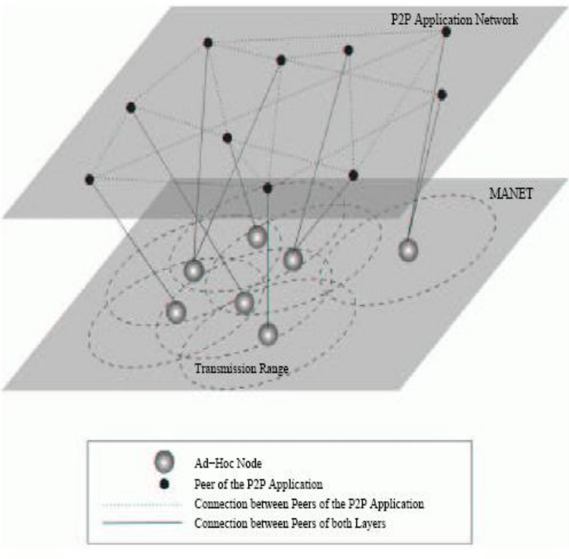


Figure 2. A diagram of a P2P application over a MANET [18]

In this research, we consider GeRaf [19], an efficient location aware transmission (MAC) and forwarding (routing) scheme, to manage the network. In our model, for each node, participation in the P2P network is discretionary. However, irrespective of its membership in the P2P network, each node participates in routing messages from one node to another as a low level service. It is being assumed that the network is equipped with low level (lower than application level) point-to point unicast primitives, and each of the devices mobile has access to some forms of location service [20]. Through this location service, a node in the network can obtain the physical location of itself or other nodes. The information from the location service is used by the lower level network management (i.e., GeRaf) as well as by the P2P modules. As a result, either the network management modules expose interfaces to share the location information or can be combined with the P2P modules as a cross-layer application.

3.1. Multi hop cellular networks (MCN)

In the field of mobile communications, an important model is multi hop ad hoc networks [21], unlike traditional single hop cellular networks (SCN); this paradigm is based on the infrastructure which is less, self-organizing and rapidly deployable without any site planning. In multi hop ad hoc networks consisting exclusively of mobile users (also known as mobile ad hoc networks or MANETs), each node relays the packets that need multiple hop transmissions, and other nodes towards their destination by acting as an intermediary station. Otherwise they can't be reached with single-hop transmission. The real world MANETs which are available today, are deployed widely based on the ad hoc mode of IEEE 802.11 standard for wireless local area networks(WLAN) with the high data rate of 54 Mbps for internet access and multimedia applications. Another concept of MCN was first proposed by Lin and Hsu [21] as an architecture that would preserve the benefits of traditional SCNs with infrastructure and incorporating the flexibility of ad hoc networking. To simplify the concept of MCN in this paper, we would introduce MCN as a general networking paradigm that combines the traditional SCNs and ad hoc networks by means of multi hop transmission, unless we specifically refer to the MCN architectures proposed in [21].

The underpinning idea of using a multi hop communication is to split an original long communication link into two or more shorter links and it thus results less requirement of transmission power of each node participating in the communication scenario. Also this reduction in transmission power could also lead to a lower interference level and shorter frequency reuse distance. In addition, the need for short-range transmission in MCNs opens the possibility of using other higher data rate wireless technologies such as IEEE 802.11, Bluetooth, or Ultra-Wideband (UWB), in conjunction with the cellular technology.

3.2. Comparison between MANETs and P2P application networks

Mobile ad hoc network is characterized as multi-hop wireless communications among mobile devices. As P2P applications and mobile ad hoc networks follow the same model, some aspects of both are common but have some jarring contrasts. General differences between both technologies are described in TABLE 1.

Table 1. Differences between P2P application networks and MANETs

	P2P Network	**MANET**
Motivation for creating the network	Create a logical infrastructure to provide a service	Create a physical infra-structure to provide connectivity
Connection between two nodes	fixed medium and direct	wireless and indirect
Connection confidence	high (physical connections, many paths)	low (wireless connections)
Peer location	any Internet point	restricted area
Structure	physical apart from logical structure	physical structure corresponds to logical structure
Routing	only reactive algorithms possible, reliable algorithms not implemented yet	reactive, pro-active and reliable algorithms exist
Peer behaviour	Fixed	mobile
Broadcast	virtual, multiple uncast	physical, to all nodes in transmission range area

3.3. Peer-to-Peer multi hop cellular

Mobile ad hoc and Peer-to-Peer networks have encountered two challenges in common: peer detection and packet routing. To mitigate these challenges, synergies between P2P networks and MANET can be used to reduce the administrational effort and to increase the performance and reliability of mobile peer to peer (MPP)

[22]. The MPP protocol suite comprises the MPP protocol as the application layer protocol, the Mobile Peer Control Protocol (MPCP) as the interlayer communication protocol and EDSR as the network routing protocol. MPP plays the roles of file transfers within the P2P network and resides in the P2P client application. MPP also gets the data upload and download facilities through use of HTTP. A communication channel, the MPCP, between the application (MPP) and the network layer (EDSR) is needed to exploit the P2P functionality in the network layer. Fig. 3 shows the layer model of the mobile P2P network.

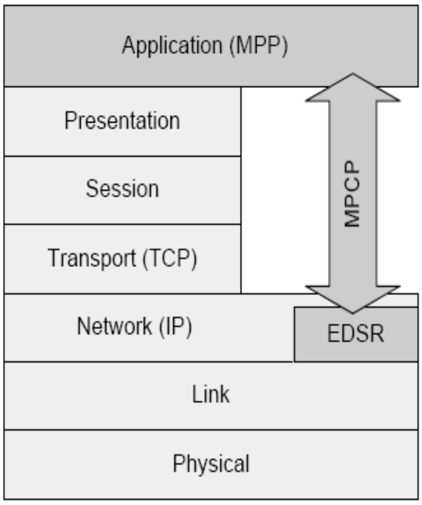


Figure 3. Network layer and protocols [22].

3.4. Physical and network layer significance of Peer-to-Peer multi hop cellular

Initially, the P2P application on the mobile device announces itself to the EDSR layer with the MPCP. Fig. 4 shows the process of searching and transferring files within the mobile Peer-to-Peer network as a message sequence chart. MPCP forwards the request to EDSR which transforms it into a search request (SREQ) upon the initialization of data search. As with DSR route requests (RREQ), EDSR floods SREQs through the MANET. EDSR nodes receiving the request, forward the request to the registered P2P application over MPCP. Consequently the P2P application can determine data satisfying the request's criteria. If a file on the node matches to a request, the application initializes an EDSR file reply to be sent back to the source node containing all necessary information for the file transfer. In a similar way to DSR route replies (RREP), a file reply (FREP) includes the complete path between source and destination.

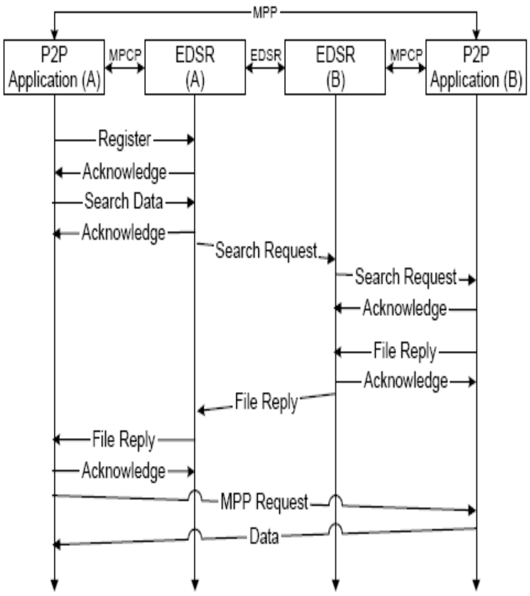


Figure 4. Message sequence chart for data search and download process in the mobile P2P network[21]

3.5. Optimizing P2P over MANETs

Our proposed modifications to the unstructured approaches are proposed to this section. As mentioned before, the performance of the unstructured P2P approaches depends on the amount of messages sent. As noted by J. Borg, content searching in P2P is similar to routing in MANETs [23]. So, we used concepts of routing to improve unstructured P2P networks. A classic routing algorithm named Gossiping is introduced where each node forwards its requests to each of its neighbors according to a pre-defined probability [24]. This operation significantly reduces energy consumption and network load due to a smaller number of messages sent, however the number of hops travelled increases [25]. Reconfiguration of gossip protocol by exchanging a number of neighbors between peers will make neighbors closer to each other. The concept of exchanging neighbors follows from the shuffle operation of PROOFS [17]. It is done to provide randomness in the network, whereas, reconfiguration makes attempts to being neighboring peers closer to each other. We hope that reconfiguration reduces link level hop count between neighboring peers. Firstly if a neighbor is located at a nearby location then it results in reduction in number of hops between the peers. So that helps in reducing of number of link level messages which helps in reducing the total bandwidth consumption to forward P2P messages. Furthermore, lower hops mean reduced message latency. As reduced number of link level messages slows down energy consumption and boosts battery life of mobile devices and these two properties are appeared extensively in wireless mobile application.

Indicate the nodes in the network as *v*1, *v*2, *vNtotal*. Distance between nodes *vi* and *vj* can be defined as *Dij* = |*dist (vi, vj)*|. Let *E*[*nij*] is the expected number of link level hops between *vi* and *vj*. From [19], it can be shown that,

$$\frac{D_{ij}/R-1}{E[\zeta(D_{ij}/R)]}+1 \leqslant E[n_{ij}] \leqslant \frac{D_{ij}/R}{E[\zeta(1)]}+1$$

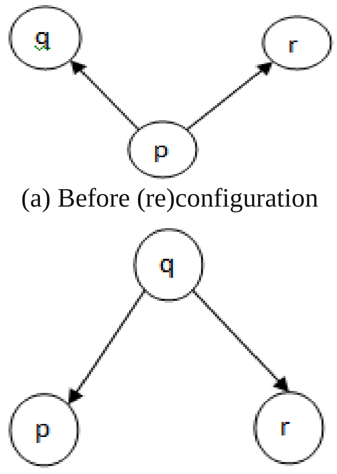
Where *R* is the radio range of a node. *E* [*ζ (D)*] is the expected one hop advancement towards the destination where D (expressed in unit radio range) is the distance between the current and the destination nodes. The average link level hop count can then be defined as,

$$\frac{1}{N_{total}(N_{total}-1)}\sum_{i=1}^{N_{total}}\sum_{\substack{j=1\\ j\neq i}}^{N_{total}} E[n_{ij}]$$

We introduce the concept of **'distance gain'** to have peers located at a close geographic area. During a (re)configuration procedure between peers *p* and *q*, if the initiating peer *p* forwards another P2P neighbor *r* to *q*, the distance gain is the reduction of the distances between the pairs *p* and *r* and the second pair *q* and *r*. Fig. 5 shows a (re)configuration step where a directed edge from any peer *x* to another peer *y* means that *y* is a neighbor of *x*. Now, the distance gain is formally given by:

$$d^{p}_{q,r} = |dist(p,r)| - |dist(q,r)|$$

Where *dist(x, y)* is the distance between *x* and *y*. When a peer *p* wants to engage in a reconfiguration process, it finds the peer so that it results the maximum distance gain.



(b) After (re)configuration

Figure 5. (Re) configuration process.

3.6. The function of (Re) configuration in P2P over mobile ad hoc networks

Generally the wireless devices have a restricted transmission range due to its limited power supply. So, data search should preferentially be made within a small distance. It can be happened in two ways. In general, the first solution provides continuous access to an information network like the Internet or a private intranet with high costs, using a fixed infrastructure. The second way does not need a fixed infrastructure: a set of mobile devices, on their own, forms the ad hoc network and nearby devices become important sources of data for each other. This is similar to the P2P paradigm that incorporates a network element which acts as client, server and router at the same time. Once the ad hoc and the P2P networks take place in different layers, they do not substitute each other.

Conversely, they seem to be naturally complementary**.** The easiness of forming the network is one of the main advantages of a P2P network, since it is not necessary to have an infrastructure nor it depends on a central server. Examples of possible uses of P2P over ad hoc networks include applications that alert us to the presence of friends at a crowded public space or identify people we want to meet taking into account our preferences and interests; systems that spread rumors, facilitate the exchange of personal information, or support us in more complex tasks.

Another thing is that P2P over ad hoc networks is a very dynamic combination that demands, among other things, special attention regarding (re)configuration [23] issues as discussed below. P2P networks are virtual networks which focus on sharing data. For a better performance in the search mechanism the protocols are designed so that data can be found more rapidly and more frequently. The P2P network topology, however, might be very different from the physical one, that is, an efficient P2P protocol may have a negative impact on the ad hoc network. Generally, P2P protocols have in common a mechanism that consists in having a list of neighbors to which the queries can be forwarded. In fixed networks, the status of a neighbor would usually vary between is available and is not available. In mobile scenarios, however, a close peer may become distant in a short period of time, maintaining it's is available status. In this case, the P2P distance, measured in hops, would be the same but the ad hoc distance would increase which in turn generating a greater traffic on the physical layer. Data search could even perform as before but the ad hoc network would be overloaded and nodes would demand much more of their power supply. It is important to monitor the references between nodes, (re)configuring the P2P network taking into account the ad hoc topology. The (re)configuration algorithms [23] are described below:

3.7. (Re) configuration algorithms

(Re) configuration algorithms [23] for decentralized P2P network being proposed. There are three different decentralized algorithms, which are called Basic, Regular and Random[42].In the description of all these algorithms, it will be said that the nodes are connected, trying to connect, maintaining a connection and other similar terms. It is important to notice, however, that we are dealing with wireless networks and, thus, there are no real connections, e.g., a TCP connection, between nodes. Here connections actually mean references, that is, they represent the knowledge of the addresses of some reachable nodes. Thus, if a node A keeps a reference to node B whereas B also refers A, then it is considered as regular connection. On the other hand, Irregular connections also exist and are used in the Basic algorithm.

3.7.1. Decentralized algorithms

The proposed system's model forwards the message over different hops from a peer to the next in order to establish connections and search for data. In spite of having the same proposition, the three decentralized algorithms have distinct behaviors, as we have shown below. The Basic algorithm has been discussed in the next section. Another unavoidable concept will be briefly described after that: the small-world effect, which is the key point for turning the Regular algorithm into the Random algorithm. The Random algorithm has been described at the end.

3.7.2. The basic algorithm

The Basic algorithm is the basis for comparison and serves as a simple (re)configuration algorithm. It partially ignores the dynamic nature of the network. The Basic algorithm which is shown below uses three constants named MAX_N_CONN, N_HOPS and TTL. MAX_N_CONN represents the maximum number of connections per node. N_HOPS is the number of hops a message can travel and TTL stands for the time interval between two attempts to establish connections. The algorithm works as described below. When a node is starting to participate in the P2P network, it broadcasts a message to discover other nodes within N_HOPS away in the neighborhood. Each and every node that listens to this message answers it. The node establishes a connection to the neighbor which sent the response, as soon as the response arrives, until the limit of MAX_N_CONN connections. In case the number of responses is less than MAX_N_CONN and whenever else it has less than MAX_N_CONN connections then the node keeps trying to create the rest of the connections. Here the time interval is the rescue for network to avoid the traffic overload during consecutive trials. The node waits for a time interval—TTL—in order to avoid traffic overload in the network. The validity of a connection or reference is frequently checked by sending ping once upon it is established. Whenever a node receives a ping it answers with a pong. The pong signal represents the existence of a connection whereas its absence means the neighbor is not reachable anymore and then the connection is over.

Basic algorithm:
A_Basic: Establishing connections:
While the node belongs to P2P network
If number of connections < MAX_N_CONN
Try to establish new connections to nodes within N_HOPS away up to the limit of MAX_N_CONN connections;
Wait TTL before next try;
End if
End while

Maintaining connections:
While this connection exists
Send a *ping* to the connected node;
Wait some time for the *pong*;
If the *pong* was received then
Wait some time before sending next *ping*;
Else
Close this connection;
End if
End while

3.7.3. The small-world model

In a regular graph its n vertices are connected to their nearest k neighbors but in a random graph, the connections are randomly established and k stands for the average number of edges per vertex. In this way two neighbors of a node have a greater chance of being connected to each other in regular graphs, that is, the average clustering coefficient is much greater in regular graphs. This coefficient is obtained as follows: let real_conn be the number of existent connections between all the neighbors of a node (these neighbors are connected to this given node); and let possible_conn be the number of all connections that could exist between these neighbors. The clustering coefficient is given by real_conn/possible_conn. Besides the clustering coefficient, the regular and random graphs also have very distinct characteristic path lengths. In large regular graphs with n much larger than k—for a k much larger than 1— the path length is approximately n/2k. In large random graphs this value decreases substantially and is given by log n/ log k. Interestingly, little changes in regular graphs connections are sufficient to achieve short global path lengths as in random graphs. The rewiring of some connections from neighbors to randomly chosen vertices creates bridges between clusters that are distant. These bridges diminish the path length without any considerable change in the clustering coefficient.

Graphs that have high clustering coefficients and, at the same time, short global path lengths are called small world graphs. Our Random (re)configuration algorithm, presented next, aims to construct a Peer-to-Peer network as a small-world graph. Before presenting the Regular and Random algorithms, we will list their variables and constants, most of which are present in both algorithms. There are three variables: hops, randhops and timer. The first one represents the number of hops a message looking for a regular connection can travel. It is initialized with the value N_HOPS_INITIAL, which is greater than 1, and has MAX_N_HOPS as an upper limit. The Second one has a similar meaning but it is only applied to random connections; it does not need to be initialized. The third variable stands for the time interval a node waits between two attempts to establish connections. It is initialized with TTL_INITIAL and can increase up to MAXTIMER. Finally, there are two remaining constants not explained yet: MAX_N_CONN, which is the maximum number of connections per node, and MAX_DIST, which is the maximum distance allowed between two connected neighbors (measured in number of hops).

3.7.4. The regular algorithm

Initially, there is the ad hoc network and some (or all) of its nodes that want to build the P2P network execute the algorithm. In this case, the node broadcasts a message saying that it is available to establish connections. Messages are expected to travel a specific number of hops (nhops). When receiving this message, a node willing to connect starts a three-way handshake with the sender, aiming to establish a regular connection. If, within that radius, less than MAX_N_CONN neighbors can be regularly connected, the node will make another broadcast with a larger number of hops (nhops +2). Before the new broadcast, however, it waits for a timer time interval. As in the Basic algorithm, this interval is an attempt to avoid traffic overload. This mechanism is repeated until the maximum of MAX_N_CONN connections or the maximum of MAX_N_HOPS hops is achieved, whichever occurs first. When nhops is set to 0 it means that the node tried all possible values for nhops without connecting to MAX_N_CONN neighbors. In this case, the time interval timer is doubled before the next cycle of trials, in which nhops will restart with the N_HOPS_INITIAL value. Once a connection is successfully built, the node starts its maintenance as presented below. The connection is frequently checked using pings. As we are dealing with regular connections, only the node that started the process of establishing the connection will send pings. The reception of pings is controlled by the other node with the use of a timer. Whenever it receives a ping, it answers with a pong and reinitializes the timer. In case a timeout occurs, it closes the connection. Upon receiving a pong, the other node knows its neighbor is still reachable, but this is not enough to maintain the connection. To remain connected, the distance between the nodes must be less than MAX_N_HOPS hops. In case the distance is greater than that, the connection is closed. The same occurs in the absence of a pong.

Regular and random algorithm:

A_Regular: Establishing connections
While the node belongs to P2P network
If number of connections < MAX_N_CONN then
If nhops ≠ 0 then
Try to establish new and regular connections to
Nodes within nhops away up to the limit of
MAX_N_CONN connections;
Wait timer before next trial;
Else
Timer = min (timer × 2, MAX_TTL);
End if
Nhops = (nhops+2) mod (MAX_N_HOPS + 2);
End if
End while
A_Random: Establishing connections
While the node belongs to P2P network
If number of connections < MAX_N_CONN then
If nhops ≠ 0 then
Try to establish new and regular connections
To nodes within nhops away up to the limit of
MAX_N_CONN – 1 regular connection;
Wait timer before next trial;
Else
Timer = min (timer × 2, MAX_TTL);
End if
If a random connection is needed then
Set randhops to a randomly chosen value between
Nhops and 2 × MAX_N_HOPS;
Try to establish one new and regular random
Connection to the farest node possible within
Randhops away;
End if
Nhops = (nhops + 2) mod (MAX_N_HOPS + 2);
End if
End while
A_Regular and A_Random: Maintaining connections:
While this connection exists
If it is the node that asked for the connection
Send a ping to the connected node;
Wait some time for the pong;
If the pong was received then
If this is a random connection then
If the node is nearer than 2× MAX_DIST then
Wait some time before sending next ping;
Else close this connection;
End if
Else
If the node is nearer than MAX_DIST then
Wait some time before sending next ping;
Else close this connection;
End if
End if
Else close this connection;
End if
Else wait some time for the ping;
If the ping was received then
Send a pong;
Else close this connection;
End if
End if
End while

This algorithm has four improvements compared to the Basic algorithm. First, the number of hops a message that is looking for connections may travel is increased gradually. Once this kind of message is sent by broadcast, controlling the number of hops means less traffic in the network. The traffic is also potentially diminished by the control of the distance between connected nodes since the pings and pongs exchanged will span a narrower area. This was the second improvement, which is complemented by the third one: the number of pings and pongs was cut half because only one vertex checks whether the connection is active by sending pings. These three actions added together have a positive impact on wireless networks, which have bandwidth and energy constraints. Last, but not least, there is the fourth improvement, related to the timer and which was inspired by the dynamic nature of our network, together with the traffic concern. The time interval between two broadcasts has not a fixed value. Instead, it doubles every time a cycle of attempts to establish connections is over, diminishing the overall traffic. Besides, if it is being difficult to connect to other nodes, while waiting for a longer interval the network can change to a more favorable configuration. Then, it may be easier to establish the desired connections. One detail not presented in the pseudo-code is that, whenever a connection is done, the timer is reset to its initial value. This is because this new connection may indicate a better network configuration.

3.7.5. The random algorithm

Adopting the Regular algorithm, each node preferentially connects to its nearest neighbors. In a dense P2P network, the connections will be established within a low number of hops. Probably, this will lead to a network whose characteristics resemble regular graphs, mainly in the sense of long global path lengths. Aiming to avoid this and to gain the small-world characteristics, our Regular algorithm was changed to the Random algorithm. The establishment of the first MAX_N_CONN−1 connections follows exactly the same steps mentioned in the Regular algorithm. For this reason, they will be called regular connections. The difference of the two algorithms lies in the last connection, as it can be seen in the algorithm .As discussed before, few rewiring can turn a regular graph into a small-world graph. To promote this rewiring, the node does not try to establish its last connection within nhops away. Instead, it chooses a random number (randhops) between nhops and2 × MAX_N_HOPS. Then, it broadcasts a message looking for connections to all nodes within randhops hops away. It waits some time for responses to arrive, analyzes them, and only continues the three-way handshake with the most distant neighbor. Once a connection is established this way, it is called a random connection and, whenever it goes down, it must be replaced by another random connection. The maintenance of the existing connections follows the scheme presented in algorithm. The final effect expected is that some of the overall connections will link distant peers, and, therefore, will act as bridges. This will turn the path length shorter while maintaining the clustering coefficient high, and achieving the small world Effect.

IV. SIMULATIONS

Simulation can be defined as the process of designing a model of a real system and conducting experiments with this model for the purpose of understanding the behaviour of the system and/or evaluating various strategies for its control. To make this process useful, the behaviour of the model is expected to realistically initiate the response of the system under study. The operation of wireless applications or protocols in the perspective of Mobile Ad-hoc Networks (MANETs) frequently requires moving through a simulation phase. For the results of the simulation to be significant, it is important that the model on which is based the simulator matches as strictly as possible the reality. There exist several popular simulators, such as OPNET Modeller, NS-2 or GloMoSim MATLAB. In this paper the simulation results being presented of a straightforward algorithm using NS-2 and MATLAB. We have used Microsoft Excel to performance evaluation to generate graph from the extracted data of NS-2.

4.1. Matrices for performance evaluation

In order to measure the performance of our proposed platform, we have used the following matrices:
Average End-to-end Throughput: Throughput is the average bit rate at the destination node. The end-to-end throughput is measured as the average of the end-to-end throughputs of all flows in the network. Throughput= (total no of bytes received/simulation time)*(8/1000) kbps.
For calculating Average end-to-end Throughput from our extracted data of NS-2 we have used the following formula:
Average end-to-end Throughput is the total of all bits (or packets) successfully delivered to individual destinations over total-time / total-time (or over bits-total / total time)
Average End-to-end Delay: The average delay is the delay a packet takes to travel from the source to the destination node. The end-to-end delay is the end-to-end delay experienced by flows in the network. Since flows traverse multiple hops in an ad-hoc network, this metric gains importance as the delay is directly proportional to the hop length of the path and inversely proportional to the end-to-end throughput.
For calculating average end-to-end delay from our extracted data of NS-2 we have used the following formula:
Average end-to-end Delay=Sum of delay experienced by each packet making up the flow) / number of packets

4.2. Simulation environments

In order to evaluate the proposed platform, we performed simulations in the NS-2 simulator version 2.29.3 and MATLAB. NS-2 is a highly modular discrete event simulator, developed for simulating the behaviour of network and transport layer protocols in a complex network topology. It is freely available and has been extensively enhanced by the Monarch project at CMU for using simulating mobile ad hoc networks. In our NS-2 simulations the P2P applications run on top of the UDP protocol, since TCP does not perform well in this type of environment. We choose DSDV for routing for its best performance under a P2P applications in the most common MANET scenarios[35]. Nodes are configured with typical PDA network parameters(11 Mbps IEEE 802.11b with 50m of range). The interface queue (IFQ) length is set to 50 packets and energy consumption is 230 mW for reception and 330mW for transmission. Radio propagation follows the two-ray-ground model. We assume a network of varying number of nodes scattered in a 500mX500m grid area. Nodes move accordingly to the random way-point mobility model, with a pause time of 2.0s and maximum speed of 10.0m/s. The application traffic pattern consists of CBR sources running on UDP that start at staggered times.CBR source destination pairs are generated randomly. In our simulations we fixed the packet size and the number of CBR flows. We made the variation in the node number from 5 to 20. These 4 variations were combined with 5 mobility scenarios as input files to the network simulators. Altogether we performed 20 simulations.

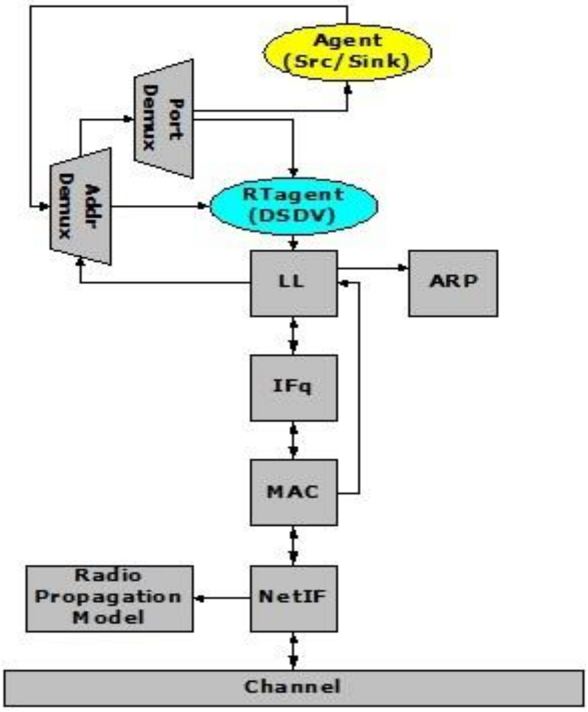


Figure 6. Wireless simulation in NS-2(mobile node diagram-DSDV)[ns manual].

4.2.1. Simulation parameters

TABLE 2. Summaries of all the constant parameters of the simulations

Table 2. Static parameters.

Parameters	Value
Terrain size	500mx500m
Node placement	Uniform
MAC protocol	802.11 without RTS/CTS
Bit rate	4.0 kbps
Wireless Propagation model	Two-ray-ground
Antenna type	Omni directional
Mobility Model	Random waypoint
Pause time	2.0s
Maximum node speed	10.0m/s
Simulation time	200s

4.2.2. Creating random traffic pattern for wireless scenarios

Random traffic connections of TCP and CBR can be setup between mobile nodes using a traffic-scenario generator script. This traffic generator script is available under ~ns/indep-utils/cmu-scen-gen and is called cbrgen.tcl. It can be used to create CBR and TCP traffics connections between wireless mobile nodes. In order to create a traffic-connection file, we need to define the type of traffic connection (CBR or TCP), the number of nodes and maximum number of connections to be setup between them, a random seed and incase of CBR connections, a rate whose inverse value is used to compute the interval time between the CBR pkts. So the command line looks like the following:

cbrgen.tcl [-type cbr|tcp] [-nn nodes] [-seed seed] [-mc connections] [-rate rate]

For our simulations we generated five scripts varying the number of nodes.

4.2.3. Creating node-movements for wireless scenarios

The node-movement generator is available under ~ns/indep-utils/cmu-scen-gen/setdest directory and consists of setdest {.cc, .h} and Makefile. For creating node movement scenarios the command looks like this,

/setdest [-n num_of_nodes] [-p pause time] [-s max speed] [-t sim time] \[-x max x] [-y max y] >[out dir/movement-file]

Five scenarios with varying number of nodes have been considered.

4.2.4. Extracting data from trace files

We extracted data from the trace files for calculating throughput and delay by the following OTcl command-
awk '/^s/ && /AGT/ {print}' filename.tr > output file

4.3. Performance evaluation of simulation results

This section presents the results collected during simulation with NS-2 and MATLAB simulators. The simulation was performed for five different scenarios by varying node number. We varied node number from five to twenty. We focused our analysis on two metrics: Average end-to-end delay and average end-to-end throughput.

4.3.1. Average end-to-end delay analysis

This includes delay perceived by a user requesting some content, including the time for transmitting the query to the network, locating the desired content in the network, and returning a response back to the user.

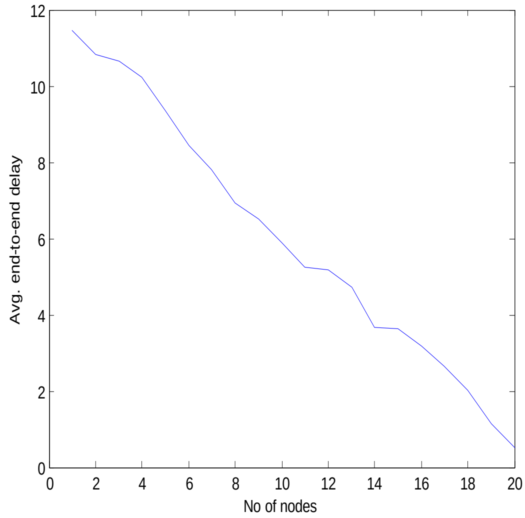


Figure 7. MATLAB result for average end-to-end delay.

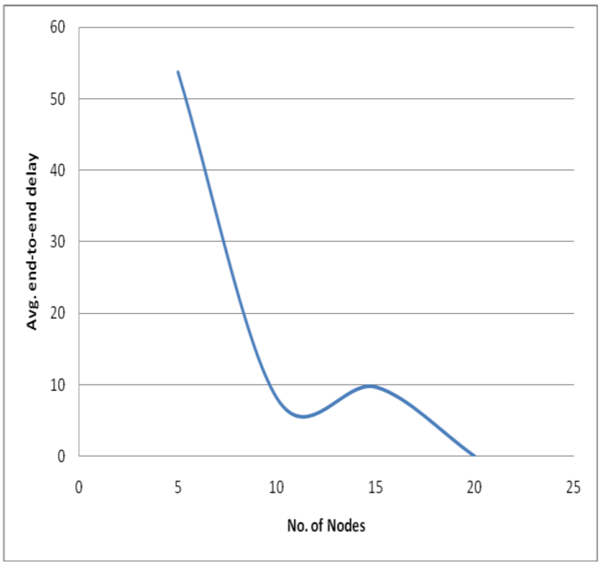


Figure 8. NS-2 result for average end-to-end delay.

From the result it can be seen that average end-to-end delay decreases as the number of node increases. These in terms reduce the no of link hops.

4.3.2. Average end-to-end throughput analysis

Throughput in this research is calculated for the total no. of packets received during the simulation time of 200sec.

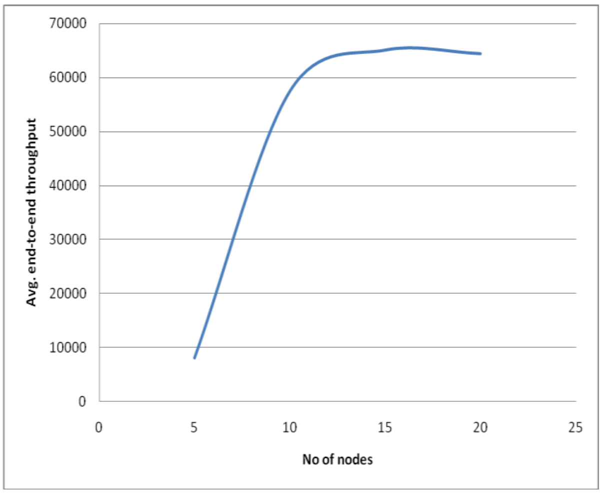


Figure 9. NS-2 result for average end-to-end throughput.

From the result it can be seen that average end-to-end throughput increases as the number of node increases. This in terms increases the throughput.

4.3.3. Discussion of simulation results

Our study is based on Gnutella. Due to use of flooding Gnutella is inefficient in large-scale networks or in situations where no. of nodes is high. Using gossiping reduces number of messages sent, however number of hop travel increases[25].We have used (re)configuration algorithm[23] for reducing number of hop count. It is expected that reformation reduces link level hop count between neighboring peers [10].From the simulation results it is clear that the as the no. of nodes increases the throughput increases and delay decreases. Reduced delay means reduction in the number of hops between the peers. Increased throughput means increase of bandwidth as bandwidth is proportional to throughput. Note that both of these properties are very much desirable for wireless mobile applications. As reduced number of hops reduces number of link level messages which slows down energy consumption and boosts battery life of mobile devices.

V. CONCLUSIONS

Mobile ad hoc networks (MANETs) are infrastructure less and nodes may act as both clients and servers, which increases fault tolerance and data availability. That's why Peer-to-Peer (P2P) networks are a well fit for content distribution on such networks. The contribution of this proposed platform is to reduce the cost of Peer-to-Peer mobile wireless networks. According to the existing theory it is possible to minimize the consumption of bandwidth and energy of mobile wireless devices and to increase the throughput of the network. This proposed platform has presented different viewpoints to improve the performance of Peer-to-Peer networks. We have used inexpensive gossip protocol to manage the network. We have considered distance between neighbours as a biasing factor. In ado wireless networks, each node is able to setup a point-to-point communication to other nodes within its radio signal range, without the need of a fixed infrastructure. To uphold an efficient sharing of information and improve network durability, we have implemented (re)configuration [23] algorithms that provide configuration, maintenance and reformation of the P2P network over an ad hoc network. To analyze the performance of the algorithms we have performed some simulations using the NS-2 and MATLAB. The proposed platform is robust and flexible with high fault tolerance.

5.1 Future work

As our future work, we will study different approaches to design a low cost platform by improving the performance of P2P networks over MANETs. We are most interested in analyzing the effects of response time, density of nodes, energy, mobility of nodes in ad hoc and P2P layers. Caching [24] is one promising approach, where nodes would pro-actively store the most frequently accessed items. This would reduce the response time and the number of messages in the on the network by resolving queries in less hops. Variable transmit power into ad hoc routing protocols allows nodes some control over their local densities.

REFERENCES

[1] Lv Q, Cao P, Cohen E, Li K, Shenker S (2002) 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.

[2] Clarke I, Miller S, Hong T, Sandberg O, Wiley B (2002)Protecting free expression online with freenet.In:*IEEE Internet Comput* 6(1):pp 40–49.

[3] Ratnasamy S, Francis P, Handley M, Karp R, Shenker S(2001) A scalable content Addressable network. In: *ACMSIGCOMM*.

[4] Rowstron A, Druschel P (2001) Pastry: scalable, distributed object location and routing for large-scale Peer-to-Peer systems.In*:IFIP/ACM international conference on distributed systems platforms* .Germany, pp 329–350.

[5] Jing Tie; Hai Jin; Shengli Li; Xuanhua Shi; Hanhua Chen; Xiaoming Ning(2006) Peer-Tree: A Hybrid Peer-to-Peer Overlay for Service Discovery.In*: 20th International Conference on Advanced Information Networking and Applications*, pp 999-1004.

[6] Ramaswamy L, Gedik B, Liu L (2005) A distributed approach to node clustering in decentralized Peer-to-Peer networks. In: *IEEE Trans Parallel Distrib Syst* 16(9): pp 814-829.

[7] Chakraborty D, Joshi A, Yesha Y, Finin T (2006) Toward distributed service discovery in pervasive computing environments.In*:IEEE Trans Mob Comput* 5 (2):pp 97–112.

[8] Ghandeharizadeh S, Krishnamachari B, Song S (2004) Placement of continuous media in wireless Peer-to-Peer networks.In*:IEEE Trans Multimed* 6(2):pp 335–342.

[9] Ding G, Bhargava B (2004) Peer-to-Peer file-sharing over mobile ad hoc networks.In: *IEEE conference on pervasive computing and communications workshops.* FL, USA, pp 104–108.

[10] M Akon,X Shen,S Naik, A Singh, Q Zhang(2008) An inexpensive Unstructured platform for wireless mobile Peer-to-Peer networks.In:*Peer-to-Peer networking and applications*, Springer New York.

[11] Stoica I, Morris R, Karger D, Kaashoek MF, and Balakrishnan H (2001) Chord: a scalable Peer-to-Peer lookup service for internet applications.In: *ACM SIGCOMM* 2001. CA, USA,pp 149–160.

[12] Castro M, Druschel P, Kermarrec AM, Rowstron A (2002)SCRIBE: a large- scale and decentralised application-level multicast infrastructure.In:*IEEE J Sel Areas in Commun*(JSAC) 20(8):100–110.

[13] Zhao BY, Kubiatowicz JD, Joseph AD (2001) Tapestry: An infrastructure for fault-tolerant wide-area location and routing.In:*Tech. Rep. UCB/CSD-01-1141*, U. C. Berkeley.

[14] Freedman MJ, Sit E, Cates J, Morris R (2002) Introducing Tarzan, a Peer-to-Peer anonymizing network layer. In: *1st intl.workshop on Peer-to-Peer systems*, pp 121–129.

[15] Passarella A, Delmastro F, Conti M (2006) XScribe: a stateless, cross-layer approach to P2P multicast in multi-hop ad hoc networks. In: *International conference on mobile computing and networking.* Los Angeles, California, pp 6–11.

[16] Druschel P, Rowstron A (2001) PAST: a large-scale, persistent Peer-to-Peer storage utility. In: *HotOS VIII*. Germany, pp 75–80.

[17] Stavrou A, Rubenstein D, Sahu S (2004) A lightweight, robustP2P system to handle flash crowds.In:*IEEE J Sel Areas in Commun* 22(1):pp6–17.

[18] Oliveira L. B., Siqueira I. G., and Loureiro A. A. F.(2003) Evaluation of Ad hoc Routing Protocols under a Peer-to-Peer Application. In: *IEEE Wireless Communications and Networking Conference (WCNC'03)*. New Orleans/LA.pp. 1143-1148.

[19] Zorzi M. and Rao R.R. (2003) Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance, In: IEEE *Trans. Mobile Computing*, 2 (4).

[20] Cleary DC, Parke DC (2005) "Finding Yourself" building location services in a Peer-to-Peer wireless world. In:*EUROCON 2005*, pp 60–63.

[21] Perkins C.E, Ad Hoc Networking, Addison-Wesley, Boston MA, USA, 2001.

[22] Hsieh H.Y, Sivakumar R (2001) Performance Comparison of Cellular and Multihop Wireless Networks: A Quantitative Study *Proc. ACM SIGMETRICS*, pp. 113-122.

[23] Franciscani F. P, Vasconcelos M. A, Couto R. P, and Loureiro A. A. F (2003) Peer- to-Peer over Ad Hoc Networks: (Re) Configuration Algorithms. In: *Proceedings of the International Parallel and Distributed Processing Symposium*, pp.32-2.

[24] Hedetniemi S. M, Hedetniemi T. S, and Liestman A. L (1988)A survey of gossiping and broadcasting in communication networks," *Networks*, pp. 319–349.

[25] W. R. Heinzelman, J. Kulik, and H. Balakrishnan(1999). Adaptive protocols for information dissemination in wireless sensor networks.In: *Proceedings of the fifth annual ACM/IEEE international conference on Mobile computing and networking*,pp. 174–185.