Evaluation of Ad-Hoc Routing Protocols under a Peer-to-Peer Application

Leonardo Barbosa e Oliveira, Isabela Guimarães Siqueira and Antonio Alfredo Ferreira Loureiro
Federal University of Minas Gerais
Computer Science Department
Belo Horizonte, Minas Gerais, Brazil
Email: {leob,isabela,loureiro}@dcc.ufmg.br

***Abstract*— Mobile Ad-Hoc Networks (MANETs) and Peer-to-Peer (P2P) applications are emerging technologies based on the same paradigm: the Peer-to-Peer paradigm. Motivated, respectively, by the necessity of executing applications in environments with no previous infra-structure and the demand for applications that share, in a satisfying manner, files through the Internet, MANETs and P2P applications have brought onto themselves some interest from the community. As a characteristic of the distributed model which they follow, such technologies face a difficult task of routing requests in a decentralized environment. In this paper we conducted a detailed study of a Gnutella-like application running over a Mobile Ad-hoc Network where three different protocols were considered. The results show that each of the protocols analyzed performed well in some scenarios for some metrics yet had drawbacks in others.**

I. Introduction

The recently disseminated Peer-to-Peer (P2P) paradigm is the basis for both Mobile Ad-hoc Networks (MANETs) and popular Internet P2P applications. The P2P paradigm has as its most significant particularity the fact that central units, which are responsible for managing and meeting the needs of the network, are non-existent. In this model, nodes have equivalent functionalities and provision capabilities and, as a consequence, are called "peer" entities. Every peer is able to send and reply to request messages originated from another one. This shows the dual interface of these peers, since they might play the role of servers and clients simultaneously. That is the reason why they are also named "servents" (servers/clients).

Similarly to the architecture on which they are based, MANETs and P2P applications have only recently drawn attention to themselves. The growth of computing resources for mobile devices has been the key contributing factor to the dissemination of mobile ad-hoc networks. Moreover, the launch of new applications – such as rescue team management in disaster situations or the exchange of information in combat areas – generated an increase in demand for networks without previous infra-structure. On the other hand, the spread out of P2P applications can be attributed to their success as file sharing platforms – specially those that distribute MP3-compressed music tracks [1].

Based on the same paradigm, P2P application networks, which are composed by a set of servers implementing a P2P application, and MANETs have common characteristics and functionalities. In essence, both are self-organizing networks, have dynamic topology, and are responsible for routing queries in a distributed environment.

Many researchers have already carried out work in order to evaluate the performance of routing protocols in MANETs [2]–[4] and also in P2P application networks [5]. Others have highlighted a few of their theoretical similarities [6]. However, it seems that there are no evaluations of both systems acting together yet.

A novel diagram of a P2P application running over a MANET is shown in Fig. 1.

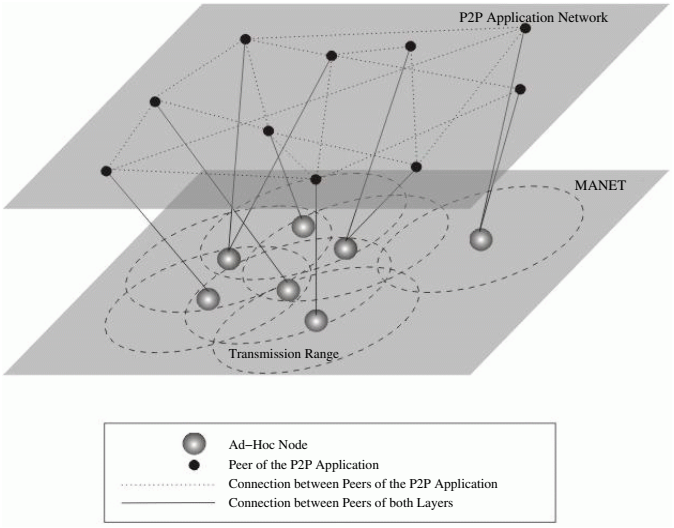


Fig. 1. A diagram of a P2P Application over a MANET

Because nodes in mobile ad-hoc networks usually have low computing capacity and therefore are unable to play the role of servers all the time, – or even supply many clients simultaneously – P2P application appears to be a power tool to disseminate information on this type of scenario. In other words, since a P2P application network does not possess a unique service provider at a certain time, but many servents playing this role, the assignment of distributed network tasks among nodes prevents them to become overloaded.

The main goal of this work is to simulate a scenario containing a P2P application running over a MANET in order to evaluate the performance of ad-hoc routing algorithms. The Destination-Sequenced Distance-Vector Routing (DSDV) [7], the Dynamic Source Routing Protocol (DSR) [8], and the Ad-

0-7803-7700-1/03/$17.00 (C) 2003 IEEE

hoc On Demand Distance Vector (AODV) [9] are the protocols evaluated. The reasons for the choice are twofold. First, the nature of those algorithms is distinct. While DSDV is proactive, DSR and AODV are reactive – though the last employs methods of the two formers. Second, the three have already been exhaustively tested and validated [2]. The results point out that each of the protocols analyzed performed well in some scenarios yet had drawbacks in others. This confirms the importance of considering characteristics of both application and network in order to have the best integrated solution.

The rest of this paper is organized as follows. Next section presents a comparison between MANETs and P2P application networks. In Section III the P2P application is discussed. The analysis methods as well as the simulation environment are described in Sections V and IV, respectively. Section VI reveals the simulation results. Section VII provides a discussion about the P2P application in comparison with the Client/Server results, and finally, Section VIII presents our conclusions.

II. Comparison between MANETs and P2P Application Networks

P2P applications and mobile ad-hoc networks have several aspects in common since they follow the same model. Both MANETs and P2P application networks lack managing and centralizing units, since the network is established as soon as the participants opt to interact with one another. The decision to connect to the network can be taken at distinct moments, so variance is constantly introduced in the environment.

Another similarity is their dynamic topology, which is a result of the constant changes in connections used by peers. In mobile ad-hoc networks these alterations are mainly caused by a node mobility. That is, as a node moves, it might leave the transmission range area of its current neighbors and have its links broken as a consequence. Thus, in order to reestablish contact with peer entities, the peers must set new connections. On the other hand, what causes the dynamic topology of P2P applications networks is the low availability of their peers. In this scenario applications are executed mostly over fixed networks and the main reason for link breakage is not the mobility of nodes, but the short session duration. According to [10] the average availability of peers is 0.083 and the average session duration is 2 hours per day. Because P2P applications are usually built over a network which is based on the Client/Server model, their networks present some characteristics that differ from the P2P paradigm. MANETs, in contrast, have their own communication mechanism and, therefore, are more faithful to the distributed model.

As previously mentioned, in the P2P architecture peers can communicate with one another without intervention of any centralized access point. Paradoxically, P2P applications are, in fact, clients of services provided by external servers – such as DHCP (Dynamic Host Configuration Protocol), DNS (Domain Name Service), and Web servers. In MANETs, requests are really handled by any network participant. Another evidence that MANETs are more in conformity with the P2P paradigm than P2P application networks is the fact that in the former, their peers are only a single-hop away from their neighbors, whereas in the last, the neighbors are just logic ones and might be geographically many hops apart.

Typical differences between both technologies [6] are described in Table I.

TABLE I
Differences between P2P application networks and MANETs

	P2P Network	MANET
Motivation for creating the network	Create a logical infra-structure 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 behavior	fixed	mobile
Broadcast	virtual, multiple unicasts	physical, to all nodes in transmission range area

III. Description of the P2P Implemented Protocol

In order to achieve the previously described objectives, it was required to implement a P2P application in the simulator. The adoption of special strategies was entailed, which would be dispensable in a Client/Server architecture. This is due to the P2P decentralization and dynamic nature and also to the role played by the servents in the P2P application network, which alternates from server to client and from client to server.

The implemented protocol is mainly based on Gnutella [11] protocol, which is used for peer-to-peer communication in Gnutella decentralized file-sharing system in the Internet. The main reason for choosing Gnutella protocol is the simplicity of its communication model. Since it was developed neither for best performance nor for best scalability, it is very suitable for evaluating network performance. Furthermore, Gnutella protocol is regarded as being able to adapt very well to dynamically changing peer populations, what is a very important characteristic.

Although Gnutella was taken as a reference, the protocol was altered for the simulator environment and also for ad-hoc networks. The strategies adopted are described in the following.

Searching. The fact that a P2P application network does not possess a server that centralizes information complicates the the task of locating data. In a Client/Server architecture, this is not a problem since the client knows the server address in advance. The strategy widely adopted, which has also been used in this work, is sending a "query-send" message throughout the network in order to gather information. This message contains the required file identification – its name – and the identification of the peer that is consulting, the "query-source".

The transmission of queries in the P2P application network is carried out through controlled flooding. The servent that receives a query message will forward it in case the file wanted is not stored in its node. The process goes on until the information source is found or the message is dropped due to a

TTL (Time-to-Live) expiration. Whenever a source is located, i.e., when a "query-hit" event occurs, the peer that owns the file wanted ("file-source") sends a reply to the "query-source" peer validating its availability for file transfer.

Joining the network. A peer desiring to join the P2P application network starts by sending a "broadcast-send" message throughout the network in order to elect the "neighbors", which may be used for message flooding. The initial replies will settle virtual connections between the new servent and each answerer. It is important to notice that virtual neighbors are not equivalent to physical neighbors although this is likely to occur at the beginning, since answers of near hosts tend to arrive more quickly.

Transferring files. After receiving the first reply, the "query-source" servent establishes an end-to-end communication with the "file-source". The file is fragmented in small pieces and each piece is sent inside a "pull-data" message from the "file-source" to the "query-source". The service for transferring data is datagram, typical of wireless environments.

Controlled Flooding. The query messages are uniquely identified by the pair (*query-id, query-source*). Each peer of the network maintains a cache which stores the last received messages, avoiding redundant processing, i.e., if a node receives the same message during a short period of time, the message will not be computed again.

The P2P message header has a TTL field to prevent a message being forwarded infinitely in the P2P application network. The idea is similar to TTL field of the Internet Protocol (IP). The next hop of a node in P2P, though, might be another node which is not directly connected.

Neighborhood Control. P2P application networks have a dynamic behavior, as mentioned before. Peers can leave the network at anytime, what implies the employment of a special control scheme for maintaining an up-to-date list of neighbors. To solve the problem in the implemented protocol, all peers have to send periodical "ping" messages to their neighbors to check if they are still "alive". When no answer is detected, i.e., when a "pong" message is not received, the related peer is removed from the neighbors list and a "broadcast-send" message is sent to find another neighbor.

Message Size. Messages that circulate among peers may have a variable size and the maximum value is 210 bytes, based on the Gnutella Protocol.

IV. Simulation

The simulator used in this work to evaluate the ad-hoc routing protocols was the Network Simulator (*ns-2*) [12]. The simulation was conducted using the following default settings. It was constructed a 200m $\times$ 200m [13] topology composed of 40 mobile nodes, 12 of which implementing a single instance of the P2P application.

For the mobility model, a pause-time of 50s and a maximum speed of 0.5 m/s were used. The mobility scheme employed was the Random Way Point.

The transmission range of all nodes was set for 50m using the Shadowing Propagation Model with a rate of 95% of correct reception within the range area. The IEEE 802.11 was the protocol used in the MAC layer, with 2 Mbits/s of bandwidth. The radio interface chosen was the 914 MHz Lucent WaveLAN. The total simulation time for all scenarios was set to be 300s.

In respect to P2P application parameters, a peer could have at most 3 neighbors and the initial number of files per peer was set to be 10. The choice of the initial file names as well as their sizes follow the normal distribution model. The average file size is adjusted to 10 KB. This reduced value was estimated taking into account the low bandwidth of mobile scenarios, as well as memory and energy constraints of mobile devices.

The "query-send" message in particular had its TTL set to 3 – large enough for queries to reach most of the peers in the simulated scenario – and the choice of the file to be searched follow the normal distribution model. The "ping" messages were sent with a default rate of 6 per minute and the "pong" messages were waited for no longer than 10s. The "broadcast-send" interval time was 2s.

In the simulation, nodes start with 100 Joules of energy each. The power loss for transmission was set to 0.280 Watts and 0.204 Watts for reception.

During the simulation, both the peer entrance time and the exit time were uniformly distributed in order to simulate the dynamic topology of the P2P application network.

Each scenario was simulated 33 times in a machine with the following hardware configuration: Pentium III 870 MHz processor and 256 MB RAM. The operating system and ns versions were Linux Mandrake 8.0 and ns-2.1b8a respectively.

V. Analysis Methods

The performance evaluation of ad-hoc routing protocols supporting a P2P application [14] examined four properties: workload, mobility, network density and peer quantity. The metrics were chosen considering its significance for each evaluation parameter.

Workload. It shows the workload introduced into the network by the P2P application. Its increase might place undesirable changes in network performance, such as latency,[1] packet dropping and control overhead.[2] As a consequence, the network may not provide a good service for the application. It was chosen to vary the number of queries generated by P2P peers as well as the average size of the files transferred through the network in order to investigate the protocol scalability. The results are presented in the next section. The following metrics were evaluated: number of file transfers initiated, throughput, percentage of queries not responded, delivery ratio, energy loss, and routing overhead associated to the ad-hoc network.

Mobility. Simulation experiments considering mobility were conducted, and the protocol capability in adapting to distinct

[1]The term latency suggests the amount of time spent for a specific event to happen, such as a query-hit or the reception of a response.

[2]The overhead was measured only in terms of packets, since the cost to access the medium to transmit a packet is significantly more expensive than the cost of adding a few extra bytes to an existing packet.

speed and pause time applied to ad-hoc nodes was analyzed regarding path length, connectivity among application peers, and latency.

Network Density. It affects greatly the performance of the ad-hoc network, and therefore is an important point of analysis. Simulation experiments considering different values of transmission range and number of nodes that populate the network were performed. In the last case, the number of peers was maintained at 30% of the total number of nodes. Percentage of queries not responded, path length, latency, routing overhead, connectivity among application peers, and delivery ratio were the metrics employed to evaluate the three protocols.

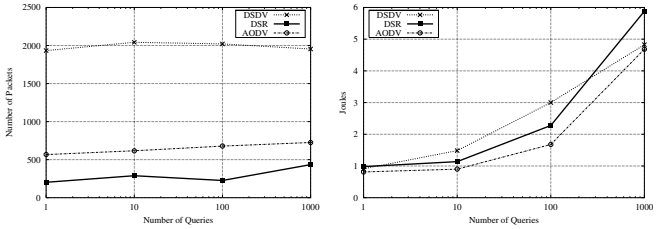
Peer Quantity. In order to investigate the amount of peers influence over the protocols, the number of nodes was left unmodified and the number of peers was increased. This type of analysis is important because it demonstrates the scalability of the ad-hoc routing protocols taking into account the number of application instances that run over the network. This is essential for selecting the best algorithm in case of deploying a P2P application. The metrics chosen were routing overhead, latency, path length, throughput, and energy loss.

VI. Simulation Results

This Section presents the results according to the four properties described above.

Workload. The three routing protocols introduced distinguishing amounts of overhead when the number of queries by a node was varied. As shown in Fig. 2(a), the DSDV exhibits the most overhead, followed by AODV and then DSR. The former, for one query, introduced ten times more control packets than DSR. Comparing DSDV to AODV and DSR on-demand protocols, the considerable increase in overhead obtained was due to route update messages that are constantly triggered by DSDV. Although DSDV produced more overhead, it demonstrated to have a steady behavior considering workload increase. The others, in contrast, did not suggest to be as scalable – DSR overhead, specifically, duplicated from one extreme of the x axis to the other.

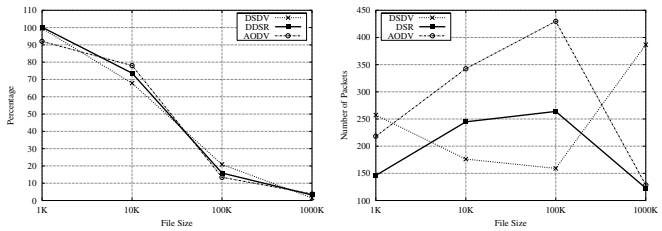
DSDV was the protocol which consumed the greatest amount of energy for lower and medium loads, as depicted in Fig. 2(b), as a result of its high overhead. For 1000 queries per peer, DSR nodes spend 20% more energy than AODV and DSDV ones. This was mainly due to DSR strategy of source routing. An increase in the number of queries causes the need of more packets being forwarded.



(a) Routing Overhead (b) Energy Loss

Fig. 2. Number of Queries Variation

Fig. 3(a) depicts the fraction of messages delivered to the application, as the shared file sizes were incremented. For all protocols, the curves assumed almost identical shapes. It can be noticed that while for 1 KB files the delivery ratio is higher than 90%, for 1000 KB practically all packets were dropped. This is due mainly to the low bandwidth available in the ad-hoc network.



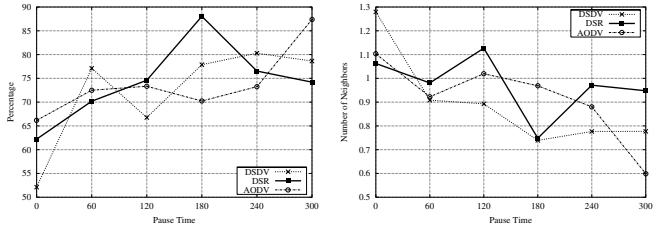
(a) Delivery Ratio for Application (b) "Pull-Data" Received

Fig. 3. File Size Variation

The results for file transfers initiated indicate that for all protocols the best performance is achieved when the average file size is 10 KB. This metric is represented by the number of "pull-data" messages received in Fig. 3(b), which can also be considered a result of throughput.

On the whole, DSDV performed better for extremely high loads. It obtained the lowest number of queries not responded, the highest throughput and more files successfully transferred. From Fig. 3(b) it is clear that DSR and AODV did not support the application requirements, in contrast to DSDV. This is due to the huge congestion generated, what caused difficulties for them to find routes on demand.

Mobility. Figs. 4(a) and 4(b) show the behavior of the connections among peers. Is is noticed that none of the protocols had a remarkable performance compared to the others. For lower mobility, the average number of neighbors and the amount of "ping" messages sent were reduced, while the number of "broadcast-send" messages and the number of queries not responded grew for all protocols. That is, DSDV, DSR and AODV produced more information unavailability and worse P2P connectivity.

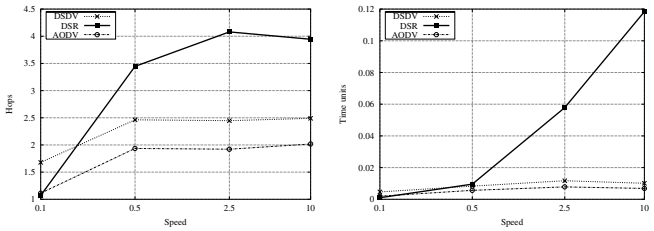


(a) Queries not Responded (b) Neighbors

Fig. 4. Pause Time Variation

As the mobility was incremented, surprisingly, the connectivity degree rose. This apparently anomalous behavior was mainly caused by the partitioning of the network. When the mobility is low, the network might isolate peers during the whole simulation, whereas in a higher mobility scenario these partitions are eliminated because of the peers movement. Paradoxically, Fig. 5(a) demonstrates that the increase in

speed did not have significant influence after 2.5 m/s. It is important to observe, though, that the DSDV and AODV curves stabilized earlier than the DSR curve.



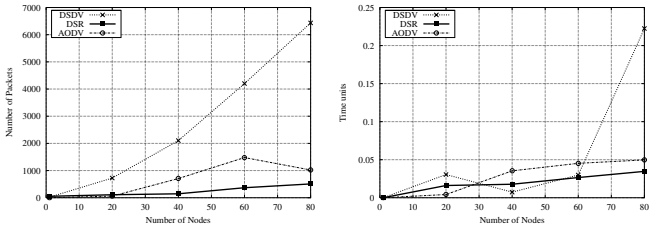
(a) Hops for Finding Information (b) Latency for Query-Hit

Fig. 5. Speed Variation

Fig. 5(b) shows the time elapsed for a query-hit to happen after the "query-send" message was sent. Both DSDV and AODV protocols had similar behaviors and showed to be insensitive to the node speed, whereas DSR was very sensitive to the node speed.

DSR was the protocol which presented the highest number of hops[3] and latency, when mobility was increased. Due to its source routing nature, in case nodes move at high speeds, a route generated might become outdated, even at the time when the packet is traversing the network from the source to the destination. As a result, more time and hops are consumed with routing.

Network Density. Concerning routing overhead, as shown in Fig. 6(a), DSDV was badly affected by the increase in the number of the network nodes, as it requires periodic routing updates and broadcasting of triggered beacon messages. In contrast, this scenario modification did not influence the other two protocols – which indicated to scale gracefully. Curiously, this performance declination did not appear when the transmission range was extended.



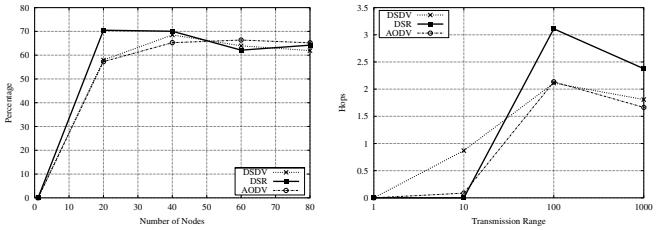
(a) Routing Overhead (b) Latency for a "Query-Reply"

Fig. 6. Nodes Variation

The three protocols behaved equivalently for both number of nodes and transmission range variation with respect to connectivity among application peers. The protocols had their latency intensified for a denser network, as presented in Fig. 6(b). Particularly, DSDV appears to be less scalable regarding this metric due to its routing overhead, as previously highlighted.

Regarding both number of queries not responded and network delivery ratio, DSDV, DSR, and AODV performances were similar. The former protocol, despite producing more routing overhead, managed to maintain the same delivery ratio for a denser network. Fig. 7(a) shows the results obtained for the delivery ratio.

In respect to path length, it was observed that this metric is very dependable on network density, as shown in Fig. 7(b). The highest average of hops and forwarded packets was offered by DSR and DSDV protocols, for denser and less dense networks, respectively. The former result can be easily explained, as DSR does not take into account path optimality when routes are generated. The last, though, can be considered a positive result, since the others obtained nearly zero average hops. In other words, this result means that DSDV is the only protocol which really delivers packets and provides support to the P2P application layer in less dense scenarios. Regarding the curve shapes of the three protocols, a change in the behavior could be detected. At this point, the number of hops falls suddenly, as a consequence of the proximity of the desired information. That is, when the number of existent nodes in the network is higher, it is more likely that the required information is stored on a near or easily reachable node. Furthermore, when the transmission range is expanded, the packets predictably tend to arrive in the destination with less hops.



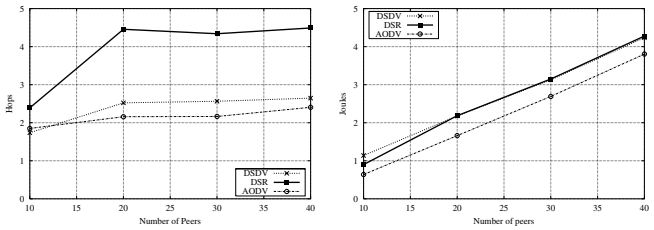
(a) Delivery Ratio for Application (b) Hops for Finding Information

Fig. 7. Nodes and Transmission Range Variation

Peers. Fig. 8(a) indicates that the DSR protocol needs more hops to find information than the others (nearly 2 times more hops than AODV, in the worst case), in agreement with the previously described results. Nevertheless, the shape of the curves is similar for the three protocols. When the network is populated with less instances of P2P applications, the desired information tends to be found in a fewer number of P2P hops. Also in this case, the amount of P2P neighbors of a peer is lower, since the number of reachable peers is lower as well. As a result, the network is likely to become partitioned, and in the rare cases in which the information is found, it will be located in one or two hops apart. The growth in the number of peers, by contrast, may expand the route lengths of the P2P application layer, allowing information to be found in a greater amount of P2P hops – and obviously the same for network hops. After a certain point in the increase of peers, the number of neighbors reached its maximum value and no more influence was detected.

All protocols were affected equivalently by the throughput. AODV was responsible for the best performance concerning routing overhead, while DSDV, as usual, generated more routing control packets. AODV also achieved better results respecting time. DSR, in contrast, presented the highest latency

[3]The term hops suggests the average amount of hops for a query to reach an information source.



(a) Hops for Finding Information (b) Energy Loss

Fig. 8. Peers Variation

– not only because it does not provide an optimal path, but also due to the fact that packets to be transmitted are held in its buffer until the path to destination is found.

Finally, Fig. 8(b) presents the energy loss. It is possible to observe that the shape of the curves for all the protocols evaluated was similar, considering the increasing of peers. AODV, however, provided less loss (35% approximately), since it possess the lower overhead.

VII. Comparison between P2P and Client/Server Applications

In this section a performance comparison between applications based on P2P and Client/Server paradigms is presented. The results concerning the Client/Server application were obtained mainly in [2].

First, regarding mobility, P2P and Client/Server applications exhibit considerable differences. Unlike the Client/Server model, in which the shortest path was obtained by DSR and DSDV in the simulated application, AODV was the protocol that presented the best performance, delivering the queries with the lowest hops average. This result shows that the merge between hop-by-hop routing of DSDV and route-discovery of DSR is less affected by the mobility property when a P2P application is considered.

Second, in both paradigms DSDV demonstrated to have the highest and steadiest overhead. However, the discrepancy between it and the other protocols is higher with the P2P application execution. It happens due to the growth in the topology dynamism caused by this type of application. Many link breakages occur and, as a result, a great amount of update messages need to be triggered. Despite the fact that AODV still has more overhead than DSR, the separation between the performance of both has decreased 60% at most, comparing with the results of a Client/Server application.

In respect to the delivery ratio, the network performance supporting a P2P application presented a worse result. With the increase in the amount of nodes, the ratio achieved at most 80%, whereas in Client/Server applications, it was achieved nearly 100%.

The most interesting result, possibly, was obtained with the mobility variation. Contrary to the scenarios which run applications based on a Client/Server model, the results achieved in this work reveal that P2P applications are not suitable for low mobility scenarios. First of all, it is important to emphasize that this kind of distributed applications are built to function in high dynamic scenarios, so it is reasonable that they present a low performance in more stable scenarios. Furthermore, when a peer moves, despite being likely to lose physical connections in the ad-hoc network level, it might not only maintain its P2P application links, but also establish others. And since the peers are highly dependable on their set of neighboring peers to communicate with the rest of the P2P application network, an increase in their amount of neighbors becomes an advantage.

VIII. Conclusion and Future Work

In the last few years, Mobile Ad-hoc Networks and Peer-to-Peer applications have started to be deployed, leading to a greater interest in the network community due to their distinct characteristics from traditional networks. Both MANETs and P2P applications have several points in common since they are based on the same model. Therefore, it is natural to study both MANETs and P2P applications together. In this paper we conducted a detailed study of a Gnutella-like application running over a Mobile Ad-hoc Network where three different protocols were considered. It is interesting to notice that each of the protocols analyzed performed well in some scenarios for some metrics yet had drawbacks in others. This conclusion shows the importance of identifying more precisely characteristics of the P2P application itself (workload, peer quantity) and characteristics of the network and mobile devices (mobility, network density) before committing to a particular protocol.

Future work will focus on improving performance of P2P applications over MANETs. Strategies which may be adopted are: modify existent ad-hoc protocols or even propose another one, use multicast protocols to disseminate information, and develop a middleware so that one layer – network or application – can take advantage of the other.

References

[1] "Openp2p," http://www.openp2p.com.
[2] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Mobile Computing and Networking*, 1998, pp. 85–97.
[3] S. R. Das, C. E. Perkins, E. M. Royer, and M. K. Marina., "Performance comparison of two on-demand routing protocols for ad hoc networks," in *IEEE Personal Commun. Mag., Special Issue on Ad hoc Networking*, Feb. 2001, pp. 16–28.
[4] H. Jiang and J. Garcia-Luna-Aceves, "Performance comparison of three routing protocols for ad hoc networks," in *IEEE ICCCN2001*, Feb. 2001, pp. 16–28.
[5] I. C. Society, "IEEE internet comput." Jan. 2002.
[6] R. Schollmeier and I. Gruber, "Routing in peer-to-peer and mobile ad hoc networks: A comparison," in *Workshop on Peer-to-Peer Computing, held in conjunction with IFIP Networking 2002*, May 2002.
[7] C. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, 1994, pp. 234–244.
[8] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," *IEEE Computer Communications*, vol. 3, pp. 1405–1413, 1997.
[9] C. Perkins, "Ad hoc on demand distance vector (AODV) routing," citeseer.nj.nec.com/article/perkins99ad.html, Nov. 1997.
[10] S. Saroiu, P. K. Gummadi, and S. Gribble, "A measurement study of peer-to-peer file sharing systems," May 2001.
[11] "The Gnutella protocol specification v0.4," http://www.gnutella.co.uk/.
[12] "Network Simulator," http://www.isi.edu/nsnam/ns.
[13] A. Jacobson, "Metrics in ad hoc networks," Master's thesis, Tekniska Universitet, 2000.
[14] N. W. Group, "Rfc2501 - mobile ad hoc networking (manet): Routing protocol performance issues and evaluation considerations," http://www.ietf.org/rfc/rfc2501.txt, Jan. 1999.