Available online at www.sciencedirect.com
SCIENCE DIRECT®
J. Parallel Distrib. Comput. 65 (2005) 1337–1347
Journal of Parallel and Distributed Computing
www.elsevier.com/locate/jpdc

On the performance of ad hoc routing protocols under a peer-to-peer application

Leonardo B. Oliveira, Isabela G. Siqueira*, Antonio A.F. Loureiro

Computer Science Department, Federal University of Minas Gerais, ICEx, Av. Antônio Carlos, 6627 Pampulha, Belo Horizonte, Minas Gerais, CEP 31270-010, Brazil

Received 1 January 2004; accepted 11 May 2005
Available online 15 July 2005

Abstract

Mobile ad hoc networks (MANETs) and peer-to-peer (P2P) applications are emerging technologies based on the same paradigm: the P2P paradigm. Motivated, respectively, by the necessity of executing applications in environments with no previous infra-structure and the demand for applications that share files or distribute processing through the Internet, MANETs and P2P applications have received some interest from research 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 MANET where three different protocols were considered. The results show that each protocol that were analyzed performed well in under some conditions and for some metrics, while had drawbacks in others.

Keywords: P2P over MANETs; MANETs; Peer-to-Peer applications; Gnutella; Ad hoc routing protocol; Performance evaluation; Simulation

1. Introduction

The recently introduced peer-to-peer (P2P) paradigm [15] is the basis for both Mobile Ad hoc Networks (MANETs) [7] and popular Internet P2P applications (e.g. SETI@home, Napster, Gnutella, Freenet). One of the most significant characteristics of the P2P paradigm is 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 each other. 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).

Similar to the architecture on which they are based, MANETs and P2P applications have recently attracted both research community and media attention. The growth of computing resources for mobile devices has been the key contributing factor for the focus on MANETS. Moreover, the launch of new applications—such as rescue team management in disaster situations or the exchange of information in battle fields [26,1]—generates an increase in demand for networks without previous infra-structure. The spread out of P2P applications, on the other hand, can be attributed to their success as content sharing and distributed processing platforms [23,1]—where parallel applications run on available peers.

Based on the same paradigm, both P2P application networks—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. Figs. 1(a) and (b) exhibit a MANET and a P2P Application diagram, respectively.

* Corresponding author. Fax: +55 31 3499 5858.
E-mail addresses: leob@dcc.ufmg.br (L.B. Oliveira), isabela@dcc.ufmg.br (I.G. Siqueira), loureiro@dcc.ufmg.br (A.A.F. Loureiro).

0743-7315/$ - see front matter © 2005 Elsevier Inc. All rights reserved.
doi:10.1016/j.jpdc.2005.05.023

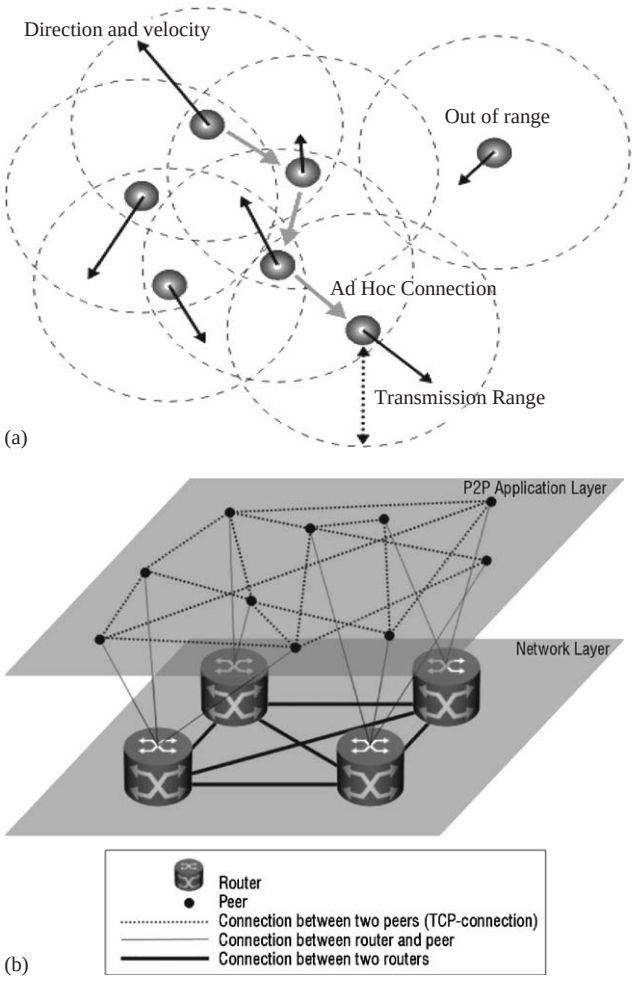


Fig. 1. MANET and P2P application diagrams: (a) A MANET diagram; (b) a P2P application diagram [21].

Because nodes in MANETS usually have low computing capacity and, therefore, are unable to play the role of servers all the time,—or even supply many clients simultaneously—a P2P application appears to be a powerful tool to spread 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 that play this role, the assignment of distributed network tasks among nodes prevents them to become overloaded. In addition, we envision that some applications enabled by MANETs (e.g., rescue team communication in disaster situations and exchange of information in battle fields [1,26]) will have each instance working in cooperation with the others (i.e., sending and replying to queries like peers). For instance, a rescue team participant might require information about nearest neighbor location. That is true a central server could be responsible for store information, but this approach not only would be more expensive (this would require more hops and constant location updates), but also less resilient—a single point o failure is not desirable in rescue team situations and servers would be target of attacks in battle field contexts. A novel diagram of a P2P application running over a MANET is shown in Fig. 2.

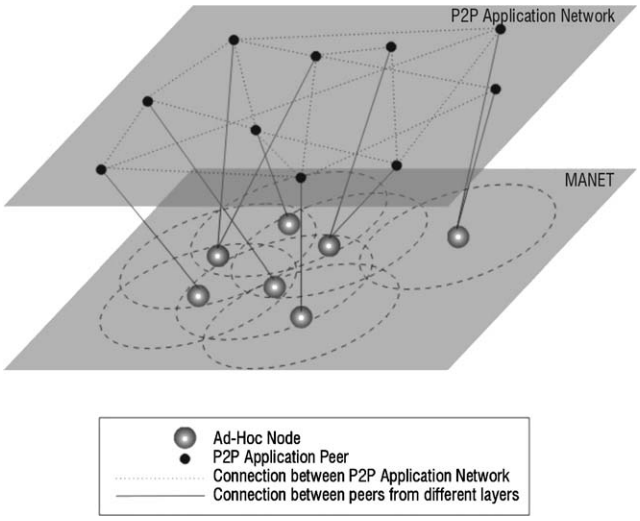


Fig. 2. A diagram of a P2P application over a MANET.

The main purpose of this work is to learn about the performance of ad hoc routing algorithms in a scenario in which a P2P application runs over a MANET. In order to accomplish the desired goals, we have conducted simulation experiments of some well-known ad hoc routing protocols. The results help clarifying the differences between P2P applications and client/server ones showing the usability of such networks, and discovering possible improvements in MANET routing.

The destination-sequenced distance-vector routing (DS-DV) [18], the dynamic source routing protocol (DSR) [10], and the ad hoc on demand distance vector (AODV) [19] are the protocols evaluated in this work. The reasons for the choice are twofold. First, the nature of those algorithms is distinct. While DSDV is pro-active, 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 one of these protocols 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. Section 2 discusses the related work and Section 3 presents a comparison between MANETs and P2P application networks. Section 4 briefly describes the routing protocols used in this work. In Section 5, the P2P application is discussed. The simulation scenarios are described in Section 6 and the performance evaluation metrics in Section 7. Section 8 discusses the simulation results. Section 9 provides a discussion about the P2P application in comparison with the client/server results. Finally, Section 10 presents our conclusions.

2. Related work

It was only recently that the scientists realized the synergy between MANETs and P2P networks and started studying both systems acting together. This was a very important step towards providing more applicability for MANETs. Nevertheless, still many open issues remain.

Schollmeier et al. [21] and Borg [1] discussed similarities and differences of MANETs and P2P networks. The former focus mainly in routing aspects and the latter discusses content discovery, security, quality of service, etc.

Kortuem presented Proem [13,12], a middleware platform for developing and deploying P2P applications tailored to personal area networks (PANs), a special class of MANETs.

Hu et al. [8] proposed dynamic P2P source routing (DPSR), a routing ad hoc protocol that integrates strategies used by DSR routing protocol and Pastry P2P protocol [20] to improve scalability.

Papadopouli and Schulzrinne [16,17] and Klemm et al. [11] presented P2P data sharing systems tailored to MANbreak ETs namely *seven degrees of separation* (7DS) and *optimized routing independent overlay network* (ORION), respectively. 7DS focus on enabling the exchange of data among peers not directly connected to the Internet by exploring peer mobility, while ORIOM concentrates on file sharing applications by setting up overlay routes on demand.

Franciscani et al. [6] concentrated on minimizing the impact of the highly dynamic topology obtained through the combination of P2P networks and MANETs. They proposed algorithms for configuring and reconfiguring these networks. In their algorithms, three combinations of neighborhood assignment are compared: (1) *regular*, where P2P neighborhood corresponds to the physical neighborhood; (2) *random*, where authors try to achieve the small-world [14,25] phenomenon by picking each neighbor at random among online peers; (3) and *hybrid*, where links are built following a hierarchy and each peer communicates through an intermediate.

Ding and Bhargava [4] performed a theoretical comparison between P2P systems over MANETs (broadcast over broadcast; broadcast; DHT over broadcast; DHT over DHT; and DHT) and presented important results in *O*-notation. Nevertheless, they do not evaluate real P2P systems and do not take into account practical aspects (e.g., mobility and channel error) in their work.

3. Comparison between MANETs and P2P application networks

P2P applications and MANETs have several aspects in common [22,9,13,1]. 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 MANETs these alterations are mainly caused by node mobility. That is, as a node moves, it might leave the transmission range area of its current neighbors and has its links broken as a consequence. Thus, in order to reestablish contact with peer entities, the peers must set new connections. Conversely, what causes the dynamic topology of P2P application networks is the low availability of 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.

Curiously, 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, on the other hand, 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, the 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 [22] are described in Table 1.

Table 1
Differences between P2P application networks and MANETs

Item	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

4. Ad hoc routing protocols

In this section, we briefly describe the routing protocols used in this work.

4.1. DSDV

The DSDV [18] is a variation of the distance vector routing protocol modified for ad hoc networks. The changes were performed in order to reduce looping properties that would be present in the original protocol. DSDV is a hop-by-hop routing and pro-active protocol that provides each node a routing table that lists the next-hop information for each reachable destination. Thus, it requires periodic broadcasting of routing updates and triggered beacon messages, which leads to an increase in routing overhead.

4.2. DSR

DSR [10] employs an on-demand approach regarding route discovery and maintenance processes. The key difference of the protocol from other on-demand routing protocols is the fact that it adopts the source routing strategy—as its own name indicates. That is, the complete path from the source to destination is carried in each packet. Such path is discovered through routing query broadcasts. DSR also provides each node a route cache for decreasing the number of control messages sent. In order to update its respective caches, every intermediate node makes use of the source route information available in the packet it forwards.

The main advantage of the approach adopted by DSR is that no additional mechanism is necessary to detect routing loops. The disadvantage, clearly, is the overhead caused by the introduction of source routing information in the header of the data packet.

4.3. AODV

The AODV [19] is a reactive protocol which combines both DSR and DSDV characteristics. It borrows the basic route discovery and route-maintenance of DSR as well as hop-by-hop routing, sequence numbers and beacons of DSDV. When a source node desires to establish a communication session, it initiates a route discovery process to locate the destination node, by generating a "route request" message, which might be replied by the intermediate nodes in the path to destination or the destination node itself. At the time of arrival, the "route reply" message contains the whole path to destination. To handle the case in which a route does not exist, or the query or reply packets are lost, the source node rebroadcasts the query packet if no reply is received by the source after a time-out.

5. Description of the P2P implemented protocol

In order to achieve the previously described purposes, 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 protocol, which is used for P2P 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, which 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 below.

5.1. Joining the network

A peer desiring to join the P2P application network starts by sending a broadcast-send message through the network in order to elect its "neighbors", which may be used for message flooding. The initial replies will settle virtual connections between the new servent and each answerer. Each one of these connections is maintained by an entry in a neighbors list which has predefined maximum size. 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.

5.2. Content discovery

The fact that a P2P application network does not possess a server that centralizes information complicates 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 through the network 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.

Table 2
Messages transmitted in P2P application network

Message type	Function	Size (Bytes)
broadcast-send	Look for neighbors	23
broadcast-reply	Answer a broadcast-send	38
ping	Check the activity of a peer	23
pong	Answer a ping	38
query-send	Search for a file	26
query-forward	Retransmit a query originated by another peer	26
query-reply	Answer a query (a *query-hit* has occurred)	26
push-request	Require the transfer of a file	51
pull-request	Transmit data (pieces of a file)	210 (maximum)

5.3. Content dissemination

After receiving the first reply, the "query-source" servent establishes an end-to-end communication with the "file-source". The file is fragmented into 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.

5.4. Controlled flooding

Each peer of the network maintains a cache in order to avoid duplicate query processing. This is possible since the query message is uniquely identified by the pair (*query-id*, *query-source*).

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.

5.5. Neighborhood control

P2P application networks have a dynamic behavior, as mentioned before. Peers can leave or join the network at anytime they necessitate. This 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 neighbor list and a broadcast-send message is sent to find another neighbor.

5.6. Message size

Messages that circulate among peers may have a variable size and the maximum value is 210 bytes, based on the Gnutella Protocol. Table 2 presents the message types sent by the peers with their respective functions and sizes.

6. Simulation scenarios

In order to evaluate the accomplish the purposes of this work, we have conducted simulation experiments using the network simulator (*ns*-2) [5] and its CMU wireless and mobility extension [24]. We have considered a set of default settings. Some of them were varied throughout the simulation experiments. The variations can give findings on the performance of the algorithms for diverse scenarios.

The default settings taken into consideration are the following. It was constructed a $200 \times 200\,m^2$ topology composed of 40 mobile nodes, 12 of which implementing a single instance of the P2P application.

The mobility scheme employed was the random way point (since it is frequently used for individual movements [2]). We have chosen as default settings a pause-time of 50 s and a maximum speed of 0.5 m/s.

The transmission range of all nodes was set to 50 m. The radio propagation model chosen was 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 300 s.

In respect to P2P application parameters, the maximum size of the neighbors list, for neighborhood control, was set to 3. Also, 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.

For controlled flooding, the TTL of the query-send messages were set to 3—large enough for queries to reach most of the peers in the simulated scenario. During the simulation, 10 searches are scheduled for each peer. The scheduling time is uniformly distributed and at each time the search is carried out only if the peer is part of the P2P network at the moment.

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 10 s. The broadcast-send interval time was 2 s.

In the simulation, nodes start with 100 J of energy each. The power loss for transmission was set to 0.330 and 0.230 W for reception [3].

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 simulation was run 33 times, with different seeds for the random number generator, on ns-2.1b8a [5] and its CMU wireless and mobility extension [24]. The results represent the average values on the 33 runs.

7. Evaluation metrics

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

7.1. 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, packet dropping and control overhead. 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. 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.

As a consequence, the network may not provide a good service for the application. We have chosen to vary the total number of queries generated by P2P peers (1, 10, 100, and 1000) as well as the average size of the files transferred through the network (1, 10, 100, and 100 kB) with the aim of investigating the protocols scalability. The results, for low, medium and high workload, are presented in Section 8. The following metrics were evaluated: number of initiated file transfers, throughput, percentage of queries not responded, delivery rate, energy consumption, and routing overhead associated to the ad hoc network.

7.2. Mobility

It represents the speed and pause time applied to the ad hoc nodes. Simulation experiments considering mobility were conducted, and the protocol capability in adapting to distinct speed (0.1, 0.5, 2.5, and 10 m/s) and pause time (0, 60, 120, 180, 240, and 300 s) of node values was analyzed. The metrics chosen were path length, connectivity among application peers, and latency.

7.3. 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 (1, 10, 100, and 100 m) and number of nodes (0, 20, 40, 60, and 80) 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 rate were the metrics employed to evaluate the three protocols.

7.4. Peer quantity

In order to investigate the influence of the amount of peers over the protocols, the number of nodes was left unmodified and the number of peers was varied (10, 20, 30, and 40). 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 consumption.

8. Simulation results

This section presents the results according to the four metrics described above.

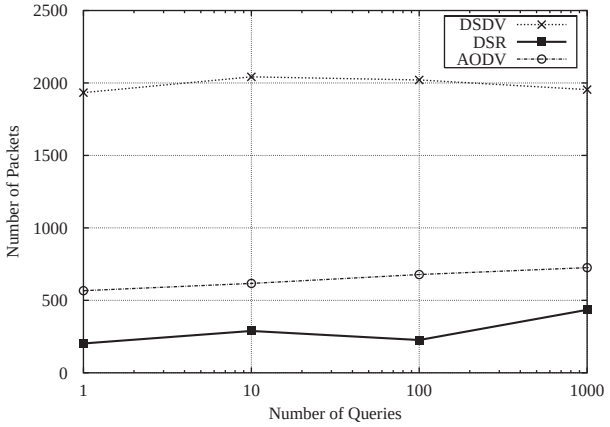
8.1. Workload

The three routing protocols introduced distinguishing amounts of overhead when the number of queries by a node was varied. As shown in Fig. 3(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—the DSR overhead, specifically, duplicated from one extreme of the *x*-axis to the other.

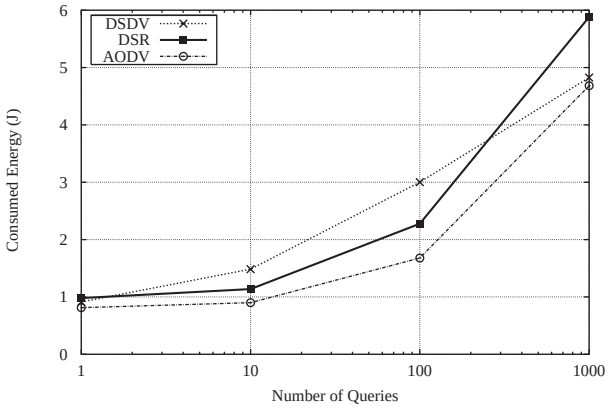
DSDV was the protocol which consumed the greatest amount of energy for lower, medium, and higher loads, as depicted in Fig. 3(b). This is a result of the higher number of control packets sent and received by the nodes.

Fig. 4(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 rate 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.

The results for initiated file transfers 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

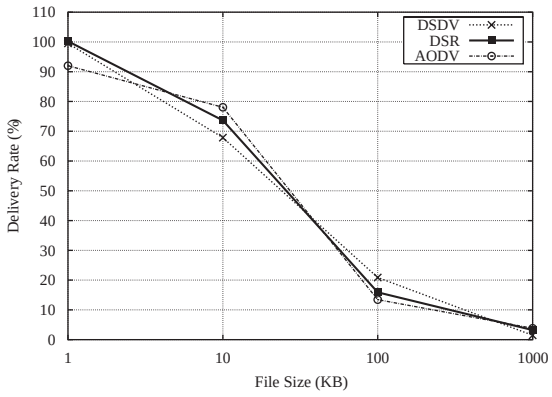


(a) Routing Overhead

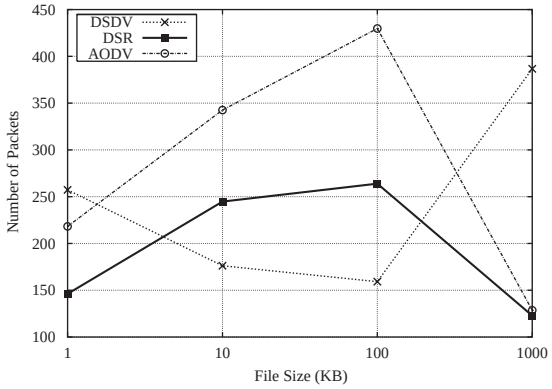


(b) Energy Consumption

Fig. 3. Number of queries variation: (a) Routing overhead; (b) energy consumption.



(a) Delivery Rate for Application



(b) pull-data Messages Received

Fig. 4. File size variation: (a) Delivery rate for application; (b) pull-data messages received.

of pull-request messages received in Fig. 4(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. 4(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, which caused difficulties for them to find routes on demand.

8.2. Mobility

Figs. 5(b) and (a) 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.

As the mobility was incremented, surprisingly, the connectivity degree rose (see Fig. 5(a)). 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 a peer movement. As a result, for high pause-time values, i.e., for low mobility, the number of queries not responded is also high.

Paradoxically, Fig. 6(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.

Fig. 6(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 and latency, when mobility was increased. The term hops suggests the average amount of hops for a query to reach an information source. 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.

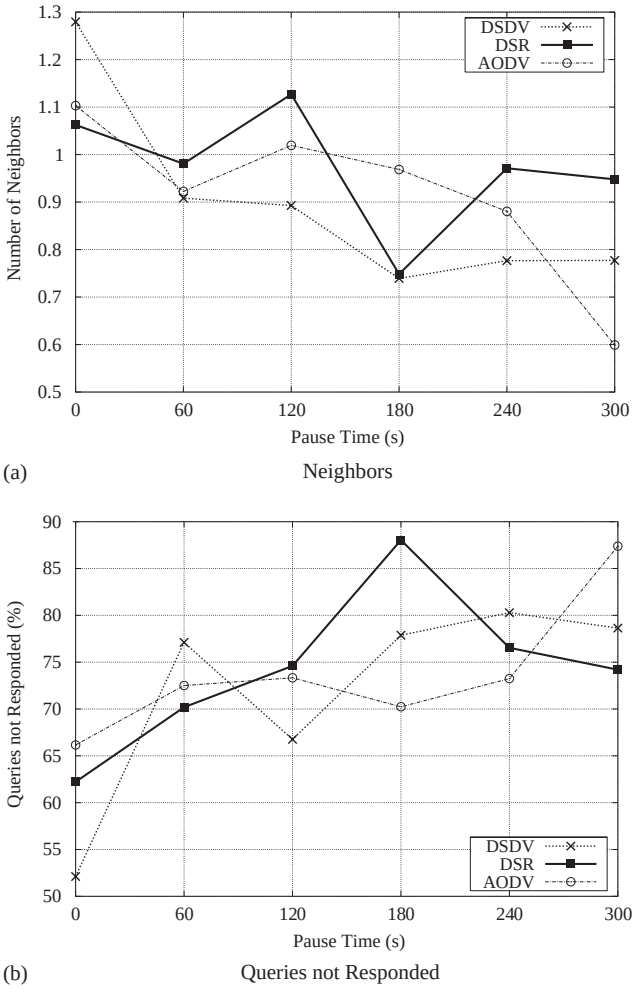


Fig. 5. Pause time variation: (a) Neighbors; (b) queries not responded.

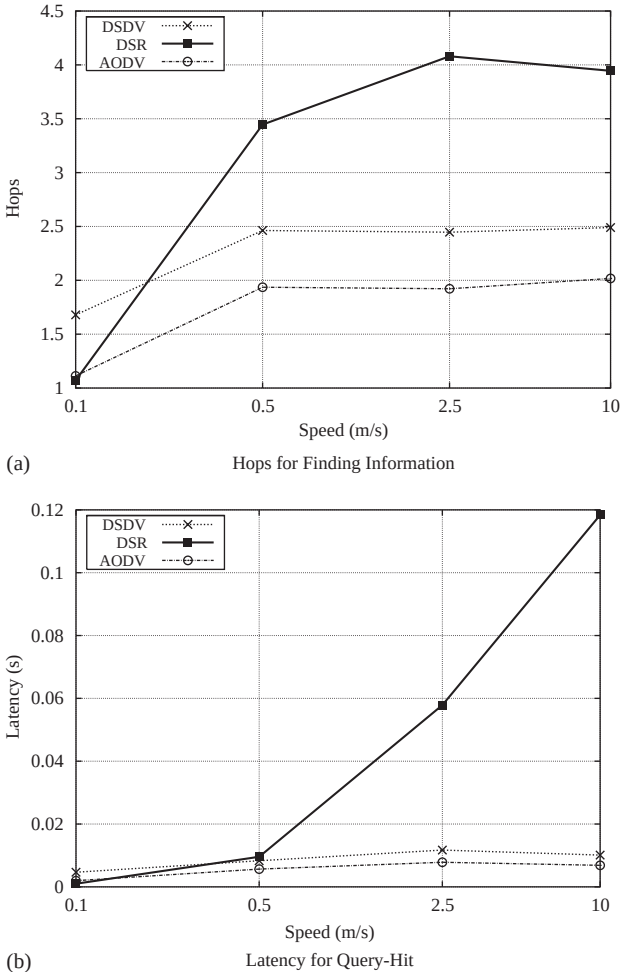


Fig. 6. Speed variation: (a) Hops for finding information; (b) latency for query-hit.

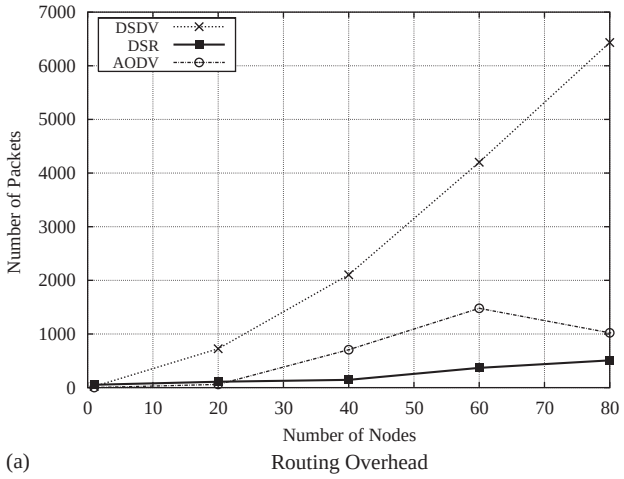
8.3. Network density

Concerning routing overhead, as shown in Fig. 7(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.

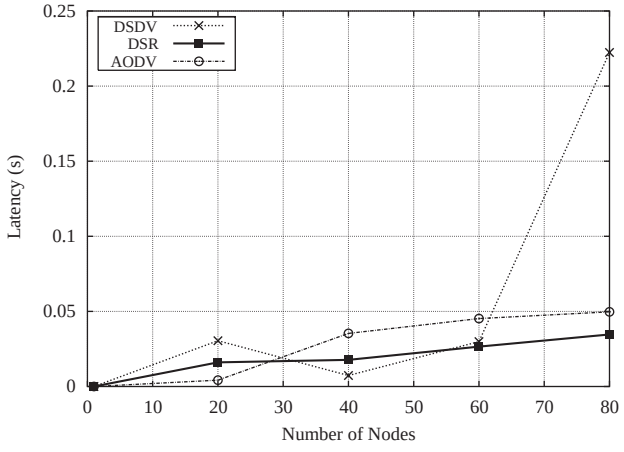
The three protocols behaved equivalently for both number of nodes and transmission range variation with respect to connectivity among application peers. The protocols had their latency intensified for a denser network, as presented in Fig. 7(b). Particularly, DSDV appears to be less scalable regarding this metric due to its routing overhead, as previously highlighted.

Regarding both number of queries not responded and network delivery rate, DSDV, DSR, and AODV performances were similar. The former protocol, despite producing more routing overhead, managed to maintain the same delivery rate for a denser network. Fig. 8(a) shows the results obtained for the delivery rate.

In respect to path length, it was observed that this metric is very dependable on network density, as shown in Fig. 8(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 that 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.

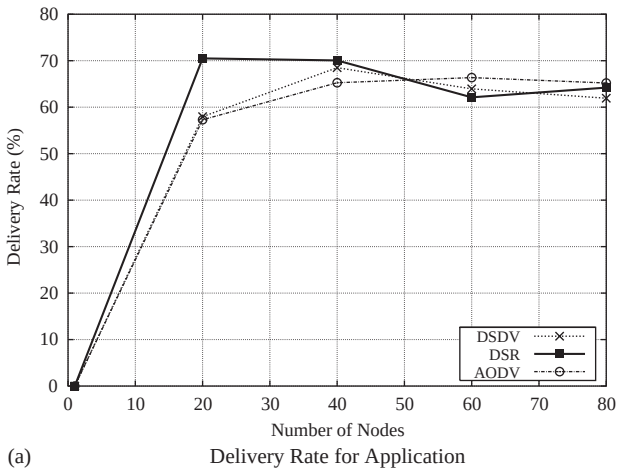


(a) Routing Overhead

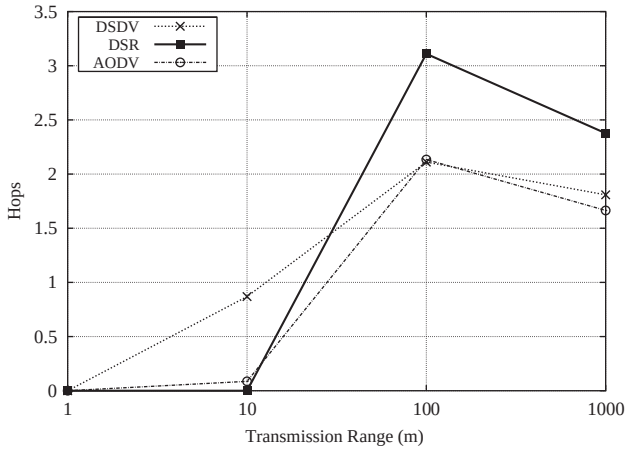


(b) Latency for Receiving a query-reply

Fig. 7. Nodes variation: (a) Routing overhead; (b) latency for receiving a query-reply.



(a) Delivery Rate for Application



(b) Hops for Finding Information

Fig. 8. Nodes and transmission range variation: (a) Delivery rate for application; (b) hops for finding information.

8.4. Peers

Fig. 9(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 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. 9(b) presents the energy consumption. 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 consumption (0.35 J less, approximately), since it possess the lower overhead.

9. 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 from [2].

First, regarding mobility, P2P and client/server applications exhibit considerable differences. Unlike the

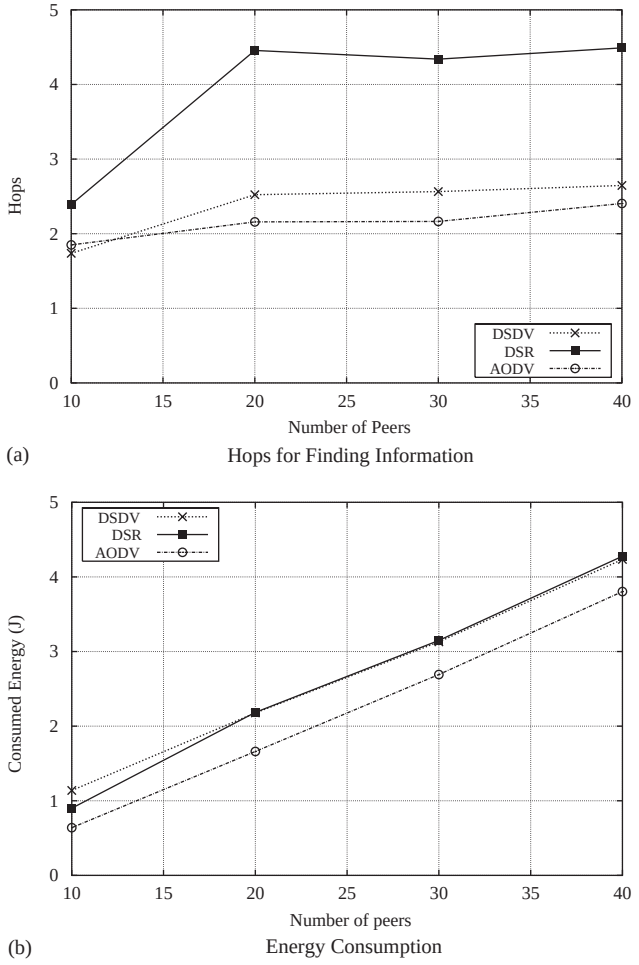


Fig. 9. Peers variation: (a) Hops for finding information; (b) energy consumption.

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 rate, the network performance supporting a P2P application presented a worse result. With the increase in the amount of nodes, the rate 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 that 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.

10. Conclusion and future work

In the last few years, mobile ad hoc networks (MANETs) 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 MANET where three different routing protocols were considered. It is interesting to notice that each of the protocols that was 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. Some of the strategies, which may be adopted are: modify existent ad hoc protocols or even propose other ones, use multicast protocols to disseminate information, and develop a middleware so that one layer—network or application—can take better advantage of the services of the other layers.

References

[1] J. Borg, A comparative study of ad hoc & peer to peer networks, Master's Thesis, University College London, 2003.

[2] J. Broch, D.A. Maltz, D.B. Johnson, Y.-C. Hu, J. Jetcheva, A performance comparison of multi-hop wireless ad hoc network routing protocols, in: Proceedings of the Fourth annual ACM/IEEE International Conference on Mobile Computing and Networking, ACM Press, Dallas, Texas, United States, 1998, pp. 85–97.

[3] J.-C. Cano, P. Manzoni, A performance comparison of energy consumption for mobile ad hoc network routing protocols, in: Eighth International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS'00), IEEE Computer Society, San Francisco, CA, 2000, pp. 57–64.

[4] G. Ding, B. Bhargava, Peer-to-peer file-sharing over mobile ad hoc networks, in: Second IEEE Annual Conference on Pervasive Computing and Communications Workshops, Orlando, Florida, 2004, pp. 104–108.
[5] K. Fall, K. Varadhan, Network Simulator Notes and Documentation, The VINT Project, February 2001.
[6] F.P. Franciscani, M.A. Vasconcelos, R.P. Couto, A.A.F. Loureiro, (Re)Configuration algorithms for peer-to-peer over ad hoc networks, J. Parallel Distrib. Comput. (JPDC) 65 (2) (2005) 234–245.
[7] Z.J. Haas, J. Deng, B. Liang, P. Papadimitratos, S. Sajama, Wireless ad hoc networks, in: J.G. Proakis (Ed.), Wiley Encyclopedia of Telecommunications, Wiley, New York, 2002.
[8] Y.C. Hu, S.M. Das, H. Pucha, Exploiting the synergy between peer-to-peer and mobile ad hoc networks, in: HotOS-IX: Ninth Workshop on Hot Topics in Operating Systems, Lihue, Kauai, Hawaii, 2003.
[9] Y.C. Hu, S.M. Das, H. Pucha, Exploiting the synergy between peer-to-peer and mobile ad hoc networks, in: Proceedings of the Ninth Workshop on Hot Topics in Operating Systems (IX HotOS), 2003.
[10] D.B. Johnson, D.A. Maltz, Dynamic source routing in ad hoc wireless networks, in: C.E. Perkins (Ed.), Ad Hoc Networking, Addison-Wesley, Reading, MA, 2001, pp. 139–172, (also appeared in IEEE Computer Communications).
[11] A. Klemm, C. Lindemann, O.P. Waldhorst, A special-purpose peer-to-peer file sharing system for mobile ad hoc networks, in: IEEE Semiannual Vehicular Technology Conference (VTC2003-Fall), 2003.
[12] G. Kortuem, Proem: a middleware platform for mobile peer-to-peer computing, SIGMOBILE Mobile Computing and Communication Review, vol. 6, No. 4, 2002, pp. 62–64.
[13] G. Kortuem, J. Schneider, D. Preuitt, T.G.C. Thompson, S. Fickas, Z. Segall, When peer-to-peer comes face-to-face: collaborative peer-to-peer computing in mobile ad hoc networks, in: IEEE First International Conference on Peer-to-Peer Computing, Linkopings, Sucia, 2001, pp. 75–91.
[14] S. Milgram, The small-world problem, Psychol. Today 1 (1) (1967) 60–67.
[15] A. Oram, Peer-To-Peer: Harnessing the Power of Disruptive Technologies, First ed., O'Reilly, 2001, iSBN:0-596-00110-X.
[16] M. Papadopouli, H. Schulzrinne, Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices, in: Second ACM International Symposium on Mobile ad hoc Networking & Computing, ACM Press, New York, 2001, pp. 117–127.
[17] M. Papadopouli, H. Schulzrinne, A performance analysis of 7ds a peer-to-peer data dissemination and prefetching tool for mobile users, in: Advances in Wired and Wireless Communications, IEEE Sarnoff Symposium Diges, Ewing, USA, 2001.
[18] C.E. Perkins, P. Bhagwat, Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers, in: Proceedings of ACM Conference on Communications Architectures (SIGCOMM'94), Protocols and Applications, ACM Press, London, United Kingdom, 1994, pp. 234–244.
[19] C.E. Perkins, E.M. Royer, Ad hoc on-demand distance vector routing, in: Proceedings of the Second IEEE Workshop on Mobile Computing Systems and Applications, New Orleans, LA, 1999, pp. 90–100.
[20] A. Rowstron, P. Druschel, Pastry: scalable, distributed object location and routing for large-scale peer-to-peer systems, in: Proceedings of IFIP/ACM International Conference on Distributed Systems Platforms (Middleware'01), 2001, pp. 329–350.
[21] R. Schollmeier, I. Gruber, M. Finkenzeller, Routing in peer-to-peer and mobile ad hoc networks: a comparison, in: International Workshop on Peer-to-Peer Computing, Pisa, Italy, 2002, held in conjunction with IFIP Networking 2002.
[22] R. Schollmeier, I. Gruber, M. Finkenzeller, Routing in peer-to-peer and mobile ad hoc networks: a comparison, in: Revised Papers from the NETWORKING 2002 Workshops on Web Engineering and Peer-to-Peer Computing, Springer-Verlag, Pisa, Italy, 2002, pp. 172–186.
[23] D. Talia, P. Trunfio, Toward a synergy between p2p and grids, IEEE Internet Comput. 7 (4) (2003) 94–95.
[24] The cmu monarch projects wireless and mobility extension to ns, work in Progress, September 2004.
[25] D. Watts, S. Strogatz, Collective dynamics of = small-world' networks, Nature 393 (6) (1998) 440–442.
[26] L. Zhou, Z.J. Haas, Securing ad hoc networks, IEEE Network 13 (6) (1999) 24–30.

Leonardo B. Oliveira received his B.Sc. (2003) and M.Sc. (2004) degrees in Computer Science from Federal University of Minas Gerais (UFMG), Brazil. He is currently pursuing a doctoral degree at University of Campinas (UNICAMP), Brazil. Leonardo's primary research interests include P2P networks over MANETs and security in sensor ad hoc networks.

Isabela G. Siqueira received her B.Sc. degree in Computer Science from the Federal University of Minas Gerais (UFMG), Brazil, in 2003. She is currently pursuing her M.Sc. degree in Computer Science at the same university. Her main research interests include Peer-to-Peer computing, MANETs, and topology control in sensor networks.

Antonio A.F. Loureiro holds a B.Sc. and a M.Sc. in Computer Science, both from the Federal University of Minas Gerais (UFMG), and a Ph.D. in Computer Science from the University of British Columbia, Canada. Currently he is an Associate Professor of Computer Science at UFMG. His main research areas are mobile Computing, distributed algorithms, and network management.