Cross-layer Interception Caching for MANETs

F.J. González-Cañete, E.Casilari, and A. Triviño-Cabrera

Dpt. Tecnología Electrónica, University of Málaga
Campus de Teatinos, ETSI Telecomunicación, Málaga, Spain
+34 952 13 71 76
{fgc,ecasilari,atc}@uma.es

Abstract. In this work we study the interception of the requests performed by the mobile nodes in a wireless network. This interception can be achieved because a local cache is implemented in each wireless device. In that way, the nodes can serve the documents instead of forwarding the requests to the data servers. In our proposal, the interception is enhanced so it is also implemented when the mobile nodes create the route to the data servers. Using cross-layer information, the routing algorithm can discover the location of the documents disseminated across the network. By means of simulations, we evaluate the performance of the proposed interception mechanism. We study the effect of the network load, the expiration time of the documents, the requests pattern and the cache size on these cache mechanisms. Under all the assumptions the proposed scheme reduces the delay, the network traffic and the amount of timeouts when the servers are not reachable.

Keywords: MANET, local caching, interception caching, AODV.

1 Introduction

In the mobile computing paradigm, devices are expected to demand the access to the Internet anywhere and anytime. Some geographic and economic constraints may limit the access through infrastructure networks. In order to overcome this restriction, the MANET (Mobile Ad hoc NETwork) technology outstands as a feasible solution. Mobile ad hoc networks are composed of wireless devices that communicate among themselves without any specific router. Actually, routing tasks are transferred to the mobile nodes so two distant devices (not directly connected by a wireless link) can exchange packets through the collaboration of intermediate nodes that retransmit and route the packets to the final destination. As packets traverse multiple links or hops, these networks are also known as multihop wireless networks.

One member of the MANET can be an Internet Gateway [1] through which the network can access some external hosts, DHCP (Dynamic Host Configuration Protocol.) servers or HTTP (Hyper Text Transfer Protocol) servers. However, the use of the Internet Gateway must cope with the peculiarities of mobile devices in a MANET. Firstly, the Internet Gateway may not be permanently available for the MANET as the mobility of the nodes may lead to situations where this element is not reachable. Additionally, the wireless medium has restricted bandwidth. In particular, the links to the Gateway will be notably saturated when mobile nodes inject traffic to the Internet. This could provoke a bottleneck effect in the Internet Gateway. Therefore, web technologies in a MANET must take into account the temporary connection to the Internet while they should also reduce the signalling and the traffic load through the Internet Gateway.

Taking into account these new requirements, the traffic could benefit from caching techniques. When using caching, devices store some documents which were previously requested to a server. Mobile devices in a MANET can take advantage of others' storage so that the documents can be served without accessing the server and, in turn, without occupying the wireless links to the Gateway. Furthermore, the requests can be satisfied even when the Gateway is not reachable. In addition, the traffic generated to get the document is reduced as an intermediate node serves it. This operation is called interception caching. Although interception caching is not exclusively conceived for ad hoc networks, the technique has been adapted and optimized for a MANET context. Specifically, this paper evaluates a cross-layer technique by which interception caching uses some routing resources to expand the use of the caches.

The paper is structured as follows. Section 2 presents the related work. In section 3, the caching scheme that intercepts the requests is described. This scheme exchanges some cross-layer information using the routing protocol in order to know where the documents are located in the MANET. Section 4 details the simulation model. Section 5 studies the performance of a MANET when the interception scheme proposed in Section 3 is employed. The study takes into account the time between requests, the expiration time of the documents, the cache size and the distribution of the requests. Finally, section 5 outlines the main conclusion and suggests possible future work.

2 Related work

Although there are many works devoted to caching technology in the wired environment the particular literature about caching in MANETs is not so abundant.

Some caching schemes employ broadcast message as the first choice in order to find the documents in the network. These broadcast messages can be sent to the entire network, as in the case of MobEye [2] or following a more restrictive approach that limits the distance of the messages as used in SimpleSearch [3]. Similarly, the caching scheme proposed by Moriya in [4] sends the broadcast messages to the neighbourhood so that, if the document is not found, the request is transmitted to the server.

Other caching schemes employ information of the location of the documents in the network. Nodes obtain this information by analysing the messages that they forward. As examples of this kind of caching schemes we can mention: DGA (Distributed Greedy Algorithm) [5], Wang [6], Cho [7], CacheData, CachePath, HybridCache [8] and GroupCaching [9].

Some caching schemes assign a predefined role to every node in the wireless network. In that way, they can perform as caching nodes, requesting nodes, coordinator nodes, gateway nodes, etc. CC (Cluster Cooperative) [10] and Denko [11] are examples of this kind of caching policy.

However, some caching schemes follow a hybrid policy. Thus, the caching schemes COOP [12] and IXP/DPIP (IndeX Push/Data Pull/Index Push) [13] employ network information and broadcast requests. On the other hand, COACS (Cooperative and Adaptive Caching System) [14] and GROCOCA (Group-based Cooperative Caching) [15] are role-based caching schemes that also utilize information obtained from the network.

3 Caching in MANETs

In this section we define the formulation of caching in MANETs following a modified version of the notation presented in [16].

For a mobile node in a MANET that requests information to a data server or to the Internet through a Gateway, the first strategy to reduce the traffic in the network is by implementing a local cache for the requested documents. This cache allows that future requests to the same documents will be served by the local cache even if the Gateway is not available, thus they will not produce any traffic in the ad hoc network.

Formally, let $MN = \{w_1, w_2, ..., w_w\}$ be the set of w mobiles nodes in a MANET. Let $U = \{1, 2, ..., n\}$ be the universe of n documents that can be requested by the mobile nodes, where $s(i)$ represents the size of the document i with $1 \leq i \leq n$. $TTL(i)$ is defined to be the time when the document i expires and becomes obsolete. Let $R_j = \{r_{j1}, r_{j2}, ..., r_{jm}\}$ be the sequence of requests performed by the node j, where r_{jk} denotes the request of mobile node j in the k instant taking into account that $r_{jk} \in U$. The destination of the requests in R_j is a fixed node w_{DS} (Data Server) in the Internet that has access to all the documents in U. The data server is supposed to be accessed through the Internet Gateway.

Let us implement a local cache in each mobile node in the MANET. B_{jk} denotes the set of documents stored in the cache of node j at time k, where $B_{jk} \subset U$. The set of documents stored in the caches must satisfy the properties (1) and (2):

$$\sum_{i \in B_{jk}} s(i) \leq S_j \qquad \textbf{(1)}$$

$$\forall i \in B_{jk} \Rightarrow TTL(i) > k \qquad \textbf{(2)}$$

where S_j is the size of the cache in node j. Property (2) states that the documents stored in the caches cannot be obsolete. Therefore, the sequence of states $(B_{j0}, B_{j1}, ..., B_{jm})$ indicates the states of the cache in node j as the requests in R_j are resolved. B_{j0} is the initial state when the cache is empty and B_{jm} is the state when all the requests have been served.

Let $p_{k,ij} = \{w_i, w_u, w_v, ..., w_j\}$ be the active route between node i and node j at instant k, defined as the set of mobile nodes that are necessary to reach node j from node I at the instant k. Only two consecutive nodes in a path are directly connected, that is, a wireless link between them is available. If $p_{k,ij} = \varnothing$ then there is no route created to reach node j from node i, otherwise $card(p_{k,ij}) \geq 2$, where $card(p_{k,ij})$ is the cardinality of the set $p_{k,ij}$. We can define the distance between node i and j at the instant k as (3):

$$dist(w_i, w_j)_k = card(p_{k,ij}) - 1 \tag{3}$$

This distance defines the number of hops needed to reach node j from node I at the instant k. When the distance between nodes i and j is one, the nodes are neighbour. On the other hand, we consider that $dist(w_i, w_j)_k = \infty$ if there is not a route created between the nodes i and j at the instant k ($p_{k,ij}$=∅).

We define the local cache hit sequence in node j to be $(h^l_{j1}, h^l_{j2}, ..., h^l_{jm})$ where h^l_{jk} is defined in (4).

$$h^l_{jk} = \begin{cases} 1 & \text{if } r_{jk} \in B_{j(k-1)} \\ 0 & \text{if } r_{jk} \notin B_{j(k-1)} \end{cases} \tag{4}$$

When h^l_{jk} equals *1* a local cache hit is considered in node j, otherwise a local cache miss is assumed as the document required is not in the local cache at that moment.

The cache replacement policy decides which documents must be evicted from the cache in order to make room for the new ones. The goal of the caching scheme is to retain in the cache the documents that have a high probability to be requested again near in time.

Given a request sequence R_j in node j, a cache of size S_j and an initial state B_{j0}, the replacement policy produces a cache state sequence $(B_{j1}, ..., B_{jm})$, we have for k=1,…,m the equation (5):

$$B_{jk} = \begin{cases} (B_{j(k-1)} - \varepsilon_{jk}) \cup \{r_{jk}\} & \text{if } r_{jk} \notin B_{j(k-1)} \\ B_{j(k-1)} & otherwise \end{cases} \tag{5}$$

In the first case considered in the previous equation, the document requested is not stored in the cache of node j (cache miss). The term ε_{jk} (with $\varepsilon_{jk} \subset B_{j(k-1)}$) denotes the set of documents that have to be evicted from the cache in order to make room for the new one in the request r_{jk}. If there is enough room for the new document, ε_{jk} is an empty set. In the second case there is a local cache hit and the cache remains unchanged.

The next step compels the mobile nodes to cooperate between them in order to respond to the requests of the other nodes using their local caches. As an example, Fig. 1 shows a snapshot of a MANET where *DS* indicates a Data Server node, that is, the node that physically stores all the documents. This node is accessed through a Gateway (*GW*). Nodes *1*, *2*, *3* and *4* are user nodes that request documents to *DS*. The connections between the nodes indicate the existing wireless links created by the routing protocol.

Let us suppose that node *2* requests the document *A*. The request will be forwarded using the routing protocol to the Gateway. The data server will respond with the document *A* using the reverse route and the node *2* will store *A* in its local cache. In the case that node *3* requests the same document *A* to *DS*, the request will reach node *2*, which can respond to node *3* with the copy of the document *A* stored in its local cache. The interception of the requests reduces the number of hops necessary to obtain the document and hence the number of forwarding messages along the network.

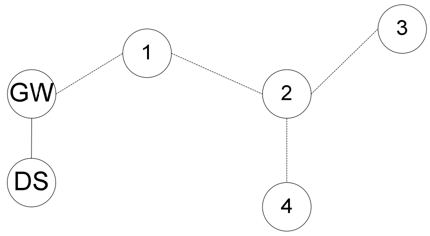


Fig. 1. Example of a MANET connected to a Data Server (DS).

We define the interception cache hit sequence in node *j* to be $(h^i_{j1}, h^i_{j2}, ..., h^i_{jm})$ where h^i_{jk} is defined in (6):

$$h^i_{jk} = \begin{cases} 1 & \text{if } r_{jk} \notin B_{j(k-1)} \wedge r_{jk} \in B_{i(k-1)} \mid i \neq j \wedge i \neq DS \wedge w_i \in p_{k,jDS} \\ 0 & otherwise \end{cases} \quad \textbf{(6)}$$

where w_i indicates an intermediate wireless node that has a copy of the document requested in its local cache. The node w_i is located in the route of the request r_{jk} from the node *j* to the server node *DS*. When $h^i_{jk}=1$ an interception cache hit is considered in node *j*. As $dist(w_j, w_i)_k < dist(w_j, w_{DS})_k$, the number of hops needed to satisfy the request r_{jk} is diminished as well as the time to serve the request.

The proposed interception caching procedure works in the cases when the routing paths from the source of the requests to the data servers are created. However let us suppose the situation presented in Fig. 2 (which follows a similar network topology of the example in Figure 1): Node *1* has moved outside the coverage area of *DS* and hence node *2* cannot access *DS* directly. Nodes *3* and *4* are located in the coverage area of node *2* but there is no route created between them. When node *4* proceeds to request the document *A* to *DS*, it detects that there is no path to the Gateway and therefore, DS is unreachable. The request is not served even when node *2* has a valid copy of the document *A* in its local cache.

This problem can be solved if the routing algorithm is involved in the process of looking for the documents in the MANET. Let us suppose that the AODV protocol [17] is utilized. The broadcast RREQ (Route Request) message sent in order to create the route to the gateway could include information about the demanded document, that is, a field with the document identification is 'piggybacked' (inserted) into the RREQ message. In our example, node *4* will broadcast a RREQ in order to create the route to *DS* with the information of the document *A*. When node *2* receives the RREQ, it checks if a document request is piggybacked in the routing message. If this is the case, the node extracts the information and verifies if there is a copy of the document in its local cache. If so, node *2* responds to node *4* using a RREP (Route Reply) message including the piggybacked identification of the document. As the node *4* receives the RREP and hence the route between node *4* and *2* is created, node *4* directly sends the document request to node *2*.

By using this procedure nodes in the MANET can access to the disseminated documents even if the data server is inaccessible.

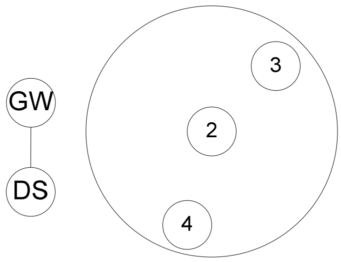


Fig. 2. Example of an isolated MANET.

We define the route creation interception cache hit sequence in node *j* to be $(h^{rci}_{j1}, h^{rci}_{j2}, ..., h^{rci}_{jm})$ where h^{rci}_{jk} is defined in (7):

$$h^{rci}_{jk} = \begin{cases} 1 & \text{if } r_{jk} \notin B_{j(k-1)} \wedge r_{jk} \in B_{i(k-1)} \mid i \neq j \wedge i \neq DS \wedge p_{k,jDS} = \varnothing \\ 0 & otherwise \end{cases} \quad (7)$$

where w_i indicates a wireless node that replies to the request of a route from node *j* to the server node *DS* because it has a copy of the document requested in its local cache when it receives the RREQ message. When $h^{rci}_{jk}=1$ a route creation interception cache hit is considered to have occurred in node *j*. Again, as $dist(w_j, w_i)_k < dist(w_j, w_{DS})_k$ the amount of hops required to satisfy the request r_{jk} (and consequently the delay perceived to obtain the document) is diminished.

In order to implement the proposed piggyback method using AODV routing protocol, the RREQ and RREQ messages must be modified. Fig. 3 shows the modified version of the RREQ message. The idea is that RREP packets also includes the identifier of the document.

0 7	8					13 23	24 31
Type	J	R	G	D	U	Reserved	hop count
RREQ id							
destination IP address							
destination sequence number							
originator IP address							
originator sequence number							
document id							

Fig. 3. Modified AODV RREQ message.

Typical AODV RREQ message contains a field reserved for future use but, unfortunately, they are 11 and 9 bits long respectively. That space is not enough to handle the identifier of a document, so we propose, for simulation purposes, to add a 32 bit field including the identification (e.g. the URL) of the document to be requested compressed using Bloom Filters [18].

4 Simulation model

By means of simulations, we have evaluated the performance benefits of the caching techniques proposed in section 2, especially the cross-layer interception scheme. The simulations are based on the network simulator NS-2.33 [19] which is the most popular simulator for the researches on ad hoc networks [20].

Table 1 summarizes the main simulation parameters. We will consider W=50 mobiles nodes distributed in a 1000 meters by 1000 meters area. The mobile nodes follow the Random Waypoint mobility pattern [21] moving at a constant speed of 1 m/s with no pause time when they reach the destination.

Table 1. Simulation parameters.

Parameter	Default	Tested values
Simulation area (meters)	1000x1000	
Number of nodes	50	
Number of Documents	1000	
Documents size (bytes)	1000	
Number of requests per node	10000	
Simulation time (s)	20000	
Timeout (s)	3	
TTL (s)	2000	250-500-1000-2000-∞
Mean time between requests (s)	25	5-10-25-50
Traffic pattern (Zipf slope)	0.8	0.4-0.6-0.8-1.0
Replacement policy	LRU	
Cache size (number of documents)	100	25-50-100-200
Warm-up (requests)	2000	
Ad hoc routing protocol	AODV	
MAC protocol	802.11b	
Propagation model	Two Ray Ground	
Coverage radio (meters)	250	
Mobility pattern (Random WayPoint)	Min. and max. speed: 1m/s Pause time: 0 s	

We consider n=1000 different documents (identified by a specific number) distributed between two fixed (immobile) servers. Those servers are located at positions (x,y)=(0,500) and (x,y)=(1000,500) in the simulation area respectively. Each server is directly connected to a Gateway. For our study, we assume that the Gateway and the servers are the same nodes. With this assumption, we avoid that the effects of the connection between these two elements alter our analysis. To distribute the traffic, documents with an odd identifier are placed in a server while even-numbered documents are located in the other one. In addition, each document has an associated TTL time that determines the instant when the document expires and it is considered obsolete. The expired documents stored in the local caches are evicted in order to make room for fresh documents. We have considered an exponential distribution with a mean between 250 and 2000 seconds for the TTL of the documents. In that way we model both a high and low variability of the documents. In addition we also consider the case of an infinite TTL for the documents, that is to say, the case where the documents never expire. The size of all the documents is constant and set to 1000 bytes.

Each node that is not a server is programmed to generate requests to the servers along the simulation time. When a request is served another request is generated by the same node. The idle time between the reception of a response and the next request follows an exponential distribution with a mean between 5 and 50 seconds. Using this variety of values we evaluate a wide range of patterns for the node activity (and consequently for the networks load). A document is requested again if the response of the present request is not served before a defined timeout is triggered.

The pattern of requests of the documents follows a Zipf-like distribution that has been demonstrated to properly characterize the popularity of the Web documents in the Internet [22]. The Zipf law asserts that the probability *P(i)* for the *i-th* most popular document to be requested is inversely proportional to its popularity ranking as shown in eq. (8)

$$P(i) = \frac{\beta}{i^{\alpha}} \tag{8}$$

The parameter α is the slope of the log/log representation of the number of references to the documents as a function of its popularity rank (i) while the β parameter is the displacement of the function. In our simulations, the slopes selected to generate the requests are 0.4, 0.6, 0.8 and 1.0.

Finally, every node implements a local cache that employs the Least Recently Used (LRU) replacement policy. This replacement policy evicts the documents that were referenced longest ago. All nodes have the same cache size, which has been configured to fit 25, 50, 100 and 200 documents. In order to avoid cold start influences, that is to say, cache misses because the cache is empty, the local caches are "warmed up" using the first 20% of the requests.

The simulation time (k) has been set to 20000 seconds.

5 Performance evaluation

Each scenario has been executed five times using the same TTL and time between requests but using a different request distribution. The performance evaluation presented is the mean of the results obtained for the five simulations. All the figures of this section include the 95% confidence interval for every measured parameter.

As performance metrics we utilise:

- Delay: Defined as the time elapsed between the request of a document and the reception of the response.
- Percentage of timeouts: Defined as the proportion of requests that must be requested again because the response does not reach the destination before a timeout.

Also, we define the local hit ratio (LHR), interception hit ratio (IHR) and AODV interception hit ratio (RCHR) in (9), (10) and (11) respectively:

$$LHR = \frac{1}{w}\sum_{j=1}^{w}\frac{\sum_{k=1}^{m} h_{jk}^{l}}{card(t_j)}. \quad \textbf{(9)}$$

$$IHR = \frac{1}{w}\sum_{j=1}^{w}\frac{\sum_{k=1}^{m} h_{jk}^{i}}{card(t_j)} \quad \textbf{(10)}$$

$$RCHR = \frac{1}{w}\sum_{j=1}^{w}\frac{\sum_{k=1}^{m} h_{jk}^{rci}}{card(t_j)} \quad \textbf{(11)}$$

where $t_j \subset R_j$ represents the subset of requests that the node *j* has performed until the end of the simulation time *t* with $t \le m$.

We will compare the performance of a MANET in four situations: 1) the nodes do not implement any cache mechanism (No cache), 2) the nodes only use the local cache (LC), 3) the nodes implement local cache whilst the intermediate nodes can also intercept the requests (I) and 4) the nodes implement the previous schemes as well as the route creation interception using AODV (AODV Interception).

5.1 Effects of the mean time between requests

Fig. 4 represents the mean delay, timeouts and cache hits as a function of the mean time between requests. As it can be observed in Fig. 4 a) the delay is decreased if a local cache is used and this reduction is greater as we implement the interception and AODV interception caching. The delay reduction is independent of the mean time between requests and is about 40% for the last case, in which all caching schemes are utilised. On the other hand Fig. 3 b) shows that for highly loaded networks (low time between requests) the timeouts are slightly diminished using the local cache while they are drastically reduced (almost 50% in high loaded networks) if the interception is implemented. This is due to the fact that with the interception mechanism the number of access to the highly saturated servers is dramatically reduced. As the network load decreases (high time between requests) the interception techniques also outperform the local caching approach, although the reduction in the percentage of timeouts is less meaningful. Finally, Fig. 3 c) illustrates that the local cache hits are reduced as the time between requests increases. This is due to the expiration time of the documents. As the nodes perform fewer requests, the documents stored in their local caches expire and they have to be requested again. The interception caching is also reduced as the time between requests increases because the routes to the servers expire and they must be created again, that is, the AODV interception can operate and that is why the number of AODV interception increases as the network load diminishes.

This study reveals that our proposal of routing interception outperforms the standard interception for medium and low loaded networks. In the best case, the reduction of the delay is about 14% and the number of timeouts is reduced by 5%. For high loaded networks the improvement is not very significant.

5.2 Effects of the TTL

Fig. 5 illustrates the delay, timeouts and cache hits as a function of the TTL of the documents. As the mean TTL increases the documents can be stored longer in the local caches because they are considered to be obsolete later. Fig. 5 a) and b) show that the delay and percentage of timeouts are reduced as the TTL increases if we use any of the caching techniques proposed. As in the case when traffic load is varied, results indicate that the combination of caching schemes outperforms the local and interception caching. Fig. 5 c) reveals that all the cache hits are increased as the mean TTL increases. Supposing that documents do not expire the percentage of cache hits almost reach 65%, i.e. 65% of the requests are served by the local caches or other nodes instead of by the servers. In that way the servers load is dramatically reduced.

This study demonstrates that the proposed routing interception outperforms the standard interception scheme for a low variability of the documents. If the documents do not expire the reduction of the delay is about 20% and the number of timeouts is reduced by 10%. As the TTL of the documents is decreased the difference between the standard interception and our proposal is reduced. For high variable documents the improvement is not significant.

5.3 Effects of the Zipf slope α

Fig. 6 displays the delay, timeouts and cache hits as a function of the Zipf slope (α parameter) selected for the pattern of the requests. As the Zipf slope is close to 1.0 there is a small set of documents that are requested more times and hence the local cache will also hit more frequently. Fig. 6 c) confirms this behaviour. On the other hand the interception and AODV interception hit rate remain practically unchanged as we vary the Zipf slope. As in the previous studies, Fig. 6 a) and b) show that the policy using all the techniques outperforms the other methods, considering both the delay and the percentage of timeouts. As the Zipf slope raises, the benefits of employing a caching scheme is increased compared to the scheme without cache.

We can conclude that the use of the routing interception outperforms the standard interception for all the studied Zipf slopes. In fact, the interception mechanism reduces the delay by 10% and the number of timeouts by 5%. Only the local cache hits are incremented as the Zipf slope is increased.

5.4 Effects of the cache size

Fig. 7 illustrates the delay, timeouts and cache hits as a function of the cache sizes. These figures show that for a cache size greater than 50KB (50 documents – 5% of the documents) there is not any relevant improvement in the network performance. The delay, timeouts and cache hits remain constant for cache sizes greater than 50 documents. Further studies will be needed in order to evaluate the behaviour with smaller caches. Anyway the routing interception caching outperforms the standard interception for all the cache sizes studied, reducing the delay by 10% and the number of timeouts by 7%.

6 Conclusions

We have evaluated a caching scheme for MANETs that includes the implementation of a local cache in each mobile device in the ad hoc network. With this scheme, the mobile nodes also have the ability to intercept the requests from other nodes. By using cross-layer information the routing protocol can also piggyback the requests information in the routing protocol messages, so that other nodes can respond with the location of the document when the route to the data source is being created.

By means of simulations, we have studied the influence in performance of the cross-layer interception caching technique of the mean time between requests (the network load), the mean document lifetime, the requests pattern and the cache sizes. Taking into account the network load, the benefits of using the interception caching are more significant for highly loaded networks. Results show that as the network load decreases the local caches reduce their performance although the amount of AODV interception hits increases. In this sense, the TTL of the documents is a crucial parameter for the cache behaviour. As the TTL increases all the cache hits are incremented and hence there is a progressive reduction of the delay and number of timeouts. As expected, results are also proved to be very dependent on the popularity of the document. In particular, if the popularity of the documents is modelled by a Zipf distribution and the Zipf slope is close to 1.0, the cache hits increase because fewer documents are requested more often provoking more cache hits. Finally, the cache size also influences the delay and timeouts but for cache sizes greater than 5% of the document, the performance is not improved.

We can conclude that the interception of the requests when the route between the source node of the request and the server is already created obtains a reduction of the delay and timeouts for all the studied parameters. On the other hand, AODV interception, specially designed when there is not an active route between the source of the requests and the server, in conjunction with the previous interception techniques reduces even more the delay and the occurrence of timeouts. As a consequence of the cache hits the server load is widely reduced. The reduction of the load in the MANET is expected to improve its performance.

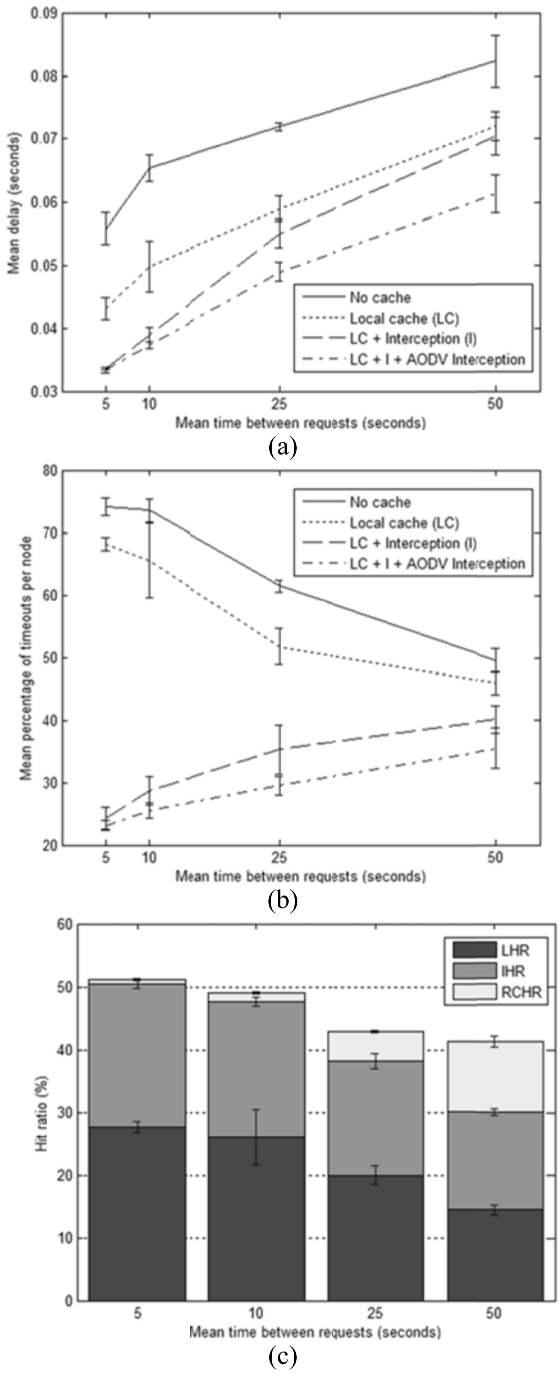
As a future work we propose to study the influence of the node speed as well as the mobility pattern. Another parameter to study is the node density in the network.

Acknowledgments. We would like to thank Adela Isabel Fernandez Anta for revising the syntax and grammar of this article. This study was partially supported by the National Project No. TEC2009-13763-C02-01.

References

1. Wakikawa, R., Malinen, J.T., Perkins, C.E., Nilsson, A., Tuominen, A.J.: Global Connectivity for IPv6 Mobile Ad Hoc Networks. Internet Draft, Internet Engineering Task Force (2006)

2. Dodero G., Gianuzzi, V.: Saving Energy and Reducing Latency in MANET File Access. In: 26th International Conference on Distributed Computing Systems Workshops (ICDCSW'06), pp. 16--20 (2006)
3. Lim, S., Lee, W.C., Cao, G., Das, C.R.: A novel caching scheme for improving Internet-based mobile ad hoc networks performance. Ad Hoc Networks, vol. 4, no. 2, pp. 225--239 (2006)
4. Moriya, T., Aida, H.: Cache Data Access System in Ad Hoc Networks. In: 57th IEEE Semiannual Vehicular Technology Conference (VTC 2003), vol. 2, pp. 1228--1232 (2006)
5. Tang, B., Gupta, H., Das, S.R.: Benefit-Based Data Caching in Ad Hoc Networks. IEEE Transactions on Mobile Computing, vol. 7, no. 3, pp. 289--304 (2008)
6. Wang, Y.H., Chen, J., Chao, C.F., Chuang, C.C.: A Distributed Data Caching Framework for Mobile Ad Hoc Networks. In: 2006 International conference on Wireless communications and mobile computing, pp. 1357--1362 (2006)
7. Cho, J., Oh, S., Kim, J., Lee, K.H., Lee, J.: Neighbor Caching in Multi-Hop Wireless Ad Hoc Networks. IEEE Communications Letters, vol. 7, no. 11, pp. 525--527 (2003)
8. Yin, L., Cao, G.: Supporting Cooperative Caching in Ad Hoc Networks. IEEE Transaction on Mobile Computing, vol. 5, no. 1, pp. 77--89 (2006)
9. Ting Y., Chang, Y.: A Novel Cooperative Caching Scheme for Wireless Ad Hoc Networks: GroupCaching. In: International Conference on Networking, Architecture and Storage (NAS 2007), pp. 62--68 (2007)
10. Chand, N. Joshi, R.C., Misra, M.: Cooperative Caching in Mobile Ad Hoc Networks Based on Clusters. International Journal on Wireless Personal Communications, no. 43, pp. 41--63 (2007)
11. Denko, M.K.: Cooperative Data Caching and Prefetching in Wireless Ad Hoc Networks, International Journal of Business Data Communications and Networking, vol. 3, no. 1, , pp. 1--15 (2007)
12. Du, Y., Gupta, S.: COOP – A cooperative caching service in MANETs. In: Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services (ICAS-ICNS 2005), pp. 58--63 (2005)
13. Chiu, G., Young, C.: Exploiting In-Zone Broadcast for Cache Sharing in Mobile Ad Hoc Networks. IEEE Transactions on Mobile Computing, vol. 8, no. 3, pp 384--397 (2009)
14. Artail, H., Safa, H., Mershad, K., Abou-Atme, Z., Sulieman, N.: COACS: A Cooperative and Adaptive Caching Systems for MANETs. IEEE Transactions on Mobile Computing, vol. 7, no. 8, pp. 961--977 (2008)
15. Chow, C.Y., Leong, H.V., Chan, A.: Group-based Cooperative Cache Management for Mobile Clients in a Mobile Environment. In: 33rd International Conference on Parallel Processing (ICPP'04), pp. 83--90 (2004)
16. Hosseini-Khayat, S.: Optimal solution of off-line and on-line generalized caching. Technical Report WUCS-96-20 (1996)
17. Perkins, C. E., Belding-Royer, E. M., Das, S.: Ad Hoc On Demand Distance Vector (AODV) Routing. IETF RFC 3561 (2003)
18. Bloom, B., Space/time Trade-offs in Hash Coding with Allowable Errors, ACM Communications, vol. 13, no. 7, pp. 422--426 (1970)
19. NS-2 Home page: http://isi.edu/nsnam/ns/
20. Kurkowski, S., Camp, T., Colagrosso, M.: MANET Simulation Studies: The Incredibles. ACM's Mobile Computing and Communications Review, vol. 9, no. 4, pp. 50--61 (2005)
21. Broch, J., Maltz D., Johnson, D., Hu, Y., Jetcheva, J.: Multi-Hop wireless ad hoc network routing protocols. In: ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM '98), pp. 85--97 (1998)
22. Adamic, L.A., Huberman, B.A.: Zipf's law and the Internet. Glottometrics, vol. 3, pp. 143--150 (2002)

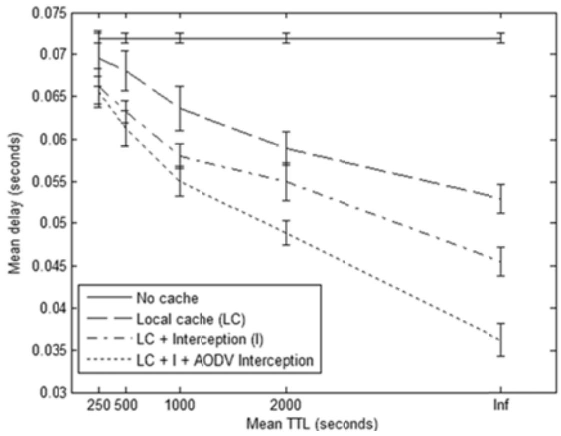


(a)

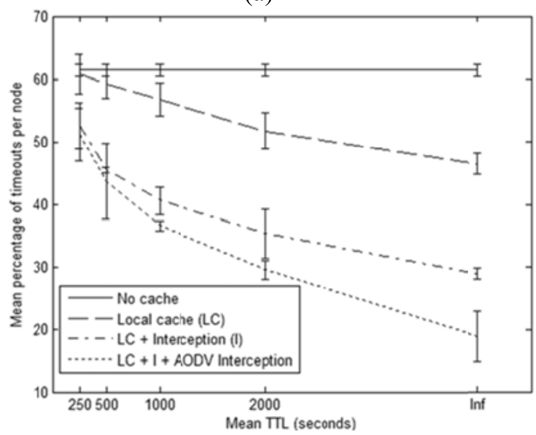
(b)

(c)

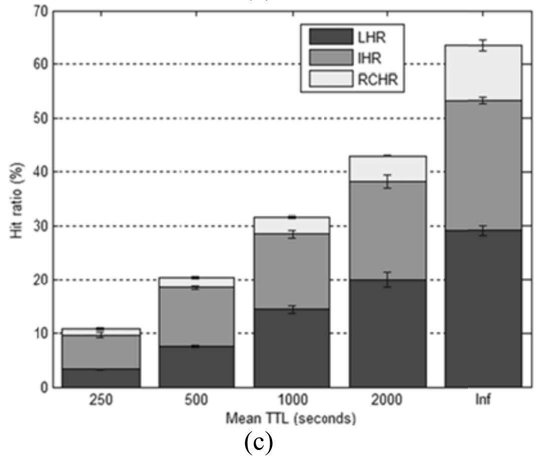
Fig. 4. Delay (a), percentage of timeouts (b) and cache hits (c) as a function of the mean time between requests.



(a)



(b)



(c)

Fig. 5. Delay (a), percentage of timeouts (b) and cache hits (c) as a function of the mean TTL of the documents.

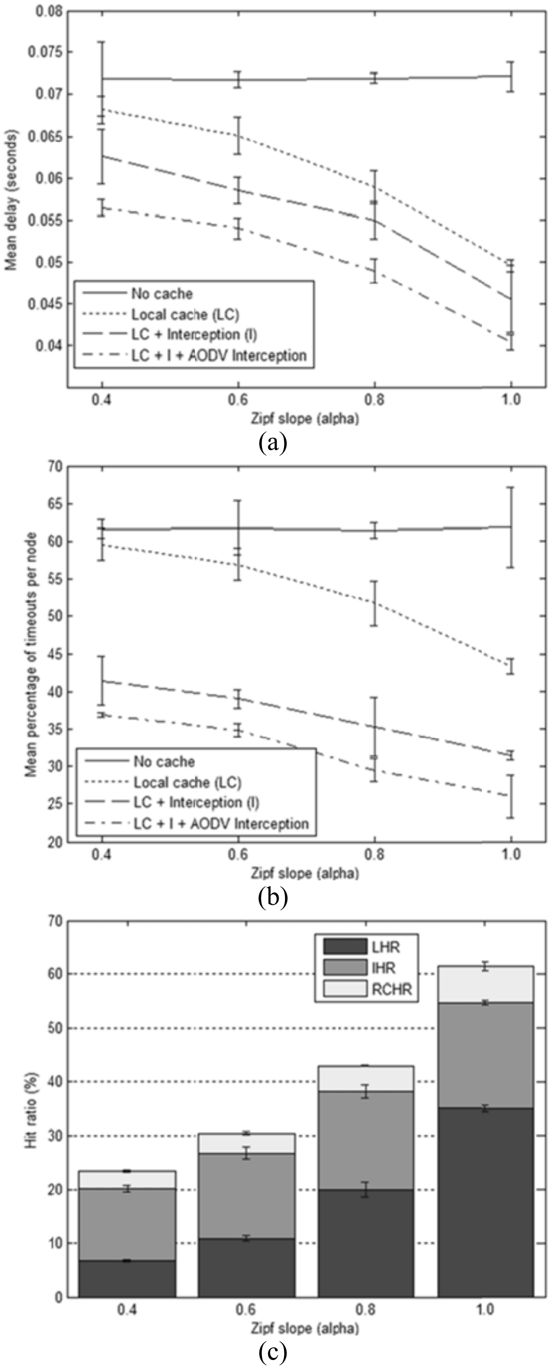


Fig. 6. Delay (a), percentage of timeouts (b) and cache hits (c) as a function of the Zipf slope.

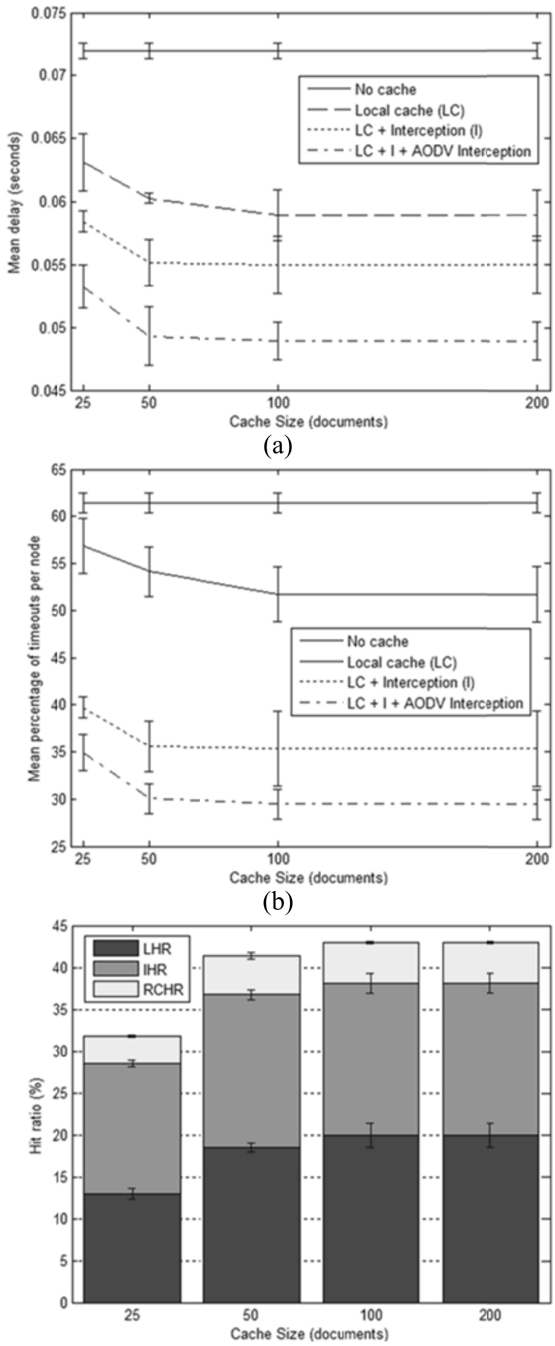


Fig. 7. Delay (a), percentage of timeouts (b) and cache hits (c) as a function of the cache size.