Proposal and Evaluation of an Application Level Caching Scheme for Ad Hoc Networks

F.J. González-Cañete, E. Casilari, A. Triviño-Cabrera
Dpto. Tecnología Electrónica
University of Málaga
Málaga, Spain
+34 952 13 71 76
{fgc, ecasilari, atc}@uma.es

ABSTRACT
In this paper, we describe a caching scheme for ad hoc networks. In this proposal the wireless nodes implement a local cache so that consecutive requests to the same document can be served directly by their own local cache instead of accessing the remote server. The nodes can also intercept the forwarding requests and serve the documents requested directly using their local cache. On the other hand the nodes learn where the documents are located using the information of the request and response messages that they forward. Using this information the nodes can redirect the requests to other nodes that are known to have the document requested and that are closer than the original destination of the request. By means of simulations we demonstrate that these proposals reduce the latency perceived by the nodes.

Categories and Subject Descriptors
I.6.5 [***Simulation and Modeling***]: Model development C.2.1 [***Computer Communication Networks***]: Network architecture and design

General Terms
Management, Measurement, Performance.

Keywords
Caching, replacement policy, ad hoc networks, performance.

1. INTRODUCTION
Mobile Ad Hoc Networks (MANETs) are composed of wireless devices that intercommunicate without any infrastructure. To get this goal, nodes collaborate retransmitting and routing the packets of others nodes so that they can reach their final destination. Due to their autonomy, MANETs were initially conceived for disaster or battlefield operations where the deployed telecommunication network may not be available.

However, the success of wireless communications has prompted the use of MANETs in commercial applications such as conferences or vehicular ad hoc networks. In these new scenarios, the users demand access to external networks, especially to the Internet.

Concerning the connection to the Internet, some web technologies must be adapted to the specific features of mobile ad hoc networks. Among them, the following characteristics outstand:

- Limited Processing Capabilities. Some light weight devices are constrained in their processing and visual capabilities.
- Limited Batteries. Mobile devices operate with batteries. In order to maximize their lifetimes, the messages that they generate should be reduced.
- Scarce Bandwidth. Wireless Medium has restricted bandwidth so signalling traffic should be minimized in MANETs.
- Temporary Connection to the Internet. The integration of MANET into external networks is guaranteed through Internet Gateways. However, the mobility of the MANET may provoke the Gateway to be temporarily unavailable. Under these circumstances, the HTTP Server is not reachable and web technologies must adapt to this condition.

Taking into account these characteristics, HTTP traffic could benefit from web caching. When using web caching, devices store in their internal cache some documents which were previously requested to an HTTP server. Mobile devices in a MANET can take advantage of the storage space of other nodes, so that the documents can be served without accessing the HTTP server. With this operation, HTTP requests are satisfied even when the Internet Gateway is not reachable which may happen in real MANET applications. Furthermore, the traffic generated to get the document is reduced as an intermediate node in the route to the server serves it.

This paper studies how the web caching approach can work in multihop wireless networks. Furthermore, as explained in this paper, the adaptation must cope with collisions in the wireless medium. The rest of this paper is structured as follows. In Section 2, the caching scheme is described. Section 3 details the simulation model and the results of the simulations. Finally, Section 4 outlines the main conclusion and suggests possible future work.

IWCMC'09, June 21–24, 2009, Leipzig, Germany.

2. CACHING SCHEME

In this section we present an application level caching scheme for ad hoc networks. In this scheme the network nodes request documents that are located in data servers. Due to the limited capabilities of wireless devices, we assume that the data servers are not part of the MANET but they are accessed through the Internet Gateways. The Internet Gateways are fixed nodes in the MANET as specified in [1]. The Web caching procedure works as follows: a node requests a document to a data server; the request is routed through the ad hoc network using the routing algorithm defined for this network. When the data server receives this request it responds sending the document to the node. This client-server scheme is very similar to the one used in the Internet for the HTTP Web traffic. To improve this procedure we compare the following three caching schemes.

2.1 Local caching

As in the case of HTTP traffic, the simplest way in which a caching scheme can be implemented is allocating a local cache for each user of the network, that is, for each MANET node. This cache will store the documents requested by the node once they are received from the data server. The next time the node requires the same document it will be served directly from its local cache. This situation is called a cache hit and it drastically reduces the traffic over the ad hoc network, as well as the energy consumption and the latency to receive the documents.

The local cache requires to define some parameters: the cache size, the replacement policy and the expiration of the documents. In ad hoc networks the characteristics and capabilities of the nodes can be very heterogeneous, thus the storage space that can be reserved for the local cache will vary depending of each node. The replacement policy is the algorithm that decides which document or documents that are stored in the cache have to be evicted to make room for the new one that has just arrived to the node. Many replacement policies have been proposed for HTTP traffic at different levels [2]: user, proxy and server cache. Those algorithms have been designed to take advantage of the characteristics of the HTTP traffic over the Internet and hence they can be adapted or adopted in the ad hoc networks if the traffic shares some of those characteristics. Finally, the cache must also take into account the expiration time of the documents. A document is considered to be valid during a period of time defined by its lifetime. The expired documents can be evicted from the cache because the information that they contain is considered to be obsolete. Thus, if needed, the node will have to request the document again to the data server.

2.2 Interception caching

In HTTP traffic, the proxy cache is a component of the network situated between the users and the Web servers [3]. This proxy cache is located close to a group of users with similar characteristics, for example the employers of a company or a university and hence they share analogous interests, so they usually request similar documents. When a user requests a document to a server, the document is stored in the proxy cache and if another user requests the same document, it will be directly served from the proxy instead of the remote server. This interception of the requests reduces the latency perceived by the users because the document is served from a proxy which is closer than the server. The proxy interception also decreases the Internet traffic in the outgoing link and the load of the remote servers.

Since proxies are not available in MANETs, this interception proxy caching scheme must be adapted to the ad hoc environment. To satisfy this demand, the proxy functionalities are transferred to the MANET nodes in a similar way to the routing procedures. With this additional task, every node in the path of a request from a node to the data server can respond to this request if it has a valid copy of the requested document in its own local cache.

Figure 1 shows an example of an ad hoc network where DS is 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.

In the case that node 2 requests a document A, the request will pass through node 1 to DS using an ad hoc routing algorithm. The data server will respond with the document using the path from node 1 to 2. Finally the node 2 will store the document A in its local cache. If node 3 requests the same document A to the DS the request will reach node 2 that checks if there is a valid copy of the document A in its local cache and, if so, it will respond to node 3 with the document. This interception of the request reduces the number of hops from 6 (3-2-1-DS-1-2-3) if there is not an interception, to 2 (3-2-3) and consequently the latency perceived by node 3. This mechanism also saves energy in the nodes and reduces the traffic because terminals do not have to forward the requests and responses. In addition, the interception reduces the possible bottlenecks in the data servers because they do not process all the generated requests.

The situation when a node in the path of a request to the DS intercepts the request is called an interception hit.

2.3 Redirection caching

Aiming at reducing the route length to the serving node, the previous scheme is extended to take into account some information about the distribution of the documents in the MANET. Specifically, we make nodes keep information about the distance (measured as the number of hops) where the served documents can be found. This information is dynamically extracted from the forwarded messages (requests and responses). From these data, each node can know that a document is stored in the local cache of a node that is closer than the data server.

To illustrate this procedure, let us suppose that node 4 in Figure 1 requests the document A to DS. The request will pass through nodes 2 and 1 to DS so node 2 knows that node 4 will have the document A and that node 4 is one hop away. Similarly, the node 1 will know that node 4 will own the document A and that node 4 is two hops away. When the DS responds with the document A through nodes 1 and 2 to node 4, nodes 1 and 2 will register that

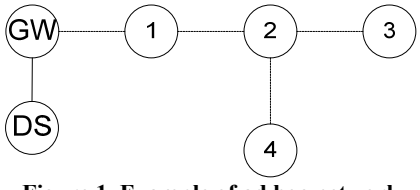

Figure 1. Example of ad hoc network

DS has the document A and it is one and two hops away respectively. If node 3 requests the same document A to DS, the request will reach node 2. In that situation node 2 knows that node 4 and DS have the document A and they are one and two hops away respectively so node 2 will redirect the request to node 4 because the path to the document is shorter. This redirection of the request reduces the number of hops from 6 (3-2-1-DS-1-2-3) if there is not redirection, to 4 (3-2-4-2-3). Hence the latency perceived by node 3 is diminished.

The previous policy takes into account the redirection in the path of the request, but it also can be considered the direct redirection at the source of the request. Let us suppose that in situation described in the previous example node 2 requests the document A. Instead of sending the request to DS node 2 can decide to request it to node 4 directly because it knows that node 4 is one hop away while DS is 2 hops away.

Unfortunately, the redirection caching has some drawbacks that have to be considered: the mobility and disconnection of nodes and the replacement of the documents.

The mobility of the nodes in the ad hoc networks causes links between nodes to break so that the information kept in the table becomes obsolete. This may turn into redirections do not reach the destination. This circumstance also occurs if the nodes are disconnected from the network because they leave the coverage area or because the device is turned off. A timeout system must be implemented in the request node to notice this situation in order to request the document again.

The other problem of the redirection consists in the replacement of the documents in the local cache of the nodes. In the previous example node 2 knows that node 4 has the document A, but node 4 could have evicted the document from its local cache because of the replacement policy. At that point if node 3 requests the document A and node 2 decides to redirect the request to node 4, the document could not be served because node 4 does not have the document at the present moment. This situation is called a cache redirection miss. A solution to this problem is that node 4 sends a special response message informing node 3 that it does not have the document. Making so, every node in the path from node 4 to node 3 could become aware that node 4 does not have the document A and consequently they could update their knowledge about the cache in node 4. Finally, node 3 will request the document A again to DS and node 2 will not redirect this request. The interception misses decrease the performance of the network because they generate more traffic and the latency of the response. In the previous example the number of hops is 10 (3-2-4-2-3-2-1-DS-1-2-3) for the interception miss instead of the situation without interception that requires 6 hops (3-2-1-DS-1-2-3).

In order to decrease the number of redirection misses caused by the replacement of the documents stored in the local caches an expiration for the knowledge of the document situation is proposed. Each node calculates the mean time that the documents are stored in its local cache. Supposing that the cache space of the other nodes is the same the mean time the documents will be stored in the other caches will be similar to the calculated locally. In that way, when the redirection information of a document is stored the expiration time of this information will be the minimum between the Time To Live (TTL) of the document and the mean time the documents are stored in the local cache.

The learning method proposed can also be expanded if we consider the promiscuous mode of the nodes in the ad hoc network. Under this mode, the nodes could learn about the situation of the documents not only taking into account the request and responses that they receive and forward but also listening to any request or response forwarded by their neighbour node.

3. Performance evaluation

In this section the simulation model and the performance evaluation of a static multihop ad hoc network are presented. In this work we study the performance of the caching scheme presented in the previous section taking into account only the application level traffic. In that way, although we have used the AODV routing protocol [4], we suppose that the ad hoc routing protocol is configured to the mobility conditions, especially the Active Route timeout parameter. Thus, we assume that routes created by AODV do not expire. As the routes are created only once (as nodes are static), the AODV traffic will be insignificant compared with the application level traffic.

3.1 Simulation model

The simulations are based on the network simulator NS-2.33 [5] that is the most popular simulator for the researches on ad hoc networks [6]. The area where the nodes are located is 1000 metres x 1000 metres. We study three scenarios which nodes are uniformly distributed forming grid of three different densities: 5x5, 7x7 and 9x9 nodes. As the radio range of the nodes is 250 meters the connectivity between the nodes is different in each mesh. Figure 2 shows the connectivity (neighbour nodes) for the above mentioned distributions. As can be observed, when the density increases, the amount of available nodes at one hop also increases.

There exist 1000 different documents (identified by a specific number) distributed between two servers that are located in opposite corners of the simulation area. 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 odd identification are situated in a server and even documents are located in the other one. In addition, each document has an associate TTL time that determines when the document expires and hence it is considered to be 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 mean between 500 and 5000 seconds for the TTL of the documents. In

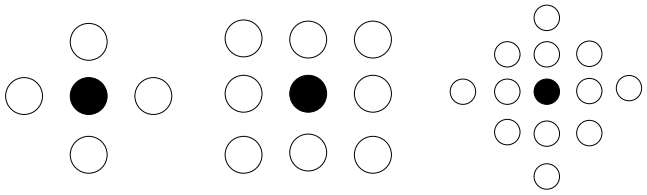

Figure 2. Connectivity for 5x5, 7x7 and 9x9 nodes grids

that way we model both a high variability and a low variability of the documents. In addition we also consider an infinite TTL for the documents, that is, the documents never expire. The size of all documents is constant and equal to 1000 bytes.

Each node that is not a server generates 10000 requests to the servers along the simulation time. When a request is served another request is generated by the same node. The waiting time between the reception of a response and the next request follows an exponential distribution with a mean between 5 and 100 seconds. Using this range of values we evaluate a wide range of the node activity (and consequently the networks load). A document is requested again if the response of the actual request is not served before a defined timeout is triggered.

The pattern of requests of the documents follows a Zipf like distribution as this law governs the request of documents in the Internet [8]. 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. 1.

$$P(i) = \frac{\beta}{i^{\alpha}} \qquad \text{with } \alpha \text{ close to 1} \qquad \text{Eq. 1}$$

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) and the β parameter is the displacement of the function. In out simulations, the slopes selected to generate the requests are 0.4, 0.6, 0.8 and 1.0.

Finally, each node implements a local cache that employs the Least Recently Used (LRU) replacement policy [8]. 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 cache misses because the cache is empty, the local caches are "warmed up" using the first 20% of the requests.

As the amount of requests of each node is constant and the mean time between requests varies and the 20% of the requests are used to "warm-up" the cache, the simulation time also varies between 40000 and 800000 seconds. Table 1 summarizes the main simulation parameters.

3.2 Performance evaluation

Each simulation has been executed five times using the same TTL, time between requests and requests distribution. The performance evaluation presented is the mean of the results obtained for the five simulations.

Table 1. Simulation parameters

Parameter	Default	Values
Simulation area	1000x1000	
Number of nodes		5x5 – 7x7 – 9x9
#Documents	1000	
#Requests per node	10000	
TTL (s)	2000	500-1000-2000-4000-5000-Inf
Mean time between queries (s)	10	5-10-50-100
Zipf slope	0.8	0.4-0.6-0.8-1.0
Replacement policy	LRU	
Cache size (documents)	100	25-50-100-200
Warm-up (requests)	2000	

We use the following metrics to quantify the performance of the network:

- Delay: It is defined as the time elapsed between the request of a document and the reception of the response.
- Hit Ratio: It is defined as the number of documents served by the cache in the network nodes divided by the total number of requests.

At this point we distinguish a type of hit ratio for each functionality studied: the local hit ratio, the interception hit ratio and the redirection hit ratio. At each node, the local hit ratio defines the proportion of documents served by the local cache, the interception hit ratio is the proportion of the documents requested by a node that have been served by a node in the path to the server, and the redirection hit ratio is the proportion of documents requested by a node that have been served by another node after a redirection of the request.

We will compare the performance of an ad hoc network in four situations: 1) the nodes do not implement any cache mechanism (No cache), 2) the nodes only use the local cache (LC - Local cache), 3) the intermediate nodes can intercept the requests (I – Interception) and 4) the nodes implement the redirection of requests (Redirection) as well as the interception.

Although the simulations were performed using the 5x5, 7x7 and 9x9 grids, the figures shown correspond to the 5x5 nodes network because the results obtained by the 7x7 nodes and 9x9 nodes are similar.

Figure 3 represents the delay and hit ratio as a functions of the mean time between requests. The use of local caching drastically reduces the delay perceived by the nodes especially when time between requests is short. As the time is increased this difference is decreased because of the expiration of the documents in the local caches. This fact causes the reduction of the local cache hit ratio and hence the amount of documents that have to be requested again to the server is increased. The Interception scheme outperforms the local caching and also reduces the delay. Finally, the Redirection slightly outperforms the Interception although the improvement with a high request rate it is not very significant. For low loaded networks (a low request rate) the reduction of the delay if the combined caching scheme is adopted is about 18% compared with the schema without caching, while the reduction can reach the 43% for very loaded networks (with a high request rate). In very loaded networks with very active nodes (mean waiting time of 5 or 10 seconds) the amount of requests served by the local cache or another intermediate cache is about 55%. This fact reduces drastically the traffic in the servers distributing the load among the nodes.

Figure 4 shows how the mean TTL of the documents influences on the delay and the hit ratio. As the TTL of the documents increases the time they can be stored in the caches increases and hence they can be useful during more time because they expire later. As it can be observed from the figure the delay is reduced as the TTL increases. For TTLs up to 2000 seconds the delay is slowly reduced asymptotically until the optimal value where the TTL is infinite, that is, the documents do not expire. In the case of infinite TTL the percentage of hits reaches near 60%. This fact causes the reduction of the delay in about 50% compared to the

scheme without caching. In the case of short living documents (low TTL) the reduction of the delay is about 25%.

Figure 5 compares the delay and hit ratio as we change the slope of the Zipf distribution of the request pattern. As the slope increases the local cache hit also is increased due to the fact the most popular documents are frequently requested. On the other hand the interception hits are decreased as the slope increases because most traffic is served by the local caches. Consequently the delay is widely reduced as the Zipf slope increases following a similar behaviour as the previous studies. The delay is reduced about a 25% and a 50% for the 0.4 and 1.0 slope respectively.

Finally the Figure 6 shows the delay and hit ratio as a function of the cache size. The performance improvement reaches to the limit when the cache size is 100 Kbytes (100 documents) as the results obtained for the 200 Kbytes cache size (200 documents) are similar. For a cache of 25 documents the improvement obtained with the Redirection caching is close to zero but the redirection hit ratio increases as the cache size increases. For the 25 documents cache the reduction of the delay is about 25% meanwhile for largest caches it is reduced a 40%.

4. Conclusions

In this work we have proposed an application level caching scheme for ad hoc networks. The scheme suggests to implement a local cache at each node of the ad hoc network in order to intercept or redirect the requests by the intermediate nodes from the source to the destination request. In that way the number of hops is reduced, and hence, the delay perceived by the ad hoc nodes. As the number of hops is decreased, the number of forwarding messages also decreases and the power consumption is reduced. We have studied by mean of simulations the influence of the mean time between queries (which defines the request rate), the effect of the TTL of the documents, the influence of the traffic pattern and the size of the caches.

We can conclude that the use of local caching combined with the use of the interception and redirection caching reduces drastically the delay perceived by the nodes. The reduction of the delay can vary between 18% and 50% depending on the characteristics of the traffic and cache size.

As a future work we propose to refine the replacement policy of the cache in order to take into account parameters such as the TTL, the frequency of the requests and the number of hops from node that served the document. The study can also be expanded using another routing protocols such as the Optimized Link State Routing protocol (OLSR) [9] that is more appropriate for static wireless networks. Finally, we also propose to extend the study taking into account the mobility of the nodes.

5. ACKNOWLEDGMENTS

We would like to thank Adela Isabel Fernández Anta for revising the syntax and grammar of this paper.
This work was partially supported by the public Project TEC2006-12211-C02-01.

6. REFERENCES

[1] Wakikawa, R., Malinen, J.T., Perkins, C.E., Nilsson, A., Tuominen, A.J. 2006. Global Connectivity for IPv6 Mobile Ad Hoc Networks, draft-wakikawa-manet-globalv6-05.txt, Internet Draft, Internet Engineering Task Force

[2] Khayari, R.A. 2003. Workload-Driven Design and Evaluation of Web-Based Systems, Thesis, Der Andere Verlag, Osnabrueck, Germany.

[3] Luotonen, A. and Altis, K. 1994. World-Wide Web Proxies. *Computer Networks and ISDN Systems,* Vol. 27, No. 4, pp. 147-154.

[4] Perkins, C. E., Belding-Royer, E. M., and Das, S. 2003. Ad Hoc On Demand Distance Vector (AODV) Routing. IETF RFC 3561.

[5] NS-2 Home page: http://isi.edu/nsnam/ns/

[6] Kurkowski, S., Camp, T., Colagrosso, M. 2005. MANET Simulation Studies: The Incredibles. ACM's Mobile Computing and Communications Review, vol. 9, no. 4, pp. 50-61.

[7] Adamic, L.A., Huberman, B.A. 2002. Zipf's law and the Internet. Glottometrics, vol. 3, pp. 143-150.

[8] Coffman, E.G., Dennings, E.J. 1973. Operating Systems Theory. Prentice-Hall.

[9] Clausen, T., Jacket, P. 2003. Optimized Link State Routing protocol (OLSR), IETF RFC 3626.

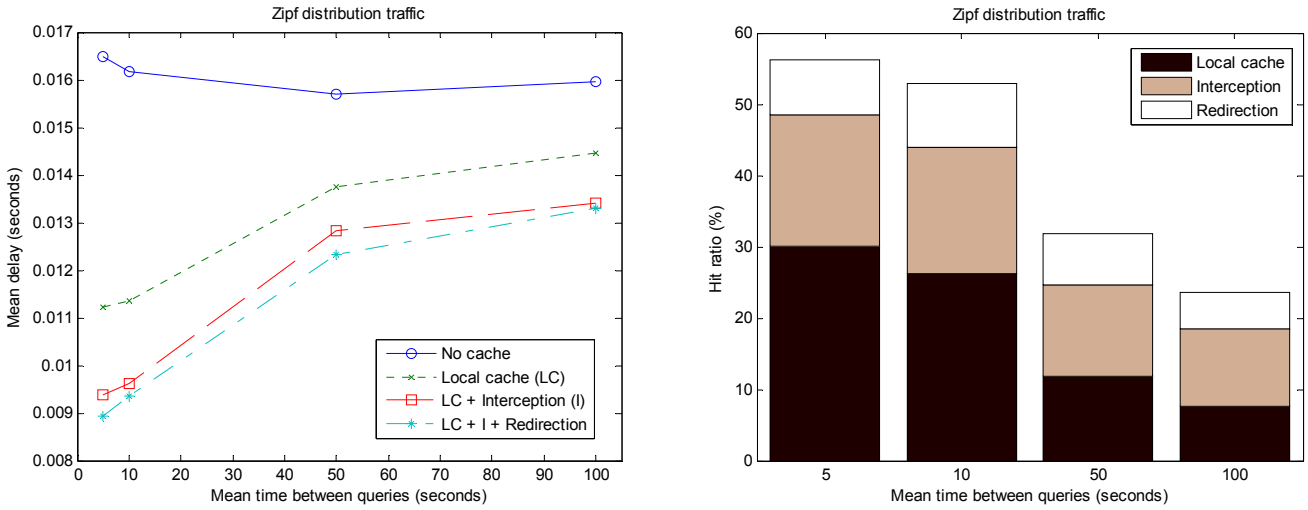

Figure 3. Delay (left) and Hit ratio (right) as a function of the mean time between queries

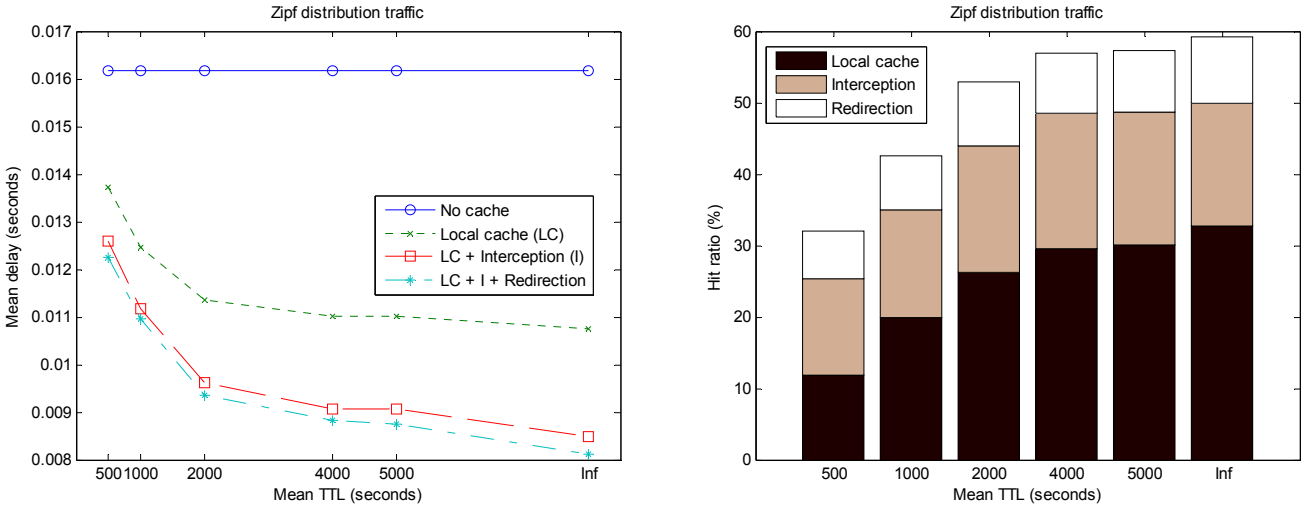

Figure 4. Delay (left) and Hit ratio (right) as a function of the mean TTL of the documents

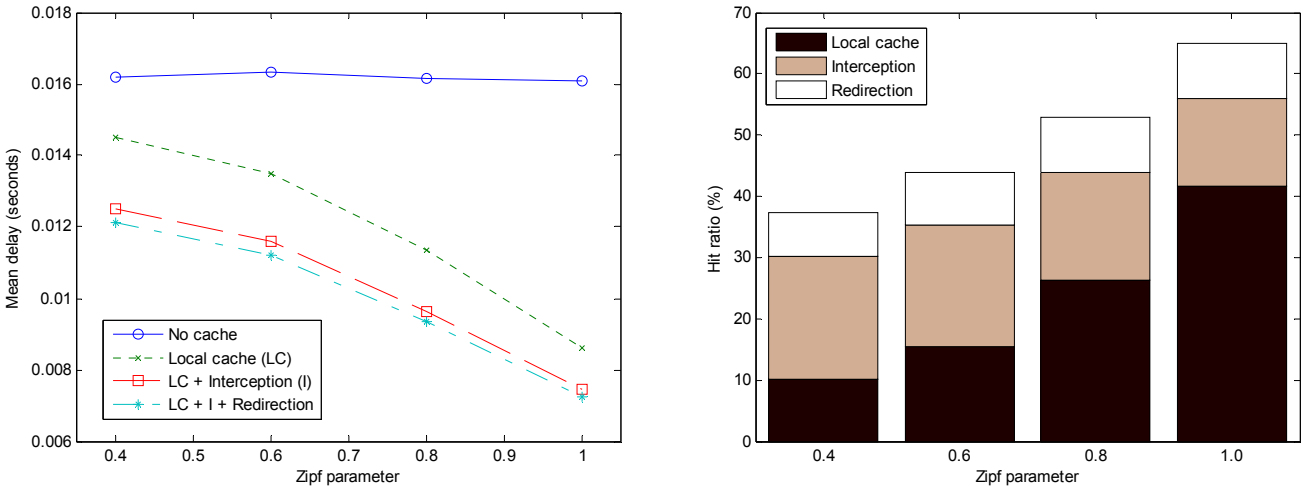

Figure 5. Delay (left) and Hit ratio (right) as a function of the Zipf parameter

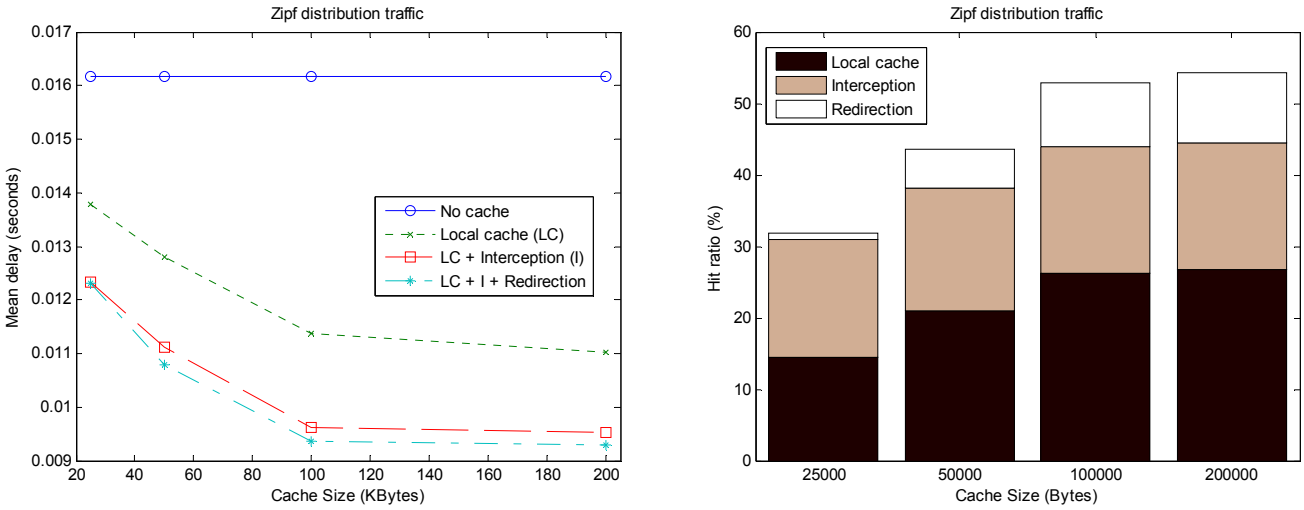

Figure 6. Delay (left) and Hit ratio (right) as a function of the cache size