Wireless Pers Commun
DOI 10.1007/s11277-017-4984-7

Improving Cache Effectiveness Based on Cooperative Cache Management in MANETs

Ali Larbi[1] · Louiza Bouallouche-Medjkoune[1] · Djamil Aissani[1]

Abstract In wireless mobile Ad Hoc networks, cooperative cache management is considered as an efficient technique to increase data availability and improve access latency. This technique is based on coordination and sharing of cached data between nodes belonging to the same area. In this paper, we studied the cooperative cache management strategies. This has enabled us to propose a collaborative cache management scheme for mobile Ad Hoc networks, based on service cache providers (SCP), called cooperative caching based on service providers (CCSP). The proposed scheme enabled the election of some SCPs mobile nodes, which receive cache's summaries of neighboring nodes. Thus, nodes belonging to the same zone can locate easily cached documents of that area. The election mechanism used in this approach is executed periodically to ensure load balancing. We further provided an evaluation of the proposed solution, in terms of request hit rate, byte hit rate and time gains. Compared with other caching management schemes, the simulation results show that the proposed CCSP scheme improves significantly the cache effectiveness and the network performances. This is achieved by improving data availability and reducing both overall network load and latencies perceived by end users.

Keywords Mobile Ad Hoc networks · Data service management · Cooperative caching · Service cache provider · Cache management · Simulation

✉ Ali Larbi
larbiali2@mail.com

Louiza Bouallouche-Medjkoune
Louiza_medjkoune@yahoo.fr

Djamil Aissani
lamos_bejaia@hotmail.com

[1] Research Unit LaMOS, "Modeling and Optimization of Systems", Faculty of Exact Sciences, Bejaia University, Targa Ouzemmour, 06000 Bejaïa, Algeria

Published online: 07 October 2017

1 Introduction

The last decade has seen a rapid growth in computer and wireless communication technologies, which stimulated the development of mobile communication systems. Consequently, two categories of mobile networks have been developed: (i) infrastructure based and (ii) Ad Hoc networks [1]. The infrastructure based network uses fixed network access points (APs or base stations) with which mobile terminals interact for communicating. APs are responsible for receiving the signal from end devices and retransmitting them to the destinations. Its role is also to interconnect the wireless LAN to external networks such as internet. However, these networks are frequently unsuitable for many applications and environments (hostile environment, need for a rapid deployment,...), thus, the deployment of fixed infrastructure is either too expensive, or impossible. Due to their flexibility and ease of deployment, distributed solutions, known as mobile Ad Hoc NETworks (MANETs), are potentially more adapted for applications such as battlefield, disaster recovery environments or outdoor assemblies. MANET must be seen as complimentary of infrastructure-based network rather than competitive [2, 3]. Indeed, in infrastructure based, the network coverage can be extended by Ad Hoc networks.

The posed constraints in the particular context of the mobile Ad Hoc networks are different, and especially more complex than those encountered in the wired architecture [4, 5]. These networks may suffer from scarce bandwidth, frequent network disconnections and limited local resources like memory's capacity. However, with the spectacular progress of microelectronics, the storage capacities of memories and the data transfer rates have significantly improved. Still, this asset remains insufficient in comparison with the tendency to increase the number of end users and exchanged information volume between them. This new deal will reduce increasingly the level of quality-of-service required by mobile end users, particularly their perceived latencies. The use of cooperative caching is proved to be efficient solution to improve latencies perceived by users and to increase overall storage capacity of caches [1, 6–10]. The cache refers to the storage mechanism of usually or recently required information, for a future re-use [5, 9]. The basic idea is due to the fact of storing the data items in the local cache of the mobile nodes; the mobile Ad Hoc network members can effectively reach required information without request servers. This technique has been widely studied mainly to minimize average latency perceived by the end users, to save bandwidth and to reduce energy consumption [6, 11]. However, in mobile Ad Hoc networks, nodes belonging to the same geographic area may have similar tasks and share common interest, thus cache replacement strategies that perform replacement individually cannot be suitable [12, 13]. Cooperative caching based on the sharing and coordination of cached data among multiple clients, allows the cache systems to be optimized, and the use of all network resources can be more effective. There are a very great number of cache management mechanisms in literature and most of them were proposed in the field of Web. Cache management techniques designed for wired networks may not be applicable to wireless Ad Hoc networks. In this paper, a new approach was established for a cooperative cache management in mobile Ad Hoc networks based on service cache providers (SCP). This new approach allows the election of some mobile nodes to be SCPs that receive summaries caches of neighboring nodes. Consequently, nodes belonging to the same area can reach all requested documents, which are locally cached by nodes of this zone. We evaluate the performance of our CCSP scheme by comparing it with CC scheme [14] (it is the "closer" competitor) and SimpleCache scheme (SC) [15]. We perform an experimental evaluation of these schemes and simulation results attests that the proposed CCSP scheme can significantly improve data accessibility and bandwidth saving.

The rest of this paper is organized as follows. Section 2 reviews some works related to cache replacement policies in MANETs and cooperative caching schemes. Section 3 exposes first the system model and the presentation of used local cache replacement policy. Then describes the proposed cooperative cache scheme. In Sect. 4, simulation model and summary of used notations are presented. Section 5 is devoted to the presentation and to the discussion of obtained simulation results. The concluding remarks are given in Sect. 6.

2 Existing Works

2.1 Cache Replacement Policies in MANETs

The cache replacement policies in MANETs can be either uncoordinated or coordinated, in case of uncoordinated cache replacement policy, the selection process of documents to delete from the cache is made based only on document's characteristics and local cache state. However, in case of coordinated cache replacement policy, In addition to previous parameters, the selection process take into account all cache states of nodes belonging to the same managed zone. Thus, all local caches of this zone will be managed as a single large cache.

There is a very great number of cache replacement policies in literature, most of them were proposed in the field of Web. However, those policies do not take into account the characteristics of MANETs like mobility. Among the cache replacement policies, which were proposed within the framework of these networks, we can quote the TDS policy (time and distance sensitive) reported in [16]. This policy takes into account a distance factor and a temporal factor for the replacement. TDS calculates a value $v(i)$ for each document i, $v(i) = \delta_i * \tau_i$, where δ_i is the distance measured by the number of hops towards the access points or mobile terminals, having the requested data. $\tau_i = 1/(t_{cur} - t_{epdate_i})$, where t_{cur} and t_{epdate_i} are respectively the current time and the last update time of δ_i. TDS does not consider the frequency and the size of the documents, which are very important parameters and influence considerably on the caches performances. Authors in [17] proposed another cache replacement scheme, called MARS, mobility-aware replacement scheme. It takes into account various factors that are important when making cache replacement decisions. MARS uses a cost function for the replacement; it comprises a temporal score and a spatial score. When the cache of a mobile client becomes full and a new object needs to be cached, the cost function is used to generate a cost value for each cached object. The object with the lowest value is evicted from the client's cache and replaced by the new one. Dimokas et al. in [18] were also interested on wireless multimedia sensor networks (WMSNs), and they proposed a cache replacement policy, called NICC associated with a cache consistency mechanism.

2.2 Cooperative Caching

The cooperative caching is based on coordination and sharing of cached data by nodes belonging to the same area. In this technique, a number of caches work together to cache more objects collectively, thus, more client's requests are satisfied without overloading a single server. Cooperative caching in Ad Hoc networks has been widely studied in literature, particularly in the context of data accessibility issue [11, 15, 19, 20]. Among the cache management mechanisms that were proposed within the framework of Ad Hoc networks, we can quote the work of Du et al. in [11], they proposed a cooperative caching scheme for on-demand data access applications in MANETs called COOP. This scheme attempt to discover data sources which induce less communication overhead by utilizing cooperation zones, historical profiles, and hop-by-hop resolution. In [19], Sailhan and Issarny proposed a cooperative caching scheme to increase data accessibility by peer-to-peer (P2P) communication among mobile clients, when they are out of bound of a fixed infrastructure. Yin et al. in [15], were also interested on data access issue by using cooperative caching, they proposed three schemes: CacheData which caches the data, Cachepath which caches the data path and Hybridcache which associate the two schemes CacheData and Cachepath. In case of Ad Hoc networks with high mobility nodes, Cachepath or Hybridcache might be unsuitable, because of extra processing overhead which can be induced. Tang et al. in [20] consider the cache placement problem of minimizing total data access cost in Ad Hoc networks, they developed efficient strategies to select data items to cache at each node. The studies achieved by Fiore et al. in [21] attempt to consider all information of each zone of the MANET as a whole. They proposed a cooperative scheme called Hamlet integrating cache management strategy that considers both cases of nodes with large and small cache sizes. Their scheme allows nodes to decide whether to cache some documents and for how long. Chand et al. in [22] were interested on caches in MANETs and proposed cooperative scheme called zone cooperative (ZC). As its name indicates, this scheme is based on zone formed by nodes belonging to the neighborhood of a given client. In [16], authors proposed another cooperative caching strategy in mobile Ad Hoc networks based on clusters. Chan et al. in [6] studied Energy-saving issues and formulated the energy-efficient coordinated cache replacement problem (ECORP), it takes into account a coordinated replacement in the context of a mobile Ad Hoc network. Group caching (GC) is the technique proposed in [23]. It maintains localized caching status of 1-hop neighbors. When a data request is received in a mobile host (MH), it forms with its 1-hop neighbors a group by using the “Hello”? message. When caching placement and replacement need to be performed, the MH selects the appropriate group member to execute the caching task in the group. In this scheme, the data accessibility is increased but the energy consumption and constrain of wireless bandwidth are not much considered. In [24], group-based cooperative caching scheme called (GCC) is proposed. It is based on the concept of group caching, in which it is allowed for each mobile host and its k-hop neighbors to form a group. A directory of cached data items is maintained in each mobile host. Each MH obtains the directories of its k-hop group members through broadcasts in the group. In this technique, a consequent overhead may be generated by mobile host’s requests. The study made by Zhang et al. [10] is a continuity of previous works reported in [25, 26]. Throw these studies, they were interested on the cost efficiency of in-network caching in Long-Term Evolution by considering two cases, caching in data center only, and caching in both data center and in-network. They conclude that caching in all network mobile nodes significantly reduces traffic load and improves latency perceived by end-users. Chand et al. in [14] proposed cooperative caching scheme, called cluster cooperative (CC), based on geographic clusters. The Ad Hoc network is partitioned into non-overlapping geographic clusters. In this cooperative caching scheme, the clusterization process is achieved easily because the affiliation of each node to given cluster depends on its geographic position, however, in the situation of high mobility, an extra processing overhead can be induced. In the research work of Sureshkumar et al. [27], authors propose a mobility with clustering based energy efficient cooperative caching method (MCE2C2).

This method integrates clustering algorithm, cache consistency mechanism and cache replacement policy.

3 Cooperative Cache Management Scheme

In this section, we firstly expose the system model and the presentation of local cache replacement policy, then we describe the proposed cooperative cache scheme.

3.1 System Model

The system model consists of a set of mobile nodes that can communicate with each other using Ad Hoc communication protocols. The network topology is thus represented by an undirected graph $G = (E, V)$ where V is the set of mobile nodes $MN_1, MN_2, \ldots, MN_n$, $|V| = n$ and $E = V \times V$. E is the set of links between nodes. Nodes are free to move randomly, thus the network topology may change rapidly and unpredictably. Each node acts as a router and is willing to forward data of other nodes. The nodes outside the communication range of the access points can access the document in the access points through a number of hops using Ad Hoc network routing protocols. Thus, the document request is forwarded hop-by-hop until it reaches the access point. During its routing, the request may flows through many nodes, and it is possible that some of this nodes hold in their caches a copy of the requested document. In this case, the document is downloaded from the node that holds a copy of this document, without continuing the routing to access point.

3.2 Used Local Cache Replacement Policy

The design of the cache replacement policies intended for mobile Ad Hoc networks, or generally for mobile environments, is strongly influenced by the characteristics of Web traffic, as well as the specific characteristics of these networks. Among these characteristics, we can cite document frequencies and their sizes for Web traffic. We can also cite the distances variation that separates the nodes from/to each other's due to node's mobility.

In the following, we present the used cache replacement policy, FSDV Frequency, Size and Distance based Value. It is based on the calculation of a value for each document present in the cache. The calculation of this value includes four parameters, (i) access frequency, (ii) distance in a number of hops between requester node and that one which answers the request, (iii) time passed since the last update of this same distance and (iv) document size.

Thus, for each document i, we calculate the value H_i, as expressed in Eq. (1). Then, we choose document having smallest value for replacement.

$$H_i = \frac{f_i \times d_i}{s_i \times t_i} \tag{1}$$

where

f_i : is the access frequency to the document i;
d_i : is the hops number between requester node for a document i and that one which answers the request;
s_i : is the size of the document i;

t_i : is the time since the last update of s_i.

Upon receiving a new document, the cache decision is made locally, for each node of the network. According to remainder cache free space, if it is insufficient to cache the new document, cache replacement policy is invoked to liberate space, by selecting one or more cached documents for eviction. This eviction decision is also made locally.

3.3 Used Cache Consistency Mechanism

Cache consistency mechanisms ensure that each cached document copy is eventually updated, to reflect changes to the original document. There are several cache consistency mechanisms currently in use: time-to-live (TTL) fields, client polling and invalidation protocols. In our system model, we used TTL-based Cache consistency mechanism [28], however, the CCSP scheme can be associated with any other advanced cache consistency mechanism. With TTL approach, the origin server assign a TTL-value for each document. This value is an estimate of the cached document's lifetime, after this time value, the document is regarded as invalid, and the cached copy of document is deleted. Thus, the next request for this document will be satisfied by local cache of another node (through the CCSP scheme) or, in the worst case, it will be redirected to the origin server.

3.4 Cooperative Cache Scheme Description

The basis of cache's cooperation is to allow a node in situation of a local cache miss, to be able to reach this document from other cache nodes.

In our cooperative cache scheme, some nodes proportional to the total number of node in the network, are elected to play a role of cache services providers. This election is made according to an algorithm executed periodically to designate another group of SCP, in order to ensure a good load balancing.

When a given node is elected as SCP, it broadcasts invitation message to the neighboring at one or more hops, in order to request them to send the summary of their caches. The node receiving multiple invitation messages can reply at most, to three providers. After sending their respective summaries to SCPs, each node is required to validate its summary periodically, in all SCPs with which it is associated. If the provider sees that a given cache summary is still not validated after the end of third validation period, this summary will be deleted. Additionally, when a node realizes that he is out of communication range of all the SCPs that have copy of its cache summary, it broadcasts a request to join another SCP.

The behavior of the cooperative caching scheme based on service cache providers is shown in Fig. 1. When a node initiates request for document, it first looks for the data item in its own cache. If there is no copy of requested document in the local cache, the node sends lookup message to the nearest SCP, this one consults all summaries of local cache nodes that it holds. If there is an entry for the requested document, so this is a hit, it responds with the identity of the node having requested document. Otherwise, requester node contacts another SCP, if still no node that caches the requested document is found, this document will be recovered directly from the source.

Upon receiving requested document from SCP or from source, requester node checks if its cache free space is sufficient to cache this document, in which case it is inserted directly into the cache. Otherwise, the cache replacement policy is invoked to make space for the new document.

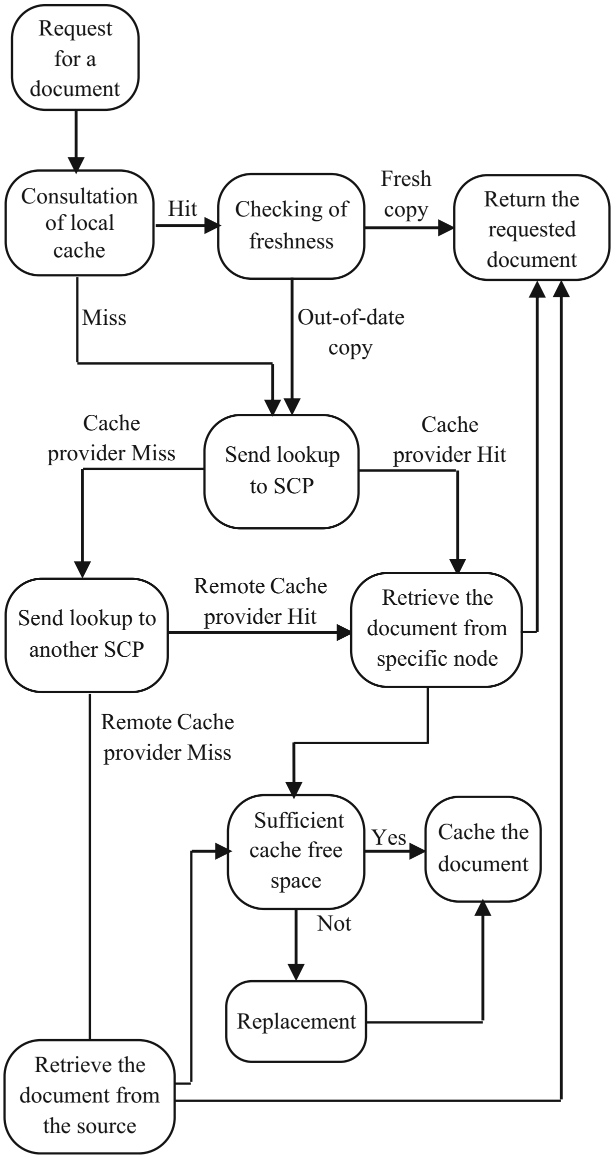


Fig. 1 Behavior of the cache receiving a request for a document

3.5 Election of the Services Cache Providers

As shown in Fig. 2, the election of SCPs is executed periodically following rounds in order to balance workload, and energy consumption between nodes. Each node calculates $R_{div}(i)$ as defined in Table 1, those finding zero will be elected as SCPs. After this step, all elected SCPs send invitation messages to the neighboring at one or more hops, according to j of $S_{Nb(i,j)}$ defined in Table 1, in order to request them to send their respective cache summary. Upon receiving multiple invitation messages, non-SCPs nodes can reply, at most, to three

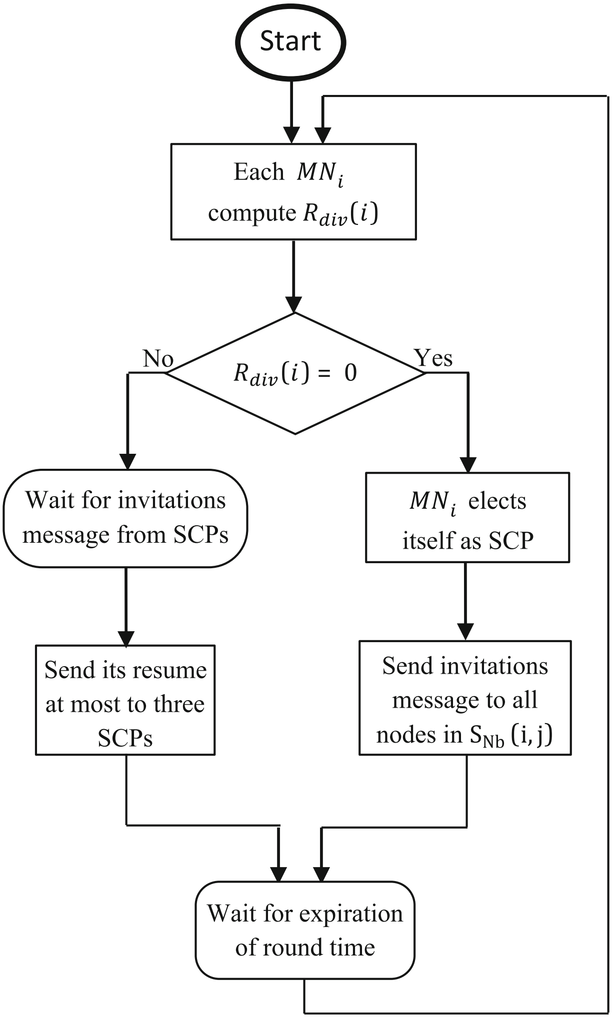


Fig. 2 Flowchart of the election mechanism of SCPs

providers. After sending their respective summaries to SCPs, non-SCPs nodes are required to validate their summaries periodically, in all SCPs with which they are associated.

4 Simulation Model

We model our system as a set of mobile nodes (MN) moving in a rectangular area, which spontaneously formed an Ad Hoc network. Among the mobile nodes, some of them can directly connect to the Internet; those nodes are considered as Access Points (AP) for the rest of mobile nodes in the Ad Hoc network. We assume that each access point contains all requested documents.

Table 1 List of simulation notations and their significations

Notations	Significations
N	Number of nodes in the network
MN_i	Mobile node $i, i = 1\ldots N$
L	Service cache range in number of hops
ID_i	Node identifier which can take values between 1 and N
$p_{provider}$	Percentage of cache service providers
$S_{nb}(i,j)$	Set neighbors of MN_i at j hops
N_{round}	The number of the current round
$R_{div}(i)$	The remainder division of $(ID_i + N_{round})$ by $(1/p_{provider})$
D	Number of documents
L_RH_i	Local request hit rate of mobile node i,$i = \overline{1\ldots N}$
L_BH_i	Local byte hit rate of mobile node i, $i = \overline{1\ldots N}$
$SCP_j\_RH_i$	Request hit at the j $thSCP$ of mobile node i, $i = \overline{1\ldots N}$, $j = \overline{1\ldots 3}$
$SCP_j\_BH_i$	Byte hit at the j $thSCP$ of mobile node i, $i = \overline{1\ldots N}$, $j = \overline{1\ldots 3}$
G_LRH	Global request hit rate of Local caches of all mobile nodes
G_LBH	Global byte hit rate of Local caches of all mobile nodes
$G\_SCP_j\_RH$	Global request hit at the j $thSCP$ of all mobile nodes, $j = \overline{1\ldots 3}$
$G\_SCP_j\_BH$	Global byte hit at the j $thSCP$ of all mobile nodes, $j = \overline{1\ldots 3}$
Bd_i	Size of document d_i (byte)
LH_MN_i	Set of requests that have had a local hit in mobile node i
$SCP_jH\_MN_i$	Set of requests that have had a hit at the jth SCP of mobile node i
R_MN_i	Set of requests received by mobile node i
LH_SD_i	Set of all documents concerned by requests that have had a local hit in mobile node i
$SCP_jH\_SD_i$	Set of all documents concerned by requests that have had a hit at the jth SCP of mobile node i
R_SD_i	Set of all documents concerned by requests received by mobile node i

4.1 Notations

According to the system model described above, we can express some notations summarized in Table 1. The evaluation metrics described in Table 1, can be expressed in Eqs. (2), (3), (4) and (5).

$$L\_RH_i = \frac{|LH\_MNi|}{|R\_MNi|} \tag{2}$$

$$SCP_j\_RH_i = \frac{|SCP_jH\_MNi|}{|R\_MNi|} \qquad j = \overline{1\ldots 3} \tag{3}$$

$$L\_BH_i = \frac{\sum_{d_l \in LH\_SD_i} Bd_l}{\sum_{d_k \in R\_SD_i} Bd_k} \tag{4}$$

$$SCP_j\_BH_i = \frac{\sum_{d_l \in SCP_jH\_SD_i} Bd_l}{\sum_{d_k \in R\_SD_i} Bd_k} \qquad j = \overline{1\ldots3} \tag{5}$$

Global rates are the average of local rates, so they can be expressed in Eqs. (6), (7), (8) and (9).

$$G\_LRH = \frac{1}{N}\sum_{i=1}^{N} L\_RH_i \tag{6}$$

$$G\_LBH = \frac{1}{N}\sum_{i=1}^{N} L\_BH_i \tag{7}$$

$$G\_SCP_j\_RH = \frac{1}{N}\sum_{i=1}^{N} SCP_j\_RH_i \qquad j = \overline{1\ldots3} \tag{8}$$

$$G\_SCP_j\_BH = \frac{1}{N}\sum_{i=1}^{N} SCP_j\_BH_i \qquad j = \overline{1\ldots3} \tag{9}$$

4.2 Node Mobility Model

Each mobile node is deployed randomly within the simulation area; we use the Random Waypoint Movement model to simulate the moving pattern of the mobile nodes. At the beginning, each mobile node selects a random destination, which is a point on border of simulation area. To reach its destination, the mobile node moves for a random period of time and pauses for another random period in alternation, according to the following sequence: Moving period, pause period and speed. All are distributed uniformly between a minimum value and a maximum value (Refer to Table 2). After reaching the destination, the mobile node selects another destination and repeats the movement pattern.

4.3 Performance Evaluation Parameters

In our simulation, we have mainly used three performance parameters. Each of these parameters is motivated by its direct impact on a specific performance criterion of mobile Ad Hoc networks. The first one is response time or latency perceived by users (expressed as latency gain). The latency gain varies whether the request is satisfied from local cache of requester node or from local cache of another node designated by CCSP scheme. The second one is request hit rate, which is closely linked to data accessibility performance criteria. In fact, due to mobility of network nodes, a source of documents could not be reachable from a given node, so all document requests of this node can be satisfied from a local cache of this or other network nodes. The third one is byte hit rate. This parameter has a greater impact on reducing the network workload. Indeed, in situation without using cache mechanism, upon formulation of request by a given node, the requested document should transit the network from original server until the requester node.

Table 2 Simulation parameters

Parameters	Values
Bandwidth	1 Mbit/s
Number of nodes	100
Inter-request maximum time	3 s
Number of updating request, which can be ignored by node before deleting its cache summary from SCP	2
Inter-round time period	70 s
Time period of SCPs to request all managed cache nodes to update their summaries	10 s
Inter-invitation time of each SCP to neighbors nodes to send their local cache summaries	5 s
Simulation time	200 s
Cache size	32 Mbits
Simulation area	100 × 100 (m)
Interval of pause time	[0–5] (s)
Interval of walking time	[2–8] (s)
Interval of walking speed	[2–20] (m/s)
Radius of communication range	30 m
The percentage of SCP node (service cache provider)	20%

Synthetic workload parameters

Parameters	Values
Parameter of the Zipf law distribution for documents frequencies; it is between 0 and 1	0.8
Parameter of the pareto law distribution for documents size; it is between 0 and 2	1.5
Proportion of distinct documents relative to the total number of requests; it is between 25 and 40%	30
Proportion of 1-timer documents relative to the total number of distinct documents; it is between 40 and 70%	50
Smallest document size (bytes)	77
Largest document size (bytes)	1,066,229
Mean document size (bytes)	8840

4.4 Network Workload Trace

In our simulation, the used workload trace is web traffic. This is supported by our study on mobile Ad Hoc networks (MANETs), indeed the main application used by nodes of MANETs is the world wide web (WWW). The web traffic has specific characteristics, much works and studies in scientific literature have focused on workload characterization [29–32]. These studies are based on statistics and probabilistic methods. As result, several tools have been developed for the reproduction of web traffic that has characteristics as close as possible to its original and real characteristics. For our simulation we have used synthetic web traffic generator tool called ProWGen [33, 34]. The Characteristics of our synthetic web traffic, generated with this tool are as follows:

1. The web documents frequencies follow Zipf probability law, its detailed parameters and formula can be found in [33].
2. The web documents sizes follow heavy-tailed probability distribution [13, 23]. Detailed parameters and formula of used probability distribution law can be found in [33].
3. The proportion of distinct documents, relative to the total number of requests is 30%.
4. The percentage of one-timers documents (documents which are consulted only once) is 50%.

Table 2 summarizes the important simulation parameters values of our approach.

4.5 System Modeling Description

In our simulation, mobile nodes are the mains entities, around which; we have set up three processes: (i) mobility process, (ii) cooperation scheme maintenance process and (iii) document research process. Each process consists of succession of events. Flowchart of events execution of these three processes is illustrated in Figs. 3, 4 and 5 respectively.

These three processes are executed from beginning to end of the simulation. Their execution includes one or more mobile nodes. All events created throw this execution are placed into execution queue or events list. Each event is associated with a timestamp (it is a precise occurrence time in which the event must be executed).

The execution of each event induces one or more events. The induced events, in turn, are stamped with their timestamp and are placed in the events list. Simulation loops by extracting and executing event with the smallest timestamp, from the event list. It runs until the event list is empty or the simulation stop time is reached.

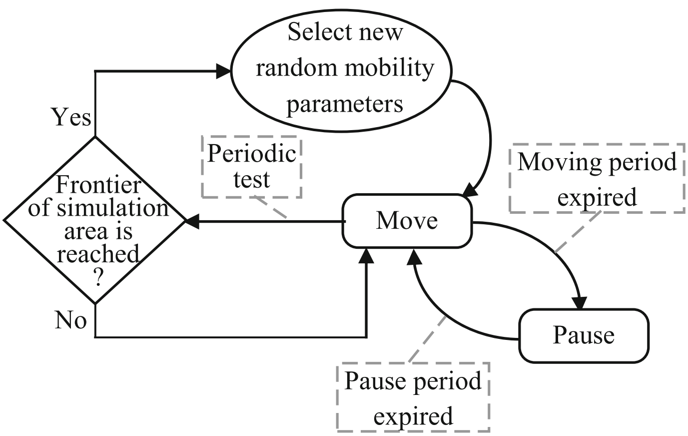


Fig. 3 Description of mobility process

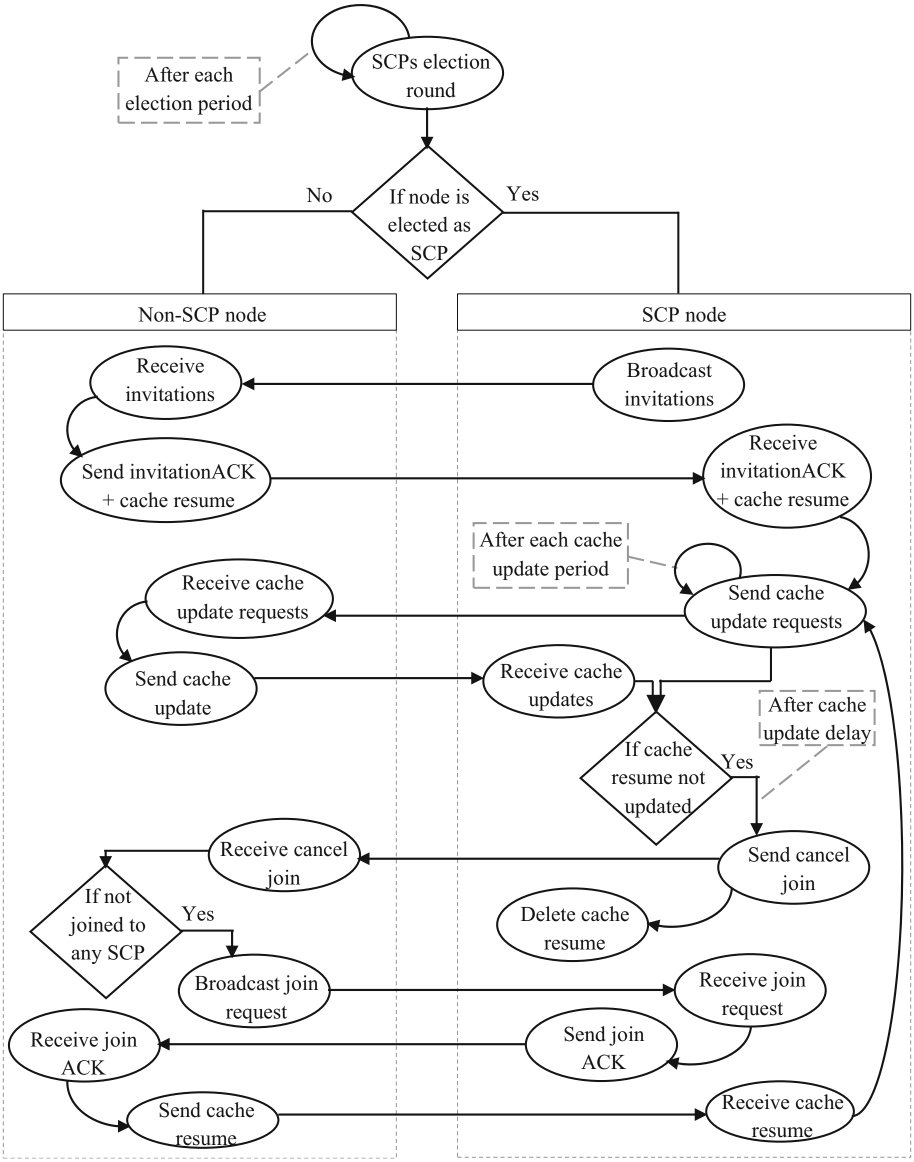


Fig. 4 Description of cooperation scheme maintenance process

5 Simulation Results and Discussion

In this section, we present the simulations experiments of cooperative caching scheme CCSP by comparing it with SimpleCache scheme (SC) reported in [15] and CC scheme reported in [14]. In SimpleCache scheme each node has a local cache, and in case of local cache miss, node's requests are forwarded to the node source.

In the proposed CCSP scheme, network nodes can be associated to 0, 1, 2 or 3 SCPs at the same time. The node mobility leads to connections/disconnections of SCPs, which in

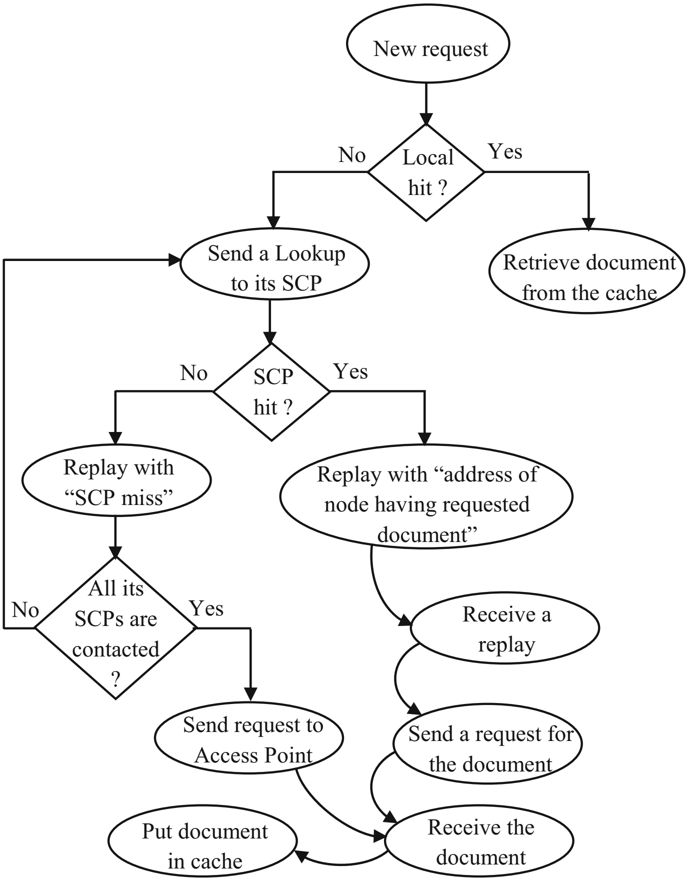


Fig. 5 Description of document research process

turn varied continually the association number of node with the SCPs. Figure 6 shows the average rate of these associations. It can be clearly seen that after 89.28% of simulation time, all nodes are associated to 3 SCPs simultaneously, which allows us to achieve a high level of cooperation. On the other hand, only 0.42% of simulation time where nodes are not associated to any SCPs is obtained. In that case, the nodes are not able to cooperate in any way, with other nodes. So, they can only use their local cache. The global request hit rate refers to the satisfaction rate of all requests made on the network. In CCSP scheme, a document request can be satisfied whether from local cache of requester node, or from one of its three SCPs. Figure 7 shows the part of contribution of each SCP. One can notice local cache contribution remains insufficient despite the gradually filling up over time of caches. On the other hand, the contribution of the first SCP (SCP1) is the more important one in terms of gain in latency. In the beginning, this contribution is widely assisted by the second SCP (SCP2) and the third SCP (SCP3), but over time, by the progressive filling of local caches, the contribution of SCP1 becomes more important comparatively to SCP2 and SCP3.

Fig. 6 Average period of nodes associated to different number of SCPs

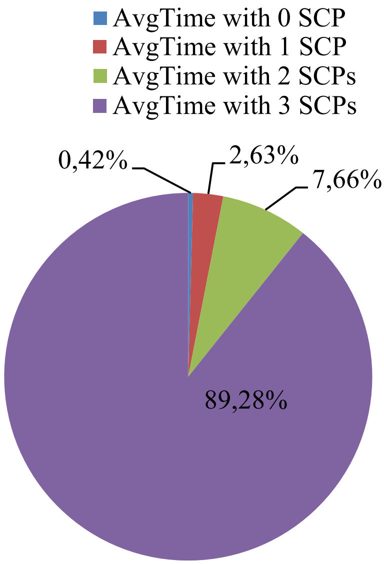
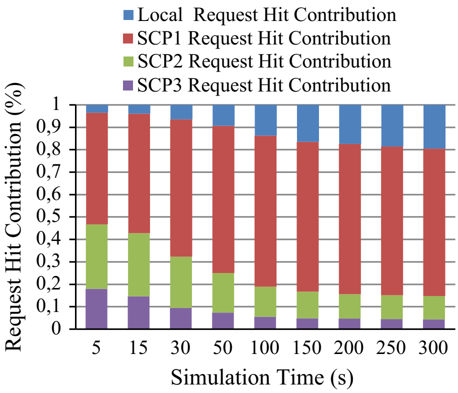


Fig. 7 Request hit contribution of local cache, SCP1, SCP2 and SCP3 versus simulation time

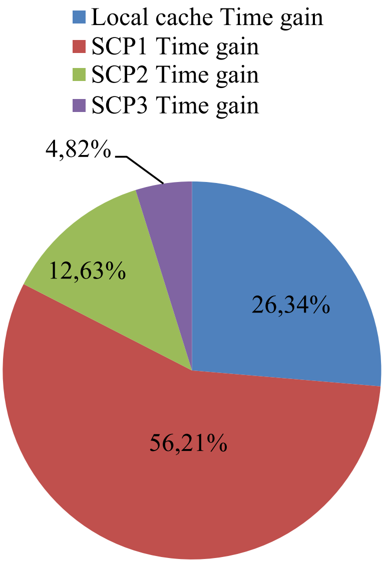


To explain in more detail Fig. 8, which represents the gains realized on latency, we define the following two functions: OSL_{Time} and $CCSP_{Time}$, describing latency.

$$OSL_{Time},\ CCSP_{Time}:\ Q \rightarrow R$$

where Q is the set of request and R is the set of real numbers representing latency times. The first functionOSL_{Time} (Original Source Latency Time) returns for each request the latency time from the original source of the document. In other words, it returns latency time of request without using caches. The second one, $CCSP_{Time}$ (Cooperative Caching based on Service Providers) returns latency time for each request by using CCSP scheme. The request could be satisfied either from the local cache of requester node or from one of the three SCPs. Given the foregoing background together, it is now possible to identify the gains on latency for the local cache, SCP1, SCP2 and SCP3, which are given respectively by LC_TG and SCP_j_TG where $j = \overline{1\ldots3}$, associated formulas are expressed in Eqs. (10) and (11).

Fig. 8 Comparison of time saving



$$LC\_TG = \sum_{1}^{N} \left(\sum_{q \in LH\_MN_i} \left(OSL_{Time}(q) - CCSP_{Time}(q) \right) \right) \quad (10)$$

$$SCP_j\_TG = \sum_{1}^{N} \left(\sum_{q \in SCP_jH\_MN_i} \left(OSL_{Time}(q) - CCSP_{Time}(q) \right) \right) \quad (11)$$

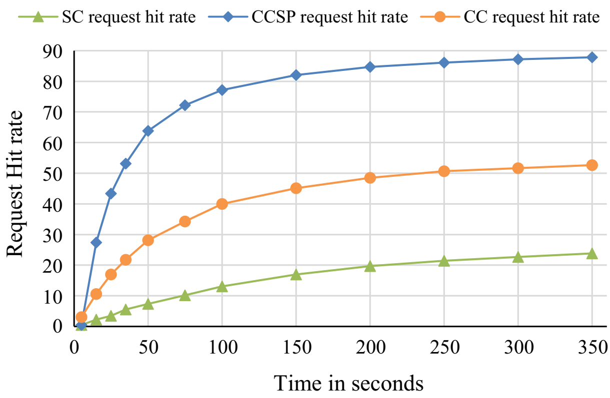
where $j = \overline{1 \ldots 3}$.

In order to satisfy each document request formulated by a mobile node, local cache is the first solicited then, at each cache miss, the SCPs cited above, are solicited sequentially. The SCP can be viewed as a very large cache memory formed by the association of all local caches of mobile nodes managed by the SCP. It can be seen from Fig. 8, that the most dominant gain in latency concerns SCP1. So, a very large number of requests that are not being met by the local cache are redirected to SCP1, where they could be satisfied. The few missing remaining requests with miss in SCP1 will be redirected to SCP2, and a very few missing request in SCP2 will be redirected to SCP3. This may explain the higher control of $SCP_1 TimeGain$ than SCP_2 and $SCP_3 TimeGains$. The weak $SCP_3 TimeGain$ can also be explained by the fact that document research process may take a long time when the request is redirected until SCP3. In some cases, this time may be very close to the one exhausted by the request to get document from the origin server. Therefore, as revealed in Fig. 8, a low $SCP_3 TimeGain$ is achieved.

Local request hit rate and local byte hit rate are equivalent to performances, which can be obtained with network operating without cooperation of local caches. Request hit rate is closely linked to data accessibility. In fact, due to mobility of network nodes, a source of documents could not be reachable from a given node, so all document requests of this node can be satisfied from a local cache of this or other network nodes.

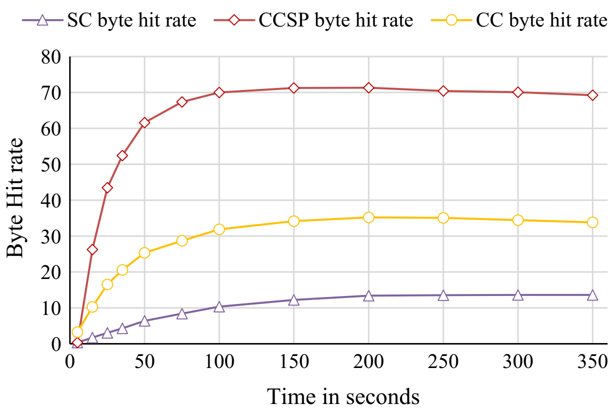
As we can see, in Fig. 9, the gain in request hit rate is very significant by using CCSP scheme. The request hit rate is more stable over time, it reaches more than 85% in case of

Fig. 9 Evolution of request hit rate versus time in seconds



CCSP scheme and at the same time, that of CC scheme is about 52% and SC request hit rate was just above the 20%. This represents a gain of about 65 and 35% of satisfied requests through caches cooperation, compared respectively with CC scheme and SC scheme. CCSP scheme provides also considerable gain for byte hit rate. A byte hit may be either local byte hit or remote byte hit, in CCSP scheme a remote hit can occur either in SCP1, SCP2 or SCP3. In CC scheme, remote hit occurs when requested document is found within a node belonging to a cluster along the routing path to the source node. In SC byte hit case, the requested document is recovered without any data being transited through the network. However, in CCSP byte hit, the requested document should transit from holder node of that document, to the requester node. This implies that SC byte hit rate will have a greater impact on reduction of the network load, nevertheless, the remote byte hit rates impact, mainly the CCSP byte hit rate are not negligible. One can see from Fig. 10 that when CCSP byte hit rate reached a peak of about 71%, SC byte hit rate is only 14%, and CC byte hit rate achieves 35%. This makes a gain of 57% by using CCSP scheme against the use of only SC scheme. These gains were almost realized by the first SCPs (SCP1). The details of all contributors to these gains are illustrated in Fig. 7. From Figs. 9 and 10, we can also note that contrary to request hit rate, the byte hit rate recorded slight decrease. This may be explained by the used cache replacement policy, frequency, size and distance based value (FSDV) with both CCSP scheme and CC scheme. This policy promotes

Fig. 10 Evolution of byte hit rate versus time in seconds



elimination of large documents to free up more space in the cache and therefore increases the global request hit rate by admission of more documents in the cache.

Figure 11 shows the variation of request hit rate in relation to cache size. We compare request hit rates provided by CCSP, CC or SC schemes. SC request hit rate is achieved without any cooperation unlike CCSP scheme and CC scheme. Therefore, Fig. 11 assesses the gap, between SC request hit rate and those of CCSP and CC schemes. this gap is very much impacted by the cooperation. It should be noted that the essence of cache cooperation is related to the limited capacity of local cache memory. Cooperative caching allows management of all local caches of the same zone by a given node, this provides a global cache with large storage capacity. Therefore, the number of documents present in global cache is increased, and request hit rate is increased too. Figure 11 shows a significant difference between CCSP request hit rate and CC request hit rate, this can be explained by the fact that unlike CCSP scheme, which use for each node three SCPs at the same time, the CC scheme use for each node only one CSN. It can be shown also in Fig. 11 that all request hit rates are constantly evolving with a growth of cache size. It is also interesting to notice some stability of all request hit rates with a growth of cache size after 96 MB. This stability may be explained by the satisfaction of a large number of requests from caches due to the abundant cache free space.

Figures 12 and 13 show the variation of respectively, the number of exchanged messages in the network and the size of network load, in relation to cache size. The first finding which can be made in these figures, is that a number of exchanged messages and a size of network load are closely linked, as illustrated by the curve profiles. The second finding is the decrease of number of exchanged messages and size of network load by increasing cache size. Indeed, increasing cache size allows caching more documents, thus, more requests can be served by intermediate nodes without forwarding lookup messages further towards the source nodes. Consequently, this will induce significant reduction of network load. In Fig. 12, it can be observed also that for cache size of 8–16 MB, CC scheme generates less number of messages than CCSP scheme, but for all others cache size over than 16 MB, it becomes the other way around. However, we can note that this difference is not significant, since it varies between 5 and 8% of the total number exchanged messages.

Figure 14 shows average latencies time of all requests in case of CCSP scheme and CC scheme, depending on the cache size, from 8 to 128 MB. We note that the curve profiles are inversely proportional to cache size, which is logical, because, with the growth of cache

Fig. 11 Evolution of request hit rate versus cache size

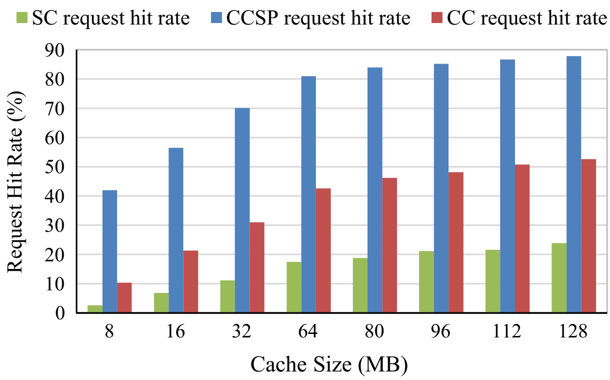


Fig. 12 Evolution of number of exchanged messages versus cache size

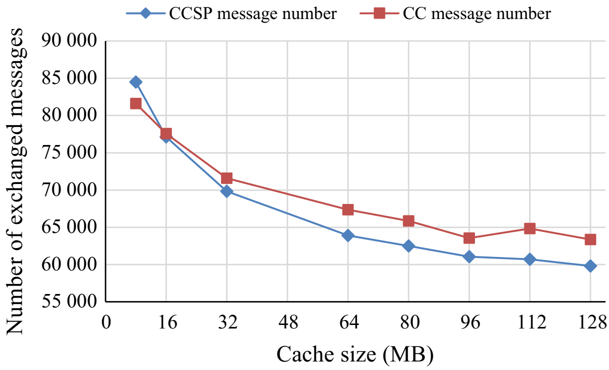


Fig. 13 Evolution of network load size versus cache size

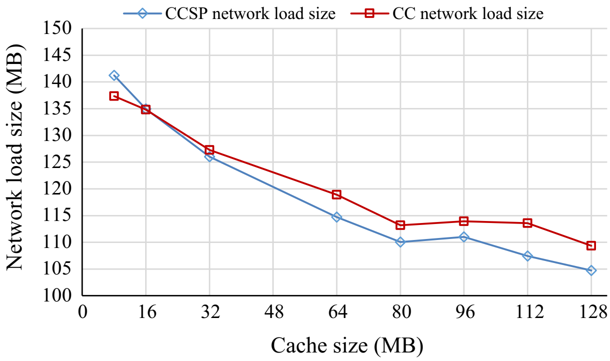
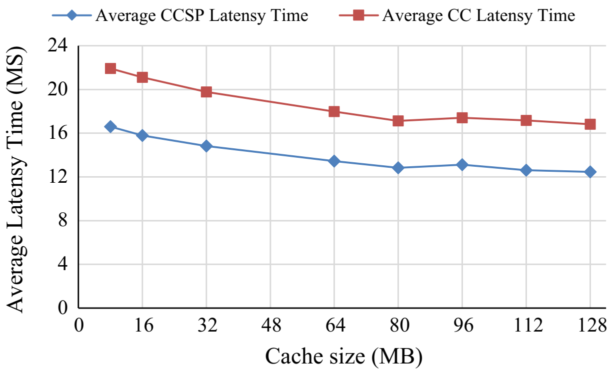


Fig. 14 Evolution of average latency time versus cache size



size, more and more requests are served from intermediate cache nodes without forwarding lookup messages further towards the source nodes. Therefore, latency is reduced. We note in Fig. 14 that, although the difference in terms of request latency time between the two schemes is clearly visible, but in reality, this difference is not significant, because it does not exceed 5.5 ms.

Figure 15 shows the variation of time gain depending on the cache size, from 8 to 128 MB. We compare time gain achieved by SC scheme, CCSP scheme and CC scheme. Despite the slight difference in terms of request latency time between the two schemes

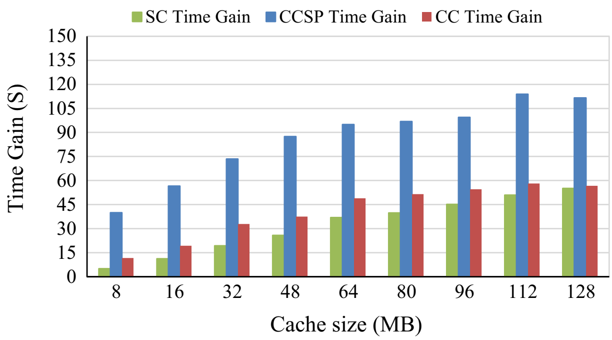


Fig. 15 Evolution of time gain versus cache size

CCSP and CC that is illustrated in Fig. 14, in Fig. 15, we note a significant gap between the time gains of these two schemes. This can be explained by the higher request hit rate achieved by CCSP scheme, compared with that of CC scheme (This is illustrated in Fig. 11).

6 Conclusion

The use of cache memories by each node in mobile Ad Hoc networks and its association with adequate cooperative cache management scheme constitutes one of the best solutions for improving data accessibility, reducing user's latency and overall network load. In this paper, we have proposed the CCSP scheme (cooperative caching scheme based on service providers).

We were interested on time gain induced by the CCSP scheme in different levels, namely: local, SCP1, SCP2 and SCP3. The biggest gain was revealed in SCP1. With increasing in SCP levels, the time gain decreases. The gain remained positive until in SCP3. Beyond this level, the gain become negative. Indeed, the process of looking for document in local caches of network nodes may take more time than to bring it directly from an accessible original server. Simulation results showed that CCSP scheme performed a request hit rate of over than 85%, this performance criteria is closely linked to data accessibility.

As regards the byte hit rate performance criteria, it was shown that the local byte hit rate had a greater impact on reduction of the network load. Moreover, the CCSP byte hit rates, mainly the SCP1 byte hit rate had also non-negligible impact on offloading.

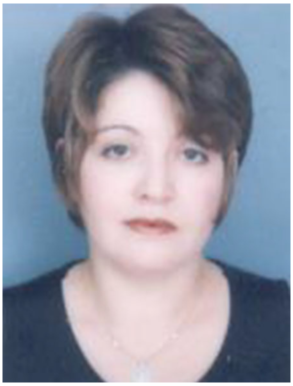
This study showed that the criteria of multi-level document research, adopted by CCSP scheme is very beneficial for improving data accessibility. Moreover, this proposition contributes effectively on offloading of mobile Ad Hoc network, in term of volume of exchanged information. As prospects, it will be interesting to explore possible improvements by using control admission mechanism for each candidate document to be cached. It will be interesting also to apply coordinated cache replacement policy by considering all documents cached locally in all nodes of the same neighborhood.

References

1. Perkins, C. E. (2008). *Ad hoc networking* (1st ed.). Reading, MA: Addison-Wesley Professional.
2. Martikainen, O. E. (2006). Complementarities creating substitutes—Possible paths towards 3g, wlan/wimax and ad hoc networks. *Info*, *8*(4), 21–32.
3. Remy, G., Senouci, S. M., Jan, F., & Gourhant, Y. (2013). Coupling lte and network coding for efficient dissemination in vehicular networks. *Global Information Infrastructure Symposium - GIIS*, *2013*, 1–7.
4. Arlitt, M., Cherkasova, L., Dilley, J., Friedrich, R., & Jin, T. (2000). Evaluating content management techniques for web proxy caches. *SIGMETRICS Performance Evaluation Review*, *27*(4), 3–11.
5. Aggarwal, C., Wolf, J. L., & Yu, P. S. (1999). Caching on the world wide web. *IEEE Transactions on Knowledge and Data Engineering*, *11*(1), 94–107.
6. Chan, E., Li, W., & Chen, D. (2009). Energy saving strategies for cooperative cache replacement in mobile ad hoc networks. *Pervasive and Mobile Computing*, *5*(1), 77–92.
7. Abrams, M., Standridge, C. R., Abdulla, G., Williams, S., & Fox, E. A. (1995). Caching proxies: Limitations and potentials. Technical report, institution, Blacksburg, VA, USA.
8. Reddy, M., & Fletcher, G. P. (1998). Intelligent web caching using document life histories: A comparison with existing cache management techniques. In *3rd International WWW caching workshop* (pp. 35–50).
9. Cao, P., & Irani, S. (1997). Cost-aware www proxy caching algorithms. In *Proceedings of the USENIX symposium on internet technologies and Systems*, USITS'97, Berkeley, CA, USA (p. 18). USENIX Association.
10. Zhang, N., Kimmerlin, M., Costa-Requena, J., & Hämmäinen, H. (2016). Cost efficiency of mobile in-network caching. *International Journal of Network Management*, *26*(1), 44–55 (NEM-15-0111.R1).
11. Du, Y., Gupta, S. K. S., & Varsamopoulos, G. (2009). Improving on-demand data access efficiency in MANETs with cooperative caching. *Ad Hoc Networks*, *7*(3), 579–598.
12. Xu, J., Hu, Q., Lee, W.-C., & Lee, D. L. (2004). Performance evaluation of an optimal cache replacement policy for wireless data dissemination. *IEEE Transactions on Knowledge and Data Engineering*, *16*(1), 125–139.
13. Wu, K.-L., Yu, P. S., & Chen, M.-S. (1996). Energy-efficient caching for wireless mobile computing. In *Proceedings of the twelfth international conference on data engineering* (pp. 336–343).
14. Chand, N., Joshi, R. C., & Misra, M. (2007). Cooperative caching strategy in mobile ad hoc networks based on clusters. *Wireless Personal Communications*, *43*(1), 41–63.
15. Yin, L., & Cao, G. (2006). Supporting cooperative caching in ad hoc networks. *IEEE Transactions on Mobile Computing*, *5*(1), 77–89.
16. Lim, S., Lee, W.-C., Cao, G., & Das, C. R. (2006). A novel caching scheme for improving internet-based mobile ad hoc networks performance. *Ad Hoc Networks*, *4*(2), 225–239.
17. Lai, K. Y., Tari, Z., & Bertok, P. (2004). Location-aware cache replacement for mobile environments. In *Global telecommunications conference, 2004. GLOBECOM '04. IEEE* (Vol. 6, pp. 3441–3447).
18. Dimokas, N., Katsaros, D., & Manolopoulos, Y. (2010). Cache consistency in wireless multimedia sensor networks. *Ad Hoc Networks*, *8*(2), 214–240.
19. Sailhan, F., & Issarny, V. (2003). Cooperative caching in ad hoc networks. In *4th International conference on mobile data management: MDM 2003* (pp. 13–28), Melbourne, Australia.
20. Tang, B., Gupta, H., & Das, S. R. (2008). Benefit-based data caching in ad hoc networks. *IEEE Transactions on Mobile Computing*, *7*(3), 289–304.
21. Fiore, M., Casetti, C., & Chiasserini, C. F. (2011). Caching strategies based on information density estimation in wireless ad hoc networks. *IEEE Transactions on Vehicular Technology*, *60*(5), 2194–2208.
22. Chand, N., Joshi, R. C., & Misra, M. (2005). Cooperative cache management in mobile ad hoc networks. In *2005 2nd Asia Pacific conference on mobile technology, applications and systems* (pp. 1–7).
23. Chang, Y.-K., & Ting, Y.-W. (2007). A novel cooperative caching scheme for wireless ad hoc networks: Groupcaching. *2007 International conference on networking, architecture, and storage* (pp. 62–68).
24. Ting, I.-W., & Chang, Y.-K. (2013). Improved group-based cooperative caching scheme for mobile ad hoc networks. *Journal of Parallel and Distributed Computing*, *73*(5), 595–607.
25. Zhang, N., & Hämmäinen, H. (2015). Cost efficiency of sdn in lte-based mobile networks: Case finland. In *2015 International conference and workshops on networked systems* (*NetSys*) (pp. 1–5).
26. Naudts, B., Kind, M., Westphal, F. J., Verbrugge, S., Colle, D., & Pickavet, M. (2012). Techno-economic analysis of software defined networking as architecture for the virtualization of a mobile network. In *2012 European workshop on software defined networking* (pp. 67–72).

27. Sureshkumar, A., & Samson, R. S. (2016). Swarm and fuzzy based cooperative caching framework to optimize energy consumption over multimedia wireless sensor networks. *Wireless Personal Communications*, *90*(2), 961–984.
28. Feng, B., Zhou, H., Zhang, H., Jiang, J., & Yu, S. (2016). A popularity-based cache consistency mechanism for information-centric networking. In *2016 IEEE global communications conference* (*GLOBECOM*) (pp. 1–6).
29. Crovella, M. E., & Bestavros, A. (1997). Self-similarity in world wide web traffic: Evidence and possible causes. *IEEE/ACM Transactions on Networking*, *5*(6), 835–846.
30. Christensen, K. J., & Javagal, N. J. (1997). Prediction of future world wide web traffic characteristics for capacity planning. *International Journal of Network Management*, *7*(5), 264–276.
31. Arlitt, M., Friedrich, R., & Jin, T. (1999). Workload characterization of a web proxy in a cable modem environment. *SIGMETRICS Performance Evaluation Review*, *27*(2), 25–36.
32. Mahanti, A., Williamson, C., & Eager, D. (2000). Traffic analysis of a web proxy caching hierarchy. *Network Magazine of Global Internetworking*, *14*(3), 16–23.
33. Busari, Mudashiru, & Williamson, Carey. (2002). Prowgen: A synthetic workload generation tool for simulation evaluation of web proxy caches. *Computer Networks*, *38*(6), 779–794.
34. Markatchev, N., & Williamson, C. (2002). Webtraff: A gui for web proxy cache workload modeling and analysis. In *Proceedings on 10th IEEE international symposium on modeling, analysis and simulation of computer and telecommunications systems* (pp. 356–363).

Ali Larbi received the engineer degree in Computer Science from University of Tizi-Ouzou (Algeria) in 2006 and the magister degree in Networks and Distributed Systems from University of Bejaia (Algeria) in 2009. Currently, he is Ph.D student at University of Bejaia, Member of LaMOS Research Unit (Modeling and Optimization of Systems). His research interests include mobile networks (Ad Hoc networks, sensors networks…), their architecture, application and resource management.



Louiza Bouallouche-Medjkoune received the engineer degree in Computer Science from University of Setif (Algeria), the Magister degree in Applied Mathematics from University of Béjaïa (Algeria); the Doctorate degree in Computer Science from University of Setif in 2006 and the HDR from the University of Constantine in 2009. She works as a teacher at the department of Computer Science of University of Béjaïa and as a researcher at LaMOS Research Unit of University of Béjaïa. Bouallouche-Medjkoune is head of the computer science teaching staff since 2008, and head of the research team EPSIRT since 2005. She is also head of department of Operational Research since January 2010 to July 2015. Her publications have appeared in various publishers: Taylor&Francis, Springer, AMS, BCS. Her research Interests are: Performance Evaluation of Computer Systems and telecommunication networks, Stability of Systems, Markov Chains, Queueing theory, Computer Networks (ad hoc networks, sensors networks...), Quality of Service.

Djamil Aissani was born in 1956 in Biarritz (Basque Country, France). He started his career at the University of Constantine in 1978. He received his Ph.D in 1983 from Kiev State University (Soviet Union). He is at the University of Bejaia since its opening in 1983/1984. Director of Research, Head of the Faculty of Science and Engineering Science (1999–2000), Director of LaMOS (Modeling and Optimization of Systems) Research Unit at Béjaïa University, Scientific Head of the Computer Science Doctorate School, he has taught in many universities (USTHB Algiers, Annaba, Tizi-Ouzou, Rouen, Dijon, ENITA,...). He has published many papers on Markov chains, queueing systems, reliability theory, performance evaluation and their applications in such industrial areas as electrical and telecommunication networks and computer systems.