2013 IEEE 27th International Conference on Advanced Information Networking and Applications

Detecting Energy-Efficient Central Nodes for Cooperative Caching in Wireless Sensor Networks

Nikos Dimokas
Engineering Informatics and Telecommunications
University of Western Macedonia
Kozani, Greece
Email: dimokas@delab.csd.auth.gr

Dimitrios Katsaros
Centre for Research & Technology Hellas
& University of Thessaly
Volos, Greece
Email: dkatsar@inf.uth.gr

***Abstract*—The deployment of wireless sensor networks in many application areas like environment control, target tracking in battlefields, requires an optimization to the communication among the sensors so as to serve data in short latency and with minimal energy consumption. Cooperative data caching has been proposed as an effective and efficient technique to achieve these goals concurrently. The design of protocols for such networks depends mainly on the selection of the sensors which will take special roles in coordinating the procedure of caching and take forwarding decisions. This article introduces a new metric to aid in the selection of such nodes. Based on this metric, we propose a new energy efficient cooperative caching protocol, which is compared against the state-of-the-art competing protocol. The simulation results attest the superiority of the proposed protocol.**

***Keywords*-wireless sensor networks, cooperative caching, latency, energy conservation, social network analysis.**

I. Introduction

The rapid technological advances in low-power hardware design have enabled the development of tiny battery-powered sensor nodes which are able to compute, sense physical "parameters" and communicate with each other. A wireless sensor network (WSN) is a network of large numbers of sensors nodes, where each node is equipped with limited on-board processing, storage and radio capabilities. Sensor nodes are quasi-stationary, densely deployed and with limited capabilities. Nodes sense and send their signals towards a data center which is called the "information sink". The design of protocols and applications for such networks has to be energy aware in order to prolong the lifetime of the network because it is quite difficult to recharge node batteries. Additionally, it has to take into account the multi-hop communication nature. Communication in a WSN between any two nodes that are out of one another's transmission range is achieved through intermediate nodes, which relay messages to set up a communication channel between the two nodes.

The success of the applications running over WSNs will be determined at a large degree by the optimization of the communication among the sensors. It is critical for the majority of applications to serve the requested data in short latency and with minimal energy dissipation. The cooperative data caching has been proposed as an effective and efficient technique to achieve these goals [1], [2], [3] (for more details cf. Section II). In cooperative caching, multiple sensor nodes share and coordinate cache data without always having to visit the data centers. Since the battery lifetime can be extended if we manage to reduce the "amount" of communication, caching the useful data for each sensor either in its local store or in the near neighborhood can prolong the network lifetime and reduce the communication overhead and the data sources workload. Although the definition of the network lifetime depends on the applications' semantics, a widely accepted definition is the time until the first/last node of the network depletes its energy [4].

In order to address the latency and energy consumption requirements of WSNs, we attempt to identify some sensor nodes as being more significant than the others, w.r.t. coordinating the procedure of caching. This fact helped realize the significance of borrowing concepts from the field of Social Network Analysis [5] (SNA) to the design of more efficient cooperative caching protocols. Social network analysis is based on an assumption of the importance of relationships among interacting units. The social network perspective encompasses theories, models, and applications that are expressed in terms of relational concepts or processes. Along with growing interest and increased use of network analysis has come a consensus about the central principles underlying the network perspective. SNA views social relationships in terms of nodes and ties. Nodes are the individual actors within the networks, while ties are the relationships between the actors.

SNA have attracted significant interest initially from the social and behavioral communities, later from the data mining (Abdallah [6], Hwang et al. [7]) and only recently from the networking community (Katsaros et al. [8], Taghizadeh et al. [9]). SNA comprehends the study and exploitation of the structural information present in the network, such as existence and strength of communities (Saravanan et

Research supported by the EU Network of Excellence "Internet Science" under FP7-ICT Grant Agreement No.288021.

1550-445X/13 $26.00 © 2013 IEEE
DOI 10.1109/AINA.2013.120

al. [10]), node centralities, network robustness to node removal, topology evolution over time (Gilbert et al. [11]) and so on. Among the most significant tasks involved in SNA is the calculation of centrality measures [12]. Point centrality in communication is based on the concept of betweenness. According to betweenness centrality, a node is central to the degree that it stands between others. Various other measures of centrality have been proposed to determine the importance of a node within a graph [13] (cf. Section II).

The fundamental aspect in all the proposed cooperative caching schemes for sensor networks is the identification of the nodes which will implement the aspects of the cooperation concerning the caching decisions, i.e., towards which nodes will the data request will be forwarded? which nodes will decide about which data will be cached in which nodes? and so on.

A. Motivation and contributions

The early proposals for cooperative caching in WSNs and Mobile Ad hoc Networks [1], [14] (MANETs), did not pay attention to the selection of nodes that will have special roles in the cooperation protocol. The work [2] pointed out the singificance of the careful selection of these nodes; it was argued there, that these nodes should be "central" in the sensor network topology. Based on this, the authors proposed a cooperation scheme where the sensors with special role were selected based on their *ability to influence the communication between pairs of other nodes*. This ability was quantified by calculating the *Node Importance index* – NI for each sensor, which is a localized version of the well-known betweenness centrality index used in social network analysis.

Network lifetime is prolonged by distributed energy consumption. Nevertheless, the calculation of NI metric did not take into account the remaining energy of sensor nodes in order to determine the significant nodes. NI metric elects always the same nodes as significant nodes. This has as a consequence that central nodes will become points of failure for WSNs. Thus, WSN can be partitioned due to absence of an energy balancing policy.

Motivated by the aforementioned shortcoming, this article proposes a new energy efficient metric to evaluate the significance of a sensor to undertake special roles in the cooperation, and based on this it describes a new cooperative caching protocol. In particular, the article makes the following contributions:

- It describes a new centrality metric for sensor nodes, the EBC (Energy Betweenness Centrality), which is a generalization of NI metric. The new metric is energy efficient since its computation exploits the remaining energy of sensor nodes. Additionally, the election of significant nodes is not a static procedure, avoiding fast depletion of their energy.
- It develops a new cooperative caching protocol for WSNs, the EBCCoCa (EBC Cooperative Caching); this protocol is compared against the state-of-the-art protocol via simulation analysis, which attests the superiority of the proposed protocol in terms of energy consumption.

The rest of this article is organized as follows: In Section II we survey the most important works relevant to this article, while in Section III we describe the new centrality metric and the component of the proposed cooperative caching protocol. Section IV presents the simulation environment that was built to investigate the performance of the proposed protocol, and also describes the experiments and obtained results after the comparison of the protocol with the competing state-of-the-art schemes. Finally Section V conludes the article.

II. Relevant work

The issue of cooperative caching has attracted significant attention in the literature concerning various types of distributed systems; in the Web [15], in file servers [16], and so on. Nevertheless, the very limited capabilities of the sensor nodes (in terms of energy, storage, and computation), the particularities of the wireless channel (varible capacity), and the multi-hop fashion of communication, turns the solutions proposed in the aforementioned environments, of limited usefulness.

In distributed systems over wireless networks based on multi-hop communication, cooperative caching has been proven a very efficient strategy to shorten the communication latency and conserve energy. Nuggehalli et al. [17] addressed the problem of energy-conscious cache placement in wireless ah hoc network, and [18] considered the cache placement problem of minimizing total data access cost in ad hoc networks with multiple data items and nodes with limited memory capacity, and presented a polynomial-time centralized approximation algorithm to attack the problem, since it is NP-hard. Though these works address cache placement issues.

The most important relevant works are those reported in [1], [2], [14]. The work reported in [14] considered cache replacement issues for wireless ad hoc networks but in the context of a very limited form of cooperation; a node which requests a data searches either in its local cache or in the caches of its 1-hop neighbors (otherwise forwards the request to a ficticious data center). Thus, remote hits can not happen take in this protocol. Yin & Cao proposed the Hybrid cooperative caching protocol, which exploited both data and node locality in an homogeneous manner, but this policy was proved inferior to NICoCa, described in [2] which took special consideration to select appropriate "central" nodes to carry out and coordinate the cooperation. However, the selection of "central" nodes does not take into consideration the remaining energy of sensor nodes. Thus, the energy consumption of significant nodes will have as a consequence the reduction of network lifetime and finally the network fragmentation.

There are various measures of centrality and the most prominent of them are going to be presented in the following section.

A. Centrality Metrics

One major concept for the analysis of social networks is centrality. Centrality metrics have been used to identify the role of individual nodes in a network and study their relationship to their neighboring nodes. Even though one of these metrics, betweenness centrality [13], was introduced in the 70s, the research community did not apply social network techniques to sensor networks until only the last couple of years. Betweenness is a centrality measure of a vertex within a graph. Vertices that occur on many shortest paths between other vertices have higher betweenness than those that do not.

The simplest centrality metric is degree centrality [5] and refers to the number of direct connections a node has to its neighbors. Degree is often interpreted in terms of the immediate risk of node for catching whatever is flowing through the network (some information). Degree centrality favors over nodes that have many one-hop neighbors. However, some times this is misleading especially in cases where many one-hop neighbors are at the edge of the network. Another popular centrality metric is closeness centrality. Closeness centrality [5] describes the efficiency of information propagation from one node to all the others, and it is defined as the inverse of the sum of the distances between a given node and all other nodes in the network. The distance can be measured in number of hops, delays, and so on. Closeness centrality gives an estimate of how long it will take information to spread from a given node to the rest of the network actors.

In [7] Hwang et al. proposed a centrality metric called Bridging Centrality (BC). The metric focuses on what the authors call bridging nodes, which are the nodes that are located in between highly connected regions and are therefore crucial for the connectivity and routing inside the network. The main drawback of the algorithm was that it was centralized and therefore global network knowledge was necessary.

All of the above centrality metrics have been defined in a centralized fashion (i.e., taking into account all the network nodes). Such centralized computations are prohibitive for sensor networks due to the communication complexity of learning the whole network topology, and thus localized versions of them have been used in the literature of protocol design. Localized centrality metrics have been proposed in the recent literature [2], [8], [19]. However, these centrality metrics does not take into account the energy consumption of sensor nodes. Thus, the sensor nodes elected by centrality metrics will deplete their battery quite fast.

III. The New Cooperative Caching Scheme

One of the main parts of the proposed protocol is the estimation of the importance of sensors relative to the network topology and the remaining energy. The intuition is that if we discover those nodes which have enough remaining energy and reside in a significant part of the (short) paths connecting other nodes, then these are the "important" nodes; then they may be selected as coordinators for the caching decisions, i.e., as "mediators" to provide information about accessing the requested data or even as caching points.

A. Measuring sensor node importance

A wireless sensor network is abstracted as a graph $G(V, E)$, where V is the set of its nodes, and E is the set of radio connections between the nodes. An edge $e = (u, v),\ u, v \in E$ exists if and only if u is in the transmission range of v and vice versa. All links in the graph are bidirectional, i.e., if u is in the transmission range of v, v is also in the transmission range of u. The network is assumed to be in a connected state. The set of neighbors of a node v is represented by $N_1(v)$, i.e., $N_1(v) = \{u : (v, u) \in E\}$, while the set of two-hop nodes of node v is represented by $N_2(v)$, i.e., $N_2(v) = \{w : (u, w) \in E$, where $w \neq v$ and $w \notin N_1$ and $(v, u) \in E\}$. The combined set of one-hop and two-hop neighbors of v is denoted as $N_{12}(v)$.

A *path* from $u \in V$ to $w \in V$ has the common meaning of an alternating sequence of vertices and edges, beginning with u and ending with w. The *length* of a path is the number of intervening edges. We denote by $d_G(u, w)$ the *distance* between u and w, i.e., the minimum length of any path connecting u and w in G, where by definition $d_G(v, v) = 0,\ \forall v \in V$ and $d_G(u, w) = d_G(w, u),\ \forall u, w \in V$. Note that the distance is not related to network link costs (e.g., latency), but it is a purely abstract metric measuring the number of hops.

Let $\sigma_{uw} = \sigma_{wu}$ denote the number of shortest paths from $u \in V$ to $w \in V$ (by definition, $\sigma_{uu} = 0$). Let $\sigma_{uw}(v)$ denote the number of shortest paths from u to w that some vertex $v \in V$ lies on. Let $sep_{uw} = sep_{wu}$ denote the shortest energy path from node $u \in V$ to node $w \in V$. Shortest energy path is defined as the multi-hop path that connects two nodes with the minimum energy dissipation among the intermediate nodes. The shortest path consists of those nodes that have the biggest amount of remaining energy and sep_{uw} value is the fraction of the summation of edges' weights that correspond to the path to the number of edges. Initially, each node represents its remaining energy E_r as percentage of the initial energy E_i. Then, the computation of weight involves the exchange of remaining energy among neighboring nodes. Each edge $e = (u, v), u, v \in E$ is assigned with weight $w_{uv} = (E_u + E_v)/2$. For instance, the energy path values for the edges connecting two nodes B, E that belong to a local network (see Figure 1) are indicated in Table I. Then, we define the *energy betweenness centrality* index $EBC(v)$ of a vertex v as:

Definition 1: The $EBC(v)$ of a vertex v is equal to:

$$EBC(v) = \sum_{u \neq v \neq w \in V} a \cdot sep_{uw}(v) + (1-a) \cdot \frac{\sigma_{uw}(v)}{\sigma_{uw}}. \quad (1)$$

Large values for the EBC index of a node v indicate that this node v can reach others in relatively short paths and there are short energy paths that pass through node v. Thus, a node v with large value of EBC index can connect other nodes while prolongs the network lifetime. Parameter a determines the degree of participation of shortest energy path in calculation of EBC index. It is obvious that for $a = 0.0$ EBC index is equal to NI index.

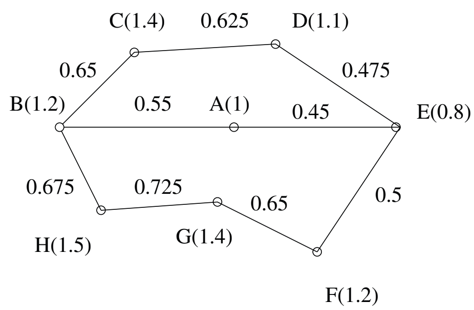

Figure 1. A local network. Each edge is characterized by a weight. The numbers in parentheses denote the remaining energy of the respective node.

Path	Energy Path Value
BAE	0.5
BCDE	0.58
BHGFE	0.63

Table I
THE PATHS THAT CONNECT NODES B AND E AND THE CORRESPONDING VALUES .

B. The EBC Cooperative Caching protocol

Without loss of generality and adopting the model presented in [1], we assume that the ultimate source of data is a *Data Center*. This is not restrictive at all and simply guarantees that every request, if it is not served by other sensor nodes and if does not expire, will finally be served by the Data Center.

At the very first step, it is supposed that each sensor is aware of the number, remaining energy and identity of its 2-hop neighbors; this is achieved with the exchange of "HELLO" messages. We assume that we are able to determine an assignment of time slots to the sensor nodes such that no interference occurs, i.e., no two nodes transmit in the same time slot. Such a scheme can be found using the D2-coloring algorithm from [20]. Then, every node calculates the EBC index of its 1-hop neighbors. The node uses this information in order to characterize some of its neighbors as *mediator* nodes; the minimum set of neighbors with the larger EBC which "cover" its 2-hop neighborhood are the mediator nodes for that node; The node is responsible for notifying its neighbors about which of them are its mediators. Thus, a node can be either a mediator or an ordinary node.

It is supposed that each sensor is aware of its remaining energy and of the free cache space; Additionally, each sensor node stores the following data/metadata:

- The dataID, and the actual data item.
- The latency to obtain a data item (using exponential smoothing).
- The size $Size_i$ of datum i.
- A TTL interval (Time-To-Live) for each datum.
- For each cached item, the timestamps of the K most recent accesses to that item (usually, $K = 2$ or 3).
- Each cached item is characterized either as O (i.e., own) or H (i.e., hosted). If an H-item is requested by the caching node, then its state switches to O.

When a sensor node issues a request for a data item, it searches its local cache. If the item is found there (a local cache hit) then the K most recent access timestamps are updated. Otherwise (a local cache miss), the request is broadcasted and received by the mediators. If none of them responds (a "proximity" cache miss), then the request is directed to the Data Center.

When a non one-hop mediator node receives a request, it searches its local cache. If it deduces that the request can be satisfied by a neighboring node (a remote cache hit), then stops the request's route toward the Data Center, and forwards the request to this neighboring node. If more than one nodes can satisfy the request, then the node with the largest residual energy is selected. If the request can not be satisfied by this mediator node, then it does not forward it recursively to its own mediators. This is due to the fact that these mediators will most probably be selected by the routing protocol as well (AODV) and thus a great deal of savings in messages is achieved. Therefore, during the procedure of forwarding a request toward the Data Center, no searching to other nodes is performed apart from the nodes which reside on the path toward the Data Center.

For every issued request one of the following four cases may take place:

1) Local hit (LH): the requested datum is cached by the node which issued the request. If this datum is valid (the TTL has not expired) then the $EBCCoCa$ is not executed.
2) "Proximity" hit (PH): the requested datum is cached by a node in the 2-hop neighborhood of the node

which issued the request. In this case, the mediator(s) return to the requesting node the "location" of the node which stores the datum.

3) Remote hit (RH): the requested datum is cached by a node and this node has at least one mediator residing along the path from the requesting node to the Data Center.
4) Global hit (GH): the requested datum is acquired from the Data Center.

C. The cache replacement component

A cache replacement policy is required when a sensor attempts to cache an object, but the cache is full. In replacement operation one or more objects are evicted out of the local cache (due to providing sufficient space) and new one is cached. The $EBCCoCa$ protocol employs the a value based policy. Initially each sensor node removes the object that it has cached on behalf of some other node. Each cached item is characterized either as O (i.e., own) or H (i.e., hosted). In case of a local hit, then its state switches to O.

If the available cache space is still smaller than the requested object, then for each cached object i the following function is calculated: $cost(i) = \frac{Lat_i * Size_i}{TTL_i * AR_i}$. The objects with the greatest values are those that are removed from the cache. When a sensor gets a reply message, it calculates the incurred latency (Lat). The smaller the latency of an object is, the more likely to remain to cache. The access rate (AR) indicates the frequency that a cached item is being requested, while time-to-live (TTL) value determines the validity of a cached object. An object remains in cache when AR and TTL are big. Finally, the bigger the size of an object is, the more likely to be removed from the cache.

During the final step, cache node notifies the mediators about the candidate victim. If the object is also cached by another node in the neighborhood, then mediators broadcast a delete message and the object is evicted out of the local cache. Otherwise, each mediator send a message that contains the node that has the largest residual energy and enough space to cache the object. In this case, the node purges the object and send it to the node with the largest residual energy. Finally, mediators update their cached metadata about the new state.

IV. Performance evaluation

We evaluated the performance of the $EBCCoCa$ protocol through simulation experiments. We conducted a large number of experiments with various parameters, and compared the performance of $EBCCoCa$ to the state-of-the-art cooperative caching policy for WSNs, namely NICoCa [2].[1] For the interest of space, we present here only the most important, representative experiments and respective results.

[1] In [2], the NICoCa protocol was compared against the Hybrid caching scheme [1], for many data/request distributions and many network topologies, and NICoCa proved superior in all cases.

As stated before $NICoCa$ protocol constitutes a special case of $EBCCoCa$ protocol. This happens when parameter a equals 0.0.

A. Simulation model

Both protocols have been implemented and evaluated with the J-Sim wireless network simulator [21]. In our simulations, the AODV [22] routing protocol is deployed to route the data traffic in the wireless sensor network. We use IEEE 802.11 as the MAC protocol and the free space model as the radio propagation model. The wireless bandwidth is 2 Mbps. The radio characteristics used in our simulations are summarized in Table II.

Operation	Energy Dissipated
Transmitter/Receiver Electronics	$E_{elec} = 50nJ/bit$
Transmit Amplifier if $d_{toBS} \leq d_0$	$e_{fs} = 10pJ/bit/m^2$
Transmit Amplifier if $d_{toBS} \geq d_0$	$e_{mp} = 0.0013pJ/bit/m^4$
Data Aggregation	$E_{DA} = 5nJ/bit/signal$

Table II
RADIO CHARACTERISTICS.

The protocols has been tested for a variety of sensor network topologies, to simulate sensor networks with varying values of node degree, from 4 to 10. Thus, we are able to simulate both sparse and dense sensor deployments. We experiment with various sizes of the sensor networks; we present here the results for two cases, namely when the number of sensors is 100 and 500. The distribution of the sizes of the data items is uniform between 1KB and 10KB.

The generated network topology consists of many square grid units where one or more nodes are placed. The number of square grid units depends on the number of nodes and the node degree. The topologies are generated as follows: the location of each of the n sensor nodes is uniformly distributed between the point $(x = 0, y = 0)$ and the point $(x = 500, y = 500)$. The average degree d is computed by sorting all $n*(n-1)/2$ edges in the network by their length, in increasing order. The grid unit size corresponding to the value of d is equal to $\sqrt{2}$ times the length of the edge at position $n * d/2$ in the sorted sequence. Two sensor nodes are neighbors if they placed in the same grid or in adjacent grids. The simulation area is assumed of size $500m \times 500m$ and is divided into equal sized square grid units. Beginning with the lower grid unit, the units are named as $1, 2, \ldots$, in a column-wise fashion.

The client query model is similar to what have been used in previous studies [2], [1]. The access pattern of sensor nodes follow the well-known Zipfian distribution with parameter θ (for $\theta = 0.0$, we get a uniform access pattern; for values of θ around 1, the access pattern is highly skewed). The sensors residing in neighboring grids (25 grids with size $100m \times 100m$) have the same access pattern. We conducted experiments with varying θ values between 0.0 and 1.0. Here, we present the results for $\theta = 0.8$. Each sensor node generates 200 requests.

Similar to [2], [1], two Data Centers are placed at opposite corners of the simulation area. Data Center 1 is placed at point $(x = 0, y = 0)$ and Data Center 2 is placed at point $(x = 500, y = 500)$. There are $N/2$ data items in each data center. Data items with even ids are stored at Data Center 1 and data items with odd ids are stored at Data Center 2. We assumed that data items are not updated. The system parameters are listed in Table III.

Parameter	Default value	Range
# items (N)	1000	
# requests per node	200	
S_{min} (KB)	1	
S_{max} (KB)	10	
# nodes (n)	500	100–1000
Bandwidth (Mbps)	2	
Waiting interval (t_w)	10 sec	
Cache size (KB)	800	200 to 800
Zipfian skewness (θ)	0.8	0.0 to 1.0

Table III
SIMULATION PARAMETERS.

The measured quantities include the number of hits (local, remote and global), the network lifetime of the sensor network, the average latency for getting the requested data and the number of packets dropped. It is evident that large number of remote hits proves the effectiveness of cooperation in reducing the number of global hits. A large number of local hits does not imply an effective cooperative caching policy, unless it is accompanied by small number of global hits, since the cost of global hits vanishes the benefits of local hits.

B. Evaluation

We performed a large number of experiments varying the size of the sensor networks (in terms of the number of its sensor nodes), varying the access profile of the sensor nodes, and the cache size relative to the aggregate size of all data items. In particular, we performed experiments for cache size equal to 1% and to 5% of the aggregated size of all distinct data, for access pattern with θ starting from 0.0 (uniform access pattern) to 1.0 (highly skewed access pattern), and for average sensor node degree equal to $4, 7$ (sparse and moderate dense sensor networks, respectively) and 10 (dense sensor network). For the interest of space we present a small subset of the results obtained (shown in Figures 2 – 6). In these experiments we evaluate the impact of the a parameter and the amount of the sensor storage on the number of hits and latency. Additionally, we evaluate the impact of the a parameter and the number of sensors in network lifetime. $NICoCa$ protocol corresponds to $EBCCoCa$ protocol with parameter a equals 0.0. We didn't depict the energy consumption during the exchange of "HELLO" messages, since the protocol requires only four bytes to implement the exchange and thus the overhead to the overall performance is negligible.

As expected, both cooperative caching schemes exhibit better performance for all metrics with increasing cache size; therefore caching is indeed a useful technique, irrespectively of the network topology. As the cache space in each sensor increases toward an infinite cache, that could ideally accommodate all items, the actual performance gap (latency and travelling messages) between the protocols diminishes. The second generic observation is that the proposed EBCCoCa protocol prolongs network lifetime for large values of parameter a while it achieves a better performance in dense networks in terms of latency and remote cache hits.

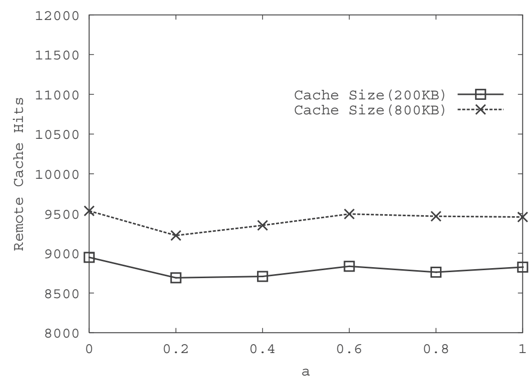

Figure 2. Impact of a on remote cache hits in a sparse ($d = 4$) WSN with $\theta = 0.8$.

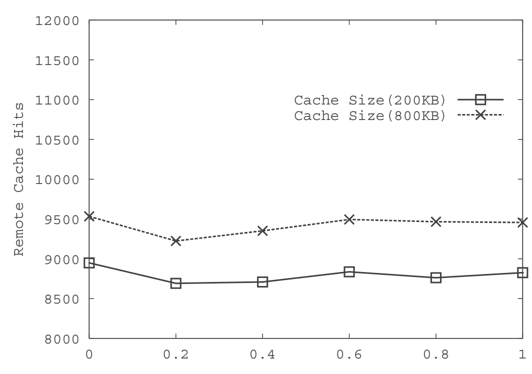

Figure 3. Impact of a on remote cache hits in a dense ($d = 10$) WSN with $\theta = 0.8$.

It is interesting to note that both protocols achieve almost the same number of remote cache hits in sparse ($d = 4$) WSNs. This is because of small number of nodes participating in local network. Both protocols elect almost the same number of mediators and in many cases the same nodes as mediators. However, in dense ($d = 10$) WSNs EBCCoCa performs slightly better than NiCoCa for a greater than 0.5. This is due to the selection of mediator nodes. For large values of parameter a EBCCoCa protocol is affected more by shortest energy path. The shortest energy path contains more nodes than shortest path without weights on edges. Thus, more nodes become mediators and a great number of requests are satisfied by mediators without reaching in Data Centers.

The performance of the protocols with respect to the average latency incurred for varying cache sizes and a values for both sparse and dense sensor network deployments are depicted in Figures 4 and 5. The dominant observation is that caching is more beneficial for sparse networks, since it can balance the (relatively) longer paths to the data that increase the latency. The second observation is the superiority of EBCCoCa protocol in dense WSNs for large values of parameter a. The relative results follow the same trends that we observed in the previous experiment; there is a close connection between the number of remote cache hits and latency, since the more the number of remote hits, the less the latency in accessing the data.

In general, the performance of each protocol gets better in dense sensor networks, since we constrain more sensor nodes to be dispersed in the same geographical region, thus creating more replicas of the same data and providing more alternative paths to the data. This better performance is reflected to the access latency, hit ratio and message overhead.

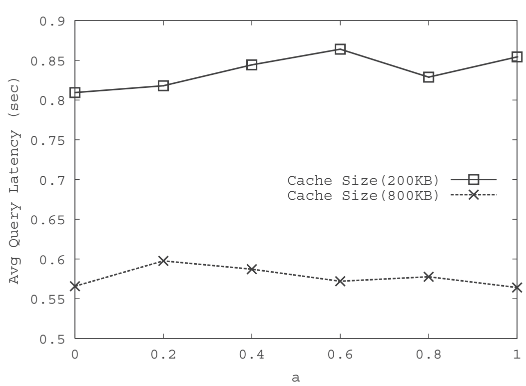

Figure 4. Impact of a on latency in a sparse ($d = 4$) WSN with $\theta = 0.8$.

Finally, we evaluated the performance of the algorithms with respect to the energy consumption for varying a values and cache sizes for both sparse and dense sensor network deployments. The results are depicted in Figures 6 and 7. The prolongation of the network lifetime is the major objective of EBCCoCa protocol. Due to EBC index that comprehends the remaining energy of nodes in calculation of shortest paths, the mediators will not be always the same.

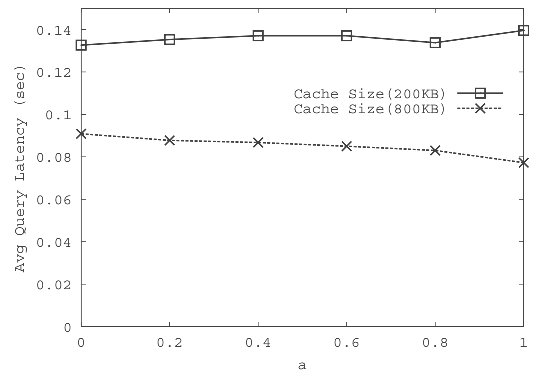

Figure 5. Impact of a on latency in a dense ($d = 10$) WSN with $\theta = 0.8$.

Thus, EBCCoCa protocol achieves a better energy balance among sensors in WSNs. The lifetime of dense networks is prolonged from 15% to 30% by EBCCoCa protocol.

In summary, for dense network topologies $EBCCoCa$ is able to better capture the energy "significant" nodes in the sensor network. Thus, $EBCCoCa$ can prevent the network partition due to energy dissipation of mediator nodes, while minimizes latency.

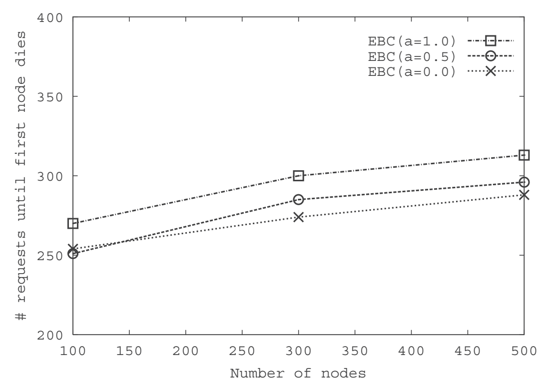

Figure 6. Performance of EBCCoCa in a sparse ($d = 4$) WSN with $\theta = 0.8$. Network lifetime (first node death).

V. Conclusions

The majority of applications based on wireless sensor networks depends mainly on the ability of the underlying protocols to scale to large number of sensors, to conserve energy and provide answers in short latency. Cooperative data caching has been proposed as an effective and efficient technique to achieve these goals concurrently. The design of these protocols depends mainly on the selection of the sensor nodes which will take special roles in running the caching and request forwarding decisions. The article introduced a new centrality metric that aid to select such nodes, and proposed a new energy efficient cooperative caching protocol

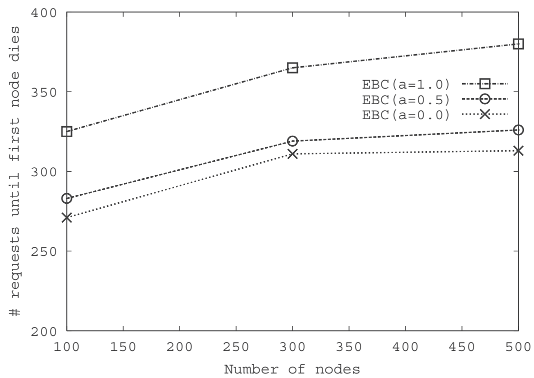

Figure 7. Performance of EBCCoCa in a dense ($d = 10$) WSN with $\theta = 0.8$. Network lifetime (first node death).

(EBCCoCa), which proved superior to the state-of-the-art competing protocol.

References

[1] L. Yin and G. Cao, "Supporting cooperative caching in ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 5, no. 1, pp. 77–89, 2006.

[2] N. Dimokas, D. Katsaros, and Y. Manolopoulos, "Cooperative caching in wireless multimedia sensor networks," *ACM Mobile Networks and Applications*, vol. 13, no. 3–4, pp. 337–356, 2008.

[3] T. P. Sharma, R. C. Joshi, and M. Misra, "Cooperative caching for homogeneous wireless sensor networks," *International Journal of Communication Networks and Distributed Systems*, vol. 2, no. 4, pp. 424–451, 2012.

[4] O. Younis, M. Krunz, and S. Ramasubramanian, "Node clustering in wireless sensor networks: Recent developments and deployment challenges," *IEEE Network*, vol. 20, no. 3, pp. 20–25, May/Jun. 2006.

[5] S. Wasserman and K. Faust, *Social network analysis: methods and applications. Structural analysis in the social sciences.* Cambridge University Press, 1994.

[6] S. Abdallah, "Generalizing unweighted network measures to capture the focus in interactions," *Social Netw. Analys. Mining*, pp. 255–269, 2011.

[7] W. Hwang, T. Kim, M. Ramanathan, and A. Zhang, "Bridging centrality: graph mining from element level to group level," in *Proceedings of the ACM Special Interest Group on Knowledge Discovery and Data Mining*, 2008, pp. 336–344.

[8] D. Katsaros, N. Dimokas, and L. Tassiulas, "Social network analysis concepts in the design of wireless ad hoc networks protocols," *IEEE Network Magazine*, vol. 24, no. 6, pp. 23–29, 2010.

[9] M. Taghizadeh, K. Micinski, and S. Biswas, "Distributed cooperative caching in social wireless networks," *IEEE Transactions on Mobile Computing*, vol. 1, no. 4, pp. 271–286, 2012.

[10] M. Saravanan, G. Prasad, S. Karishma, and D. Suganthi, "Analyzing and labeling telecom communities using structural properties," *Social Network Analysis and Mining*, vol. 1, no. 4, pp. 271–286, 2011.

[11] F. Gilbert, P. Simonetto, F. Zaidi, F. Jourdan, and R. Bourqui, "Communities and hierarchical structures in dynamic social networks: analysis and visualization," *Social Network Analysis and Mining*, vol. 1, no. 1, pp. 83–95, 2011.

[12] P. Bonacich, "Power and centrality: a family of measures," *American Journal of Sociology*, vol. 92, no. 5, pp. 1170–1182, 1987.

[13] L. C. Freeman, "Centrality in social networks: Conceptual clarification," *Social Networks*, vol. 1, no. 3, pp. 215–239, 1979.

[14] W. Li, E. Chan, and D. Chen, "Energy-efficient cache replacement policies for cooperative caching in mobile ad hoc network," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2007, pp. 3349–3354.

[15] L. Fan, P. Cao, and A. Z. Almeida, J. M. Broder, "Summary cache: A scalable wide-area Web cache sharing protocol," *IEEE/ACM Transactions on Networking*, vol. 8, no. 3, pp. 281–293, 2000.

[16] S. Annapureddy, M. J. Freedman, and D. Mazières, "Shark: Scaling file servers via cooperative caching," in *Proceedings of the USENIX Symposium on Networked Systems Design & Implementation (NSDI)*, 2005, pp. 129–142.

[17] P. Nuggehalli, V. Srinivasan, and C.-F. Chiasserini, "Energy-efficient caching strategies in ad hoc wireless networks," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*, 2003, pp. 25–34.

[18] B. Tang, H. Gupta, and S. R. Das, "Benefit-based data caching in ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 7, no. 3, pp. 289–304, 2008.

[19] W. Gao, Q. Li, B. Zhao, and G. Cao, "Multicasting in delay tolerant networks: a social network perspective," in *Proceedings of the ACM International Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*, 2009, pp. 299–308.

[20] R. Gandhi and S. Parthasarathy, "Fast distributed well connected dominating sets for ad hoc networks," Computer Science Department, University of Maryland at College Park, Tech. Rep. CS-TR-4559, 2004.

[21] A. Sobeih, J. C. Hou, L.-C. Kung, N. Li, H. Zhang, W.-P. Chen, H.-Y. Tyan, and H. Lim, "J-Sim: A simulation and emulation environment for wireless sensor networks," *IEEE Wireless Communications magazine*, vol. 13, no. 4, pp. 104–119, 2006.

[22] C. E. Perkins and E. Royer, "Ad hoc On-demand Distance Vector routing," in *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, 1999, pp. 90–100.