Cooperative Caching in Wireless Multimedia Sensor Networks*

Nikos Dimokas
Informatics Department,
Aristotle University,
Thessaloniki
Greece
dimokas@delab.csd.auth.gr

Dimitrios Katsaros
Computer & Communication
Engineering Department,
University of Thessaly,
Volos, Greece
dkatsaro@csd.auth.gr

Yannis Manolopoulos
Informatics Department,
Aristotle University,
Thessaloniki
Greece
manolopo@csd.auth.gr

ABSTRACT

The recent advances in miniaturization and the creation of low-power circuits, combined with small-sized batteries have made the development of wireless sensor networks a working reality. Lately, the production of cheap CMOS cameras and microphones gave birth to what is called *Wireless Multimedia Sensor Networks (WMSNs)*. WMSNs will boost the capabilities of current wireless sensor networks, and will fuel several novel applications. WMSNs introduce several research challenges, related to the delivery of application-level Quality-of-Service (e.g., latency minimization).

To address this goal in an environment with extreme resource constraints, with variable channel capacity and with requirements for multimedia in-network processing, the efficient and effective caching of multimedia data, exploiting the cooperation among sensor nodes is vital. This paper presents for the first time a cooperative caching solution particularly suitable for WMSNs. The proposed solution is evaluated extensively in an advanced simulation environment, and it is compared to the state-of-the-art cooperative caching algorithm for mobile ad hoc networks.

1. INTRODUCTION

Wireless Sensor Networks [2] (WSNs) have emerged during the last years due to the advances in low-power hardware design and the development of appropriate software, that enabled the creation of tiny devices which are able to compute, control and communicate with each other. A wireless sensor network consists of wirelessly interconnected devices that can interact with their environment by controlling and sensing "physical" parameters. The constantly growing interest in WSNs can be attributed to the many new exciting applications that were born as a result of the deployment of large-scale WSNs. Such applications range from disaster relief, to environment control and biodiversity mapping, to machine surveillance, to intelligent building, to precision agriculture, and to pervasive health applications.

More recently, the production of cheap CMOS cameras and microphones, which can acquire rich media content from the environment, created a new wave into the evolution of wireless sensor networks. For instance, the Cyclops imaging module [16] is a light-weight imaging module which can be adapted to MICA2[1] or MICAz sensor nodes. Thus, a new class of WSNs came to the scene, the *Wireless Multimedia Sensor Networks (WMSNs)* [1]. These sensor networks, apart from boosting the existing application of WSNs, will create new applications: a) multimedia surveillance sensor networks which will be composed of miniature video cameras will be able to communicate, to process and store data relevant to crimes and terrorist attacks; b) traffic avoidance and control systems will monitor car traffic and offer routing advices to prevent congestion; c) industrial process control will be realized by WMSNs that will offer time-critical information related to imaging, pressure, etc.

It is emphasized in [1] that these applications demand a reconsideration of the computation-communication paradigm of traditional WSNs, which has mainly focused on reducing the energy consumption, targeting to prolong the longevity of the sensor network. Though, the applications implemented by WMSNs have a second goal, as important as the energy consumption, to be pursued; this goal is the delivery of application-level quality of service (QoS) and the mapping of this requirement to network layer metrics, like latency. This goal has (almost) been ignored in mainstream research efforts on traditional WSNs. The goal of Internet QoS in multimedia content delivery has been pursued in architectures like Diffserv and Intserv, but these protocols and techniques do not face the severe constraints and hostile environment of WSNs.

In particular, WMSNs are mainly characterized by:

- Resource constraints: sensor nodes are battery-, memory- and processing-starving devices.
- Variable channel capacity: the multi-hop nature of WMSNs, which operate in a store-and-forward fashion because of the absence of base stations, implies that the capacity of each wireless link depends on the interference level among nodes, which is aggravated by the broadcasting operations.
- Multimedia in-network processing: sensor nodes are required to store rich media, e.g., image, video, needed

*Research supported by a *ΓΓΕΤ* grant in the context of the project "Data Management in Mobile Ad Hoc Networks" funded by *ΠΥΘΑΓΟΡΑΣ ΙΙ* national research program.

MOBIMEDIA 2007, August 27-29, Nafpaktos, Greece

DOI 10.4108/ICST.MOBIMEDIA2007.1845

[1]http://www.xbow.com

for their running applications, and also to retrieve such media from remote sensor nodes with short latency.

Under these restrictions/requirements, the goal of achieving application-level QoS in WMSNs becomes a very challenging task. There could be several ways to attack parts of this problem, e.g., channel-adaptive streaming [9], joint source-channel coding [7]. Though, none of them can provide solutions to all of the three aforementioned issues. In this paper, we investigate the solution of *cooperative caching multimedia content* in sensor nodes to address all three characteristics. In cooperative caching, multiple sensor nodes share and coordinate cache data to cut communication cost and exploit the aggregate cache space of cooperating sensors. The plain assumption we make, is that each sensor node has a moderate local storage capacity associated with it, i.e., a flash memory. Although, there exist flash memories with several gigabytes storage capacity, e.g., the NAND flash [11] and the trend is that they become cheaper, larger and more common, we do not assume extreme storage capabilities, so as to capture a broader field of applications and architectures.

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. Additionally, caching can be very effective in reducing the need for network-wide transmissions, thus reducing the interference and overcoming the variable channel conditions. Finally, it can speed-up the multimedia in-network processing, because, as it is emphasized in [1], the processing and delivery of multimedia content are not independent and their interaction has a major impact on the levels of QoS that can be delivered.

This paper investigates for the first time the technique of caching in the context of WMSNs. The need for effective and intelligent caching policies in sensor networks has been pointed out several times [6, 13] in the very recent literature, but no appropriate sophisticated policies have been proposed, although there are quite a lot of caching protocols in other fields (see relevant work in Section 2). This article proposes a novel and high-performance cooperative caching protocol, the *NICoCa* protocol named after the words *Node Importance-based Cooperative Caching*, and compares it with the state-of-the-art cooperative caching policy for mobile ad hoc networks, which is the "closer" competitor. Using the J-Sim simulation environment [18], we perform an experimental evaluation of the two methods, which attests that the proposed *NICoCa* cooperative caching policy prevails over its competitor.

The rest of this article is organized as follows: in Section 2 we review the relevant work and record the contributions of the article. In Section 3 we present the details of the *NICoCa* protocol, and in Section 4 we present the results of the performance evaluation of the methods; finally, Section 5 concludes the paper.

2. RELEVANT WORK

The technique of caching has been widely investigated in the context of operating systems and databases and it is still a very fertile research area [12]. Similarly, caching on the Web has been thoroughly investigated for cooperative and for non-cooperative architectures [14]. These environments though do not face the extreme resource constraints of WMSNs, and thus the proposed caching protocols are not appropriate for the case of WMSNs.

There are also a number of caching approaches for wireless broadcast cellular networks [10]. These policies assume more powerful nodes than the sensor nodes, and one-hop communication with resource-rich base stations, which serve the needed data.

The most relevant research works to our protocol are the cooperative caching protocols which have been developed for mobile ad hoc networks (MANETs). The main motive for the development of these protocols is the mobility of the nodes, and thus they all strive to model it or exploit it. For instance, the GroCoca [5] attempts to organize nodes into groups based on their data request pattern and their mobility pattern. The ECOR [17], the Zone Co-operative [3] and the Cluster Cooperative [4] protocols attempt to form clusters of nodes based either in geographical proximity or utilizing widely known node clustering algorithms for MANETs. The only protocols that deviated from such approaches and tried to exploit both data and node locality in an homogeneous manner are described in [20] and are the following: *CachePath*, *CacheData*, and *HybridCache*. In CacheData, intermediate nodes (the nodes between a requesting node and the node which holds the requested data) cache the data to serve future requests instead of fetching data from the "Data Center". In CachePath, mobile nodes cache the data path and use it to redirect future requests to the nearby node which has the data instead of the faraway Data Center. A high performance amalgamation of them is the HybridCache, which can further improve the performance by taking advantage of CacheData and CachePath while avoiding their weaknesses.

The only works on caching in wireless sensor networks concern the placement of caches [15, 19] and thus they are not strictly relevant to our considered problem.

2.1 Motivation and contributions

The protocols proposed so far for cooperative caching in MANETs present various limitations. Those protocols which first perform a clustering of the network and then exploit this clustering (actually, the cluster-heads, CH) in order to coordinate the caching decisions, inherit the shortcomings of any bad CH selection. For instance, in [3, 4], the nodes which form the cluster are assumed to reside within the same communication range, i.e., they are with on-hop distance from the other nodes of the cluster. Additionally, the nodes do not cache the data originating from an one-hop neighbor. Thus, CHs which do not reside in a significant part of data routes can not serve efficiently their cluster members, because they do not have fast access (short latency) to requested data. The cooperation zone which is formed in [17] by selecting an optimal radius, implies a large communication overhead, because every node within that radius must send/receive any changes to the caches of the other nodes within the radius. Finally, the HybridCache policy is tightly coupled to the underlying routing protocol, and thus if a node does not reside in the route selected by the routing protocol can not cache the data/path, or conversely, can not serve the request even if it holds the requested data.

Motivated by the weaknesses of the current cooperative protocols and the unique requirements of the WMSNs, which are mainly static and not mobile, we propose a cooperative caching policy which is based on the idea of exploiting the sensor network topology, so as to discover which nodes are more important than the others, in terms of their position in the network and/or in terms of residual energy. Incorporating both factors into the design of the caching policy we ensure both network longevity and short latency in multimedia data retrieval. In summary, the article's contributions are the following:

- Definition of a metric for estimating the importance of a sensor node in the network topology, which will imply short latency in retrieval.
- Description of a cooperative caching protocol which takes into account the residual energy of the sensors.
- Development of algorithms for discovering the requested multimedia data, and maintaining the caches.
- Performance evaluation of the protocol and comparison with the state-of-the-art cooperative caching protocol for MANETs, using an established simulation package (J-Sim).

3. THE *NICoCa* COOPERATIVE CACHING PROTOCOL FOR WMSNS

One of the main parts of the proposed protocol is the estimation of the importance of sensors relative to the network topology. The intuition is that if we discover those nodes which 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.

3.1 Measuring sensor node importance

A wireless multimedia 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. 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., $N_1(v) = \{u : (v, u) \in E\}$. The set of two-hop nodes of node v, i.e., the nodes which are the neighbors of node v's neighbors except for the nodes that are the neighbors 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 $N_{12}(v)$.

Definition 1 (Local network view of node v). *The local network view, denoted as LN_v, of a graph $G(V, E)$ w.r.t. a node $v \in V$ is the induced subgraph of G associated with the set of vertices in $N_{12}(v)$.*

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. Then, we define the *node importance* index $\mathcal{NI}(v)$ of a vertex v as:

Definition 2. *The $\mathcal{NI}(v)$ of a vertex v is equal to:*

$$\mathcal{NI}(v) = \sum_{u \neq v \neq w \in V} \frac{\sigma_{uw}(v)}{\sigma_{uw}}. \tag{1}$$

Large values for the $\mathcal{NI}$ index of a node v indicate that this node v can reach others on relatively short paths, or that the node v lies on considerable fractions of shortest paths connecting others. Figure 1 illustrates this metric.

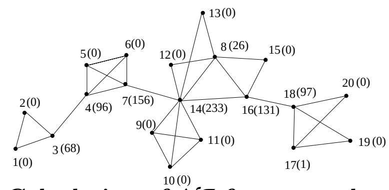

Figure 1: Calculation of $\mathcal{NI}$ for a sample graph. The numbers in parentheses denote the $\mathcal{NI}$ index of the respective node considering the whole WMSN topology.

Apparently, when estimating the $\mathcal{NI}$ index for each sensor node using the whole network topology we obtain a very informative picture of which nodes reside in a large number of shortest paths between other nodes. Fortunately, it is very easy to confirm that, even when we calculate the $\mathcal{NI}$ indexes of the nodes taking into account only their k-hop ($k = 2$ or 3) neighborhood, the picture about the relative importance of the nodes remains very accurate.

3.2 Housekeeping information in the *NICoCa* protocol

W.l.o.g. and adopting the model presented in [20], we assume that the ultimate source of multimedia data is a *Data Center*. Firstly, it is assumed that each node is aware of its 2-hop neighborhood. This information is obtained through periodic exchange of "beacon" 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 [8]. Then, every node calculates the $\mathcal{NI}$ 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 $\mathcal{NI}$ 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.

The sending of requests for data is carried out by an ordinary sensor (or ad hoc) routing protocol, e.g., AODV. A node always caches a datum which has requested for. A node is aware of its remaining energy and of the free space in its cache. Each sensor node stores the following metadata related to a cached multimedia item:

- the dataID, and the actual multimedia data item,
- the data size (s_i),
- a TTL interval (Time-To-Live),
- 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 node acquires the multimedia datum it has requested for, then it caches it and broadcasts a small index packet containing the dataID and the associated TTL, its remaining energy and its free cache space. The mediator nodes which are also 1-hop neighbors of this node store this broadcasted information. Please, notice here that this set of mediator nodes includes the mediators that the broadcasting node has selected, and also any other mediators which have been selected by nearby nodes. In summary, every mediator node stores the remaining energy and the free cache space for each one of its 1-hop neighbors, and for each dataID that has heard through the broadcasting operation, the TTL of this datum and the nodes that have cached this datum.

3.3 The cache discovery component protocol

When a sensor node issues a request for a multimedia 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 *NICoCa* 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.

3.4 The cache replacement component protocol

Even though the cache capacity of individual sensors may be in the order of gigabytes (e.g., NAND flash) the development of an effective and intelligent replacement policy is mandatory to cope with the overwhelming size of multimedia data generated in WMSNs. The *NICoCa* protocol employs the following four-step policy:

STEP 1. In case of necessity, before purging from cache any other data, each sensor node first purges the data 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 required, execute Step 2.

STEP 2. Calculate the following function for each cached datum i: $cost(i) = \frac{s_i}{TTL_i} * \frac{now - t_{K-th\ access}}{K}$, where $t_{K-th\ access}$ is the timestamp of the K-th access. The candidate cache victim is the item which incurs the largest cost.

STEP 3. Inform the mediators about the candidate victim. If it is cached by some mediator, then this information returns back to the node and purges the datum. If the datum is not cached by some mediator(s), then it is forwarded to the node with the largest residual energy and the datum is purged from the cache of the original node. In any case, the mediators update their cached metadata about the new state.

STEP 4. The node which caches this purged datum, informs the mediators with the usual broadcasting procedure.

4. PERFORMANCE EVALUATION

We evaluated the performance of the *NICoCa* protocol through simulation experiments and compared its performance to that of a state-of-the-art cooperative caching policy for MANETs, namely *HybridCache* [20].

4.1 Simulation model

We have developed a simulation model based on the J-Sim simulator [18]. In our simulations, the AODV 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 assumed to be 2 Mbps.

We tested the protocols for a variety of sensor network topologies, to simulate sensor networks with varying levels of node degree, from 4 to 10. Each topology consists of many square grid units where one or more nodes are placed. 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 $(x = 100,\ y = 100)$. 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 are placed in the same grid or in adjacent grids.

The simulation area is $100x100\ m^2$. The client query model is similar to what have been used in previous studies [20]. Each sensor node generates read-only queries. After a query is sent out, if the sensor node does not receive the data item, it waits for an interval (t_w) before sending a new query. The access pattern of sensor nodes is: a) location-independent, that is, sensor nodes decide independently the data of interest; each sensor node generates accesses to the data following the uniform distribution, and b) Zipfian with $\theta = 0.8$, where groups of nodes residing in neighboring grids (25 grids with size $20x20\ m^2$) have the same access pattern.

Similar to [20], 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

Parameter	Default value	Range
# items (N)	1000	100–10000
S_{min} (KB)	1	
S_{max} (KB)	10	
S_{min} (MB)	1	
S_{max} (MB)	5	
Zipfian θ	0.8	
# nodes (n)	100	50–1000
Bandwidth (Mbps)	2	0.5–2
Waiting interval (t_w)	10 sec for items with KB size 100 sec for items with KB size	
Client cache size (KB)	800	200 to 1200
Client cache size (MB)	125	25 to 250

Table 1: Simulation parameters.

($x = 100,\ y = 100$). Each data center contains $N/2$ data items. The size of each data item is uniformly distributed between s_{min} and s_{max}. We assumed that data items are not updated. The data centers serve the queries on a FCFS basis. System parameters are listed in Table 1.

The measured quantities include the number of hits (local, remote and global), the residual energy level of the sensor nodes, the average latency for getting the requested data, the number of packets dropped. For the interest of space, we present only the results that concern the number of hits as a representative of latency, of collisions and of energy consumption. It is evident that a small number of global hits implies less network congestion, and thus fewer collisions and packet drops. Moreover, 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.

4.2 Evaluation

We performed a large number of experiments varying the size of the sensor network (in terms of both its geographic area and the number of its sensor nodes), varying the number of multimedia items, varying the wireless link bandwidth, the access profile of the sensor nodes, and so on. In the interest of space we present a small subset of the results obtained. In these experiments we evaluate the impact of the amount of the sensor flash storage on the number of hits. We examine the case of small multimedia files (their size is in order of kilobytes), and the case of large multimedia files (their size is in order of megabytes). In any case, the local sensor flash storage ranges from 1%, to 5% to 10% of the total size of all distinct multimedia data.

Figures 2, 3, and 4 present the performance of the algorithms for uniform access pattern and for multimedia data whose size is in the range of a few megabytes, e.g., video data. The first generic observations is that for all network topologies (sparse, dense and very dense), *NICoCa* achieves more remote hits and less global hits than *HyrbidCache*. This performance gap widens in favor of *NICoCa* as we move from sparse to denser WMSNs. It is striking that for very dense sensor deployments, *NICoCa* achieves double the remote hits of *HyrbidCache* and only half of its global hits.

Examining the local hits, we observe that for sparse WMSNs *HyrbidCache* achieves slightly more local hits than does *NICoCa*, but this gap vanishes completely when moving to denser network. Besides, this small gain of *HyrbidCache* for sparse topologies is not advantageous at all, since it incurs global hits as many as twice the number of its local hits.

Figures 5, 6, and 7 present the performance of the algorithms for Zipfian (skewed) access pattern with $\theta = 0.8$ and for multimedia data whose size is in the range of a few

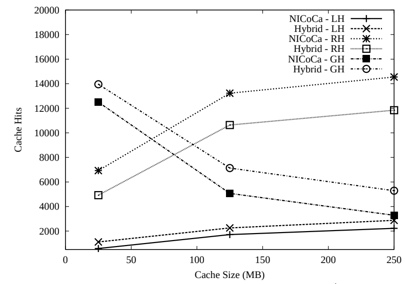

Figure 2: Impact of cache size on hits (large multimedia files and uniform access) in a sparse WMSN ($d = 4$).

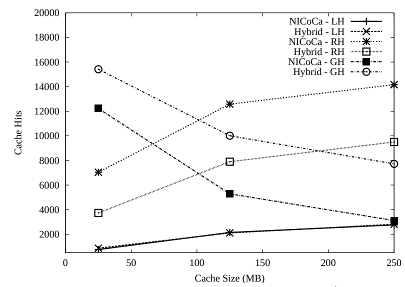

Figure 3: Impact of cache size on hits (large multimedia files and uniform access) in a dense WMSN ($d = 7$).

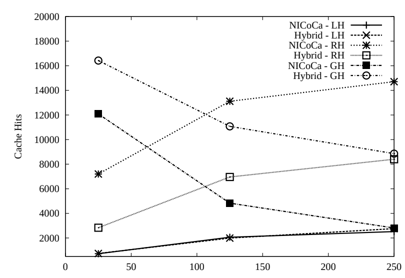

Figure 4: Impact of cache size on hits (large multimedia files and uniform access) in a very dense WMSN ($d = 10$).

kilobytes. The general trend is that *NICoCa* achieves significantly smaller number of global hits and larger number of remote hits than *HyrbidCache* does.

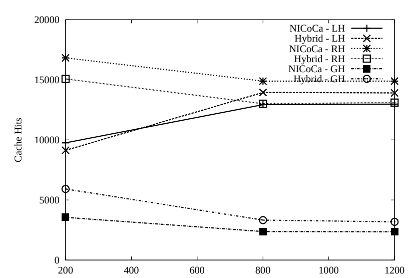

Figure 5: Impact of cache size on hits (small multimedia files and Zipfian access) in a sparse WMSN ($d = 4$).

This trend becomes more evident in denser deployments and it is the case that for very dense WMSNs, the requests that reach the Data Center in the *NICoCa* protocol are less than half those of *HyrbidCache*! It is also important to note that *NICoCa*'s global hits do not vary significantly with varying network topologies and varying local sensor storage. When the topology becomes denser, *NICoCa*'s remote hits increase, which means that it is able to exploit the co-

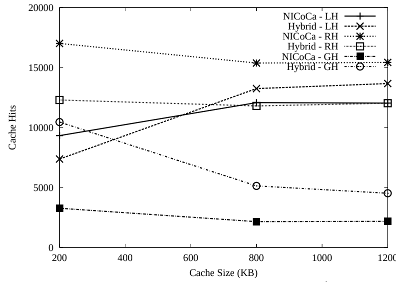

Figure 6: Impact of cache size on hits (small multimedia files and Zipfian access) in a dense WMSN ($d = 7$).

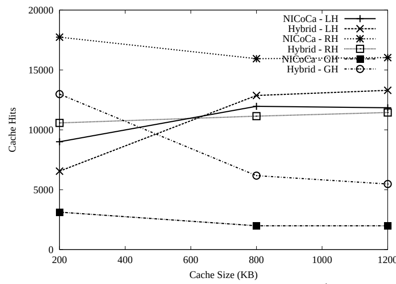

Figure 7: Impact of cache size on hits (small multimedia files and Zipfian access) in a very dense WMSN ($d = 10$).

operation in a better way. On the contrary the global hits of *HyrbidCache* are severely affected by the topology and the cache size. For cache capacity equal to 1% of the total data size, *HyrbidCache*'s global hits increase at a pace of 50%! The results for Zipfian access pattern on megabyte-sized data was even more impressively in favor of *NICoCa*.

5. SUMMARY AND CONCLUSIONS

The recent advances in miniaturization, the creation of low-power circuits, and the development of cheap CMOS cameras and microphones gave birth to *Wireless Multimedia Sensor Networks*. WMSNs are expected to fuel many new applications and boost the already existing ones. The unique features of WMSNs call for protocol designs that will provide application-level QoS, an issue that has largely been ignored in traditional WSNs. Taking a first step toward this goal, this article develops a cooperative caching protocol, the *NICoCa* protocol, suitable deployment in WMSNs. The protocol "detects" which sensor nodes are most "central" in the network neighborhoods and gives to them the role of mediator in order to coordinate the caching decisions. The proposed protocol is evaluated with J-Sim and its performance is compared to that of a state-of-the-art cooperative caching protocol for MANETs. The obtained results attest the superiority of the proposed protocol which is able to reduce the global hits at an average percentage of 50% and increase the remote hits due to the effective sensor cooperation at an average percentage of 40%. The performance of the protocol is particularly high for the delivery of large multimedia data.

6. REFERENCES

[1] I. Akyildiz, T. Melodia, and K. R. Chowdhury. A survey of wireless multimedia sensor networks. *Computer Networks*, 51:921–960, 2007.

[2] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey of wireless sensor networks. *IEEE Communications magazine*, 40(8), 2002.

[3] N. Chand, R. C. R. C. Joshi, and M. Misra. Cooperative caching strategy in mobile ad hoc networks based on clusters. *Wireless Personal Communications*, 2006.

[4] N. Chand, R. C. R. C. Joshi, and M. Misra. A zone co-operation approach for efficient caching in mobile ad hoc networks. *International Journal of Communication Systems*, 19:1009–1028, 2006.

[5] C.-Y. Chow, H. V. Leong, and A. T. S. Chan. GroCoca: Group-based peer-to-peer cooperative caching in mobile environment. *IEEE Journal on Selected Areas in Communications*, 25(1):179–191, 2007.

[6] Y. Diao, D. Ganesan, G. Mathur, and P. Shenoy. Rethinking data management for storage-centric sensor networks. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, pages 22–31, 2007.

[7] Y. Eisenberg, C. E. Luna, T. N. Pappas, R. Berry, and A. K. Katsaggelos. Joint source coding and transmission power management for energy efficient wireless video communications. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(6):411–424, 2002.

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

[9] B. Girod, M. Kalman, Y. J. Liang, and R. Zhang. Advances in channel-adaptive video streaming. *Wireless Communications and Mobile Computing*, 2(6):573–584, 2002.

[10] D. Katsaros and Y. Manolopoulos. Web caching in broadcast mobile wireless environments. *IEEE Internet Computing*, 8(3):37–45, 2004.

[11] G. Mathur, P. Desnoyers, D. Ganesan, and P. Shenoy. Ultra-low power data storage for sensor networks. In *Proceedings of the ACM International Conference on Information Processing in Sensor Networks (IPSN)*, pages 374–381, 2006.

[12] N. Megiddo and D. S. Modha. ARC: A self-tuning, low overhead replacement cache. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST)*, pages 115–130, 2003.

[13] S. Nath and A. Kansal. FlashDB: Dynamic self-tuning database for NAND flash. In *Proceedings of the ACM International Conference on Information Processing in Sensor Networks (IPSN)*, pages 410–419, 2007.

[14] G. Pallis, A. Vakali, and J. Pokorny. A clustering-based prefetching scheme on a Web cache environment. *Computers & Electrical Engineering*, 2007. To appear.

[15] K. S. Prabh and T. F. Abdelzaher. Energy-conserving data cache placement in sensor networks. *ACM Transactions On Sensor Networks*, 1(2):178–203, 2005.

[16] M. Rahimi, R. Baer, O. I. Iroezi, J. C. Garcia, J. Warrior, D. Estrin, and M. Srivastava. Cyclops: In situ image sensing and interpretation in wireless sensor networks. In *Proceedings of the ACM International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 192–204, 2005.

[17] H. Shen, S. K. Das, M. Kumar, and Z. Wang. Cooperative caching with optimal radius in hybrid wireless networks. In *Proceedings of the International IFIP-TC6 Networking Conference (NETWORKING)*, volume 3042 of *Lecture Notes on Computer Science*, pages 841–853, 2004.

[18] 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*, 13(4):104–119, 2006.

[19] B. Tang, S. Das, and H. Gupta. Cache placement in sensor networks under update cost constraint. In *Proceedings of the (ADHOC-NOW)*, volume 3738 of *Lecture Notes on Computer Science*, pages 334–348, 2005.

[20] L. Yin and G. Cao. Supporting cooperative caching in ad hoc networks. *IEEE Transactions on Mobile Computing*, 5(1):77–89, 2006.