International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:6, No:3, 2012

# Cooperative Data Caching in WSN

Narottam Chand

***Abstract*****—**Wireless sensor networks (WSNs) have gained tremendous attention in recent years due to their numerous applications. Due to the limited energy resource, energy efficient operation of sensor nodes is a key issue in wireless sensor networks. Cooperative caching which ensures sharing of data among various nodes reduces the number of communications over the wireless channels and thus enhances the overall lifetime of a wireless sensor network. In this paper, we propose a cooperative caching scheme called ZCS (Zone Cooperation at Sensors) for wireless sensor networks. In ZCS scheme, one-hop neighbors of a sensor node form a cooperative cache zone and share the cached data with each other. Simulation experiments show that the ZCS caching scheme achieves significant improvements in byte hit ratio and average query latency in comparison with other caching strategies.

***Keywords*****—**Admission control, cache replacement, cooperative caching, WSN, zone cooperation

## I. Introduction

RECENT advances in miniaturization of devices and low power system design have led to an increasing interest in wireless sensor networks (WSNs). WSNs consist of low cost, battery operated wireless sensors, which can be used for wide range of surveillance and control applications. Environment monitoring, habitant monitoring, disaster warning system (land slide and avalanche monitoring), military surveillance, etc. are some of the applications, where these small size, battery operated, wireless sensors can be used. WSNs have made it possible for such applications to achieve unprecedented success which otherwise would not have been possible. Due to the low cost of wireless sensors, these can be deployed in large numbers. Apart from sensing, sensor nodes are equipped with data processing and communication capabilities. Due to deployment of wireless sensors in unattended harsh environment, it is not possible to charge or replace their batteries. Therefore, energy efficient operation of wireless sensors to prolong the lifetime of overall wireless sensor network is of utmost importance. Energy consumption mainly occurs due to three types of operations: (i) sensing, (ii) data processing, and (iii) data communication. The applications in WSNs demand to reduce the number of communications among the sensors to serve the requested data with lower latency and minimum energy consumption. Network lifetime of WSN can be enhanced if the rate of nodes' energy depletion is reduced, which is possible if the amount of communication is reduced. This can be achieved by caching useful data for each wireless sensor either in its local store or in the nearby neighborhood.

Caching plays a pivotal role in reducing the number of communications from sensor node (SN) to sink by caching the useful data.

Narottam Chand is Associate Professor with the Department of Computer Science & Engineering, National Institute of Technology, Hamirpur, 177 005 India. E-mail: nar.chand@gmail.com

Providing continuous information to the sink with uninterrupted communication is a big challenge in designing large-scale sensor networks. A lot of research in data routing [1, 2], data compression [5] and in-network aggregation [3, 4] has been carried out in WSNs during recent years. This paper targets the problem of efficient data dissemination and tries to solve it by utilizing the memory of sensor nodes by caching the data items in it. Caching if implemented optimally can reduce network traffic and enhance data availability to the users through sink.

Various researches have been carried out by exploiting data caching either in some intermediate nodes or at a location nearer to the sink in the wireless sensor networks. Jinbao Li et al. [6] propose a caching scheme for the multi-sink sensor network. The sensor network forms a network tree for particular sink. A common subtree is formed out of such trees and the root of the common subtree is selected as the data caching node to reduce the communication cost.

Md. A. Rahman et al. [7] propose effective caching by data negotiation between base station and the sensors, developing expectancy of data change and data vanishing. J. Xu et al. [8] proposed a waiting cache scheme which waits for the data of the same cluster until it becomes available within a threshold, aggregating it with the packet from the lower cluster and then sending it to the sink, thus reducing number of packets travelling in the network. K.S. Prabh et al. [9] consider the whole network to be a Steiner Data Caching Tree which actually is a binary tree and buffers data at some intermediate node (data cache) such that it reduces the network traffic during multicast. In [10], M.N. Al-Ameen et al. exploit caching for faulty nodes in WSNs and propose a mechanism to handle the packets when node fails. T.P. Sharma et. al [11] proposed a cooperative caching scheme which exploits cooperation among various sensor nodes in a defined region. Apart from its own local storage, a node uses storages of nodes from certain region around it to form larger cache storage known as cumulative cache. A token based cache admission control scheme is devised where node holding the token can cache or replace data item. Disadvantage of proposed model is that, there are overheads to maintain and rotate the token. N. Dimokas et. al [12, 17], have identified various goals which are required to be optimized such as energy consumption, access latency, number of copies of data items to be placed at different locations. Disadvantage of schemes is that node importance (NI) considers neighborhood of a particular node. So, overhead to find NI for all the nodes consumes energy which in turn reduces the lifetime of sensor network.

In this paper, we propose a cooperative caching scheme for WSN called Zone Cooperation at Sensors (ZCS). Sensor nodes belonging to the one-hop neighborhood (zone) of a given sensor form a cooperative cache system for this node since the cost for communication with them is low both in terms of energy consumption and message exchanges. In ZCS, each sensor node caches the frequently accessed data items in its non-volatile memory such as flash memory. The data items in the cache satisfy not only the node's own requests but also the data requests passing through it from other nodes. For a data miss in the local cache, the node first searches the data in its zone before forwarding the request to the next node that lies on a path towards the data source. The process of cache admission control is based on the distance criteria of a node from the sink and gives higher priority to the nodes located nearer to the sink. Utility based data replacement policy has been devised to ensure that more useful data is retained in the local cache of a node.

The rest of the paper is organized as follows. The system environment is described in Section II. Section III describes the proposed ZCS cooperative caching scheme. Section IV is devoted to performance evaluation and presents simulation results. Section V concludes the paper.

## II. System Environment

We assume a wireless sensor network consisting of sensor nodes (SNs) that interact with the environment and sense the physical data. A SN that senses and holds the original copy of a data item is called source for that particular data item. A data request initiated by a sink is forwarded hop-by-hop along the routing path until it reaches the source and then the source sends back the requested data. Sensor nodes frequently access the data, and cache some data locally to reduce network traffic and data access delay. As sensor nodes do not have sufficient cache storage e.g. for multimedia data, cooperative caching may be more useful where cached data at sensor node may also be shared by the neighboring nodes.

WSN comprises a group of sensor nodes communicating through omni-directional antennas with the same transmission range. The WSN topology is thus represented by an undirected graph G = (V, E), where V is the set of sensor nodes $SN_1$, $SN_2$, ..., and $E \subseteq V \times V$ is the set of links between nodes. The existence of a link $(SN_i, SN_j) \in E$ also means $(SN_j, SN_i) \in E$, and that nodes $SN_i$ and $SN_j$ are within the transmission range of each other, and are called one-hop neighbors of each other. The set of one-hop neighbors of a node $SN_i$ is denoted by $SN_i^1$ and forms a zone. The combination of nodes and transitive closure of their one-hop neighbors forms a wireless sensor network. The sensor nodes might be turned off/on at any time, so the set of alive nodes varies with time.

*We make the following assumptions in this system environment:*

- Sensor nodes are static, the communication links are bidirectional, and the sensors communicate using multi-hop.
- The WSN is homogeneous i.e. the computation and communication capabilities are the same for all sensor nodes.
- Each sensor node is aware of its geographical coordinates (x, y) through some localization method [13].
- Unique node identifier is assigned to each sensor node in the system. The system has total of M nodes and $SN_i$ ($1 \le i \le M$) is a node identifier. The set of one-hop neighbors of a node $SN_i$ is denoted by $SN_i^1$.
- The set of data items is denoted by $D = \{d_1, d_2, \ldots\ldots d_N\}$, where N is the total number of data items and $d_j$ ($1 \le j \le N$) is a data identifier. $D_i$ denotes the actual data for item $d_i$.
- Data items have varying sizes. The size of data item $d_i$ is denoted with $s_i$.
- The original of each data item is at particular source.
- Each sensor node has a cache space of C bytes and can cache a number of data items depending upon the size of items.
- Data value sensed at a source may change with time. After a data item is updated, its cached copy maintained on one or more nodes may become invalid.

## III. ZCS Cooperative Caching

This section describes our Zone Cooperation at Sensors (ZCS) caching scheme for data retrieval in WSNs. In ZCS caching, it is advantageous for a sensor node to share cached data with its neighbors located in the zone, i.e. sensor that are accessible in one-hop. Sensor nodes belonging to the zone of a given node then form a cooperative cache system for this node since the cost for communicating with them is low both in terms of energy consumption and message exchange. Fig. 1 shows the behavior of ZCS caching strategy for a data request. For each request, one of the following four cases holds:

Case 1: Local hit occurs when copy of the requested data item is stored in the cache of the sensor node. If the data item is valid, it is used to serve the query and no cooperation is necessary.

Case 2: Zone hit occurs when the requested data item is stored in the cache of one or more one-hop neighbors of the requester. Message exchange within the home zone of the requester is required during the cache discovery.

Case 3: Remote hit occurs when the data is found with a node belonging to a zone other than home zone of the requester along the routing path to the data source.

Case 4: Global hit occurs when data item is retrieved from the original data source.

### A. Cache Discovery

ZCS uses a cache discovery algorithm to find the sensor node who has cached the queried data item. When a data request is initiated at a node/sink, it first looks for the data item in its own cache. If there is a local cache miss, the node broadcasts *request* packet to check if the data item is cached in other sensors within its home zone. When a sensor receives the *request* and has the data item in its local cache (i.e., a zone cache hit), it will send an *ack* packet to the requester to acknowledge that it has the data item. In case of a zone cache miss, the request is forwarded to the neighboring sensor along the routing path. Before forwarding a request, each sensor along the path searches the item in its local cache or zone. If the data item is not found on the zones along the routing path (i.e., a remote cache miss), the request finally reaches the data source. When a sensor receives an *ack* packet, it sends a *confirm* packet against first *ack* packet. The *ack* packets for the same item received from other nodes are discarded without further processing. When a sensor/source receives a *confirm* packet, it responds back with the actual data value to the requester.

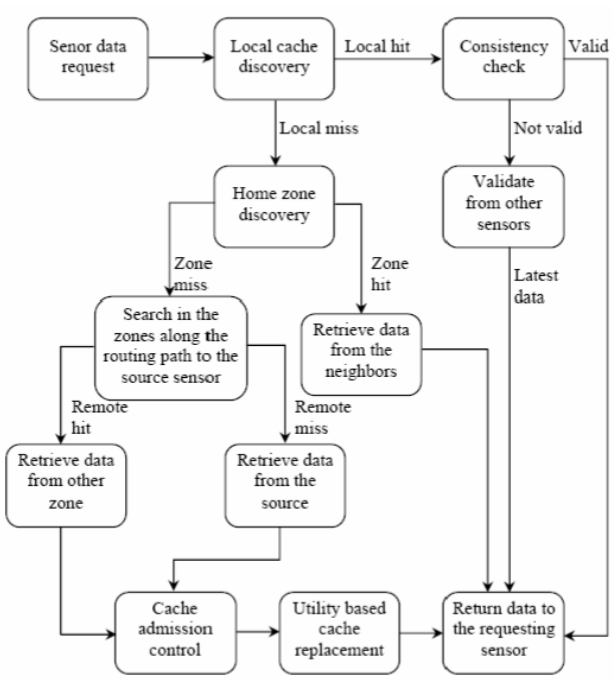



Fig. 1 A data request being serviced in ZCS cooperative caching

### *B. Cache Admission Control*

When a sensor node receives the requested data or a data item passes through it, a cache admission control is triggered to decide whether it should be stored into the cache of the node or not. Inserting a data item into cache might not always be favorable because incorrect decision can lower the probability of cache hits and also makes poor utilization of the limited storage. In ZCS, the cache admission decision at a node $SN_i$ is based on two distance parameters: (i) number of hops $H_i$ between $SN_i$ and sensor/source of the data item from where the cached copy is shared, and (ii) number of hops $H_s$ between $SN_i$ and sink.

If the sensor/source is less than Δ hops away from the requesting node $SN_i$ i.e. $H_i < \Delta$, then it does not cache the data; otherwise it caches the data item. In general, same data items are cached at least Δ hops away. A tradeoff exists between query latency and data accessibility. With a small Δ, the number of replicas for each data item is high and access delay for this data item is low. On the other hand, with a larger Δ, each data item has a small number of replicas, and the access delay can be little longer. Advantage is that sensor nodes can cache more distinct data items and still serve requests when the data source is not accessible. In ZCS, we have used $\Delta = 2$, i.e. if the cache/source of the data resides in the same zone of the requesting node, then the item is not cached, because it is unnecessary to replicate data item in the same zone since cached data can be used by closely located sensors. So in ZCS, the same data item is replicated at least two hops away.

A node $SN_i$ is allowed to cache a data item only if $H_s$ is having a value lower than the specified threshold Ω i.e. $H_s < \Omega$. To increase proximity of the data items nearer to sink, it is better to start caching data items at the sink. Initially all the data items are cached at sink and the next level for cache storage will be a SN along the routing path towards source. Initially we assume $\Omega = 1$ and value is gradually increased with the decreasing free cache space on the sensor nodes located nearer to the sink.

### *C. Cache Consistency*

Cache consistency ensures that sensor nodes only access valid states of the data and no stale data is used to serve the queries. Two widely used cache consistency models are the weak consistency and the strong consistency model. In the weak consistency model, a stale data might be returned to the node. In the strong consistency model, after an update completes, no stale copy of the modified data will be returned to the node.

Due to multi-hop environment, limited bandwidth and energy constraints in wireless sensor networks, the weak consistency model is more attractive [14]. The ZCS caching uses a simple weak consistency model based on Time-To-Live (TTL), in which a SN considers a cached copy up-to-date if its TTL has not expired. The node considers a data item as victim for replacement if its TTL expires. A SN refreshes a cached data item and updates its TTL if a fresh copy of the same data passes by.

### *D. Cache Replacement Policy*

We have developed *utility* based cache replacement policy, where data items with the lowest *utility* are removed from the cache. Four factors are considered while computing *utility* value of a data item $d_i$ at a sensor node:

Popularity ($P_i$), Distance ($\delta_i$), Consistency ($TTL_i$) and Size ($s_i$).

Based on these factors, the $utility_i$ value for a data item $d_i$ is computed as:

$$utility_i = \frac{P_i.\delta_i.TTL_i}{s_i}$$

The objective is to maximize the total utility value for the data items kept in the cache. To achieve this, we use a heuristic that removes a cached data item $d_i$ having least $utility_i$ value until the free cache space is sufficient to accommodate the incoming data.

## IV. Performance Evaluation

In this section, we evaluate the performance of ZCS cooperative caching through simulation experiments.

### *A. Simulation Model*

During the simulation, AODV [15] has been used as underlying routing algorithm to route the data traffic in the wireless sensor network. The number of nodes chosen is 500

International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:6, No:3, 2012

802.11 as the MAC protocol and the free space model as the radio propagation model.

The time interval between two consecutive queries generated from sink follows an exponential distribution with mean $T_q$. After a query is sent out the sink does not generate new query until the pending query is served. The sink generates accesses to the data items following Zipf distribution [16] with a skewness parameter θ. If θ = 0, sink uniformly accesses the data items. As θ is increasing, the access to the data items becomes more skewed. We chose θ to be 0.8.

TABLE I
RADIO CHARACTERISTICS

| Operation | Energy Dissipated |
|---|---|
| Transmitter/receiver electronics ($E_{elec}$) | 50 nJ/bit |
| Transmit amplifier if $d_{toBS} \leq d_0$ ($e_{fs}$) | 10 pJ/bit/m$^2$ |
| Transmit amplifier if $d_{toBS} > d_0$ ($e_{mp}$) | 0.0013410 pJ/bit/m$^4$ |
| Data aggregation ($E_{DA}$) | 5 nJ/bit/signal |

TABLE II
SIMULATION PARAMETERS

| Parameter | Default value | Range |
|---|---|---|
| Number of data items (N) | 1000 | 100~1000 |
| Number of sensor nodes (M) | 500 | 100~500 |
| $s_{min}$ | 1 KB | |
| $s_{min}$ | 10 KB | |
| Bandwidth | 2 Mbps | |
| Transmission range (r) | 40 m | 15~40 m |
| Mean query generate time | 5 sec | 2~100 sec |
| Cache size (C) | 800 KB | 200~1400 KB |
| TTL | 300 sec | 100~300 sec |
| Skewness parameter (θ) | 0.8 | 0.0~1.0 |

The data items are updated only at the source nodes. The sensor/source serves the requests on FCFS (first-come-first-serve) basis. When the source sends a data item to a sink, it sends the TTL value along with the data. The exponential distributed TTL value is used for each data item. After the TTL expires, sensor node/sink has to get the new version of the data item either from the source or from other sensors (having maintained the data item in its cache) before serving the query.

The radio parameters are given in Table I and simulation parameters are given in Table II.

*B. Performance Metrics*

We evaluate following two performance metrics:

Average query latency ($T_{avg}$). The query latency is the time elapsed between the query is sent and the data is transmitted back to the requester, and average query latency ($T_{avg}$) is the query latency averaged over all the queries.

Byte hit ratio (B). Byte hit ratio is defined as the ratio of the number of data bytes retrieved from the cache to the total number of requested data bytes. Here byte hit ratio (B) includes local byte hit ($B_{local}$), zone byte hit ($B_{zone}$) and remote byte hit ($B_{remote}$).

*C. Results*

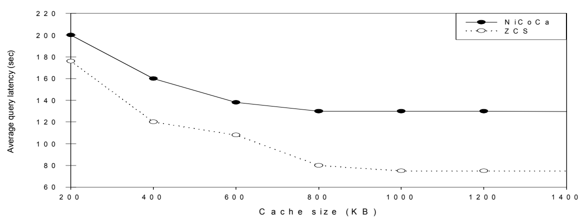
Here we examine the impact of cache size on the performance of proposed ZCS caching strategy. For performance comparison with ZCS, NICoCa [17] is also simulated.

From Fig. 2(a), we can see that the proposed scheme ZCS performs much better than NICoCa scheme due to cooperation within a zone. When the cache size is small, more required data items can be found in local+zone cache for ZCS as compared to NICoCa which utilizes only the local cache. Thus, the need for accessing the remote and global cache in ZCS is alleviated. When the cache size is large enough, the nodes can access most of the required data items from local, zone and remote cache, that reduces query latency.

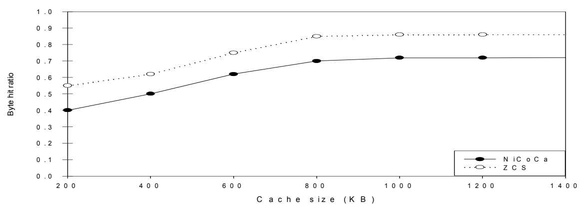
Fig. 2(b) shows that the byte hit increases with the increasing cache size because with large cache size more data can be stored locally and the size of zone cache increases. Due to use of utility based replacement, the ZCS has the higher byte hit ratio at all cache sizes. Due to cooperation within a zone, the byte hit ratio of ZCS is always higher than NICoCa because each node shares caches of its one-hop neighbors. When the cache size is small, the contribution due to zone hit and remote hit is more significant.

V. CONCLUSION

This paper presents a cooperative caching scheme ZCS to improve the performance of the wireless sensor networks. The scheme enables nodes in a zone to share their data which helps alleviate the longer query latency and limited data accessibility problems at a node and prolongs the lifetime of WSN. The ZCS caching scheme includes a cache discovery process, distance based admission control, consistency check and utility based cache replacement policy. The admission control prevents high data replication by enforcing a minimum distance between the same data item, while the utility based replacement policy helps in improving the byte hit ratio.




(a)




(b)

Fig. 2 Effects of cache size on (a) average query latency, and (b) byte hit ratio

## REFERENCES

[1] A. Tripathi, P. Gupta, Aditya Trivedi and Rahul Kala, "Wireless Sensor Node Placement Using Hybrid Genetic Programming and Genetic Algorithms," International Journal of Intelligent Information Technologies, Vol. 7, No. 2, pp. 63-83, 2011.

[2] A. Abbasi and M. Younis, "A Survey on Clustering Algorithms for Wireless Sensor Networks," ACM Journal of Computer Communications, Vol. 30, No. 14-15, pp. 2826-2841, 2007.

[3] T.P. Sharma, R.C. Joshi and M. Misra, "Dual Radio Based Cooperative Caching for Wireless Sensor Networks," IEEE International Conference on Networking, pp. 1-7, 2008.

[4] E. Fasolo, M. Rossi, J. Widmer and M. Zorzi, "In-Network Aggregation Techniques for Wireless Sensor Networks: A Survey," IEEE Wireless Communications, Vol. 14, No. 2, pp. 70-87, 2007.

[5] N. Kimura and S. Latifi, "A Survey on Data Compression in Wireless Sensor Networks," International Conference on Information Technology: Coding and Computing, Vol. 2, pp. 8-13, 2005.

[6] J. Li, S. Li and J. Zhu, "Data Caching Based Queries in Multi-Sink Sensor Networks," International Conference on Mobile Ad-hoc and Sensor Networks, pp. 9-16, 2009.

[7] Md. A. Rahman and S. Hussain, "Effective Caching in Wireless Sensor Network," International Conference on Advanced Information Networking and Applications Workshops, Vol. 1, pp. 43-47, 2007.

[8] J. Xu, K. Li, Y. Shen and J. Liu, "An Energy-Efficient Waiting Caching Algorithm in Wireless Sensor Network," International Conference on Embedded and Ubiquitous Computing, Vol. 1, pp. 323 - 329, 2008.

[9] K.S. Prabh and T.F. Abdelzaher, "Energy-Conserving Data Cache Placement in Sensor Networks," ACM Transactions on Sensor Networks, Vol. 1, No. 2, pp. 178–203, 2005.

[10] M.N. Al-Ameen and Md. R. Hasan, "The Mechanisms to Decide on Caching a Packet on Its Way of Transmission to a Faulty Node in Wireless Sensor Networks Based on the Analytical Models and Mathematical Evaluations," International Conference on Sensing Technology, pp. 336-341, 2008.

[11] H. Luo, F. Ye, J. Cheng, S. Lu and L. Zhang, "A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks," International Conference on Mobile Computing and Networking, pp. 148-159, 2002.

[12] N. Dimokas, D. Katsaros, L. Tassiulas and Y. Manolopoulos, "High Performance, Low Complexity Cooperative Caching for Wireless Sensor Networks," Springer International Journal of Wireless Networks, Vol. 17, No. 3, pp. 717-737, 2011.

[13] B. Xiao, H. Chen and S. Zhou, "Distributed Localization Using a Moving Beacon in Wireless Sensor Networks", IEEE Transactions on Parallel and Distributed Systems, Vol. 19, No. 5, pp. 587-600, 2008.

[14] 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.

[15] C. Perkins and E.M. Royer, "Ad Hoc On-Demand Distance Vector Routing," IEEE Workshop on Mobile Computing Systems and Applications, pp. 90-100, 1999.

[16] L. Breslau, P. Cao, L. Fan, G. Phillips and S. Shenker, "Web Caching and Zipf-Like Distributions: Evidence and Implications," IEEE International Conference on Computer Communications, Vol. 1, pp. 126-134, 1999.

[17] N. Dimokas, D. Katsaros and Y. Manolopoulos,"Cooperative Caching in Wireless Multimedia Sensor Networks," Springer Journal of Mobile Network Applications, pp. 337-356, 2008.

**Dr. Narottam Chand** received his Ph.D. degree from IIT Roorkee in Computer Science and Engineering. Previously he received M.Tech. and B.Tech. degrees in Computer Science and Engineering from IIT Delhi and NIT Hamirpur, respectively.Presently he is working as Associate Professor, Department of Computer Science and Engineering, NIT Hamirpur. He has served as Head, Department of Computer Science & Engineering, during Feb 2008 to Jan 2011 and Head, Institute Computer Centre, during Feb 2008 to July 2009.His current research areas of interest include mobile computing, mobile ad hoc networks and wireless sensor networks. He has published more than 120 research papers in International/National journals & conferences and guiding PhDs in these areas. He is member of ACM, IEEE, ISTE, CSI, International Association of Engineers and Internet Society.