

Research Article

An Probability-Based Energy Model on Cache Coherence Protocol with Mobile Sensor Network

Jihe Wang,¹ Bing Guo,¹ and Meikang Qiu²

¹ Computer Science College, Sichuan University, Chengdu, Sichuan 610064, China

² Department of Electrical and Computer Engineering, University of Kentucky, Lexington, KY 40506, USA

Correspondence should be addressed to Meikang Qiu; mqiu@engr.uky.edu

Received 12 January 2013; Accepted 3 April 2013

Academic Editor: Jianwei Niu

Copyright © 2013 Jihe Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Mobile sensor networks (MSNs) are widely used in various domains to monitor, record, compute, and interact the information within an environment. To prolong the life time of each node in MSNs, energy model and conservation should be considered carefully when designing the data communication mechanism in the network. The limited battery volume and high workload on channels worsen the life times of the busy nodes. In this paper, we propose a new energy evaluating methodology of packet transmissions in MSNs, which is based on redividing network layers and describing the synchronous data flow with matrix form. We first introduce the cache coherence layer to the protocol stack of MSNs. Then, we use a set of energy probability matrices to describe and calculate the energy consumption of each state in the protocol. After that, based on our energy model, we will give out an energy evaluating method of the MSNs design, which is suitable for measuring and comparing the energy consumption from different implements of hardware/software. Our experimental results show that our approach achieves a precision with less than 2% error and provides a credible quantitative criterion for energy optimization of data communication in MSNs.

1. Introduction

A *Mobile sensor network* (MSN) is a wireless sensor network which is widely used in many industrial and consumer applications, such as industrial process monitoring and control and machine health monitoring. Various kinds of MSNs, such as *vehicular ad hoc networks* (VANETs), *underwater sensor networks* (UWSNs), and *wireless body area networks* (WBANs), have been widely researched and developed to provide ubiquitous solutions for real-time monitoring [1]. A sensor network usually comprises some node sensors, wireless communication devices, microcontrollers, and power source. As a mobile device, the limited battery volume cannot supply the durable power to the sensor node [2], known as *power wall*. Sensors in a wireless sensor network are prone to failure, due to energy depletion [3]. In order to save the energy consumption on communication, some base stations are added into the network as a center of a cluster of near nodes. These base stations are able to send queries and gather the data from the sensor nodes. Depending on the applications, the sensors are deployed randomly or using a

systematic approach to gather the information from the environment [4]. Making effective use of multiple memory modules remains difficult, considering the combined effect of performance and power requirement [5]. Because fetching and synchronizing the data in remote node is a high energy consuming procedure, caching the data in the memory media of local node can effectively decrease the energy consumption on data communication. However, to measure and evaluate the energy consumption of cache coherence among a large number of nodes is not easy. (1) The application's data operations are randomly happening on each node, which make the energy consumed in communication unpredictable. (2) In a distributed cache system, synchronizing a data between a pair of nodes includes several steps, such as requiring, responding, and transmitting, and each step is independent from the other one, which increase the uncertainty of the procedure. So, precisely describing the complicated energy consumed in the procedure with mathematical tools is the key solution of this issue.

Generally, the energy of a node can be divided into three kinds: listening energy, mobility energy, and communication

energy. The listening energy is consumed when the sensor node is in inactive state, in which, the sensor's radio module is ready to transmit or receive data at any time. The mobility energy comes from mobile sensor physical movements, and the power dissipation through the movement of sensors is dependent on the number of sensors moved during the deployment phase. The communication energy is the energy consumed when the message is exchanging among sensors [6]. We focus on the communication energy in MSNs because (1) the listening energy is the static part of the energy unrelated with applications; (2) in our architecture, the mobility energy can be ignored when a mobile node only moves around its master base station. As the dynamic part of the energy consumption, the communication energy is completely driven by the applications running on each node, which can be seen as the source of remote data fetching and caching.

It is foreseen that huge gains are achievable both in terms of overall MSN energy saving and data availability, if caching is properly implemented in MSN. Since queries can be serviced at nearby node or local cache, caching reduces data access time and thus obviates the need for query to travel to actual data source that would be quite far from the querying node [7]. There are two valid cache strategies of caching in MSNs: one is to cache the recently used data in the mobile node waiting for being reused by the local (or nearby) sensor; another one is to cache the data in the master base station of the mobile node so that all sensors in one cluster can share the cache data stored in their public base station. Our energy model supports both of the two cache strategies.

With the traditional ISO/OSI network, the stack is divided into seven layers, from the physical layer to the application layer. However, this partition cannot be directly migrated to the MSN with cache coherence, because more conditions should be considered carefully: (1) the data communication comes from the sensor's remote fetching operation, instead of application's *send* and *receive* functions; (2) if the remote data is written by an arbitrary node and has not been synchronized over the network, the received message might contain dirty data; Therefore, we redivide the upper layers of OSI network so that the cache coherence protocol becomes the kernel application in MSN. Based on the new network stack, our energy model of cache coherence protocol can be built to describe the energy dissipating over the network.

In this paper, we propose a new energy evaluating methodology of packet transmissions in MSNs, which is based on redividing network layers and describing the synchronous data flow with matrix form. Specifically, our main contributions are as follows.

- (1) A new cache coherence protocol is inserted into the network stack of MSNs, based on which, all data communication in the MSNs can be replaced by the movements of state in the protocol.
- (2) We propose a high accuracy probability-based energy model for the cache coherence protocol. It combines the energy model of network components, such as routing unit and link on base stations to measure the energy distribution of single message in MSNs.

- (3) We propose a new energy comparison method to evaluate different types of cache coherence protocols for single message, which strongly depend on cache organizations, network design, and run-time environments.

The remainder of this paper is organized as follows. Architecture and data fetching procedure of the cache coherence protocol are presented in Section 2. Our probability-based energy model for the the protocol is described in Section 3. The new energy comparison method is depicted in Section 4. Experimental results are shown in Section 5. Section 6 contains some related work. Finally, Section 7 concludes the paper.

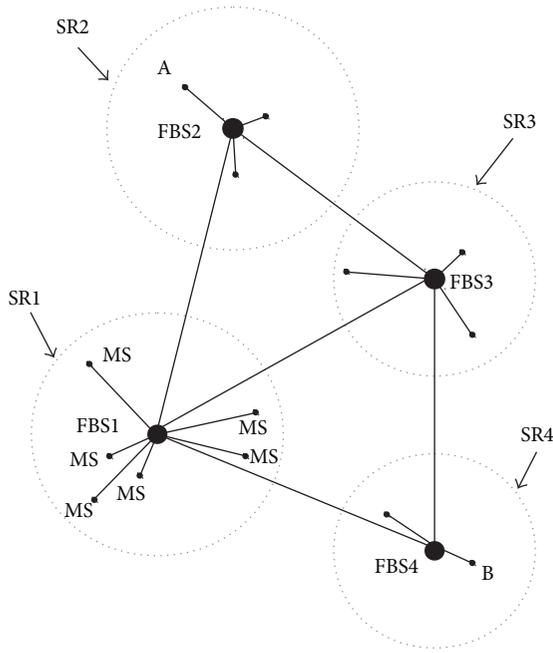
2. Architecture and Data Fetching Procedure

A typical mobile wireless sensor network includes some interconnected fixed base stations (FBSs), more mobile sensors (MSs) around these FBSs, and the sensing regions (SRs) of each FBS define the corresponding FBS' control area. Based on the architecture, a network stack is set up to provide the network services to the applications running in each MS. Using caching mechanism, the remote data reading and writing services can be provided by the cache coherence protocol layer in the stack, so that all applications can invoke the synchronization functions identically.

2.1. Architecture Overall. As mentioned above, components of an MSN, for example FBSs and MSs, are organized as a hierarchical fashion. Figure 1 provides an illustration of the typical architecture of the MSN. End MSs communicate with each other with the mechanisms available through their directly connected master FBS.

Mobile Sensor. Abstractly speaking, a sensor is a device capable of measuring a physical quantity and transforming it into a format that can be correctly interpreted by an instrument, namely, a computer or digital device. However, the current mobile sensors, such as the ends of mobile internet, have high performance on computing. They are able to process some simple and complicated tasks with local processors, which decreases the workload of communication, while demands more complex data synchronization mechanism for the distributed sensor. In this situation, the energy consumed by network is mainly used for fetching the remote data from other locations.

Fixed Base Station. A FBS stays in the center of an SR and gathers data from sensor nodes through shot-range communications, so that the energy consumption of each sensor is reduced, since fewer relays are needed for the sensor to send and receive its message to and from its master FBS [8]. The communication between a pair of FBSs is a more reliable channel than the one between MS and its master FBS, such as a wired connection based on TCP. In this architecture, we assume that the relative locations of any two FBSs are concrete so that they invoke fixed routing algorithm to find the target FBS and MS.



FBS: Fixed base station
 MS: Mobile sensor
 SR: Sensing region

FIGURE 1: The architecture of typical mobile sensor network.

In the architecture, different MSs communicate with each other by the FBSs. For example, in Figure 1, if an MS A in SR2 wants to communicate with another MS B which resides in SR4 remotely, A has to send the requesting message to its master fixed base station FBS2 as an agency, and FBS2 retransmits the message to FBS4 which is the master FBS of MS B through several hops in the network, for example, FBS2 → FBS3 → FBS4. The energy consumption of this procedure combines the energy of each network component in the transition path of the message. When the application on the sensors synchronize data with a high frequency, the remote connection above is set up over and over, and the energy is wasted if the remote data does not change. Therefore, a cache coherence protocol layer is added into the traditional MSN stack so that MS can get the remote data with a prompt response.

2.2. Protocol Stack of MSN. As a packet-switching network, MSNs need a protocol stack to support various communication services at different levels. However, limited resource and power on the mobile sensors can hardly sustain an integrated stack as complex as *open system interconnection* (OSI). Therefore, a simple network topology and light stack are needed to provide high-efficiency communication services with less energy consumption. Two significant differences exist between MSN and OSI protocol stacks: (1) the goal of designing OSI is to meet the communication demands of most network environments and conditions. However, in MSN design, the cache organization affects the communication, since different cache capability on each node determines the hit rate of the sensor’s applications. Thus, this may affect

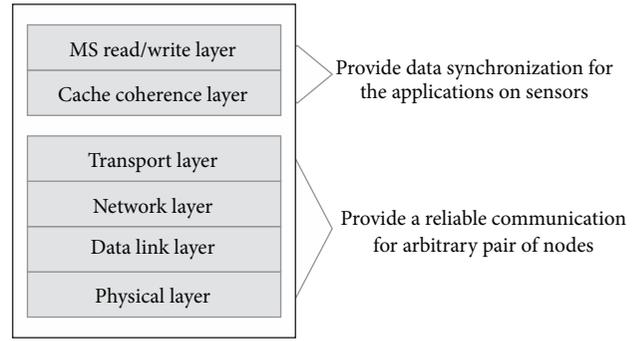


FIGURE 2: A new layer division of MSN stack.

communication patterns over the MSN. (2) The operating environment of OSI cannot be predicted precisely, and a series of safeguarding mechanisms are implemented to maintain the integrity and correctness of data units in each layer. However, MSN resides on distributed sensors and base station, and designers have to focus on how to utilize the limited resources and reduce the energy consumption.

As shown in Figure 2, we insert a cache coherence protocol layer to MSN stack to control the cache coherence transaction. When a read/write operation of a MS is launched, a data request is added to a request queue of its master FBS if the required data is missing in local FBS’s cache. These requests will be sent into backbone network by the FBS’ *network interface* (NI), and a remote data cache will respond with a synchronizing message to the requester. In this procedure, the memory address is mapped to the remote node’s identifier and path to the remote node.

FBS cache missing is the reason of communication activities, and the knowledge of FBS cache is enough to set up an energy consumption model because the location of the remote responder can be deduced and the transferring path is determined by the cache coherence protocol. At any time, any node that produces a transaction can locate the required data in the shortest time. As one of the most widely used unit in cache coherence protocol, the directory table utilizes the distributed directory items to record the states of memory blocks, such as locations, sharing members, and owners. With this information, the requester traces the newest copy of a memory block, and a shortest path between the responder and the requester is reserved. We will focus on the energy consumption in this protocol because all location information can be determined and it is possible to determine the hops of a message in a specific network typology, considering fixed routing mechanism.

2.3. Data Fetching Procedure. The cache coherence protocol can set up a 3-way data fetching procedure to load the remote data into local FBS cache. We describe the fetching procedure as shown in Figure 3.

Request Step. The *request node* produces the *home node’s* location i ; that is, the block directory information required by the *request node* may be stored into the i node’s directory table. Then, the *request node* sends the *home node* a request

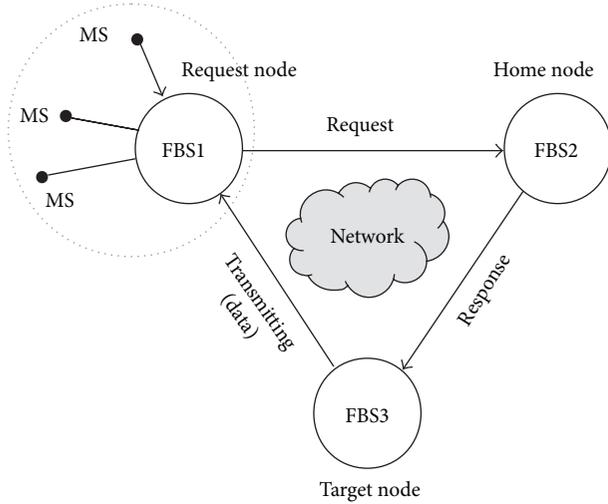


FIGURE 3: Procedure of data synchronization in MSN.

message which includes the memory address of the required block.

Response Step. After the *home node* receives the request message, it uses the block address to search the directory item in its directory table. If the directory item shows that there is an available image of the required memory block in a *target node*, the *home node* sends the *target node* a transfer request message which includes the required block address and the *request node's* location.

Transmitting Step. When the *target node* receives a transfer request message, it sends a copy of the required memory block to the *request node*. Finally, remote data synchronization finishes.

For example, in Figure 3, MS launches a remote memory accessing to its master FBS1. When the FBS1 finds that the required data is not stored in the local cache, a request message to the home node of the data is sent, FBS2. The directory item in FBS2 indicates that the newest copy of the data can be obtained from a target node, FBS3, then FBS2 will notice the FBS3 to send a copy of the data to the original requester, FBS1. From the viewpoint of MS, this cache coherence protocol provides consistent accessing to any data over the network. For simplifying model, we neglect some competition-to-avoid mechanism, which is related to waiting time of the requester, while independent from the energy consumption model.

This fetching procedure is able to balance the network load when all directory information is fairly distributed in each FBS. We will further analyze the energy consumption of the 3-way data fetching procedure and build its energy evaluation model.

3. Energy Model of Data Fetching

As a widely used cache coherence unit, a directory item includes a 3-way locating procedure to obtain the newest copy of memory blocks from remote nodes, as shown in

Section 2.3. This procedure simplifies the communication because write and read operations are performed in the same transaction, and each transaction also corresponds to a single packet travelling in the network. Our energy consumption model takes into account each step of the data transfer procedure.

3.1. Location of Remote Data. In this section, we use uniform memory address to locate the data in the whole MSN system. With the distributed architecture of the memory, each FBS is mapped to a piece of consecutive memory address. As the master of a set of nearby MSs, an FBS provides cache service to the MSs in its SR. Each MS in an SR can directly access the cache of its master FBS. Thus, message passing mechanism is used for an MS to access memory modules located at a remote node in the network. Therefore, cache access by an SM is not uniform since it depends on which memory address the SM wants to access. It is referred to a *nonuniform memory access* (NUMA) system. Many different cache mapping strategies have been designed for distributed cache organization to achieve a higher hit rate. We use a linear address model that is simple but scalable to find directory items in a remote FBS node.

A directed graph $MSN(S_N, L)$ is used to model an MSN with N FBS nodes, where $s_i \in S_N$ is the i th routing nodes in the graph, i represents the location of node s_i , $0 \leq i \leq N - 1$, and $l_{i,j} \in L$ is a directed edge between s_i and s_j . A one-dimensional flat memory can be expressed as a two-tuple $M(l, b)$, where l is the number of the basic blocks in the memory space and b is the number of parts, and each part includes l/b memory blocks. A number of distributed directory record tables map all blocks in memory and store their current states, while all records items of a part are stored in the same *home node*. If the number of parts equals the number of nodes, that is, $b = N$, then a mapping function from a block address in memory to its *home node's* location in MSN can be represented as

$$i = \left\lfloor \text{block_addr} \times \frac{b}{l} \right\rfloor, \quad (1)$$

where *block_addr* is the basic address of the block in memory. The addresses of all memory units in one block use the same *block_addr*. This equation can determine the location of *home node* where the block's directory record is stored. Then, the *request node* needs to launch a fetching process to obtain the state information of the block in the *home node*. We will discuss the fetching process in the following section.

3.2. Distribution of Request and Response. Our energy consumption model is based on the transitivity of packet sending probability between the request and response steps. We utilize the independency between the request and response steps to provide the matrix expressions of the transitive relationship. We also describe how to combine these probability matrices with the energy consumption of a single packet to estimate the energy distribution on node pairs of an FBS network. In MSN, all request messages requiring remote data can be seen as random events. Thus, we use $q_{i,j}$ to represent the probability of FBS i 's SM asking for the block whose record item

is stored in node j . Then we utilize a matrix \mathbf{Q} to describe the request sending probability distribution of the whole MSN. Specifically, (1) if the diagonal elements $q_{i,i} > 0$, a part of requests from node i hits the local cache of the master FBS, and node i does not need to look up the directory item in remote nodes. If $q_{i,i} = 0$, the required data block is always found in the remote FBS' cache. (2) Since \mathbf{Q} is a probability matrix, it is natural that the sum of a row in \mathbf{Q} must equal 1. (3) $q_{i,j} (i \neq j)$ could be any value between 0 and 1.

Matrix \mathbf{Q} is a probability matrix that contains more hints about the transmission energy. If requests from node i are mainly requests to nearby nodes rather than to distant ones, the energy consumed in the request process could be lower. For example, in a 2×2 mesh network, consider a probability vector $\mathbf{q} = (q_{1,0}, q_{1,1}, q_{1,2}, q_{1,3})$, a row of the 4×4 matrix \mathbf{Q} . If the probability of this row is mainly contributed by $q_{1,0}$, $q_{1,1}$, and $q_{1,2}$, then the energy dissipation from the requests to distant nodes, $q_{1,3}$, cannot increase the overall energy consumption significantly. However, if $q_{1,3}$ is dominant, the requests from node 1 to node 3 can seriously impact overall energy consumption.

We adopt a similar approach to describe the response probability distribution. A matrix \mathbf{P} can be used to express the relationship between home and target nodes. Its element $p_{j,k}$ is the probability with which the home node j sends the transfer request message to the target node k . In some specific implementations of the cache coherence protocol, designers employ different swap-out methods to delete the trashy image or spread a block's image in a limited range, where some blocks can never appear in a node or will always be accessed by a node. These optimal strategies are viewed as special probability distributions in our matrix space, which may lead to better performances, such as fewer average transmission hops and lower energy consumption. We do not intend to restrict this model in a specific implementation, so it is reasonable to assume that an image of any memory block can resident in any node, and a specific strategy always has corresponding probability distribution matrix \mathbf{P} .

We also list the conditions for matrix \mathbf{P} : (1) if the diagonal elements $p_{j,j} \neq 0$, a part of transfer request messages hits in the local home node; that is, node j is also the target node during the *response step*. If $p_{j,j} = 0$, the available copy of required data block is never found in the home node j , so all transfer request messages will be sent to the remote nodes for a copy of memory image. (2) Since \mathbf{P} is a probability matrix, the sum of a row in \mathbf{P} must equal 1. (3) $p_{j,k} (j \neq k)$ could be any value between 0 and 1, and that depends on the run-time environment and the swap-out strategy of each node.

In the following section, we will show how to build the energy consumption distribution using \mathbf{Q} and \mathbf{P} .

3.3. Energy Consumption on FBS Pairs. A packet traveling through MSN motivates the activities of the components in its routing path, such as routers and links. Routing algorithms always choose the shortest path between a pair of source and sink nodes for a packet travelling through MSN. Thus, we can count the number of hops between the source node and the destination node to estimate the energy incurred by the transfer of a packet. If each node in the routing path has a

homogeneous routing unit and works in the same frequency, then the energy consumption by a packet E_{packet} can be expressed as

$$E_{\text{packet}} = (h + 1)E_R + hE_L, \quad (2)$$

where h is the hop along the path of the packet transferring between a node pair; E_R and E_L are the average energy consumption of a single packet passing through a routing unit and a link, respectively.

Consider a mesh backbone MSN network with $R \times C$ nodes connected together, and these nodes take identifiers from 0 to $R \times C - 1$. If a packet travels from node i to node j , the shortest Manhattan path between the two nodes should be chosen, which includes a constant number of routers and links. On the mesh network, in order to reach node j , the packet has to go through $|(j - i) \bmod(c)| + 1$ routers along the horizontal axis. Yet, along the vertical axis, the shortest path contains $\lfloor j/c \rfloor - \lfloor i/c \rfloor + 1$ routers. Thus, from (2), the energy consumed by a single packet between node i and j is

$$e_{i,j} = \begin{cases} \left[|j - i| \bmod(c) + \left| \left\lfloor \frac{j}{c} \right\rfloor - \left\lfloor \frac{i}{c} \right\rfloor \right| \right] (E_R + E_L) + E_R; & i \neq j, \\ 0; & i = j. \end{cases} \quad (3)$$

The key idea is to count the number of hops of a Manhattan path and then sum up all the energy from the components that transport a packet. In general, there are several shortest paths between two nodes that contain the same number of routers and links. However, the energy consumption is not affected by the choice of different shortest paths.

3.4. Energy Consumption of Data Transmitting. Since the energy of transmitting step takes the greatest portion of energy in MSN design, we need to decide the probability distribution of single packet's transition which includes the block data for cache synchronization.

3.4.1. Independence between Request and Response Steps. For directory table-based cache systems, synchronization is always a costly operation. A request node has to look up the directory items stored in home node before a data transmission action starts. That means, on the one hand, the requester does not know where an available copy of the block is stored in the network until the search finishes in the home node. On the other hand, when a node is searching in its directory table, it does not need to reference the location of the requester on the network. If we use one random event $A_{i,j}$ to represent that node i sends a request to node j and another random event $B_{j,k}$ to represent that node j finds a needed data stored in node k , then the events $A_{i,j}$ and $B_{j,k}$ are statistically *independent* from each other:

$$\Pr(A_{i,j} \cap B_{j,k}) = \Pr(A_{i,j}) \Pr(B_{j,k}). \quad (4)$$

In some protocol implementations, several copies of one cache block may be distributed in different nodes. These

Require: A MSN prototype; Sequence of memory accessing operation (Opt) from each MS.

Ensure: Data transmission probability matrix \mathbf{T} .

- (1) Get a cache missing event Opt_{RN} in request node RN
- (2) $ADDR \leftarrow$ accessing address of Opt_{RN}
- (3) Home node $HN \leftarrow ADDR$ with (1)
- (4) $req_home_{RN,HN} ++; req_total_{RN} ++$
- (5) Send request message to HN
- (6) **if** $ADDR$ is cold **then**
- (7) Forward request message to main memory
- (8) **else**
- (9) Get target node TN from directory table
- (10) $trans_target_{HN,TN} ++; trans_total_{TN} ++$
- (11) **end if**
- (12) Send transfer request message to TN
- (13) **for** Each group of three nodes $i, j, k \in NoC$ **do**
- (14) $Q_{i,j} = req_home_{i,j} / req_total_i$
- (15) $P_{j,k} = trans_target_{j,k} / trans_total_j$
- (16) $T_{k,i} += Q_{i,j} * P_{j,k}$
- (17) **end for**
- (18) Send data to RN

ALGORITHM 1: Forming a data transmission matrix \mathbf{T} .

nodes constitute a candidate set. Any node in the set could be a potential target node in the transmitting step. We only select one node located closest to the requester in order to shorten the transmission distance, which leads to lower energy consumption. Since the candidate set is *independent* from the location of the requester, the selected target node's position is also independent from the requester, no matter how close the target node will be from the request node. Based on this independence, we construct the data transmission energy model in the following section.

3.4.2. Energy Consumption of a Single Packet Transmission.

The final goal of the cache coherence protocol is to send the latest memory block image from the target node to the request node. Considering that if a packet is transferring from node k to node i , two independent events, $A_{i,j}$ and $B_{j,k}$, defined in Section 3.4.1, have happened before the data transfer. The probability of a packet transferring from node k to node i can be formulated as

$$\Pr(C_{k,j}) = \sum_{j=0}^{N-1} \Pr(A_{i,j}) \Pr(A_{i,j} | B_{j,k}), \quad (5)$$

thus

$$\Pr(C_{k,j}) = \sum_{j=0}^{N-1} \Pr(A_{i,j}) \Pr(B_{j,k}), \quad (6)$$

where $C_{k,i}$ is the event that a packet transfers from node k to node i . We use an $N \times N$ matrix \mathbf{T} to represent the probability distribution of data transferring; then \mathbf{T} can be deduced from (6) as

$$\mathbf{T} = (\mathbf{QP})^T. \quad (7)$$

The result needs to be transposed to fit the correct direction of data transmission. The meaning of elements of matrix \mathbf{T} is similar to that of \mathbf{Q} and \mathbf{P} . Element $t_{k,i}$ is the probability of sending a packet from node k to node i . Algorithm 1 shows the procedure forming the data transmission matrix \mathbf{T} .

In (3), we use $e_{i,j}$ to denote energy consumed by a packet travelling from node i to node j . Thus, $e_{i,j}$ could be a fixed parameter of the shortest path between a pair of nodes in the network. We multiply $e_{i,j}$ to each element of the data transferring probability matrix \mathbf{T} , shown in (8), and get the energy distribution of fetching a single block over MSN:

$$\mathbf{E} = (e_{i,j} t_{i,j})_{N \times N}. \quad (8)$$

The matrix \mathbf{E} can be used to display the energy consumption distribution of data transferring in network. In particular, (1) the sum of a row $\sum_{i=0}^{N-1} e_{k,i} t_{k,i}$ is the expected energy when sending a packet through MSN from node k . (2) The diagonal element $e_{k,k} t_{k,k}$ equals 0, because $e_{k,k} = 0$ (see (3)).

With our energy consumption model above, the data update can be achieved conveniently. Referring the description in Section 2, a remote data update includes three steps. At first, a copy of remote data is fetched from a remote node, which can be seen as a regular data fetching procedure in our model. Then, the data is written in the local memory. If the data is set with a write-through flag, this copy of data should be sent back to the remote data so that the other node is able to use the latest data version, which is also another regular data fetching procedure:

$$E_{\text{update}} = 2 * E_{\text{fetching}} + E_{\text{localwrite}}. \quad (9)$$

Therefore, the energy consumption of a data update procedure is the sum of two regular data fetching energy and one local data writing energy, see (9).

4. Energy Estimation with Random Sending

To use our energy model to evaluate different cache organizations and application algorithms on sensors, we present a random sending solution that does not include any optimizing information. By comparing with the random sending solution, we can determine the optimization gain of a given solution. Using the expected energy of a single packet in directory protocols, we can define a random sending solution which is a complete, fair packet sending instance that satisfies $q_{i,j} = q_{i,k}$ and $p_{i,j} = p_{i,k}$ for any node identifier i, j , and k . In a random sending solution, all elements in matrix \mathbf{P} , \mathbf{Q} , and \mathbf{T} equal $1/N$, where N is the node quantity in the MSN. This is because $q_{i,j} = p_{j,k} = 1/N$ for any node i, j , and k . Then $t_{k,i} = q_i p_k = N \times (1/N)^2 = 1/N$, where q_i and p_k are the i th row vector of matrix \mathbf{Q} and k th column vector of matrix \mathbf{P} , respectively.

In the random sending solution, the sending probability between any pair of FBS nodes is uniform, so there is no information about data aggregation and address optimization in this solution. In other words, it's unnecessary to adopt a solution whose energy performance is worse than the random one. We use a scalar α to represent the energy optimization gain for a given solution, and $e_{i,j}^{\text{given}}$ and $e_{i,j}^{\text{random}}$ are elements of matrix \mathbf{E} of a given and random solution, respectively

$$\alpha = \frac{\sum_{i,j} e_{i,j}^{\text{random}}}{\sum_{i,j} e_{i,j}^{\text{given}}}. \quad (10)$$

Based on the uniformization, α is able to evaluate the energy optimal depth of any given solution because the random solution can be calculated without any measurement. A bigger α indicates that the given solution can save more energy during the transmission of a single packet from target node to the requester. We will use this ratio to evaluate different energy optimal solutions in our experiments section.

5. Model Evaluation

In order to verify the effectiveness of our energy model, we perform two experimental cases. Firstly, we use simulator to mimic the behaviours of cache coherence protocol and test the accuracy that characters the performance of this model in a more practical context. Secondly, the model is inserted into mobile sensors network to help to measure the energy consumption of a set of application we developed.

5.1. Evaluation Methodology. In the evaluation, we developed our simulator as our experimental platform. Our simulator contains 16 FBSs organized as a 4×4 mesh network. Each of the FBS nodes master a random number of MS, from 1 to 255, and a directory table. The cache stores the cached data by the FBSs; the directory table keeps the locating records for part of memory. We use (1) to map a specific address to the position of its directory table. To demonstrate the accuracy of our energy model, we implement a random packet sending system as a reference. Three packet sending patterns are used to test the model in different environments.

Evaluation Methodology. The basic method to measure the energy consumption of the MSN is to set an energy monitor in each link and router. When a packet arrives at a monitor of a link, a basic energy value is added to the link's energy property. When a packet enters a router, the energy consumption of router's components, such as *buffer* and *crossbar* are also accounted for using an average energy value for one packet. The statistic data of links and routers are used at the end of simulation to calculate the overall energy consumption of the MSN system.

Configuration of Network. As an event-driven network simulator, our simulator provides flexible interfaces to create various network units and sample these units' parameters. In our 4×4 mesh network, each pair of neighbouring nodes is connected with a link which has 10 Mbps bandwidth. This bandwidth guarantees that there is no packet loss. The link's basic energy value mentioned above is also initiated based on this active frequency. When a data transmission event occurs, the synchronous packets have a length of 512 bytes which fit one data block of memory and cache line. In the NI of a target node, a packet is divided into 4 messages, and the routers in their transfer path only launch the routing algorithm for the head message. After the tail flit leaves the router, all resources allocated to this packet, such as buffer space and crossbar path, are released. We use the simple X-Y routing algorithm in the mesh to determine the shortest path between a pair of source and sink nodes.

Packet Sending Patterns. During a specific interval, node is either in the bursting state or in the idling state. In the experiments, we use the proportion of bursting time in one second to control the expected sending rate. Since a time interval is set to one second, the expectations of bursting time $E(\text{BT})$ and idling time $E(\text{IT})$ of one node satisfy $E(\text{BT}) + E(\text{IT}) = 1$. The sending probability of a node can be calculated as $E(\text{BT})/[E(\text{BT}) + E(\text{IT})] = E(\text{BT})$. We design three packet sending patterns to represent different task mapping results of an application: 3×3 , 5×5 , and 7×7 windows. For example, when the 3×3 window is chosen, the node in position of (i, j) , where i and j are the row and column identifiers, sends packets only to the nodes in $(i-1, j-1)$, $(i-1, j)$, $(i-1, j+1)$, $(i, j-1)$, $(i, j+1)$, $(i+1, j-1)$, $(i+1, j)$, and $(i+1, j+1)$, forming a 3×3 sending window. This type of sending model satisfies the random feature in the sending window of node (i, j) area. The basic idea of this kind of design is that, if a task is mapped to several neighbouring nodes, the communication between the neighbours will happen with much higher probabilities than the remote node pair.

5.2. Experimental Setup and Applications. Our simulators were conducted using Intel Core i5-2410M CPU which integrates 2.30 GHz quad core. The system memory used in this machine is an 8 GB DDR3 memory with 1600 MHz clock frequency. This machine runs Linux with the 3.5.0-17-generic Kernel.

Our experiments are designed based on NS-2 to set up the simulation environment, which includes the architecture of our network design and the infrastructures of the MSN system, such as node sensors, wireless devices microcontroller,

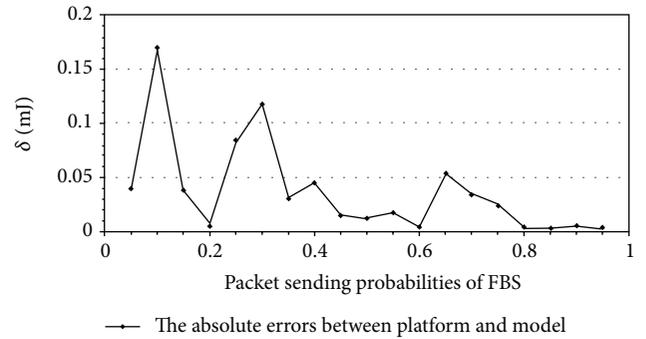
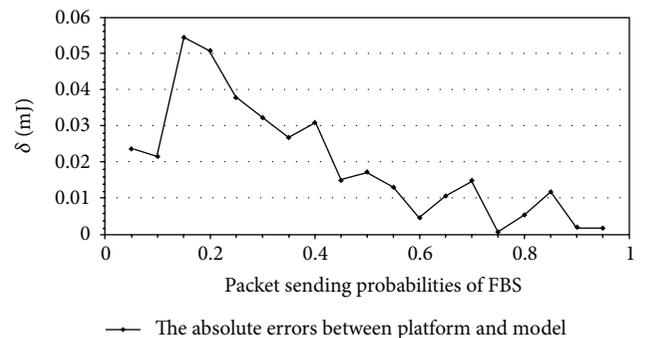
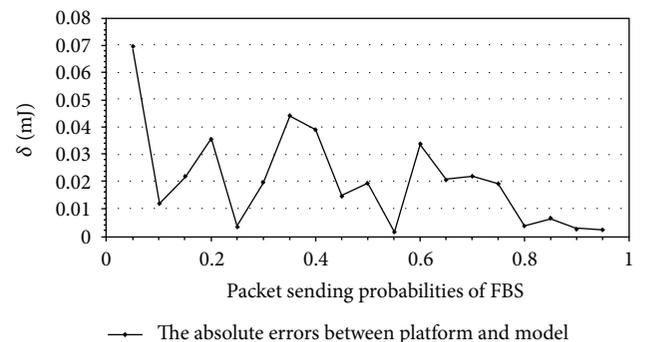
TABLE I: Single packet energy comparison and errors.

Sending type	Packet number	Platform energy (J)	Model average energy (J)	Mean error	Maximum error
7×7	6,700,208	95.6	95.8	0.18%	1.76%
5×5	5,022,234	78.3	78.2	0.07%	0.69%
3×3	2,357,410	51.2	51.3	0.14%	1.35%

and power source. There are two kinds of node sensors in the MSN system: *base station* and *movable device*. In our experiments, there are 16 *base stations* which are organized as a 4 by 4 mesh network. Each pair of neighbor base stations is connected with a 10 MHz frequency Ethernet. The positions of these base stations are stable so that the energy consumption of wireless communications between base station and movable device can be computed quickly. The movable devices in our simulator are initialed with random positions in a rectangle area. Their positions keep changing with every time interval so that the distances with their base station also vary with time. The energy consumption counting system in our simulator is designed based on packet level. When a packet passes through the network components, such as the link and router, an activity count is accumulated to compute the energy consumption for the component. If the packet arrive its target base station, the energy consumption of the wireless communication is calculated based on the current distance between the base station and the sink movable device. The similar case also happens in the packet sending action.

5.3. Accuracy Comparison of Energy Consumption. The simulation results in Table 1 shows that the stable single packet energy could be obtained without the effects from different sending probability. The different values of packets' hops only came from the different sending patterns. In 7×7 pattern, some packets are sent to farther nodes; then the average hop count is greater. While, in 3×3 pattern, all packets are limited in a small area, which leads to a shorter average transmission range. Table 1 shows the energy simulation results of the three types of sending patterns in Section 5.1. Since our simulator sends hundreds of thousands of packets to transfer the data in the 16-node MSN system, the energy results are very close to the model's forecast. Even if in the worst case, the error between the simulation's measurements and model's prediction is less than 2%. This outcome is acceptable as a validation of the energy evaluation model.

The model's stability is also important in evaluating the energy consumption of MSN. We also run the simulator with different packet sending probabilities for each type of pattern, from 0.05 to 0.95. The errors between the platform and our model are shown in Figures 4, 5, and 6. The results shows that a higher sending speed, or a larger probability, always results in less variances. This illustrates a better prediction of energy consumption. The reason is rooted in the law of large numbers. When the sending probability is higher than 0.8 each node, the simulation results stably corresponded to the model predictions and maximum errors are less than 0.2%. This results show that our model's prediction approaches the *limit* of a single packet's energy consumption.

FIGURE 4: 7×7 window's packet energy errors between platform and model.FIGURE 5: 5×5 window's packet energy errors between platform and model.FIGURE 6: 3×3 window's packet energy errors between platform and model.

We implement the static energy consumption analyser for a distributed MSN system. The analyser takes the data fetching operations as inputs to record each sensor's cache access. Starting from the records, we identify the remote data accesses from local data hit and generate the remote accessing

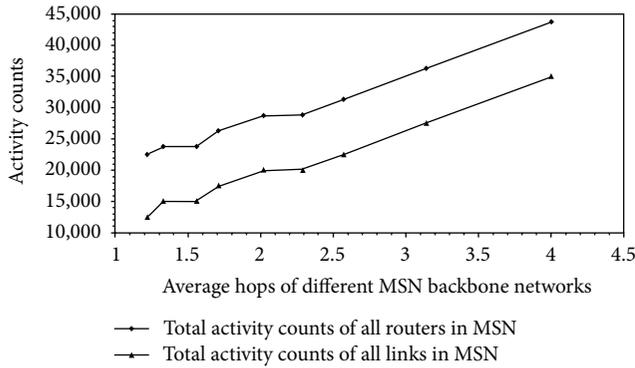


FIGURE 7: Energy consumption optimization for different transmitting hops.

list for each thread. During the energy evaluating stage, we iterate different shapes of windows and various locations of main thread with our energy model to get the best thread mapping pattern. The reason for iterating the location of master sensor is that master sensor always has the most communication with other subsensors, which intensively impacts the energy consumption of MSN. Figure 7 shows the relationship between average hop and the activity count of components, such as routers and links. We can conclude from the figure that the energy consumption has a linear increasing with the average hop of the network, which is irrelevant with the topology of network.

6. Related Works

Research works on energy model of mobile sensor network mainly focus on energy modelling and estimation. Reference [6] proposes a comprehensive energy model to estimate the overall energy consumption through power dissipation, for both static and mobile sensors in the potential sensor fields. Their model can calculate the total remaining lifetime of mobile sensor networks from the total power dissipation, and an optimum network strategy can be designed for a given application. Reference [9] gives an efficient hybrid method for message relaying and load balancing which is proposed in low mobility wireless sensor networks. Taking a mathematical approach, the system parameters are adjusted so that all the sensor nodes dissipate the same amount of energy, so that the problem of losing connectivity due to the fast power drainage of the closest node to the fixed sink is resolved. A 3D model of energy consumption for deploying nodes proposed in [10] describes a mathematical model for the power consumption of mobile node in wireless sensor networks. Each source node must send all its locally generated data to the other nodes and vice versa. To maximize mobile node's lifetime, it is essential to have optimum monitored region and radio range of each source node of wireless sensor networks. Reference [11] proposes a transmission scheme for power-adjustable radio to optimize transmit energy efficiency subject to given overflow and delay constraints. An analytical model is developed to estimate the unit energy, data throughput, and delay for a sensor node only in the single-hop case. Some

work in [12] proposes to split the lifetime of a sensor network into equal periods of time referred to as rounds and model the energy constrained routing during a round as polynomial-time solvable flow problems. The flow information from an optimum solution to a flow problem is then used as a basis for an energy-efficient routing protocol. Reference [13] proposes a mobile cluster which is applied to a vehicle equipped with a sensor node and consists of a mobile cluster head and mobile cluster members. Their analytical results show that a mobile cluster applied to the vehicle can perform data transmission using less power than direct communication applied to the vehicle. In [14], they explore an optimal barrier coverage-based sensor deployment for object tracking wireless sensor networks where a dual-sink model was designed to evaluate the energy performance of all the static sensors, static sink, and mobile sink simultaneously. Reference [15] gives a distributed target localization and pursuit scheme based on discrete measurements of the energy intensity field produced by mobile targets. In their new strategy, all robots are categorized into two groups: the leaders, responsible for the target pursuit, and the followers, responsible for the formation and connectivity maintenance. Most of these previous work have the ability to describe the activities of wireless components and get the energy consumption in each router node or link. Some of them achieve high accuracy for single device energy consumption when simulating some applications of wireless sensor network. However, they are weak in predicting the energy consumption of the overall MSN system, because they only set up their energy model on hardware layers so that the software (coherence protocol) impact on energy consumption is ignored by the model.

7. Conclusions

Energy consumption of MSNs is a vital factor in the design of a large-scale network. A low power consumption design of wireless sensor network has to satisfy heavy workload on communication and high power-efficiency at the same time. In this paper, we analyse the data fetching procedure of a 3-way cache coherence protocol and proposed a methodology to build the energy model for it. In this model, an independence assumption is provided to guarantee that any application can be seen as a sequence of data fetching action on the wireless sensor network, and the fetching procedure is decided by the first two steps of the cache protocol. Then, we use our model to evaluate any given application that transits copy of data between different node in the MSNs system. Our model is also suitable for measuring and comparing the energy consumption from different implements of hardware and software. Our experimental results shows that our approach achieves a precision with less than 2% error and provides a credible quantitative criterion for energy optimization of cache coherence protocols with MSNs system.

Acknowledgments

Jihe Wang and Bing Guo are supported in part by the National Natural Science Foundation of China under Grant nos. 61272104 and 61073045; Sichuan Science Fund for

Distinguished Young Scholars under Grant no. 2010JQ0011; the Fund from State Key Laboratory of Computer Architecture, Institute of Computing Technology, Chinese Academy of Sciences under Grant no. ICT-ARCH201003. Meikang Qiu is supported by NSF CNS-1249223 and NSFC 61071061.

References

- [1] H. C. Chen, H. L. Fu, P. Lin, and C. H. Hsu, "Energy-aware transmission scheduling in mobile sensor networks," in *Proceedings of IEEE Global Telecommunications Conference (GLOBECOM '11)*, pp. 1–15, December 2011.
- [2] M. Qiu, C. Xue, Z. Shao, M. Liu, and E. H.-M. Sha, "Energy minimization for heterogeneous wireless sensor networks," *Journal of Embedded Computing*, vol. 3, no. 2, pp. 109–117, 2009.
- [3] M. Qiu, J. Liu, J. Li, Z. Fei, Z. Ming, and E. H. M. Sha, "A novel energy-aware fault tolerance mechanism for wireless sensor networks," in *Proceedings of IEEE/ACM International Conference on Green Computing and Communications (GREENCOM '11)*, pp. 56–61, Washington, DC, USA, 2011.
- [4] S. Choudhury, S. Akl, and K. Salomaa, "Energy efficient cellular automaton based algorithms for mobile wireless sensor networks," in *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '12)*, pp. 2341–2346, April 2012.
- [5] Q. Meikang and W. Jiande, "Energy saving for memory with loop scheduling and prefetching," in *Proceedings of the 18th ACM Great Lakes Symposium on VLSI (GLSVLSI '08)*, pp. 155–158, New York, NY, USA, March 2008.
- [6] M. Tariq, M. Macuha, Y. J. Park, and T. Sato, "An energy estimation model for mobile sensor networks," in *Proceedings of the 4th International Conference on Sensor Technologies and Applications (SENSORCOMM '10)*, pp. 507–512, July 2010.
- [7] T. P. Sharma, R. C. Joshi, and M. Misra, "Dual radio based cooperative caching for wireless sensor networks," in *Proceedings of the 16th IEEE International Conference on Networks (ICON '08)*, pp. 1–7, December 2008.
- [8] O. Jerew and W. Liang, "Prolonging network lifetime through the use of mobile base station in wireless sensor networks," in *Proceedings of the 7th International Conference on Advances in Mobile Computing and Multimedia (MoMM '09)*, pp. 170–178, New York, NY, USA, December 2009.
- [9] G. F. Zaki, H. M. Elsayed, H. H. Amer, and M. S. El-Soudani, "Energy balanced model for data gathering in wireless sensor networks with fixed and mobile sinks," in *Proceedings of the 18th International Conference on Computer Communications and Networks (ICCCN '09)*, pp. 1–6, August 2009.
- [10] R. Dutta and I. Bhattacharya, "Generalize 3d model of energy consumption for deploying nodes in sensor network," in *Proceedings of IEEE Recent Advances in Intelligent Computational Systems (RAICS '11)*, pp. 124–128, September 2011.
- [11] C. Wang and P. Ramanathan, "Energy efficient transmission scheme for data-gathering in mobile sensor networks," in *Proceedings of the 3rd Annual IEEE Communications Society on Sensor and Ad hoc Communications and Networks (Secom '06)*, vol. 2, pp. 498–507, September 2006.
- [12] S. Gandham, M. Dawande, and R. Prakash, "An integral flow-based energy-efficient routing algorithm for wireless sensor networks," *Proceedings of IEEE Wireless Communications and Networking Conference (WCNC '04)*, vol. 4, pp. 2341–2346, 2004.
- [13] I. Joe and M. Shin, "An energy-efficient mobile cluster-based approach for vehicular wireless sensor networks," in *Proceedings of the 6th International Conference on Networked Computing (INC '10)*, pp. 1–5, May 2010.
- [14] J. Chen, M. B. Salim, and M. Matsumoto, "Modeling the energy performance of object tracking in wireless sensor network using dual-sink," in *Proceedings of the 16th Asia-Pacific Conference on Communications (APCC '10)*, pp. 204–209, November 2010.
- [15] J. Hu, L. Xie, and C. Zhang, "Energy-based multiple target localization and pursuit in mobile sensor networks," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 1, pp. 212–220, 2012.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

