Article:
Kabashi, AH and Elmirghani, JMH (2016) Optimal and quasi-optimal energy-efficient storage sharing for opportunistic sensor networks. Wireless Communications and Mobile Computing, 16 (14). pp. 2001-2015. ISSN: 1530-8669

https://doi.org/10.1002/wcm.2664

Optimal and Quasi-Optimal Energy-Efficient Storage Sharing for Opportunistic Sensor Networks

A. H. Kabashi and J.M.H. Elmirghani

Abstract—This paper investigates optimum distributed storage techniques for data preservation, and eventual dissemination, in opportunistic heterogeneous wireless sensor networks where data collection is intermittent and exhibits spatio-temporal randomness.The proposed techniques involve optimally sharing the sensor nodes' storage and properly handling the storage traffic such that the buffering capacity of the network approaches its total storage capacity with minimum energy. The paper develops an integer linear programming (ILP) model, analyses the emergence of storage traffic in the network, provides performance bounds, assesses performance sensitivities and develops quasi-optimal decentralized heuristics that can reasonably handle the problem in a practical implementation. These include the Closest Availability (CA) and Storage Gradient (SG) heuristics whose performance is shown to be within only 10% and 6% of the dynamic optimum allocation, respectively.

Index Terms—; Algorithm/protocol design and analysis; Distributed Systems; Dynamic Storage Sharing; Energy Efficiency; Opportunistic Connectivity; Optimization; Wireless Sensor Networks.

1 INTRODUCTION

The recent evolution in the research and development of wireless sensor networks (WSN) has made them one of the key enabling elements of the next generation heterogeneous networks; featuring ubiquitous computing, technology convergence and context-aware applications. This is enabled by advances in micro-processing, device miniaturization, wireless communications and networking techniques [1-3]. Research in this field spans a wide scope of topics including protocol design, topology control, survivability, energy efficiency and network management [1].

Due to the wide scope of the application domain of WSNs, their architectures, design requirements and modes of operation are quite diverse [1, 3]. For instance, the essential data collection and dissemination functionality can employ a variety of approaches ranging from conventional fixed base-stations to data Mules, city buses, aerial vehicles and robots [4]. As such, an instantaneous and continuous data

- *A. H. Kabashi is with the Electronic & Communication Eng. Dept., Umm Al-Qura University, KSA. E-mail: {amarkabashi@hotmail.com}.*
- *J.M.H. Elmirghani is with the School of Electronic & Electrical Eng, University of Leeds, Leeds, LS2 9JT, UK. E-mail: {j.m.h.elmirghani @leeds.ac.uk} and with King Abdulaziz University, KSA*

collection capability is not a practical assumption in many cases. This is due to either an employed intermittent data collection mode or an imposed connectivity problem in the core and edge of the network that causes the network to devolve and partition frequently. Intermittent collection might be inherent in the application context [5] or intentionally designed through sleep scheduling for energy conservation purposes [6]. Network devolution and fragmentation might arise due to radio-wave propagation impairments, mobility, failures or misbehavior [7-8]. This study proposes storage-aided operation in such highly opportunistic scenarios by optimally and dynamically sharing the distributed (nodal) storage in such a way that avoids jeopardizing the network lifetime by excessive energy consumption.

2 RELATED WORK AND CONTRIBUTION

The use of the ad hoc and sensor network's internal storage for additional functionalities has recently become feasible with the rapid advancements in non-volatile storage manufacturing techniques. For instance, the pioneering work in [9,10] shows that equipping the MicaZ WSN platform with NAND flash memory can dramatically alter the capacity and energy efficiency of local storage making its energy consumption two orders of magnitude lower than communication, and comparable to computation. This opens the door to a variety of in-network processing possibilities that were once hindered by the storage limitations.

The use of optimization techniques in networked storage systems has been around for a while within the area of file assignment and sharing in wired networks [11, 12]. The objectives here are reducing access time and flow and improving the network's robustness to disk failures, at the cost of increased memory capacity and complexity of file-consistency control. This is especially empowered by using coded storage through utilizing Maximum Distance Separable codes (MDS). Recent advancements in this area have also looked at the problem of optimal storage allocation to coded information that maximizes the probability of successful recovery from a random subset of unreliable disks [13]. Distributed data caching has been proposed for ad-hoc and peer-to-peer networks for content distribution and accessibility benefits [14].

In WSNs optimal buffering schemes have been recently proposed to help achieve power efficient operation [15]. These are local (non-networked) buffering approaches that achieve energy conservation by utilizing batched transmissions and amortize the radio wake-up energy costs among larger data units. The interesting study in [15] gives the optimum buffer sizes and buffering times for a large range of operational conditions.

This paper proposes the use of optimal distributed storage for data preservation, and eventual dissemination, in heterogeneous sensor networks with intermittent and random data collection. Overloaded nodes use other nodes in the network as temporary storage nodes until a collection opportunity arises. The goal is to find optimum storage strategies that consider energy consumption as well as data perseveration. The essential question to be answered is: given a data generation profile, how to make the buffering capacity of the network approach its total storage capacity with minimum energy. This involves finding an optimum 'storage plan' that specifies the optimum storage location for each packet generated and properly handles the storage traffic such that the distributed storage in the network is maximally utilized with the minimum possible energy. The study targets a generic sense-store-collect context where:

- Sink (collector) is not always available for continuous upload. This is either due to connectivity problems or due to scheduled data collection.
- Sink does not have a fixed location. Hence the sink can 'appear' virtually anywhere in the vicinity of the network. This incorporates applications where the access to the WSN is through a mobile agent or through a form of mobile ad-hoc network. An example of the former is agricultural sensing where data is collected by an 'E-bike' that traverses a large agricultural area to collect the data from groups of sensors [16]. An example of the latter is an environmental sensor network that exploits the movements of vehicles to upload its data [2, 4].

Hence there is an inherent spatio-temporal randomness associated with the data collection process.

Figure (1) illustrates the scenario.

The paper first develops an integer linear programming (ILP) optimization model to solve the distributed storage problem and analyses the emergence of storage traffic in the network. A comprehensive 'off-line' study is then conducted to assess different performance sensitivities. The paper then examines dynamic optimum storage sharing in a realistic WSN scenario and develops efficient decentralized heuristics that can reasonably handle the problem in a practical implementation. Finally the paper shows how the ILP model can be utilized to find performance bounds for the distributed storage problem and proposes future directions to achieve further efficiencies.

3 OPTIMIZING DISTRIBUTED STORAGE SHARING

3.1 Objectives of optimum distributed storage

The objectives of optimizing distributed storage in the mentioned application contexts are twofold. First we want to minimize data loss at overloaded nodes that have exhausted local storage. Due to the lack of a continuous upload feature (continuous sink availability and global connectivity) such overloading conditions lead to dropping new packets or overwriting old packets, resulting in data loss in both cases. To minimize data loss we try to maximum the utilization of the distributed storage in the network by using appropriate peer nodes as storage nodes.

Second, the distributed storage energy costs have to be minimized so that the network lifetime is extended. This involves finding an optimum selection of storage nodes for each piece of data (where to store?) and optimum storage routing (how to deliver?) so that the overall storage energy consumption in the network is minimized throughout its operation cycle.

3.2 Optimization model

To jointly achieve the objectives outlined in the previous section, the optimization model developed employs different concepts from the Network Optimization and Operations Research. The main concepts are summarized below:

- The optimum storage selection objective is modelled as a *minimum energy resource allocation* problem. Resource allocation has been extensively used in network optimization at different levels; e.g. routing, rate control and channel allocation - see [17, 18] for more examples.

- The optimum storage routing problem is modelled by utilizing conventional *network flow* optimization techniques [18, 19].
- The minimum data drop rate objective is modelled as a storage-supply deficit minimization problem. This ensures that the storage demand (generated data) is met as far as the storage supply (available storage capacity) allows, hence maximizes storage utilization and minimizes drop rate. Furthermore, the model drives the optimizer to prioritize deficit minimization over energy minimization by proper weighting, as will be explained next.

A complementary consideration is to optimize the final collection of data from the network. This problem has already been studied in the WSN optimization-based routing literature [20-22]. Note that the spatio-temporal randomness characterizing the targeted application context enables the separation of the data generation and storage phase from the final collection phase without loss of rigour.

3.2.1 Node energy consumption model

The relevant energy consuming components in this work are the transceivers and storage elements. For the transceiver energy we adopt a model close to the widely used radio energy dissipation model proposed in [23]. In this model the transmission energy consumption comprises of a transmitter electronics component and a transmitter's RF amplifier component. The reception energy consumption, on the other hand, is approximated by the electronics consumption only. The energy consumed by the transmitter's RF amplifier depends on the distance to the receiver (s) and the path-loss exponent(α). Hence we assume a log-distance radio propagation model and assume that the node can adaptively change its transmission power level to reach a given destination. Accordingly, a byte consumes $(\varepsilon_e + \varepsilon_a \cdot s^{\alpha})$ Joules in transmission and (ε_e) Joules in reception. The values of ε_e and ε_a used are shown in Table 1. These were calculated from the values given in [21-23].

For storage we assume the use of the recently developed energy-efficient NAND Flash technology

[9, 10]. Flash memories differ in the way they perform different operations (e.g. access, Write, Read, Erase). To reach a reasonable estimation for energy consumption of storage per byte (ε_s) we used the results in [10] to estimate storage cost such that the proportionality is preserved with our selected radio parameterization. The estimated value of (ε_s) is also shown in Table 1.

3.2.2 The ILP model

To enable modelling our optimization problem we first model the different flows traversing a node. This conceptual model of the inside of a node is shown in Figure (2) for an arbitrary node j where (h_j) denotes the data load at the node (in bytes), (d_{ij}) denotes the *storage flow* (in bytes) from node i to node j and (c_j) denotes the memory capacity of node j. The data of each node is injected to the network's storage at a single point. With reference to Figure (2), (d_{jj}) represents the amount of data of node j that is injected in the network's storage. Part of it may be stored locally and the rest can flow out of the node to be stored externally. Hence (d_{jj}) represents the total network's "storage supply" to node j.

Based on this, the per-node components of the objective of the optimization model are:

1- The energy consumption of node j's receiver; expressed as: $\left(\sum_{\forall i\neq j}(\varepsilon_e)\cdot d_{ij}\right)$.

2- The energy consumption of node j's transmitter; expressed as: $\left(\sum_{\forall i\neq j}\left(\varepsilon_e+\varepsilon_a.s_{ji}^{\alpha}\right)\cdot d_{ji}\right)$.

3- The energy consumption of node j's Flash memory; expressed as: $\left(\left(\sum_{\forall i} d_{ij}-\sum_{\forall i\neq j} d_{ji}\right)\cdot\varepsilon_s\right)$.

4- The amount of lost data of node j ; expressed as: $\left(h_j-d_{jj}\right)$.

The Optimum Distributed Storage Allocation ILP Model (ODSAM), jointly minimizing **storage energy** and **data loss**, is thus constructed as follows:

Minimize:

$$\mathcal{F}=\sum_{\forall j}\left(\left(\sum_{\forall i\neq j}(\varepsilon_e)\cdot d_{ij}\right)+\left(\sum_{\forall i\neq j}\left(\varepsilon_e+\varepsilon_a.s_{ji}^{\alpha}\right)\cdot d_{ji}\right)\right)+\sum_{\forall j}\left(\left(\sum_{\forall i} d_{ij}-\sum_{\forall i\neq j} d_{ji}\right)\cdot\varepsilon_s\right)+\lambda\cdot\sum_{\forall j}\left(h_j-d_{jj}\right)\quad(1)$$

Subject to:

$$\sum_{\forall i} d_{ij}-\sum_{\forall i\neq j} d_{ji}\leq c_j\quad ;\forall j\qquad(2)$$

$$d_{jj} \leq h_j;\ \forall j \tag{3}$$

$$\sum_{\forall i} d_{ij} - \sum_{\forall i \neq j} d_{ji} \geq 0 \quad ;\forall j \tag{4}$$

$$d_{ij} \geq 0;\quad \forall i, \forall j \tag{5}$$

$$d_{ij} \in \mathbb{Z}^{+};\quad \forall i, \forall j \tag{6}$$

($i, j \in \boldsymbol{V}$, where $\boldsymbol{V}$ denotes the set of nodes in the network, $|\boldsymbol{V}| = N$).

Table 2 outlines the symbols used in ODSAM.

The constraint in (2) represents the *storage load* at node j which is upper-bounded by the storage capacity. It is important to distinguish between the *data load* of a node which is the amount of generated data at the node that needs to be stored somewhere in the network, and the *storage load* at the node which is the amount of data granted storage at the node and which might come from different sources. Constraint (3) ensures that the supply doesn't exceed the demand. In (4) the storage flow is conserved at each node, while considering possible storage at the node, hence the inequality. Constraints (5) and (6) are the non-negativity and integerality constraints of the flows, respectively.

The objective function is composed of three components with direct physical correspondence. The first term aggregates the total energy consumed by the transceivers in carrying the total storage traffic (including both transmission and reception). The second term calculates the total storage energy consumption. The third term aggregates the total deficit in meeting the storage demands. Constant λ is a prioritization weighting factor and will be discussed next. The number of variables to be evaluated by this model is N^2, where N is the number of nodes in the network.

As any storage flow d_{ij} will be counted twice in the first term of (1); one in the transmitter cost of node i and another in the receiver cost of node j, we can rewrite the first term in the objective in a more compact form by considering the flow cost rather than the nodal cost. Furthermore, the second term can be simplified by considering the storage supplies rather than storage loads. The simplified objective can now be rewritten as follows:

$$\mathcal{F} = \sum_{\forall i} \sum_{\forall j \neq i} \left(\left(2\varepsilon_e + \varepsilon_a . s_{ij}^{\alpha}\right) \cdot d_{ij} \right) + \varepsilon_s \cdot \sum_{\forall j} d_{jj} + \lambda \cdot \sum_{\forall j} \left(h_j - d_{jj}\right). \quad (7)$$

The weighting factor (λ):

Parameter λ is the weighting factor that drives the optimizer to prioritize storage-supply deficit minimization (i.e. data loss minimization) over energy minimization. For the prioritization to work properly, the weighting factor λ has to be greater than the largest value of the summation of the first two terms in the objective. This will automatically drive the optimizer to minimize the deficit before minimizing energy as any positive deficit will appear as a higher cost than any possible energy cost.

Note that the maximum flow that can be received by a node j is $\sum_{\forall i \neq j} h_i$, and the maximum flow that can leave a node (assuming the node doesn't store any data) is $\sum_{\forall i} h_i$. Note also that the total storage load of the network is equivalent to $\sum_{\forall i} h_i$. Consequently we can parameterize λ as:

$$\lambda \geq \sum_{\forall j} \left((\varepsilon_e) \cdot \sum_{\forall i \neq j} h_i \right) + \sum_{\forall j} \left((\varepsilon_e + \varepsilon_a . s_{max}^{\alpha}) \cdot \sum_{\forall i} h_i \right) + \varepsilon_s \cdot \sum_{\forall i} h_i.$$

Relaxing the restriction ($i \neq j$) we can safely select the value of λ as:

$$\lambda = \left(\sum_{\forall i} h_i \right) \cdot \left(N \cdot (2\varepsilon_e + \varepsilon_a . s_{max}^{\alpha}) + \varepsilon_s\right) \quad (8)$$

where s_{max} is the largest distance between two nodes in the network. It is worth noting that (λ) is dimensioned in (Joules/byte) in order to maintain the physical consistency of the objective function.

4 OFF-LINE STUDY

In this section we will utilize the developed ILP model ODSAM to provide a quantitative analysis of the distributed storage problem so as to gain insights into the different sensitivities, trends and performance bounds. This will be an off-line static study where the optimization model is fed aggregate loads at different sites (nodes). The off-line approach enables us to capture more fundamental limits and basic behaviors. Later in Section 5 we will consider a more realistic dynamic context where instantaneous loads are considered and decentralized solutions (heuristics) are developed.

4.1 Emergence of storage traffic (ρ) and its analytical bounds

Before proceeding with the off-line study it is worth making a basic note about the storage traffic and its occurrence in the network. For storage traffic to emerge a positive storage gradient has to build up in the network. That is; some nodes' storage has to be overloaded while others underutilized. This implies that when all nodes generate data at the same rate, hence simultaneously reaching their storage capacity, no storage traffic will occur (unless some sort of controlled overwrite is allowed at some nodes). Hence heterogeneity in data generation is implicit here.

With this in mind, we can develop an analytical expression for the lower bound on storage traffic in the network. Once storage traffic emerges in the network, then it can be calculated and lower bounded as follows:

$$\rho = \sum_{\forall i} \sum_{\forall j \neq i} d_{ij}, \qquad (9.a)$$

$$\rho \geq \min\left\{ \sum_{\forall j: h_j > c_j} (h_j - c_j), \left| \sum_{\forall j: h_j < c_j} (h_j - c_j) \right| \right\}. \qquad (9.\mathrm{b})$$

Note that if the storage gradient doesn't exist then there is no storage traffic and equation (9.b) is not applicable.

The lower bound in (9.b) follows from the fact that the basic amount of data responsible for the emergence of storage traffic is the minimum of either the data overload or the storage surplus over the whole network. The actual traffic may diverge from the lower bound set in (9.b) mainly due to multi-hopping in delivering the storage traffic. As will be shown later, multi-hopping might become superior under certain conditions. Another reason for the possible increase is the non-optimal selection of storage nodes which leads to the eventual situation where the storage nodes themselves be in need for external storage.

4.2 Investigation Methodology

For this study we generate random network instances by distributing N nodes uniformly at random in a square area of dimensions ($L1 = L2 = L$). A set of data loads $\{h_j\}$ is generated randomly and assigned to the deployed nodes. The set of loads is generated such that its summation exactly packs the total storage in the network, (i.e. $\sum_{j=1}^{N} h_j = N \cdot c$), while the individual loads take arbitrary values to mimic the rate heterogeneity in the network.

The study investigates the following performance metrics and assesses their sensitivity to different operational parameters:

- **Total energy consumption** (E_{total}): this is calculated from the inter-nodal storage flow matrix $\boldsymbol{d}$ as follows:

$$E_{total} = \sum_{\forall i} \sum_{\forall j \neq i} \left(\left(2\varepsilon_e + \varepsilon_a . s_{ij}^{\alpha} \right) \cdot d_{ij} \right) + \varepsilon_s \sum_{\forall j} d_{jj} . \qquad (10)$$

- **Storage traffic (**ρ**):** as calculated from (9.a).
- **Average storage hop-count**(ϑ): this metric characterizes the multi-hopping behaviour in routing the storage traffic. It is estimated as follows:

$$\vartheta \triangleq \frac{total\ storage\ flow}{total\ over\_capacity\ load} = \frac{\sum_j \sum_{\forall i \neq j} d_{ij}}{\sum_{\forall j: h_j > c_j} \left(h_j - c_j \right)} \qquad (11)$$

The estimation in (11) is based on the fact that the original stimulus for inter-nodal storage flow is data overload at specific nodes. Multi-hopping is implied whenever the aggregate storage flow exceeds the aggregate over-capacity data load in the network. The ratio of the two aggregate values gives a good estimation for the average storage hop-count.

Max-to-average energy consumption ratio is also analysed in the study. Furthermore, **optimum storage distribution topologies** are also investigated, but not shown here for brevity. These are represented by the directed graphs resulting from taking the 'sign' of the inter-nodal storage flow matrix. They give a pictorial representation of how storage is selected and storage flow is routed over the network. Furthermore they also help superimpose energy consumption and data loading over the spatial characteristics of the network (e.g. node locations). Some insights from this metric will be briefly considered in the discussion.

The operational parameters considered include Network size($N, L1, L2$), path-loss exponent (α) and different loading effects. The first part of the investigation is conducted on a reference network generated by randomly and uniformly distributing 20 nodes over an area of initial dimensions of 100x100 metres and then scaled up to 500x500 metres. The purpose of this part is to make an initial evaluation of the performance sensitivities and trends. The second part is done on a large number of random realizations in order to get more statistically representative outcomes. Collective results are shown in Figure (3) and are discussed next. Figure (3.a) through Figure (3.f) also show the results of a restricted routing case where the ILP model of section 3.2.2 is amended by adding the following constraint:

$$\sum_{\forall j \neq i} d_{ij} \leq max\{0, (h_i - c_i)\}; \qquad \forall i. \qquad (12)$$

This constraint allows a node to send storage traffic out only if it suffers data overload and for as much as its own overload only. This restricts the emergence of storage routers and enables quantifying the effect of storage routing on other performance metrics.

4.3 Results and discussion

The effect of network size and path-loss exponent on the total network's energy consumption well resembles the $\propto (s^{\alpha})$ relationship of energy dissipation at the transceivers. Figure (3.a), where $\alpha = 2$, clearly reflects the quadratic increase of energy consumption with network dimensions, while Figure (3.c) shows a more rapid exponential increase with path-loss exponent.

These figures also reflect the fact that restricted routing cases consume more energy, and this is more evident when the network dimensions and/or path-loss exponent are large. Under these conditions single-hop routing loses its superiority from an energy point of view.

By looking at the storage distribution topologies (not shown here for brevity) we recognize an interesting pattern. As the network size and path-loss exponent increase, the storage topology itself evolves indicating changes in the optimum storage solution. Part of this change is a gradual migration from single hop routing to multi-hopping and the emergence of storage routers. This is due to the single hop routing becoming more expensive as the link length or path-loss exponent grows, and hence multi-hopping becomes superior in energy consumption. This effect is fully captured in the storage hop-count results Figures (3.b & 3.d) where we recognize the slow increase in average hop-count from 1 to

2. A closer analysis of the resulting storage topologies shows that, in addition to multi-hopping, the optimum solution might adaptively change its storage selection decisions resulting in a change in the list of storage nodes which serve an overloaded node. Hence the changes in network dimensioning or propagation characteristics make the optimum distributed storage solution update both the storage selection decisions and routing decisions.

Figures (3.e & 3.f) reflect the effects of network dimensioning and path-loss exponent on the generated storage traffic. The lower bound on traffic developed in (9.b) is also depicted in these figures. It is clear from these figures that storage traffic for the optimum (unrestricted) case grows rapidly with both network dimensions and path-loss exponent, hence diverges from the lower bound. The trend of traffic growth follows that of average storage hop-count, shown in Figures (3.b & 3.d). On the other hand the restricted routing cases converge to the lower bound throughout. These observations render multi-hopping as the main cause for storage traffic growth in the optimum (unrestricted) cases. Note that despite traffic growth, the unrestricted cases keep their superiority in terms of energy consumption, as observed previously from Figures (3.a & 3.c). It can be concluded that the energy consumption of distributed storage is governed by the compound effect of storage traffic superimposed on both network geometry and propagation conditions. Consequently it is for the general good of the network, in terms of distributed storage, to have the heterogeneous data rates well interleaved spatially so that storage availability is globally maintained as close as possible to overloaded nodes and overloaded zones are avoided.

To get a more statistically representative picture further experiments were conducted over a large number of random realizations. Appropriate results are shown in Figures (3.g-h). For this set of results the experiments were run over 1000 randomly generated networks (i.e. 1000 random node distributions over 100m x 100m area) and associated 1000 randomly generated load sets$\{h_j\}$. The obtained results were then averaged over all realizations.

Figure (3.g) shows the averaged results of the average storage hop-count with both α and N. The hop-count steadily increases with α, agreeing with the previous results on our reference network, and the rate of increase generally drops as α gets sufficiently large. The new insight gained from Figure (3.g) is the behaviour of the storage hop-count with N, where this result gives a more comprehensive idea about the effect of network density on hop-count than that given before by Figure (3.b) when the network size was changed. A fall in hop-count with increased node density can be seen in this result, agreeing with the previous result in Figure (3.b) - (note that an increase in the network dimensions can also be viewed as a decrease in node density). However, Figure (3.g) shows the additional observation that this relationship is not strictly preserved for all conditions. The curves at higher values of α show that the, more accurate, trend is a sharper increase followed by a long-tailed decrease, resulting in a peaking behaviour at intermediate values of node density. The peak occurs at different values of N that are dependent on, and increase with, path-loss exponent; $\{(N = 5\ or\ less, \alpha = 2), (N = 10, \alpha = 2.4), (N = 15, \alpha = 2.6), (N = 20, \alpha = 2.8), (N = 30, \alpha = 3)\}$. This behavior can be explained as follows: In both the sparse and sufficiently dense cases the distance to a storage node is comparable to the distance to potential router nodes, on average, resulting in the superiority of less hopping. In the intermediate cases, on the contrary, there are higher effective differences among the inter-nodal distances in the overloaded-storage zones. This gives multi-hopping more degrees of freedom to emerge. Note that uniform node distribution is dealt with here. Another potential reason (especially for the dense case) is a better spatial data rate interleaving which keeps storage availability closer to storage need.

Figure (3.h) shows the ratio of max-to-average energy ratio (MAER) among the deployed nodes. The ratio increases with both N and α and ranges from 2 in the low (N, α) corner to over 7 in the high corner. The increase is relatively slow with N and more rapid with α. Load heterogeneity is a basic cause for this relatively high MAER. Other causes include locations of storage availability, as compared to location of storage need, and how the storage traffic is routed. Note that in our optimization formulation we have focused on minimizing the overall energy consumption and this might yield considerable variation in the nodal energy consumption and an increased MAER. An interesting extension to this work might be including some metric of uniformity or fairness in nodal energy consumption in the optimization model and then quantifying the resultant loss of efficiency in the overall energy consumption.

5 DYNAMIC OPTIMUM STORAGE SHARING

This part introduces two extensions. First the optimization model developed for the distributed storage problem (***ODSAM***) is used in a realistic dynamic scenario, rather than the static off-line scenario used previously. Using the model dynamically, where the model is fed instantaneous storage demands and availabilities, enables achieving a dynamic optimum allocation of the distributed storage. In fact this is the practical way where the optimization model can be envisioned to run in a network. The previous theoretical off-line study assumes that the aggregate loads can be known a-priori and hence exploits additional knowledge and gives performance bounds rather than practical results. A comparison between both approaches will be presented at the end of this section.

Second, decentralized efficient heuristics are developed for the storage problem and compared to the optimum solution. These heuristics represent the practical solution that can readily be employed in a real network, rather than ILP model which needs a centralized coordinated operation and whose complexity can be prohibitive in a sensor node.

5.1 Dynamic setting and data generation model

In the dynamic scenario data generation follows a discrete model where a packet generation is allowed only at discrete time epochs along the time dimension. Depending on the current data generation rate at each node a new data packet originates at appropriate time epochs. At each epoch, nodes specify their instantaneous loads depending on whether or not a data packet is generated at the node, according to the current rate. This set of instantaneous loads $\{h_j^\tau\}$ along with the instantaneous storage availabilities $\{c_j^\tau\}$ are fed to the model ***ODSAM*** and the optimization result is then used to update the storage availabilities according to:

$$c_j^{\tau+1} = c_j^{\tau} - \left(\sum_{i=1}^{N} d_{ij}^{\tau} - \sum_{i=1, i\neq j}^{N} d_{ji}^{\tau} \right) \tag{13}$$

where $c_j^0 = c$. The scenario then continues with the instantaneous flows d_{ij}^{τ} accumulated in the global inter-nodal flow matrix $[d_{ij}]$such that:

$$d_{ij}(t) = \sum_{\tau=0}^{t} d_{ij}^{\tau}. \tag{14}$$

This matrix is eventually used to calculate the results. These steps are summarized in Figure (4).

Hence the dynamic optimum allocation approach yields a centralized coordinated data storage plan that jointly minimizes energy and drop rate on a *per epoch* basis.

5.2 Decentralized distributed-storage heuristics

To reduce the complexity and centrality drawbacks of the ILP solution, decentralized heuristics are developed here. Decentralized heuristics enable nodes to take independent decisions without the need for a central coordinator. Despite this, nodes may coordinate in exchanging information that helps all of them take better decisions independently. Although at the price of accuracy, the use of heuristics in real communications systems is in many cases the only way for practical implementation [17, 18].

In the following sections two heuristics are presented and thoroughly discussed.

5.2.1 Closest Availability Single-Hop (CASH)

CASH is a simple heuristic that elects the closest node with storage availability as the storage node for each newly generated packet at an overloaded node. The intuition behind CASH is that delivery costs dominate the distributed storage costs and hence CASH tries to minimize the costs by storing as near as possible.

CASH can be summarizes as follows:

- **Decision policy**: Select the closest node that has storage availability.
- **Metric**: inter-node distances.
- **Information needed**:

1- Distances to other nodes. This can be estimated through various localization approaches or by simpler signal strength approaches (e.g. Radio Signal Strength Indicator RSSI).

2- Storage availability at all nodes. This can be collected through continuous declaration (broadcast) of storage loads. The energy cost of exchanging this information can be minimized if it is integrated with the various broadcast operations used in conventional routing protocols in ad-hoc and sensor networks. However, more efficient approaches can use load estimation locally by exploiting rate information. Here nodes exchange only their rate information and only when rate change happens.

Hence CASH consists of three functional blocks; **distance estimation block**, **load estimation block** and **storage selection block**. The execution of CASH is illustrated in Figure (5).

5.2.2 Storage Gradient Single-Hop (SGSH)

One of the potential deficiencies in CASH is the blind selection of storage closest availability without considering the amount of storage load in the selection metric. The problem with this approach is that some selected storage nodes might soon become overloaded themselves and hence re-generate storage traffic. This increases the total storage traffic in the network and drives the total energy consumption high. A better strategy would then be a more storage-load-aware mechanism that jointly considers distance and amount of current storage loads at the potential candidates. The storage gradient heuristic (SGSH) pursues this approach. It favours closer nodes with higher storage availability. SGSH is summarised in the following points:

- **Decision policy**: select the node with highest storage gradient from the source node.
- **Metric**: storage gradient.

The storage gradient is defined as the ratio of storage load difference and inter-nodal distance. The storage gradient from node i to node j at time τ is expressed in a normalized form as:

$$g_{ij}^{\tau} = \frac{\left((\bar{c}_i^{\tau} - \bar{c}_j^{\tau})/c\right)}{\left(s_{ij}/\sqrt{(L_1^2 + L_2^2)}\right)} \tag{15}$$

where (L_1, L_2) are the dimensions of the deployment area, $(\bar{c}_i^{\tau})$ denotes the instantaneous storage load and (c) denotes the maximum memory capacity in the network. Note that the sub-divisions in the numerator and denominator of (15) is for normalization purposes which produces a metric independent of the exact values of the operational parameters: L_1, L_2 *and* c. Note also that the definition of the storage gradient in (15) is similar to that mentioned in Section 4.1 with an added weighting by distance reciprocal.

- **Information needed**: same as CASH.

Refining the metric of SGSH:

The storage gradient as expressed in (15) can be better refined by appropriate dimensioning and weighting. In its current form the metric suffers from two potential weaknesses. First the numerator features rapid variations due to the continuously evolving operation of the network. Second the contributions from both sub-metrics (storage load and distance) are kept equal, hence not allowing the distance factor to stabilize and drive the overall metric in a proper way. Hence we need to compress the storage load contribution in a way that yields a diminishing return behavior (i.e. giving high storage load differences less 'gain' than smaller ones). On the other hand the distance contribution has to be magnified with inverse proportionality (i.e. giving short distances more 'gain' than longer distances). In this way we guarantee that distance (especially shorter one) is properly weighted and at the same time large storage-load differences are not allowed to mislead the metric.

Different approaches can be adopted to implement the suggested modifications. A good choice is to use a $(log(x))$ function for the storage load compression and, given that the distance appears in the denominator, use a $(\sqrt[r]{x})$ function for the distance. The refined storage gradient metric is expressed as:

$$\breve{g}_{ij}^{\tau} = \frac{log\left(1 + \left(\left((\bar{c}_i^{\tau} - \bar{c}_j^{\tau})/c\right)\cdot f_1\right)\right)}{\sqrt[r]{\left(\left(s_{ij}/\sqrt{(L_1^2 + L_2^2)}\right)\cdot f_2\right)}}. \tag{16}$$

In (16) we have also included scaling factors (f_1 and f_2) to enable projecting the normalized values properly on the x-axis of the selected functions[1]. The values of these factors will be a design choice learned through experimental experience. Generally the best values of these constants would be those maximally supporting the compression/magnification effects of the functions used. The value of r is another design value. It is intuitively expected that we will need to use lower r values for higher path-loss exponents in order to disadvantage larger distances (less weight). SGSH heuristic is functionally summarized in Figure (6).

To avoid data collisions at a storage node, due to simultaneous transmissions, or data loss due to unsuccessful delivery, standard hand-shaking mechanisms are proposed for both heuristics.

5.3 Results and discussion

Using the dynamic data generation model highlighted in Section 5.1, a dynamic scenario was simulated and used to evaluate the relative performance of the optimum distributed storage model and the sub-optimum heuristics developed. The general parameters used in these experiments are typical of those used before in Table 1. The path-loss exponent was set at ($\alpha = 2$) and the number of nodes at ($N = 10$) for these experiments. The optimum model was employed within the simulator as highlighted in Figure (4), while the heuristics were implemented according to Figures (5-6). The resulting global inter-nodal flow matrices, as calculated through (14), were used to obtain the results presented next. In addition, the averaged results of runs over large number of random realizations are also presented at the end of the section. The experiments assess the performance sensitivities to different factors including data load distributions, network scaling and network density.

Figure (7.a) shows the total energy consumption of distributed storage, as calculated from the inter-nodal flow matrices through (10), for different load settings. For all of the 7 loading cases the total amount of data generated across the network ($h_{total} = \sum_{i=1}^{N} h_i$) is kept constant while arbitrary varying the nodal data loads (h_i) from one case to the other. This is to get a fair idea about relative performance over a wide range of load distributions. The figure compares the performance of the heuristics

[1] Recall that the storage gradient initial values were normalized to the interval [0-1].

(CASH, SGSH) to that of the dynamic optimum ILP solution (opt_dyn). The performance of the direct storage gradient heuristic without the refinement introduced in (16) is also shown.

This result reflects a very good performance for both CASH and SGSH heuristics as they are both capable of approaching the optimum performance to a good extent. The SGSH heuristic is, however, consistently superior to the CASH heuristic. The direct storage gradient heuristic (SGSHD), using metric (15) directly, on the other hand, shows severely worse performance. While it is still able to mimic the general trend of the optimum case, its gap from the optimum energy consumption is large. This verifies the reasons we had when refining the SG metric to (16). We will later present a proper quantification of the relative performance gaps.

The network scaling effect is shown in Figure (7.b) where the deployment area side length was varied from 10m to 1000m. This result shows that as the network dimensions are scaled the heuristics performance diverge from the optimum. In our earlier off-line study, Figure (3.b), network scaling resulted in an increased average storage hop-count. To check this for the current experiment, Figure (7.c) plots the average storage hop-count for the dynamic optimum solution. The rise in hop count with network scaling is clear.

This should have contributed to the divergence observed as the developed heuristics employ direct single hop delivery only. To quantify the effect of this, a multi-hopping extension was added to the developed heuristics such that, after storage node selection, data is delivered through minimum energy routes. This is done on a per-hop basis where each node elects best router node through which 2-hop routing is better than direct delivery. The same is repeated at each intermediate router until reaching the storage node. The new results are also shown in Figure (7.b) where we see that the multi-hopping capability (CAMH, SGMH) has enabled the heuristics' performance to converge back towards the optimum consistently. This indicates that it was the routing part that has caused the divergence, not the storage selection part.

To demonstrate the benefits of using distributed storage, Figure (8) compares the performance of the developed heuristics to that of a simple local storage policy where overloaded nodes overwrite older (undelivered) data. The results were obtained by using an aggregate data load that starts at around 80% of the total network's storage and then increased gradually. Individual data loads were assigned randomly to the deployed nodes. In terms of data drop rate, Figure (8.a) reflects the considerable gains achieved by employing distributed storage, especially when the aggregate data load is below the network's capacity. Due to the lack of an external storage capability, a local storage policy loses much of the network's data even when the aggregate data load is less that the network's capacity, due to the existence of overloaded nodes. Figure (8.b) compares the energy consumption, where the price of external storage is clear (recall the difference between the energy cost of storage and radio as shown earlier in Table I).

To characterize the performance of the developed heuristics we conducted statistical quantification of the performance gaps from the optimum. To achieve this, the experiment was run over 100 randomly generated networks (i.e. 100 random node distributions over 100m x 100m area) each with a corresponding random load distribution. For each realization the energy consumption performance gaps were calculated and then averaged over all realizations. The whole experiment was then repeated for a number of node densities. The resulting percentages of the averaged gaps are shown in Figure (8). These results show that the heuristics yield a distributed storage performance that is only 10% away from the optimum, on average. The sub-optimality of the Closest Availability heuristic's average performance is within the range (8%-10%) while that of the Storage Gradient is only (3%-6%).

5.4 Notes on SG heuristic design constants

The constants incorporated in the metric of the storage gradient heuristic (r, f_1, f_2) were said to be design parameters learned through experimental experience.

To this end, Table 3 summarizes the best values obtained by looking into a large range of values and combinations. These are not claimed to be absolute optimum values, as the search was not meant to be strictly exhaustive, but they rather provide a general guide and 'rule of thumb'. As seen in the table, the values of these constants were found to be sensitive to the value of path-loss exponent used. Higher values of α favour lower values of r, and f_1. This confirmed our previous expectation regarding r, as the influence of the distance factor has to be reduced at higher α values. This is done through using root functions with lower r values (recall that distance contributes to the denominator of the metric).

The observed trend for the scaling factor f_1 can be explained by noting that higher values of this factor lead to further 'activation' of the log compression behavior in the numerator. Table 3 implies that this behavior is more desirable in low path-loss cases where the distance contribution to the SG metric is more stressed. In higher path-loss exponent cases, on the other hand, the compression behavior is better marginalized and this is achieved by using the more linear part of the log function (low f_1 value).

5.5 Performance bounds

Figure (7) also shows bounding performance results which were obtained by using the ODSAM ILP model in a static 'off-line' manner, similar to that of Section 4. Obviously the off-line approach uses more information than the dynamic one. That is; in the off-line approach we feed the optimizer the aggregate data loads at each node and hence enable it to take better decisions and yield an optimum global storage plan. For the dynamic approach, as in a realistic setting, this information is not known a-priori and hence the optimizer yields the optimum network-wide storage plan for each time epoch, given current conditions. The heuristics try doing the same in a decentralized way. Hence the off-line static solution is expected to yield absolute lower bounds for the performance. The appropriate results in Figure (9) confirm this expectation as the off-line solution always lower-bounded all other results.

The off-line static approach is obviously impractical. However, the performance gap seen in these results indicates that there are still significant savings that might be achieved if more knowledge about load evolution can be incorporated in the dynamic approaches. This opens the door for future extensions to this work by investigating estimation-based techniques.

6 Conclusions

This paper has studied the problem of optimum storage sharing in heterogeneous sensor networks where data collection is spatio-temporally opportunistic. The study focused on finding optimum storage strategies that consider energy consumption as well as data preservation. Static and dynamic optimum performances have been thoroughly investigated and decentralized sub-optimal heuristics were proposed. Moreover, the emergence and bounds of storage traffic have been characterized analytically and the overall performance has been assessed under different conditions.

The storage traffic was shown to be initiated by the emergence of storage gradient in the network, and further driven by the dynamics governing the optimality of energy consumption and storage selection decisions. The effect of network size and path-loss exponent on the total energy consumption well resembled the relationship of energy dissipation at the transceivers. Changes in the operational parameters were found to cause the optimum distributed storage solution to update both the storage selection decisions and routing decisions. The energy cost of distributed storage was found to be determined by the compound effect of storage traffic superimposed on both network geometry and propagation conditions. For the benefit of the network, in terms of distributed storage, it is better to have the heterogeneous data rates well interleaved spatially. The results indicated that while optimum distributed storage achieves minimum total energy consumption it might introduce non-uniformities in the energy budget of the nodes. Max-to-average energy ratio can also reach considerable values.

The developed heuristics have shown very good performance compared to the dynamic optimum solution, especially when having the multi-hop routing capability. Their sub-optimality incurs an additional energy consumption of within only 10% for the Closest Availability heuristic and 6% for the Storage Gradient heuristic, on average. Absolute performance bounds indicated that further energy savings might be achieved if more knowledge about load evolution can be incorporated proactively, through techniques such as load estimation.

REFERENCES

[1] H. Karl and A Willing, Protocols and Architectures for Wireless Sensor Networks, Wiley, 2005.
[2] I. Akyildiz et al., "Wireless sensor networks: a survey", *Computer Networks*, 2002. 38(4): p. 393-422.
[3] P. Baronti et al, "Wireless sensor networks: a survey on the state of the art and the 802.15.4 and ZigBee standard", *Comp. Commun*, 2007, 30, 1655-1695.
[4] M. Di Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey." *ACM Trans. on Sensor Net.*, Vol.8, no. 1, 7, August 2011.
[5] Ho, M. and K. Fall. Delay Tolerant Networking for Sensor Networks. In IEEE Conference on Sensor and Ad Hoc Communications and Networks (SECON 2004).
[6] N. A. Pantazis, D. D. Vergados, "A survey on power control issues in wireless sensor networks," *IEEE Communications Surveys & Tutorials*, vol. 9, issue 4, fourth quarter 2007.
[7] A. Kabashi and J. Elmirghani, "The Devolution Effects of Flat Fading on Connected Wireless Networks under Shadowing", in Proc. IEEE IWCMC'11, Istanbul, Turkey, July 2011.
[8] F. Xing and W. Wang, "On the survivability of wireless ad hoc networks with node misbehaviors and failures."*IEEE Transactions on Dependable and Secure Computing*, 2010, 7(3): pp. 284-299.
[9] G. Mathur et al, "Ultra-low power data storage for sensor networks", *ACM Trans. On Sensor Net.*, Vol.5, no. 4, 33, Nov 2009.
[10] Y. DIAO et al, "Re-thinking data management for storage-centric sensor networks." 3rd Biennial Conference on Innovative Data Systems Research (CIDR'07).
[11] L. W. Dowdy and D. V. Foster, "Comparative models of the file assignment problem'', *Computing Surveys*, vol. 14, no. 2, pp. 287-313, 1982.
[12] M. Naor and R. M. Roth, "Optimal file sharing in distributed networks", *SIAM J. Comput.*, vol. 24, no. 1, pp. 158-183, 1995.
[13] D. Leong, A. G. Dimakis, and T. Ho, "Distributed Storage Allocations," *IEEE Trans. Inf. Theory, vol. 58, no. 7, July 2012*.
[14] Tang, B., H. Gupta, and S. Das, Benefit-Based Data Caching in Ad Hoc Networks. *IEEE Transactions on Mobile Computing*, 2008. 7(3): p. 289-304.
[15] Y Ling, C. Chen and S. Chen, "Analysis of Power-aware Buffering Schemes in Wireless Sensor Networks", *ACM Trans. On Sensor Net.*, Vol.7, no. 3, 26, Sept. 2010.
[16] A. Kabashi and J. Elmirghani, "A Technical Framework for Designing Wireless Sensor Networks for Agricultural Monitoring in Developing Regions", In Proc. IEEE NGMAST'08, Cardiff, Wales, UK 16-19 Sept 2008.
[17] M. Resende and P. Pardalos (Eds), Handbook of Optimization in Telecommunications. New York: Spinger Science, 2006.
[18] S. Shakkottai and R. Srikant, "Network Optimization and Control", Foundations and Trends in Networking, Vol. 2, No. 3 (2007), pp 271–379.
[19] M. Pioro and D. Medhi, Routing, Flow and Capacity Design in Communication and Computer Networks. San Francisco, CA: Morgan Kaufmann, 2004.
[20] F. Ordonez and B. Krishnamachari, "Optimal information extraction in energy-limited wireless sensor networks," *IEEE J. Select. Areas Commun.*, vol. 22, no. 6, pp. 1121–1129, 2004.
[21] N. Sadagopan and B. Krishnamachari, "Maximizing data extraction in energy-limited sensor networks," in Proc. IEEE INFOCOM 2004.
[22] W. Ye and F. Ordonez, "A sub-gradient algorithm for maximal data extraction in energy-limited wireless sensor networks," in Proc. IEEE WirelessCom 2005.
[23] W. R. Heinzelman et al, "Energy efficient communication protocol for wireless micro-sensor networks." Proceedings of IEEE Hawaii International Conference on System Sciences (2000).

TABLE 1

NODE ENERGY CONSUMPTION PARAMETERS.

Tx amplifier energy cons. (ε_a)	800 pJ/byte/ m^α
TxRx electronics energy cons. (ε_e)	400 nJ/byte
Storage energy consumption (ε_s)	4 pJ/byte

TABLE 2

KEY OF SYMBOLS USED IN ODSAM MODEL

Symbol	Representation	Unit
ε_e	TxRx electronics energy consumption	J/byte
ε_a	Tx amplifier energy consumption	J/byte/ m^α
ε_s	Memory energy consumption	J/byte
d_{ij}	Storage flow from node i to j	byte
h_j	Data load at node j	byte
c_j	Memory capacity of node j	byte
s_{ij}	Euclidean distance between two nodes	m
s_{max}	largest distance between two nodes	m
α	Path-loss exponent	-
N	Number of nodes in the network	node
λ	Weighting factor	J/byte

TABLE 3

SELECTED VALUES FOR SG HEURISTIC'S CONSTANTS.

Constant	$\alpha = [2 - 2.5]$	$\alpha = [2.6 - 3.0]$
r	3.2	0.01
f_1	10000	10
f_2	1	1

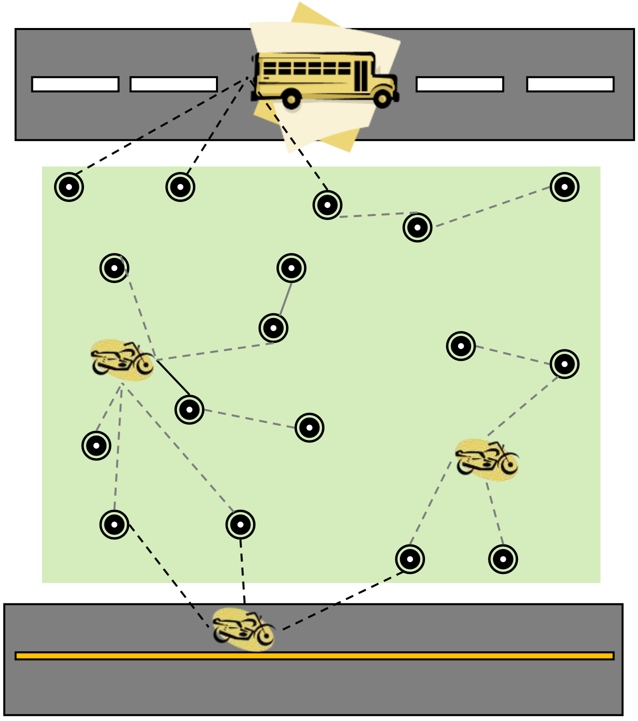


Figure 1. Opportunistic Data Collection Scenario

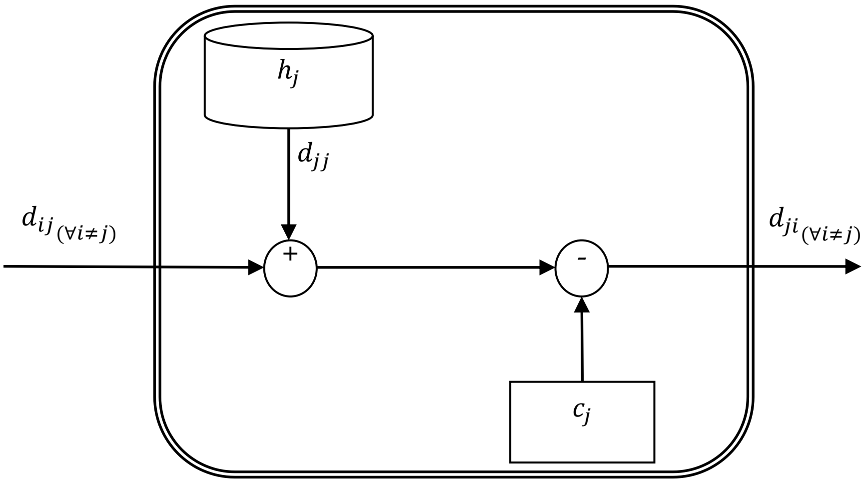


Figure 2. Node's flow model

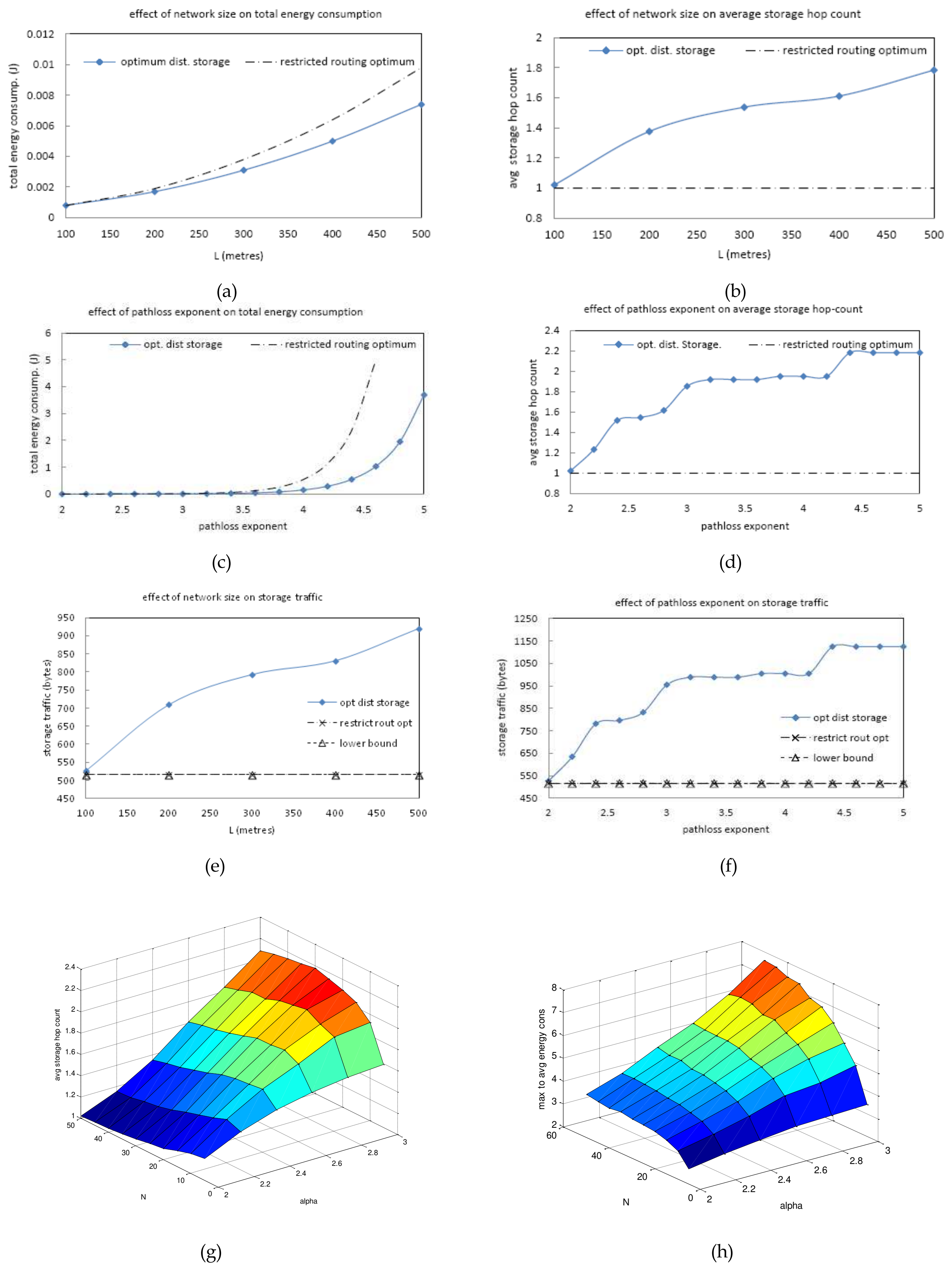


Figure 3. Optimum distributed storage results (*off-line solutions*): (a) effect of network size on energy consumption (b) effect of network size on storage hop-count (c) effect of path-loss exponent on energy consumption (d) effect of path-loss exponent on storage hop-count (e) effect of network size on storage traffic (f) effect of path-loss exponent on storage traffic (g) storage hop-count with node density and path-loss exponent (averaged). (h) MAER with node density and path-loss exponent (averaged).

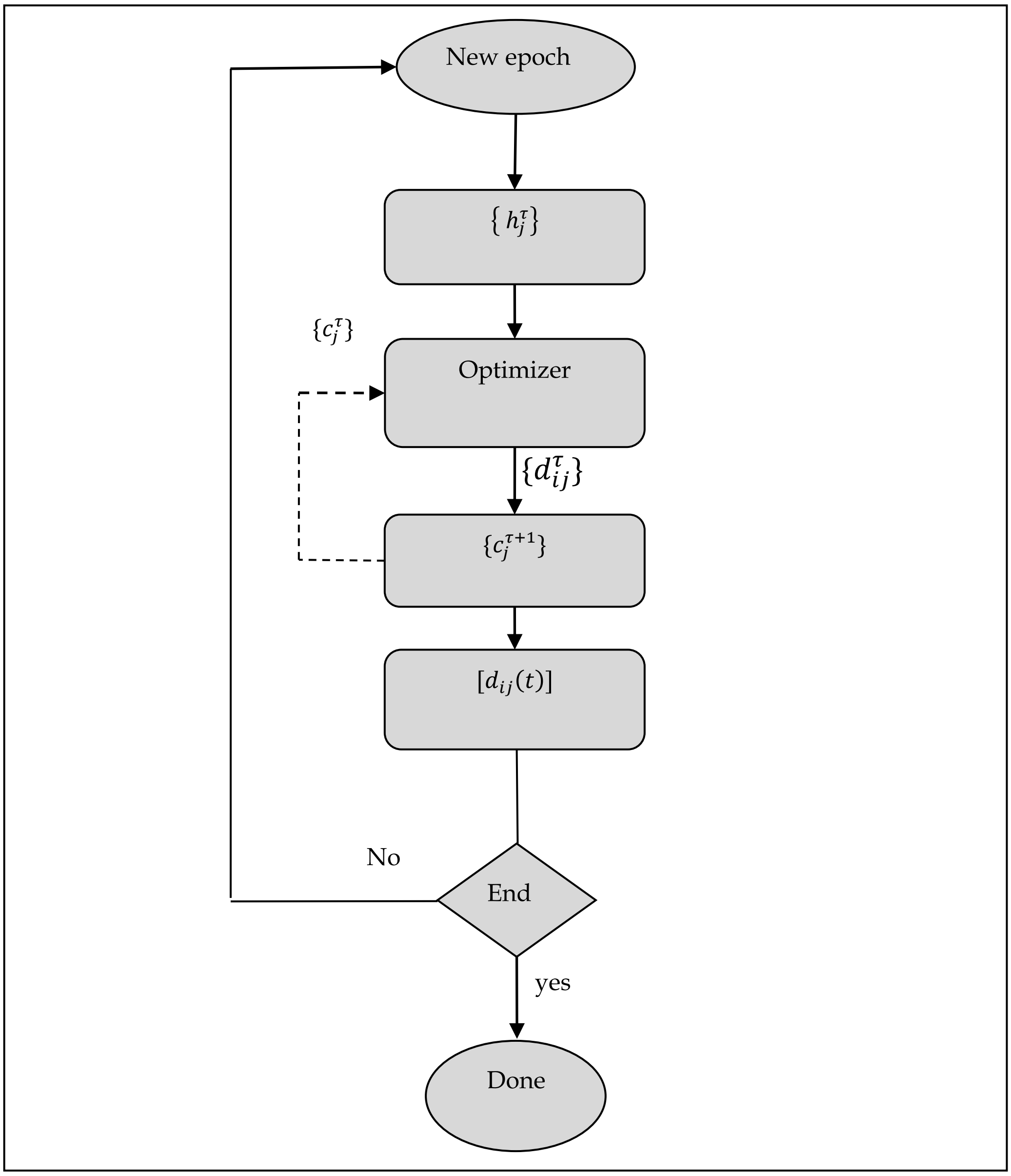


Figure 4. Using the optimization model dynamically

Input: $\boldsymbol{S} = \{s_{ij}\}$, $\boldsymbol{SL} = \{c_j^{\tau}\}$; $\forall j \in \boldsymbol{V}$

Output: sn /* storage node */

Initialization: $\overline{\boldsymbol{S}} \longleftarrow$ INDICES $\{$SORT_ASCEND $\{\boldsymbol{S}\}\}$

/*list of nodes sorted according to their distances from node i */

New_Pkt

$w = 1$ /* indicator */

if $c_i^{\tau} < c$

$sn = i$

$w = 0$

else

for $q = 2{:}N$

if $\boldsymbol{SL}\left(\overline{\boldsymbol{S}}(q)\right) < c$

$sn = \overline{\boldsymbol{S}}(q)$

$w = 0$

exit **for**

end if

end for

end if

if $w = 1$

$sn = \emptyset$

end if

STORE (New_Pkt, sn) /* when $(sn = \emptyset)$ the packet is discarded, when $(sn \notin \{\emptyset, i\})$ the packet is routed to external storage*/

Figure 5. CASH heuristic

Input: $\boldsymbol{S} = \{s_{ij}\}$, $\boldsymbol{SL} = \{c_j^{\tau}\}$; $\forall j \in \boldsymbol{V}$

Output: sn /* storage node */

Initialization I: SET$\{f_1\}$, SET$\{f_2\}$, SET$\{r\}$

/* set heuristic constants */

Initialization II: $\overline{\boldsymbol{S}} \longleftarrow$ INDICES $\{$SORT_ASCEND $\{\boldsymbol{S}\}\}$

/*nodes sorted according to their distances from node $\boldsymbol{i}$ */

New_Pkt

$w = 1$ /* indicator */

if $c_i^{\tau} < c$

  $sn = i$

  $w = 0$

else

  $\boldsymbol{G} = \{\breve{g}_{ij}^{\tau}\}$, $\forall j \in \boldsymbol{V}$

  $\overline{\boldsymbol{G}} \longleftarrow$ INDICES $\{$SORT_DESCEND $\{\boldsymbol{G}\}\}$

  for $q = 1 : (N - 1)$

    if $\boldsymbol{SL}\left(\overline{\boldsymbol{G}}(q)\right) < c$

      $sn = \overline{\boldsymbol{G}}(q)$

      $w = 0$

      exit **for**

    end if

  end for

end if

if $w = 1$

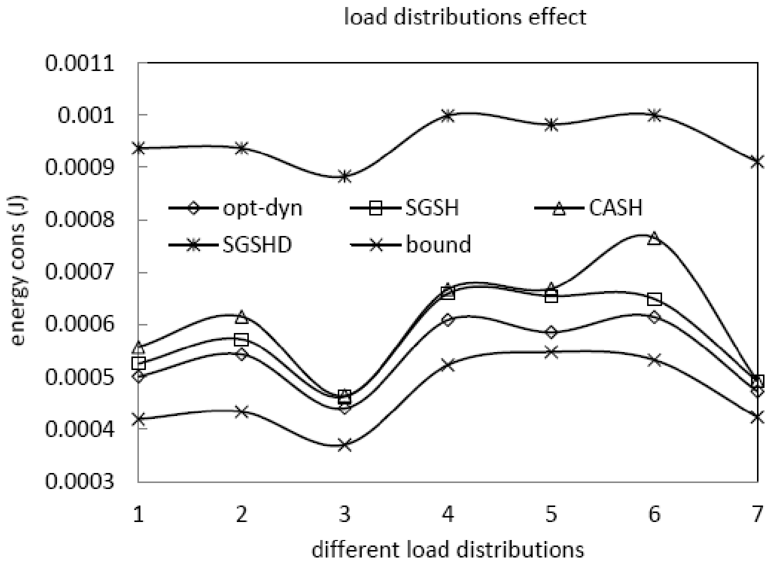
  $sn = \emptyset$

end if

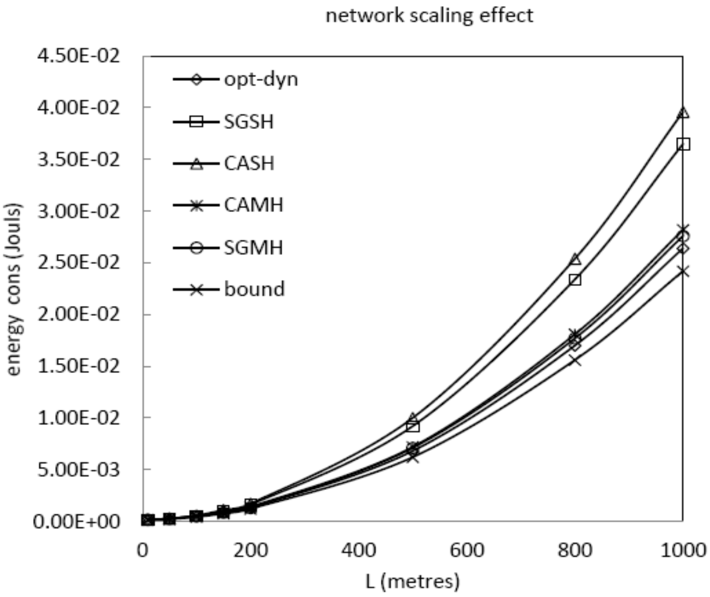
STORE (New_Pkt, sn) /* when $(sn = \emptyset)$ the packet is discarded, when $(sn \notin \{\emptyset, i\})$ the packet is routed to external storage*/

Figure 6. SGSH heuristic

(a)



(b)



(c)

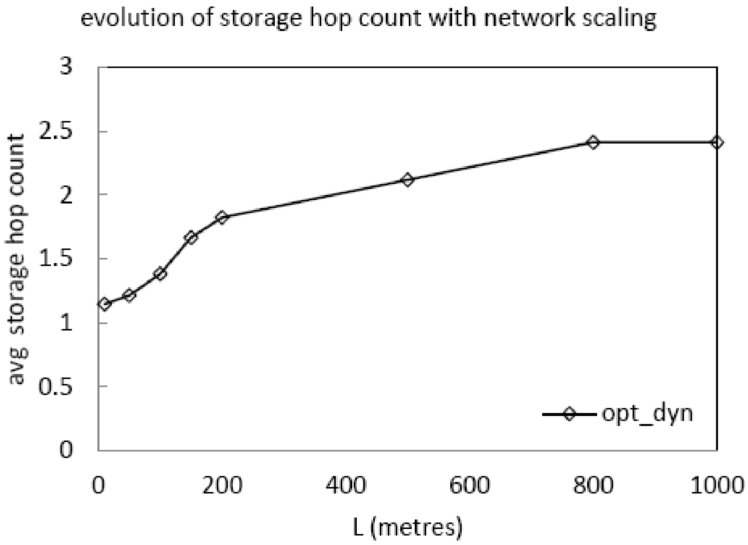
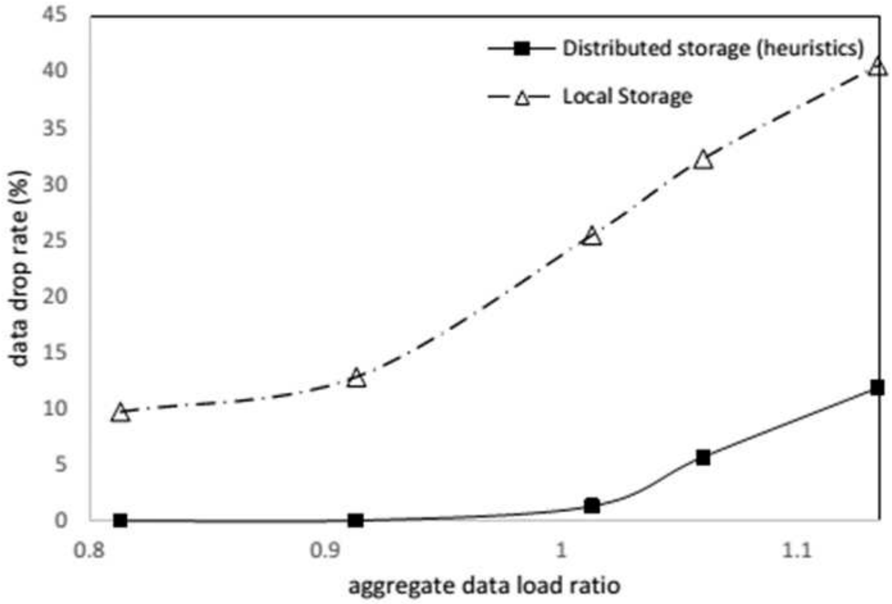
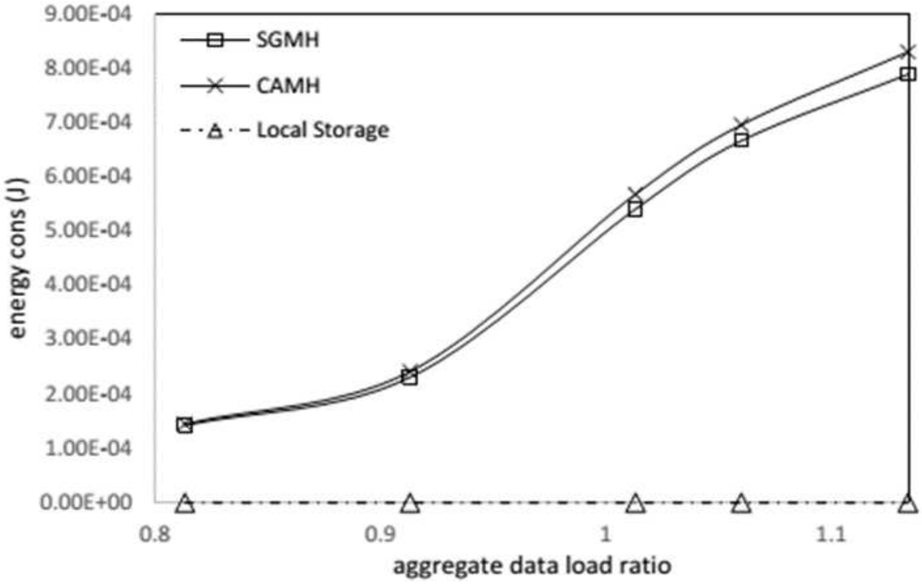


Figure 7. Dynamic storage sharing results



(a)



(b)

Figure 8. Comparing the performance of distributed and local storage

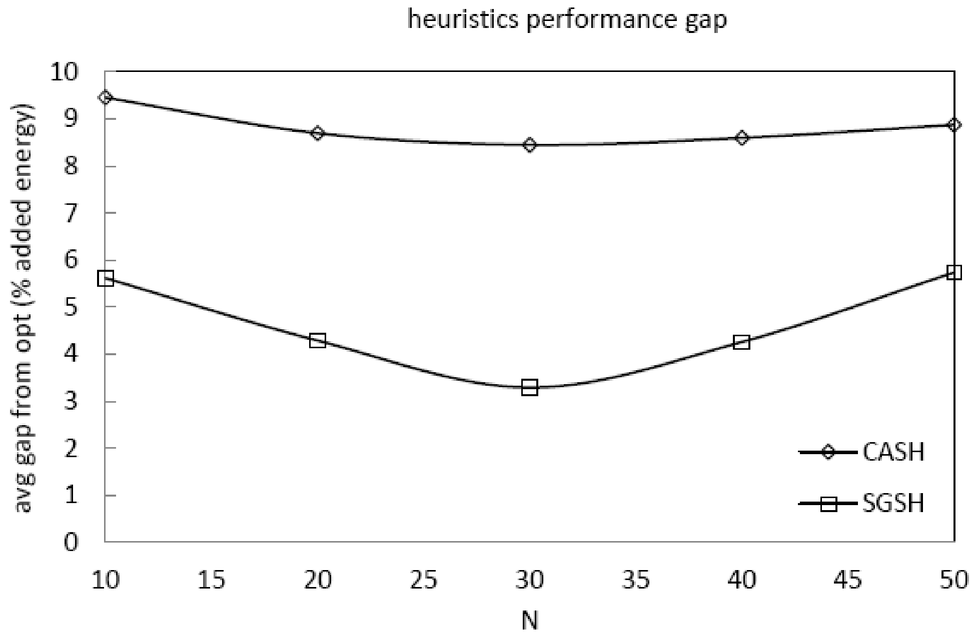


Figure 9. Heuristic average performance gap from optimum.