

3DQS: Distributed Data Access in 3D Wireless Sensor Networks*

Jun Luo Feng Li Ying He

School of Computer Engineering, Nanyang Technological University (NTU), Singapore

Emails {junluo, fli3, yhe}@ntu.edu.sg

Abstract—This paper proposes novel mechanisms to access sensory data in a distributed fashion in 3D wireless sensor networks. As these networks have their nodes deployed in 3D volumes, we first propose a volume parametrization algorithm to transform irregular volumes into a regular one; it also allows us to extend network protocols from 2D to 3D (which would otherwise be highly non-trivial). Based on this transformation, we propose a new quorum system, 3DQS, to handle distributed data access. The tunability of 3DQS enables it to adapt to different application requirements. We demonstrate the efficacy and efficiency of 3DQS through both analysis and simulations.

Index Terms—3D wireless sensor networks, data access, quorum systems, volume parametrization.

I. INTRODUCTION

Although *wireless sensor networks* (WSNs) were traditionally deemed as networks in *two-dimensional* (2D) areas, the interests in deploying wireless sensors nodes in *three-dimensional* (3D) spaces have been becoming increasingly keen. These interests are motivated by applications such as underwater surveillance and atmospheric monitoring. In particular, these 3D WSNs do not simply consist of nodes distributed on a 3D surface, but the networked sensors rather form 3D volumes. This new feature has brought several difficulties to the protocol designs for these 3D WSNs.

One of the issues that pertains to 2D WSNs but becomes harder to solve in 3D is *data access*. Specifically, the issue concerns mechanisms that enable human users to obtain desired data out of the sensor productions. As a conventional solution, a WSN may gather data from a large set of nodes to a particular (often small) set of nodes that also process user queries. Unfortunately, the resulting *convergecast* type of data transmission pattern leads to a very unbalanced load distribution, hence “kill” those heavily loaded nodes first [1]. Moreover, since the arrival of queries may be spatially and temporarily distributed, concentrating all data and queries at a few locations is neither energy efficient nor flexible. As an improvement, the *in-network data storage* approach (e.g., [2]) suggests to replicate data at various nodes, to which a later data query is directed. Though this latter approach does make the data access more flexible, the replication point for a given data type (which is determined by, for example, a hash of the data type [2]) again becomes a hotspot [3]. Therefore, the load balancing problem is left unsolved.

Compared with its 2D counterpart, the load balancing problem gets exacerbated in a 3D WSN due to the dramatically increased node quantity in order to fill a 3D volume. In this paper, we apply the principle of *quorum systems* to address the load balancing issue. Although quorum systems exist in distributed systems [4] and have been applied to wired and wireless networking (e.g., [5], [6], [3], [7]), we propose a non-trivial extension from 2D to 3D based on *volume parametrization*. Our design methodology significantly extends recent developments in using geometric principles to guide the protocol implementations in WSNs, e.g., [8], [7]. We further propose *3D Quorum Systems* (3DQS) where the quorums are formed by curves. Tuning the curve parameters allows 3DQS to maintain load balancing and energy efficiency when facing different application requirements. We also show that 3DQS outperforms the existing in-network data storage approach in terms of load balancing, and we demonstrate its superiority over a trivial extension from a 2D quorum system to 3D. In summary, our main contributions are:

- A novel volume parametrization based quorum design methodology; it applies to 3D WSNs with any shape of the network volumes.
- A specific quorum system, 3DQS, formed by curves with tunable parameters, allowing a flexible adaptation to different application requirements.

In the following, we first discuss backgrounds and related literature in Sec. II. We focus on the data access issue in Sec. III: we first describe our volume parametrization in Sec. III-A, then we present our quorum system design along with the corresponding analysis. We report the simulation results in Sec. IV and conclude our paper in Sec. V.

II. BACKGROUNDS AND RELATED WORK

In this section, we first discuss the basics of quorum systems, as well as a few existing designs in 2D spaces. Then we explain why extending to 3D spaces is not trivial.

A. Quorum Systems in Distributed Systems

A quorum system is defined upon a finite set (also termed *universe*) $\mathcal{U} = \{u_1, u_2, \dots, u_n\}$ of nodes. In particular, the following definition characterizes a quorum system.¹

Definition 1 (Quorum System): A quorum system $\mathcal{Q} \subset 2^{\mathcal{U}}$ consists of two disjoint sets, \mathcal{Q}^R and \mathcal{Q}^W , of subsets of \mathcal{U} ,

*This work is supported in part by AcRF Tier 1 Grant RG 32/09 and 69/07.

¹We actually define the quorum system in an asymmetric fashion [6], differing from its original (symmetric) version [4].

such that each subset in \mathcal{Q}^R intersects every subset in \mathcal{Q}^W . Each subset Q^R (resp. Q^W) in \mathcal{Q}^R (resp. \mathcal{Q}^W) is called a *read* (resp. *write*) *quorum*.

Given a quorum system \mathcal{Q} , networked nodes may make use of it to coordinate data access. A node may choose to *access* a quorum by either *writing* to or *reading* from it. Thanks to the intersection property, a read access will find the desired data from some quorum that stores the data written by another node. A typical design is illustrated in Fig. 1. As conventional distributed systems are supposed to be built

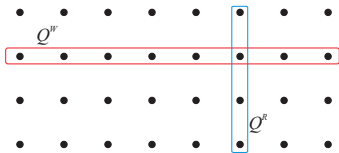


Fig. 1. A typical quorum system design on a grid universe.

upon overlay networks (or the middleware layer), each node is virtually connected to every other node. Consequently, any logical structure of a universe is possible, whereas a grid is commonly used due to its simple topology. Therefore, though the quorum design appears to be rigid, the system flexibility to cope with different application requirements actually relies on the arbitrary shuffling of nodes within the underlying topology.

B. Distributed Data Access in WSNs

As discussed in Sec. I, distributed data access is enabled in WSNs by the in-network storage approach (e.g., [2]), which replicates sensory data at nodes other than the sources and allows the queries to originate anywhere and anytime. However, as these data replication locations become new hotspots, the load balancing problem is not fully solved [3]. To this end, quorum systems are applied as a special in-network storage approach to further balance the load [3], [7]. In WSNs, the grid-like quorum design shown in Fig. 1 is no longer flexible as the underlying topology cannot be arbitrarily shuffled anymore.² Therefore, the system flexibility has to be embodied by the quorum design, as demonstrated by the geometric approach to quorum systems in [3], [7]. This geometric approach suggests using projective map to “lift” the 2D network area onto a sphere, then quorum systems designed on the sphere are projected back to the 2D area. As the design on a sphere allows more diversity in “shaping” the quorums, it hence has a potential to deliver more flexible systems.

C. Extension from 2D to 3D is Highly Nontrivial

Unfortunately, direct extensions from 2D to 3D do not really work. Firstly, one could extend the grid-like quorum design. However, as nodes in a WSN can rarely form a regular topology such as a grid, we need geometric maps to transform an irregular topology to a regular one (as we will show in Sec. III-A). Also, a trivial extension will have scalability issue, because a grid-like quorum system involves nodes on a line

²As the bandwidth resource is very scarce in WSNs, constructing a middleware layer that enables a fully connected topology is almost impossible.

(row or column) in each quorum for 2D, the extension to 3D would require each quorum to include nodes in a plane, significantly reducing the energy efficiency of data access.

Secondly, one may believe that, as it is possible to use the projection proposed in [9] to map a 3D volume to a 4D sphere, the idea presented [3], [7] could be directly applied to 3D WSNs. Unfortunately, although two geodesics on a 3D sphere (two great circles) always intersect each other, there is no such a guarantee in 4D. Therefore, subsets of nodes induced by geodesics on a 4D sphere do not necessarily intersect each other; in other words, they are not quorums. In general, quorums in a 3D WSN have to involve subsets of nodes induced by surfaces or planes (or curves that fill planes).

III. ACCESSING SENSORY DATA IN 3D WSNs

In this section, we first introduce our volume parametrization algorithm that provides a bijection between an arbitrary 3D volume and a solid cube. Then we present quorum system designs for 3D WSNs that utilize this bijection.

A. Algorithm to Parameterize Irregular 3D Volumes

Although surface parametrization has been widely studied in geometric modeling community, techniques on surface parametrization cannot be generalized to volumes directly, due to the fundamental difference between volumes and surfaces. In addition, existing harmonic map based volumetric mapping [10] is not bijective, so a design in a (map) image may lead to ambiguity in the preimage (a WSN). In this section, we present the algorithm for parameterizing 3D WSNs. The algorithm takes a 3D point cloud \mathcal{U} as the input, and it parameterizes \mathcal{U} to a solid cube \mathcal{C} . We assume the boundary of \mathcal{U} is topologically equivalent to a sphere, i.e., \mathcal{U} is a topological ball. We first sketch our proposed *direct product volume parametrization* by Algorithm 1, and we also illustrate part of the algorithmic pipeline in Fig. 2 and provide detailed explanations as well.

Algorithm 1: DPVP_Algo

Input: Point cloud \mathcal{U}

Output: A bijection $\phi : \mathcal{U} \rightarrow \mathcal{C}$, a solid cube \mathcal{C}

- 1 Construct \mathcal{M} , the 3D Delaunay tetrahedralization of \mathcal{U} ;
 - 2 Parameterize the boundary surface $\partial\mathcal{M}$ to the boundary of the cube $\partial\mathcal{C}$. The map is denoted by $\psi : \partial\mathcal{M} \rightarrow \partial\mathcal{C}$;
 - 3 Obtain the partition (induced by ψ) of the boundary surface $\partial\mathcal{M}$ into floor B_0 , ceiling B_1 and walls W , i.e., $\partial\mathcal{M} = B_0 \cup B_1 \cup W$;
 - 4 Compute the harmonic function $\tau : \mathcal{M} \rightarrow \mathbb{R}$, $\Delta\tau = 0$, with Dirichlet boundary condition $\tau(B_0) = 0$ and $\tau(B_1) = 1$;
 - 5 Parameterize \mathcal{M} to \mathcal{C} by tracing the integral curves of the gradient vector field $\nabla\tau$ for every vertex of \mathcal{M} .
- return** ϕ, \mathcal{C}
-

Step 1. In order to parameterize a point cloud to a solid volume, our first step is to construct a 3D tetrahedralization of

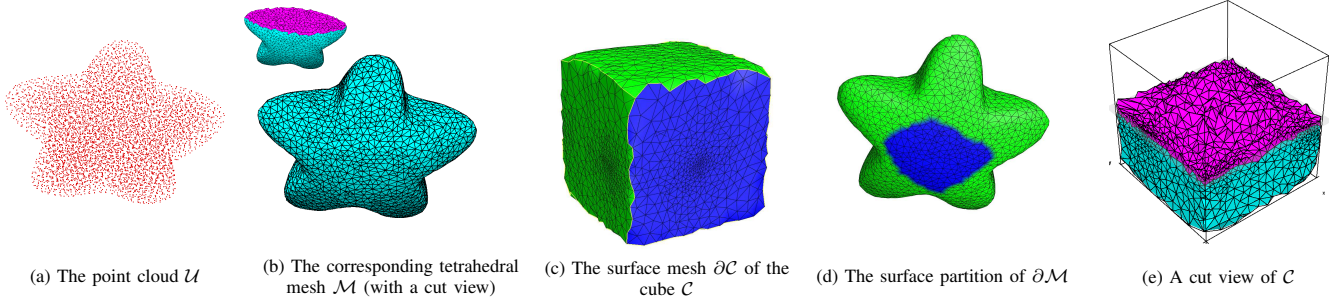


Fig. 2. Taking an arbitrary genus 0 volume and a point set \mathcal{U} inside it as input (a), Step 1 starts with a Delaunay tetrahedralization to obtain \mathcal{M} (b). Then the boundary surface $\partial\mathcal{M}$ is parameterized to that of the unit cube $\partial\mathcal{C}$, through Step 2 of DPVP_Algo (c). The boundary surface $\partial\mathcal{M}$ is partitioned into ceiling (red, not shown), floor (blue) and walls (green) according that of $\partial\mathcal{C}$ in Step 3 (d). Finally, the inner mesh of \mathcal{C} (e) is obtained by Steps 4 and 5.

the point cloud. In our implementation, we use Tetgen [11].
Step 2. The parametrization from $\partial\mathcal{M}$ to $\partial\mathcal{C}$ is guided by the following commutative diagram:

$$\begin{array}{ccc} \partial\mathcal{M} & \xrightarrow{\psi} & \partial\mathcal{C} \\ f \downarrow & & \downarrow g \\ \mathbb{S}^2 & \xrightarrow{h} & \mathbb{S}^2 \end{array}$$

We first parameterize $\partial\mathcal{M}$ and $\partial\mathcal{C}$ to the unit sphere \mathbb{S}^2 using spherical conformal parametrization [12]. Then we find a conformal map between the unit spheres, $h : \mathbb{S}^2 \rightarrow \mathbb{S}^2$. Here we choose the identity map, which is a conformal map. Since all maps g , h and l are conformal, the composite map $\psi = g^{-1} \circ h \circ f$, mapping $\partial\mathcal{M}$ to $\partial\mathcal{C}$, is also conformal.

Step 3. We partition the boundary surface of $\partial\mathcal{C}$ into three disjoint sets, i.e., floor (the bottom square), ceiling (the top square) and walls (the remaining four squares). Note that the boundary map $\psi : \partial\mathcal{M} \rightarrow \partial\mathcal{C}$ is conformal, its inverse $\psi^{-1} : \partial\mathcal{C} \rightarrow \partial\mathcal{M}$ induces a partition on $\partial\mathcal{M}$. Let $B_0 \in \partial\mathcal{M}$ (resp. $B_1 \in \partial\mathcal{M}$) denote the floor (resp. ceiling) of $\partial\mathcal{M}$, and $W \in \partial\mathcal{M}$ denote the walls. Then $\partial\mathcal{M} = B_0 \cup B_1 \cup W$.

Step 4. We compute a harmonic function $\tau : \mathcal{M} \rightarrow \mathbb{R}$ in \mathcal{M} ,

$$\nabla^2 \tau(v) = 0, \quad \forall v \notin B_0 \cup B_1$$

such that $\tau(B_0) = 0$ and $\tau(B_1) = 1$. The above Laplace equation can be solved readily using the *finite element method* on the tetrahedral mesh \mathcal{M} . It can be shown that any integral curve of the gradient vector field $\nabla\tau$ must have two ending points, one on the floor and the other on the ceiling. Furthermore, any two arbitrary integral curves γ_1 and γ_2 do not intersect. Therefore, given an arbitrary point $v \in \mathcal{M}$, there is a *unique* integral curve $\gamma \in \mathcal{M}$ such that γ passes through v and γ follows the gradient direction of τ .

Step 5. The volume parametrization $\phi : \mathcal{M} \rightarrow \mathcal{C}$ is constructed as follows. For arbitrary non-floor point $v \in \mathcal{M}$ and $v \notin B_0$, let $\gamma \in \mathcal{M}$ be the unique integral curve of the gradient vector field $\nabla\tau$ that γ passes through v . As mentioned above, γ has two ending points on B_0 and B_1 respectively. Without loss of generality, say, $v_0 \in B_0$ and $v_1 \in B_1$. By using the floor map (i.e., restricting the boundary map $\psi : \partial\mathcal{M} \rightarrow \partial\mathcal{C}$ to the floor $B_0 \in \partial\mathcal{M}$), we compute $v'_0 = \psi(v_0)$. Obviously, v'_0 is on the floor of \mathcal{C} . Then the image of v under map $\phi : \mathcal{M} \rightarrow \mathcal{C}$, $\phi(v)$, is given by $\phi(v) = v'_0 + (0, 0, \tau(v))^T$.

Remarks: The constructed map $\phi : \mathcal{M} \rightarrow \mathcal{C}$ is a bijection as both the floor map and the harmonic function are bijective.

B. Network Model

We represent a WSN by \mathcal{U} , with $u_i \in \mathcal{U}$ being a sensor node. \mathcal{U} is also the point cloud input to DPVP_algo and the universe upon which a quorum system can be defined. We make the following assumptions on the network:

- Each node is aware of its own location by, e.g., [13].
- Nodes on the network boundary are aware of their status through a boundary detection mechanism.
- The node density is sufficiently high to allow *trajectory forwarding* [14], where routing paths are given by predefined curves.

Given an arbitrarily deployed 3D WSN (which forms a 3D volume), we perform the following three steps during the initialization phase in a centralized fashion. Firstly, we use DPVP_algo to map the 3D volume to a solid cube. Secondly, we design quorum systems along with the routing paths that allow nodes to access their corresponding quorums in the cube. Thirdly, the inverse images of the design routing paths are obtained and they are coded before disseminated to all nodes. During the WSN operation phase, the trajectory forwarding [14] is used along with the coded routing paths whenever a read or write access is initiated by some node.

C. Quorum System in 3D

We present two quorum system designs in this section. As explained in Sec. III-B, the design is done in a cube. The first design is a trivial extension from the 2D grid-like quorum systems, and the second design, 3DQS, represents our idea of improving system flexibility using tunable quorums. We will compare these two designs through both analysis and simulations, in Sec. III-D and Sec. IV respectively.

1) *Grid Quorum Systems (GQS):* We simply use two sets of perpendicular planes to form read and write quorums. The boundary of a quorum is the intersections between the defining plane and the network boundary, and a quorum consists of nodes that are traversed by the routing path that guides a data access. In particular, we have:

- **Write quorum:** defined by the horizontal plane passing through the node that performs a write access. The

routing path to access the quorum is defined by a space-filling curve that fills the area.

- **Read quorum:** defined by a vertical plane passing through the node that performs a read access. The routing path consists of two rays starting from that node. One ray hits the boundary and the other stops upon intersecting with the targeted write quorum.

We show a pair of read/write quorums in Fig. 3. Note that

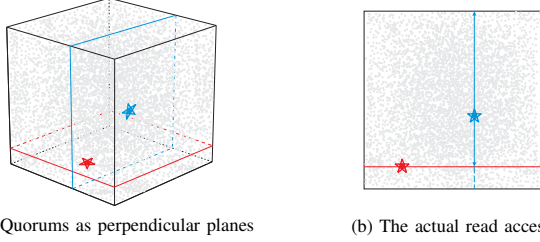


Fig. 3. GQS in a cube. We use red (resp. blue) color to indicate a write (resp. read) quorum, respectively. We also use pentagrams to represent the accessing nodes. The read access in (b) is illustrated on the defining plane of the corresponding read quorum.

the nodes that form a quorum may not be exactly on the defining plane or curve, but they are usually close to it, due to the property of trajectory forwarding [14].

Remarks: GQSs inherit the rigidity of their 2D counterpart, but the negative effect is amplified in 3D. For example, the communication load incurred by a write access is much higher than that of a read access. Therefore, the system becomes extreme inefficient under high write access rate.

2) *3DQS: Quorum Systems with Tunability:* Our design is dedicated to address the weaknesses of GQS. We continue using perpendicular planes to define quorums, but we use parameters to control the boundary of write quorums and the routing paths of read quorums. More specifically,

- **Write quorum:** a rectangular area on the horizontal plane passing through the node that performs a write access. The width w of the rectangular area is a parameter. The routing path is again given by a space-filling curve.
- **Read quorum:** though the same planes are used to define these quorums, the routing paths include parallel line segments spaced by a distance d . The curves end when hitting either a write quorum or the network boundary.

Suppose that all read quorums are aligned with x -axis, then all write quorums have to be aligned with y -axis. We show a pair of read/write quorums in Fig. 4.

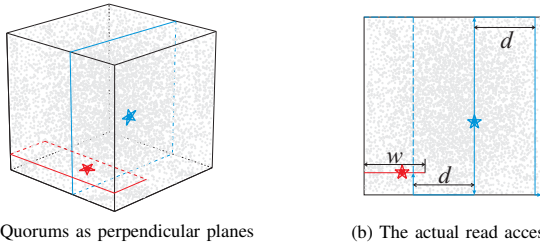


Fig. 4. 3DQS in a cube, with tunable parameters shown in (b).

The following result is stated without proof.

Proposition 1: A read quorum intersects a write quorum at no less than k locations if $w \geq kd$.

As $k \geq 1$ is required, we need to have $w \geq d$. Obviously, a larger value of k makes 3DQS more robust against node failures, but it also reduces the energy efficiency, as a routing path for read access becomes longer for a certain w .

D. Analysis of Energy Efficiency

We perform a geometric analysis to compare the two designs presented in Sec. III-C in terms of their energy efficiency. We assume the network cube has an edge length of ℓ . We also assume that the node set involved in a read (resp. write) quorum is equivalent to a pipe centered around the routing path and with a cross-section area of δ^2 (resp. a rectangular cuboid with its bottom given by the quorum defining plane and a height of δ). Let n_R (resp. n_W) denote the number of read (resp. write) nodes, and let λ_R (resp. λ_W) denote the rate of read (resp. write) accesses. To simplify the analysis, we assume that each read access targets at a relatively large set of write quorums. This effectively implies that each read access almost always goes through the whole routing path, i.e., including also the dash line part in both Fig. 3 and Fig. 4.

For GQS, the total energy consumption is represented as $E_{GQS} = n_R \lambda_R \ell \delta^2 + n_W \lambda_W \ell^2 \delta$, up to a constant multiplier. And that of 3DQS is represented as $E_{3DQS} = n_R \lambda_R \ell \delta^2 (1 + \lceil \frac{\ell}{d} \rceil) + n_W \lambda_W \ell w \delta$, up to the same constant multiplier. Let $d = w$ and $\frac{n_R \lambda_R}{n_W \lambda_W} = \rho$, we have

$$\begin{aligned} \frac{E_{3DQS}}{E_{GQS}} &= \frac{n_R \lambda_R \ell \delta^2 (1 + \lceil \frac{\ell}{d} \rceil) + n_W \lambda_W \ell w \delta}{n_R \lambda_R \ell \delta^2 + n_W \lambda_W \ell^2 \delta} \\ &= \frac{\rho \delta (1 + \lceil \frac{\ell}{d} \rceil) + w}{\rho \delta + \ell} \geq \frac{w}{\ell} \end{aligned} \quad (1)$$

The lower bound can be a good approximation if $\rho \gg 1$ and $\frac{\delta}{w} \ll 1$; the later is true as δ is usually in the order of a one-hop distance and w is in the order of the network dimension ℓ . Also, if $\rho < 1$, it would justify to take smaller value of w and hence results in even bigger advantage of 3DQS.

IV. SIMULATIONS

We randomly put nodes in 5 arbitrary 3D volumes, as shown in Fig. 2(a) (Net1, 6501 nodes) and Fig. 5 (Net2 to Net5). We follow the steps described in Sec. III-B to

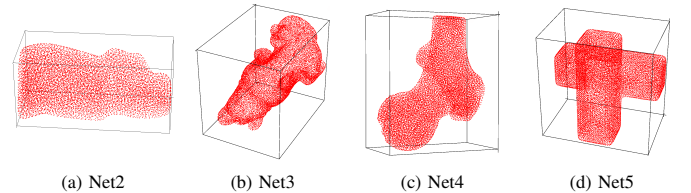


Fig. 5. 3D WSNs with size (a) 10817, (b) 20069, (c) 18304, and (d) 19573.

design quorum systems and the corresponding routing paths. We assume the energy consumption of individual nodes is dominated by wireless communications. In order to compute the load distribution and the total energy consumption, we take

the following rule: as soon as a curve (i.e., the corresponding routing path) passes through a tetrahedron, all the four vertices are charged with a unit of communication load. This stems from the broadcast nature of wireless communications and the need for local coordination in the trajectory based forwarding. We assume 500 writing nodes ($n_W = 500$), 100 reading nodes ($n_R = 100$), and 50 locations used by a GHT based in-network storage system (termed GHT hereafter). We take $\frac{\lambda_W}{\lambda_R} = 1$, $\ell = 1$ and $w = 0.25$. Due to space limitation, we omit the investigations the impact of varying these values.

A. Improved Load Balancing

In this section, we compare the load balancing effect of three systems, namely GHT, GQS and our new design 3DQS. We use the *maximum load* (i.e., the maximum load taken by a single node within the whole network) to indicate how balanced the load is. Obviously, the smaller this value is, the better the load is balanced. To provide an intuitive illustration of the load distribution, we choose Net1 and render its volume based on a color map in Fig. 6 to represent the actual load distribution. It can be easily observed that 3DQS leads to an almost perfect load distribution. The comparison of the maximum loads for

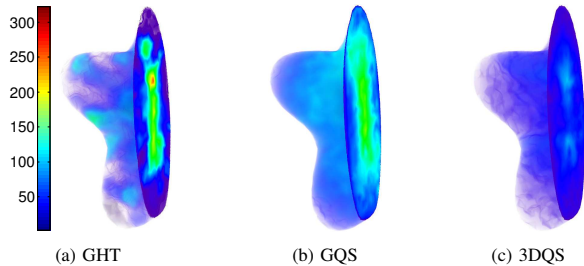


Fig. 6. The load distribution for the three systems in Net1.

the three systems are given in Fig. 7(a). It is clear that, though using a quorum system (either GQS or 3DQS) always leads to more balanced load distribution, 3DQS does a much better job than GQS: the maximum load of 3DQS can be below $\frac{1}{2}$ of that of GQS and $\frac{1}{3}$ of that of GHT.

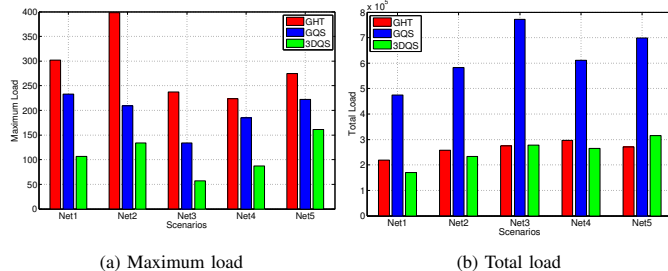


Fig. 7. Comparing maximum and total loads for the three systems.

B. Better Energy Efficiency

For WSNs, load balancing is not the only concerned issue. It is equally important that balancing energy consumptions among nodes does not lead to an excessive total energy

consumption (or *total load* hereafter). In other words, energy efficiency of the whole network is equally important. Our claim here is that 3DQS does not sacrifice energy efficiency to achieve its superb load balancing. The comparison made in Fig. 7(b) on total load strongly corroborates our statement. It is clear that, although a trivial design like GQS does trade energy efficiency for load balancing, 3DQS has almost the same (or even lower) total load as that of GHT. As a result, 3DQS is far more energy efficient than GQS.

V. CONCLUSION

In this paper, we focus on the distributed data access in WSNs deployed in 3D volumes, and we propose a novel quorum system, 3DQS, as the solution. In particular, we parameterize the network volume to a solid cube, in which 3DQS is designed with tunable parameters. According to our analysis and simulations that compare 3DQS with existing methods, 3DQS apparently has a big advantage in terms of load balancing while preserving energy efficiency. Moreover, we are allowed to tune the parameters of 3DQS to maintain its advantage under different application requirements.

Our paper is based on the assumption that nodes are deployed uniformly within a network volume, but non-uniformly node distribution is possible in practice. We are on the way of tackling this issue as part of our future work.

REFERENCES

- [1] J. Luo and J.-P. Hubaux, "Joint Sink Mobility and Routing to Increase the Lifetime of Wireless Sensor Networks: The Case of Constrained Mobility," *IEEE/ACM Trans. on Networking*, vol. 18, no. 3, pp. 871–884, 2010.
- [2] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, "GHT: A Geographic Hash Table for Data-Centric Storage in Sensor Networks," in *Proc. of the 1st ACM WSNA*, 2002.
- [3] R. Sarkar, X. Zhu, and J. Gao, "Double Rulings for Information Brokerage in Sensor Networks," in *Proc. of the 12th ACM MobiCom*, 2006.
- [4] D. Malkhi and M. Reiter, "Byzantine Quorum System," *Springer Distributed Computing*, vol. 11, no. 4, pp. 569–578, 1998.
- [5] L. Zhou, F. Schneider, and R. van Renesse, "COCA: A Secure Distributed On-line Certification Authority," *ACM Trans. on Computer Systems*, vol. 20, no. 4, pp. 329–368, 2002.
- [6] J. Luo, P. Eugster, and J.-P. Hubaux, "Pilot: Probabilistic Lightweight Group Communication System for Ad Hoc Networks," *IEEE Trans. on Mobile Computing*, vol. 3, no. 2, pp. 164–179, 2004.
- [7] J. Luo and Y. He, "GeoQuorum: Load Balancing and Energy Efficient Data Access in Wireless Sensor Networks," in *Proc. of the 30th IEEE INFOCOM (mini-conference)*, 2011.
- [8] W. Zeng, R. Sarkar, F. Luo, X. Gu, and J. Gao, "Resilient Routing for Sensor Networks Using Hyperbolic Embedding of Universal Covering Space," in *Proc. of the 29th IEEE INFOCOM*, 2010.
- [9] F. Li, S. Chen, Y. Wang, and J. Chen, "Load Balancing Routing in Three Dimensional Wireless Networks," in *Proc. of IEEE ICC*, 2008.
- [10] X. Li, X. Guo, H. Wang, Y. He, X. Gu, and H. Qin, "Meshless Harmonic Volumetric Mapping Using Fundamental Solution Methods," *IEEE Trans. on Automation Science and Engineering*, vol. 6, no. 3, pp. 409–422, 2009.
- [11] H. Si, "TetGen: A Quality Tetrahedral Mesh Generator and a 3D Delaunay Triangulator." [Online]. Available: <http://tetgen.berlios.de>
- [12] X. Gu, F. Luo, and S.-T. Yau, "Recent Advances in Computational Conformal Geometry," *Comm. Inf. Sys.*, vol. 9, no. 2, pp. 163–196, 2009.
- [13] J. Luo, H. Shukla, and J.-P. Hubaux, "Non-Interactive Location Surveying for Sensor Networks with Mobility-Differentiated ToA," in *Proc. of the 25th IEEE INFOCOM*, 2006.
- [14] D. Niculescu and B. Nath, "Trajectory Based Forwarding and its Applications," in *Proc. of the 9th ACM MobiCom*, 2003.