Energy-Efficient and Adaptive Algorithms for Constructing Multipath Routing in Wireless Sensor Networks

Shaohua Wan

Energy-Efficient and Adaptive Algorithms for Constructing Multipath Routing in Wireless Sensor Networks

Shaohua Wan

School of Information and Safety Engineering, Zhongnan University of Economics and Law,
430073 Wuhan, China
shwanhust@gmail.com

Abstract. In this paper, we design and implement a k-multipath routing algorithm that allows a given source node send samples of data to a given sink node in a large scale sensor networks. Construction and dynamic selection of alternative routing structures, for the purpose of extending the networks lifetime, while providing a balance between QoS(Quality of Service) requirements and the minimization of the variance of the energy. The proposed multipath routing algorithm tries to keep multipath as node disjoint routes. Our view is that balancing load distribution while meeting acceptable delays for applications can lead to significant power savings. The simulation results demonstrate that our multipath routing algorithm can not only achieve load balancing, but also can be help to prolong the life-span of network, compared with the shortest path routing(single path routing SPR).

1   Introduction

Wireless Sensor networks are composed of hundreds, or possibly thousands of tiny low-cost nodes which, once deployed in a particular physical environment, can measure various values of interest, perform some limited computation and communicate with other in order to achieve a desired task in a cooperative manner. Sensor networks can be deployed in large geographic areas to actively monitor a variety of operations ranging from long-term ones (e.g., security alerts) to short-term ones with high degree of dynamics (disaster management). A major research problem, critical to the real world operation of sensor networks, is to design networks that are efficiently and dynamically adaptable to the energy and QoS requirements. The main goal of the proposed research will be to develop energy- efficient and adaptive algorithms for constructing routing trees.

An important aspect of energy-efficiency is performing in-network aggregation while routing data from source sensors through intermediate nodes in the network. Servicing an aggregate query, say Q_i, involves disseminating the query from a given sink s_i, that requested it to all the target sensing nodes relevant for its processing; and sending the results from each of the target nodes back to the sink. An effective way of disseminating the queries and gathering the query answer is using a tree structure rooted at the sink. Once the tree is constructed, each of its nodes has a dual role:

forward the answer-sets measured by the children towards the sink; and perform some local aggregation of the data, in order to reduce the communication overhead.

The rest of the paper is organized as follows. In section 2, we provide related work into the area of multipath routing for wireless sensor networks. We model the query component and formulate the general construction point-to-point routes problem in section 3. Section 4 discusses route establishment, data transmission and route maintenance of the k-multipath routing. Based on simulation results, section 5 presents a detailed analysis of load distribution, energy consumption, lifetime for both multipath and the shortest path routing mechanisms. Section 6 concludes the paper.

2   Related Work

There has been recent research on distributed algorithms for construction of low-weight connected sub graphs and spanners in the context of wireless sensor networks(motivated by energy-efficient routing and fault- tolerant deployment) [1,2,3,4,5,9,10,11]. There has been little work on localized and distributed construction of routing trees for data aggregation in wireless sensor networks. Li et al. give a local algorithm to construct a low-weight sub graph (called as k-Local MST) that has many desirable properties: connectivity, sparseness, spanner, bounded degree, and planarity; but it is not a tree. A structure is low weight if its total edge length is within a small constant factor of the total edge length of the minimum spanning tree. Since the structure is not a tree, it is not suitable for query applications where an aggregation tree is needed. However, low weight structures and spanners, in particular, are useful in reducing the complexity of the underlying graph. The tree construction algorithms can be run on "top" of such spanner structures. MST cannot be constructed in a purely localized manner, i.e., each node cannot determine which edge is in the defined structure by using only the information of the nodes within some constant hops.

Data aggregation has been studied extensively [6, 7, 8]. The main motivation is to minimize the transmission of packets containing individual measurements whenever the semantics of the application needs a summarized picture of the environment, e.g., a weighted sum of the signals, and allows for functional decomposition when calculating the statistical values. When the data based on the actual measurements is categorical, even pattern identification techniques can be used for aggregation. Important parameters that impact data aggregations in wireless sensor networks has been addressed from the perspective of database-like query processing and, recently, the energy efficiency of node clustering with data aggregation trees has been studied.

The focus is on efficient processing of a mix of aggregate queries and not on constructing the routing trees themselves in an energy-efficient fashion. In contrast, the objective of our research is to construct such trees considering the evolution of the network as a sequence of generated queries with different semantics and adapting the routing structures both in the sense of constructing a new one and modifying the existing ones.

3 Query Model and Problem Formulation

3.1 Query Model

Users need to be able to interact with the sensor network, typically by connecting to a (sink) node and submitting queries of interest for which the network must provide accurate answers in a timely manner. While the most common, standard query specification language is SQL, which is typically used in database systems, it has also been adopted in wireless sensor network application based on the abstraction that the network represents, in fact, a largely distributed database system. However, specific aspects that distinguish a typical database from a sensor network infrastructure brought modifications to SQL and the most recognized SQL-specification for wireless sensor network is TinySQL. Regardless of the query specification language, a wireless sensor network needs to provide mechanisms for query processing that are both energy and bandwidth conscious. For example, as we have shown in the following, it is imperative, for resource usage efficiency, that a sensor network to implement a triggering (or similar) mechanism in order to better implement monitoring queries (continuous queries). The typical SQL style is as follows: a network user connects to one of the sink nodes, formulates and submits a query of the following form:

Q: SELECT   ALL/MIN/MAX/AVG (measurement)
      FROM   Region ($R_1(x_1, y_1, ... , x_n, y_n)$)
      WHERE   Condition (measurement)
      FOR   Lifetime
      SAMPLE EVERY Sampling Interval

R_1 represents the geographical bounds of the region in which the samples for the query are to be collected from. If R_1 is not explicitly specified, it defaults to the entire sensor network deployment area. Sampling interval indicates the frequency each node must acquire the measurements and ship the data towards the sink. The sensor must stop sensing and sending the data towards the sink node after the lifetime period expires. We only consider the sink node is outside of region R_1.

Figure 1 gives an illustration of this case that we will exclusively consider. Therefore, we will have to construct point-to-point routes from the aggregation root node, which is situated inside the sampling region, to the ultimate destination, the sink node. The aggregation results should be shipped along with these paths.

3.2 Problem Formulation

Sensor nodes that are outside the sampling region are also important as they might be used in data-relay duties, making the connection between the producer, in the sampling region, and its consumer, the sink node. For each source-destination pair, a single (shortest) path is always discovered and used for data transmission, as seen in Figure 1, the aggregated information will be sent to the sink through the bold intermediate nodes. Obviously, in wireless sensor network, with a high density of

nodes the shortest paths connecting any pair of nodes tend to be very close to the line segment connecting those two nodes. Hence, that area close to the line segment will be very likely to develop hot-spots, which is the situation we are trying to avoid in the first place. The fact that these nodes are overused is one of the major causes for hot spots. This paper provides a new multipath protocol for mitigating the sensor network hot- spot problem, considering load balancing as well as quality of data.

4 K-Multipath Routing Algorithm

4.1 Route Establishment

We assume that each node knows its location and the location of its neighbors. This simulator provides us with the heartbeat algorithm, which is already implemented and executed in the first hour of the simulation, and finds the neighbors for us.
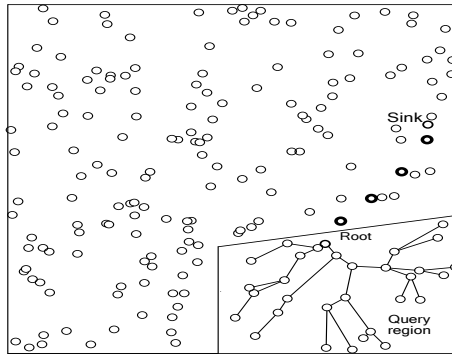

Fig. 1. Sink node is physically located outside the sampling region.

The algorithm for constructing the routing structure should be as follows. For a given source-sink pair of nodes, the sink will unicast on a shortest path routing (along a straight imaginary line) the query request to the source node. Subsequently, based on the query specification, referring to Figure 2, we will draw a segment orthogonal to source-sink line segment. We will split the segment in k places, which will correspond to k intermediate destination points (breakpoints) of the paths between the sink and the source. The distance between two consecutive paths on the line which is orthogonal to the source-sink segment will be equal. For each line segment, we forward the data packet using nodes closest to the line segment. Since we have k breakpoints for a given source-sink pair, we will establish k multipath to offer more opportunities for regulating the traffic over the network.

Multipath routing protocol can try to find node disjoint, link disjoint, or non-disjoint routes. Node disjoint routes, also known as totally disjoint routes, have no nodes or links in common. Link disjoint routes have no links in common, but may have nodes in common. Non-disjoint routes can have nodes and links in common.

Since we assume the whole topology is known, finding node-disjoint multiple paths is not a difficult task. Figure 2 shows an example of how to construct k-multipath which are node-disjoint routes given a source-sink pair of nodes. As can be seen in this Fig., that k-multipath from source to sink does not interfere with each other except that they share the resources at source and sink. To the best of my knowledge, even if the multiple paths are node-disjoint, transmissions along the routes may interfere if some nodes among the routes are in the same collision domain. When we establish multiple paths, it is important to establish paths that are as independent as possible to ensure the least interference between the paths.
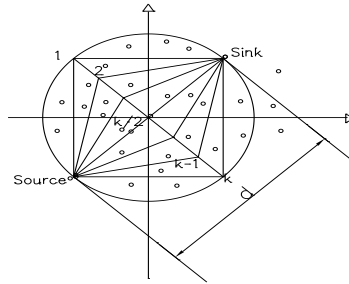

Fig. 2. K-multipath routing protocol model.

The theoretical principle is adopted in our approach relies on the concept of Bezier curves, developed by Paul de Casteljau (1959) and independently by Pierre Bezier (1962). In its general form, a Bezier curve of a given set of n+1 points: Pi (i=0, 1, 2······n), and a parameter t∈[0, 1], is defined as:

$$p(t) = \sum_{i=0}^{n} P_i B_{i,n}(t) \tag{1}$$

Where B$_{i,n}$(t) Bernstein polynomials, defined as:

$$B_{i,n}(t) = \frac{n!}{i!\,(n-i)\,!}\, t^i (1-t)^{n-i} \tag{2}$$

Sum to one for any t in [0, 1],

$$\sum_{i=0..n} B_i^n(t) = 1 \tag{3}$$

For 3 control points, n = 2,

$$p(t) = (1-t)^2 p_0 + 2t(1-t)p_1 + t^2 p_2 \tag{4}$$

For 4 control points, n = 3,

$$p(t) = (1-t)^3 p_0 + 3t(1-t)^2 p_1 + 3t^2(1-t)p_2 + t^3 p_3 \tag{5}$$

$$p(t) = (-t^3 + 3t^2 - 3t + 1)\mathbf{p}_0 + (3t^3 - 6t^2 + 3t)\mathbf{p}_1 \\ + (-3t^3 + 3t^2)\mathbf{p}_2 + (t^3)\mathbf{p}_3 \tag{6}$$

$$p(t) = (-\mathbf{p}_0 + 3\mathbf{p}_1 - 3\mathbf{p}_2 + \mathbf{p}_3)t^3 + (3\mathbf{p}_0 - 6\mathbf{p}_1 + 3\mathbf{p}_2)t^2 \\ + (-3\mathbf{p}_0 + 3\mathbf{p}_1)t + (\mathbf{p}_0)1 \tag{7}$$

If we regroup the equation by terms of exponents of t, we get it in the standard cubic form. This form is very good for fast evaluation, as all of the constant terms (a, b, c, d) can be recomputed. The cubic equation form obscures the input geometry, but

there is a one-to-one mapping between the two and so the geometry can always be extracted out of the cubic coefficients.

$$\mathbf{a} = \left(-\mathbf{p}_0 + 3\mathbf{p}_1 - 3\mathbf{p}_2 + \mathbf{p}_3 \right)$$
$$\mathbf{b} = \left(3\mathbf{p}_0 - 6\mathbf{p}_1 + 3\mathbf{p}_2 \right)$$
$$\mathbf{c} = \left(-3\mathbf{p}_0 + 3\mathbf{p}_1 \right) \tag{8}$$
$$\mathbf{d} = \left(\mathbf{p}_0 \right)$$

$$p(t) = \mathbf{a}\,t^3 + \mathbf{b}\,t^2 + \mathbf{c}\,t + \mathbf{d}$$

$$p(t) = \begin{bmatrix} t^3 & t^2 & t & 1 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} \qquad \begin{bmatrix} \mathbf{a} \\ \mathbf{b} \\ \mathbf{c} \\ \mathbf{d} \end{bmatrix} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 3 & -6 & 3 & 0 \\ -3 & 3 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} \mathbf{p}_0 \\ \mathbf{p}_1 \\ \mathbf{p}_2 \\ \mathbf{p}_3 \end{bmatrix} \tag{9}$$

We can rewrite the equations in matrix form. This gives us a compact notation and shows how different forms of cubic curves can be related. It also is a very efficient form as it can take advantage of existing 4x4 matrix hardware support. For example, given three points P0, P1, and P2, a quadratic Bezier curve is the path traced by the equation (4) as shows in Figure 3.
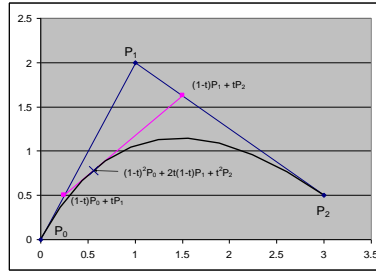

Fig. 3. Construct a Bezier curve from given three points

The curve passes through the first, P0 and last vertex points, P_n. The tangent vector at the starting point P0 must be given by $P_1 - P_0$ and the tangent P_n given by $P_n - P_{n-1}$. The properties of the Bernstein polynomials ensure that all Bezier curves lie in the convex hull of their control points. Hence, even though we do not interpolate all the data, we cannot be too far away. Figure 4 describes convex polygon formed by connecting the control points
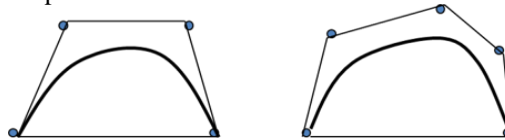

Fig. 4. Curve resides completely inside its convex hull

We have relied on the flexibility of the Bezier curves in order to overcome some of the cause of premature lifetime termination: lack of appropriate workload balancing in the most critical of the network-around the sink and source nodes. By using the Bezier curve as the trajectory model, we are able to control the coverage of the sink/source nearby nodes implicated in the routing and attain, in a practical setting, a near 100% utilization of them. Figure 2 shows that the routing coverage, for workload balancing,

by means of trajectory-based alternating routes, in comparison with the ones which provide routes that approximate, to some degree, the shortest path routing.

The route's information, which is represented through a Bezier curve, must be transmitted from node for the routing purposes. A node, however, needs only to communicate the coordinates of the control points in order to be able to generate an entire Bezier curve, no matter its shaped and length. This property will save both time and energy, a clear advantage of using parametric curves such as Bezier. Moreover, not only on curve can be computed from a set of control points, but an entire family of curves can be also generated, based on the affine property of Bezier curves, which will prove benefit when managing multiple routes between two source-sink points. Bezier curves add flexibility to the routes. If we consider two fixed endpoints, the shape of the curve can be adjusted by using the remaining control points. Figure 5 shows the type of shapes Bezier trajectories can take by simply relocating these control points given the source-sink endpoints p_0 and p_1.
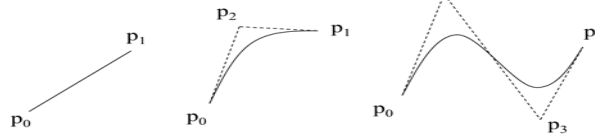

Fig. 5. Linear, quadratic and cubic.

4.2   The analysis of finding multiple node- disjoint paths

Due to the independence of the paths, disjoint paths have been received considerable attention in the recent literature. The main reason is that disjoint paths offer certain advantages over non-disjoint paths. When using non-disjoint routes, a single link or node failure can result in multiple routes to fail while in node or link disjoint routes, a link failure will only cause a single route to fail. What is more, both the nodes and the wireless links are error-prone, which leads to multiple paths that share those nodes and links to fail in non-disjoint paths. Hence, node-disjoint paths can provide the highest degree fault-tolerance.

Many algorithms to find node-disjoint paths make use of request/reply cycles. Typically, a source node initiates a route discovery procedure by broadcasting a Route Request packet, and then this ROUTE REQUEST message is flooded to the entire network. Contrary to the above general algorithm, our k-multipath algorithm to build node-disjoint routes does not generate too much RREQ/RREP packets and then increases the routing overheads. Instead, we fully make use of geometrical knowledge to discover routes. According to the Fig. 2, all the coordinates of k breakpoints can be more easily calculated, in a segment orthogonal to source-sink line segment. In each of k-1 surrounded regions, we only construct one route from the source node to the sink node. Those intermediate nodes in one route are selected in a way that they should be closest to source-sink line segment. This is easily explained by the fact that it takes more time to deliver packets along the path farther away from the source-sink line segment. Multiple paths may present differences in the end-to-end delay of each path. Such scenarios require that data coming from different flows needs to be

buffered till the flow from the path with the highest delay arrives for reordering the data correctly. This solution poses another problem, as high speed memory is extremely expensive, and therefore we should minimize the differential delay. In our simulation, we don't need to consider packet reordering.

5   Performance Evaluation

Our simulation setting is as follows. We create 500 nodes uniformly deployed in a $2 \times 2km2$ area, which use 802.11 protocol at the MAC layer, and the heartbeat node discovery protocol in order to determine the neighbors. We randomly pick up one pair of nodes as source-sink nodes from the physical terrain, and we don't consider the characteristic of the mobility of the nodes. Although how the number of the paths affects the performance remains unknown, there are 5 paths to be used in the simulation. We choose a path randomly from the multiple paths with the same probability. Moreover, we try to keep the number of paths odd.

We study two different ways to use the multiple paths. In one method, called multipath routing 1, we choose a path randomly from the multiple paths with the same probability. The other method, called multipath routing 2, is to choose a path with a probability inversely proportional to the length of the path. We vary the sampling rate in order to observe the effects of packet loss in the nodes due to the interferences among the multipath. We will compare the performance of the shortest path routing and multipath routing in different aspects. We evaluate the performance according to the following metrics:

・The load distribution: This metric provides the average relayed traffic in packets as function of the distance to the network center, in accordance with the Pham and Perreau's analytical model [3]. We use load distribution as a metric to evaluate the load balancing.

・Average energy consumption: The energy consumption is averaged over all nodes in the network.

・The lifetime extension: The metric studied is the number of hours of communication achieves when 1 percent, 25 percent, 50 percent, and 100 percent of the nodes die using multipath routing and the shortest path routing.

・Query Turn-Around Time: A measure of the initial responsiveness of the query; the time lapsed from the moment the query is submitted to the network by the user until the very first data-packet is received at the user. This will measure and penalize the multipath-construction algorithm with a high set-up time.

These measures are intended to provide insight into the ability of the protocols to route packets to their intended destination, and the energy efficiency of the protocols in accomplishing that task. The routing overhead is defined as the ratio of the number of routing messages generated by a routing protocol to the number of received data packets at the destinations. This metric is a measure of how many routing messages are needed to receive one data packet. It captures the efficiency of the routing protocol. Since the routing overhead is similar and much lower between multipath and the shortest path routing in our simulation, we don't present the results of routing overhead in this paper.

Figure 6 portrays the load distribution of the two protocols as function of the distance from the network center. In our simulation, the center is the midpoint of the segment between a pair of source-sink nodes. With the increase of the distance to the center, there is a much more slight decrease of the load for the multipath routing while the load is greatly reduced for the shortest path routing. This simulation shows that our multipath routing can achieve better load balancing. This result can be explained by the fact that the traffic of the network is evenly regulated to the different paths while the single path always chooses the geographic-based shortest path, which will unfairly distribute more loads to the nodes along this optimal route than their neighboring nodes. According to this Figure, we conclude that the shortest path is likely to be overloaded because this route is across or very close to the center. In addition, due to the fact that we adopt load balancing policy, theoretically, all the nodes should experience approximately the same loads in the multipath routing 1, however, there exists a smooth decrease of the loads as the distance increases. The possible reason is that those packets that travel through longer routes are dropped due to more latency.

Still, we notice that as the distance from the network center increases, the number of average load for multipath routing 2 drops faster than multipath routing 1. This can be explained that multipath routing 2 is to choose a path with a probability inversely proportional to the length of the path. In other words, the further the distance to the center, the lower the probability that the nodes are used to relay the packets. Moreover, since our load balancing policy is not optimal, those nodes close to the optimal route have to be assigned more traffic in comparison with ones at the rear. Nevertheless, it is important to stress the fact that our multipath routing outperforms the shortest path routing in terms of load balancing.
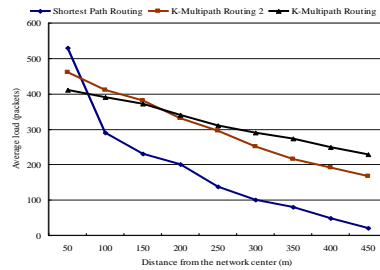

Fig. 6. The average load distribution as function of distance from the network center

Figure 7 portrays the average energy consumption of the three protocols as function of the number of the hours of communication. Clearly, both two multipath routing have smaller energy consumption than that of the shortest path routing. This demonstrates that both multipath routing can distribute the traffic load more fairly than the shortest path routing. This result can be explained by the fact that the traffic of the network is evenly regulated to the different paths while the single path always chooses the geographic-based shortest path, which will unfairly distribute more loads to the nodes along this optimal route than their neighboring nodes. According to this Figure, we conclude that the nodes along this shortest path are likely to be overused because this route is only one. Therefore, the energy of those nodes on this route will drop faster than the other nodes. Moreover, we also notice that there exists a slightly

improvement of energy consumption between multipath routing 1 and multipath routing 2, even compared to the shortest path routing. This is because the nodes, even with no routing tasks, have to passively listen to neighboring nodes' radio transmission, which inevitably consumes battery energy. Even though our multipath routing is not optimal, it is important to stress the fact that our multipath routing outperforms the shortest path routing in terms of energy consumption.
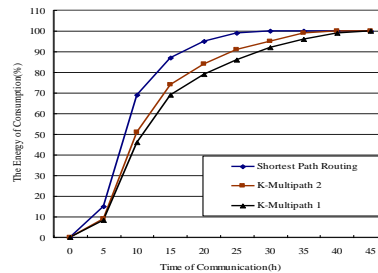

Fig. 7. The average energy consumption as function of the number of hours of communication

Figure 8 illustrates the lifetime extension as function of the percentage of node death. We study the number of hours of communication when 1 percent, 25 percent, 50 percent, and 100 percent of the nodes die using k-multipath routing and the shortest path routing. As can be seen in Figure 10, both of the two multipath routing can yield improvements over the shortest path routing in all cases while the lifetime extension of the multipath routing 1 is trivial compared to the multipath routing 2. To the best of our knowledge, the battery energy of a network node is mainly consumed on forwarding control and data packets. Multipath routing usually increases the energy consumption on the transmission of data messages because some data packets traverse sub-optimal paths. On the other hand, it will decrease the energy consumption on the transmission of control messages. This reveals that k-multipath routing 1 can achieve the best balanced energy dissipation among the sensor nodes to have full use of the complete sensor network. In Figure 9 we have plotted query turn-around time as function of sampling interval. One can see that there is less responsive time in the shortest path. But for the long term and continuous query, since the nodes close the shortest path routing, which are called "hot spots", will quickly deplete the sensor nodes' energy, the impact brought by link failure will greatly increase query response time. This confirms the more significance that we have explored K-Multipath Routing Scheme for Energy Efficient Wireless Sensor Networks.
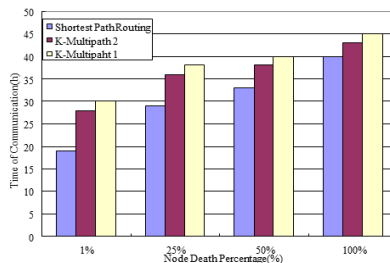

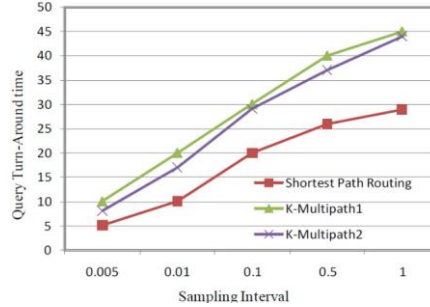**Fig. 8.** The lifetime of the network as function of the percentage of node death

Fig. 9. Query turn-around time as function of sampling interval

6 Conclusions and Future Work

We present a novel load-balancing mechanism for wireless sensor networks. The new scheme is simple but very effective to achieve load balancing and congestion alleviation. We have explored an experimental comparison between k-multipath routing and the shortest path routing. Our performance study shows that the network traffic can be distributed more evenly onto multipath routing. Load balancing is important to fairly distribute the routing task among the nodes of the network. It can also protect a node from failure considering that a node with heavy duty is likely to deplete its energy quickly.

Although it takes much more time for the packet delivery along those multiple paths than the shortest path, the packet delivery fraction of our technique has been improved obviously and the network resource can be utilized efficiently. Also, we use the k-multipath routing to balance the energy dissipation to maximize the lifetime of the nodes in a sensor network. However, minimizing energy in isolation has drawbacks because energy efficiency often brings additional latency. Clearly, several practical applications set limits on acceptable latency, as specified by QoS requirements. For example, the data transmission delay per packet may have a bound. Beyond this bound, this packet may be dropped. Therefore, our motivation of this paper is to investigate the trade-off that arises between the end-to-end delay of the data transmission and the lifetime extension of the individual nodes and the network as a whole. We have explored an experimental comparison between k-multipath routing and the shortest path routing. So through distributing the energy load among the nodes, we can increase the lifetime and quality of data in a sensor network. However, multipath routing techniques are not without pitfalls. It is worthwhile to mention them clearly, even though they are not captured in our simulation. The primary disadvantages of multipath routing protocols compared to single path protocol are complexity and overhead. We have to consider the overheads that we maintain multiple routes to a destination, which leads to large routing tables at intermediate nodes. Also, we need to take into consideration how to allocate the packets to the multiple routes. Hence, we still need to expand our design to provide the solution to the above-mentioned problems in the future.

References

1. X.Chen, M.Chamania, A. Jukan, A. C. Drummond, N. L. S. da, Fonseca: QoS-Constrained Multi-path Routing for High-End Network Applications. IEEE INFOCOM2009 High-Speed Networks Workshop, Rio de Janeiro, Brazil, April 2009.
2. Y. Ganjali, A.Keshavarzian: Load balancing in ad hoc networks: single-path routing vs. multipath routing, in Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2004), March 2004.
3. P.P. Pham and S. Perrau: Performance Analysis of Reactive Shortest Path and Multipath Routing Mechanism with Load Balance, Proc. IEEE INFOCOM Conf., pp. 251-259, Apr. 2003.
4. Sungoh Kwon and Ness B. Shroff: Analysis of Shortest Path Routing for Large Multi-Hop Wireless Networks, IEEE/ACM TRANSACTIONS ON NETWORKING, VOL. 17, NO. 3, pp. 857-869, JUNE 2009.
5. Shaohua Wan, Yanxiang He: Performance analysis of single-tree and split-tree approach in wireless sensor networks, CyberC 2009: International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 132-135, October 2009.
6. Kai-Wei Fan, Sha Liu, and Prasun Sinha: Structure- free Data Aggregation in Sensor Networks. IEEE Transactions on Mobile Computing (TMC), Vol. 6, Number 8, pp. 929-942, August 2007, an earlier version also appeared in INFOCOM'06.
7. P. Skraba, Q. Fang, A. Nguyen, and L. Guibas: Sweeps over wireless sensor networks, 5th Int'l Conference on Information Processing in Sensor Networks (IPSN), pp. 143-151, 2006.
8. Shrivastava N., Buragohain C., Agrawaid., et al: Medians and beyond: new aggregation techniques for sensor networks. In: Proc. of the Second International Conference on Embedded Networked Sensor Systems (SenSys'04). New York, ACM Press, 2004, 239-249.
9. Prasenjit Chanak, Tuhina Samanta, Indrajit Banerjee: Fault-tolerant multipath routing scheme for energy efficient wireless sensor networks, International Journal of Wireless & Mobile Networks (IJWMN) Vol. 5, No. 2, April 2013.
10. Sudarshan Vasudevan, Micah Adler, Dennis Goeckel, Don Towsley: Efficient Algorithms for Neighbor Discovery in Wireless Networks. IEEE/ACM Trans. Networking. 21(1): 69-83 (2013).
11. Parveen Patel, Deepak Bansal, Lihua Yuan, Ashwin Murthy, Albert G. Greenberg, David A. Maltz, Randy Kern, Hemant Kumar, Marios Zikos, Hongyu Wu, Changhoon Kim, Naveen Karri: Ananta: cloud scale load balancing. SIGCOMM 2013: 207-218.