TOLB: A Traffic-Oblivious Load-Balancing Protocol for Next-Generation Sensornets

Mohamed Aly[1] and Anandha Gopalan[2]

[1] Google, Inc.
[2] Dept. of Computer Science, University of Pittsburgh
maly@google.com, axgopala@cs.pitt.edu

Abstract. The multiple expected sources of traffic skewness in Next-Generation SensorNets (NGSN) will trigger the need for load-balanced point-to-point routing protocols. Driven by this fact, we present in this paper a load-balancing primitive, namely Traffic-Oblivious Load-Balancing (TOLB), to be used on top of any point-to-point routing protocol. TOLB obliviously load balances traffic by pushing the decision-making responsibility to the source of any packet without depending on the energy status of the network sensors or on previously taken decisions for similar packets. We present theoretical bounds on TOLB's performance for special network types such as mesh networks. Additionally, we ran simulations to evaluate TOLB's performance on general networks. Our experimental results show the high benefit (in terms of network lifetime and throughput) of applying TOLB on top of routing schemes to deal with various traffic skewness levels in different sensor deployment scenarios.

1 Introduction

Early sensornet deployments targeted data collection and push-based querying, e.g. [7, 36]. Hence, most current sensornet code bases, e.g. [18, 14], mainly offer tree-based many-to-one and one-to-many routing, query broadcasts, and aggregation during the data collection process [25]. Early sensornet applications were based on this model, e.g. monitoring and surveillance applications [37]. However, researchers and practitioners envision Next-Generation Sensor Nets (NGSN) to be composed of sensors deployed everywhere, together with gateways connecting sensors to Internet users/applications [26, 24]. Gateways may be stationary base stations [10], or mobile ones, such as robots, cell phones, and PDAs [35, 34]. Due to their huge number, sensors will tend to be clustered into geographic areas and geographically addressed *relatively* in each area rather than being assigned GPS-based addresses. In this model, querying loads will mostly be composed of pull-based ad-hoc queries issued by mobile users and/or Internet users. Ad-hoc queries trigger the need for using point-to-point routing [20, 12], a different routing paradigm from the old sensornet data collection model.

The large and continuously varying number of query sources in NGSN highly complicates the task of predicting the query distributions. Furthermore, the possibility of traffic skewness, when the sources and/or the destinations of most queries belong to a fairly small subset of sensors, is high. Query hotspots [2], where most queries access a small number of sensors simultaneously, represent one of the major traffic skewness examples. Query hotspots may be in the form of *Data-Centric Storage(DCS) range* queries [2], e.g. many queries asking for a small range of temperature readings stored in one or two sensors, or *geo-centric* queries [26], when many users are simultaneously interested in data generated by sensors in a particular area (e.g. find free parking spots in downtown area). In general, traffic skewness is a major problem that may result in the early death of sensors, network partitioning, and a subsequent reduction in network lifetime. The expected traffic skewness in NGSN introduces the need for robust load-balanced point-to-point routing schemes.

In this paper, we present the *Traffic-Oblivious Load-Balancing (TOLB)* protocol, a load-balancing protocol to be used on top of any point-to-point routing scheme. Our major design goal is *simplicity*. To achieve this goal, we adopt two main concepts: *Traffic-Obliviousness* and *Multipath Routing*. Traffic-Obliviousness means that the route of a packet $p = (s, t)$ is determined independently from the routes of previously issued (s, t) packets throughout the network operation [28, 8, 1]. A *stateless* distributed oblivious routing scheme is one where routing decisions are taken by individual sensor nodes solely based on local information, i.e. with no dependence of the load (energy) status of the remaining network nodes. On the other hand, Multipath Routing means that packets $p = (s, t)$ are routed through different network paths throughout the network operation. Previously presented multipath routing schemes were based on having a *paths enumeration* phase, where a set of network paths P is determined for each packet type (s, t) prior to the network operation. Individual paths of P are interchangeably used to route (s, t) packets based on the load-status of network nodes in a way to balance the energy consumption among all sensors. To blend both concepts together, TOLB substitutes the paths enumeration phase and the dependence on state information in taking routing decisions by *randomization*.

At its core, TOLB is based on a variation of the famous two-stage randomized routing, originally presented by Valiant [33] for bounding congestion in interconnection networks. In plain two-phase routing, an (s, t) packet is first routed to a random intermediate node r before being routed to the final destination t. To maintain obliviousness, TOLB only assumes the ability of each sensor to estimate its location and the approximate boundaries of the network service area. Furthermore, TOLB presents additional optimization heuristics that exploit the power of applying admission-control, making two random choices, and applying partial load-balancing in order to deal with various levels of skewness of both, traffic and node deployments. Through extensive simulations, we show that the major advantages achieved by TOLB are:

- Significantly increasing the network lifetime and throughput against skewed traffic distributions compared to the plain underlying routing scheme. The performance gains achieved by TOLB highly increase when considering typical query semantics such as query-reply pairs and query hotspots. TOLB also exhibits good performance when node densities increase.
- Maintaining a good level of fault tolerance against temporary node failures.

– Maintaining a comparable level of Quality of Service (QoS) and real-time guarantees to that offered by the underlying routing protocol.
– Maintain good performance even under skewed node deployments.

Organization of the paper: The rest of the paper is organized as follows. Related work is presented in Section 2 and the components of TOLB are presented in Section 3. Section 4 presents experimental results and Section 5 concludes the paper and discusses future work.

2 Related Work

In this section, we first provide a quick review of the load-balanced and oblivious routing protocols already presented in literature. Then, we briefly classify the currently available point-to-point routing protocols.

Load-Balanced Routing: Unlike TOLB, all previously presented load-balancing paradigms were embedded in routing protocols. Many of these protocols were based on multipath routing, where a set of paths are determined for each packet type prior to the network operation and paths are interchangeably used afterwords [31, 11]. Directed diffusion [19] presented the idea of finding multiple routes from multiple sources to a single destination while applying in-network data aggregation. Many multipaths routing schemes were later presented based on *Directed Diffusion*, e.g. [13, 29, 6]. Ganesan et al. [13] suggested the use of *braided multipaths* to achieve high resilience and fault-tolerance. Later, Raicu et al. [29] presented a diffusion-based algorithm where load-balancing decisions are made locally using location, power, and load as metrics in order to achieve global energy-efficiency.

Oblivious Routing: The idea of distributed oblivious routing in general communication networks was originally presented by Räcke's seminal paper [28], which later triggered many subsequent improvements, such as [17, 15, 16]. Recently, Busch et al. [8] presented the first theoretical analysis of a valiant-based oblivious routing algorithm on special types of geometric networks like mesh networks and uniform disc graphs. Later, Aly and Augustine [1] addressed the packet admission-control and oblivious routing problem for the first time in sensor networks. The work presented theoretical guidelines for any oblivious routing algorithm to maintain polylogarithmic competitiveness, w.r.t. throughput, against an offline routing algorithm. TOLB uses both, the Valiant paradigm and the admission-control ideas, based on the theoretical guidelines presented by the previous two papers.

Point-to-Point Routing: The need for point-to-point routing has recently increased as many current sensornet applications assume its usage, e.g. data-centric storage (DCS) [32, 4] and muti-dimensional range queries [23, 3, 2]. The first point-to-point routing schemes presented for wireless and sensor networks were based on *geographic routing*, e.g. GPSR [20], where nodes are identified by their geographic coordinates and routing is done greedily. Later, it was pointed that geographic schemes suffer from various limitations, e.g. the inability of current radios to conform with the current planarization algorithms and the unrealistic requirement of GPS-equipped sensors [12, 21]. Driven by these problems, schemes like NoGeo [30] and GEM [27] suggested the use of synthetic *virtual* coordinates assigned by iteratively embedding nodes in a Cartesian plane. Two recent schemes, BVR [12] and Logical Coordinates [9], used a collection of ideas from both geographic and virtual coordinates schemes. The basic idea of both schemes is to let nodes obtain coordinates from a set of landmarks. Routing then minimizes a distance function on these coordinates.

3 The TOLB Protocol

TOLB is designed to run on top of any point-to-point routing protocol to load-balance the traffic in the sensor network. The protocol is composed of a basic load-balancing algorithm and three optimization heuristics that can be set to run based on need. TOLB assumes each sensor node knows its location, the locations of its direct neighbors, and the approximate boundaries of the network service area. Throughout the rest of the paper, we refer to routing any packet $p = (s, t)$ using the underlying point-to-point routing protocol as *routing greedily.* We denote the network service area by A.

We start by presenting TOLB's core load-balancing algorithm.

3.1 The Core Load-Balancing Algorithm

The basic idea of the algorithm is to assign the load-balancing task solely to the source node of any packet. This conforms with the traffic-obliviousness property that TOLB maintains. The algorithm can be described in the following high-level steps:

1. For any packet $p = (s, t)$, the source s selects a location $R = (x_r, y_r) \in A$ at random and routes p *greedily* to R.
2. Let the closest node to the location R be r. Upon receiving p, r routes it greedily to its destination t.

Before describing the implementation details of TOLB, it should be noted that the mechanism for forwarding a packet from a sensor to one of its neighbors depends on the underlying routing algorithm. For example, geographic routing algorithms like GPSR [20] use physical coordinates while algorithms like GEM [27] use logical coordinates. The exact mechanism by which a node determines the location of any sensor in the network is beyond the scope of this paper. Furthermore, the random location selected is based on the assumption that a sensor knows the approximate physical boundaries of the network service area, A (in case of physical coordinates) or the virtual boundaries of A (in case of virtual coordinates).

Upon receiving $p = (s, t)$, s selects a random $R = (x_r, y_r) \in A$. It then sets the *first* destination of p to be R and the *second* destination to be t while setting a *destination flag* to 0 to indicate that the packet should be sent to its first destination. Then, s greedily routes p to its first destination R. Upon receiving p, each intermediate node first checks the destination flag to determine whether to forward p to the first or the second destination. When an intermediate node determines that it is closer to R than any of its neighbors, it sets p's destination flag to 1 indicating that p should be forwarded to its second destination. The process ends when p reaches its final destination t.

The load-balancing effect of the algorithm results from the randomized selection of R. As packets are routed greedily, this results in rotating the use of the different available paths in the network for routing (s, t) packets as time progresses. It is possible to prove theoretical bounds on the performance of the algorithm on special network types, such as mesh networks where each sensor can communicate with its four direct neighbors. The performance is measured in terms of stretch and congestion. Stretch is defined as the maximum ratio of a path length (in hops) of any (s, t) packet to that of the respective shortest path of that packet and congestion is defined to be the maximum number of paths using any node in the network. Let C_{opt} be the minimum congestion of the optimal offline load-balancing algorithm. The following theorem, whose proof directly follows from the proofs presented by Busch et al. [8], describes the performance of our algorithm on mesh networks.

Theorem 1. *TOLB achieves an $O(1)$ stretch and an $O(\log n) * C_{opt}$ congestion on both mesh networks and uniform disk graphs assuming $S = A$ and any underlying greedy routing protocol.*

We now present additional heuristics that would improve the performance of the TOLB load-balancing algorithm in special network settings, e.g. skewed node deployments.

3.2 The Admission-Control Protocol

The importance of the presence of an admission-control protocol in an oblivious routing scheme was raised by the following theorem proved by Aly and Augustine [1]. Let an always-send routing algorithm be the one where a packet p received by any node k is forwarded toward its destination as long as k has enough energy to forward p to one of its neighbors.

Theorem 2. *[1] Given a balanced binary tree $T(V, E)$ and a set of demands D, an always-send distributed oblivious routing algorithm A_{as} cannot maintain polylogarithmic competitiveness if either:* 1. *D is a set of adversarial demands, or follows a general distribution that is unknown to all sensor nodes; or* 2. *an adversary sets the tree node capacities (internal nodes or leaf nodes).*

This theorem shows that any distributed oblivious routing algorithm needs a concrete admission-control protocol in order to achieve polylogarithmic competitiveness with respect to throughput in the context of sensor networks. Although our goal is not directly to achieve polylogarithmic competitiveness, however, we use this theorem as an indication showing that the presence of an admission-control protocol improves the performance of any oblivious routing algorithm. The TOLB protocol is intended to maintain obliviousness on top of any routing algorithm. Thus, the combination of TOLB and the underlying routing algorithm can be considered as a distributed oblivious routing algorithm. In light of the above theorem, we append TOLB with a packet admission-control protocol. The basic idea that we present here is that some local information may be available at individual sensors. The usage of this information would improve TOLB's performance without ruling out its obliviousness property.

Before presenting the protocol, we define the *counters-list* as a list of counters maintained by each sensor node and containing a counter corresponding to each of the node's neighbors. Based on this definition, our admission-control technique can be summarized in the following points.

– Initially, all counters are set to zero. Whenever a packet is forwarded to neighbor j from node i, i increments the counter corresponding to j by 1.
– Whenever a packet p, arising in node s, is to be greedily forwarded to neighbor j of s, s compares j's counter value to the values of the counters of the rest of its neighbors. If the ratio of j's counter to the sum of all counters exceeds a threshold c, s reruns TOLB's core load-balancing algorithm to get another neighbor j'. The process is repeated for several times till an unloaded neighbor j_u is selected and p is then forwarded to j_u.

It is important to note that the admission-control protocol presented above did not use any information passing technique. Instead, all information used was inferred by keeping track of the number of packets sent through each direction (and subsequently through each neighbor). The intuition behind this is that the number of packets forwarded to each neighbor can be considered as an approximation of the total number of packets that passed through all the paths on which the neighbor falls. Subsequently, this can show us rough indications about the energy status of nodes along these paths.

The importance of the above admission-control protocol lies in achieving load-balancing for networks with skewed node deployments. A first example of such a setting is a network containing different node densities in different areas. For simplicity, we can think of a network with two sides, a left side with scarce sensor deployment and a right side with dense sensor deployment. We define the *skewness path* to be the path between the source and destination for some s-t pair that constitutes a large portion of the total network traffic. Let the skewness path be between the two sides of the network and the source be falling on the intersection line between the two sides. When applying TOLB's core load-balancing algorithm, the source's neighbors falling on the left side will be more loaded that those falling on the right side. Thus, these neighbors would be depleted with a faster rate than the right-side neighbors. In such a case, applying the admission-control protocol would help in improving the level of load-balancing achieved.

A second setting where admission-control shines is when s is the source of some traffic skewness in the network and one of its neighbors is close to death. Note that this information can be easily deduced from the counter value corresponding to this neighbor (thus, the number of packets sent through this neighbor), i.e., without any wireless communication dedicated for such reason assuming all sensors start the network operation with equal amounts of energy. In such a case, the admission-control may decide to take this symptom as a sign showing that most of the nodes on the path that will be followed by the packet are either dead or closed to death. This information can be used to take a decision of not sending any further packets along this path.

3.3 The Two-Choices Paradigm

We now move on to present the two-choices paradigm whose main goal is to enforce load-balancing in another type of skewed deployment. The idea of this paradigm comes from the famous balls-and-bins model. It is well knows that, given n balls that are thrown at random, one at a time, into n bins, the maximum load of a bin is approximately $\log n / \log \log n$ with high probability. Azar et al. [5] showed that in case, for each ball, two random bin selections are made and the ball is thrown in the least loaded bin among the two, the maximum load of a bin drops to $\theta(\log \log n)$, with high probability.

The important implication of the above result is that even a small amount of choice can lead to a significantly improved performance of randomized load-balancing algorithms. Using this intuition, we try to exploit the power of making two choices in TOLB. The paradigm maintains a *counters-list* (already defined in Section 3.2) at each sensor and it works as follows. Whenever a packet p arises in s, s makes two random choices rather than one by selecting two locations R_1 and R_2, both within the boundary of A. Among the two routes s-R_1-t and s-R_2-t, the idea is to try to route p through the route with least loaded nodes (i.e. with higher energy). However, knowing information about the paths' energy status contradicts with traffic-obliviousness. To cope with this problem, s determines the two neighbors j_1 and j_2 that will be involved in greedily routing the packet to R_1 and R_2, respectively. s then picks the location R_i whose j_i's counter has the smaller value and uses this location as the intermediate destination for p.

Like the admission-control protocol, the two-choices paradigm uses the values of the counters, representing messages sent through different neighbors, to get an approximate idea on the energy status of the nodes in the directions (and subsequently areas) corresponding to these neighbors. This idea is exploited by the paradigm to achieve load-balancing for skewed network settings such as *networks with gaps*. As an example, consider a network with randomly distributed sensors, but containing one or more gaps with no sensors in them (due to geographic obstacles, temporary or permanent node failures, etc). Let the source s of some skewness path be existing on the border of one of these gaps. Looking at the neighbors of s, we realize that its direct neighbors falling on the border of the gap will be more loaded than neighbors falling in other locations. For such a setting, applying the two-choices paradigm would be beneficial as it would be unlikely for the two random choices to fall on the border of the gap.

3.4 The Partial Load-Balancing Heuristic

Although load-balancing is an extremely important primitive for any routing protocol, a robust load-balancing protocol should be able to deal with different levels of traffic skewness. Subsequently, we present the option of partial load-balancing in TOLB in order to account for the possibility of regular or slightly skewed traffic loads.

In its high level, the partial load-balancing heuristic can be summarized by applying the TOLB protocol for a subset of the packets injected in the network rather than for every packet arising in any network node. This can be done by defining the *load-balancing factor* $0 \leq \epsilon \leq 1$ which has a unique value for all sensor nodes. Whenever a packet arises in a sensor s, s applies the TOLB protocol with probability ϵ and *greedily* routes the packet immediately to its destination t with a probability $(1 - \epsilon)$. The value of ϵ should be adaptively set in direct proportion to the expected traffic skewness level, thus, high when traffic is expected to be highly skewed and low otherwise. This mixed usage of the two versions of the routing algorithm, with and without TOLB, results in a limited load-balancing effect that is still better than solely using the underlying routing algorithm when a low level of skewness occurs in the network. It is worth mentioning that the traffic skewness level may be determined based on mining the query load history using machine learning techniques that are orthogonal to our concern in this paper.

Now that we have described all the components of TOLB, we move on to experimentally validate its performance in the following section.

4 Experimental Results

In order to evaluate the performance of TOLB, we implemented both TOLB and GPSR using the Glomosim wireless network simulator [22]. We simulated a sensornet cluster of the NGSN. The network is assumed to have *multiple base stations*, both stationary and mobile, acting as sources of the (skewed) traffic. In our experiments, sensors are assumed to be randomly distributed in A (unless otherwise mentioned). Sensors have an equal starting energy amount of $e = 100$ units. Sending or receiving one packet consumes 1 energy unit. Whenever a sensor s sends a packet to one of its neighbors t, only s and t consume energy for sending and receiving this packet, respectively. Furthermore, the wireless medium is assumed to be reliable and does not contribute to any packet loss. Each sensor is assumed to know the approximate boundaries of the service area, A. Also, a sensor is assumed to know its location and the approximate location of any sensor in the network.

To evaluate TOLB's *scalability*, we ran simulations for networks of sizes varying between 1000 and 2000 sensors. The network service area, A spanned a 200x200 square. At the start of every simulation, node locations are picked at random (except for the case of skewed deployments) and multiple stationary base stations are picked at random locations. The number of these base stations is fairly small compared to that of sensors. Initially, each node sends 1 broadcast message to know its neighbor's locations and it receives as many messages as the number of its direct neighbors. No maintenance messages are further sent during the simulation.

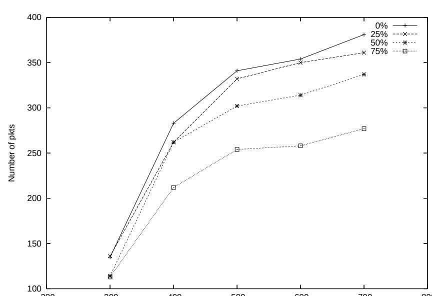
Traffic is generated as follows. At the start of the network operation, a small number of stationary base stations are randomly selected. Then, a small amount of destinations are randomly selected. Then, a high percentage of the generated packets are sent between the base stations and the selected destinations. We define this percentage to be the *skewness factor*, x. The rest of the packets are sent from random sources to random destinations. These are meant to be queries issued by mobile nodes, picked by a nearby sensor, and targeting another sensor in the network.

We implemented our protocol on top of GPSR. In measuring TOLB's performance in the different simulations, we focused on two metrics: *Network Lifetime* and *Throughput*. We define the network lifetime to be the time elapsed before the first node death in the network. Throughput denotes the number of successfully sent packets by all network nodes before the network dies. Network lifetime gives an idea of how TOLB load-balances the energy consumption among the different sensor nodes. Throughput on the other hand shows how load-balancing skewed traffic increases the network performance in terms of the number of successfully sent packets.

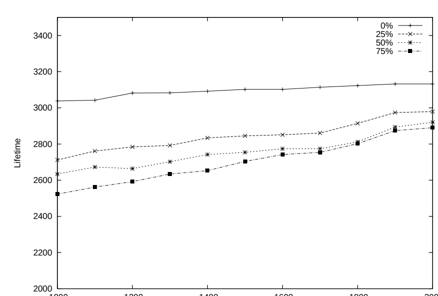
Simulation results are presented in the following subsections. Note that we only present part of our findings due to space constraints. In each of the graphs below, a point represents the average of 10 runs. It is worth mentioning that we were aware of the standard deviation in all simulation runs and we did not encounter a relatively large variance in any of the simulations.

4.1 Effect of Traffic Skewness Degree

In the first set of simulations, we changed the skewness factor x from 0% to 75% to get an idea on TOLB's performance for different traffic skewness levels. Figure 1 shows results in terms of both throughput and network lifetime. Figure 1(a) shows that the difference in throughput is almost constant between the 0% and the 75% cases. Furthermore, Figure 1(b) shows that the difference in lifetime between both cases decreases with the increase in network size. An important TOLB characteristic that can be deduced from both Figures is that its performance does not highly degrade or depend on the skewness factor.



(a) Throughput



(b) Network Lifetime

Fig. 1. TOLB Performance against Different Skewness Levels

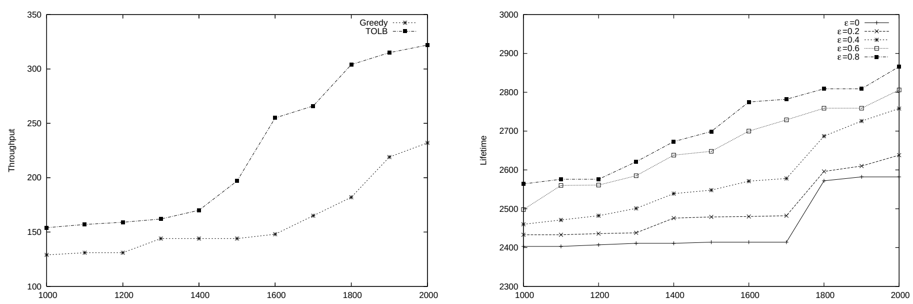
4.2 Performance Gain Over Greedy

We now present the results that show the effect of our basic TOLB protocol on increasing the throughput when compared to the plain GPSR algorithm.

Figure 2(a) demonstrates this comparison for a skewness factor of 70%. The Figure is an example of the big performance gain that TOLB achieves based on load-balancing traffic when compared to plain GPSR.

4.3 Benefit of Partial Load-Balancing

We study here the effect of TOLB's partial load-balancing heuristic on increasing network lifetime when traffic is close to regular. We set x to be 30%. For this lowly skewed traffic, we change ϵ from 0 to 0.8. Figure 2(b) shows that the lifetime increases proportionally with ϵ. This implicitly shows that the overheard imposed by TOLB on the different sensors due to using longer paths is totally dominated by the gain achieved by its load-balancing functionality.



(a) TOLB vs Plain Greedy (b) Partial Load Balancing

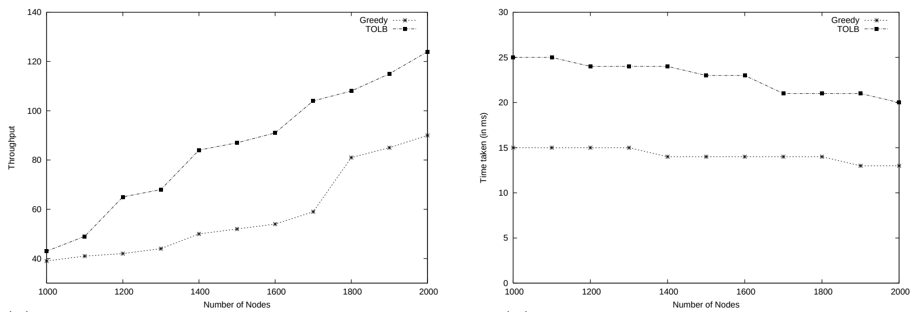
Fig. 2. TOLB-Greedy Comparison and A Study of Different Levels of Partial Load-Balancing

4.4 Fault Tolerance

Failures can occur in the sensor network *temporarily* because of environmental conditions or due to the application of a specific energy saving scheme or *permanently* because of node deaths. It is important to test TOLB's performance against the various types of failures. We focused on temporary node failures as they capture the typical sensor network behavior. Thus, we assumed sensors have two modes: On and Off. We then introduced a random distribution to model temporary node failures. For $x = 70\%$, Figure 3(a) shows that the difference in throughput between TOLB and plain GPSR increases proportionally with the network size. This is a direct consequence of the multipath routing that TOLB imposes on GPSR for load-balancing.

4.5 Quality of Service (QoS)

Real-time applications represent an important characteristic of next-generation sensornets. In our application, for example, a user issuing a query can not tolerate waiting a long time without receiving a result. Motivated by this fact, we compared TOLB to plain GPSR in terms of the average time taken by packets to reach their destination. This metric is a twin of the average packet path lengths (Note that the stretch, which is the maximum path length, has been used in the theoretical analysis of TOLB already presented in Section 3.1). Figure 3(b) shows that the difference between TOLB and plain GPSR is almost constant for different network sizes when $x = 70\%$. An important implication of this result is that TOLB does not impose a large or increasing degradation in QoS on the underlying routing algorithm.



(a) TOLB Performance against Node Deaths

(b) QoS in terms of Packets Timing

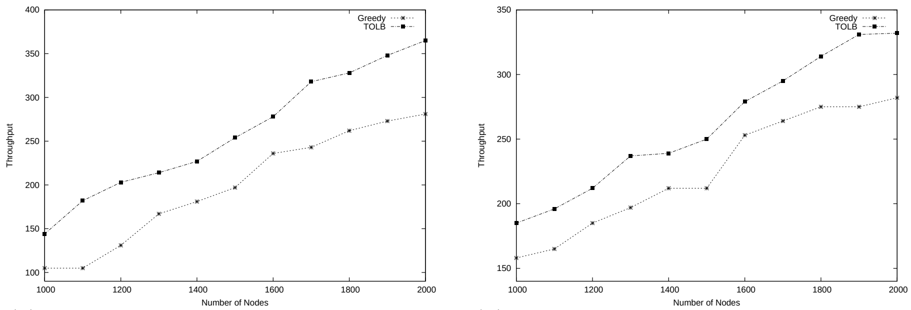
Fig. 3. Fault Tolerance and Quality of Service

4.6 Effect of Correlated Requests

Motivated by our underlying application, a user issuing a query is expected to get an "immediate" answer for such a query. When mapped to requests, this means that whenever an $(s - t)$ is issued, a $(t - s)$ is issued accordingly. Furthermore, if the query spans more than one request, this would mean that multiple requests with the same source s would be issued to many destinations t_j, and replies will be sent back from these destinations to s. Figure 4 presents simulation results modeling these two types of correlated requests. As GPSR acts deterministically, the effect of traffic skewness is magnified. However, load-balancing traffic through multiple $(s - t)$ paths helps in reducing the overhead imposed on individual paths. This is obvious by the relatively large performance gain achieved by TOLB for both types of requests (Figures 4(a) and 4(b)).

4.7 TOLB vs Skewed Node Deployments

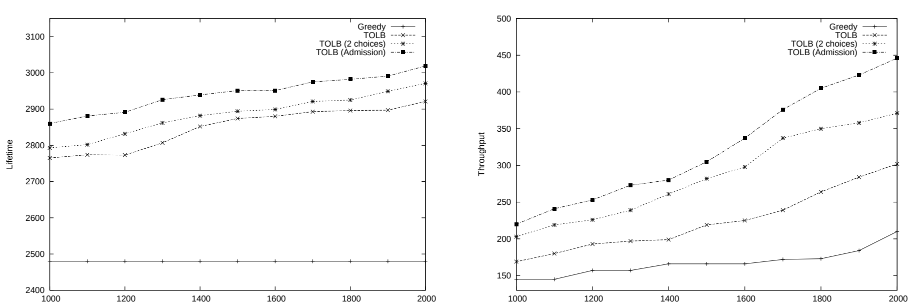
Due to environmental conditions, achieving random or uniform node distributions is difficult. This problem is largely magnified in next-generation sensornets as sensors are stationary and new node deployments are not frequent after the network is initially deployed. For such reason, we simulated skewed distributions of node deployments and networks with *gaps* to compare the effects of applying TOLB's admission-control protocol and two-choices paradigm on top of TOLB with basic TOLB and plain GPSR. Figure 5 shows the results of these simulations for $x = 70\%$. Figure 5(a) shows how applying the admission-control



(a) Source-Destination-Source Requests (b) Source-Destinations-Source Requests

Fig. 4. TOLB vs Correlated Requests

protocol results in increasing the network lifetime achieved by TOLB when compared to the basic TOLB protocol. Of course, the difference between basic TOLB and GPSR is considerably large because of the deterministic behavior of GPSR and the ability of basic TOLB to overcome this problem due to the use of randomization. It can be seen that TOLB with the two-choices paradigm achieves a better performance than basic TOLB. Though this performance is not as good as that of TOLB with admission-control, however, this demonstrates the effect of making two random choices and that it is comparable to using the full knowledge of local information as in the admission-control option. Similar intuitions are valid for Figure 5(b). However, in this case the difference between the effect of the admission-control protocol and that of the two-choices paradigm decreases because of the random distribution of the sensors (outside the gap). Note that we used a threshold $c = 60\%$ for the admission-control protocol. This is just an indicative value and further analysis can provide us with the optimal c.



(a) Scarce-Dense Networks (b) Networks with a Gap

Fig. 5. Skewed Node Deployments

5 Conclusions and Future Work

In this paper, we presented the Traffic-Oblivious Load-Balancing protocol (TOLB), a load-balancing primitive to be run on top of any point-to-point routing scheme in order to deal with traffic skewness. TOLB is based on the famous Valiant two-phase randomized routing paradigm previously presented used in interconnection networks and communication networks. Additionally, TOLB presents three optimization heuristics that apply admission control, exploit the power of two random choices and partial load-balancing to maintain load-balancing for skewed sensor deployments. We evaluated TOLB theoretically and experimentally to show its ability to load-balance different levels of traffic skewness.

We are currently implementing TOLB on sensornet testbeds to physically test its performance for various network sizes and settings. In the future, we would like to devise traffic skewness detection techniques that can quickly determine the level of traffic skewness and adaptively set the TOLB parameters to deal with the specific skewness level.

References

[1] M. Aly and J. Augustine. Online packet admission and oblivious routing in sensor networks. In *Proc. of ISAAC*, 2006.

[2] M. Aly, P. K. Chrysanthis, and K. Pruhs. Decomposing data-centric storage query hot-spots in sensor networks. In *Proc. of MOBIQUITOUS*, 2006.

[3] M. Aly, N. Morsillo, P. K. Chrysanthis, and K. Pruhs. Zone Sharing: A hot-spots decomposition scheme for data-centric storage in sensor networks. In *Proc. of DMSN*, 2005.

[4] M. Aly, K. Pruhs, and P. K. Chrysanthis. KDDCS: A load-balanced in-network data-centric storage scheme in sensor network. In *Proc. of CIKM*, 2006.

[5] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal. Balanced allocations. *SIAM Journal on Computing*, 1999.

[6] S. J. Baek and G. de Veciana. A scalable model for energy load balancing in large-scale sensor networks. In *Proc. of 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2006.

[7] P. Bonnet, J. Gehrke, and P. Seshadri. Towards sensor database systems. In *Proc. of MDM*, 2001.

[8] C. Busch, M. Magdon-Ismail, and J. Xi. Oblivious routing on geometric networks. In *Proceedings of SPAA*, 2005.

[9] Q. Cao and T. Abdelzaher. A scalable logical coordinates framework for routing in wireless sensor networks. In *Proc. of RTSS*, 2004.

[10] H. Dai and R. Han. Unifying micro sensor networks with the internet via overlay networking. In *Proc. of LCN*, 2004.

[11] S. Dulman, T. Nieberg, J. Wu, and P. Havinga. Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks. In *Proc. of WCNC*, 2003.

[12] R. Fonseca, S. Ratnasamy, J. Zhao, C. T. Ee, D. Culler, S. Shenker, and I. Stoica. Beacon Vector Routing: Scalable point-to-point routing in wireless sensornets. In *Proc. of NSDI*, 2005.

[13] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5, 2001.

[14] L. Girod, T. Stathopoulos, N. Ramanathan, J. Elson, D. Estrin, E. Osterweil, and T. Schoellhammer. A system for simulation, emulation, and deployment of heterogeneous sensor networks. In *Proc. of SenSys*, 2004.

[15] M. Hajiaghayi, J. H. Kim, T. Leighton, and H. Räcke. Oblivious routing in directed graphs with random demands. In *Proc. of STOC*, 2005.
[16] M. Hajiaghayi, R. D. Kleinberg, T. Leighton, and H. Räcke. New lower bounds for oblivious routing in undirected graphs. In *Proc. of SODA*, 2005.
[17] C. Harrelson, K. Hildrum, and S. Rao. A polynomial-time tree decomposition to minimize congestion. In *Proc. of SPAA*, 2003.
[18] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In *Proc. of ASPLOS*, 2000.
[19] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions on Networking (TON)*, 11, February 2003.
[20] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless sensor networks. In *Proc. of ACM Mobicom*, 2000.
[21] Y.-J. Kim, R. Govidan, B. Karp, and S. Shenker. On the pitfalls of geographic face routing. In *Proc. of DIALM-POMC*, 2005.
[22] L. Bajaj, M. Takai, R. Ahuja, R. Bagrodia and M. Gerla. Glomosim: A scalable network simulation environment. Technical Report 990027, UCLA, May, 1999.
[23] X. Li, Y. J. Kim, R. Govidan, and W. Hong. Multi-dimensional range queries in sensor networks. In *Proc. of ACM SenSys*, 2003.
[24] T. Luckenbach, P. Gober, S. Arbanowski, A. Kotsopoulos, and K. Kim. Tinyrest - a protocol for integrating sensor networks into the internet. In *Proc. of REALWSN*, 2005.
[25] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. volume 36, pages 131–146, New York, NY, USA, 2002. ACM Press.
[26] S. Nath, J. Liu, J. Miller, F. Zhao, and A. Santanche. Sensormap: a web site for sensors world-wide. In *Proc. of SenSys*, 2006.
[27] J. Newsome and D. Song. GEM: Graph embedding for routing and data centric storage in sensor networks without geographic information. In *Proc. of SenSys*, 2003.
[28] H. Räcke. Minimizing congestion in general networks. In *Proc. of FOCS*, 2002.
[29] I. Raicu, L. Schwiebert, S. Fowler, and S. K. Gupta. Local load balancing for globally efficient routing in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 1, 2005.
[30] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica. Geographic routing without location information. In *Proc. of ACM Mobicom*, 2003.
[31] R. C. Shah and J. M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, 2002.
[32] S. Shenker, S. Ratnasamy, B. Karp, R. Govidan, and D. Estrin. Data-centric storage in sensornets. In *Proc. of HotNets-I*, 2002.
[33] L. G. Valiant. A scheme for fast parallel communication. *SIAM Journal on Computing*, 11, 1982.
[34] Z. Vincze, D. Vass, R. Vida, A. Vidacs, and A. Telcs. Adaptive sink mobility in event-driven multi-hop wireless sensor networks. In *Proc. of InterSense*, 2006.
[35] C. Westphal. Scaling properties of routing protocols in sensor networks with mobile access. Technical report, Nokia, July 2006.
[36] Y. Yao and J. Gehrke. Query processing for sensor networks. In *Proceedings of CIDR*, 2003.
[37] J. Zhao, R. Govindan, and D. Estrin. Sensor Network Tomography: monitoring wireless sensor networks. *Computer Communication Review 32(1)*, 64, 2002.