Overload Traffic Management for Sensor Networks

CHIEH-YIH WAN
Intel Corporation
SHANE B. EISENMAN
Columbia University
ANDREW T. CAMPBELL
Dartmouth College
and
JON CROWCROFT
Cambridge University

There is a critical need for new thinking regarding overload traffic management in sensor networks. It has now become clear that experimental sensor networks (e.g., mote networks) and their applications commonly experience periods of persistent congestion and high packet loss, and in some cases even congestion collapse. This significantly impacts application fidelity measured at the physical sinks, even under light to moderate traffic loads, and is a direct product of the *funneling effect*; that is, the many-to-one multihop traffic pattern that characterizes sensor network communications. Existing congestion control schemes are effective at mitigating congestion through rate control and packet drop mechanisms, but do so at the cost of significantly reducing application fidelity measured at the sinks. To address this problem we propose to exploit the availability of a small number of all wireless, multiradio *virtual sinks* that can be randomly distributed or selectively placed across the sensor field. Virtual sinks are capable of siphoning off data events from regions of the sensor field that are beginning to show signs of high traffic load. In this paper, we present the design, implementation, and evaluation of *Siphon*, a set of fully distributed algorithms that support virtual sink discovery and selection, congestion detection, and traffic redirection in sensor networks. Siphon is based on a Stargate implementation of virtual sinks that uses a separate longer range radio network (based on IEEE 802.11) to siphon events to one or more physical sinks, and a short-range mote radio to interact with the sensor field at siphon points. Results from analysis, simulation and an experimental 48 Mica2 mote testbed show that virtual sinks can scale mote networks by effectively managing growing traffic demands while minimizing any negative impact on application fidelity. Additionally, we show the scheme is competitive with respect to energy consumption compared to a network composed of only motes.

This article extends work published in the proceedings of ACM SenSys 2005.
Authors' addresses: C.-Y. Wan, Intel Research, Intel Corporation, Hillside, OR; Shane B Eisenman; email: shane@ee.columbia.edu.

Categories and Subject Descriptors: C.2.1 [**Computer-Communications Networks**]: Network Protocols—*Wireless communication*

General Terms: Algorithms, Design, Performance, Experimentations

Additional Key Words and Phrases: Simulations, system design, testbeds

ACM Reference Format:

Wan, C. -Y., Eisenman, S. B., Campbell, A. T., and Crowcroft, J. 2007. Overload traffic management for sensor networks. ACM Trans. Sens. Netw. 3, 4, Article 18 (October 2007), 38 pages. DOI = 10.1145/1281492.1281493 http://doi.acm.org/10.1145/1281492.1281493

1. INTRODUCTION

There is a growing understanding that existing experimental mote networks of any appreciable size (e.g., 40+ motes) will operate only under fairly light workloads and are easily driven into overload conditions, and at the extreme, congestion collapse [Hull et al. 2004], rendering these networks nonoperational at the exact moment they need to be used to report a certain phenomenon. There is considerable interest by the research community in developing better radios [Chipcon 2006], MACs, and distributed control algorithms [Wan et al. 2003] that can boost the performance of these networks, allowing them to operate in a stable manner under varying workloads while delivering suitable *application fidelity* (e.g., as simple as events/sec, or more complex) [Tilak et al. 2002] at the sinks. Many technical barriers stand in the way of this goal, however.

One significant challenge is that sensor networks exhibit a unique *funneling effect* where events (e.g., periodic, discrete, and impulse traffic) generated under varying workloads (e.g., light, moderate, high loads) move quickly toward one or more sink points, as illustrated in Figure 1. The flow of events has similarities to the flow of people from a large arena after a sporting event completes. A major limitation in the design of existing sensor networks is that they are ill equipped to deal with the funneling of events and increasing traffic demands. This leads to increased transit traffic intensity [Wan et al. 2004], congestion [Wan et al. 2003], and large packet loss [Zhao and Govindan 2003] (which translates into wasted energy and bandwidth). As a result, the sensors nearest the sink will use energy at the fastest rate, shortening the operational lifetime of the network.

A number of aggregation and congestion control techniques have been proposed that help counter this funneling effect. The aggregation [Intanagonwiwat et al. 2000] of data events can help offset congestion and the disproportionate amount of energy consumed by forwarding nodes located nearer the sink by trading off computation and communications resources. Because of the buildup of traffic close to the sink, loss of aggregated data packets is also more likely [Wan et al. 2004]. This can severely impact the reporting capability (i.e., the fidelity) of the network to meet the application's needs. While there has recently been some advances in developing general theoretical models for aggregation, it is unlikely that existing aggregation techniques [He et al. 2004] alone can resolve the congestion problem and funneling effect.

Recently, a number of congestion control schemes [Sankarasubramaniam et al. 2003; Wan et al. 2003; Hull et al. 2004] have been proposed for sensor networks. However, these schemes do not adequately address the funneling

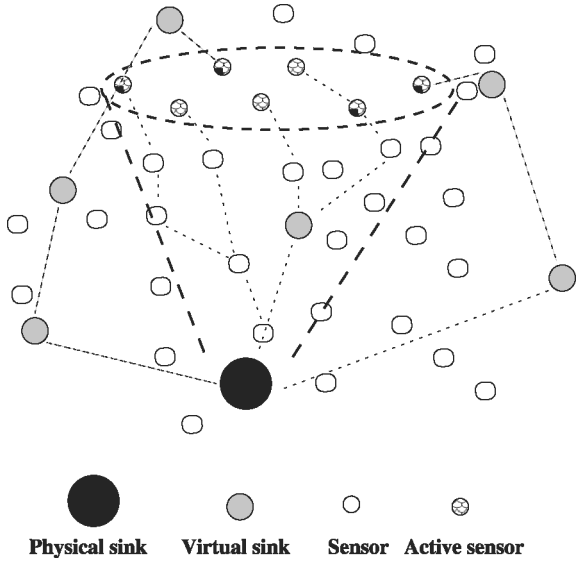

Fig. 1. The funneling effect. Sensors within the range of an event region/epicenter (enclosed by the dotted ellipse) generate data events that travel along a propagation funnel (enclosed by dotted line) toward the sink when an event occurs.

effect. Typically these congestion avoidance mechanisms assume that all nodes are equal (with the exception of the sink) in reacting to the onset of congestion. In this case the congestion avoidance algorithms are distributed across the sensor field and act uniformly. When congestion occurs and the channel is saturated, the application fidelity measured at the sink degrades because the congestion control policy at the sources and intermediate forwarding nodes is to rate control the traffic or even drop event packets during periods of transient and persistent congestion. This raises the question of whether alternative or complementary control solutions exist that could maintain application fidelity at the sink, even in the case of increased workload or overload conditions.

To address this problem we propose to randomly distribute or selectively place (as the case may be) a small number of all-wireless multiradio *virtual sinks* that are capable of offering overload traffic management services to the existing low-power sensor network. While such special nodes can be exploited to support a variety of application-specific (e.g., aggregation, coding, low-delay transport) and common network functions (e.g., storage, localized activation), we focus in this article on their ability to selectively "siphon off" data events from regions of the sensor field that are beginning to show signs of overload. In essence, virtual sinks operate as safety valves in the sensor field that can be used on demand to divert selected packets from areas of high load, alleviating the funneling effect, in order to maintain the fidelity of the application signal at the physical sink.

While a multiradio sensor platform (e.g., [Stargate 2006]) is feasible, the monetary cost is still much higher than a single-radio platform (e.g., Berkeley

motes series [Hill et al. 2000]). Therefore, the cost of deploying a large number of multiradio sensor platforms in sensor networks is prohibitive. Recently, small world studies have shown that a small fraction of shortcut nodes randomly distributed in a network is enough to effectively reduce the network diameter, resulting in a fast distribution network. In Wan et al. [2004], we show that when less than 5% of the nodes are shortcut nodes (e.g., virtual sinks) then the average distance between nodes is halved for a 100-node test bed. This indicates that only a small number of virtual sinks would be needed to form a fast, low-diameter secondary distribution radio network to redirect traffic to a physical sink. This result motivates our work and underpins the viability of the virtual sink concept, both technically and economically.

We call these specialized nodes virtual sinks to distinguish them from the *physical sinks* that typically provide a gateway to the Internet via a wireline, satellite, or wide area cellular interface. Virtual sinks are equipped with a secondary long-range radio interface (e.g., IEEE 802.11, perhaps WiMAX in the future), in addition to their primary low power mote radio. Virtual sinks are capable of dynamically forming a secondary ad hoc radio network that is rooted at a physical sink. Rather than rate controlling event flows or dropping events as is the case with existing congestion control techniques, virtual sinks take selected traffic off the low-powered sensor network (i.e., off the primary radio network) before the onset of congestion, and move it to the physical sink using the secondary radio network.

We present the design, implementation, and evaluation of *Siphon*, a set of fully distributed algorithms that support virtual sink discovery and selection, congestion detection, and traffic redirection in sensor networks. Siphon is based on a Stargate realization of virtual sinks, with a longer-range radio based on IEEE 802.11 to siphon events to one or more physical sinks, and a short-range mote radio to interact with the sensor field at siphon points.

The article is organized as follows. Section 2 discusses the detailed design of the Siphon algorithms. While our design addresses overload traffic management in sensor networks we believe the Siphon algorithms are generally applicable to a broader class of applications that need to exploit special nodes with additional capability (e.g., multiradio, more computational capability or more storage, expedited or low-delay transport services). Section 3 studies the performance properties of Siphon using the ns-2 simulator, which is enhanced to support dual radio virtual sink nodes. The analysis emphasizes scaling issues beyond our existing experimental network. We also study how Siphon operates with Directed Diffusion [Intanagonwiwat et al. 2000]. Section 4 provides an experimental evaluation of Siphon in a 48 Mica2 mote test bed running Surge using a small number of Stargates as virtual sinks. We study a number of different configurations and workloads and show that virtual sinks are capable of increasing the fidelity of the mote network while minimizing the energy tax [Wan et al. 2003]. Additionally, we investigate the energy consumption implications of using virtual sinks in our testbed and show that the Siphon scheme is at least competitive in terms of energy consumption when used in an *on-demand* fashion despite the relatively higher transmission energy of the Stargate's long-range radio compared to a mote's short range radio. Section 5

presents the related work. Finally, we offer some concluding remarks in Section 6. This article presents an extended version of work [Wan et al. 2005] first presented at ACM SenSys 2005.

2. SIPHON DESIGN

In what follows, we discuss the detailed design of the Siphon algorithms for (i) virtual sink discovery and visibility scope control, (ii) congestion detection, (iii) traffic redirection, and finally (iv) congestion avoidance in the secondary network.

2.1 Virtual Sink Discovery and Visibility Scope Control

Like many new services, we envision Siphon may be deployed in an incremental fashion, either for logistical reasons or in response to anticipated traffic characteristics. Specifically, the physical sink might not be equipped with a secondary radio. As a result, there is no guarantee that the virtual sinks can form a connected secondary network rooted at a physical sink through their long-range radio. Furthermore, due to the relatively sparse required concentration of virtual sinks, as discussed in Section 3.8, there is no assurance that a virtual sink is adjacent to a congested region. Consequently, a congested node requires a method to discover, in an energy-efficient manner, the existence of a local virtual sink that could be multiple hops away.

We propose an in-band signaling approach that embeds a *signature byte* into any periodic control packets originated by a physical sink. In typical sensor network applications, a physical sink is required to send periodic signaling into the network for management purposes. For example, Directed Diffusion requires periodic interest refreshes [Intanagonwiwat et al. 2000], and in MultiHopRouter [Woo and Culler 2003], a routing protocol included in TinyOS [2006] for mote-based sensor networks, route control messages are periodically broadcast from each node in the network to estimate the routing cost and monitor link quality. In these cases the Siphon signature byte can ride for almost free, allowing for nearly zero-overhead virtual sink discovery. For applications that do not require periodic sink control messages, an independent signature byte application is invoked that broadcasts low rate (once per few minutes) virtual sink signaling messages from a physical sink, resulting in a small overhead. This overhead can be minimized through smart management at the sink, as discussed in Section 2.2. As shown in the following discussion, the embedded signature byte approach to virtual sink discovery is also used for controlling the visibility of the virtual sink to its neighbors.

This signature byte contains a VS-TTL (virtual sink TTL) field that specifies the scope (hop count) over which a virtual sink is advertised. A VS-TTL of l allows nodes up to l hops from a virtual sink to utilize Siphon's overload traffic management services. Clearly, a larger value of l allows more nodes to utilize a local virtual sink, but increasing l does not necessarily lead to better network performance. First, packets from nodes reached only by a large l have longer paths to the virtual sink and may not benefit from its use. Also, a broad virtual sink scope advertisement increases the chance of localized congestion around a

virtual sink (where each virtual sink potentially creates a mini-funneling effect similar to the original problem). On the other hand, a smaller value of l implies shorter redirect paths, improving delivery latency and energy consumption, but confines the benefit to fewer nodes. Section 3.5 investigates the trade-offs involved in determining the best initial value of l.

The handling of signature byte messages is different for virtual sink and non-virtual sink nodes in the network; the process flow for each case is outlined below. Note that physical sinks that do not have a secondary radio broadcast Siphon control packets (i.e., any nondata packets that include a Siphon signature byte) with the VS-TTL set to NULL; otherwise, physical sinks set VS-TTL to l. For virtual sink nodes, for any incoming nondata (control) packet, if a signature byte is embedded then identify the forwarder of this packet as the next Siphon hop. Further, if such a packet arrives via the secondary radio, then set the VS-TTL to l. Then, and also in the case where no signature byte is embedded, forward the packet through both radio interfaces. Note that virtual sinks receiving control packets containing the Siphon signature byte via their low power radios leave the VS-TTL as NULL and thus do not advertise their presence to the neighborhood. Such a virtual sink has no path to a physical sink via its secondary network and, thus, other nodes derive no extra benefit by forwarding packets through this node. However, the Siphon protocol definition allows for a graph of virtual sinks not connected to any dual-radio physical sink to carry traffic on its secondary network. We evaluate this scenario in Section 3.7 and discuss whether this yields any performance benefits. For nonvirtual sink nodes, for any incoming nondata (control) packet, if a signature byte is embedded and the VS-TTL is greater than 0, then identify the forwarder of this packet as a virtual sink neighbor and decrement the VS-TTL. Then, and also in the case where no signature byte is embedded, forward the packet.

Note that the existence of a *virtual sink neighbor* indicates a virtual sink is located in the neighborhood and can be reached through this specific neighbor. Through this procedure a sensor maintains a list of neighbors through which neighborhood virtual sinks are accessible. This list is maintained according to a soft state approach whereby if the Siphon signature byte is not periodically received from a virtual sink neighbor (e.g., virtual sink/node failure, radio dynamics) then that neighbor is removed from the virtual sink neighbor list. In our current implementation we set the soft state time out equal to twice the Siphon signaling period. Because of the small fraction of virtual sinks in the network, there is usually only one neighbor in the list. Therefore, the memory overhead for maintaining a virtual sink list is negligible. In many cases the overhead could be reduced to a single bit in each neighbor entry of the routing table.

While the initial setting of l is driven by the physical sink, if due to traffic load dynamics this initial value of l leads to persistent congestion at the primary radio interface of the virtual sink (measured using the techniques described in Section 2.2.1), the virtual sink can autonomously reduce the value of VS-TTL in the incoming control message by 1 before rebroadcasting. Since virtual sink neighbor lists are maintained using a soft state approach, this autonomous reduction of the VS-TTL by a virtual sink leads to reduced load once

the virtual sink neighbors outside of the reduced VS-TTL time out. Conversely, under a persistently sparse traffic regime the virtual sink can autonomously increase the value of VS-TTL in the incoming control message by 1 before rebroadcasting. While our current implementation of Siphon does use this soft state approach to virtual sink neighbor list maintenance, we defer reporting on dynamic scope adjustment to future work.

2.2 Congestion Detection

Accurate and efficient congestion detection plays an important role in the Siphon framework inasmuch as it indicates the proper time a sensor should attempt to utilize any virtual sinks it has discovered. We describe two techniques for congestion detection control and actuation of the virtual sink infrastructure: (i) *node-initiated congestion detection*; and (ii) *physical sink initiated "post-facto" congestion detection*. In what follows, we discuss these two techniques and their application in Siphon.

2.2.1 *Node-Initiated Congestion Detection.* CODA [Wan et al. 2003] describes a CSMA-based, energy-efficient congestion detection technique where wireless receivers use a combination of the present and past channel loading conditions, obtained through a low-cost sampling technique, and the current buffer occupancy to infer congestion. In Siphon, we adopt these mechanisms proposed in Wan et al. [2003] to determine the local congestion levels that a node is experiencing.

While the congestion detection techniques in CODA are CSMA-based or contention-based, the idea can be generalized to other MACs that are often used in sensor networks, including schedule-based [Clare et al. 1999; Rajendran et al. 2003] and hybrid-based MACs (e.g., S-MAC [Ye et al. 2002], T-MAC [Dam and Langendoen 2003]). For pure schedule-based MACs that attempt to guarantee collision-free communication, queue occupancy provides a good measure of the congestion level. For hybrid-based MACs such as T-MAC [Dam and Langendoen 2003], a good measure is a combination of the queue occupancy and the duty cycle length of the scheduled activity of a node.

However the congestion level is measured, when the local channel load approaches or exceeds the theoretical upper bound of the channel throughput [Wan et al. 2003], or when the buffer occupancy grows beyond a high water mark, a sensor node located within the visibility scope of a virtual sink will activate its redirect algorithm (see Section 2.3 for details) to divert designated traffic (e.g., data impulses, prioritized traffic, etc.) out of the neighborhood, utilizing the virtual sinks. In Li et al. [2001] show that an ideal wireless ad hoc multihop forwarding chain should be able to achieve 25% of the throughput that a single-hop transmission can achieve. This observation has important implications in dealing with our requirement of early congestion detection in Siphon. If the sensed channel load reaches a certain fraction (e.g., 25% [Li et al. 2001]) of the capacity, with high probability congestion will occur in a region located further downstream in the propagation funnel. To best counter the funneling effect, it is essential to redirect overload event traffic as early as is possible in the propagation funnel. However, in order not to diminish any possible aggregation

effort of correlated data in the network (aggregation is most effective deep in the funnel), it is beneficial to redirect traffic later in the funnel. To achieve a balance, it is best to redirect data at a location just before congestion is most likely to occur in the funnel. In Section 3.4 we verify this conjecture.

2.2.2 *Post-Facto Congestion Detection.* As an alternative approach to the node initiated congestion detection discussed in the previous section, we consider the post-facto activation of the virtual sink infrastructure via congestion inference at a physical sink. The physical sink, as a point of data collection in the funnel, can do smart monitoring of the event data quality and the measured application fidelity [Tilak et al. 2002], and initiate virtual sink signaling only when the measured application fidelity degrades below a certain threshold. In this approach, the siphoning service is enabled only after congestion or fidelity degradation is measured in the primary low-power radio network. As such, the approach has limited capabilities dealing with transient congestion deep in the network, but may be adequate when congestion occurs closer to the physical sink. This technique has the advantage of not requiring underlying congestion detection support at each node. To propagate the signal in a timely manner from the physical sink, a control message is broadcast through its noncongested secondary radio network (a connected secondary network is required). Because the traffic siphoning in the post-facto approach is based on the perceived performance measured at the physical sink but not the congestion levels experienced in the network, we conjecture that it also has the advantage of avoiding premature traffic siphoning especially when network-wide aggregation [He et al. 2004] is used. In Section 4, we examine the effectiveness of the post-facto congestion approach in a sensor network testbed.

2.3 Traffic Redirection

Traffic redirection in Siphon is enabled by the use of one *redirection bit* in the network layer header. We consider two approaches in setting the redirection bit: (i) *on-demand redirection*, in which the redirection bit is set only when congestion is detected; and (ii) *always-on redirection*, in which the redirection bit is always set. We discuss the tradeoffs of these two approaches in Section 3.6. The basic redirection mechanism is as follows. A sensor that receives a packet with the redirection bit set, forwards the packet to its virtual sink neighbor, a process through which the redirected packet would eventually reach a virtual sink. If the redirection bit is not set then routing follows the paths determined by the underlying data dissemination/routing protocol.

When a virtual sink receives a redirected packet, it forwards the packets to the neighbor from which it most recently received a control message embedding the signature byte. As discussed in Section 2.1, such control packets can arrive either through a virtual sink's primary or secondary radio interface. In the best case, all virtual sinks are connected to a physical sink via the secondary network overlay and all physical sink-bound packets are routed through the virtual sink and forwarded on a fast track all the way to the physical sink. When the secondary network is partitioned, the last virtual sink (closest to the sink) in the secondary network fragment must direct all sink-bound packets

back onto the primary network, specifically to the sensor it has identified in the discovery phase. From here, packets are again routed to the physical sink according to the default routing paths.

Recent experimental studies [Zhao and Govindan 2003; Woo and Culler 2003] show that sensor networks using low-power radios often suffer from highly variant wireless link quality that is both time and location dependent. To ensure that traffic siphoning through the virtual sink infrastructure does not degrade the network's primary packet forwarding service, only neighbors with good link quality are utilized to redirect packets to a virtual sink. Many routing protocols (e.g., MultiHopRouter [Woo and Culler 2003]) maintain a neighbor table that includes a continuously updated link quality estimation for a selected set of neighbors. When a sensor located within the visibility scope of a virtual sink detects congestion while forwarding event packets, it makes a decision to redirect a specific type of data packet based on local policy.

As a general policy rule for traffic redirection in Siphon, the link to the virtual sink neighbor must have a link quality estimate that is within 15% (lower bound) of the link estimate of the currently chosen next hop. If the link quality estimate to the virtual sink neighbor is worse than this bound (e.g., virtual sink/node failure, fading, etc.) the virtual sink neighbor should not be utilized. If the redirect policy parameters are met, the congested sensor marks the redirection bit in the routing header of the data packet being forwarded and redirects it to a virtual sink neighbor selected from its local list. Conformance with an appropriate policy allows use of the virtual sink infrastructure to improve application data fidelity by bypassing funnel congestion around the primary physical sink, without potentially incurring an unacceptable level of packet loss through use of low quality links to the local virtual sink.

As mentioned in Section 2.1, the virtual sink advertisement scope (VS-TTL) is engineered to offer a virtual sink path to the largest number of nodes while avoiding the creation of local congestion around the virtual sink. Section 3.5 investigates the tradeoffs involved in determining the optimal value of this advertisement scope under different conditions. However, due to load generation dynamics, congestion can still happen in the locality of a virtual sink. In the case of transient congestion, existing techniques such as CODA's open-loop control [Wan et al. 2003] already running in the network can provide relief. For persistent congestion, dynamic adjustment of the virtual sink advertisement scope (including disabling advertisements entirely), as discussed in Section 2.1, is used.

Virtual sinks offer shortcuts and possibly higher-bandwidth pipes for data delivery in sensor networks. Traffic siphoning through virtual sinks may subtly impact the routing protocols operating in the primary and secondary network only if the routing metrics used are sensitive to enhanced service characteristics, such as the delay or loss associated with the data delivery paths in the network. For example, data-centric dissemination protocols such as Directed Diffusion [Intanagonwiwat et al. 2000] and variants of DSDV-like routing protocols, are capable of choosing empirically good paths that dynamically adapt to changing network conditions. These protocols are therefore delay-sensitive and

their routing decisions could be impacted by traffic siphoning in a subtle way.[1] In Section 3.2 we discuss these interactions and propose a simple method to deal seamlessly with such behavior. As examples, we show how Siphon interworks with Directed Diffusion and Surge, in Sections 3 and 4.5, respectively.

2.4 Virtual Sink Placement

The benefit derived from Siphon is strongly dependent on where the virtual sinks are located in relation to where congestion most often arises in the network, and the ability of the underlying primary radio network to redirect traffic to the nearest virtual sink. However, in general, the occurence of congestion is not only driven by packet generation rate but is also physical and routing topology dependent. This fact makes it difficult to describe general heuristics for virtual sink emplacement in nonuniform topologies.

Given a particular deployment topology, it is possible to identify the most likely positions for virtual sinks through a two-step prediction and correction process. First, predict likely congestion hotspots by identifying routing bottlenecks in the network (e.g., the funnel region of the physical sink). This can often be done a priori from a simple visual inspection of the physical topology, or in situ through use of a network management tool [Tolle and Culler 2005; Deb et al. 2003]. If channel load measurements (see Section 2.2.1) are exposed to the network management system, congestion hotspots can be identified with a higher level of accuracy. After the initial virtual sink deployment, use of online monitoring, either in situ using a network monitoring system or using a post facto approach (see Section 2.2.2) to refine virtual sink placement. Based on our experimental testbed experience, in the case where there are more persistent congestion hotspots than available virtual sinks, it is preferrable to place a virtual sink near each in a subset of the identified hotspots rather than placing virtual sinks between hotspots in the middle of the network in an attempt to cover all hotspots. This is because virtual sink service is confined to the advertisement scope area, and the larger scope required to reach nonadjacent hotspots would require longer multihop paths to redirect packets to the virtual sink implying a higher packet loss probability. Also, a broad virtual sink scope advertisement increases the chance of localized congestion around a virtual sink.

In Section 3 we provide average case analysis and simulation based on randomized virtual sink placements in random topologies; in Section 4.5 we provide results from a mote testbed in a grid topology with emplaced Stargate virtual sinks.

2.5 Congestion in the Secondary Network

The traffic siphoning service is complementary to the first generation congestion control schemes such as CODA [Wan et al. 2003] and Fusion

[1]Only protocols that base their routing decisions on the actual data delivery service perceived at the receiver are potentially impacted. Other protocols, that base their routing decisions on fixed routing metrics, such as shortest path routing, geographical routing [Navas and Imielinski 1997] or the routing on a curve [Nath and Niculescu 2003] approach are not affected.

[Hull et al. 2004], and as such can be run in parallel with these techniques on the primary and secondary networks. When the secondary network is also overloaded, traffic redirection through virtual sinks offers little benefit. Therefore, a virtual sink always monitors its own congestion levels on both primary and secondary radio channels and does not advertise its existence when either one of its radio networks is consistently overloaded. For the IEEE 802.11 radio (which we use in our experimentation), Murty [2004] propose an algorithm to calculate the normalized collision-induced bit error rate as part of their scheme to predict congestion and dynamically adjust the MAC parameters for throughput optimization. We use this technique as a reliable scheme to detect congestion on Siphon's secondary IEEE 802.11 network. This forces a virtual sink to refrain from offering overload traffic management service or to reduce its scope of service according to the level of detected congestion.

When both primary and secondary networks are overloaded, the congestion levels on both networks will eventually rise beyond certain thresholds. In that case, CODA's backpressure mechanism (or the similar mechanism in Fusion) will be triggered, (i.e., the system falls back to the traditional schemes that rate-control the source and forwarding nodes to alleviate congestion). In general, virtual sinks are less likely to be congested since they can send and receive packets at the same time through the two different radios, in channels with different characteristics (fading, throughput, delay, etc.).

3. SIMULATION EVALUATION

We use packet-level simulation to obtain preliminary performance evaluation results for Siphon, and study scaling issues that are not easily evaluated in our experimental testbed, as discussed in Section 4. We also discuss the implications of our results on the design choices that shape Siphon.

3.1 Simulation Environment

We implement Siphon as an extension to the ns-2 simulator in its simplest instantiation. First, to model a virtual sink node we add support for a second long-range radio interface that has a transmission range of $250m$. The primary low-power radio used in our simulations is configured to have a $40m$ transmission range to model a typical sensor node. We use Directed Diffusion [Intanagonwiwat et al. 2000] as the routing core in the simulations, which allows the simulations to shed light on Siphon's interaction with a realistic data routing model where congestion can occur.

Our simulations use the 2 Mbps IEEE 802.11 MAC provided in ns-2 with some modifications. We add code to the MAC to measure channel loading using the epoch parameters ($N = 3, E = 200ms, \alpha = 0.5$), as defined in Wan et al. [2003]. The MAC of a node sets a congestion flag in the routing agent when the measured channel load exceeds a threshold of 70% [Wan et al. 2003]. To perform early congestion detection, as discussed in Section 2.2, it is beneficial to set the congestion flag at a lower threshold. We verify this in Section 3.4.

In all our experiments, we use random topologies with different network sizes. For each network size, our results are averaged over five different

generated topologies and each value is reported with its corresponding 95% confidence interval. In most of our simulations, we use a fixed workload that consists of six sources and two physical sinks. A sink subscribes to three data types corresponding to three different sources [Intanagonwiwat et al. 2000]. Note, however, that the network dynamics in the simulations are nondeterministic because each sink subscribes at a random time to a set of sources that is randomly chosen over different simulation sessions. Thus, the congestion periods and areas are nondeterministic due to Directed Diffusion's ability to choose empirically good paths that are dynamically adapted to the network conditions.

To model impulse type data traffic that is generated from an event epicenter, all sources are located in a neighborhood of a node randomly selected from nodes in the network. Sinks are uniformly scattered across the sensor field.

3.2 Delay Device and Directed Diffusion

Next, we describe a scheme to seamlessly integrate Siphon redirection with data-centric dissemination protocols using Directed Diffusion as an example. In Directed Diffusion, the sources initially generate low rate data packets that are marked *exploratory* and are disseminated through multiple paths toward the physical sink. Based on the measured delivery performance, the sink later reinforces one or more empirically good paths capable of delivering high quality data traffic, (i.e., with lowest latency and highest fidelity delivery). Subsequently, the sources generate higher rate data packets, no longer marked exploratory, which are transported along the reinforced paths.

As mentioned in Section 2.3, when routing protocols are delay-sensitive, the enhanced service offered by Siphon can affect routing decisions. This is certainly the case for Directed Diffusion, which is used as the routing protocol for both primary and secondary networks in our simulations. Specifically, exploratory data packets traversing the low-delay paths provided by the virtual sink secondary network will almost certainly reach the physical sink before packets passing through the primary network. As a result, paths using the virtual sink secondary network will always be reinforced, regardless of the congestion state of the primary network when using delay sensitive protocols such as Directed Diffusion. In Section 3.6, we discuss the merits of such *always-on* operation of the secondary network. In general, however, a mechanism that allows for the conditional (i.e., on-demand) usage of the low-delay virtual sink paths is required.

To this end, we implement a *delay device* on each virtual sink that operates on the secondary radio interface and is activated whenever a virtual sink forwards an exploratory data packet through the long-range radio. The device delays the forwarding of exploratory data packets via the secondary radio by D seconds. D should be large enough that these exploratory data packets will not be the first to be delivered to the physical sink, instead allowing packets to reach the physical sink via the primary radio network first. For example, in our simulations we use the maximum round-trip delay between the two nodes furthest apart from each other in the network as the value of D for the delay device. In this

manner, paths on the primary network (instead of the secondary network via virtual sinks) are reinforced by Directed Diffusion, and this situation persists while the network is in a noncongested state.

When a node within the visibility scope of a virtual sink detects congestion while forwarding data packets, it takes action such that the virtual sink secondary network becomes more attractive to Directed Diffusion, allowing traffic to be siphoned from the congested region through the virtual sinks. Specifically, such a node selectively duplicates a data packet (e.g., one in every fifty data packets), marks it exploratory (using the Directed Diffusion exploratory bit) and sets the redirection bit (the same bit as described in Section 2.3), and forwards to its virtual sink neighbors. Note that the original packet is still forwarded along the existing routing paths during this period. A virtual sink receiving an exploratory data message with the redirection bit set will disable the delay device and forward the message immediately through both interfaces (assuming they have matching gradient entries). Without the delay added by the delay device, a dissemination path over the virtual sink secondary network is likely to be reinforced by the sink. Subsequently, high rate data will be redirected over the secondary network, until the congestion ends. At that point, the node that had originally signaled the congestion stops setting the redirection bit in data packets it forwards through the virtual sink. The virtual sink will reinstitute the delay device, and Directed Diffusion will ultimately again reinforce the best path(s) on the primary network.

3.3 Performance Metrics

We define the following metrics to analyze the performance of Siphon on sensing applications.

—*Energy Tax = (Tot.*[2] *pkts dropped in the network)/(Tot. pkts rcvd at the physical sink)*. Since packet transmission/reception consumes the main portion of the energy of a node, the average number of wasted packets per received packet directly indicates the energy saving aspect of Siphon.

—*Energy Tax Savings = ((Avg E.Tax w/o Siphon)—(Avg E.Tax w/ Siphon))/(Avg E.Tax w/o Siphon)*. This metric indicates the average Energy Tax improvement or degradation from using Siphon.

—*Fidelity Ratio = (Pkts rcvd at the physical sink w/ Siphon)/(Pkts rcvd at the physical sink w/o Siphon)*. The ratio indicates the average fidelity improvement or degradation from using Siphon.

—*Residual Energy = (Remaining energy)/(Initial energy)*. We use the ns-2 energy model for IEEE 802.11 network to measure the remaining energy of each node at the end of a simulation. The residual energy distribution allows us to examine the load balancing feature of Siphon and to estimate the effective network lifetime.

[2]Dropped packets include the MAC signaling (e.g., RTS/CTS/ACK and ARP), event data, and Diffusion messaging packets.

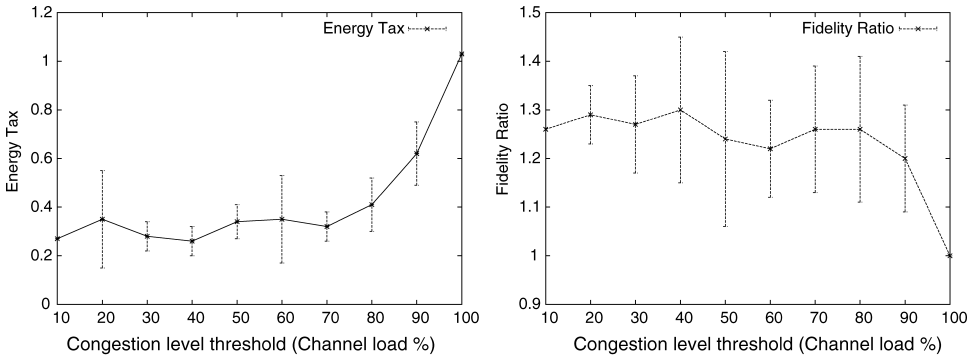

Fig. 2. Early Congestion Detection. Energy Tax and Fidelity Radio performance for different congestion level thresholds that can avoid congestion down the funnel.

We use these metrics to evaluate and quantify the benefits of using Siphon under different scenarios and configurations in the following sections.

3.4 Early Congestion Detection

Using fidelity and energy tax performance as a guide, we first search for a congestion (channel load) threshold that will trigger traffic siphoning to best avoid congestion in the funnel. We simulate a network of 30 nodes, where 2 nodes are randomly selected as virtual sinks, one of which is also selected as the physical sink. There is only one virtual sink within the network that can be utilized to redirect data traffic. Six nodes are randomly selected as sources. Each source generates 15 pkts/sec sent toward the physical sink starting at a random time distributed uniformly from 10 to 15 seconds into the simulation, and runs for 100 more seconds. In the simulations, we vary the congestion threshold at which we should start redirecting data traffic to a nearby virtual sink. Figure 2 plots both fidelity and energy tax against different congestion level thresholds.

In the simulation, we strategically place the virtual sink at a location within a few hops of the propagation funnel toward the physical sink. Figure 2 shows that as long as the virtual sink is utilized for traffic siphoning, the data fidelity is improved regardless of the congestion level threshold. However, the energy tax of the network rises quickly when the threshold is set higher than 80%. In our simulations, we observe that a channel utilization of 80% is where the channel saturates and suffers from frequent collisions between neighboring nodes. Note that this is also the threshold chosen to trigger CODA's open-loop backpressure scheme in Wan et al. [2003]. This indicates that when the threshold is set too high it is too late to divert traffic at a location that is deep in the funnel. Considering that Siphon is a complementary scheme (to CODA) that prevents congestion by diverting traffic earlier in the funnel, Figure 2 indicates that a threshold that is slightly lower than the channel saturation level would be appropriate. For example, 70% is appropriate in this simulated network since the energy tax is only slightly higher than that incurred at the lower thresholds.

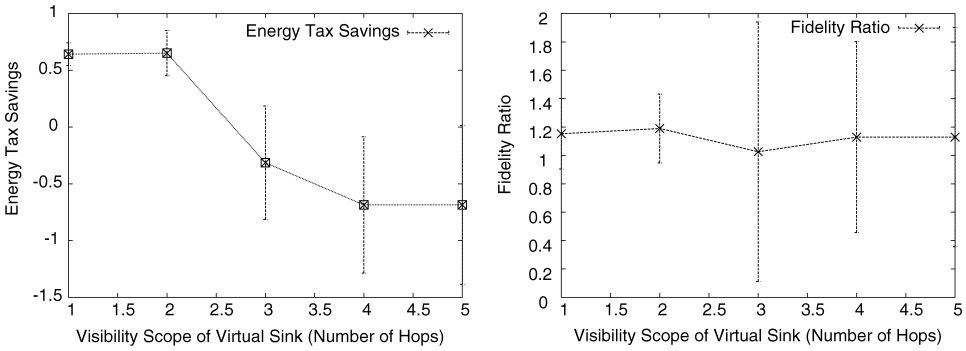

Fig. 3. The impact of the visibility scope of a virtual sink for a network of 30 nodes.

Notice that utilizing a virtual sink at a lower threshold means its energy is more quickly drained. While a high buffer occupancy can also serve as a good indicator for congestion, we observe [Wan et al. 2003] that in our simulator it grows at a much slower rate than the channel load. In Section 4.2 we investigate an appropriate buffer occupancy level threshold that best predicts congestion in our sensor testbed.

3.5 Virtual Sink's Visibility Scope Impact

In what follows, we investigate the visibility scope of a virtual sink. We vary the scope l from 1 to 5 and measure the fidelity ratio as well as the average energy tax. In Figure 3, the energy tax is normalized such that it represents the energy tax savings when using Siphon.

Figure 3 shows that for all values of l, the average fidelity ratio is larger than 1 (despite the high variability when l is larger than 2), indicating that fidelity can be improved whenever a virtual sink is utilized. However, the energy tax savings decreases when l is larger than 2, and drops rapidly below zero, indicating that the nodes actually consume more energy when they utilize and redirect data traffic to a virtual sink that is more than two hops away. Through careful examination of the details of our simulation, we observe that when l is larger than 2, it often creates local congestion around the virtual sink (i.e., a mini-funneling effect) as more nodes within the funnel are trying to redirect data through the same virtual sink. This causes frequent collisions and therefore more packet drops and more retransmissions. Figure 3 shows that when l is 2, both the fidelity gain (20%) and energy tax savings (60%) are the highest, and have smaller confidence intervals, indicating that an l equal to 2 is an appropriate scope.

3.6 Always-On versus On-Demand Virtual Sinks

An always-on virtual sink continuously powers up the secondary radio to help forward nearby data traffic, regardless of the congestion conditions in the neighborhood. This can help to enhance data delivery service in the field at the expense of consuming more of its own energy. On the other hand, an on-demand

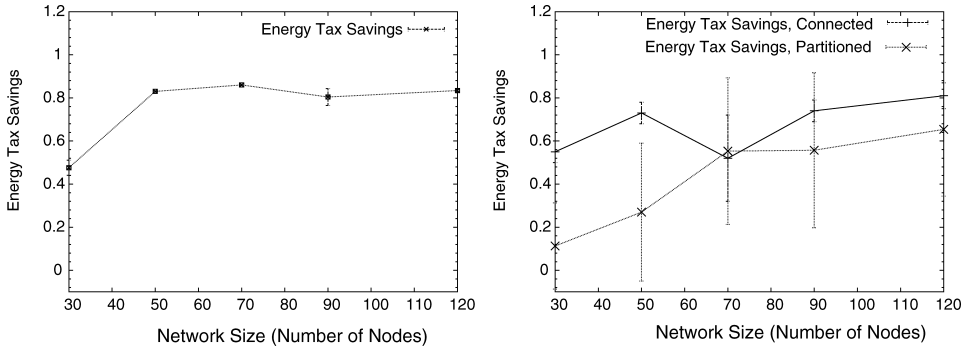

Fig. 4. (a) Energy Tax performance in a network with always-on virtual sinks. (b) Energy Tax performance in a network where virtual sinks are activated only when congestion is detected, (i.e., on-demand).

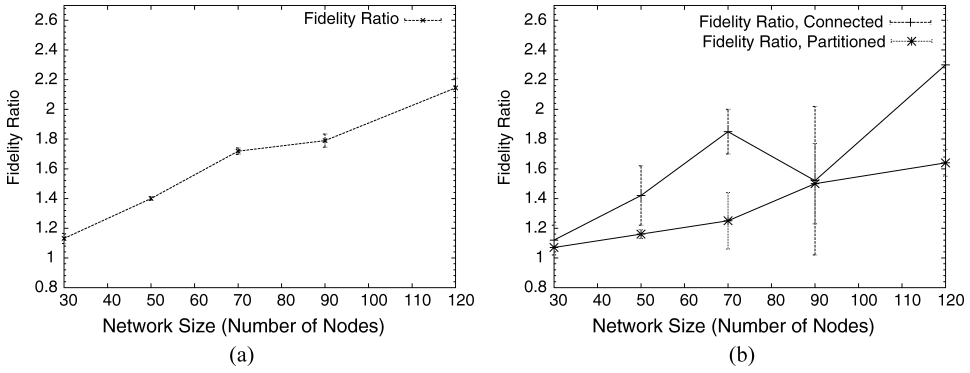

Fig. 5. (a) Fidelity performance in a network with always-on virtual sinks. (b) Fidelity performance in a network where virtual sinks are activated only when congestion is detected, (i.e., on-demand).

virtual sink will not power up its long-range radio unless its visibility scope overlaps a congestion region. In what follows, we specifically investigate the trade-off of these two approaches. To model an always-on virtual sink, we simply disable the delay device on the node, as discussed in Section 3.2.

Figures 4(a), 4(b), 5(a), and 5(b) present the fidelity ratio and energy tax saving performance of these two approaches in a set of networks of different sizes. Six sources and two physical sinks form two propagation funnels in the network. In each simulation, 5% of the nodes are randomly selected to be the virtual sinks. In this scenario, the 5% virtual sinks are uniformly distributed across the field and form a connected secondary network over long range radios. Figures 4(b) and 5(b) also include another set of plots that present the scenario when the virtual sinks can not form a connected secondary network, as discussed in Section 3.7.

Always-on virtual sinks are utilized whenever they are able to deliver data event to a physical sink with lower delay and higher fidelity. Figures 4 and 5

show that Siphon is able to obtain greater fidelity gain in a larger network, although the energy gain does not follow the same trend. The fidelity gain increases almost linearly with increasing network size, while the gain in energy tax levels off after a network size of 50 nodes. This indicates that when the number of nodes in the network increases, the number of dropped packets increases almost linearly because of a longer propagation path and more intense funneling effect. But with Siphon, the virtual sinks are able to siphon off events to maintain the fidelity level regardless of the linearly increasing number of packet drops. Without Siphon, the packet delivery service degrades linearly while the number of packets dropped (wasted) increases rapidly. This indicates that the energy tax of Siphon degrades much more slowly than the vanilla Directed Diffusion. When the network size increases, Siphon can continue to obtain larger fidelity gains, although the energy benefit obtained does not keep pace.

Figures 4(b) and 5(b) closely agree with Figures 4 and 5, respectively, except with a much higher degree of variability (indicated by the error bars that represent the 95% confidence intervals). This indicates that instead of utilizing virtual sinks in an always-on fashion, possibly exhausting the energy of the virtual sink (recall that the virtual sinks are not line-powered), on-demand virtual sinks that power up the secondary long range radio only in times of congestion and can achieve almost as good energy savings and fidelity improvement may be preferred. Figures 4(b) and 5(b) also shows the efficacy of our congestion detection scheme since it enables the nodes within the visibility scope of a virtual sink to correctly detect congestion and utilize the nearby virtual sink. However, the on-demand nature of this approach increases the dynamics and introduces more disturbance into the network, hence the high degree of variability in the plot. This result clearly illustrates the tradeoff between data delivery, service stability, and energy consumption of the virtual sinks.

One further issue to consider when using the virtual sink overlay on demand is the availability of the next hop on the mesh when traffic is to be redirected via the overlay mesh to the physical sink. Since one option when using virtual sinks on-demand is to activate the secondary radio only when redirected traffic is received via the primary radio (a local condition), coordination among the members of the mesh is necessary to guarantee that traffic transmitted along the secondary network mesh will reach the physical sink. The problem may be addressed by provisioning some local buffering at each virtual sink and using any of a number of scheduled MACs (e.g., Ye et al. [2002]; Dam and Langendoen [2003]). Clearly, the choices made in solving this problem have an impact on the delay performance of packet delivery over the secondary radio mesh, and should thus take into consideration the requirements of the application. Note that for small networks where the secondary network is one hop to the physical sink this issue disappears, assuming the secondary radio of the physical sink is always on. Alternatively, we can change the definition of "on-demand" to mean only *transmitting* via the secondary radio when redirected traffic is received via the primary network, but keeping the secondary radio powered at all times. Clearly, this will guarantee availability of the next hop on the overlay mesh, but

implies a high idle listening cost. We comment further on this energy tradeoff in Section 4.6.

3.7 Partitioned Secondary Network

If only a small number of virtual sinks are deployed in the network or if the physical sink does not support a secondary radio, then the virtual sinks may not form a connected network and the primary short-range channel must be used to deliver packets between virtual sinks. To model this, we move virtual sink functionality from one of the physical sinks to another node. This action conserves the number of virtual sinks, while partitioning the secondary network. Each of the simulations described in previous section is repeated. The result is presented in Figures 4(b) and 5(b), which show that both the fidelity and energy tax gains are much smaller (and have higher variability) than their connected network counterparts, especially for energy tax performance in smaller networks. For example, in a 30-node network, the energy tax sometimes is even higher than without siphoning, indicated by the error bars of energy tax savings that include negative values in Figure 4(b). We observe that in smaller networks, the paths that connect virtual sinks through the primary channel often coincide with the original propagation funnels toward the physical sinks. Although this could improve load balancing by diverting traffic through virtual sinks and their surrounding neighbors, it does not eliminate network bottlenecks caused by the funneling effect. This result suggests that a connected secondary network is required to reap a consistent benefit from traffic siphoning for the purpose of congestion avoidance and overload management.

3.8 Virtual Sink Density Impact

In Section 3.5 we demonstrate that a visibility scope of 2 hops is most appropriate. Section 3.7 describes the two disadvantages of having a partitioned secondary network. These results influence the distribution of virtual sinks in a network. Kumar and Xue [Xue and Kumar 2004] proved an asymptotic result for full connectivity within a randomly distributed wireless network; that is, in a wireless network consisting of n nodes, the network is asymptotically connected if each node connects to greater than $5.1774 \log n$ nearest neighbors. This result provides an analytical foundation for connected ad hoc network deployment. Consider a radio communication range of r, using the uniform independent and identically distributed node placement assumption in Xue and Kumar [2004], we can derive that a randomly distributed network of N nodes can fully cover an area and guarantee full connectivity therein if it fulfills the constraint:

$$Area = \frac{N * \pi * r^2}{5.1774 \log N}, \tag{1}$$

Meanwhile assuming a visibility scope of l hops, we can calculate the number of virtual sinks required in this network deployment as:

$$N_{v-sink} = \frac{Area}{\pi (lr)^2} = \frac{N}{5.1774 \log N * l^2}. \tag{2}$$

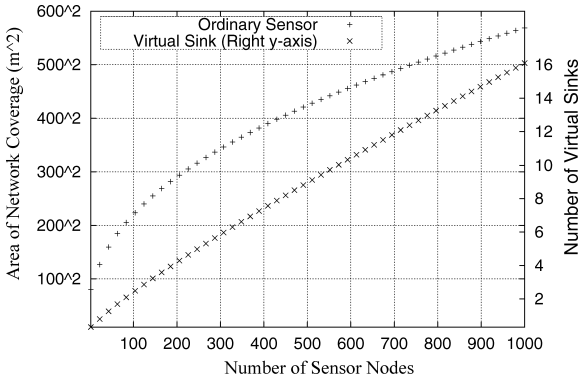

Fig. 6. Number of sensor nodes required to ensure connectivity in the corresponding areas of network coverage as well as the number of virtual sinks (right vertical axis) required to ensure performance improvements.

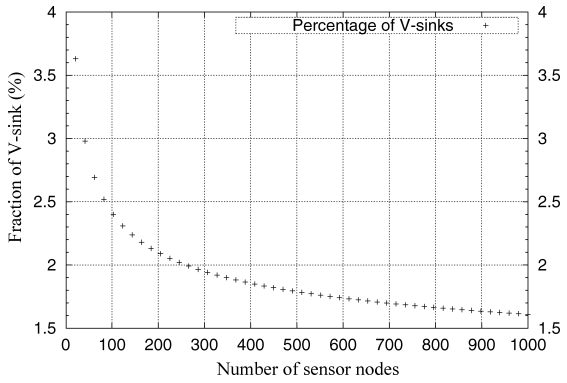

Fig. 7. Fraction of Virtual Sinks needed to assure improved network performance. As the network size increases, the relative cost decreases, e.g., only 1.6% of nodes needed to be virtual sinks for a 1000-node network.

Using Equations (1) and (2), one can determine the number of sensors and virtual sinks required to populate a designated area of interest. Figure 6 plots the above expressions numerically against number of sensors N. The radio communication range of a sensor is $r = 40m$, while the long-range radio communication range of a virtual sink is $250m$ and the visibility scope l is 2 hops. With this specific setup, according to Figure 6, an area of 600^2m^2 would require 1000 randomly distributed sensors to ensure network connectivity, while 16 virtual sinks are enough to guarantee performance improvement from siphoning. Furthermore, from Equation (2) the fraction of virtual sinks needed to assure improved network performance decreases logarithmically in N (plotted numerically in Figure 7).

On the other hand, the connected secondary network requirement imposes a lower bound on the ratio between the transmission range of the long-range radio and the low-power radio. In Figure 8, we consider the network area covered by the secondary long-range radio of virtual sinks with different transmission

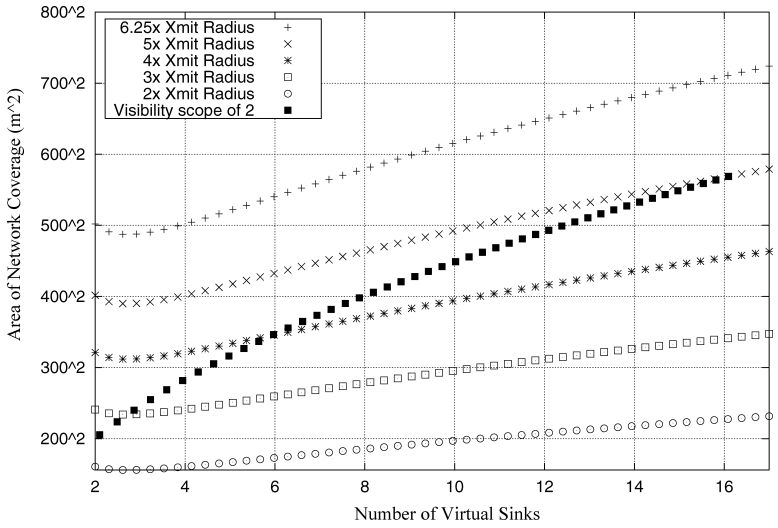

Fig. 8. Requirement for a connected secondary network. The transmission range of the long-range radio is expressed as multiples of the transmission radius of the low-power radio. The visibility scope requirement that assure both energy tax and fidelity improvements is plotted as filled square in the figure.

range ratios, and plot them against the number of virtual sinks in the network. We observe that for a visibility scope of $l = 2$, when the transmission ratio > 5, the network coverage of the required number of virtual sinks is always smaller than the network coverage of the same number of virtual sinks needed to ensure connectivity in secondary network. In other words, for a specific area, the number of virtual sinks required for a visibility scope of 2 is always larger than that required to ensure full connectivity in the same area. This indicates that if the transmission range of a virtual sink's long-range radio is at least 5 times that of its low-power radio, then the number of virtual sinks required for visibility scope of 2 also guarantees a connected secondary network. Together, Figure 6 and Equation (2) offer an appropriate roadmap for network deployment.

3.9 Load Balancing Feature

In what follows, we study the load balancing feature of Siphon in terms of its energy impact on the network. We simulate a moderate-size network of 70 nodes. Three virtual sinks are scattered at random locations in the network. One of these virtual sinks is selected as the physical sink, subscribing to six randomly designated sources that generate data at 10 packet per second. We measure the residual energy of each node at the end of each simulation and plot the average complementary cumulative distribution frequency (CDF) of the residual energy distribution of the network in Figure 9.

Figure 9 shows that the minimum residual energy of the network increases from 67% to 72%, meaning each node has a residual energy larger than 72% of its initial energy capacity. However, the plot also shows that the probability of any nodes having residual energy larger than 86% is 0%, while without

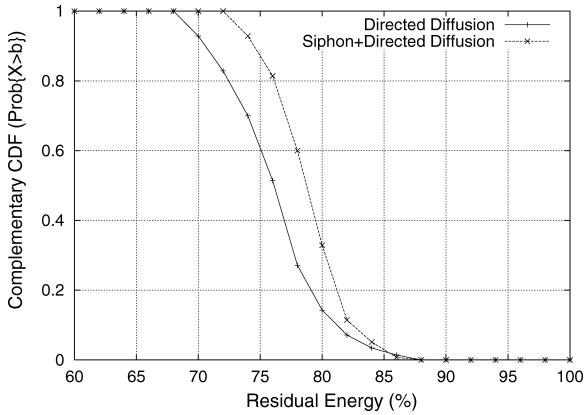

Fig. 9. Energy distribution (CDF) of a 70-node network with 3 virtual sinks scattered randomly across the network. With Siphon's load balancing feature more nodes share the energy load. Therefore, fewer nodes have residual energy larger than 85%, but more nodes have larger residual energy (e.g., the percentage of nodes having residual energy larger than 75% increases from 60% to 85%), effectively increasing the lifetime of the network.

Siphon the probability is slightly higher. The maximum residual energy among the nodes decreases because more nodes are involved in forwarding packets with Siphon. More nodes share the energy consumption, indicating the load balancing feature of Siphon. Note that even without Siphon there is no node possessing residual energy of more than 88%; all nodes at least spend some energy. This is because of the periodic interest flooding requirement of Directed Diffusion. In summary, Siphon can balance the load in the network so that more nodes have higher residual energy as more nodes share the energy load, effectively increasing the operational lifetime of the network.

4. TESTBED EVALUATION

In this section, we discuss the implementation of Siphon on a real sensor network using [TinyOS 2006] on Mica2 motes [Hill et al. 2000] and the Stargate platform [Stargate 2006]. We equip our Stargates with IEEE 802.11b PCMCIA cards, enabling each Stargate to be a virtual sink that talks to both the long-range IEEE 802.11b network as well as the short-range CC1000 radio network formed by Mica2 motes. We report evaluation results, including appropriate threshold values for congestion levels that should trigger the traffic redirection, and an evaluation of Siphon with a generic data dissemination application as compared to CODA [Wan et al. 2003]. We evaluate a "post-facto" approach that activates the virtual sinks only after congestion has occurred and impacted the application's fidelity, as discussed in Section 2.1. Finally, we report on the performance of Siphon in a network running Surge, a commonly used application included in the TinyOS-1.1.0 distribution [TinyOS 2006].

4.1 Mote Testbed Configuration

Our testbed comprises 48 Mica2 motes arranged in a 6×8 grid. In order to construct a fairly strict and dense multihop radio environment for our experiments,

we first ran calibration experiments to determine appropriate node spacing and transmission power. First, a centrally located node A in the grid is chosen as a transmitter at the maximum rate. Reception rates are then measured at the nodes one and two hops to the North, Northeast, East, Southeast, South, Southwest, West, and Northwest of A, for a given transmission power. The same procedure is then repeated but reversing the roles of receiver and transmitter (still only one transmitter at a time) so that A is the receiver and nodes in the cardinal and ordinal positions are transmitters. Five trials of 100 packet transmissions are done for each (transmission power, spacing) pair. Systematically stepping through transmit power settings and node spacings we identify values such that one-hop neighbors achieve $> 80\%$ delivery, while two-hop neighbors achieve $< 20\%$ delivery in our grid. For repeatability, the exact positions of the motes are maintained through out all experiments so that the multipath environment is fairly constant.

Even though the Mica2 voltage regulation circuitry is much more effective at providing constant power to the radio across a broad range of battery voltages, as compared to earlier generations of the Berkeley mote architecture (e.g., Rene2), we anecdotally experience best performance in this regard when the battery voltage is above 2.5 volts. Therefore, during the course of all our experiments, the battery voltage is maintained above 2.6 volts on all motes at all times.

Sections 4.2, 4.3, and 4.4 evaluate Siphon performance in a focused funnel environment. Thirty of the 48 motes are activated for the experiments in these sections. The topology resembles a hub-and-spoke arrangement, though the "hub" is just a normal mote with no special requirements or capabilities. Section 4.5 evaluates Siphon performance using the entire 6×8 grid, with the physical sink node, viewing the grid outline as a rectangle, located in the middle of a long edge. The physical sink also has virtual sink functionality and communicates over IEEE 802.11b directly with the other active virtual sinks in the various experimental scenarios.

4.2 Traffic Redirection Decision

Following the rationale suggested in Section 3.4 from the simulation results, a channel load threshold that is slightly lower than the channel saturation level would be appropriate to trigger traffic siphoning to avoid congestion. We therefore run an experiment to measure the channel load versus offered load to guide the choice of an appropriate redirection threshold in our Mica2 test bed. Given that in our testbed the channel saturates at about 36% (Figure 10), a redirection threshold that is between 20 and 30% should be appropriate. Note that the saturation level in the test bed is markedly lower than in the simulated network (Section 3.4), because the simulated network uses the IEEE 802.11 MAC/PHY and does not suffer from real world radio phenomena.

Queue management is often used in traditional data networks for congestion detection, that is, congestion is signalled when a node's buffer occupancy grows beyond a high water mark level. However, as discussed previously in Section 3.4, we observe through simulations that the channel load provides

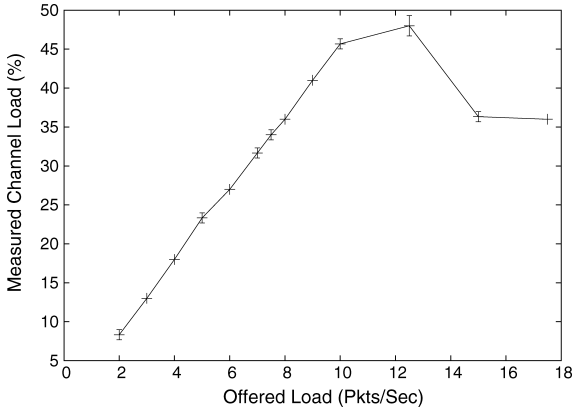

Fig. 10. An appropriate choice for the traffic redirection threshold is a value slightly below the congestion point of the network.

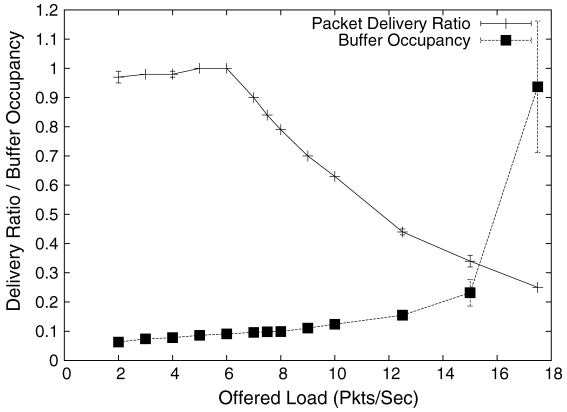

Fig. 11. Queueing performance and buffer occupancy threshold for congestion avoidance.

a much faster and more reliable indication of network congestion than buffer occupancy. While the same observation holds true for our mote-based sensor testbed, we observe one exception when the time-varying channel suffers from occasional deep fades for an extended time period. During this period, while the measured channel load is low, few packets can be delivered between forwarding nodes. Therefore, the queue of the sending node grows quickly (when link-ARQ is used) and eventually overflows and starts dropping packets. Based on this observation, it is beneficial to determine an appropriate buffer occupancy level that can reliably indicate congestion in addition to channel load indication.

In our testbed, we generate data packets at different rates and measure the average queue size of the nodes in a small neighborhood that share the wireless medium. Figure 11 plots the measured normalized average buffer occupancy against the offered load. We also plot the packet delivery ratio between neighboring nodes in the same figure. We observe that the buffer occupancy is small

($\leq$ 10%) when the channel quality is excellent and the packet delivery ratio is high. On the other hand, when the buffer occupancy $>$ 10%, the packet delivery ratio falls below 80% and continues to drop quickly, signifying a congested state. The offered load at which the buffer occupancy touches 10% also coincides with the first point the measured channel load reaches the saturation value of 36% (see Figure 10). Based on this result, we set the buffer occupancy threshold to 10% in our test bed for all experiments discussed in next section. Note that by the offered load at which the buffer occupancy saturates, the data packet delivery ratio has already sagged below 30%, indicating severe congestion. This result illustrates the insufficiency of a high watermark buffer occupancy as a stand-alone indicator of congestion.

4.3 Generic Data Dissemination Application

In what follows, we evaluate Siphon using a realistic data dissemination application and compare the result with CODA's open loop control. CODA's [Wan et al. 2003] open-loop control function supports priority forwarding of packets from a list of predefined data types. This includes a channel load measurement MAC module as described in Wan et al. [2003] on Mica2. To support Siphon, we use Stargates as virtual sinks and implement the traffic redirecting function as well as the virtual sink visibility scope control function on both Mica2 and Stargate platforms. In the following experiments we use a static visibility scope of 1 to demonstrate the minimum benefit that can be obtained by the use of virtual sinks; we defer reporting on dynamic scope adjustment (Section 2.1) to future work.

We use the network topology as discussed in Section 4.1 to carry out the experiments, with the addition of two Stargate nodes, one of which is also a physical sink. The other is a virtual sink that is placed at arbitrary locations in the test bed for different experiments.

For every scenario, we collect data for three different cases:

—The base case without any congestion control/avoidance mechanism (no CODA, no Siphon).
—CODA open loop control with priority support enabled. One of the sources (Src-3) leverages CODA's priority mechanism [Wan et al. 2003] and generates data packets with higher priority than the other two.
—Siphon with one virtual sink that is placed at arbitrary locations for different experiments (no CODA).

Five independent experiments are conducted for each case, and we calculate the average energy tax savings and fidelity ratio. Both metrics are normalized to the results obtained for the case without any congestion control/avoidance mechanism, allowing us to demonstrate the performance gain of both CODA and Siphon, and also the relative merit of each technique. Figure 12 presents this relative performance in bar charts for each respective source. From the figure, we observe that in terms of energy tax, CODA's open-loop hop-by-hop backpressure scheme has limited benefits in this scenario over the base case (i.e., energy tax savings is close to zero and fidelity ratio is close to one for Src-1

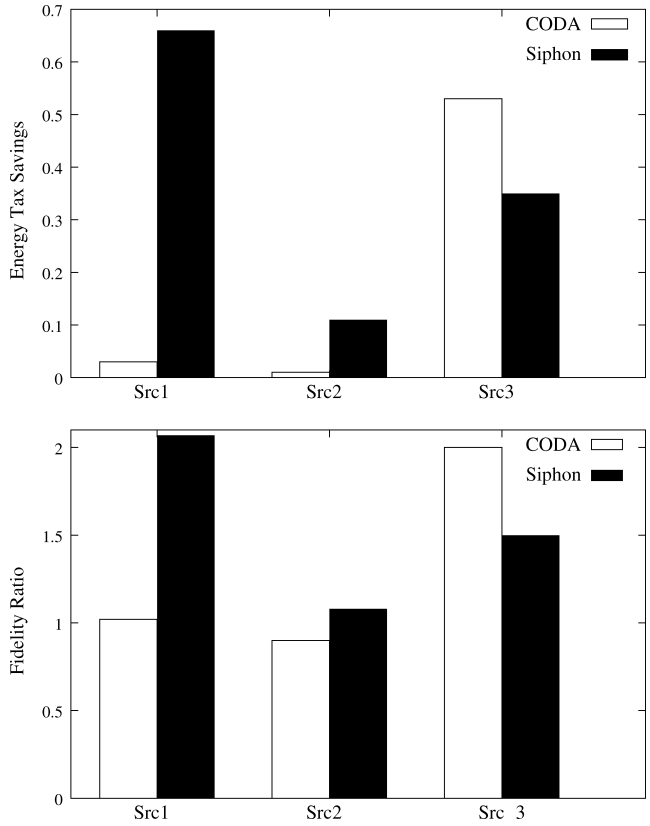

Fig. 12. Siphon performance in a real sensor network of 30 nodes. CODA priority favors Src-3; radio physical layer effects plague Src-2. Regardless, Siphon is able to improve both Fidelity Ratio and Energy Tax Savings for each source, repectively.

and Src-2) since the hotspot is far away from the sources and the congestion is persistent. The sources keep driving the channel beyond the congestion threshold with no feedback to lower the source rates, leading to collisions. However, CODA's priority support [Wan et al. 2003] for Src-3 improves both its energy tax (up to 55%) and fidelity (up to 200%) over the base case. On the other hand, Siphon improves both the energy tax (12% to 68%) and fidelity ratio (10%–110%) for all sources without any prioritization, by routing packets around the congestion areas, across highly reliable, high-bandwidth links. Note that because of testbed radio physical layer artifacts, the route that packets from Src-2 take to the sink suffers higher packet loss, leading to the observed lower energy tax savings and a lower fidelity ratio compared to the other sources in both the CODA and Siphon scenarios.

4.4 Post-Facto Traffic Siphoning

As discussed in Section 2.2, a physical sink can infer congestion by monitoring the event data quality, and enable "post-facto" traffic siphoning through the secondary network only when the measured application fidelity is degraded

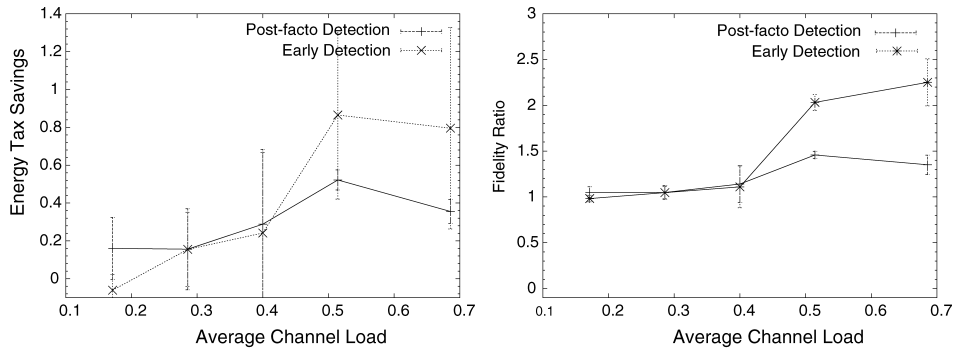

Fig. 13. (a) Comparing post-facto traffic redirection versus the early-detection approach. Energy Tax Savings is plotted against fractional channel load. (b) Comparing post-facto traffic redirection versus the early-detection approach. Fidelity Ratio is plotted against fractional channel load.

below a certain threshold. We implement an application agent that analyzes in real time the event data delivery ratio of each source at the physical sink. The agent calculates the moving average of the data delivery ratio using a window of five seconds, and initiates virtual sink signaling when the measured delivery ratio is lower than 60% for at least 10 seconds. Figure 13 presents the results as compared to its node-initiated early-detection based counterpart under different traffic loads.

Figure 13 plots fidelity ratio and energy tax savings against fractional channel load, and shows that while the post-facto approach does not perform as well as the node-initiated early-detection approach under high traffic load scenarios ($\geq$ 50% channel load), it performs as well in the lower traffic load region. In fact, the post-facto approach performs better than the node-initiated early-detection approach at traffic loads lower than 30% utilization. We observe that under low traffic load, the network sometimes suffers from poor connectivity or frequent collisions due to hidden terminals during the periods in which both the measured channel load and buffer occupancy are low. As a result, the measured data delivery ratio degrades and triggers post-facto traffic siphoning that improves subsequent data delivery, while in node-initiated early-detection approach the virtual sink is not utilized because of the perceived low channel load and buffer occupancy.

4.5 Evaluation with Surge Application

Surge periodically reports ADC readings to the sink at a rate that is programmable over-the-air using a control message. The Surge application employs the services of the MultiHopRouter [Woo and Culler 2003] component to set up and maintain a forwarding tree, based on packet-time granularity link quality estimation. In our test bed environment, during the course of preliminary measurements we found the addition of a small amount of application level randomization when sending the ADC sample appreciably reduces loss. We use Surge, with this small randomization, as the basis for all experiments described in the remainder of Section 4.

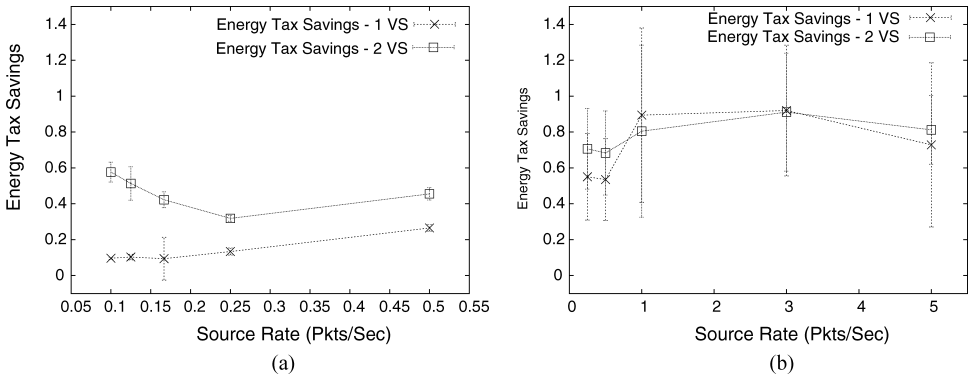

Fig. 14. (a) All nodes are Surge sources. Siphon provides increasing energy tax savings as the channel load increases in the network. Activating an additional virtual sink boosts this gain. The "2 virtual sink" curves exhibit a routing algorithm artifact at low rates. (b) A cluster of 4 nodes are Surge sources. Siphon provides an increasing fidelity ratio as channel load increases in the network. Activating an additional virtual sink has little effect due to virtual sink placement and packet propagation patterns.

We experimentally evaluate the performance of virtual sinks in terms of energy tax savings and fidelity ratio. Intuitively, when using Siphon, the energy tax savings and fidelity ratio should increase with increasing traffic load and through the deployment of more virtual sinks. Fixing the network traffic load and increasing the number of virtual sinks will give a larger fraction of the sensors access to siphon's overload management service. Conversely, fixing the number of virtual sinks in the network and increasing the network traffic load past the traffic redirection threshold (chosen as discussed in Section 4.2) results in a larger percentage of packets traversing the more reliable IEEE 802.11 secondary network to the physical sink, rather than the increasingly congested links of the primary mote network. Note that in general adding more virtual sinks is not always helpful because doing so may not provide 1-hop access to any new motes. Similarly, increasing the network traffic load beyond the point where the redirection paths to the virtual sinks become saturated does not increase the benefit of Siphon. We test this intuition by measuring energy tax savings and fidelity ratio for several combinations of network traffic load and virtual sink participation. While one can conceive of many possive traffic patters that may arise in a given sensor network, we restrict our evalation in this section to two common cases: a network-wide periodic monitoring pattern, and a localized event detection pattern.

4.5.1 *Traffic PatternI—All Nodes as Sources.* Figures 14(a) and 15(a) show the results when each mote in the grid is a Surge source transmitting at a common specified rate, creating a spatially uniform, temporally periodic traffic pattern. As the common source rate increases, this creates traffic in the network which moves from light load to overload. Results reflecting the performance of the 0, 1 and 2 virtual sink cases are shown in terms of the energy tax savings and the fidelity ratio of the 1 and 2 virtual sink cases with respect to the no virtual sink case. Each data point on the figure represents the average of six

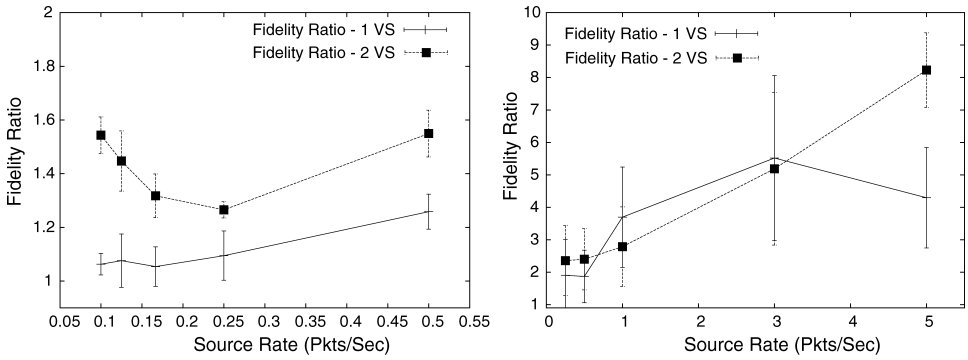

Fig. 15. (a) All nodes are Surge sources. Siphon provides increasing energy tax savings as the channel load increases in the network. Activating an additional virtual sink boosts this gain. The "2 virtual sink" curves exhibit a routing algorithm artifact at low rates. (b) A cluster of 4 nodes are Surge sources. Siphon provides an increasing fidelity ratio as channel load increases in the network. Activating an additional virtual sink has little effect due to virtual sink placement and packet propagation patterns.

10-minute trials with error bars marking the 95% confidence interval. The virtual sink visibility scope is set to 1, and the redirection threshold is set to 20% in accordance with the result from Section 4.2.

Regardless of the traffic load, the transition from no virtual sink to 1 virtual sink yields at least a 9% energy tax savings, while the addition of a second virtual sink increases the savings by at least 18% more. Similarly, the addition of the first virtual sink yields at least a 5% increase in fidelity measured at the physical sink, while the addition of a second virtual sink increases the fidelity by at least an additional 17%.

The concavity exhibited by the fidelity ratio and energy tax savings curves for the 2-virtual-sink case is an interesting artifact of the MultiHopRouter component's link quality estimation technique. MultiHopRouter chooses the next hop node (i.e., parent node) based on several factors, one of which is the bidirectional quality of the link between neighbors. Counters used to calculate this link quality are updated not only in the course of exchanging route control packets, but also during the forwarding of data packets. We observe that due to constants defined inside MultiHopRouter, a low source rate (on the order of the link quality estimate update interval), in concert with occasional loss, leads to a situation where a node may not have any valid parent when it needs to forward a packet. In such a case, MultiHopRouter sends the application packet to the broadcast address to jumpstart the link quality estimation process. However, broadcast packets are not forwarded by MultiHopRouter, that is, the packets are abandoned by the routing layer. While this state may not be persistent, at low rates the effect is pronounced enough to significantly lower the probability of a good path to the physical sink for nodes that are several hops away. Using additional control motes configured as passive snoopers we observe a large number of Surge application packets are sent to the broadcast address. In fact, at low rates the parent selection technique may actually contribute to traffic

funneling anywhere in the network, since it has the effect of channeling packets along well-trodden paths. In the experiments shown in Figures 14(a) and 15(a), the virtual sinks are arbitrarily placed within the grid. In the 1-virtual-sink case, the virtual sink is placed nearer to the physical sink, while in the 2-virtual-sink case the second virtual sink is placed farther from the physical sink. Thus, while the single virtual sink provides the normal advantage in the face of congestion (relatively rare at the lowest rates we tested), the position of the second virtual sink relative to the physical sink allows it to provide the additional benefit of servicing motes disadvantaged by unstable paths to the physical sink at low rates.

As the source rate at each mote increases from $\frac{1}{10}$ pkts/sec to $\frac{1}{4}$ pkts/sec the magnitude of the aforementioned link atrophy effect diminishes. As implied by the curves for the 2-virtual-sink case, the minimum application data rate necessary to support the formation of stable paths is between $\frac{1}{6}$ pkts/sec and $\frac{1}{4}$ pkts/sec in our testbed. While this numerical value may be specific to our testbed configuration, we conjecture that such a threshold will exist in the general context too. As the source rate increases beyond $\frac{1}{4}$ pkts/sec all the energy tax savings and fidelity ratio curves trend as expected. Note that the delivery probability in general decreases with increasing number of hops, so a virtual sink placed relatively farther from the physical sink is likely to be more valuable in such a medium-scale network. However, this effect alone would not explain the observed concavity.

4.5.2 *Traffic PatternII—4 Nodes as Sources.* Figures 14(b) and 15(b) show the results when a 1×4 cluster of four motes are centrally located in the grid four hops from the physical sink. Unlike the previous experiment where all the motes were sources, in this case, these four clustered motes are the only active sources in the sensor network. Each of the four transmits at the same specified rate, creating a spatially concentrated impulse traffic pattern. Activating the source cluster emulates detection of a localized event that must be relayed to the physical sink. Each data point represents the average of six 5-minute trials with the error bars indicating the 95% confidence intervals. Again, the virtual sink visibility scope is set to 1, and the redirection threshold is set to 20%. Virtual sinks, when in use, are placed adjacent to the source cluster for the reasons discussed below.

As with the previous results, use of virtual sinks provides significant gains in terms of energy tax savings and fidelity ratio, providing at least a 53% increase in energy tax savings and at least a 87% increase in fidelity ratio. However, the difference between the 1-virtual-sink and 2-virtual-sink curves is minor and within confidence intervals, except for the fidelity ratio corresponding to the highest source rate. Furthermore, the energy tax savings is flat (within confidence intervals, and with only about a 15% increase in the mean) across the source rates. We observe that for the highest three source rates over 80% (include over 90% for the highest rate) of the delivered traffic arrives via a virtual sink, and even at the two lower rates this fraction is 55%.

Because the majority of the traffic is redirected over the highly reliable secondary long-range network, we observe a flat energy tax savings curve and a

fidelity curve that increases linearly with the source rate, as expected. The fact that both virtual sinks are placed near the source motes in this configuration also explains the relative performance between the 1-VS and 2-VS fidelity ratio curves that are shown in Figure 15(b). Above the 3 pkts/sec source rate of the 1-VS fidelity ratio curve in Figure 15(b) saturates because fewer of the sources are able to successfully redirect packet to the virtual sink because of congestion collapse in the vicinity of the virtual sink. The 2-VS fidelity ratio curve continues to increase linearly across the tested source rate range since the load of the four sources is roughly split between the two virtual sinks, keeping the network in the vicinity of the virtual sinks out of the collapse regime. The fact that most of delivered packets arrive via the secondary network is again explained by the MultiHopRouter link quality estimation technique. Recall that in this scenario, the only Surge packets originate from the four source motes. Most of the network is therefore devoid of application traffic, which, as explained previously, leads to a lack of stable multihop routes in those areas. Thus, when the direct path from the cluster of sources to the physical sink becomes congested, there is a low probability of finding an alternative path. In any case, in the grid such a path would likely be longer, implying an additional decrease in the probability of eventual delivery. As observed previously, packets are sent to the broadcast address and are not forwarded to the physical sink, in the absence of a valid next hop (parent node), whether due to actual congestion or link atrophy. We observe that for the packets not unicast to a virtual sink, a large overall percentage of packets, and an overwhelming majority for the highest three rates, are sent to the broadcast address, indicating that they had no parent node at the time of packet transmission. In fact, when virtual sinks are placed further than two hops from the source cluster, they are very rarely utilized because of this link atrophy/broadcast problem; congestion remains localized to the vicinity of the single source cluster due to the routing bottleneck and little traffic propagates to alternative paths. When the virtual sinks are both placed near the source motes such that the congestion region overlaps the advertisement scope of the virtual sinks, they are able to provide a large increase in network performance. The paths from the sources to the virtual sinks are short and have a greater probability of being well maintained by the routing protocol. Another result of this bimodal network behavior is the large degree of variability in fidelity, as indicated by the large confidence intervals in Figure 15(b). Either good paths are set up between most of the source motes and a virtual sink, in which case the fidelity is quite high, or few good paths between the source motes and virtual sinks exist and fidelity is significantly reduced since the primary network does not have stable multihop paths from the source motes to the physical sink.

4.5.3 *Time Series Traces.* Figure 16 shows fixed-rate time series traces of the network fidelity, (i.e., packets/sec received at the physical sink). Each point represents an average of a 10 second interval. The “4 sources” trace comprises three 5-minute intervals where 0 (0-300s), 1 (300-600s) and 2 (600-900s) virtual sinks, respectively, are active. The cluster of four motes sends Surge packets into the network at a rate of 5 pkts/sec. The “48 sources” trace comprises three 10-minute intervals where 0 (0-600s), 1 (600-1200s) and 2 (1200-1800s) virtual

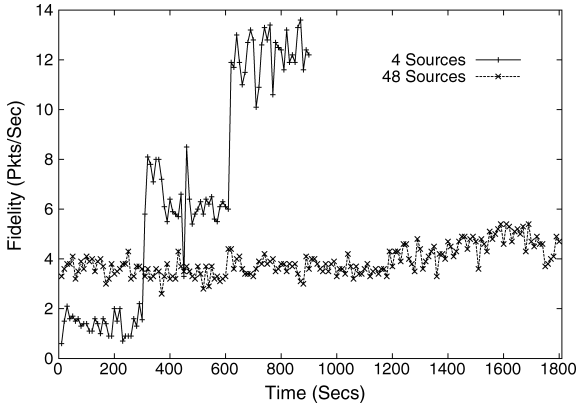

Fig. 16. A time series trace showing packet reception rate at the physical sink. The four nodes in the source cluster each send at 5 pkts/sec. When all 48 motes are sources, the rate is $\frac{1}{6}$ pkts/sec. Each trace comprises three intervals corresponding respectively to 0, 1, and 2 virtual sinks activated in the network. Performance gain due to virtual sink participation is immediate but the magnitude depends strongly on traffic pattern and virtual sink placement.

sinks, respectively, are active. All motes are sources injecting Surge packets at a rate of $\frac{1}{6}$ pkts/sec. As expected, in the "4 sources" case, the transitions between intervals are abrupt and substantial. This is a result of the virtual sink placement and MultiHopRouter parent selection strategy discussed in Section 4.5 with over 90% of the delivered traffic flowing through a virtual sink at 5 pkts/sec.

In the "48 sources" case, the interval transitions in the time series, though discernible, are much more modest. At $\frac{1}{6}$ pkts/sec, only about 20% of the delivered traffic flows through a virtual sink. Two factors contribute to this reduced virtual sink involvement, compared to the "4 sources" case. First, many sources are close to the physical sink and can deliver packets directly or in a small number of hops. Second, the source rate is on the cusp of being high enough to eliminate the MultiHopRouter link atrophy problem that plagues lower event rates. Note, however, that the addition of the second virtual sink at 1200 secs into the trace boosts the fidelity more than the addition of the first virtual sink at 600 secs into the trace. We conjecture that this indicates link atrophy is still a problem at $\frac{1}{6}$ pkts/sec.

4.6 Virtual Sink Cost Analysis

The results in previous sections show the benefit of utilizing Siphon to enhance network performance for increasing channel load. In what follows, we present results on the cost of using Siphon. In Section 3.6, we mentioned that using virtual sinks in the "always-on" mode drains energy more quickly than in the "on-demand" mode. This is strongly dependent on the particular characteristics of the radio interfaces used in the primary and secondary networks. Using a simple model that captures the energy spent in delivering packets in our testbed, we offer some additional insight into the "on-demand" versus "always-on" question and the cost of using virtual sinks in an experimental testbed.

To analyze the cost of Siphon we use a metric called *virtual sink usage cost ratio*, that normalizes the energy cost of delivering packets to the physical sink when virtual sinks are active, by the energy cost of delivering the same number of packets using only the primary network. Our energy model captures the energy spent in our testbed during transmission and reception of delivered packets, and includes protocol characteristics of sending a packet, such as the average CSMA backoff time when using TinyOS/Mica2 and the IEEE 802.11b MAC acknowledgment message. Virtual sink usage cost (VSUC) is defined as

$$VSUC = \frac{E_{delivery,VS}}{E_{delivery,noVS}}, \tag{3}$$

where

$$\begin{aligned} E_{delivery,noVS} &= \lambda N h_{avg,PS}(e_{mote,tx} + e_{mote,rx}) \\ &\quad + (1-\lambda) N e_{mote,tx}, \end{aligned}$$

and

$$\begin{aligned} E_{delivery,VS} &= \alpha\lambda N h_{avg,PS}(e_{mote,tx} + e_{mote,rx}) \\ &\quad + (1-\lambda) N e_{mote,tx} \\ &\quad + (1-\alpha)\lambda N (e_{wifi,tx} + e_{wifi,rx}) \\ &\quad + (1-\alpha)\lambda N h_{avg,VS}(e_{mote,tx} + e_{mote,rx}) \\ &= \alpha E_{delivery,noVS} \\ &\quad + (1-\alpha) N \{(1-\lambda) e_{mote,tx} \\ &\quad + \lambda(e_{wifi,rx} + e_{wifi,tx} \\ &\quad + h_{avg,VS}(e_{mote,tx} + e_{mote,rx}))\}. \end{aligned}$$

In these expressions, N is the aggregate number of packets that originate at the sources, λ is the fraction of these originated packets that are delivered to the physical sink, $h_{avg,PS}$ is the average number of hops from the sources to the physical sink, $e_{mote,tx}$ and $e_{mote,rx}$ are the transmit and receive energy, respectively, for the Mica2 MAC and physical layers, α is the fraction of delivered packets that use the mote network exclusively, $h_{avg,VS}$ is the average number of hops that source packets delivered on the WiFi network take on the mote network before reaching the virtual sink, and $e_{wifi,tx}$ and $e_{wifi,rx}$ are the transmit and receive energy, respectively, of the IEEE 802.11b MAC and physical layers.

N is determined by the specified source rate. $\lambda, \alpha, h_{avg,PS}$ and $h_{avg,VS}$ are taken from our experimental data logs. From PowerTOSSIM [2006] the measured transmit power for our CC1000 [Chipcon 2006] firmware setting (0dBm) is 41.31mW. The PowerTOSSIM [2006] receive power is 21.09mW. Including the application packet length, preamble and start symbol in the TinyOS MAC implementation for the Mica2 platform, we calculate a packet time of 27.5ms at 19.2kbps [TinyOS 2006]. We assume, somewhat optimistically, that in the mote network only the initial CSMA backoff is needed giving an average backoff time per transmission of 3.125ms. Further, the receive/transmit mode switch time is 0.4ms [Chipcon 2006]. Using these values, we calculate $e_{mote,tx} = 1210.325\mu J$

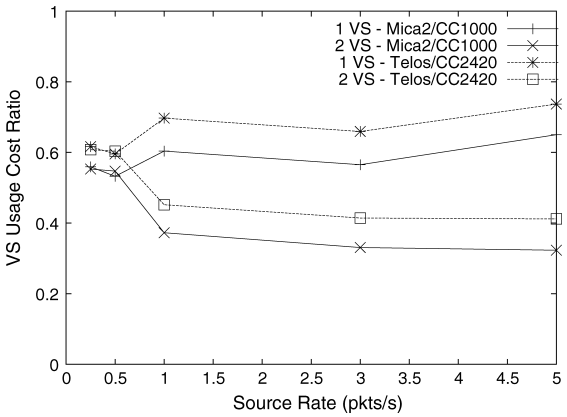

Fig. 17. Virtual sink usage cost ratio measures the energy cost of using virtual sinks to deliver packets to the physical sink normalized by the cost of delivering the same number of packets using only the primary radio network. In terms of the packet transmission costs captured by our energy model, using Siphon actually saves energy in both the CC1000 and CC2420 cases.

and $e_{mote,rx} = 597.975\mu J$. From Chen et al. [2002], the transmit and receive power are 1.4W and 1.0W, respectively. Including the average backoff, SIFS, DIFS, and DATA and ACK packet times (including TCP/IP, MAC and PLCP headers), the average transmission and receive times are calculated yielding a packet transmit energy of $e_{wifi,tx} = 1184.55\mu J$ and a packet receive energy of $e_{wifi,rx} = 581.82\mu J$. Our IEEE 802.11b network does not use RTS/CTS. Note that we do not include in our model the possibility that more than one mote receives the packet at each hop, though this is likely the case. However, given the relatively higher anticipated density of the primary mote network to the secondary network and that the calculated reception cost is also higher for the mote network than the IEEE 802.11b network, omitting these receive costs does not favor Siphon.

Figure 17 presents Siphon cost in terms of the virtual sink usage cost ratio for the four source cluster scenario discussed earlier. For the four source cluster, we see that the cost of using Siphon is actually less than using only the primary network in our testbed. Since the visibility scope of the virtual sinks is set to 1, and the virtual sinks are placed near the sources, the average hop distance for both the 1 and 2 virtual sink cases is half that of the no virtual sink case. As the per-packet energy costs using TinyOS/Mica2 and IEEE 802.11b are similar, the average hop distance from the source to the physical sink and the low α (due to the link atrophy problem described in Section 4.5.1) dominates the energy calculations, leading to the advantageous result for Siphon. However, even with a high percentage of the packets traversing the secondary network in this scenario, the IEEE 802.11b channel is still lightly loaded, implying a high percentage of idle time if the secondary radio is “always-on”. Incorporating the energy cost of this idle time into the model would greatly increase the cost of “always-on” operation, arguing for an “on-demand” virtual sink usage model. However, because of application delay requirements using

virtual sinks in an "always-on" manner may still be desireable (see discussion in Section 3.6).

In general, the exact values for λ, α, h, and e are dependent on topology, density, deployment radio environment, and radio technology (antenna characteristics, etc.). However, to get a sense of how the results shown in Figure 17 using Mica2 motes might apply to other types of mote radios, we assume a deployment configuration such that our testbed values for λ, α and h are applicable, and then substitute the values for $e_{mote,rx}$ and $e_{mote,tx}$ for the Telos platform [Polastre et al. 2005], wich uses the CC2420 radio [Chipcon 2006]. From Chipcon [2006], the transmit power drain at 0dBm is 31.32mW and the receive power drain is 35.46mW. From the measured results presented in Polastre et al. [2005], we use an approximate packet time of 18.7ms, which incorporates the MAC frame, preamble, start symbol CSMA backoff and receive/transmit mode switch times. Using these values we calculate $e_{mote,tx} = 585.684\mu J$ and $e_{mote,rx} = 663.102\mu J$. Figure 17 shows that the advantage offered by Siphon extends to CC2420-based mote platforms, though the advantage is lessened due to the higher bandwidth of the CC2420 as compared to the CC1000 [Chipcon 2006].

5. RELATED WORK

The idea of utilizing multiple coordinated radios operating over multiple channels to improve and optimize wireless network capacity was first proposed in Shih et al. [2002]. In Shih et al. [2002], the authors exploit the possibility of adding a second low-power radio of lower complexity and capability into a node in a wireless LAN network to increase the battery lifetime of the node. The main idea is to use the secondary lower-power radio to wake up a node, allowing the node to shutdown the primary radio during idle periods. There are a growing number of projects using multi-radio Stargate systems to deliver new services to sensor networks. The ExScal project [Arora et al. 2005] recently demonstrated the largest deployment of sensors and Stargates to date. By strategically positioning 200 Stargates uniformly across a sensor field, sensors could directly communicate with Stargates within one hop. Our work on overload traffic management is similar in spirit to the ExScal project, but we are primarily motivated by using the Stargate secondary network only when overload occurs in the mote network, rather than providing an always-on expedited or low delay transport service to the mote network. Another project related to our work exploits [Yarvis et al. 2005] heterogeneity in sensor networks through the use a small number of line-powered back-hauled nodes (i.e., Stargates) that connect to the wired network. Yarvis et al. [2005] analytically prove that their approach increases network reliability and the lifetime of a sensor network based on a simple grid topology. While Yarvis et al. [2005] specifically discuss the use of wired Ethernet as the secondary access, IEEE 802.11 could also be used. To the best of our knowledge our work represents the first use of multiradio sensors that deals specifically with countering the funneling effect by offering on-demand overload traffic management services to the mote network.

Basu et al. [2004] propose two centralized schemes aimed at reducing the mean end-to-end network delay in a multihop wireless network. Both schemes are proven, after simplifying assumptions, to alleviate congestion by redirecting traffic around bottleneck links. However, given the assumptions made (e.g., known topology, slow traffic dynamics, automatic power control/CDMA), the solutions are not necessarily appropriate to general wireless sensor networks. Furthermore, the many-to-one traffic pattern associated with wireless sensor networks (i.e., the funneling effect) is not well addressed by the authors' notion of avoiding bottleneck links since close to the sink all links are likely to be bottlenecks. In Liu et al. [2003] Liu and Towsley study the throughput capacity of hybrid wireless networks formed by placing base stations in an ad hoc network. Assuming a network model that is a hybrid of cellular network and wireless LAN, the authors prove the existence of a threshold for the scaling of the number of base stations with respect to the number of nodes in a network in order to gain nonnegligible capacity benefit. While their analytical result provides insights into the investment requirement for deploying a hybrid network, their model does not take into account the energy issues (the base stations are connected by a high-bandwidth wired network) which is a main concern in an all-wireless sensor network environment supported by virtual sinks.

There is a growing body of work on congestion control for sensor networks. However, these schemes impact fidelity when trying to mitigate congestion in sensor networks. Hull et al. [2004] experimentally investigate the end-to-end performance of various congestion avoidance techniques in a 55-node sensor network. They propose a strategy called Fusion that combines three congestion control techniques that operate at different layers of the traditional protocol stack. These techniques include hop-by-hop flow control, a source rate limiting scheme similar to the adaptive rate control mechanism proposed in Woo and Culler [2001] that meters traffic being admitted into the network, and a prioritized MAC layer that gives a backlogged node priority over non-backlogged nodes for access to the shared medium. ESRT [Sankarasubramaniam et al. 2003] regulates the reporting rate of sensors in response to congestion detected in the network by monitoring the local buffer level of sensor nodes. CODA [Wan et al. 2003] is based on a low-cost measurement-based channel sampling scheme, hop-by-hop backpressure that rate controls traffic, and a closed-loop multisource regulation scheme.

6. CONCLUSION

There is a growing need for improved congestion control, load balancing, and overload traffic management in emerging sensor networks. The first generation congestion avoidance mechanisms are effective at limiting packet loss due to congestion and allowing the network to find a stable operating point under increasing load. However, these mechanisms are not sufficient to deal with the new types of congestion that are an artifact of the funneling effect and a product of increasing workload. In this paper, we have proposed the use of multiradio virtual sinks that support Siphon's on-demand overload traffic management services to counter the funneling effect. We evaluated Siphon using

simulations and experimentation to gain insights into its performance and ability to interwork with Directed Diffusion and Surge, as representative applications. We plan to release the Siphon source code as part of the Armstrong Project [2006].

As a broader comment, our contribution is the exploration of general design principles that enable exploitation of special nodes, such as dual-radio virtual sinks, to increase the resilience of sensor networks with affordable cost. The idea of using special nodes can be pushed to a higher level of abstraction. For example, though in this paper we exploit a virtual sink's characteristic of longer transmission range, the same concept can be extended to nodes with higher transit bandwidth, larger storage space or enhanced computation capability. Therefore, we believe Siphon's algorithms and signaling mechanisms are more broadly applicable to a class of new applications that exploit the capabilities of such special nodes.

REFERENCES

ARMSTRONG PROJECT 2006. http://comet.columbia.edu/armstrong.

ARORA, A., RAMNATH, R., AND ERTIN, E. 2005. Exscal: Elements of an extreme scale wireless sensor network. In *Proceedings of the 11th International Conference on Embedded and Real-Time Computing Systems and Applications*. Hong Kong. IEEE, 102–108.

BASU, A., BOSHES, B., MUKHERJEE, S., AND RAMANATHAN, S. 2004. Network deformation: Traffic-aware algorithms for dynamically reducing end-to-end delay in multi-hop wireless networks. In *Proceedings of the 10th Annual International Conference on Mobile Computing and Networking*. Philadelphia, PA. ACM, 100–113.

CHEN, B., JAMIESON, K., BALAKRISHNAN, H., AND MORRIS, R. 2002. Span: An energy-efficient coordination algorithm for topology maintenance in ad hoc wireless networks. *ACM Wirel. Netw. 8*, 5, 85–96.

CHIPCON. 2006. Chipcon CC1000 and CC2420 Data Sheets 2006. http://www.chipcon.com.

CLARE, L. P., POTTIE, G., AND AGRE, J. R. 1999. Self-organizing distributed microsensor networks. In *Proceedings of 13th Annual International Symposium on Aerospace/Defense Sensing, Simulation, and Controls*. Orlando, FL. SPIE, 229–237.

DAM, T. V. AND LANGENDOEN, K. 2003. An adaptive energy-efficient mac protocol for wireless sensor networks. In *Proceedings of 1st Conference on Embedded Networked Sensor Systems*. Los Angeles, ACM, 171–180.

DEB, B., BHATNAGAR, S., AND NATH, B. 2003. Multi-resolution state retrieval in sensor networks. In *Proceedings of the 1st Workshop on Sensor Network Protocols and Applications*. Anchorage, Al. IEEE, 19–29.

HE, T., BLUM, B. M., STANKOVIC, J. A., AND ABDELZAHER, T. 2004. Aida: Adaptive application-independent data aggregation in wireless sensor networks. *ACM Trans. Embed. Comput. Syst. 3*, 2, 426–457.

HILL, J., SZEWCZYK, R., WOO, A., HOLLAR, S., CULLER, D., AND PISTER, K. 2000. System architecture directions for network sensors. In *Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems*. Cambridge, ACM, MA, 93–104.

HULL, B., JAMIESON, K., AND BALAKRISHNAN, H. 2004. Mitigating congestion in wireless sensor networks. In *Proceedings of the 2nd Conference on Embedded Networked Sensor Systems*. Baltimore MA. ACM, 134–147.

INTANAGONWIWAT, C., GOVINDAN, R., AND ESTRIN, D. 2000. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*. Boston, MA. ACM, 56–67.

LI, J., BLAKE, C., COUTO, D. D., LEE, H., AND MORRIS, R. 2001. Capacity of ad hoc wireless networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*. Rome, Italy. ACM, 61–69.

LIU, B., LIU, Z., AND TOWSLEY, D. 2003. On the capacity of hybrid wireless networks. In *Proceedings of the 22nd International Annual Joint Conference of the IEEE Computer and Communications Societies*. San Francisco, CA. IEEE, 1543–1552.

MURTY, R., QI, E. H., AND HAZRA, M. 2004. An adaptive approach to wireless network performance optimization.

NATH, B. AND NICULESCU, D. 2003. Routing on a curve. *ACM SIGCOMM Comput. Commun. Rev. 33*, 1, 155–160.

NAVAS, J. C. AND IMIELINSKI, T. 1997. Geographic addressing and routing. In *Proceedings of the 3rd Annual International Conference on Mobile Computing and Networking*. Budapest, Hongary. ACM/IEEE, 66–76.

POLASTRE, J., HUI, J., LEVIS, P., ZHAO, J., CULLER, D., SHENKER, S., AND STOICA, I. 2005. A unifying link abstraction for wireless sensor networks. In *Proceedings of the 3rd Conference on Embedded Networked Sensor Systems*. San Diego, CA. ACM, 76–89.

POLASTRE, J., SZEWCZYK, R., AND CULLER, D. 2005. Telos: Enabling ultra-low power wireless research. In *Proceedings of the 4th International Conference on Information Processing in Sensor Networks: Special Track on Platform Tools and Design Methods for Network Embedded Sensors*. Los Angeles, CA. ACM/IEEE, 364–369.

POWERTOSSIM. 2006. PowerTOSSIM Mica2 Power Benchmarks 2006. `http://www.eecs.harvard.edu/~shnayder/ptossim/mica2bench/summary.html`.

RAJENDRAN, V., OBRACZKA, K., AND GARCIA, J. 2003. Energy-efficient, collision-free medium access control for wireless sensor networks. In *Proceedings of 1st Conference on Embedded Networked Sensor Systems*. Los Angeles, CA. ACM, 181–192.

SANKARASUBRAMANIAM, Y., AKAN, O., AND AKYILDIZ, I. 2003. Event-to-sink reliable transport in wireless sensor networks. In *Proceedings of the 4th Symposium on Mobile Ad Hoc Networking & Computing*. Annapolis, MA. ACM, 177–188.

SHIH, E., BAHL, P., AND SINCLAIR, M. J. 2002. Wake on wireless: An event driven energy saving strategy for battery operated devices. In *Proceedings of the 8th Annual International Conference on Mobile Computing and Networking*. Atlanta, CA. ACM, 160–171.

STARGATE. 2006. Stargate datasheet 2006. `http://www.xbow.com`.

TILAK, S., ABU-GHAZALEH, N. B., AND HEINZELMAN, W. 2002. Infrastructure tradeoffs for sensor networks. In *Proceedings of the 1st International Workshop on Wireless Sensor Networks and Applications*. Atlanta, CA. ACM, 49–58.

TINYOS. 2006. Tinyos homepage 2006. `http://www.tinyos.net`.

TOLLE, G. AND CULLER, D. 2005. Design of an application-cooperative managment system for wireless sensor networks. In *Proceedings of the 2nd European Workshop on Wireless Sensor Networks*. Istanbul, Turkey. IEEE, 121–132.

WAN, C.-Y., CAMPBELL, A. T., AND CROWCROFT, J. 2004. A case for all-wireless dual-radio virtual sinks. In *Proceedings of the 2nd Conference on Embedded Networked Sensor Systems*. Baltimore, MA. ACM, 267–268.

WAN, C.-Y., EISENMAN, S. B., AND CAMPBELL, A. T. 2003. Coda: Congestion detection and avoidance in sensor networks. In *Proceedings of the 1st Conference on Embedded Networked Sensor Systems*. Los Angeles, CA. ACM, 266–279.

WAN, C.-Y., EISENMAN, S. B., CAMPBELL, A. T., AND CROWCROFT, J. 2005. Siphon: Overload traffic management using multi-radio virtual sinks. In *Proceedings of the 3rd Conference on Embedded Networked Sensor Systems*. San Diego, CA. ACM, 116–129.

WOO, A. AND CULLER, D. 2001. A transmission control scheme for media access in sensor networks. In *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*. Rome, Italy. ACM, 221–235.

WOO, A. AND CULLER, D. 2003. Taming the underlying challenges of reliable multihop routing in sensor networks. In *Proceedings of the 1st Conference on Embedded Networked Sensor Systems*. Los Angeles, CA. ACM, 14–27.

XUE, F. AND KUMAR, P. R. 2004. The number of neighbors needed for connectivity of wireless networks. *ACM Wirel. Netw. 10*, 2, 169–181.

YARVIS, M., KUSHALNAGAR, N., SINGH, H., RANGARAJAN, A., LIU, Y., AND SINGH, S. 2005. Exploiting heterogeneity in sensor networks. In *Proceedings of the 24th International Annual Joint Conference of the IEEE Computer and Communications Societies*. Miami, FL. IEEE, 878–890.

YE, W., HEIDEMANN, J., AND ESTRIN, D. 2002. An energy efficient mac protocol for wireless sensor networks. In *Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies*. New York, NY. IEEE, 1567–1576.

ZHAO, J. AND GOVINDAN, R. 2003. Understanding packet delivery performance in dense wireless sensor network. In *Proceedings of the 1st Conference on Embedded Networked Sensor Systems*. Los Angeles, CA. ACM, 1–13.

Received February 2006; revised December 2006; accepted January 2007