This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2010 proceedings

Fair End-to-end Bandwidth Distribution in Wireless Sensor Networks

Ying Jian Shigang Chen
Department of Computer & Information Science & Engineering
University of Florida

Shiping Chen
Network Information Center
University of Shanghai for Science and Technology

Yibei Ling
Applied Research Laboratories
Telcordia Technologies

Abstract— **End-to-end fairness in a sensor network ensures that data from each sensor has an equal (or weighted) chance to reach the sink. It eliminates the problem that data flows from sensors at some locations (close to the sink) obtain most network bandwidth, while flows from sensors at other locations (far away from the sink) are starved. Existing fairness solutions assume single-path routing, which reduces achievable throughput of the network. In this paper, we propose a multipath fairness solution (MFS) that achieves end-to-end fairness among competing flows through fully distributed operations. MFS is easy to implement, which is advantageous in a resource-scarce sensor network. More importantly, it achieves much higher network throughput and better fairness among the flows.**

I. INTRODUCTION

Sensor networks typically operate under light load and suddenly become active in response to certain important events such as fire outbreak, earthquake, or enemy movement [1]. The sudden surge of data from hundreds or even thousands of sensors must be delivered to a small number of base stations, which may cause congestion, especially near the base stations. It is a critical problem to resolve the congestion fairly, such that all data sources have equal or weighted access to the network bandwidth.

Many fairness solutions for wireless networks can only be applied to one-hop flows at the MAC layer [2], [3], [4], [5]. As we will show later, MAC-layer fairness causes extremely biased bandwidth allocations for end-to-end flows. The existing end-to-end fairness solutions also have their limitation. ESRT [6] enforces a common reporting rate for all data sources. To remove congestion, the reporting rate has to be set conservatively according to the worst hotspot in the network. Proportionally-fair congestion control in FDMA/CDMA networks is studied in [7], under the assumptions that each flow has a single routing path and that per-flow state information is maintained at intermediate nodes, which is undesirable for resource-constrained sensor networks. Fairness in TDMA networks is studied in [8], and flow-level fairness in a sensor network is investigated in [9]. Both of them assume a tree routing structure from all data sources to a sink. The rate limiting technique in [10] also requires a tree routing structure in order to achieve fairness. The rate-control mechanism in [11] ensures that the ratio between the rate of through traffic from n upstream data sources and the rate of locally generated traffic is $1 : n$ at a congested node. Fairness is achieved only with single-path routing because the ratio, $1 : n$, would be biased for through traffic if the n upstream sources could acquire additional bandwidth from other paths. Other fairness work in wireless ad-hoc networks also assumes single-path routing [12], [13]. The temporal fairness in multi-rate wireless backhaul networks is investigated in [14], but no distributed algorithm is given to realized such fairness.

The assumption of single-path routing made by the prior solutions is restrictive. It does not take advantage of forwarding flexibility and routing robustness offered by many popular routing protocols that support multipath routing. More importantly, our simulations demonstrate that single-path routing can significantly reduce network throughput and depress the average rate of end-to-end flows.

Achieving fairness in a multipath routing environment is a difficult problem. In this paper, we propose a multipath fairness solution (MFS) that allocates network bandwidth fairly among competing flows through fully distributed, localized operations. The basic idea is for a forwarding node to estimate the number of flows incoming from each adjacent link and allocate bandwidth proportional to that number. Because packets of a flow may be routed on many paths, if the flow passes a link, chances are only a fraction of the flow does so. Therefore, the number of flows passing a link is the summation of the flow fractions passing the link. Our proposed solution is able to to estimate this (fractional) number for each link. We also extend MFS for weighted fairness, which allocates network bandwidth to flows in proportion to their weights. We evaluate MFS by extensive simulations. The simulation results demonstrate that MFS achieves far better flow-level fairness than the existing approaches and the bandwidth allocation can be easily made biased towards important data sources by different weight assignments. Its average rate per flow is much higher than the rate that is achievable under single-path routing.

II. NETWORK MODEL

In a data-collection sensor network, battery-powered sensors produce data packets and forward them to the sink,

978-1-4244-6404-3/10/$26.00 ©2010 IEEE

which consists of one or multiple base stations. All sensors participate in relaying packets towards the sink though only some of them are data sources. Assume that the base stations are connected via an external network to a data collection center. It makes no difference which particular base station a packet is delivered to.

A data source generates packets at a rate Ω by default. When the routing paths do not have sufficient bandwidth, a data source will be forced to generate data at a rate smaller than Ω. Data packets generated from the same sensor constitute a *flow*. Each packet carries the data measured by a sensor, the time of the measurement, and the location of the measurement. We assume the application requires the actual data instead of their aggregate such as min/max/avg. For example, the application may need to measure the temperature distribution over a field instead of the average temperature.

A wireless communication *link* exists from one sensor to another if the latter can correctly receive the former's signal. Two sensors are neighbors if they can directly communicate with certain reliability. Assume the MAC-layer protocol is CSMA-based. In order to perform DATA/ACK exchange, only symmetric (two-way) links are used for sending data.

Let N be the set of sensors and N_x be the set of neighbors of a sensor x. Let D_x be the set of *downstream neighbors*, which are the next hops from x to the base stations. $\forall y \in D_x$, (x, y) is called a *downstream link*. Let U_x be the set of *upstream neighbors*, which uses x as a next hop to the base stations. $\forall y \in U_x$, (y, x) is called an *upstream link* of x. If y is an upstream neighbor of x, then x must be a downstream neighbor of y. D_x is determined by a routing protocol. For example, if geographic routing is used [15], [16], D_x consists of all neighbors whose distances to the nearest base stations are smaller. It is not true that increasing the size of D_i will always improve throughput. When the nodes in D_i forward packets, they compete with each other for media access, which can cause collisions. A compromise is to limit the size of D_i and pick those candidate downstream neighbors that are far from each other.

The downstream links of all sensors form a directed acyclic graph in which the base stations constitute the sink. When a packet is received from an upstream neighbor, a sensor may forward the packet to any downstream neighbor. By following any sequence of downstream links, a packet will eventually reach a base station. Packets of a flow generated from a data source are routed individually at the intermediate sensors and therefore may follow different paths even to different base stations. In this sense, we say, "a flow has multiple routing paths." It does not mean that the network explicitly establishes multiple routing paths for each specific flow.

We assume all transceivers operate at a single transmission rate, which is reasonable due to the cheap design requirement for inexpensive one-time sensors that are used in large quantities. We further assume the clocks of the sensors are loosely synchronized during the life time of the network. Sensors are statically located after deployment. We do not consider mobile sensors that form a dynamic ad-hoc network.

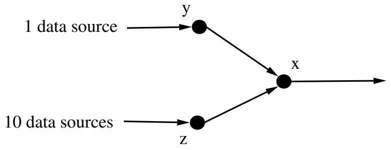

Fig. 1. MAC fairness causes end-to-end unfairness and severe packet drops.

III. Multipath Fairness Solution

In this section, we propose our end-to-end fairness solution in the context of multipath routing.

A. Multipath Challenge

In order to treat each flow equally, when a node x is congested, it should distribute its downstream forwarding capacity to upstream links in proportion to the number of flows passing each link. For example, in Figure 1, x should allocate $\frac{10}{11}$ of its forwarding capacity to z and $\frac{1}{11}$ to y, such that all 11 flows will have the same rate.

However, if multipath routing is enabled, when packets of a flow pass through a link, they are likely to account for only a small fraction of that flow. The same flow may simultaneously pass through many parallel links on different paths. In Figure 1, under multipath routing, a flow may pass both (y, x) and (z, x). Therefore, the *number* of flows passing a link is a fractional number (instead of an integer). It is the summation of the flow fractions that pass the link. How to keep track of this number for each link is a difficult problem because the flows may arbitrarily split their packets among multiple paths.

B. Flow Size Field

We introduce a *flow size field* in the header of a data packet p. The field is denoted as $s(p)$, which is l bits long. A data source x measures the number $n(x)$ of packets generated during each time interval of length $\frac{2^l-1}{\Omega}$. Recall that Ω is the maximum packet rate that a source produces. The number of packets generated in a time interval by any source is between zero to $2^l - 1$, which is able to fit in the flow size field.

Let $n_{new}(x)$ be the number of packets generated during a new time interval. $n(x)$ is updated after the interval as an exponential moving average.

$$n(x) \Leftarrow (1 - \alpha) \times n(x) + \alpha \times n_{new}(x) \qquad (1)$$

where α is a system parameter between 0 and 1. Whenever a source sends out a locally-generated data packet, it sets the flow size field $s(p)$ to be the current value of $n(x)$. The packet represents a fraction of the flow generated from the data source. The fraction is estimated as the inverse of the flow size field, i.e., $\frac{1}{s(p)}$. Clearly, the higher the rate of a data source, the more the number of packets it will generate in each time interval, and the smaller the fraction of the flow each packet will represent. Numerically, the larger the value of $n(x)$, and the smaller the fraction $\frac{1}{s(p)}$ will be.

Each node x measures the total number $F(y, x)$ of flows passing through each of its upstream links (y, x), $\forall y \in U_x$.

Let $F_{new}(y,x)$ be the measurement made in a new time interval, and $P(y,x)$ be the set of packets received by x from y during the time interval.

$$F_{new}(y,x) = \sum_{p \in P(y,x)} \frac{1}{s(p)}$$

$F(y,x)$ is updated as an exponential moving average.

$$F(y,x) \Leftarrow (1-\alpha) \times F(y,x) + \alpha \times F_{new}(y,x) \qquad (2)$$

C. Rate Limit

The proposed multipath fairness solution (MFS) performs congestion avoidance [17], which prevents an upstream neighbor from transmitting to a sensor whose buffer is full. On top of that, MFS sets rate limits to achieve flow-level end-to-end fairness.

When sensor x is congested either due to full buffer or heavy collision, it computes a rate limit $l(y,x)$ for each upstream neighbor y as follows:

$$l(y,x) = \frac{F(y,x)}{I(x) + \sum_{z \in U_x} F(z,x)} r(x) \qquad (3)$$

where $r(x)$ is the rate at which x forwards packets downstream, which can be measured locally, and $I(x) = 1$ if x is a data source or $I(x) = 0$ otherwise. Sensor x advertises $l(y,x)$ to y, possibly by piggybacking in an ACK packet to y. Sensor y enforces the rate limit such that the actual rate on the link (y,x) is bounded by the limit. If x itself is a data source, it will assign a local rate limit as follows

$$l(x) = \frac{1}{1 + \sum_{z \in U_x} F(z,x)} r(x) \qquad (4)$$

After an upstream neighbor enforces a rate limit, it may become congested because it now sends less. If its buffer is kept full, it will enforce rate limits in a similar way on its upstream neighbors. This process repeats towards the data sources. Eventually all affected data sources will adjust packet rates according to their fair bandwidth shares allowed by the rate limits. The rate limits set by all congested sensors collectively ensure a fair distribution of network bandwidth to the data sources. Note that *only* congested sensors enforce rate limits, which are updated periodically.

After a congested sensor x enforces rate limits on upstream links, the congestion condition may change either because some upstream data sources cease to produce data as the triggering events are gone, or because some downstream sensors gain additional bandwidth due to the dynamics of environmental signal interference. When x is able to forward more than it receives, its packet queue will clear up. In this case, x will artificially increase the rate limits to use up the available bandwidth. In our simulations, when the length of the packet queue at x is below a threshold, the upstream rate limit is set to be $1.1 \times l(y,x)$, where $l(y,x)$ is computed by (3), and the local rate limit is set to be $1.1 \times l(x)$, where $l(x)$ computed by (4). When the buffer space at x becomes almost empty, the upstream rate limit is overset to be $2 \times l(y,x)$ and the local rate limit is set to be $2 \times l(x)$. Increased rate limits may increase $r(x)$ (the forwarding rate of x), which in turn further increases the rate limits when (3) and (4) are computed next time.

The increase of rate limits will stop when the congestion condition returns to x. The former case happens when the rate limits are raised too high and exceed the maximum rate that x is able to send downstream.[1] Sensor x learns the return of congestion by observing the buffer's fullness, and when that happens, it sets the rate limits to be $l(y,x)$ and $l(x)$, instead of over-setting them.

A node stop enforcing rate limits if its local congestion condition is removed (as its buffer is emptied) and the rates at which it receives packets from the upstream nodes are smaller than the rate limits.

D. Weighted Fairness

Some sensors may require higher packet rates than others. We use weighted fairness to handle this issue. Each sensor x is assigned a weight $w(x)$ that is equal to or greater than one. A new k-bit *weight field* is added to the packet header. It stores the weight of the data source that generates the packet. Let $w(p)$ be the weight of the data source that produces a packet p. The packet represents a *weighted fraction* of the flow generated from the data source. The weighted fraction is defined as $\frac{w(p)}{s(p)}$. The rest of MFS stays the same except that the formulas for $F_{new}(y,x)$, $l(y,x)$, and $l(x)$ are changed as follows:

$$\begin{aligned} F_{new}(y,x) &= \sum_{p \in P(y,x)} \frac{w(p)}{s(p)} \\ l(y,x) &= \frac{F(y,x)}{w(x) + \sum_{z \in U_x} F(z,x)} r(x) \qquad (5) \\ l(x) &= \frac{w(x)}{w(x) + \sum_{z \in U_x} F(z,x)} r(x) \end{aligned}$$

Consider the set P of packets generated by a data source x during a time interval. The weighted fraction of the flow represented by each packet p is $\frac{w(p)}{s(p)} = \frac{w(x)}{n(x)}$. Because $n(x)$ packets are expected to be generated in a time interval, the summation of the weighted fractions of all packets in P is expected to be $w(x)$. The packets will acquire network bandwidth proportional to their weighted fractions along the routing paths to the base stations. Effectively, the source will receive an aggregated bandwidth from itself to the base stations in proportion to $w(x)$.

If a data source has a minimum rate requirement and finds that its actual packet rate is below the requirement, it can dynamically increase its weight. It is possible that the combined minimum requirements of the sensors exceed the total downstream capacity of the network. In that case, it is impossible to satisfy all sensors. The weight increase should stop when the weight reaches a threshold value.

---

[1] Other than filling x's buffer, no harm will be done because congestion avoidance [17] will prevent packet drops by restricting upstream neighbors from overflowing x's buffer.

E. Optimizations

Various optimizations can be made to improve the performance of the above fairness solution. Suppose each node x periodically computes its *weighted mean rate*, $m(x) = \frac{r(x)}{w(x)+\sum_{z \in U_x} F(z,x)}$, and advertises this value to the neighbors. It also learns the weighted mean rates from its neighbors.

- When x has a packet to forward, if there are two downstream neighbors and the rate limits to them are not violated, x always forwards the packet to the neighbor with the largest weighted mean rate.
- When x is congested and computes a rate limit $l(y,x)$ for an upstream neighbor y by (5), if $m(y)$ is smaller than $m(x)$, x will increase the rate limit $l(y,x)$ by a factor of $\frac{m(x)}{m(y)}$, which allows y to send more traffic to x in an effort to equalize $m(y)$ and $m(x)$.
- If x has the smallest weighted mean rate in the neighborhood, it may reduce its minimum congestion window to acquire a large portion of the channel capacity. On the other hand, if it has the largest weighted mean rate, it may increase its minimum congestion window to give up some bandwidth. Different optimization schemes can be designed based on this idea. The one used in our simulations is described below. We define the range of window variations to be

$$range = \max\{\frac{m_{max} - m_{min}}{\Omega}, 1\}$$

where m_{max} is the largest weighted mean rate in the neighborhood, m_{min} is the smallest weighted mean rate in the neighborhood, and Ω is the largest allowed rate of any data source. The minimum congestion window is adjusted by the following formula.

$$factor = 1 - \frac{range}{2} + \frac{m(x) - m_{min}}{m_{max} - m_{min}} range$$
$$min\_cong\_win = def\_win \times factor$$

where def_win is the default minimum congestion window.

IV. EVALUATION

A. Simulation Setup

We implemented a packet-level simulation testbed based on CSMA/CA with RTS/CTS/DATA/ACK exchange. The simulation setup is given as follows: 500 sensors are randomly placed in a 1000×1000 area. 8 base stations are evenly spaced along one edge of the deployment area. The transmission range of the sensors is 100. The transmission rate is 512 kilobits per unit of time. There are 100 data sources randomly selected from the 500 sensors. A data source can generate packets up to 20 per unit of time. However, the actual rate will be lower if there is congestion downstream. Carrier sensing, radio collision, collision avoidance, and exponential random backoff are implemented based on 802.11 DCF. Each data packet is 30 bytes long. Each control packet (RTS/CTS/ACK) is 3 bytes long. The buffer at each sensor can hold 30 data packets. The flow size field is 5 bits long, and the weight field is 2 bits long.

B. Simulation Results

The first set of simulations evaluates how well MFS and other solutions achieve fairness. The *delivered packet rate* (or packet rate for brevity) is the number of packets that are successfully delivered from a data source to the sink per unit of time.

Figure 2 shows the delivered packet rates of 100 data sources under CSMA/CA with multipath routing. The rates are widely distributed between zero to 20 packets per unit of time. End-to-end fairness is not achieved. Figure 3 shows that the single-path fairness solution (SFS) [9] improves fairness over CSMA/CA, however, at the expense of reduced average packet rate due to the restriction of single-path routing.

Figure 4 shows the delivered packet rates of 100 data sources when MFS is used. Comparing with Figure 3, MFS achieves much higher average rate per flow. Remarkably, it also considerably improves fairness over SFS. That is because MFS is more capable of redistributing the bandwidth from high-rate sources to low-rate sources. With multiple routing paths per source in MFS, the paths of a low-rate source have better chance to come across the paths of a high-rate source, allowing the former to acquire the bandwidth of the latter (which is implemented through appropriate rate limits that are set in proportion to the number of passing flows).

Next we demonstrate how well MFS achieves weighted fairness. In Figure 5, data sources 1-50 have weight two; their average delivered packet rate is 16.99. Sources 51-100 have weight one; their average delivered packet rate is 9.56. The results show that, roughly, MFS allocates bandwidth to the flows proportional to their weights. Note that a few sources with weight one have larger rates than some sources with weight two. The reason is that these weight-one sources do not share common routing paths with the weight-two sources. When two nodes share a common congested routing path, the bandwidth is allocated based on weights. However, if a source with weight one has an uncongested routing path while another source with weight two does not, the former will have higher rate than the latter.

In Figure 6, data sources 1-50 have weight four; their average delivered packet rate is 19.47. Sources 51-100 have weight one; their average delivered packet rate is 6.81. The ratio of 19.47 : 6.81 is not exactly 4 : 1 ratio simply because most of the data sources 1-50 are already sending at the maximum rate of 20 packets per unit of time.

Figure 7 has three different weights. Data sources 1-33 have weight three; their average delivered packet rate is 18.28. Data sources 34-66 have weight one; their average delivered packet rate is 7.79. Data sources 67-100 have weight two; their average delivered packet rate is 12.44. Note that the average rates for data sources with weight one or two are pulled up by a few outliers whose routing paths happen to have less contention. If we look at the average rates among the majority data sources whose rates are clustered together, then the ratio of the average rates will be closer to the ratio of weights.

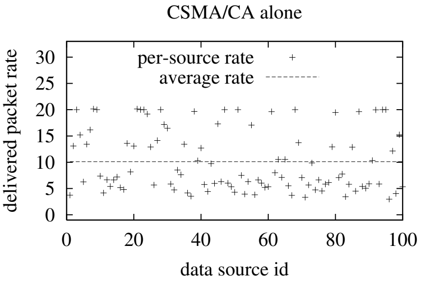

Fig. 2. Under CSMA/CA, the packet rates of 100 data sources are widely distributed between zero and 20 packets per unit of time.

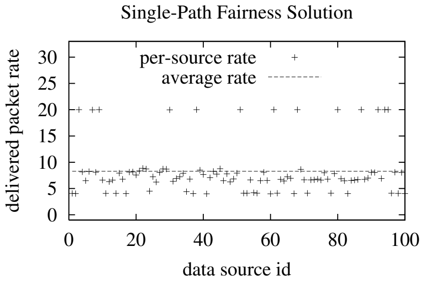

Fig. 3. The single-path fairness solution achieves fairness at reduced rates.

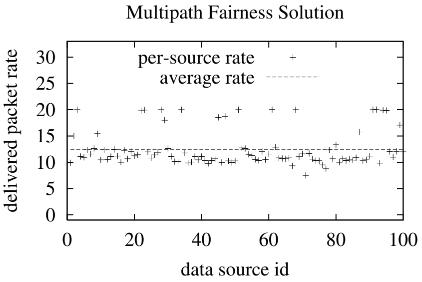

Fig. 4. The multipath fairness solution achieves both fairness and high packet rates.

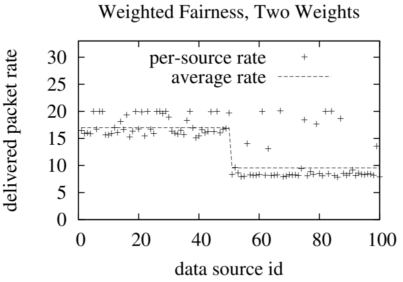

Fig. 5. Data sources 1-50 have weight two. Data sources 51-100 have weight one.

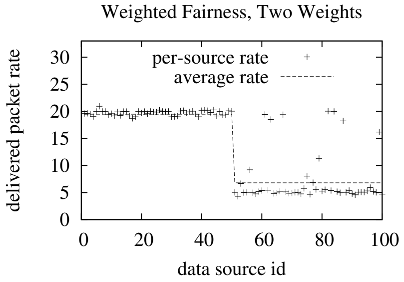

Fig. 6. Data sources 1-50 have weight four. Data sources 51-100 have weight one.

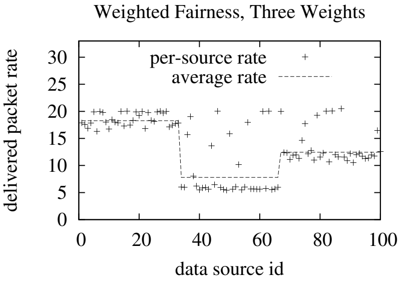

Fig. 7. Data sources 1-33 have weight three. Data sources 34-66 have weight one. Data sources 67-100 have weight two.

V. CONCLUSION

This paper studies the flow-level fairness problem in a data-collection sensor network where the flow of packets generated from a data source may be routed along numerous different paths to the base stations. We propose a multipath fairness solution that redistributes network bandwidth from high-rate data sources to low-rate sources as long as they share common routing paths. The solution is fully distributed and easy to implement. Each forwarding node performs only localized operations and does not maintain per-flow state. Collectively the forwarding nodes' operations achieve global fairness at the flow level. The proposed solution also supports weighted fairness, which allows important data sources to acquire larger shares of network bandwidth than other sources.

REFERENCES

[1] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks," *Proc. of ACM SenSys'03*, November 2003.

[2] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," *Proc. of ACM MobiCom'00*, August 2000.

[3] B. Bensaou, Y. Wang, and C. C. Ko, "Fair medium access in 802.11 based wireless ad-hoc networks," *Proc. of ACM MobiHoc'00, Boston, Massachusetts*, July 2000.

[4] X. L. Huang and B. Bensaou, "On Max-Min Fairness and Scheduling in Wireless Ad-hoc Networks: Analytical Framework and Implementation," *Proc. of ACM MobiHoc'01, Long Beach, California*, October 2001.

[5] H. Luo, J. Cheng, and S. Lu, "Self-Coordinating Localized Fair Queueing in Wireless Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, vol. 3, no. 1, 2004.

[6] Y. Sankarasubramaniam, O. Akan, and I. Akyildiz, "ESRT: Event-to-Sink Reliable Transport in Wireless Sensor Networks," *Proc. of ACM MobiHoc'03*, June 2003.

[7] Y. Yi and S. Shakkottai, "Hop-by-Hop Congestion Control over a Wireless Multi-hop Network," *Proc. of IEEE INFOCOM'04, Hong Kong, China*, March 2004.

[8] A. Sridharan and B. Krishnamachari, "Max-Min Fair Collision-Free Scheduling for Wireless Sensor Networks," *Proc. of Workshop on Multihop Wireless Networks (MWN'04)*, April 2004.

[9] C. T. Ee and R. Bajcsy, "Congestion Control and Fairness for Many-to-One Routing in Sensor Networks," *Proc. of ACM SenSys'04*, November 2004.

[10] B. Hull, K. Jamieson, and H. Balakrishnan, "Mitigating Congestion in Wireless Sensor Networks," *Proc. of ACM SenSys'04*, November 2004.

[11] A. Woo and D. Culler, "A Transmission Control Scheme for Media Access in Sensor Networks," *Proc. of ACM MobiCom'01*, July 2001.

[12] B. Li, "End-to-End Fair Bandwidth Allocation in Multi-hop Wireless Ad Hoc Networks," *Proc. of IEEE ICDCS'05*, June 2005.

[13] S. Sarkar and L. Tassiulas, "End-to-end Bandwidth Guarantees Through Fair Local Spectrum Share in Wireless Adhoc Networks," *IEEE Transactions on Automatic Control*, vol. 50, no. 9, September 2005.

[14] V. Gambiroza, B. Sadeghi, and E. W. Knightly, "End-to-End Performance and Fairness in Multihop Wireless Backhaul Networks," *Proc. of ACM Mobicom'04, Philadelphia, PA, USA*, September-October 2004.

[15] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with Guaranteed Delivery in Ad Hoc Wireless Networks," *Proc. of 3rd Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DialM'99)*, August 1999.

[16] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for Wireless Networks," *Proc. of ACM MobiCom'00*, August 2000.

[17] S. Chen and N. Yang, "Congestion Avoidance based on Light-Weight Buffer Management in Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems, Special Issue on Localized Communication and Topology Protocols for Ad Hoc Networks*, vol. 17, no. 9, September 2006.