Lexicographic Maxmin Fairness for Data Collection in Wireless Sensor Networks

Shigang Chen, Yuguang Fang, *Senior Member*, *IEEE*, and Ye Xia

Abstract—The ad hoc deployment of a sensor network causes unpredictable patterns of connectivity and varied node density, resulting in uneven bandwidth provisioning on the forwarding paths. When congestion happens, some sensors may have to reduce their data rates. It is an interesting but difficult problem to determine which sensors must reduce rates and how much they should reduce. This paper attempts to answer a fundamental question about congestion resolution: What are the maximum rates at which the individual sensors can produce data without causing *congestion* in the network and *unfairness* among the peers? We define the *maxmin optimal rate assignment* problem in a sensor network, where all possible forwarding paths are considered. We provide an iterative linear programming solution, which finds the maxmin optimal rate assignment and a forwarding schedule that implements the assignment in a low-rate sensor network. We prove that there is one and only one such assignment for a given configuration of the sensor network. We also study the variants of the maxmin fairness problem in sensor networks.

Index Terms—Multipath maxmin fairness, wireless sensor networks, data collection applications, iterative linear programming.

1 Introduction

Sensor networks have a wide range of applications in habitat observation [1], [2], health monitoring [3], object tracking [4], [5], battlefield sensing, etc. They are different from traditional wireless networks in many aspects [6]. Particularly, sensor nodes are limited in computation capability, memory space, communication bandwidth, and above all, energy supply. Intense study was carried out in recent years on the physical layer [7], [8], the MAC layer [9], [10], [11], and the network layer [12], [13], [14], [15], [16], [17].

The ad hoc nature of sensor deployment leads to unpredictable patterns of connectivity and varied node density, which causes uneven bandwidth provisioning on the forwarding paths. The data sources are often clustered at sensitive areas under scrutiny and may take similar paths to the base stations. When data converge toward a base station, congestion may occur at sensors that receive more data than they can forward. This paper primarily studies sensor networks that continuously collect data from a field for a very long time, which requires the sensors to operate at low rates. For most of the paper, we assume that, due to the lifetime requirement, the maximum forwarding rate of each sensor is set sufficiently low that the media contention is insignificant. A sensor is congested if it receives more traffic than its maximum forwarding rate. This assumption will later be removed and the implication of media contention will be discussed.

- *S. Chen and Y. Xia are with the Department of Computer and Information Science and Engineering, University of Florida, Gainesville, FL 32611. E-mail: {sgchen, yx1}@cise.ufl.edu.*
- *Y. Fang is with the Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611. E-mail: fang@ece.ufl.edu.*

Manuscript received 3 Nov. 2005; revised 19 June 2006; accepted 17 Oct. 2006; published online 7 Feb. 2007.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-0318-1105. Digital Object Identifier no. 10.1109/TMC.2007.1039.

Congestion causes many problems. When a packet is dropped, the energy spent by upstream sensors on the packet is wasted. The further the packet has traveled, the greater the waste. When a sensor x is severely congested, if the upstream neighbors attempt to send to x, their efforts (and energy) are deemed to be wasted and, worse yet, counter-productive because they compete for channel access with neighboring sensors. Finally, and above all, the data loss due to congestion may jeopardize the mission of the application. While fusion techniques [13] can be used for data aggregation, applications may require some specifics (e.g., exact locations of the reporting sensors) to be kept [6], which places a limit on how much the fusion can do.

The problem of congestion control in sensor networks is largely open [18]. A typical approach is for a congested sensor to send backpressure messages to its neighbors [18], which reduce their data rates and may further propagate the backpressure messages upstream. However, the important issue of ensuring fairness among the sensors during their rate reduction is not addressed by this approach. In ESRT [19], by monitoring the congestion notification bit carried in the packet header, the base station decides a common rate for all sensors such that no packet will be lost in the network. This approach achieves fairness but is too pessimistic because every sensor must conform its rate to the worst rate in the most congested area. Directed Diffusion [12] and SPEED [20] were not specifically designed for congestion control, but they may be adapted for this purpose to a certain degree.

This paper attempts to answer a fundamental question: What are the maximum rates at which the individual sensors can produce data without causing congestion in the network and unfairness among the peers? We define the lexicographic maxmin fairness problem for data collection in sensor networks. We prove that the maxmin rate assignment is unique and analyze an array of properties of such an assignment. We demonstrate that, although it is much harder than the classical maxmin flow-control problem [21], the maxmin rate assignment in low-rate sensor networks can be determined by iteratively solving two types of linear programs, which accumulatively find the maxmin assignment together with a congestion-free forwarding schedule that implements the assignment. To the best of our knowledge, this is the first work that solves the maxmin rate assignment problem for end-to-end flows without fixed routing paths in wireless sensor networks. We also discuss the implications caused by media contention. Overall, the paper lays out a theoretical foundation for studying fairness at the network layer in sensor networks and provides a benchmark solution against which the future distributed heuristic algorithms can be compared and evaluated.

The rest of the paper is organized as follows: Section 2 discusses the related work. Section 3 defines the network model, the notations, and the maxmin rate assignment problem. Section 4 describes an algorithm that solves a series of linear programs and finds the maxmin assignment and the forwarding schedule that implements the assignment. Section 5 takes media contention into consideration. Section 6 provides theoretical coverage on maxmin assignment with edge or mixed capacities. Section 7 addresses the weighted maxmin assignment. Section 8 draws the conclusion.

2 Maxmin Fairness and Related Work

The maxmin flow control was first proposed by Faffe [22] to distribute the network bandwidth fairly among a set of best-effort flows. The name *maxmin* comes from the strategy of *maximizing* the bandwidth allocated to those flows that receive the *minimum* bandwidth. Much further research [23], [24], [25], [26], [27], [28] has been done since then. All these works assume that each flow has a fixed routing path. Two basic properties of the maxmin flow control are:

1. *Fairness property*. At each link, any passing flow is entitled to an equal share of the link capacity unless the flow is limited to a smaller bandwidth at another link on its path.
2. *Maximum throughput property*. The entire capacity of a link must be allocated to the flows unless every passing flow has a bottleneck link elsewhere which limits the bandwidth that the flow can receive.

A bottleneck algorithm that assigns the maxmin bandwidth to every flow was described in [21], [25], [29] and is repeated here: Find the global bottleneck link that has the smallest bandwidth per flow. Assign an equal share of the link's capacity to each passing flow. Remove the link and the passing flows from the network. When a flow is removed, the capacities of all links on its routing path are reduced by the bandwidth assigned to the flow. Repeat the above process until every flow is assigned a bandwidth and removed from the network.

A wireless network has different properties than a wired network. For example, the capacity of a wireless link between two neighbors is not fixed but depends on the amount of background communication in the neighborhood. MAC-layer fairness in wireless networks was studied in [30], [31], [32], [33], [34]. Flow-level proportional fairness in FDMA/CDMA networks was studied in [35] under the assumption that each flow has a single routing path. Maxmin fairness among one-hop flows in FDMA/CDMA networks was studied in [36]. Fairness in TDMA networks was studied in [37] under the assumption of a tree routing structure from all data sources to a sink. The TAP fairness in wireless backhaul networks was investigated in [38], which achieves temporal fairness instead of throughput fairness. However, the paper does not provide an algorithm that computes the TAP rates.

For data collection in a sensor network, the packet flow generated by a sensor may take many possible paths (instead of a fixed one) to reach the base stations, which makes the maxmin assignment problem considerably harder because it is no longer clear where and which flows compete for resource. The number of possible routing paths for a flow can be exponential with respect to the distance between the source and the destination. At a forwarding node, the bandwidth should no longer be evenly divided among the passing flows because some flows may receive bandwidth from other paths while others may not. The maxmin fairness in such a context is what we will study here.

It is well known that proportional or temporal fairness is more appropriate in a *multirate* wireless network [39], [40], [41], [38], where maxmin fairness may cause severe throughput degradation. That is NOT the case for a *single-rate* wireless sensor network, which is the subject of this paper. We study the sensor networks whose transceivers operate at a single transmission rate.

In summary, the maxmin problem investigated by this paper has not been solved by the above referenced works. The technique of iterative linear programming that we will use to solve the problem did not appear in these works either.

3 Network Model and Problem Definition

3.1 Sensor Network

A sensor network consists of a number of sensors and a number of base stations. Two sensors are neighbors if they can directly communicate with each other. Different from the multirate wireless networks [39], [40], [41], [38], we assume that the transceivers of the sensors operate at a *single transmission rate* M, which is reasonable due to the cheap design requirement for inexpensive one-time sensors that are used in large quantities. Hence, the transmission rate is not modeled as a changing quantity, e.g., a function of the signal/noise ratio. If radio interference is too severe, two nodes cease to be neighbors. M is also called *media throughput*.

The sensors share the same wireless media and each packet is transmitted as a local broadcast in the neighborhood. We assume the existence of a MAC protocol, which ensures that, among the neighbors in the local broadcast range, only the intended receiver keeps the packet and the other neighbors discard the packet. The sensors are statically located after deployment. We do not consider mobile sensors that form a dynamic ad hoc network. We study data packets sent from sensors to base stations. The base stations are connected via an external network to a data collection center. A data packet may be sent to any base station as long as there is a forwarding path.

3.2 Some Notations

Let N be the set of sensors, N_x be the set of neighbors of a sensor x, and E be the set of directed communication links between neighbors. $E = \{(x, y) \mid x \in N, y \in N_x\}$.

At any moment, a sensor is in one of three states: transmitting, receiving, or staying idle. The actual rate at which the sensor forwards data, called the *forwarding rate*, is determined by the fraction of time during which it transmits. We are only concerned with the forwarding rate in a steady (nontransitional) state. Based on a lifetime requirement, the residual battery power, and the amount of energy for transmitting a packet, a sensor x can calculate its maximum allowed forwarding rate, denoted as T_x, which may be far below M due to the need for conserving energy in order to achieve the desired lifetime. The maximum rate T_x can be enforced by a token-bucket algorithm which either transmits or waits for tokens that are released at the rate of T_x. We assume T_x is small enough that media contention is insignificant and thus ignored. Such a sensor network is called a *low-rate sensor network*. For example, suppose the sensors are able to transmit at hundreds of thousands of bits per second. However, if the application requires them to continuously collect data for weeks or even months, the limited power supply may only allow the sensors to send at tens of bytes per second in order to last for such a long period. At such low rates, media contention is not a serious issue.

While this paper mainly focuses on low-rate networks, media contention will be addressed in Section 5. For now, we ignore the energy expenditure for receiving packets, which will be addressed in Section 4.5.

Let L_x be the rate at which a sensor x *generates* new data packets (to be delivered to the base stations). If $L_x = 0$, x is called an *inactive sensor*. If $L_x > 0$, it is called an *active sensor*. Let A be the set of active sensors. No matter whether a sensor is active or not, it will forward packets from other sensors.

An application may set a maximum rate Ω at which an active sensor will generate new data. Consider an environmental monitoring example, where sensors are configured to report temperature once per minute (Ω). On one hand, even when the bandwidth of a forwarding path allows a sensor to send 1,000 measurements per minute, the sensor will still report at the rate of Ω, which is preconfigured based on the application requirement. On the other hand, when there is congestion, a sensor may be forced to generate data at a rate smaller than Ω. Consequently, $L_x \leq \Omega$.

How large can $L_x, x \in A,$ be without causing congestion in the network and unfairness among peers? Below, we try to precisely model this problem.

3.3 Congestion-Free Forwarding Schedule

A *forwarding schedule* defines, for each sensor $x \in N$, the set of upstream neighbors U_x that send data to x, the set of downstream neighbors D_x that receive data from x, and the rates at which these neighbors send (receive) data to (from) x.[1] Obviously, if $x \in D_y$, then $y \in U_x$ and vice versa. If $x \in D_y$, we say there exists a *forwarding link* (y, x). The forwarding rate on (y, x) is denoted as $f(y, x)$, which is a positive real number.[2] All forwarding links form the *forwarding graph*. A sensor may have numerous paths in the forwarding graph to reach the base stations.

We can transform any forwarding graph to an acyclic one by the following procedure: Identify a forwarding loop by depth-first search. Deduct the forwarding rates along the loop by the bottleneck rate. Delete the bottleneck link whose rate becomes zero, which breaks the loop. Repeat the above process until all loops are removed. In the sequel, we assume a forwarding graph to be acyclic. In an acyclic graph, any path leads to a sink (i.e., base station).

The total rate received by x is $R_x = \Sigma_{y \in U_x} f(y, x)$. The total rate sent out by x is $\Sigma_{y \in D_x} f(x, y)$, which is bounded both by the maximum forwarding rate T_x and by the sum of the locally generated data rate L_x and the received rate R_x. Therefore, $\Sigma_{y \in D_x} f(x, y) = \min\{T_x, L_x + R_x\}$. A sensor x is *congested* if $L_x + R_x > T_x$, which means it receives more than it can forward.[3] x is *saturated* if $L_x + R_x = T_x$, which means any additional input will cause congestion at x. A forwarding schedule is *congestion-free* if the condition $L_x + R_x \leq T_x$ holds for all $x \in N$.

We call $L_A = \{L_x \mid x \in A\}$ a *rate assignment* for the active sensors. One should treat L_A as a function that maps A to real numbers, even though we often use it as a set for the purpose of convenience. A rate assignment is *feasible* if there exists a congestion-free forwarding schedule that routes all generated data to the base stations. Our goal is to find the best feasible rate assignment (as well as its congestion-free forwarding schedule) based on certain fairness/throughput criteria. Specifically, we consider the maxmin optimality.

3.4 Lexicographic Maxmin Rate Assignment in Sensor Networks

If a sensor receives more data than it can forward, congestion will occur and the sensor will have to drop the excess packets. In order to avoid congestion, the upstream sensors must redirect packets to other paths. If all forwarding paths from an active sensor to the base stations are congested, the sensor must generate data at a reduced rate. The problem becomes interesting when many active sensors share forwarding paths in an arbitrary way. For each active sensor, we want to find the highest possible rate that does not cause congestion at a downstream node. We also want all active sensors to have equal access to the transmission capacity of the network, no matter how different their forwarding paths are.

Consider an example in Fig. 1. Suppose $T_x = T_y = T_z = T_w = 1$, which are the maximum forwarding rates of the nodes, respectively. We assume that these lifetime-constrained rates are well below the media

1. A forwarding schedule is a network-layer rate assignment to the links. It is a different concept from the MAC-layer time-slot schedule.

2. An upstream node y does not continuously send to x at a rate of $f(y, x)$. It sends at rate M when it sends, but it does not send at all time. The average rate is $f(y, x)$.

3. Congestion does not necessarily mean that some sensors behave too aggressively. A large number of upstream sensors, each producing data at a small rate, may lead to congestion at a downstream sensor.

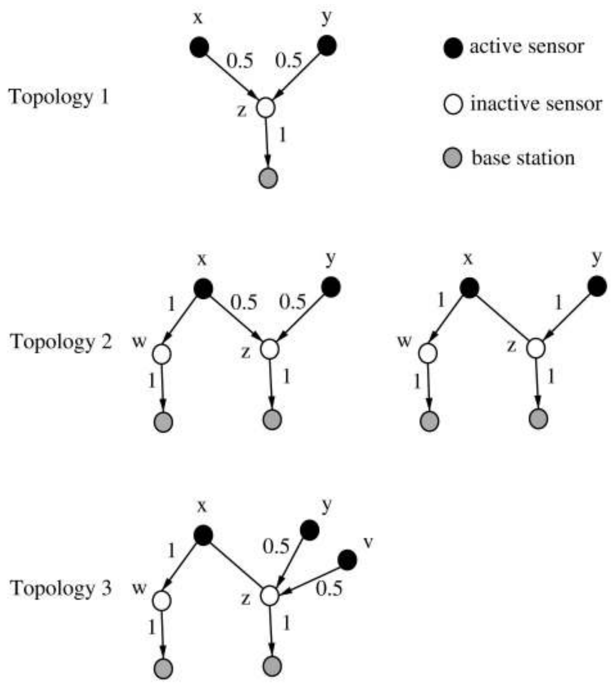


Fig. 1. The fairness property as described in Section 2 is no longer valid when there are multiple forwarding paths.

throughput. Hence, they can all be achieved due to insignificant media contention. Note that there are no fixed link capacities in a wireless network. In Topology 1, the active sensors x and y are entitled to an equal share of z's forwarding rate. But that is not true in Topology 2 because x has an additional forwarding path to a base station. If x still forwards data to z at a rate of 0.5, then it will generate data at a total rate of 1.5 while y can only generate at a rate of 0.5. Therefore, to be fair, x should not forward any data to z such that x and y will generate data at an equal rate of 1. In another twist of this example, if $T_w = 0.6$, then x should forward data to z at a rate of 0.2 such that both x and y will generate data at an equal rate of 0.8. It is not always possible for every active sensor to generate data at the same rate. In Topology 3, while x takes its own forwarding path, y and v must share the maximum forwarding rate of their common bottleneck z.

Definition 1. *Consider two feasible rate assignments, $L_A = \{L_x \mid x \in A\}$ and $L'_A = \{L'_x \mid x \in A\}$. Suppose we sort both in ascending order and treat them as ascending rate vectors. We define the following relations: 1) $L_A = L'_A$ if the two vectors are identical and 2) $L_A > L'_A$ if there exists a prefix $(r_1, r_2, \dots, r_i)$ of the sorted vector L_A and a prefix $(r'_1, r'_2, \dots, r'_i)$ of the sorted vector L'_A such that $r_i > r'_i$ and $r_j = r'_j, 1 \leq j \leq i-1$.*

The "=" operator defines *equivalent groups*, each containing a set of feasible assignments that equal one another. The ">" operator places a total lexicographic order on the equivalent groups. This ordering takes both fairness and throughput into account. In more descriptive but less precise words, a feasible assignment is "greater" than another if it is fairer or generates more throughput [42]. The "greatest" feasible assignment first maximizes the data rates from the active sensors that produce the least amount of new data, then maximizes the rates from the sensors that produce the second least amount of new data, and so on.

Definition 2. *The greatest feasible assignment $L^m_A = \{L^m_x \mid x \in A\}$ is called the* lexicographic maxmin optimal rate assignment[4] *of node x (in short, maxmin assignment). Namely, $L^m_A \geq L_A$ for any feasible assignment L_A. $\forall x \in A$, L^m_x is called the* maxmin rate *of node x.*

Due to the "=" operator, there might be multiple different maxmin assignments which are identical as sorted rate vectors but differ in the rates of specific sensors. However, the following theorem shows that the maxmin assignment is unique—there is one and only one largest rate assignment. Our goal is to find this maxmin assignment and a congestion-free forwarding schedule that implements the maxmin assignment.

Theorem 1. *There exists one and only one maxmin optimal rate assignment.*

The proof of the theorem can be found in Appendix A. Many properties of the maxmin assignment, given in the form of lemmas, are also proved in Appendix A.

We want to stress that maxmin fairness is not a maximum flow problem, which is to maximize a linear objective function $\sum_{x \in A} L_x$ without any fairness concern on individual sensors. Many rate assignments can produce a maximum flow. But, the maxmin rate assignment is unique; it has a nonlinear objective and cannot be solved by a single linear program.

Notations defined in this section are listed in Table 1 for quick reference.

4 Finding Maxmin Optimal Rate Assignment

We first solve two related problems, the *Maximum Common Rate Problem* and the *Maximum Single Rate Problem*. Based on the solutions to these problems, we will design an algorithm that finds the maxmin assignment and its forwarding schedule.

4.1 Maxmin Subset and Maxmin Subassignment

Given a real number r, the *maxmin subset* of A with respect to r is defined as

$$A(r) = \{x \mid L^m_x \leq r, x \in A\}.$$

It is the subset of active sensors whose maxmin rates are equal to or smaller than r. The corresponding maxmin subassignment with respect to r is defined as

$$L^m_A(r) = \{L^m_x \mid x \in A(r)\}.$$

We give an example in Fig. 2 to illustrate how $A(r)$ and $L^m_A(r)$ are defined. For instance,

$$A(1/2) = \{u, v, w, x, y\}$$

4. The classical definition of maxmin fairness is that any rate increase for one flow will cause the rate decrease for another flow that already has a smaller rate. We choose the lexicographic definition, which fits better in our context. These two definitions are equivalent for most but all cases.

TABLE 1
Notations

N	sensors in the network
E	communication links in the network
A	active sensors in the network
N_x	neighbors of a sensor x, $\forall x \in N$
T_x	maximum forwarding rate of a sensor x
Ω	maximum rate at which an active sensor will generate new data to be delivered to the base stations
L_x	actual rate at which a sensor x generates new data, $L_x \leq \Omega$
U_x	upstream neighbors of a sensor x in a forwarding schedule
D_x	downstream neighbors of a sensor x in a forwarding schedule
$f(y,x)$	forwarding rate on a wireless link (y,x)
L_A	a rate assignment, $\{L_x \mid x \in A\}$
L_A^m	maxmin optimal rate assignment
L_x^m	maxmin rate of an active sensor x

because the maxmin rates of these sensors are equal to or smaller than $1/2$. Sensor z does not belong to $A(1/2)$ because its maxmin rate is 1. In the rest of this section, when we say $L_A^m(r)$ is known, we imply that $A(r)$ is also known.

4.2 Maximum Common Rate Problem (MCR)

Definition 3. *Maximum Common Rate Problem (MCR): Suppose $L_A^m(r)$ is known for a real number r. Let the sensors in $A(r)$ take their maxmin rates and the sensors in $A - A(r)$ take a common rate. The problem is to find the maximum common rate $C(r)$ that can be realized by a congestion-free forwarding schedule.*

$C(r)$ is a function of r. We have the following lemmas. The proofs for the lemmas and the theorem in this section can be found in Appendix B.

Lemma 1. *$C(0)$ is the smallest maxmin rate in L_A^m.*

Lemma 2. *Let r be a maxmin rate in L_A^m. $C(r)$ is the smallest maxmin rate in L_A^m that is greater than r.*

Beginning with $r = 0$ and solving MCR for the maximum common rate $C(0)$, we find the smallest maxmin rate in L_A^m. By iteratively solving MCR with r being $C(r)$ of the previous round, we will find all maxmin rates in L_A^m.

We illustrate the above property in Fig. 3, which continues the example in Fig. 2, where there are three distinct maxmin rates, $1/3$, $1/2$, and 1. First, with the knowledge of $L_A^m(0)$ (which is an empty set), by solving MCR, we get the maximum common rate $C(0) = 1/3$, as shown in Fig. 3a. $1/3$ is the smallest maxmin rate. Second, with the knowledge of $L_A^m(1/3)$, by solving MCR, we get

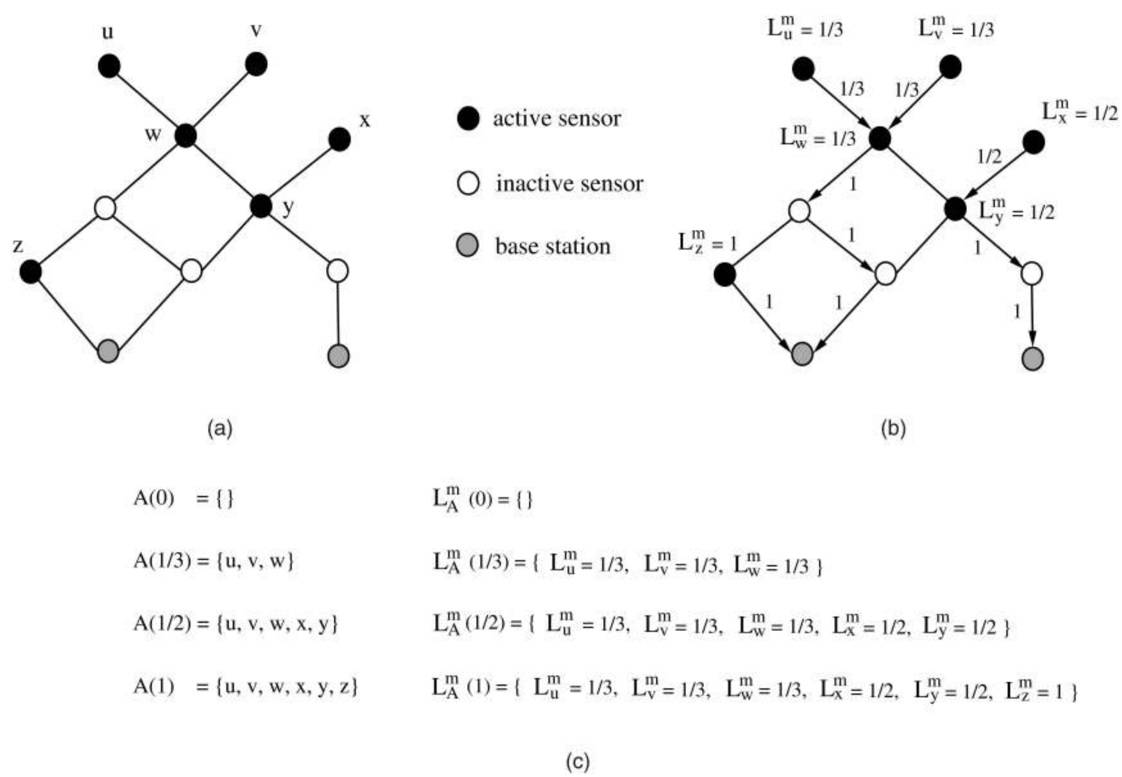


Fig. 2. Assume the maximum forwarding rates of all sensors are one. (a) Network tpology. (b) Maxmin rate assignment and congestion-free forwarding schedule. The maxmin rates are shown beside the sensors. The number beside a link is the link's forwarding rate in the forwarding schedule that implements the maxmin assignment. (c) Maxmin subsets, $A(0), A(1/3), A(1/2), A(1)$, and the corresponding maxmin subassignments, $L_A^m(0), L_A^m(1/3), L_A^m(1/2), L_A^m(1)$.

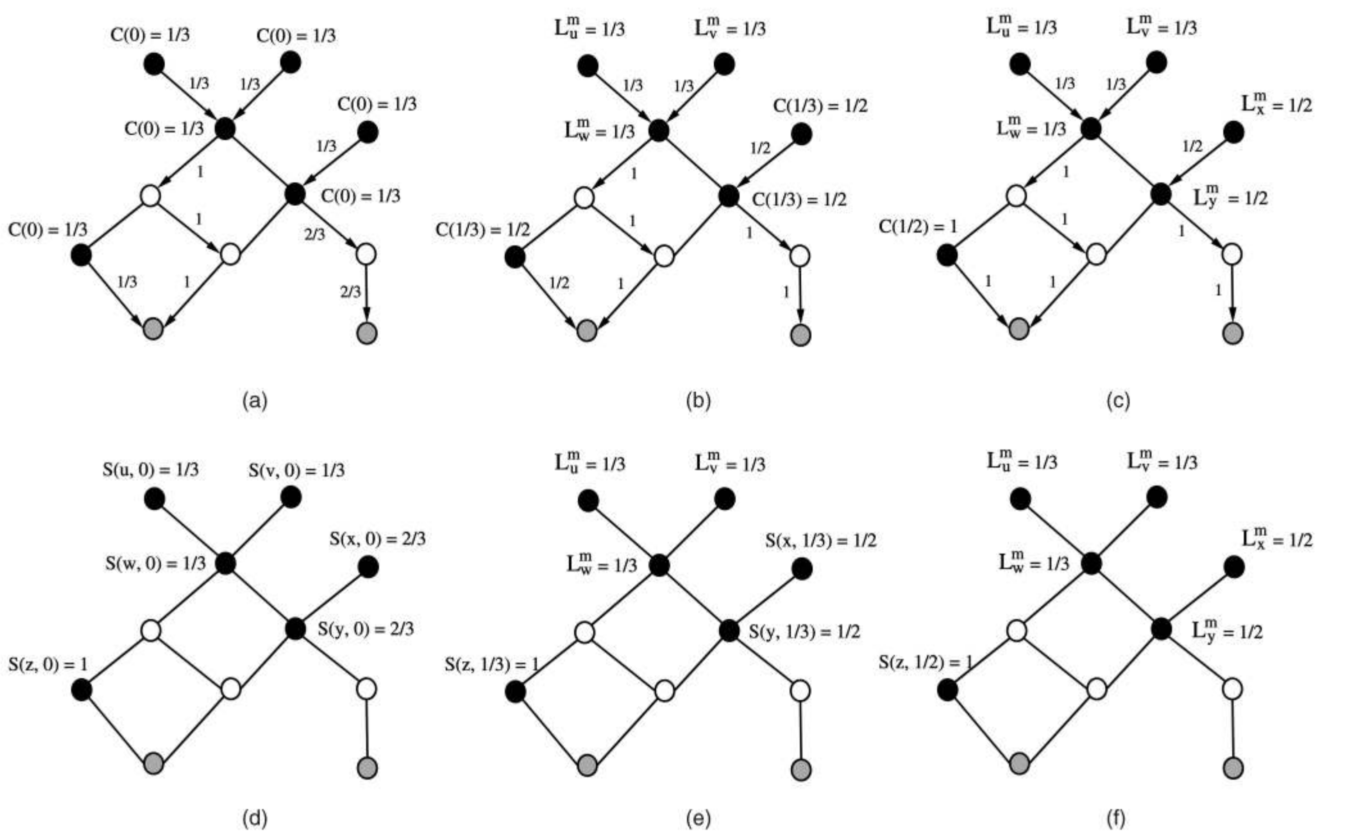


Fig. 3. (a) With $r = 0$, solving MCR gives $C(0) = 1/3$, which is the smallest maxmin rate. (b) With $r = 1/3$, solving MCR gives $C(1/3) = 1/2$, which is the next maxmin rate after $1/3$. (c) With $r = 1/2$, solving MCR gives $C(1/2) = 1$, which is the next maxmin rate after $1/2$. (d) With the knowledge of $C(0) = 1/3$, solving MSR finds the nodes (u, v, w) whose maxmin rates are $1/3$. (e) With the knowledge of $C(1/3) = 1/2$, solving MSR finds the nodes (x, y) whose maxmin rates are $1/2$. (f) With the knowledge of $C(1/2) = 1$, solving MSR finds the nodes (z) whose maxmin rates are 1.

$C(1/3) = 1/2$, as shown in Fig. 3b. $1/2$ is the second smallest maxmin rate. Third, with the knowledge of $L_A^m(1/2)$, by solving MCR, we get $C(1/2) = 1$, as shown in Fig. 3c. 1 is the last maxmin rate. The meaning of Figs. 3d, 3e, and 3f will be explained shortly.

We model MCR as a linear programming problem, which is a well-studied P problem that can be efficiently solved by various sophisticated algorithms such as interior-point methods and simplex methods.

$$
\begin{aligned}
&\mathit{Linear\ Constraints}: \\
&(1)\ \ L_x = C, \quad \forall x \in A - A(r) \\
&(2)\ \ L_x \le \Omega, \quad \forall x \in A - A(r) \\
&(3)\ \ L_x = L_x^m, \quad \forall x \in A(r) \\
&(4)\ \ L_x = 0, \quad \forall x \in N - A \\
&(5)\ \ f(x,y) \ge 0, \quad \forall (x,y) \in E \\
&(6)\ \ \Sigma_{y \in N_x} f(x,y) = \Sigma_{y \in N_x} f(y,x) + L_x, \quad \forall x \in N \\
&(7)\ \ \Sigma_{y \in N_x} f(x,y) \le T_x, \quad \forall x \in N \\
&\mathit{Optimization}: \\
&\quad \text{maximize } C
\end{aligned}
$$

Linear constraint 1 specifies the common-rate constraint. Linear constraint 2 ensures that the common rate is bounded by the maximum rate at which the active sensors generate new data. Linear constraint 3 specifies that the maxmin rates for sensors in $A(r)$ are known. Linear constraint 4 specifies that the rates of inactive sensors are zero. Linear constraint 5 ensures nonnegative forwarding rates on the links. Linear constraint 6 is the flow-conservation constraint. Linear constraint 7 is the capacity constraint.

Lemma 1 and Lemma 2 point out a way for us to iteratively find out all maxmin rates. First, find the smallest maxmin rate (Lemma 1) and then iteratively find the next greater maxmin rate (Lemma 2). There is one remaining problem: Suppose we know $L_A^m(r)$ and calculate $C(r)$ as the next greater maxmin rate. We must know $L_A^m(C(r))$, specifically, the set of sensors x with $L_x^m = C(r)$, before we can solve MCR to find the yet next maxmin rate $C(C(r))$. This problem is solved below.

4.3 Maximum Single Rate Problem (MSR)

Definition 4. *Maximum Single Rate Problem (MSR): Suppose $L_A^m(r)$ is known for a real number r. Consider an arbitrary sensor $x \in A - A(r)$. Let the sensors in A take their maxmin rates and the sensors in $A - A(r) - \{x\}$ take the same rate of $C(r)$. The problem is to find the maximum feasible rate $S(x, r)$ for x, which can be realized by a congestion-free forwarding schedule.*

$S(x, r)$ is a function of both x and r. For a given number r, we can find the value of $S(x, r)$ for any $x \in A - A(r)$ by solving the above problem.

Lemma 3. *Let r be a maxmin rate in L_A^m. $\forall x \in A - A(r)$, $L_x^m = C(r)$ iff $S(x,r) = C(r)$.*

By solving MSR for each $x \in A - A(r)$, based on Lemma 3, we know the set X of sensors x with $L_x^m = C(r)$. Consequently,

$$A(C(r)) = A(r) \cup X,$$

$$L_A^m(C(r)) = L_A^m(r) \cup \{L_x^m \mid x \in X\}.$$

An example is given in Fig. 3. After solving MCR in Fig. 3a, we know $C(0) = 1/3$. In Fig. 3d, we solve MSR for every active sensor to learn that $S(u,0) = S(v,0) = S(w,0) = 1/3$, which means $L_u^m = L_v^m = L_w^m = 1/3$. Similarly, after solving MCR in Fig. 3b, we know $C(1/3) = 1/2$. In Fig. 3e, we solve MSR to find that $S(x,1/3) = S(y,1/3) = 1/2$, which means $L_x^m = L_y^m = 1/2$. After solving MCR in Fig. 3c, we know $C(1/2) = 1$. In Fig. 3f, we solve MSR to find that $S(z,1/2) = 1$, which means $L_z^m = 1$.

MSR can also be modeled as a linear programming problem. Before solving MSR, $C(r)$ should be computed by solving MCR first.

$Linear\ Constraints:$

(1) $L_x = S$

(2) $L_x \leq \Omega$

(3) $L_y = C(r), \quad \forall y \in A - A(r) - \{x\}$

(4) $L_y = L_y^m, \quad \forall y \in A(r)$

(5) $L_y = 0, \quad \forall y \in N - A$

(6) $f(y,z) \geq 0, \quad \forall (y,z) \in E$

(7) $\Sigma_{z \in N_y} f(y,z) = \Sigma_{z \in N_y} f(z,y) + L_y, \quad \forall y \in N$

(8) $\Sigma_{z \in N_y} f(y,z) \leq T_y, \quad \forall y \in N$

$Optimization:$

maximize S

Linear constraint 1 selects a sensor x whose rate will be maximized. Linear constraint 2 ensures that the rate is bounded by the maximum rate at which any active sensor will generate new data. Linear constraint 3 ensures that all sensors in $A - A(r)$ other than x takes the rate of $C(r)$. Linear constraint 4 specifies that the maxmin rates for sensors in $A(r)$ are known. Linear constraint 5 specifies that the rates of inactive nodes are zero. Linear constraint 6 ensures nonnegative forwarding rates on the links. Linear constraint 7 is the flow-conservation constraint. Linear constraint 8 is the capacity constraint.

4.4 Finding Maxmin Assignment and Forwarding Schedule

The following algorithm computes L_A^m and its congestion-free forwarding schedule.

```
MaxminAssignment()
1.  r ← 0
2.  A(r) ← ∅
3.  while A(r) ≠ A do
4.      Compute C(r) by solving MCR
5.      X ← ∅
6.      for each x ∈ A − A(r) do
7.          Compute S(x, r) by solving MSR
8.          if S(x, r) = C(r) then
9.              L_x^m ← C(r)
10.             X ← X + {x}
11.     r ← C(r)
12.     A(r) ← A(r) + X
13. return L_A^m(r)
```

The last execution of Line 4 gives the congestion-free forwarding schedule for L_A^m. It consists of the set of nonzero $f(x,y)$ values after solving the linear program of MCR. The while loop has $O(|A|)$ iterations, each solving $O(|A|)$ linear programs. Therefore, the worst-case time complexity of MaxminAssignment() is $O(|A|^2 C(|E|))$, where $C(|E|)$ is the complexity of linear programming with $O(|E|)$ variables and $O(|E|)$ constraints. On the complexity of linear programming, readers are referred to [43].

Theorem 2. *MaxminAssignment() returns the maxmin assignment.*

The execution of MaxminAssignment() is illustrated by the example in Fig. 3, where Fig. 3a and Fig. 3d demonstrate the first iteration of the while loop, Fig. 3b and Fig. 3e the second iteration, and Fig. 3c and Fig. 3f the third iteration. Consider the first iteration. The result of Line 4 is shown in Fig. 3a, and the result of Lines 6-10 is shown in Fig. 3d, which determines that $L_u^m = L_v^m = L_w^m = 1/3$, providing the basis for the next iteration.

4.5 Considering Energy Expenditure for Receiving Packets

So far, we have ignored the energy expenditure for receiving packets. To take it into consideration, we need to modify the capacity constraint in the linear program for MCR. We replace

$$(7)\ \Sigma_{y \in N_x} f(x,y) \leq T_x, \quad \forall x \in N,$$

with

$$(7)\ \Sigma_{y \in N_x} f(x,y) + \lambda \times \Sigma_{y \in N_x} f(y,x) \leq T_x, \quad \forall x \in N,$$

where λ be the radio of the energy for receiving a packet to the energy for sending a packet. Sensor x has the energy for sending T_x packets per unit of time. If some of that energy is used for receiving packets, then the actual sending rate has to be reduced. The first term on the left is the rate at which sensor x sends packets. The second term on the left is the rate at which sensor x receives packets multiplied by λ; it is the reduction in the sending rate due to energy expenditure for receiving packets. The summation of these two terms must be bounded by T_x.

The last constraint in the linear program for MSR must be modified similarly.

4.6 Eliminating Long Forwarding Paths

The proposed algorithm and the analysis in the appendices can be applied to either the physical sensor network or a pruned one. If we apply the algorithm on a pruned network with long forwarding paths removed, the algorithm will return the maxmin assignment and its forwarding schedule for the pruned network. Suppose we want to use only the shortest paths to forward packets. For each sensor, there may exist multiple shortest paths to the base stations. All shortest paths form an acyclic graph from the active sensors to the base stations. Let E be the set of directed edges in this shortest-path graph. When we execute MaxminAssignment() with this E, the algorithm will return the maxmin assignment and the forwarding schedule using only the shortest paths.

5 Discussions on Media Contention

So far, we have assumed that, due to power constraint and long lifetime requirement, the maximum forwarding rate T_x of a sensor x is small enough such that the impact of media contention is negligible. This may be true for many applications, but not all. Below, we provide some discussions on the issue of media contention.

5.1 Contention Graph

The concept of contention graph was developed in [30], [33], [44]. Intuitively, the forwarding rate $f(x,y)$ from a sensor x to a neighbor y is constrained not only by the sensor's local constraint T_x but also by the activities of nearby sensors that compete for the same media. Two wireless links, (x,y) and (w,z), contend if x cannot transmit to y while w is transmitting to z. We denote such a contending relation as $(x,y) \bowtie (w,z)$. In general, there are three rules governing media contention. Let I_x be the set of sensors in the interfere range of x. It is normally true that $I_x \supseteq N_x$.

- *First contention rule*. A sensor cannot transmit two packets simultaneously. $\forall x \in N, \forall y,z \in N_x, y \neq z$, $(x,y) \bowtie (x,z)$.
- *Second contention rule*. A sensor cannot transmit and receive simultaneously. $\forall x \in N, \forall y,z \in N_x$, $(x,y) \bowtie (z,x)$.
- *Third contention rule*. When a sensor sends a packet, any sensor within its interference range should not be receiving another packet. $\forall x,w \in N, x \neq w$, $\forall y \in N_x$, $\forall z \in I_x$, $(x,y) \bowtie (w,z)$.

A contention graph is constructed based on the above rules. Each vertex in the graph represents a wireless link. There is an edge between two vertices, (x,y) and (w,z), if and only if the two vertices contend, i.e., $(x,y) \bowtie (w,z)$.

For specific MAC protocols such as CSMA/CA, there may be additional contention rules that are not considered above but can be easily added.

In the following, we consider four different types of constraints that may be added to MaxminAssignment() to address media contention.

5.2 Independent-Set Constraints

An independent set in the contention graph is a subset of vertices with no edge between any two of them. A proper independent set is one that is not contained by a larger independent set. All packet transmissions represented by the vertices in a proper independent set can be performed simultaneously. Different proper independent sets are scheduled for transmission serially at the MAC layer. Let Ψ be the set of proper independent sets in the contention graph. Let $t(\beta)$, $\forall \beta \in \Psi$, be the fraction of time when β is scheduled for transmission. We can modify Maxmin-Assignment() by adding the following *independent-set constraints* to the linear programs for MCR and MSR.

$$\begin{aligned} &\sum_{\beta \in \Psi} t(\beta) \leq 1 \\ &f(x,y) = M \times \sum_{\beta \in \Psi, (x,y) \in \beta} t(\beta),\ \forall (x,y) \in E, \end{aligned} \tag{1}$$

where M is the media throughput. It has been proven that any forwarding schedule, $\{f(x,y) \mid (x,y) \in E\}$, that satisfies the above constraints is schedulable at the MAC layer [44]. However, finding all proper independent sets is an NP-complete problem [44].

5.3 Clique Constraints

A clique α in the contention graph is a complete subgraph in which the vertices mutually contend. The packet transmissions represented by the vertices in a clique must be performed serially, which implies a linear constraint, $\sum_{(x,y) \in \alpha} f(x,y) \leq M$. A proper clique is a clique that is not contained by a larger clique. Let Γ be the set of proper cliques in the contention graph. We can modify Maxmin-Assignment() by adding the following *clique constraints* to the linear programs for MCR and MSR.

$$\sum_{(x,y) \in \alpha} f(x,y) \leq M, \quad \forall \alpha \in \Gamma.$$

The problem is that not all forwarding schedules that satisfy the clique constraints are schedulable at the MAC layer [44]. Because the clique constraints may allow unschedulable rate assignments, in case such an unschedulable assignment is greater than the maxmin assignment, MaxminAssignment() will return the unschedulable assignment. Therefore, with the clique constraints, MaxminAssignment() gives an upper bound of the maxmin assignment.

Next, we briefly discuss the complexity of computing the cliques. Finding the largest clique in a graph is NP-complete [45, p. 194] (transformation from CLIQUE), and the problem of finding Γ is at least as difficult. However, the contention graph has a special property that makes the problem easier. Each wireless link only contends with nearby links, more specifically, the links within one interference hop away if the previous three contention rules are considered. Therefore, the maximum size of any clique is bounded if the number of nearby wireless links are bounded, regardless how large the sensor network is. This is true for many applications such as environmental monitoring, where the number of sensors per unit of area does not need to exceed a certain number, which we call the upper threshold. Even if the randomness element in the deployment process (e.g., aerial dropping) causes the sensor density to exceed the threshold, the excess sensors can be put into sleep and woken up when active sensors die, which helps prolong the lifetime of the sensor network. Now, if both the transmission range and the sensor density are bounded by constants, the number of contending links in a one-hop neighborhood will also be bounded by a constant. In this case, the complexity of finding the proper cliques in a one-hop neighborhood is $O(1)$. The complexity of finding Γ is polynomial in the size of the network.

5.4 Complete-Contention Constraints

Consider an arbitrary wireless (x, y). Its complete contention set $\Theta(x, y)$ consists of all wireless links that (x, y) contends. We can modify MaxminAssignment() by adding the following *complete-contention constraints* to the linear programs for MCR and MSR.

$$f(x,y) + \sum_{(w,z)\in\Theta(x,y)} f(w,z) \leq M, \quad \forall (x,y) \in E.$$

Any forwarding schedule that satisfies the above constraints is schedulable at the MAC layer. It can be easily proven by induction. Suppose the statement is true for any network with k or less wireless links. Consider a network G with $k+1$ wireless links. Let $\{f(x,y) \mid (x,y) \in E\}$ be a forwarding schedule that satisfies the complete-contention constraints in G. Remove a wireless link (x, y) from G and the resulting network is denoted as G'. The forwarding schedule without $f(x, y)$ must satisfy the complete-contention constraints in G' because removing $f(x, y)$ only reduces the left side of the constraints. The forwarding schedule without $f(x, y)$ is schedulable at the MAC layer due to the induction assumption. Now, we attempt to add the link and $f(x, y)$ back. Node x can transmit to y when all contending links are idle. The total fraction of time when at least one of the contending links is busy is bounded by

$$\sum_{(w,z)\in\Theta(x,y)} \frac{f(w,z)}{M}.$$

The fraction of time when all contending links are idle is at least

$$1 - \sum_{(w,z)\in\Theta(x,y)} \frac{f(w,z)}{M}.$$

During this time, x is allowed to transmit to y, achieving a rate of

$$M - \sum_{(w,z)\in\Theta(x,y)} f(w,z),$$

which is greater than $f(x, y)$ due to the complete-contention constraint. Therefore, $f(x, y)$ is schedulable at the MAC layer after all other links are scheduled. This completes the induction proof.

The problem is that not all forwarding schedules that are schedulable at the MAC layer satisfy the complete-contention constraints. Consider a network topology $u \rightarrow x \rightarrow y \rightarrow w \rightarrow z$, where z is a base station and u and w are two active sensors. Because (x, y) contends with every other link, we have the following complete-contention constraint:

$$f(u,x) + f(x,y) + f(y,w) + f(w,z) \leq M.$$

However, $f(u,x) = f(x,y) = f(y,w) = M/4$ and $f(w,z) = M/2$ are schedulable at the MAC layer because packet transmissions on (u, x) and (w, z) can be performed in parallel.

Because the complete-contention constraints may exclude schedulable rate assignments in case the maxmin assignment is excluded, MaxminAssignment() will return a smaller assignment. Therefore, with the complete-contention constraints, MaxminAssignment() gives a lower bound of the maxmin assignment.

5.5 CDMA and Adjacent-Link Constraints

A specific MAC protocol may allow us to specify constraints that give tighter bounds. Consider CDMA as an example. Suppose the wireless links in the vicinity use different spread-spectrum codes. Two wireless links contend if and only if they share a common sensor because the sensor cannot transmit and receive more than one packet at a time. Clearly, all adjacent links of a sensor form a proper clique in the contention graph. We already know that using the clique constraints gives us an upper bound of the achievable maxmin assignment. The clique constraints for CDMA can be rewritten below.

$$\sum_{y\in N_x} (f(x,y) + f(y,x)) \leq M, \quad \forall x \in N.$$

Next, we prove that using the following *adjacent-link constraints* gives us a lower bound.

$$\sum_{y\in N_x} (f(x,y) + f(y,x)) \leq \frac{M}{2}, \quad \forall x \in N.$$

We only need to show that a forwarding schedule which satisfies the above constraints is schedulable at the MAC layer. It can be proven by induction. Suppose the statement is true for any network with k or fewer wireless links. Consider a network G with $k+1$ wireless links. Let $\{f(x,y) \mid (x,y) \in E\}$ be a forwarding schedule that satisfies the adjacent-link constraints in G. Remove a wireless link (x, y) from G and the resulting network is denoted as G'. The forwarding schedule without $f(x, y)$ must satisfy the adjacent-link constraints in G' because removing $f(x, y)$ only reduces the left side of the constraints. The forwarding schedule without $f(x, y)$ is schedulable at the MAC layer due to the induction assumption. Now, we attempt to add (x, y) back. Due to the use of CDMA, (x, y) only contends with the adjacent links of x or y. Node x can transmit to y when all contending links are idle. Below is the fraction of time when x's adjacent links other than (x, y) are busy.

$$\frac{f(y,x) + \sum_{z\in N_x,\ z\neq y}(f(x,z) + f(z,x))}{M}.$$

Due to the adjacent-link constraint, it is smaller than $\frac{1}{2} - \frac{f(x,y)}{M}$. Similarly, the fraction of time when y's adjacent links other than (x, y) are busy is also smaller than $\frac{1}{2} - \frac{f(x,y)}{M}$. Therefore, the fraction of time when all contending links of (x, y) are idle is bounded by

$$1 - \left(\frac{1}{2} - \frac{f(x,y)}{M}\right) - \left(\frac{1}{2} - \frac{f(x,y)}{M}\right) = \frac{2f(x,y)}{M}.$$

During this time, x is allowed to transmit to y, achieving a rate of $2f(x, y)$. This completes the induction proof.

5.6 Using Upper and Lower Bounds

If media contention is significant, MaxminAssignment() with the above additional constraints will return the upper and lower bounds of the maxmin assignment that is achievable at the MAC layer. Suppose we compute these bounds and distribute them to the sensors. At the beginning, the sensor network uses the upper-bound forwarding schedule. If the upper bound is not tight, congestion will occur at sensors that receive more packets than they can forward, causing buffer overflow. When sensor x is congested, it informs upstream neighbors to gradually reduce their forwarding rates until congestion is resolved. This can cause the upstream neighbors to be congested. When that happens, the upstream neighbors will inform their upstream neighbors to reduce the forwarding rates. Such backpressure will go all the way to the data sources. For details about the backpressure mechanism, readers are referred to [18], [46]. The difference is that, in our case, the forwarding rate of a link should not be reduced below the lower bound.

6 Maxmin Assignment with Edge or Mixed Capacities

In a theoretical stretch, we define the maxmin assignment problem on a graph with edge capacities instead of node capacities. Given a directed graph with edge capacities, $c(x, y), \forall (x, y) \in E$, the problem is to find the largest rate assignment (based on Definitions 1 and 2) and the congestion-free forwarding schedule that implements the assignment.

This problem can be solved in a similar way as described in Section 4. To solve MCR with edge capacities, the last constraint of the linear program in Section 4.2 must be replaced with

$$f(x, y) \leq c(x, y), \quad \forall (x, y) \in E. \tag{2}$$

To solve MSR with edge capacities, the last constraint of the linear program in Section 4.3 must also be replaced with (2). With these modifications, the same routine MaxminAssignment() can be used to solve the maxmin assignment problem with edge capacities.

We can similarly define the maxmin assignment problem on a graph with both node capacities and edge capacities. To solve this problem, we add (2) as an additional constraint to the linear programs in Sections 4.2 and 4.3.

7 Weighted Maxmin Assignment

So far, we have treated all sensors equally. In reality, sensors may carry different on-board instruments and their tasks may have different priorities. The weighted maxmin assignment captures such differences. Each sensor x is assigned a weight w_x. Intuitively, when two sensors compete for bandwidth, $w_x L_x = w_y L_y$ is considered to be fair. Given a rate assignment $L_A = \{L_x \mid x \in A\}$, the corresponding weighted rate assignment is defined as $W_A = \{\frac{L_x}{w_x} \mid x \in A\}$. W_A is feasible if L_A is feasible. Let $W_x = \frac{L_x}{w_x}$. It is called the weighted rate of x.

Definition 5. *Consider two feasible weighted rate assignments, $W_A = \{W_x \mid x \in A\}$ and $W'_A = \{W'_x \mid x \in A\}$. Suppose we sort both in ascending order and treat them as ascending rate vectors. We define the following operators: 1) $W_A = W'_A$ if the two vectors are identical and 2) $W_A > W'_A$ if there exists a prefix $(r_1, r_2, \ldots, r_i)$ of the sorted vector W_A and a prefix $(r'_1, r'_2, \ldots, r'_i)$ of the sorted vector W'_A such that $r_i > r'_i$ and $r_j = r'_j, 1 \leq j \leq i-1$.*

The *weighted maxmin assignment* is the largest assignment based on the above operators.

The following modifications are made to the algorithm in Section 4. Let $W^m_A = \{W^m_x \mid x \in A\}$ be the weighted maxmin assignment. $L^m_x = w_x W^m_x, \forall x \in A$, which is the actual date rate of x under such an assignment.

$$A(r) = \{x \mid W^m_x \leq r, x \in A\},$$

$$W^m_A(r) = \{W^m_x \mid x \in A(r)\}.$$

Definition 6. *Maximum Common Weighted Rate Problem (MCWR): Suppose $W^m_A(r)$ is known for a real number r. Let the sensors in $A(r)$ take their weighted maxmin rates and the sensors in $A - A(r)$ take a common weighted rate, i.e., $W_x = W_y$, $\forall x, y \in A - A(r)$. The problem is to find the maximum common weighted rate $C(r)$ that can be realized by a congestion-free forwarding schedule.*

The linear program for MCWR is the same as the one in Section 4.2 except that the first constraint is changed to

$$L_x = w_x C, \quad \forall x \in A - A(r).$$

Similar modifications must be made to MSR and its linear program.

8 Conclusion

This paper studies maxmin fairness for data collection in sensor networks. While the traditional maxmin flow control assumes that each flow has a fixed routing path, the maxmin assignment in sensor networks allows the packet flow from a sensor to follow multiple paths to a set of base stations, which makes the problem much harder. We prove that there is one and only one maxmin assignment for a given configuration of a sensor network. For low-rate sensor networks, we describe an algorithm that finds the maxmin assignment and its corresponding forwarding schedule in polynomial time. We also discuss the implications caused by media contention. Our contribution is to provide a theoretical foundation for the study of fairness in sensor networks from the maxmin perspective. The proposed algorithm may be used in a centralized scheme that collects information from the network, computes maxmin rates, and then disseminates the rates to the sensors. If the network conditions keep changing, distributed heuristic algorithms will be more appropriate in order to avoid excessive computation overhead, which will be studied in our future work.

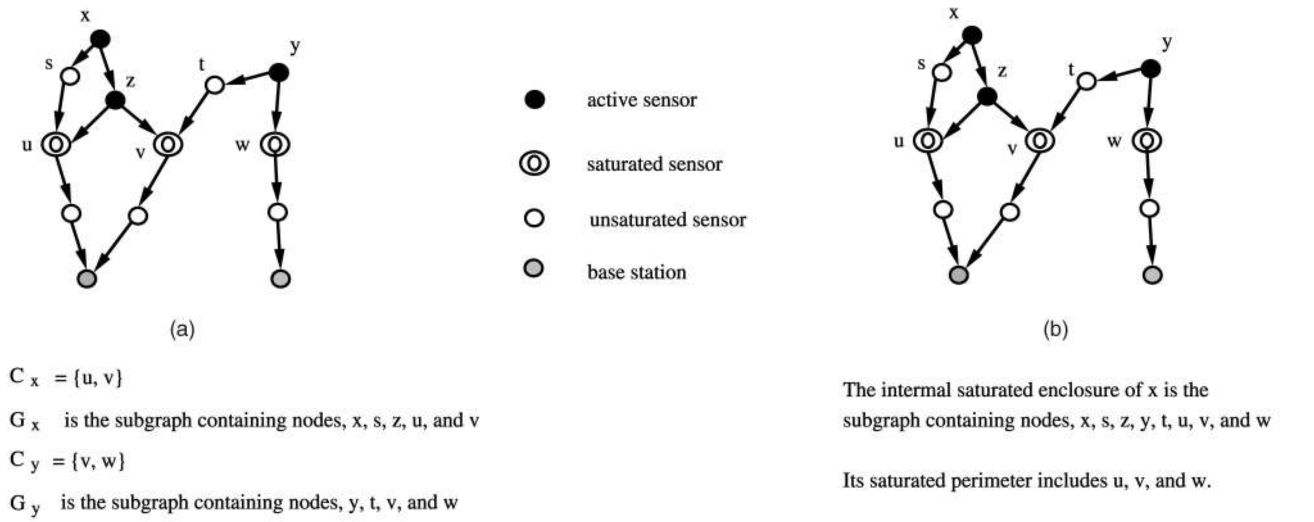


Fig. 4. Internal saturated enclosure $\mathbb{G}_x$ is the union of minimum saturated enclosures such that the traffic that saturates the perimeter nodes comes all from internal.

Appendix A

Proofs of Theorem 1 and Supporting Lemmas

Before proving Theorem 1, we give some important properties of the maxmin assignment. It establishes the theoretical foundation for our algorithm that finds the maxmin assignment and its forwarding schedule.

A *forwarding path* means a directed path P with a positive forwarding rate on every link, i.e., $f(x, y) > 0, \forall (x, y) \in P$. A *path* means a directed path who may or may not be a forwarding path.

Lemma 4. *Suppose $L_A = \{L_x \mid x \in A\}$ is a maxmin assignment. $\forall x \in A$, if $L_x < \Omega$, then there must be a saturated sensor on every path from x to any base station.*

Proof. We prove the lemma by contradiction. Suppose there exists a path P from x to a base station and no sensor on P is saturated. Let r be the residual forwarding capacity of P. L_x can be increased by x sending more on P. As long as the increment does not exceed r, the other active sensors will not be affected. The new assignment with the increased L_x is greater than L_A. This contradicts the lemma assumption that L_A is a maxmin assignment. □

Definition 7. *Minimum Saturated Enclosure (G_x, C_x): Consider a maxmin assignment L_A and an active sensor x with $L_x < \Omega$. For each path from x to a base station, we pick the saturated sensor closest to x. Let C_x be the set of closest saturated sensors picked over all paths from x to the base stations. Apparently, C_x forms a cut that separates all base stations from a subgraph G_x containing x. G_x is called the minimum saturated enclosure of x and C_x is called the saturated perimeter of G_x. Assume G_x includes C_x.*

Note that the traffic that saturates the nodes in C_x may come both from nodes in G_x and from nodes outside of G_x, as illustrated by the example in Fig. 4a.

Under the appropriate context, we also use G_x for the set of active sensors in the subgraph that it represents.

Lemma 5. *Suppose $L_A = \{L_x \mid x \in A\}$ is a maxmin assignment. $\forall x \in A$, if $L_x < \Omega$, then $L_y \leq L_x,\ \forall y \in G_x$.*

Proof. If x itself is saturated, $G_x = \{x\}$ and the lemma holds. If x is not saturated, we prove the lemma by contradiction. Suppose $\exists y \in G_x,\ L_y > L_x$. We show that a rate assignment greater than L_A can be derived.

L_A is feasible and has a congestion-free forwarding schedule. Consider a forwarding path from y to a base station. Refer to Fig. 5. The path must pass one node z in C_x in order to exit G_x. Let P_1 be the path segment from y to z.

According to the definition of C_x, there must exist a path P_2 from x to z with all intermediate sensors not saturated. Let r be the residual forwarding capacity of P_2.

With P_1 and P_2 both incident on z, we can shift a tiny forwarding rate δ from P_1 to P_2 as long as $\delta \leq r$. Namely, we first reduce L_y by δ and decrease the forwarding rate along P_1 by δ. We then improve L_x by δ and increase the forwarding rate along P_2 by δ. z remains saturated. The rates of all sensors except x and y are not affected either. Moreover, if we choose $\delta < L_y - L_x$, the new rate assignment with modified L_x and L_y is greater than L_A

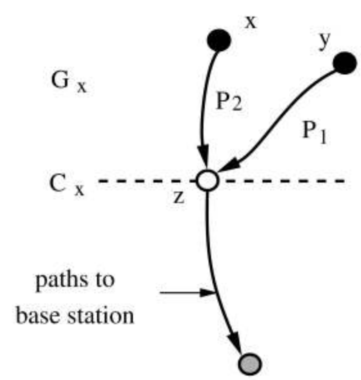


Fig. 5. Illustration for the proof of Lemma 5.

based on Definition 1, which contradicts the lemma assumption that L_A is maxmin optimal. □

Our basic technique used to prove Lemma 5 is called *rate shifting*, which is to identify two or more paths such that, given certain conditions, an amount of forwarding rate may be shifted across paths, resulting in a larger assignment. The same technique will be used in proving other lemmas.

Definition 8. *Internal Saturated Enclosure (Π_x, $\mathbb{G}_x$, $\mathbb{C}_x$): Let Π_x be the set of active sensors from which there is at least one forwarding path to a node in C_x. For example, in Fig. 4a, $y \in \Pi_x$. Obviously, $x \in \Pi_x$. If $L_x < \Omega$, we define*

$$\mathbb{G}_x = \bigcup_{y \in A,\ L_y \leq L_x} G_y. \quad (3)$$

$\mathbb{G}_x$ is the union of G_y, for all $y \in A, L_y \leq L_x$. It may or may not be a connected graph. Let $\mathbb{C}_x$ be the set of perimeter sensors that separate $\mathbb{G}_x$ from the rest of the graph.

$$\mathbb{C}_x \subseteq \bigcup_{y \in A,\ L_y \leq L_x} C_y. \quad (4)$$

$\mathbb{G}_x$ is called the internal saturated enclosure *of x and $\mathbb{C}_x$ is called the* saturated perimeter *of $\mathbb{G}_x$. We will prove a lemma that the traffic saturating the nodes in $\mathbb{C}_x$ must all come from the internal nodes in $\mathbb{G}_x$, as illustrated by the example in Fig. 4b.*

Under the appropriate context, we also use $\mathbb{G}_x$ for the set of active sensors in the subgraph it represents.

Lemma 6. *Suppose $L_A = \{L_x \mid x \in A\}$ is a maxmin assignment. $\forall x \in A$, if $L_x < \Omega$, then $\mathbb{G}_x$ does not contain a base station.*

Proof. Because G_y does not contain a base station for all $y \in A, L_y \leq L_x$, by definition. □

Lemma 7. *Suppose $L_A = \{L_x \mid x \in A\}$ is a maxmin assignment. $\forall x \in A$, if $L_x < \Omega$, then $L_y \leq L_x, \forall y \in \mathbb{G}_x$.*

Proof. Consider an arbitrary sensor $y \in \mathbb{G}_x$. By the definition of $\mathbb{G}_x$, it must be true that, $\exists v \in A,\ L_v \leq L_x$, such that $y \in G_v$. By Lemma 5, $L_y \leq L_v$. Therefore, $L_y \leq L_x$. □

Lemma 8. *Suppose $L_A = \{L_x \mid x \in A\}$ is a maxmin assignment. $\forall x \in A$, if $L_x < \Omega$, then $L_y \leq L_x$, $\forall y \in \Pi_x$.*

Proof. The proof is similar to that of Lemma 5. We still use Fig. 5 as illustration, except that y may now be outside of G_x. Because $y \in \Pi_x$, there is a forwarding path P_1 from y to $z \in C_x$. According to the definition of C_x, there exists an unsaturated path P_2 from x to z. If $L_y > L_x$, we are able to construct a rate assignment greater than L_A by shifting a tiny forwarding rate from P_1 to P_2, which contradicts the lemma assumption that L_A is maxmin optimal. □

Lemma 9. *Suppose $L_A = \{L_x \mid x \in A\}$ is a maxmin assignment. $\forall x \in A$, if $L_x < \Omega$, then there does not exist a forwarding path from a sensor outside $\mathbb{G}_x$ to a sensor in $\mathbb{C}_x$.*

Proof. We prove the lemma by contradiction. Suppose there is a forwarding path from $y \notin \mathbb{G}_x$ to $z \in \mathbb{C}_x$. By Lemma 8, $L_y \leq L_x$ and, therefore, $\mathbb{G}_x$ includes G_y, according to the definition of $\mathbb{G}_x$. Since $y \in G_y$, we have $y \in \mathbb{G}_x$. □

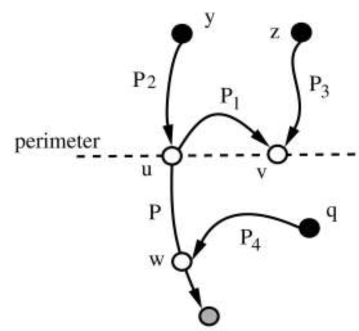


Fig. 6. Removing cross-perimeter paths.

For a maxmin assignment L_A, we attempt to remove any forwarding path P_1 from a perimeter sensor u to another perimeter sensor v in $\mathbb{C}_x$. Such a path is called a *cross-perimeter path*. As shown in Fig. 6, let P_2 be an unsaturated forwarding path from an active sensor y to u and P_3 be an unsaturated forwarding path from an active sensor z to v. Let P be an arbitrary path outside of $\mathbb{G}_x$ from u to a base station. There must be a saturated node on P because otherwise we could increase L_z by shifting a tiny forwarding rate from P_1 to P_3 while increasing the forwarding rate on P by the same amount. Let $w(\neq u)$ be the first saturated node on P. Let P' be the subpath of P from u to w. There must not be any forwarding path P_4 from an active sensor q to w, where $L_q > L_x$, because otherwise we could increase L_y by shifting a tiny forwarding rate from P_4 to the concatenated path of P_2 and P'. To remove a cross-perimeter path P_1, we add P' to $\mathbb{G}_x$ for each path P from u to a base station, which turns u an internal node of $\mathbb{G}_x$. After all cross-perimeter paths are removed, the resulting subgraph is denoted as $\mathbb{G}_x^*$ and its new perimeter is denoted as $\mathbb{C}_x^*$. The set of active sensors in $\mathbb{G}_x^*$ is the same as the set in the original $\mathbb{G}_x$. Lemma 6 and Lemma 9 hold for $\mathbb{G}_x^*$ and $\mathbb{C}_x^*$ as well.

Lemma 10. *Suppose $L_A = \{L_x \mid x \in A\}$ is a maxmin assignment. $\forall x \in A$, if $L_x < \Omega$, then*

$$\Sigma_{y \in \mathbb{G}_x} L'_y \leq \Sigma_{y \in \mathbb{G}_x} L_y$$

for any other rate assignment $L'_A = \{L'_x \mid x \in A\}$.[5]

Proof. The total rate of all active sensors in $\mathbb{G}_x$ is

$$\Sigma_{y \in \mathbb{G}_x} L_y.$$

The sum of the maximum forwarding rates of all sensors in $\mathbb{C}_x^*$ is

$$\Sigma_{y \in \mathbb{C}_x^*} T_y.$$

By Lemma 6, $\mathbb{G}_x$ is encapsulated by $\mathbb{C}_x^*$ from the base stations. Any data generated from sensors in $\mathbb{G}_x$ must pass $\mathbb{C}_x^*$ in order to reach a base station. Hence, we must have

$$\Sigma_{y \in \mathbb{G}_x} L_y \leq \Sigma_{y \in \mathbb{C}_x^*} T_y.$$

By Lemma 9, there does not exist a forwarding path from a sensor outside $\mathbb{G}_x$ to a sensor in $\mathbb{C}_x^*$. It means that the

5. Note that $\mathbb{G}_x$ is defined based on L_A, not L'_A.

forwarding throughput of $\mathbb{C}_x^*$ is consumed only by nodes inside $\mathbb{G}_x$. Yet, every sensor y in $\mathbb{C}_x^*$ is saturated. Therefore,

$$\Sigma_{y \in \mathbb{G}_x} L_y = \Sigma_{y \in \mathbb{C}_x^*} T_y. \tag{5}$$

Now, consider L'_A. The total rate of all active sensors in $\mathbb{G}_x$ is

$$\Sigma_{y \in \mathbb{G}_x} L'_y.$$

Again, because $\mathbb{G}_x$ is physically encapsulated by $\mathbb{C}_x^*$ and any data generated from sensors in $\mathbb{G}_x$ must pass $\mathbb{C}_x^*$ to reach a base station, we have

$$\Sigma_{y \in \mathbb{G}_x} L'_y \leq \Sigma_{y \in \mathbb{C}_x^*} T_y. \tag{6}$$

By (5) and (6), we have

$$\Sigma_{y \in \mathbb{G}_x} L'_y \leq \Sigma_{y \in \mathbb{G}_x} L_y.$$

□

Theorem 3. *There exists one and only one maxmin optimal rate assignment.*

Proof. The "=" operator defines equivalent groups of rate assignments. The ">" operator places a total order on the equivalent groups. Because the rates of an assignment, $L_x, x \in A$, are taken from closed real-number intervals, the set of feasible rate assignments must also form a closed space. Since a *total order* is enforced (due to ">") among all equivalent groups in a closed space, there must exist a largest one. By Definition 2, this largest equivalent group is the set of maxmin optimal rate assignments. Next, we prove by contradiction that there can be only one member in this group.

Suppose there are two different maxmin optimal rate assignments, $L_A = \{L_x \mid x \in A\}$ and $L'_A = \{L'_x \mid x \in A\}$. L_A and L'_A are identical as sorted rate vectors but their rate assignments for specific sensors differ. *Consider the set of sensors that have different rates in L_A and L'_A. Among them, let x be the one with the smallest rate in L_A.*

$$L_x \neq L'_x. \tag{7}$$

Let $\Phi_x = \{y \mid L_y < L_x,\ y \in A\}$, i.e., the set of sensors whose rates in L_A are smaller than the value L_x.

$$L_y = L'_y, \quad \forall y \in \Phi_x. \tag{8}$$

Now, let $\Phi'_x = \{y \mid L'_y < L_x,\ y \in A\}$, i.e., the set of sensors whose rates in L'_A are smaller than the value L_x. By (8), we know $\Phi_x \subseteq \Phi'_x$. We further argue that

$$\Phi_x = \Phi'_x \tag{9}$$

because, otherwise, L_A and L'_A would have different numbers of sensors whose rates are below the value L_x and, consequently, $L_A = L'_A$ would not hold.

Substituting Φ_x by Φ'_x in (8), we have

$$L_y = L'_y, \quad \forall y \in \Phi'_x. \tag{10}$$

We know $L_x \neq L'_x$ by (7). We further argue that

$$L_x < L'_x \tag{11}$$

because, otherwise, $L'_x < L_x$ would put x in Φ'_x, which, in turn, would assert that $L_x = L'_x$ by (10).

Because both L_x and L'_x are bounded by the default rate Ω, by (11), we must have

$$L_x < \Omega. \tag{12}$$

By (8), we have

$$\Sigma_{y \in \mathbb{G}_x \bigcap \Phi_x} L'_y = \Sigma_{y \in \mathbb{G}_x \bigcap \Phi_x} L_y. \tag{13}$$

By (12) and Lemma 7, we have

$$\forall y \in \mathbb{G}_x, L_y \leq L_x. \tag{14}$$

Based on the definition of Φ_x, from (14), we have

$$\forall y \in \mathbb{G}_x - \Phi_x, L_y = L_x. \tag{15}$$

From Lemma 10, we have

$$\Sigma_{y \in \mathbb{G}_x} L'_y \leq \Sigma_{y \in \mathbb{G}_x} L_y. \tag{16}$$

By combining (13) and (16), we have

$$\Sigma_{y \in \mathbb{G}_x - \Phi_x} L'_y \leq \Sigma_{y \in \mathbb{G}_x - \Phi_x} L_y. \tag{17}$$

Further, by (15), we have

$$\Sigma_{y \in \mathbb{G}_x - \Phi_x} L'_y \leq \Sigma_{y \in \mathbb{G}_x - \Phi_x} L_y = |\mathbb{G}_x - \Phi_x| \cdot L_x. \tag{18}$$

We know $L'_x > L_x$ from (11) and $x \in \mathbb{G}_x - \Phi_x$. In order for (18) to hold, there must exist $y \in \mathbb{G}_x - \Phi_x$ such that $L'_y < L_x$, which leads to the contradiction: First, $y \in \mathbb{G}_x - \Phi_x$ means $y \notin \Phi_x$. Second, $L'_y < L_x$ means $y \in \Phi'_x$. These facts contradict (9). □

APPENDIX B

PROOFS OF THEOREM 2 AND SUPPORTING LEMMAS

Lemma 1. *$C(0)$ is the smallest maxmin rate in L_A^m.*

Proof. $A(0) = \emptyset$. Let r' be the smallest rate in L_A^m. $C(0)$ cannot be smaller than r' because we can transform L_A^m and its forwarding schedule by reducing the rate of each active sensor in A to r', which would be a solution for MCR better than $C(0)$.

$C(0)$ cannot be greater than r' because, otherwise, the rate assignment from solving MCR would be greater than L_A^m.

Therefore, $C(0)$ must be equal to r'. □

Lemma 2. *Let r be a maxmin rate in L_A^m. $C(r)$ is the smallest maxmin rate in L_A^m that is greater than r.*

Proof. Let r' be the next greater maxmin rate after r in L_A^m. $C(r)$ cannot be smaller than r' because we can transform L_A^m and its forwarding schedule by reducing the rate of each active sensor in $A - A(r)$ to r', which would be a solution for MCR better than $C(r)$.

$C(r)$ cannot be greater than r' because, otherwise, the rate assignment from solving MCR would be greater than L_A^m.

Therefore, $C(r)$ must be equal to r'. □

Lemma 3. *Let r be a maxmin rate in L_A^m. $\forall x \in A - A(r)$, $L_x^m = C(r)$ iff $S(x, r) = C(r)$.*

Proof. $S(x, r) \geq L_x^m$ because, if $S(x, r) < L_x^m$, we can transform L_A^m and its forwarding schedule by reducing the rate of each active sensor in $A - A(r) - \{x\}$ to $C(r)$ while keeping the maxmin rate of x, which would be a solution for MSR better than $S(x, r)$. Because $C(r)$ is the next greater maxmin rate after r (Lemma 2), $C(r)$ must be the smallest maxmin rate among nodes in $A - A(r)$, which means $L_x^m \geq C(r)$. Therefore, $S(x, r) \geq L_x^m \geq C(r)$. It is easy to see that, if $S(x, r) = C(r)$, then $L_x^m = C(r)$.

Next, we prove that, if $L_x^m = C(r)$, then $S(x, r) = C(r)$. We rewrite (3) by replacing L_x with L_x^m and L_y with L_y^m. Namely,

$$\mathbb{G}_x = \bigcup_{y \in A,\ L_y^m \leq L_x^m} G_y.$$

The same replacement should be done when we make reference to lemmas. Recall that $\mathbb{G}_x$ contains x but no base station (Lemma 6) and it is separated from the rest of the network by a set of saturated sensors $\mathbb{C}_x$.

Because $L_x^m = C(r)$, by Lemma 7, we have

$$\forall y \in \mathbb{G}_x, L_y^m \leq L_x^m = C(r). \tag{19}$$

$C(r)$ is the next greater maxmin rate after r. By the definition of $A(r)$ and (19), we have

$$\forall y \in \mathbb{G}_x - A(r), L_y^m = C(r).$$

Consequently,

$$\Sigma_{y \in \mathbb{G}_x} L_y^m = \Sigma_{y \in \mathbb{G}_x \bigcap A(r)} L_y^m + |\mathbb{G}_x - A(r)| \cdot C(r). \tag{20}$$

Let $L'_A = \{L'_x \mid x \in A\}$ be the rate assignment from solving MSR. We know that

$$\Sigma_{y \in \mathbb{G}_x} L'_y = \Sigma_{y \in \mathbb{G}_x \bigcap A(r)} L_y^m + (|\mathbb{G}_x - A(r)| - 1) \cdot C(r) + S(x, r). \tag{21}$$

By Lemma 10, we have

$$\Sigma_{y \in \mathbb{G}_x} L'_y \leq \Sigma_{y \in \mathbb{G}_x} L_y^m.$$

Applying (20) and (21), we have

$$S(x, r) \leq C(r).$$

We know $S(x, r) \geq C(r)$ at the beginning of the proof. Therefore, $S(x, r) = C(r)$. □

Theorem 42. *MaxminAssignment() returns the maxmin assignment.*

Proof. We prove by induction that, if we know $L_A^m(r)$ for a maxmin rate r before we enter an iteration of the while loop (Lines 4-12), then we must know $L_A^m(r')$ for the next greater maxmin rate r' at the end of the iteration. The induction base is given by Lines 1-2.

By Lemma 2, $C(r)$ calculated by Line 4 is the next greater maxmin rate. By Lemma 3, Lines 6-10 find all sensors x with $L_x^m = C(r)$. After Line 12, $A(C(r))$ and $L_A^m(C(r))$ have been found. $L_A^m(C(r))$ is the union of $L_A^m(r)$ and $\{x \mid L_x^m = C(r), x \in A\}$.

Because the number of different rates in L_A^m is limited, the while loop will stop. After the last iteration, L_A^m is known. □

ACKNOWLEDGMENTS

This work was supported in part by the US National Science Foundation under grant ANI-0093241 (CAREER Award), grant DBI-0529012, and grant CNS-0644033.

REFERENCES

[1] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat Monitoring: Application Drive for Wireless Communications Technology," *Proc. ACM SIGCOMM Workshop Data Comm. in Latin America and the Caribbean,* Apr. 2001.

[2] E. Biagioni and K. Bridges, "The Applications of Remote Sensor Technology to Assist the Recovery of Rare and Endangered Species," *Int'l J. High Performance Computing Applications,* Apr. 2003.

[3] L. Schwiebert, S. Gupta, and J. Weinmann, "Research Challenges in Wireless Networks of Biomedical Sensors," *Mobile Computing and Networking,* pp. 151-165, 2001.

[4] H.T. Kung and D. Vlah, "Efficient Location Tracking Using Sensor Networks," *Proc. IEEE Wireless Comm. and Networking Conf. (WCNC '03),* Mar. 2003.

[5] R. Brooks, P. Ramanathan, and A. Sayeed, "Distributed Target Classification and Tracking in Sensor Networks," *Proc. IEEE,* vol. 91, no. 8, pp. 1163-1171, 2003.

[6] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A Survey on Sensor Networks," *IEEE Comm. Magazine,* Aug. 2002.

[7] C. Chien, I. Elgorriaga, and C. McConaghy, "Low-Power Direct-Sequence Spread-Spectrum Modem Architecture for Distributed Wireless Sensor Networks," *Proc. Int'l Symp. Low Power Electronics and Design (ISLPED '01),* Aug. 2001.

[8] R.J. Cramer, M.Z. Win, and R.A. Scholtz, "Impulse Radio Multipath Characteristics and Diversity Reception," *Proc. IEEE Int'l Conf. Comm. (ICC '98),* June 1998.

[9] E. Shih, S. Cho, N. Ickes, R. Min, A. Sinha, A. Wang, and A. Chandrakasan, "Physical Layer Driven Protocol and Algorithm Design for Energy-Efficient Wireless Sensor Networks," *Proc. MobiCom '01,* July 2001.

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

[11] K. Sohrabi, J. Gao, V. Ailawadhi, and G.J. Pottie, "Protocols for Self-Organization of a Wireless Sensor Network," *IEEE Personal Comm.,* Oct. 2000.

[12] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva, "Directed Diffusion for Wireless Sensor Networking," *ACM/IEEE Trans. Networking,* vol. 11, no. 1, pp. 2-16, Feb. 2002.

[13] W.R. Heinzelman, J. Kulik, and H. Balakrishnan, "Adaptive Protocols for Information Dissemination in Wireless Sensor Networks," *Proc. MobiCom '99,* Aug. 1999.

[14] S.M. Hedetniemi, S.L. Hedetniemi, and A. Liestman, "A Survey of Gossiping and Broadcasting in Communication Networks," *Networks,* vol. 18, 1988.

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

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

[17] Q. Fang, J. Gao, and L.J. Guibas, "Locating and Bypassing Routing Holes in Sensor Networks," *Proc. INFOCOM '04,* Mar. 2004.

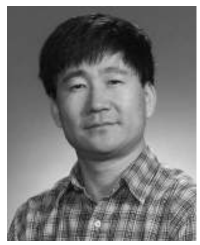
[18] C.-Y. Wan, S.B. Eisenman, and A.T. Campbell, "CODA: Congestion Detection and Avoidance in Sensor Networks," *Proc. ACM Conf. Embedded Networked Sensor Systems (SenSys '03),* Nov. 2003.

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

[20] T. He, J.A. Stankovic, C. Lu, and T.F. Abdelzaher, "SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks," *Proc. Int'l Conf. Distributed Computing Systems (ICDCS '03),* May 2003.

[21] D. Bertsekas and R. Galager, *Data Networks,* second ed. Prentice Hall, 1987.
[22] J.M. Faffe, "Bottleneck Flow Control," *IEEE Trans. Comm.,* vol. 29, no. 7, pp. 954-962, July 1981.
[23] J. Mosley, "Asynchronous Distributed Flow Control Algorithms," PhD thesis, Dept. of Electrical Eng. and Computer Science, Massachusetts Inst. of Technology, Cambridge, Mass., 1984.
[24] F. Bonomi and K. Fendick, "The Rate-Based Flow Control Framework for the Available Bit Rate ATM Service," *IEEE Network,* vol. 9, no. 2, pp. 25-39, Mar./Apr. 1995.
[25] A. Charny, D.D. Clark, and R. Jain, "Congestion Control with Explicit Rate Indication," *Proc. IEEE Int'l Conf. Comm.,* 1995.
[26] B. Awerbuch and Y. Shavitt, "Converging to Approximated Max-Min Flow Fairness in Logarithmic Time," *Proc. INFOCOM '98,* Apr. 1998.
[27] L. Georgiadis, P. Georgatsos, K. Floros, and S. Sartzetakis, "Lexicographically Optimal Balanced Networks," *IEEE/ACM Trans. Networking,* vol. 10, no. 6, 2002.
[28] P. Marbach, "Priority Service and Max-Min Fairness," *Proc. INFOCOM '02,* June 2002.
[29] Q. Ma, P. Steenkiste, and H. Zhang, "Routing High-Bandwidth Traffic in Max-Min Fair Share Networks," *Proc. ACM Special Interest Group Data Comm. (SIGCOMM '96),* Aug. 1996.
[30] H. Luo, S. Lu, and V. Bharghavan, "A New Model for Packet Scheduling in Multihop Wireless Networks," *Proc. MobiCom '00,* Aug. 2000.
[31] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving MAC Layer Fairness in Wireless Packet Networks," *Proc. MobiCom '00,* Aug. 2000.
[32] B. Bensaou, Y. Wang, and C.C. Ko, "Fair Medium Access in 802.11 Based Wireless Ad Hoc Networks," *Proc. MobiHoc '00,* July 2000.
[33] X.L. Huang and B. Bensaou, "On Max-Min Fairness and Scheduling in Wireless Ad Hoc Networks: Analytical Framework and Implementation," *Proc. MobiHoc '01,* Oct. 2001.
[34] H. Luo, J. Cheng, and S. Lu, "Self-Coordinating Localized Fair Queueing in Wireless Ad Hoc Networks," *IEEE Trans. Mobile Computing,* vol. 3, no. 1, Jan.-Mar. 2004.
[35] Y. Yi and S. Shakkottai, "Hop-by-Hop Congestion Control over a Wireless MultiHop Network," *Proc. INFOCOM '04,* Mar. 2004.
[36] L. Tassiulas and S. Sarkar, "Maxmin Fair Scheduling in Wireless Networks," *Proc. INFOCOM '02,* June 2002.
[37] A. Sridharan and B. Krishnamachari, "Max-Min Fair Collision-Free Scheduling for Wireless Sensor Networks," *Proc. Workshop Multihop Wireless Networks (MWN '04),* Apr. 2004.
[38] V. Gambiroza, B. Sadeghi, and E.W. Knightly, "End-to-End Performance and Fairness in Multihop Wireless Backhaul Networks," *Proc. MobiCom '04,* Sept.-Oct. 2004.
[39] B. Sadeghi, V. Kanodia, A. Sabharwal, and E. Knightly, "Opportunistic Media Access for Multirate Ad Hoc Networks," *Proc. MobiCom '02,* Sept. 2002.
[40] M. Heussse, F. Rousseau, G. Berger-Sabbatel, and A. Duda, "Performance Anomaly of 802.11b," *Proc. INFOCOM '03,* Apr. 2003.
[41] B. Radunovic and J. LeBoudec, "Rate Performance Objectives of MultiHop Wireless Networks," *Proc. INFOCOM '04,* Mar. 2004.
[42] S. Chen and K. Nahrstedt, "Maxmin Fair Routing in Connection-Oriented Networks," *Proc. Euro-Parallel and Distributed Systems Conf.,* July 1998.
[43] N. Megiddo, "On the Complexity of Linear Programming," *Advances in Economic Theory: Fifth World Congress,* pp. 225-268, 1987.
[44] K. Jain, J. Padhye, V.N. Padmanabhan, and L. Qiu, "Impact of Interference on MultiHop Wireless Network Performance," *Proc. MobiCom '03,* Sept. 2003.
[45] M.R. Garey and D.S. Johnson, *Computer and Intractability, A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, 1979.
[46] S. Chen and N. Yang, "Congestion Avoidance Based on Light-Weight Buffer Management in Sensor Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 17, no. 9, Sept. 2006.
[47] N. Megiddo, "Fairness in Routing and Load Balancing," *Math Programming,* vol. 7, 1974.
[48] J.M. Kleinberg, Y. Rabani, and E. Tardos, "Fairness in Routing and Load Balancing," *Proc. IEEE Symp. Foundations of Computer Science,* 1999.

Shigang Chen received the BS degree in computer science from the University of Science and Technology of China in 1993. He received the MS and PhD degrees in computer science from the University of Illinois at Urbana-Champaign in 1996 and 1999, respectively. After graduation, he worked with Cisco Systems for three years before joining the University of Florida as an assistant professor in the Department of Computer and Information Science and Engineering. His research interests include wireless networks, Internet security, and overlay networks. He received the IEEE Communications Society Best Tutorial Paper Award in 1999. He was a guest editor for the *ACM/Baltzer Journal on Wireless Networks* (WINET) and the *IEEE Transactions on Vehicle Technologies*. He served as a technical program committee cochair for the Computer and Network Security Symposium of IEEE IWCCC 2006, a vice technical program committee chair for IEEE MASS 2005, a vice general chair for QShine 2005, a technical program committee cochair for QShine 2004, and a technical program committee member for many conferences including IEEE ICNP, IEEE INFOCOM, IEEE SANS, IEEE ISCC, IEEE Globecom, etc.



Yuguang Fang received the PhD degree in systems engineering from Case Western Reserve University in January 1994 and the PhD degree in electrical engineering from Boston University in May 1997. He was an assistant professor in the Department of Electrical and Computer Engineering at the New Jersey Institute of Technology from July 1998 to May 2000. He then joined the Department of Electrical and Computer Engineering at the University of Florida in May 2000 as an assistant professor and got an early promotion to an associate professor with tenure in August 2003 and to a full professor in August 2005. He holds a University of Florida Research Foundation (UFRF) Professorship from 2006 to 2009. He has published more than 200 papers in refereed professional journals and conferences. He received the US National Science Foundation Faculty Early Career Award in 2001 and the US Office of Naval Research Young Investigator Award in 2002. He has served on several editorial boards of technical journals including the *IEEE Transactions on Communications*, the *IEEE Transactions on Wireless Communications*, the *IEEE Transactions on Mobile Computing*, and *ACM Wireless Networks*. He has also been actively participating in professional conference organizations such as serving as the steering committee cochair for QShine, the technical program vice chair for IEEE INFOCOM 2005, the technical program symposium cochair for IEEE Globecom 2004, and a member of the technical program committee for IEEE INFOCOM (1998, 2000, 2003-2007). He is a senior member of the IEEE.

Ye Xia received the BA degree in 1993 from Harvard University, the MS degree in 1995 from Columbia University, and the PhD degree in 2003 from the University of California, Berkeley, all in electrical engineering. He has been an assistant professor in the Computer and Information Science and Engineering Department at the University of Florida since in August 2003. Between June 1994 and August 1996, he was a member of the technical staff at Bell Laboratories, Lucent Technologies, New Jersey. His research interests are in the area of computer networking, including performance evaluation of network protocols and algorithms, congestion control, resource allocation, and load balancing on peer-to-peer networks. He is also interested in probability theory, stochastic processes, and queuing theory.