Providing End-to-End Delay Guarantees for Multi-hop Wireless Sensor Networks

I-Hong Hou
CESG and Department of ECE
Texas A&M University
College Station, TX, USA
ihou@tamu.edu

***Abstract*—Wireless sensor networks have been increasingly used for real-time surveillance over large areas. In such applications, it is important to support end-to-end delay constraints for packet deliveries even when the corresponding flows require multi-hop transmissions. In addition to delay constraints, each flow of real-time surveillance may require some guarantees on throughput of packets that meet the delay constraints. Further, as wireless sensor networks are usually deployed in challenging environments, it is important to specifically consider the effects of unreliable wireless transmissions.**

In this paper, we study the problem of providing end-to-end delay guarantees for multi-hop wireless networks. We propose a model that jointly considers the end-to-end delay constraints and throughput requirements of flows, the need for multi-hop transmissions, and the unreliable nature of wireless transmissions. We develop scheduling policies for two types of systems and prove that they are optimal. We also conduct extensive simulations to show that our policies outperform other policies by large margins.

I. Introduction

The advance of wireless sensor networks provides an appealing solution for real-time surveillance. In real-time surveillance, wireless sensors generate flows of surveillance data and deliver them to a sink, which makes control decisions based on the data. Examples of such applications have been demonstrated in many previous studies, such as [1]–[3].

A major challenge for real-time surveillance is to provide end-to-end delay guarantees for packet deliveries. Designing scheduling policies that provide end-to-end delay guarantees is difficult due to two reasons. As wireless sensor networks may be deployed over a large area, some flows may require multi-hop transmissions to reach the sink. Further, wireless sensor networks are usually deployed in challenging environments, such as battlefields, forests, or underwater. Within these environments, it may be impossible to ensure that all wireless transmissions can be successfully received. Thus, a desirable policy needs to explicitly address the unreliable nature of wireless transmissions.

There has also been a lot of work that considers end-to-end delay guarantees for wireless sensor networks. Jiang, Ravindran, and Cho [4] have studied the real-time capacity of wireless sensor networks. They approach this problem by decomposing end-to-end delays into per-hop delays, and then study the probability for meeting each per-hop delay independently. Wang et al [5] have used a similar decomposition approach and studied the problem of energy saving while providing end-to-end delay guarantees. Such decomposition approach inevitably leads to suboptimal solutions. Chipara et al [6] have proposed a protocol for scheduling real-time flows by taking interference among sensors into account. Wang et al [7] have investigated the distribution of end-to-end delay in wireless sensor networks. Wang et al [8] have formulated the problem of providing end-to-end delay guarantees as an optimization problem, and have proposed a heuristic for obtaining sub-optimal solutions. Li, Shenoy, and Ramamritham [9] have aimed at providing end-to-end delay guarantees by exploiting spatial reuse. However, these studies fail to provide provable performance guarantees.

In this paper, we aim to address the above difficulties. We measure the performance of each surveillance flow by its *timely-throughput*, defined as the throughput of packets that are delivered to the sink on time. We then propose a model that characterizes the hard per-packet end-to-end delay constraints and timely-throughput requirements of flows, the routing protocol for multi-hop transmissions, and the unreliable wireless channels. This model considers two types of sensors. The first type of sensors, the *orthogonal relay sensors*, can transmit and receive packets simultaneously, while the second type of sensors, the *half-duplex sensors*, cannot. We then establish a sufficient condition for a policy to be optimal based on Lyapunov analysis.

We consider designing online, tractable, and distributed scheduling policies. We propose policies for both systems with orthogonal relay sensors and those with half-duplex sensors. We prove the policy for orthogonal relay sensors is indeed *feasibility-optimal* in the sense that it fulfills all timely-throughput requirements as long as they are feasible. We also show that the policy for half-duplex sensors is feasibility-optimal among certain topologies.

In addition to theoretical studies, we also provide simulation results. We compare our proposed policies against other policies. Simulation results show that our proposed policies achieve significantly better performance than others.

The rest of the paper is organized as follows. Section II formally introduces our analytical model. Based on the framework, Section III proposes a feasibility-optimal policy for systems with orthogonal relay sensors. Section IV proposes a heuristic for systems with half-duplex sensors, and proves that the heuristic is feasibility-optimal for some topologies. Section V demonstrates our simulation results. Finally, Section VI concludes this paper.

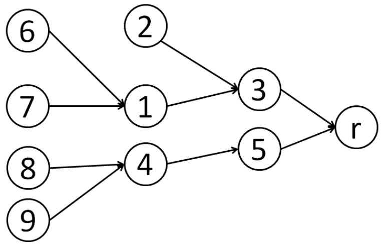


Fig. 1: An example of the system consists of 10 sensors. In the example, we have $h(3) = h(5) = r$, $h(1) = h(2) = 3$, $h(4) = 5$, etc.

II. System Model and Preliminaries

In this section, we present our model for multi-hop wireless sensor networks with end-to-end delay constraints. Our model extends a model proposed in Hou, Borkar, and Kumar [10], which only considers the delay constraints of packets and unreliability of wireless transmissions in a one-hop scenario.

Consider a sensor network with a set N of wireless sensors. One of the sensors plays the role of the sink. Sensors may generate surveillance data that need to be delivered to the sink in a timely manner, and they may relay data that are generated by other sensors. We assume that a routing tree has been constructed by some routing protocol for the sensor network. There has been a lot of work on constructing routing trees for wireless sensor networks, and Al-Karaki and Kamal [11] provides a survey of these routing protocols. In the routing tree, the sink is the root, and hence we use r to represent the sink. When a sensor n has a data packet, either one generated by itself or one that is forwarded to it from other sensors, it may forward the data to its parent, denoted by $h(n)$, in the routing tree. Fig. 1 shows an example of such a sensor network. A data packet is said to be *delivered* if it reaches the sink. A sensor may generate multiple flows of data. For example, one sensor may generate data on both temperature and humidity. We denote the set of flows in the wireless sensor network by F, and $n(f)$ as the sensor that generates data of flow f.

We assume that time is slotted and numbered by $t = 1, 2, \dots$. The length of a time slot is set to be the time needed for a sensor to transmit one data packet. Time slots are further grouped into *intervals*, where each interval consists of T consecutive time slots in $(kT, (k+1)T]$, for some k. At the beginning of each interval, each flow in F obtains some surveillance data and generates a data packet. We say that a data packet of flow f is generated at the τ_f^{th} time slot in each interval, so as to account for the latency caused by sensing and data processing. We assume that all data packets are delay-constrained, and data packets generated in one interval need to be delivered to the sink before the end of the interval. If a data packet is not delivered before the end of the interval, the packet is no longer useful for the sink, and it is dropped from the system.

We consider two types of sensors. The first type of sensors have the capabilities to transmit and receive packets simultaneously. This can be achieved by, for example, employing orthogonal frequency-division multiple access (OFDMA), which allows a node to transmit and receive on two orthogonal channels simultaneously. Many existing work, such as Zhang and Lau [12], has employed such technique for multi-hop transmissions. Some recent advances on full-duplex wireless radios, such as Duarte and Sabharwal [13] and Choi et al [14], allow nodes to transmit and receive simultaneously on the same channel. We call this type of sensors the *orthogonal relay sensors*, and a system consists of this type of sensors an *orthogonal relay system*. We assume that an orthogonal relay sensor also has the capabilities to receive multiple packets at the same time, and different transmissions do not interfere with each other. Both assumptions can be achieved by, for example, employing OFDMA and assigning interfering links on different channels.

The assumptions made for orthogonal relay systems may exceed current hardware limitations of wireless sensor network. Thus, we also consider systems with *half-duplex sensors*. In such systems, sensors cannot transmit and receive data packets simultaneously. That is, when a sensor n transmits, its parent $h(n)$ cannot transmit, or the transmission by n encounters a collision and the transmission fails. Moreover, we assume that a sensor can receive at most one transmission in a time slot. That is, if we have sensors n and m with $h(n) = h(m)$, then at most one of them can transmit in a time slot. Finally, we assume that different transmissions do not interfere with each other except the two cases discussed above. This can be done by, for example, scheduling transmissions that may interfere with each other in different channels. A system with half-duplex sensors is called a *half-duplex system*.

We consider the unreliable nature of wireless transmissions. To be more specific, we say that when a sensor n transmits a data packet to its parent, $h(n)$, $h(n)$ correctly receives the packet with probability p_n. We also assume that, by implementing ACKs, the sensor n has feedback information on whether its transmission is correctly received by $h(n)$, and it can retransmit the same packet in the case that a previous transmission fails.

We measure the performance of a flow by its long-term average number of packets that are delivered on time, which we call the *timely-throughput* of the flow. To be more specific, let $e_f(k)$ be the indicator function that the data packet of flow f in the k^{th} interval is delivered to the sink on time. The *timely-throughput* of flow f is defined as $\liminf_{K\to\infty} \frac{\sum_{k=1}^{K} e_f(k)}{K}$. We assume that each flow f has a certain *timely-throughput requirement*, denoted by q_n, for the integrity of sensed data. That is, each flow f requires that, with probability one, $\liminf_{K\to\infty} \frac{\sum_{k=1}^{K} e_f(k)}{K} \geq q_f$.

In this paper, we aim to design scheduling policies that *fulfills* timely-throughput requirements of all flows as long as they are *strictly feasible*. These terms are formally defined as follows:

Definition 1: A system, either an orthogonal relay system or a half-duplex one, is *strictly feasible* if $q_f > 0$ for all flows f, and there exists a scheduling policy under which the timely-throughput of each flow f is at least $(1+\epsilon)q_f$, for some $\epsilon > 0$.

Definition 2: A scheduling policy is *feasibility-optimal* for orthogonal relay system, or half-duplex system, if it fulfills all strictly feasible orthogonal relay systems, or all strictly feasible half-duplex systems, respectively.

We can define the *debt* of f in the K^{th} interval is defined as $d_f(K) := Kq_f - \sum_{k=1}^{K} e_f(k)$, which indicates the number of packets that a flow is behind its required timely-throughput. The following theorem then establishes a sufficient condition for a policy to be feasibility-optimal.

Theorem 1: A scheduling policy is feasibility-optimal for orthogonal relay system, or half-duplex system, if, given $d_f(k)$, it maximizes $\sum_{f\in F} d_f(k)^+ E[e_f(k+1)]$ in the $(k+1)^{th}$ interval, for all k, where $x^+ := \max\{x, 0\}$, for all orthogonal relay systems, or half-duplex systems, respectively.

Proof: The proof is very similar to the proof of Theorem 3 in [15], which is based on Lyapunov analysis. Hence, the proof is omitted. ■

III. An Online Scheduling Policy for Orthogonal Relay Systems

We now introduce an online scheduling policy for orthogonal relay systems. The policy is very simple: in each time slot, a sensor n picks the flow that has the largest debt among those that it currently holds their packets, and transmits its packet. In other words, in each time slot t of the $(k+1)^{th}$ interval, n schedules the packet from $\arg\max_{f:c_f(t)=n} d_f(k)$. We call this policy the *Greedy Forwarder*. In addition to low complexity, this policy is also distributed. Moreover, as we establish below, this simple policy is indeed feasibility-optimal.

Theorem 2: The Greedy Forwarder is feasibility-optimal.

Proof: By Theorem 1, we can show that the Greedy Forwarder is feasibility-optimal by showing that it maximizes $\sum_{f\in F} d_f(k)^+ E[e_f(k+1)]$. To show this, we prove the following two claims by induction on the size of the network,$|N|$:

1) The Greedy Forwarder maximizes $\sum_{f\in F} d_f(k)^+ E[e_f(k+1)]$.
2) Suppose a sensor generates packets from flow $f_1, f_2, \dots$, where $d_{f_1}(k) \geq d_{f_2}(k) \geq \dots$. Also assume that this sensor generates packets at time $t_1 \leq t_2 \leq \dots$, and the sensor has control over which flow among $\{f_1, f_2, \dots\}$ is to generate packets at time $t_1, t_2, \dots$, respectively. Then, with all other conditions fixed, by selecting f_1 to generate its packet at times t_1, f_2 to generate its packet at time t_2, etc, $\sum_{f\in F} d_f(k)^+ E[e_f(k+1)]$ for the whole system is maximized.

We first discuss the case when $|N| = 1$, in which the sink r is the only sensor in the system. As there is only one sensor in the system, there are no scheduling decisions to be made, and hence Claim (1) holds. Moreover, a packet of flow f is delivered, and hence $e_f(k+1) = 1$, if it is generated before the T^{th} time slot. Thus, Claim (2) holds as $\sum_{f\in F} d_f(k)^+ E[e_f(k+1)]$ is maximized by generating packets in decrement order of their debts.

Assume that both Claim (1) and Claim (2) hold for all networks with size $|N| = M$. We now show that these two claims also hold for all networks with $|N| = M + 1$. We pick a leaf node, denoted by n_0, in the routing tree. We assume that n_0 generates packets for flows $f_1, f_2, \dots$, with $d_{f_1}(k) \geq d_{f_2}(k) \geq \dots$, at times $t_1 \leq t_2 \leq \dots$, and n_0 has control over which flows among $\{f_1, f_2, \dots\}$ is to generate packets at times $t_1, t_2, \dots$, respectively. We also assume that n_0 uses a work-conserving policy, i.e., it always schedules a transmission as long as it holds a packet[1]. Under this policy, n_0 successfully transmits packets at times $\hat{t}_1 \leq \hat{t}_2 \leq \dots$. We note that $\hat{t}_1, \hat{t}_2, \dots$ are random variables whose distributions are determined by the channel reliability between n_0 and $h(n_0)$. Moreover, we have $\hat{t}_i \geq t_i$, for all i, as it is impossible to successfully transmit i packets before at least the same amount of packets are generated. Note that $\hat{t}_1, \hat{t}_2, \dots$ are not influenced by the order of packet generations and scheduling decisions, as each transmission made by n_0 is successful with probability p_{n_0}, regardless which packet is being transmitted.

Given $\hat{t}_1, \hat{t}_2, \dots$, n_0 can effectively determine which packets are to be successfully transmitted at times $\hat{t}_1, \hat{t}_2, \dots$, by choosing the order of packet generations and scheduling decisions. The only restriction for n_0 is that the packet from a flow f cannot be transmitted before it is generated. If the packet from a flow f is successfully transmitted by n_0 at time t, $h(n_0)$ receives the packet at time t and can transmit the packet starting at time $t+1$. Thus, this system is equivalent to one with n_0 removed, making the size of the network to be M, and packets from flows $f_1, f_2, \dots$ are generated at sensor $h(n_0)$ at times $\hat{t}_1 + 1, \hat{t}_2 + 1, \dots$. By the induction hypothesis on this system with M sensors, $\sum_{f\in F} d_f(k)^+ E[e_f(k+1)]$ is maximized if the packet from flow f_i is generated at time $\hat{t}_i + 1$, for all i. As n_0 can make this happen by choosing flow f_i to generate a packet at time t_i and follow the Greedy Forwarder, both Claim (1) and Claim (2) hold when n_0 is included in the system with size $M + 1$.

By induction on $|N|$, we have that both claims hold for all systems, and hence the Greedy Forwarder is feasibility-optimal. ■

We close this section by discussing some implementation issues of the Greedy Forwarder. As noted above, under the Greedy Forwarder, every sensor makes scheduling decisions solely based on the packets it holds. Still, the Greedy Forwarder requires each sensor to have the knowledge of debts for all flows in the current interval. In practice, this may be impractical. In particular, for a large network, sensors that are far away from the sink can only obtain delayed information on debts of flows. However, as we will show in Section V, when sensors apply the Greedy Forwarder with delayed information on debts, the resulting performance can still be optimal. This is because, as the net change of debt, $|d_f(k+1) - d_f(k)|$, is bounded, the difference between the delayed information on debt and the actual current debt is also bounded for all flows. Therefore, a sensor that only has delayed information on debts will make similar scheduling decisions as one that has information on the current values of debts.

IV. A Heuristic for Half-Duplex Systems

In this section, we propose a policy for half-duplex systems and show that this policy is feasibility-optimal for some particular scenarios.

Recall that there are two important limitations for half-duplex systems. First, as a sensor cannot transmit and receive simultaneously, a sensor n cannot transmit when its parent, $h(n)$, is also transmitting. Second, a sensor can only receive one packet at a time, and hence two

[1]Obviously, a policy cannot lose its optimality by making more transmissions. Thus, this assumption is not restrictive.

sensors n and m with $h(n) = h(m)$ cannot transmit simultaneously. To address theses challenges, we propose a policy, namely, the *Closest Sensor First Policy*. We first define $d_n(t) := \max_{f:c_f(t)=n} d_f(k)$. In other words, $d_n(t)$ is the largest debt of flows that n holds at the t^{th} time slot. If the sensor n does not hold any packets, $d_n(t)$ is defined to be $-\infty$. The Closest Sensor First Policy can be described iteratively as follows: First, we examine all sensors that are one-hop away from the root r, that is, we examine all sensors such that $h(n) = r$. We then schedule the sensor with the largest $d_n(t)$ to transmit, and the sensor transmits the packet from the flow with the largest debt. Next, we examine sensors that are two-hop away from the root, that is, sensors with $h(h(n)) = r$. If $h(n)$ is scheduled to transmit in the first step, then n cannot transmit. Otherwise, a sensor n is scheduled to transmit the packet from the flow with the largest debt if $d_n(t)$ is the largest among all sensors m with $h(m) = h(n)$. In the above procedure, ties are broken arbitrarily, and we carry the procedure iteratively.

In summary, a sensor n who is g-hop away from the root is scheduled in the g^{th} iteration if: (i) $h(n)$ is not scheduled in the previous iteration, and (ii) $d_n(t)$ is the largest among all m with $h(m) = h(n)$. Consider the system in Fig. 1 as an example. Assume that $d_n(t) = n, \forall n$. In the first iteration, sensor 5 is scheduled as $d_5(t) > d_3(t)$. In the second iteration, sensor 2 is scheduled as $d_2(t) > d_1(t)$. Note that sensor 4 cannot be scheduled as its parent, sensor 5, has already been scheduled. Finally, in the third iteration, sensor 7 and sensor 9 are scheduled.

Next, we show that the Closest Sensor First Policy is feasibility-optimal if all flows are generated by the same sensor n_0, i.e., $n(f) = n_0$, for all f, and each flow generates a packet at the beginning of the interval, i.e., $\tau_f = 1$, for all f. In such a system, only sensors on the path between n_0 and r are involved in forwarding messages. Thus, we call such a system as a *path-topology system*.

Theorem 3: The Closest Sensor First Policy is feasibility-optimal for path-topology systems.

Proof: By Theorem 1, we can show that the Closest Sensor First Policy is feasibility-optimal for path-topology systems by establishing that this policy maximizes $\sum_{f\in F} d_f(k)^+ E[e_f(k+1)]$ in the $(k+1)^{th}$ interval.

Let $\gamma_n[v]$ be the number of transmissions that sensor n needs to make to successfully transmit v packets to its parent. Note that this does not imply that sensor n successfully transmits v packets at the $\gamma_n[v]^{th}$ time slot in the interval, as there are time slots that sensor n is not scheduled due to the constraints of half-duplex systems. We also note that $(\gamma_n[v] - \gamma_n[v-1])$ is a geometric random variable with mean $1/p_n$, as the channel reliability between n and $h(n)$ is p_n. In practice, the values of $\{\gamma_n[v]\}$ cannot be obtained at the beginning of the interval. We will show that, even when the values of $\gamma_n[v]$ are given for all n and v, there is no policy that can achieve larger $\sum_{f\in F} d_f(k)^+ e_f(k+1)$ than the Closest Sensor First Policy, and hence than the Closest Sensor First Policy maximizes $\sum_{f\in F} d_f(k)^+ E[e_f(k+1)]$.

We order flows so that $d_{f_1}(k) \geq d_{f_2}(k) \geq \dots$. Let η and η^* be the Closest Sensor First Policy and another policy that maximizes $\sum_{f\in F} d_f(k)^+ E[e_f(k+1)]$, respectively. Further, let θ_{f_i} and $\theta^*_{f_i}$ be the times that the packet from flow f_i is delivered under η and η^*, respectively. If the packet from flow f_i is not delivered on time under η, or η^*, we set θ_{f_i}, or $\theta^*_{f_i}$, to be $T+1$. Thus, under η, or η^*, we have $e_{f_i}(k+1) = 1(\theta_{f_i} < T+1)$, or $e_{f_i}(k+1) = 1(\theta^*_{f_i} < T+1)$, respectively.

By the design of the Closest Sensor First Policy, we have $\theta_{f_1} \leq \theta_{f_2} \leq \dots$. Suppose there exists some i so that $\theta^*_{f_i} > \theta^*_{f_{i+1}}$ under η^*. We can modify η^* so that whenever it schedules the packet from flow f_i, it schedules the packet from flow f_{i+1} instead, and vice versa. Under this modification, the packet of f_i is delivered on the $(\theta^*_{f_{i+1}})^{th}$ time slot, and the packet of f_{i+1} is delivered on the $(\theta^*_{f_i})^{th}$ time slot. If both $\theta^*_{f_i}$ and $\theta^*_{f_{i+1}}$ are smaller than $T+1$, both packets are still delivered on time after this modification, and hence the value of $\sum_{f\in F} d_f(k)^+ e_f(k+1)$ is not influenced. If both $\theta^*_{f_i}$ and $\theta^*_{f_{i+1}}$ are larger than T, neither packets are delivered on time after this modification, and the value of $\sum_{f\in F} d_f(k)^+ e_f(k+1)$ is not influenced. However, if $\theta^*_{f_i} > T \geq \theta^*_{f_{i+1}}$, the packet of f_i is delivered on time and the packet of f_{i+1} is not after the modification, and the value of $\sum_{f\in F} d_f(k)^+ e_f(k+1)$ will not decrease, as $d_{f_i}(k) \geq d_{f_{i+1}}(k)$, with the modification. In sum, the value of $\sum_{f\in F} d_f(k)^+ e_f(k+1)$ will not decrease with the modification. Thus, we can repeat this procedure until $\theta^*_{f_1} \leq \theta^*_{f_2} \leq \dots$ without decreasing the value of $\sum_{f\in F} d_f(k)^+ e_f(k+1)$.

From now on, we assume that $\theta^*_{f_1} \leq \theta^*_{f_2} \leq \dots$ under η^*. We claim that, under this assumption, $\theta_{f_i} \leq \theta^*_{f_i}$ for all i. We prove this claim by induction on the number of flows. When there is only one flow in the system, the Closest Sensor First Policy schedules a transmission for flow 1 in every time slot, and hence $\theta_{f_1} \leq \theta^*_{f_1}$.

Assume that $\theta_{f_i} \leq \theta^*_{f_i}$ for all i when the system has I flows. We now consider the case when the system has $I+1$ flows. Under the Closest Sensor First Policy, whether the packet of a flow f_i with $i \leq I$ is scheduled is not influenced by whether the flow f_{I+1} is present in the system. Thus, the value of θ_{f_i} is the same as in the case when the system only has I flows, for all $i \leq I$. We then have $\theta_{f_i} \leq \theta^*_{f_i}$ for all $i \leq I$ by the induction hypothesis. Therefore, we only need to prove that $\theta_{f_{I+1}} \leq \theta^*_{f_{I+1}}$. If $\theta^*_{f_{I+1}} = T+1$, i.e., the packet of flow f_{I+1} is not delivered on time under η^*, then $\theta_{f_{I+1}} \leq \theta^*_{f_{I+1}}$ holds.

Consider the case $\theta^*_{f_{I+1}} \leq T$. Suppose that, under η^*, there is some time during the interval when the packet from flow f_{i+1} is closer to the root than the packet from flow f_i, for some i. Pick i' to be the smallest number so that the packet from flow $f_{i'+1}$ is closer to the root than the packet from flow $f_{i'}$ at some time t during the interval. Now we can pick t_1 to be the largest time before t such that the packet from flow $f_{i'+1}$ and that from flow $f_{i'}$ are held by the same sensor. Such t_1 exists as both packets are held by the sensor that generates all packets at the beginning of the interval. We then pick t_2 to be the smallest time after t such that both packets are held by the same sensor. Such t_2 exists as we assume that the packet from flow $f_{i'}$ is delivered earlier than the packet from flow $f_{i'+1}$. Thus, in any time slot in (t_1, t_2), the packet from flow $f_{i'+1}$ is always closer to the root than that from flow $f_{i'}$. Now, we can modify η^* for time slots in $[t_1, t_2]$ so that when it schedules i', it schedules $i'+1$ instead, and vice versa. After this modification, the packet from flow $f_{i'}$ is always closer to the root than that from flow $f_{i'+1}$ during (t_1, t_2). Further, this modification does not influence θ_f for any flow f. We repeat this modification until such i' does not exist. From now on, we can assume that, at any point of time, the packet from flow f_i is not closer to the root than the packet from flow f_j if $j < i$.

We prove that $\theta_{f_{I+1}} \leq \theta^*_{f_{I+1}}$ by contradiction. Note that after the packet of flow f_I is delivered, η, or η^*, needs to schedule the packet of flow f_{I+1} an addition number of $(\theta_{f_{I+1}} - \theta_{f_I})$, or $(\theta^*_{f_{I+1}} - \theta^*_{f_I})$, times before it is delivered, respectively. By the induction hypothesis, $\theta_{f_I} \leq \theta^*_{f_I}$. Therefore, if $\theta_{f_{I+1}} > \theta^*_{f_{I+1}}$, we have $\theta_{f_{I+1}} - \theta_{f_I} > \theta^*_{f_{I+1}} - \theta^*_{f_I}$, and, under η^*, there are times that the packet from flow f_{I+1} is scheduled while it would not be schedule under η. We call these times the *inversion times* and denote them by $\hat{t}_1 < \hat{t}_2 < \cdots < \hat{t}_M$. We assume that, among all policies that deliver packets at times $\theta^*_{f_1}, \theta^*_{f_2}, \dots$, η^* is one that has the smallest number of inversion times. Moreover, among those policies that have the smallest number of inversion times, η^* is one that maximizes $\hat{t}_M$.

At time $\hat{t}_M$, the sensor $c_{f_{I+1}}(\hat{t}_M)$ holds the packet from flow f_{I+1} and schedules it under η^*, while $c_{f_{I+1}}(\hat{t}_M)$ would not schedule this packet under η. There are two possibilities: first, the sensor $c_{f_{I+1}}(\hat{t}_M)$ holds another packet, and η would schedule it; and second, under η, the sensor $c_{f_{I+1}}(\hat{t}_M)$ would not transmit, as its parent, $h(c_{f_{I+1}}(\hat{t}_M))$, would be scheduled for transmission. In the first case, under η^*, the transmission of f_{I+1} cannot be successful. Otherwise, at time $\hat{t}_M + 1$, the packet of f_{I+1} is closer to the root than the other packet that $c_{f_{I+1}}(\hat{t}_M)$ holds, and violates our previous assumption. Thus, for this case, we can modify η^* so that $c_{f_{I+1}}(\hat{t}_M)$ schedules the other packet, and this modification will not influence the deliver times of packets. In this case, there are no inversion times at and after time $\hat{t}_M$ after the modification. As this modification does not create new inversion times, we obtain a policy that has a smaller number of inversion times than η^*, which contradicts our assumption in the last paragraph. Now consider the second case, that is, the sensor $c_{f_{I+1}}(\hat{t}_M)$ would not be scheduled by η because η would schedule its parent for transmission. As $\hat{t}_M$ is the largest inversion time, there will be a time after $\hat{t}_M$ such that $h(c_{f_{I+1}}(\hat{t}_M))$ is scheduled to transmit a packet, and the packet from flow f_{I+1} is not scheduled. We call this time $\hat{t}'$. At time $\hat{t}'$, either sensor $c_{f_{I+1}}(\hat{t}_M)$ or $h(c_{f_{I+1}}(\hat{t}_M))$ holds the packet from flow f_{I+1}. We can modify the schedule so that $h(c_{f_{I+1}}(\hat{t}_M))$ is scheduled for transmission at time $\hat{t}_M$, instead of $c_{f_{I+1}}(\hat{t}_M)$, and the packet from flow f_{I+1} is scheduled for transmission at time $\hat{t}'$. Note that, by our previous assumption, the packet from flow f_{I+1} is not closer to the sink than any other packets. In other words, every sensor that is farther from the sink than the one holds the packet from flow f_{I+1} does not hold any packets. Therefore, this modification will not violate any interference constraints of the half-duplex system. This modification does not influence the delivery times of packets and does not increase the number of inversion times. Moreover, after applying the modification, the largest inversion time becomes $\hat{t}'$, which contradicts our assumption that η^* maximizes the largest inversion time.

In summary, we have established that $\theta_{f_i} \leq \theta^*_{f_i}$ for all flows when there are $I+1$ flows in the system. By induction, we have $\theta_{f_i} \leq \theta^*_{f_i}$ for all flows, for all path-topology systems. Therefore, the Closest Sensor First Policy is feasibility-optimal for path-topology systems. ■

V. Simulation Results

In this section, we present our simulation results. We adopt the simulation settings in [9], where each flow generates one packet every 20 ms, and it takes 2 ms for a sensor to make a transmission. Thus, we set the duration of a time slot to be 2 ms, and each interval consists of 10 time slots.

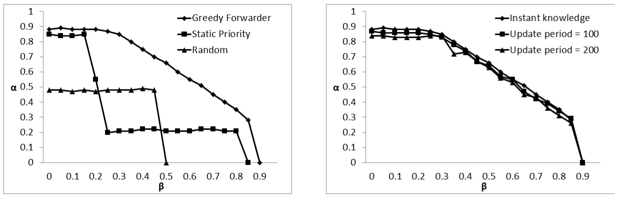
We consider the network topology as shown in Fig. 1. For each sensor n, its channel reliability, p_n, is randomly selected within $[0.4, 0.9]$. We assume that each of sensor 3, sensor 5, sensor 6, sensor 7, sensor 8, and sensor 9 generates two flows. Therefore, there are a total number of 12 flows in the system. To better illustrate our simulation results, we assume that for each of these sensors, one flow requires a timely-throughput of α, and the other requires a timely-throughput of β. We define the *timely-throughput region* of a policy to be the region consists of all (α, β) that can be fulfilled by the policy. We can then evaluate the performance of a policy by its timely-throughput region.

For all scenarios, we conduct the simulation for 3000 intervals, i.e., one minute in the simulation environment. A system is said to be fulfilled if, by the end of the simulation, the debts are less than 90 for all flows, which means the actual timely-throughput that a flow has is at least $(q_f - 0.03)$.

We consider both the orthogonal relay system and half-duplex system. We compare our proposed policies against two other policies. One policy, the Random policy, randomly selects a maximal set of packets that can be transmitted simultaneously under the restrictions of the system in each time slot. The other policy, the Static Priority policy, sorts all flows according to their timely-throughput requirements, with ties broken randomly. In each time slot, the Static Priority policy greedily selects a maximal set of packets that can be transmitted simultaneously. We note that, in half-duplex systems, the Static Priority policy is the same as the policy proposed in [6].

The simulation results for the orthogonal relay system is shown in Fig. 2a. Our proposed policy, the Greedy Forwarder, achieves the largest timely-throughput region, as it is indeed feasibility-optimal. The performance of the Static Priority policy is close to optimal when either α is much larger than β, or vice versa, as it gives higher priorities to flows with larger timely-throughput requirements. On the other hand, the Static Priority policy inevitably starves flows with smaller timely-throughput requirements. Thus, it results in poor performance when α is close to β. The performance of the Random policy is also far from optimal, as it does not take the timely-throughput requirements of flows into account.

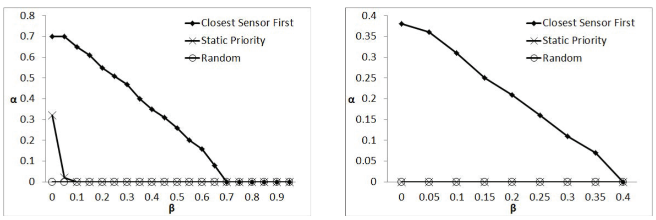
We also investigate the influence on the Greedy Forwarder when sensors only have delayed information on debts of flows. We assume that all sensors other than the sink update their information on debts every λ intervals, and we call λ the *update period*. When a sensor n updates its information, it notifies its children in the routing tree the information on debts that it currently has, and



(a) Performance comparison between different policies.

(b) Results for systems where sensors only have delayed information on debts.

Fig. 2: Simulation results for the orthogonal relay system.



(a) Results for the general topology.

(b) Results for the path-topology.

Fig. 3: Simulation results for the half-duplex system.

receives an updated information from its parent. Thus, for a sensor that is g-hop away from the sink, the information on debts that it has may be $\lambda \times g$ intervals old. We consider three scenarios: one where all sensors have knowledge of the current debts of flows, one where the update period is 100 intervals, i.e., 2 seconds, and one where the update period is 200 intervals. Simulation results are presented in Fig. 2b. It can be shown that even when sensors update their information on debts as infrequent as once every four seconds, the performance of the Greedy Forwarder is still close to optimal.

Next, we consider the half-duplex system. The simulation results of the half-duplex system is shown in Fig. 3a. The Closest Sensor First policy, which is our proposed policy, achieves the largest timely-throughput region. This suggests that the Closest Sensor First policy can still achieve good performance for general topologies, even though it is only proved to be feasibility-optimal for path-topology systems. A somewhat surprising result is that both the Random policy and the Static Priority policy have very poor performance. The Random policy fails to fulfill the system even when we set $(\alpha, \beta) = (0, 0.05)$. The reason for this behavior is because there are interference constraints for half-duplex systems, which limit the number of sensors that can transmit simultaneously. Thus, it is important to take the network topologies into account in order to deliver packets on time.

Finally, we consider a half-duplex path-topology system with six sensors and six flows, as depicted in Fig. 4. All six flows are generated by sensor 5. Three of the flows require a timely-throughput of α, while the other three flows require a timely-throughput of β. The simulation results are shown in Fig. 3b. The Closest Sensor First policy achieves the largest timely-throughput region. Moreover, both the Static Priority policy and the Random policy fail to fulfill the system even when we set $(\alpha, \beta) = (0, 0.05)$.

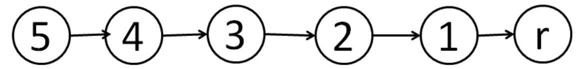


Fig. 4: The network topology for the half-duplex path-topology system.

VI. Concluding Remarks

We have investigated the problem of providing hard per-packet delay guarantees for multi-hop wireless sensor networks. We have proposed an analytical model that jointly considers the hard delay guarantees of packets, the multi-hop routing tree of the network, the timely-throughput requirements of flows, and the unreliable nature of wireless transmissions. The model can be applied for both orthogonal relay systems and half-duplex systems. We have proposed a distributed scheduling policy for orthogonal relay systems and proved that this policy is feasibility-optimal. We have also proposed a heuristic for half-duplex systems. We have proved that this heuristic is feasibility-optimal for all path-topology systems. Simulation results have suggested that our proposed policies achieve much better performance than other policies.

References

[1] I. Stoianov, L. Nachman, S. Madden, and T. Tokmouline, "Pipenet: A wireless sensor network for pipeline monitoring," in *Proc. of ACM IPSN*, pp. 264–273, 2007.

[2] S. Oh, P. Chen, M. Manzo, and S. Sastry, "Instrumenting wireless sensor networks for real-time surveillance," in *Proc. of IEEE ICRA*, pp. 3128–3133, 2006.

[3] X. Zhu, S. Han, P.-C. Huang, A. K. Mok, and D. Chen, "Mbstar : A real-time communication protocol for wireless body area networks," in *Proc. of ECRTS*, pp. 57–66, 2011.

[4] B. Jiang, B. Ravindran, and H. Cho, "On real-time capacity of event-driven data-gathering sensor networks," in *Proc. of ACM MobiQuitous*, 2009.

[5] X. Wang, X. Wang, G. Xing, and Y. Yao, "Dynamic duty cycle control for end-to-end delay guarantees in wireless sensor networks," in *Proc. of IEEE IWQoS*, 2010.

[6] O. Chipara, C. Wu, C. Lu, and W. Griswold, "Interference-aware real-time flow scheduling for wireless sensor networks," in *Proc. of ECRTS*, pp. 67–77, 2011.

[7] Y. Wang, M. C. Vuran, and S. Goddard, "Cross-layer analysis of the end-to-end delay distribution in wireless sensor networks," in *Proc. of IEEE RTSS*, pp. 138–147, 2009.

[8] Q. Wang, P. Fan, D. O. Wu, and K. B. Letaief, "End-to-end delay constrained routing and scheduling for wireless sensor networks," in *Proc. of IEEE ICC*, 2011.

[9] H. Li, P. Shenoy, and K. Ramamritham, "Scheduling messages with deadlines in multi-hop real-time sensor networks," in *Proc. of IEEE RTAS*, pp. 415 – 425, 2005.

[10] I.-H. Hou, V. Borkar, and P. R. Kumar, "A theory of QoS for wireless," in *Proc. of INFOCOM*, pp. 486–494, 2009.

[11] J. Al-Karaki and A. Kamal, "Routing techniques in wireless sensor networks: a survey," *IEEE Wireless Communications*, vol. 11, pp. 6–28, Dec. 2004.

[12] S. Zhang and V. K. N. Lau, "Resource allocation for ofdma system with orthogonal relay using rateless code," *IEEE Transactions on Wireless Communications*, vol. 7, Nov. 2008.

[13] M. Duarte and A. Sabharwal, "Full-duplex wireless communications using off-the-shelf radios: Feasibility and first results," in *Proc. of ASILOMAR*, pp. 1558–1562, 2010.

[14] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, and S. Katti, "Achieving single channel, full duplex wireless communication," in *Proc. of ACM MobiCom*, 2010.

[15] I.-H. Hou and P. R. Kumar, "Scheduling heterogeneous real-time traffic over fading wireless channels," in *Proc. of IEEE INFOCOM*, 2010.