Real-Time Convergecast Scheduling in Lossy Multi-hop Wireless Sensor Networks

Xiaohui Liu
Wayne State University
xiaohui@wayne.edu

Hongwei Zhang
Wayne State University
hongwei@wayne.edu

***Abstract*—We study the problem of scheduling packet transmissions to maximize the expected number of packets collected at the sink by a deadline in a multi-hop wireless sensor network with lossy links. Most existing work assumes error-free transmissions when interference constraints are complied, yet links can be unreliable due to external interference, shadowing, and fading in harsh environments in practice. We formulate the problem as a Markov decision process, yielding an optimal solution. However, the problem is computationally intractable due to the curse of dimensionality. Thus, we propose the efficient and greedy Best Link First Scheduling (BLF) protocol. We prove it is optimal for the single-hop case and provide an approach for distributed implementation. Extensive simulations show it greatly enhances real-time data delivery performance, increasing deadline catch ratio by up to 50%, compared with existing scheduling protocols in a wide range of network and traffic settings.**

I. Introduction

As wireless sensor networks (WSNs) mature, more and more of them are being deployed for real-time applications such as intruder detection, structure monitoring, and patient monitoring. In many such applications, it is imperative that sensed data distributed across a large region is reported to a common sink (a traffic pattern called convergecast) by a deadline, so some actions can be taken promptly, for example, sounding an alarm when an intruder is detected. Even if data reaches the sink after the deadline, it is useless and discarded. In the previous example, the intruder may be beyond range of pursuit.

Existing work in real-time WSNs usually assumes that interference-free transmissions are error-free based on some interference models. Nevertheless, WSNs are often deployed in harsh, even hazardous and hostile, environments such as forests, volcanoes, and battlefields. Besides, low-power wireless communication of sensors is susceptible to external interference, shadowing, and fading. It is thus extremely difficult, if possible at all, to ensure every transmission is successful. Despite much work in real-time WSNs, the effects of unreliable wireless links on real-time data delivery are not well studied, especially in multi-hop scenarios.

In this paper, we aim to fill the gap of delivering real-time data in lossy multi-hop WSNs. More specifically, given a collection tree rooted at the sink and the number of packets queued at each node, we study the problem of finding an optimal scheduling algorithm that maximizes the expected total number of packets delivered to the sink by a deadline, subject to interference constraints. Given the packets queued in the network initially, this is equivalent to maximizing the overall deadline catch ratio (DCR).

Recently, there is some work studying real-time scheduling over unreliable links in multi-hop WSNs [1]. However, it focuses on meeting real-time performance requirement for each individual flow/node. By contrast, we are concerned with the aggregate data gathered at the sink by a deadline. This is motivated by the observation that nodes in a WSN are usually designed to collectively accomplish a common task, e.g., detecting and tracking an intruder. The end results depend on the aggregate data gathered, not necessarily the specific amount of data from each individual node.

Our problem definition and solutions encompass two major types of real-time data collection applications: event detection and periodic sampling. In the former, an event of interest occurs. Some nodes in the vicinity detect it and generate data to notify the sink quickly to respond to the event. In the latter, nodes sense the environment in each period and generate several packets to be delivered to the sink, which processes it immediately to take certain actions depending on application requirements. In periodic sampling applications, we assume the relative deadline is no larger than the period. Thus, it suffices to run our scheduling in every period to maximize the long-term deadline catch ratio.

Contributions of this paper. Towards the objective, we make the following contributions.

1) We formulate the problem into a finite horizon Markov decision process in Section III, which enables us to compute an optimal scheduling policy based on value iteration. However, plagued by the curse of dimensionality, solving the Markov decision problem becomes computationally prohibitive when the network grows large.
2) We propose an efficient greedy scheduling algorithm called Best Link First Scheduling (BLF) to circumvent the intractability of the Markov decision process formulation in Section IV. After proving a simple greedy scheduling policy is optimal for single-hop WSNs, we extend it to multi-hop settings. The basic idea of BLF is simple: always pick the best quality links first in a top-down manner. It has the salient feature of directly taking link unreliability into consideration. We also propose

one approach for distributed implementation.

3) We carry out extensive simulations to demonstrate BLF's considerable performance improvements over existing solutions, in a wide range of network and traffic settings in Section V. BLF is consistently better and its advantage is more obvious when real-time requirements are stringent.

II. Preliminaries

A. Network and Traffic Models

We consider a multi-hop wireless network of n nodes, where node 0 is the sink. Each node is equipped with a half-duplex radio, which can either transmit or receive a packet at a time. A tree rooted at the sink is constructed by some routing protocols. When node i transmits a packet to its parent, denoted by $h(i)$, $h(i)$ receives it with probability $p(i)$, which we refer to as the quality of the link from i to $h(i)$ hereafter. Figure 1 illustrates an example. The number on each link from node i to $h(i)$ represents its quality $p(i)$. We assume that transmissions along each link and across different links are independent. We also assume transmissions along links that do not share nodes do not interfere with each other as in [1], i.e., we use the primary interference model. This can be achieved by, for instance, separating them in different channels. The absence of secondary interference enables us to isolate the effects of link unreliability on real-time packet delivery.

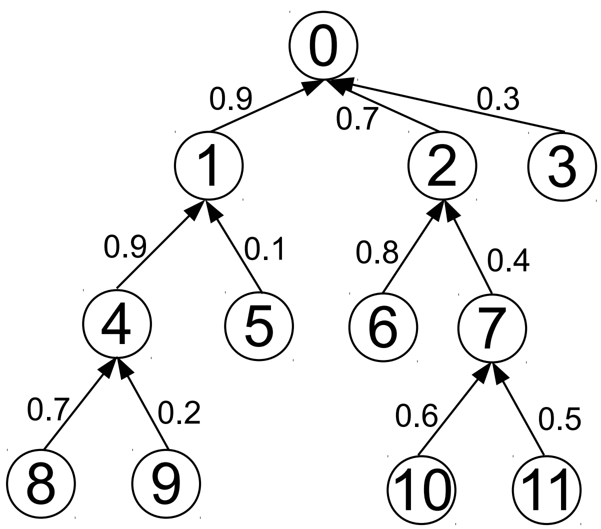

Fig. 1: An example of collection tree consisting of 12 nodes with node 0 as the root

Time is slotted and a node can transmit one packet within a slot. At time 0, node i has v_i ($v_i \geq 0$) packets to be delivered to the sink by deadline D. Note a node can generate multiple packets because it is equipped with multiple sensors for different parameters, each generating one packet. Or its sampling frequency is different from others'. Also, node i can act as pure relay and does not generate any packet itself, i.e., $v_i = 0$.

B. Problem Definition

Given $\mathbf{h} = (h_1, h_2, ..., h_{n-1})$, $\mathbf{p} = (p_1, p_2, ..., p_{n-1})$, and $\mathbf{v} = (v_1, v_2, ..., v_{n-1})$, our objective is to design a scheduling algorithm that maximizes the expected total number of packets reaching the root by deadline D, respecting the primary interference model. Note this optimality is different from the feasibility optimality in [1]. The feasibility optimality means fulfilling the DCR requirement of each individual flow if there exists a scheduling policy that can do so, while our optimality is concerned with maximizing the aggregate DCR of all flows. We do not consider in-network aggregation of packets in this paper.

III. A Mathematical Framework for Optimal Scheduling

Finding an optimal schedule involves sequential decision making in every slot, whose outcomes are stochastic in nature due to link losses. This naturally leads to formulating the aforementioned problem as a Markov decision process (MDP). The MDP's 4-tuple (S, A, P, R) are instantiated as follows.

1) State space S: a state $s \in S$ is the number of packets queued at each node.
2) Action space A: an action $a \in A$ is to schedule a set of nodes to transmit.
3) Transition probability P: P^a_{ij} is the probability state i changes to j when action a is taken. When node m is scheduled to transmit, a packet goes from m to its parent h_m with probability p_m; it stays with probability $1 - p_m$. Since transmissions across different links are independent, we can calculate P^a_{ij} as the product of the probability of each individual transmission success/failure, given the set of transmissions in action a.
4) Reward R: $R(i, a)$ is the expected reward obtained in state i when action a is taken. In each slot, a reward of 1 is gained if a packet reaches root; 0 otherwise.

A policy π is a sequence of decisions that determine which action to take at a given state. Based on this formulation, our objective translates into finding an optimal scheduling policy maximizing the expected total rewards over finite horizon D. An optimal policy can be found by the value iteration algorithm as follows:

$$V_k(i) = \max_a \{R(i, a) + \sum_j P^a_{ij} V_{(k-1)}(j)\} \quad (1)$$

$$V_0(i) = 0 \quad (2)$$

$V_k(i)$ is the k-step-to-go value of state i, i.e., the maximal reward one expects to get if one starts in state i and proceeds for k steps. The optimal policy π^* is $\{a^*_1, a^*_2, ..., a^*_D\}$ with initial state $\mathbf{v}$, in which a^*_k is the a maximizing the right hand side of Equation 1. The MDP formulation enables us to solve the original problem optimally. Unfortunately, it suffers from the curse of dimensionality, where solving for the MDP becomes computationally prohibitive when the network grows large. To see why, consider a simple case of line topology where each node has one packet, the computational complexity is $\mathcal{O}((n!)^2 2^n D)$.

IV. Best Link First Scheduling

To overcome the shortcoming of MDP, we propose an efficient greedy scheduling algorithm, Best Link First Scheduling. We demonstrate its underlying intuition in a simplified single-hop setting. After detailing the centralized BLF, we discuss one approach for distributed implementation.

A. Single-hop Scenario

Consider a one-hop network in Figure 2. Without loss

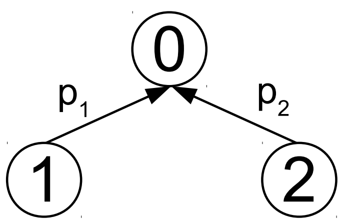

Fig. 2: A single-hop tree with 3 nodes

of generality, let us assume that $p_1 \geq p_2$. Suppose that nodes 1 and 2 each has one packet and it takes them X_1 and X_2 transmissions to successfully deliver their packet to root 0, respectively. X_1 and X_2 are geometric random variables with parameters p_1 and p_2, respectively. Nodes 1 and 2 cannot be active in the same slot. It is obvious that an optimal scheduling policy must be work-conserving, i.e., always transmitting whenever there are packets left. Thus, we restrict our discussion on work-conserving policies. Let us consider the following two policies.

- π_1: node 1 transmits first. After node 1's packet is delivered, node 2 transmits.
- π_2: node 2 transmits first. After node 2's packet is delivered, node 1 transmits.

Under π_1, root 0 receives the first packet at slot X_1, the second $X_1 + X_2$. Under π_2, root 0 receives the first packet at slot X_2, the second $X_2 + X_1$. The second packet arrives at the same time in two policies. We focus on the first packet. Given the same deadline D, let $Pr\{X_1 \leq D\}$ and $Pr\{X_2 \leq D\}$ denote the probability that the first packet arrives by D under π_1 and π_2, respectively. Since $p_1 \geq p_2$, $Pr\{X_1 \leq D\} = 1 - Pr\{X_1 > D\} = 1 - (1-p_1)^D \geq 1 - (1-p_2)^D = 1 - Pr\{X_2 > D\} = Pr\{X_2 \leq D\}$. The expected number of packets collected by D under π_1 is no less than that under π_2.

Similarly, it can be shown that other policies where the transmissions of node 1 and node 2 interleave are no better than π_1. This is because any transmission of node 2 before node 1's successful packet delivery delays it. Thus, π_1 is optimal. In fact, this argument holds even when there are more than two children and some of them have multiple packets as Lemma 1 shows.

Lemma 1. *If $h_i = 0$ for every $i > 0$, the policy π' that always schedules the non-empty node with the best quality link to the root is optimal.*

Proof. This result follows from Theorem 1 in [6]. It is also based on translating the problem into an MDP as in Section III. The difference is that packets are weighted and a reward of α_i is obtained when a packet from flow i is delivered. We let root collect data from its children, instead of transmitting data to them. After this reversal of traffic, our problem in single-hop settings turns out to be a special case of the general real-time multicast problem studied in [6], where each multicast flow degenerates into a unicast flow and is assigned the same weight (i.e., $\alpha_i = 1$ for all i). The optimal GreedyU policy that serves the user with the maximum expected weighted throughput $\alpha_i p_i$ among all unsatisfied users becomes π' above. □

B. Multi-hop Scenario

Inspired by the above result in single-hop networks, we propose Best Link First Scheduling for multi-hop networks. The core idea is to still schedule the best link first, as we do in single-hop networks, and then add more links as long as interference constraints are not violated, taking advantage of spatial reuse in a multi-hop network. When multiple links can be added, we add links closer to the root first since their packets are closer and are more likely to arrive before deadline expires. Algorithm 1 details BLF. Ties are broken arbitrarily.

Algorithm 1 Best Link First Scheduling

Input: a collection tree $\mathbf{h}$, its link qualities $\mathbf{p}$, and node queue levels $\mathbf{v}$

Output: a slot schedule $\mathcal{S}$

$\mathcal{S} = \emptyset$
k = height of $\mathbf{h}$
for i = 0 to $k-1$ **do**
  $\mathcal{A} = \{a \mid a$ is at level i and $h_a \notin \mathcal{S}\}$
  for every node $a \in \mathcal{A}$ **do**
    $\mathcal{B} = \{b \mid h_b = a$ and $v_b > 0\}$
    $b^* = \underset{b \in \mathcal{B}}{\operatorname{argmax}}\, p_b$
    $\mathcal{S} = \mathcal{S} \cup \{b^*\}$
  end for
end for
return $\mathcal{S}$

Another perspective to look at BLF is through recursion. Starting from the root, we include the non-empty child with the best quality link to it. For all other children, we repeat the same process on the subtree rooted at them recursively, using them as the new roots. For the chosen one, since its children cannot be scheduled, we instead repeat the above process on subtrees rooted at its children. BLF is greedy in the sense that in every slot, it schedules a set of nodes that maximizes the expected number of packets deliverable in that slot. BLF runs in $\mathcal{O}(nD)$ since in every slot, each link is visited once to identify the best one among those sharing a common receiver in the worst case.

As an example, let us consider the tree in Figure 1. Assume that every node has at least one packet. At level 1, nodes 1, 2, and 3 share common parent node 0. Node 1 is scheduled since it has the best quality link to node 0, i.e., $p_1 > p_2 > p_3$. At level 2, nodes 4 and 5 cannot be scheduled since their parent, node 1, has been scheduled. Out of the two children of node 2, which is not scheduled, node 6 is selected since $p_6 > p_7$. Similarly, nodes 8 and 10 are chosen at level 3. Together, nodes 1, 6, 8, and 10 are scheduled to transmit as Figure 3 shows.

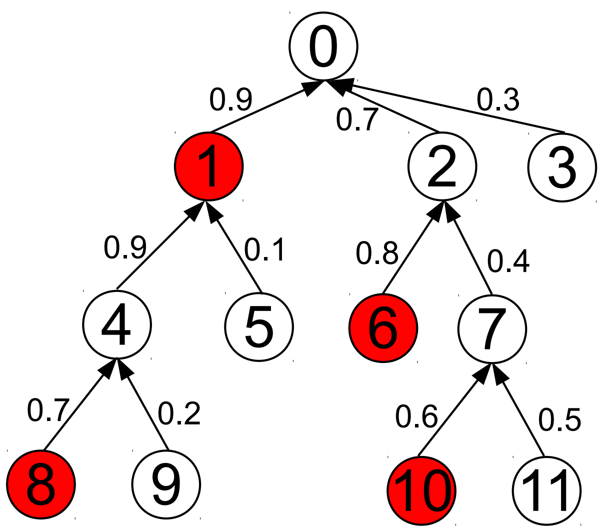

Fig. 3: Highlighted nodes are scheduled

C. *Distributed Implementation*

Algorithm 1 is a centralized algorithm. It is amenable for distributed implementation. One straightforward way to implement the distributed algorithm is as follows: precede from top to bottom in rounds to determine the set of nodes to transmit in a slot. In the first round, each child of the root tells the root whether it has any packet. This can be attained by, for example, letting each child transmit in round-robin. The root, upon receiving information from all its children, decides the non-empty child with the best link to it. It announces the decision to all its children. In the second round, nodes at level 1 and not chosen become roots and repeat the same process. This goes on till the last level but one. In the end, all chosen nodes transmit in that slot. We assume that in the bootstrap phase, each node obtains the qualities of the links to its parent and from its children through certain link estimation process. Measures can be taken to ensure reliable control signaling between a parent and a child, such as retransmissions based on known link quality. We delegate more efficient implementations to future work.

V. Evaluation

A. *Experimental Settings*

Protocols. We develop BLF and state of the art and compare them through MATLAB simulations in a wide range of network and traffic settings. The performance metric is deadline catch ratio, defined as the percentage of packets reaching the root by deadline out of all the packets in the network initially.

- LDF: we use the Closest Sensor First scheduling in [1] to represent Largest Debt First (LDF) scheduling. LDF schedules the largest debt flow first. The debt of a flow is defined as the number of timely deliveries the flow is lagging behind to satisfy its DCR requirement. For fair comparison, we set the DCR requirement of each flow as the aggregate DCR in BLF.
- CR-SLF: the Channel Reuse-based Smallest latest-start-time (LST) First (CR-SLF) protocol [3] applies techniques from traditional real-time systems theory to real-time scheduling in multi-hop wireless networks. The LSF of a flow can be regarded as its laxity, and CR-SLF is essentially least laxity first scheduling. We set a link's delay as its expected transmission count (ETX) [4] in the implementation.
- LBF: the Largest Branch First scheduling delivers packets in the largest branch/subtree first. The size of a branch is the total number of packets in the branch initially. It is an optimal minimal-length scheduling protocol when transmissions are error-free.

In general, schedule- or TDMA-based protocols are better than contention- or CSMA-based ones for providing real-time (as demonstrated in, for example, [2] and [3]), since the latter are prone to repeated collisions that result in excessive and varying exponential backoff, especially under heavy traffic load. Thus we do not compare against CSMA-based protocols in this study.

Generating networks and traffic. We randomly place n nodes in a unit square. A random node is selected as the sink. We denote transmission range as ρ. That is, two nodes within the distance of ρ can communicate with each other. We construct a collection tree rooted at the sink using the shortest-path algorithm based on hop count. We assign each link in the tree a random quality in the range of $[\alpha, 1]$. Except the root, each node generates a number of packets randomly chosen from 0 to γ.

In the following experiments, we set the default value of n, ρ, α, and γ as 100, 0.2, 0, and 20, respectively, unless otherwise stated. Figure 4 shows the distribution of hop counts to the sink in such a network. When varying a parameter to study its effect on real-time data delivery, we run 30 experiments and show the average result.

B. *Impact of Deadline*

In this experiment, we vary D while keeping all other parameters constant. Figure 5 shows the DCRs of different protocols. We can see BLF outperforms others consistently, especially when D is small. For instance, the DCR of BLF is 52% when D is 1000, versus 16%, 2%, and 15% of LDF, CR-SLF, and LBF, respectively. Also, the DCRs of all protocols increase as deadline increases since more packets reach the root by deadline.

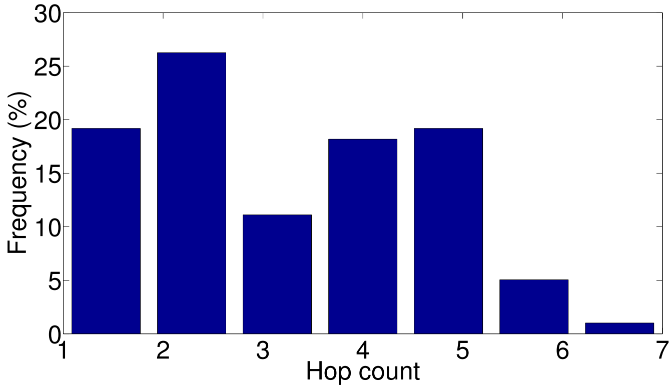

Fig. 4: Histogram of hop counts to the root in the random network

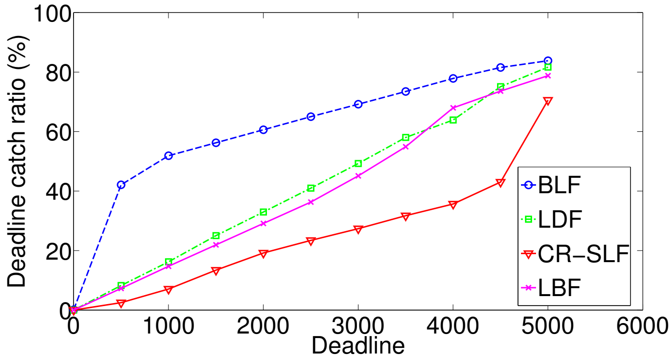

Fig. 5: Deadline catch ratios of different protocols for various deadlines

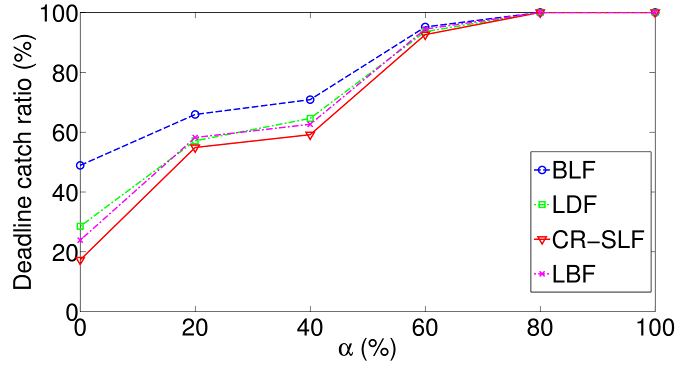

Fig. 6: Deadline catch ratios of different protocols for various link qualities

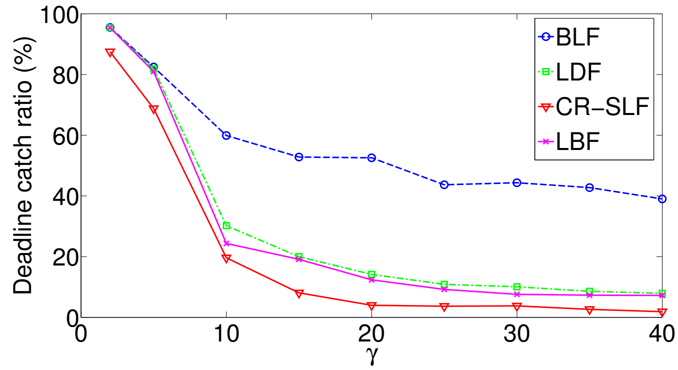

Fig. 7: Deadline catch ratios of different protocols for various traffic

C. Impact of Link Quality

In this experiment, we set D as 1000 and use the default value of all parameters except α. Figure 6 shows the DCRs of different protocols. When α is small, i.e., links are unreliable, BLF is significantly better than other protocols. This clearly shows the benefit of explicitly accounting for link loss in real-time scheduling. As α grows, links tend to become more reliable, and more packets are delivered by the same deadline. BLF is still the best among all protocols, even though its advantage gradually diminishes as links become reliable. In fact, when α is no less than 80%, BLF becomes indistinguishable from the rest.

D. Impact of Traffic Load

In this experiment, we set D as 1000 and keep all parameters intact except γ. A larger γ amounts to heavier traffic on average by the same deadline. Figure 7 shows the DCRs of different protocols. When traffic is light, i.e., γ is small, all protocols enjoy high DCRs. As traffic grows heavier, BLF still maintains high DCRs relative to other protocols, despite that all protocols' DCRs decrease. When γ is 40, BLF still obtains a DCR of 40%, while others only deliver 9% or less of packets in time.

VI. Related Work

Much work has studied scheduling real-time traffic in wireless networks, which we categorize as follows.

Work in the first category is based on the concept of debt. Each flow has a DCR requirement. The debt of a flow reflects how many timely deliveries the flow is lagging behind to satisfy its DCR requirement. Larger debt flow is favored upon smaller debt one in scheduling. [5] proposes a theory of real-time quality of service (QoS) for wireless networks, factoring in deadline, DCR requirement, and unreliable nature of wireless communication. It derives schedulability test for admission control that is both sufficient and necessary. Accordingly, largest debt first scheduling is devised and proved optimal. Though the flexible formulation allows for tractable analytical results, it only deals with single-hop topologies. [1] extends real-time QoS to multi-hop WSNs. It derives a sufficient condition for optimal scheduling. For full-duplex radios, Greedy Forwarder scheduling that transmits a packet belonging to be largest debt flow at each node is proved optimal. For half-duplex radios, only a heuristic Closest Sensor First scheduling is proposed, which is only optimal for a restrictive path-topology case. [6] considers scheduling real-time multicast traffic and shows that a policy that schedules the multicast flow whose users' sum of debt is maximal is optimal.

However, it works only for one hop scenario. All of the above work provides real-time QoS for individual flow/user, not for their aggregate as we do.

The second category leverages results from real-time systems theory. Generally, these scheduling algorithms quantify the urgency of a packet/flow based on its temporal parameters, borrowing notations from real-time systems. They schedule the most urgent (e.g., smallest latest-start time, least laxity, highest static priority) packet/flow first and then add other transmissions as long as interference and/or deadline constraints permit. [3] proposes the Channel Reuse-based Smallest latest-start-time (LST) First (CR-SLF) protocol to schedule real-time packets in multi-hop robotic sensor networks. The LST of a packet at a hop is defined as the latest time the packet must be transmitted at that hop to reach the destination in time. CR-SLF chooses the packet transmission with the smallest LST and adds other transmissions as many as possible if they do not interfere with any existing transmission nor cause any to miss deadline. [7] schedules periodic real-time flows with given routes and interference matrix using the Interference Aware Anticipatory (IAA) algorithm. It heuristically schedules flow with the smallest ratio of spatial overlap over laxity first. The spatial overlap of a flow is defined as the number of different links of other flows it interferes with, and the laxity of a flow is the tolerable waiting time during its ensuing downstream transmissions. [8] proposes Real-time Flow Scheduling (RFS), an enhanced extension of Real-Time Query Scheduling (RTQS) [9], for periodic real-time flows in WSNs under generic traffic patterns and interference models. Each flow is assigned a static priority. In the core of RFS, it schedules multiple flows concurrently if they are conflict free; when there is a conflict, it preferentially schedules higher priority flows over lower priority ones. [10] proposes an efficient heuristic algorithm, Conflict-aware Least Laxity First (C-LLF), to schedule real-time traffic in WirelessHART networks.

Work in the third category is based on Carrier Sensing Multiple Access/Collision Avoidance (CSMA/CA). The basic idea is to prioritize urgent packets by manipulating channel contention so that higher priority packets are more likely to win the channel. In [11], the authors propose multi-hop coordination, where a priority is assigned to each packet as it traverses across various broadcast regions. A packet's priority is increased at downstream nodes if it experiences excessive delay at upstream nodes and is likely to miss its end-to-end deadline, and vice versa. Within a broadcast region, higher priority packets randomly backoff in smaller contention windows in an extension of IEEE 802.11 and thus are more likely to capture the channel early on. [12] provides priority differentiation among soft real-time flows. It uses Black Burst (BB) for channel contention, which is a jamming signal that a node transmits after hearing the channel idle for a certain duration. The length of a BB is proportional to its packet priority. This ensures a packet with higher priority is favored over one with lower priority when contending the shared channel.

Another closely related line of study is to find a shortest interference-free TDMA schedule, given the number of packets at each source and conflict graph in a multi-hop collection tree. Examples include [2] and [13]. A good survey is [14]. They do not address real-time data collection directly and assume error-free transmissions.

None of the above aims to solve our problem and thus cannot be directly applied.

VII. Conclusion

In this paper, we study how to schedule packet transmissions to maximize the expected number of packets evacuated during a given time period in a lossy multi-hop wireless sensor network. We formulate it as a finite horizon Markov decision process and solve it optimally. Unfortunately, the solution becomes computationally intractable as the network and traffic grow. Thus, we propose the efficient Best Link First Scheduling protocol. BLF greedily schedules the best links first in a top-down fashion, complying with interference constraints. We prove its optimality for single-hop networks and develop an approach for distributed implementation. Through simulations in a plethora of network and traffic settings, BLF is shown to consistently outperform existing work, mainly by explicitly accounting for link losses in scheduling. Its gains are especially significant when deadline is small, links are highly unreliable, and traffic load is heavy, making it appealing to increasingly stringent real-time applications.

Considerable work is required to make BLF ultimately decentralized and field-deployable with provable properties. Interesting extensions abound and examples include, but are not limited to, the following: (1) consider more realistic interference models such as the Physical Ratio-K [15] and SINR models; (2) consider heterogeneous traffic where packets can have different arrival times and deadlines and also online traffic where packet arrivals are not known a priori; (3) consider link models other than Bernoulli such as the Gilbert-Elliott or general Markov chain model to account for burst errors; (4) develop efficient distributed implementation with low control overhead, quick convergence, and small memory footprint; (5) consider the problem when in-network aggregation of raw packets is allowed; (6) consider joint scheduling and routing to maximize expected deadline catch ratio.

References

[1] I-Hong Hou. Providing End-to-End Delay Guarantees for Multi-hop Wireless Sensor Networks over Unreliable Channels. April 2012.

[2] Wen-zhan Song, Fenghua Yuan, and Richard LaHusen. Time-Optimum Packet Scheduling for Many-to-One Routing in Wireless Sensor Networks. In *2006 IEEE International Conference on Mobile Ad Hoc and Sensor Sysetems*, pages 81–90. IEEE, October 2006.

[3] H Li, P Shenoy, and K Ramamritham. Scheduling messages with deadlines in multi-hop real-time sensor networks. *11th IEEE Real Time and Embedded Technology and Applications Symposium*, pages 415–425, 2005.

[4] Douglas S. J. De Couto, Daniel Aguayo, John Bicket, and Robert Morris. A high-throughput path metric for multi-hop wireless routing. In *Proceedings of the 9th annual international conference on Mobile computing and networking - MobiCom '03*, page 134, New York, New York, USA, September 2003. ACM Press.

[5] I hong Hou, Vivek Borkar, and P. R. Kumar. A theory of qos for wireless. In *In Proc. of IEEE INFOCOM*, 2009.

[6] Kyu Seob Kim, Chih-ping Li, and Eytan Modiano. Scheduling multicast traffic with deadlines in wireless networks. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 2193–2201. IEEE, April 2014.

[7] S. M. Shahriar Nirjon, John A. Stankovic, and Kamin Whitehouse. IAA: Interference aware anticipatory algorithm for scheduling and routing periodic real-time streams in wireless sensor networks. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*, pages 14–21. IEEE, June 2010.

[8] Octav Chipara, Chengjie Wu, Chenyang Lu, and William Griswold. Interference-aware real-time flow scheduling for wireless sensor networks. In *Proceedings of the 2011 23rd Euromicro Conference on Real-Time Systems*, ECRTS '11, pages 67–77, Washington, DC, USA, 2011. IEEE Computer Society.

[9] O. Chipara, Chenyang Lu, and G. Roman. Real-time query scheduling for wireless sensor networks. In *Real-Time Systems Symposium, 2007. RTSS 2007. 28th IEEE International*, pages 389–399, 2007.

[10] A. Saifullah, You Xu, Chenyang Lu, and Yixin Chen. Real-time scheduling for wirelesshart networks. In *Real-Time Systems Symposium (RTSS), 2010 IEEE 31st*, pages 150–159, 2010.

[11] V Kanodia, C Li, A Sabharwal, B Sadeghi, and E Knightly. Distributed multi-hop scheduling and medium access with delay and throughput constraints. *Proceedings of the 7th annual international conference on Mobile computing and networking MobiCom 01*, pages 200–209, 2001.

[12] Bach D. Bui, Rodolfo Pellizzoni, Marco Caccamo, Chin F. Cheah, and Andrew Tzakis. Soft Real-Time Chains for Multi-Hop Wireless Ad-Hoc Networks. In *13th IEEE Real Time and Embedded Technology and Applications Symposium (RTAS'07)*, pages 69–80. IEEE, April 2007.

[13] C. Florens, M. Franceschetti, and R.J. McEliece. Lower Bounds on Data Collection Time in Sensory Networks. *IEEE Journal on Selected Areas in Communications*, 22(6):1110–1120, August 2004.

[14] Ozlem Durmaz Incel and Amitabha Ghosh and Bhaskar Krishnamachari. *Scheduling Algorithms for Tree-Based Data Collection in Wireless Sensor Networks*.

[15] Hongwei Zhang, Xin Che, Xiaohui Liu, and Xi Ju. Adaptive instantiation of the protocol interference model in wireless networked sensing and control. *ACM Trans. Sen. Netw.*, 10(2):28:1–28:48, January 2014.