Scheduling Messages with Deadlines in Multi-hop Real-time Sensor Networks [*]

Huan Li, Prashant Shenoy,
Department of Computer Science,
University of Massachusetts,
Amherst, MA 01003
{lihuan,shenoy}@cs.umass.edu

Krithi Ramamritham
Dept. of Computer Science and Engineering,
Indian Institute of Technology,
Mumbai 400076, India
krithi@cse.iitb.ac.in

Abstract

Consider a team of robots equipped with sensors that collaborate with one another to achieve a common goal. Sensors on robots produce periodic updates that must be transmitted to other robots and processed in real-time to enable such collaboration. Since the robots communicate with one another over an ad-hoc wireless network, we consider the problem of providing timeliness guarantees for multi-hop message transmissions in such a network. We derive the effective deadline and the latest start time for per-hop message transmissions from the validity intervals of the sensor data and the constraints imposed by the consuming task at the destination. Our technique schedules messages by carefully exploiting spatial channel reuse for each per-hop transmission to avoid MAC layer collisions, so that deadline misses are minimized. Extensive simulations show the effectiveness of our channel reuse-based SLF (smallest latest-start-time first) technique when compared to a simple per-hop SLF technique, especially at moderate to high channel utilization or when the probability of collisions is high.

1 Introduction

Wireless sensor networks have received increased research attention in recent years. Emerging sensor applications include habitat monitoring, pollution detection, weather forecasting, and monitoring disasters such as earthquakes, fires and floods. In many applications, sensor data must be delivered with time constraints so that appropriate actions taken in real-time [16].

Consider the scenario where a group of robots are searching for people trapped inside a building on fire. Each robot is equipped with numerous devices such as camera, temperature, pressure, location, and infrared sensors. The robots communicate with one another over an ad-hoc wireless network and collaborate as a team to achieve their common goal. For instance, the robots need to pool sensory information to determine, in real-time, where to move individually and collectively. Further, if an object of interest is discovered by one robot, the discovery must be transmitted to the outside world in real-time, together with other information such as the location and the temperature map of the area. However, existing wireless protocols such as 802.11b do not provide timeliness guarantees on network transmissions due to packet collisions, exponential back-offs, and the false blocking problem [14].

In this paper, we focus on the problem of providing timeliness guarantees for multi-hop transmissions in a real-time robotic sensor application. In such applications, each message is associated with a deadline and may need to traverse multiple hops from the source to the destination. Message deadlines are derived from the validity of the accompanying sensor data and the start time of the consuming task at the destination. We show that the problem of meeting message deadlines is NP-hard even for single hop message transmissions. Consequently, we propose heuristics for online scheduling of messages with deadline constraints. Our technique (i) schedules messages based on their per-hop timeliness constraints, (ii) carefully exploits spatial reuse of the wireless channel and (iii) explicitly avoids collisions to reduce deadline misses. We evaluate our technique using simulations of various robot topologies and examine the impact of various system and sensor parameters on meeting timeliness constraints. Our results show that: the channel-reuse-based algorithm outperforms the CSMA/CA-based algorithm for a wide variety of experimental settings, and especially when (i) the channel utilization (the fraction of bandwidth the wireless channel is busy over a time interval) is high, (ii) the interference range is large, or (iii) the probability of collisions is high.

Spatial channel reuse in ad-hoc networks has been studied from the perspective of minimizing the total transmis-

---

[*] This research was supported in part by NSF grants CCR-9984030, CCR-0219520, EIA-0080119, CNS-0323597 and a gift from Microsoft.

sion time [6], or improving channel utilization, throughput and fairness [9, 10]. However, using spatial reuse to address the problem of meeting real-time constraints for multi-hop messages has not been studied, especially for sensor communication. A detailed comparison with related research efforts is presented in Section 6.

The remainder of this paper is structured as follows. Section 2 presents the application background, our system model and the problem formulation. Section 3 discusses the design issues for scheduling messages with deadlines. Section 4 describes our scheduling heuristics in detail. Experimental results are presented in Section 5, and finally, we address related work and summarize our work in Sections 6 and 7.

2 Background, System Model and Problem Formulation

2.1 Application Background

Consider a team of robots that collaborate to achieve a common task, such as searching for trapped people in a building on fire, exploring the territory of Mars, or building a map of an unknown environment. The robots are equipped with numerous sensors such as camera, temperature, pressure, infrared sensors, and positioning devices. Due to energy and weight considerations, each robot may only carry a subset of the sensors and needs to communicate with other robots for other sensor readings. Each robot is also equipped with a wireless connectivity, and the robots communicate with one another over an ad-hoc network. Due to the finite transmission range of the wireless, if a destination robot is not within the effective range of the sender, a sensor message will need to traverse through intermediate robots to reach its destination.

To achieve their common goal, each robot in the team may be assigned one or more tasks. For instance, a robot that is assigned the task of generating an up-to-date temperature map of the fire will need to periodically receive temperature readings from other robots along with their locations. Robotic teams are typically organized as *leader-follower* groups [18], where the leader robot is designed with a task of determining a "plan" for the team at each step. A plan consists of two components: a communication plan and a path plan. Communication plan tells the robots who is talking to whom; path plan tells robots where to move next.

To compute a plan, the leader takes into account (i) the current positions of follower robots, (ii) the overall goal and (iii) all "important" sensor readings that have been reported (e.g., smoke in a certain area). Each time when a new plan is constructed, it is conveyed to the follower robots which then exchange messages according to the communication plan and move according to the path plan (see Figure 1). In this model, actions required for (1) communicating the plan, (2) transmitting sensor updates, and (3) sending required sensor values, if any, back to the leader to create a new plan are the three components of the constructed plan. Thus, plans need to be computed in real-time to enable the robots to move continuously and communicate in real-time.

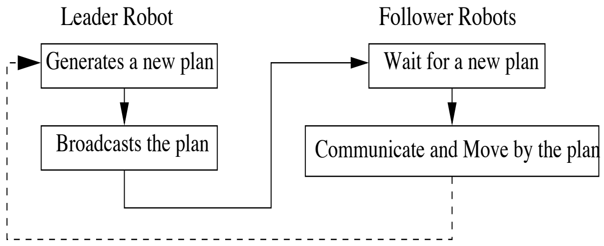

Figure 1. Plan generation and execution

2.2 Data Validity and Transmission Deadlines

In real-time sensor applications, the values of sensor data reflect the current state of the environment. Since the environment may be constantly changing, sensor readings have a temporal interval for which they are valid. For instance, the temperature readings from a few minutes back are no longer valid if the fire in a room burns out of control. We use the term *data validity* to define the time interval for which a data value produced by a sensor is valid.

In this work, we assume that sensors produce data periodically, determined by the *sensor period*. Each data value is propagated from the source robot to one or more destination robots for processing. Each value has a data validity interval associated with it, and the value must reach the destination before the validity interval expires. Further, the sensor message must arrive at the destination before the start time of the consuming task that consumes it. Thus, the effective deadline of a sensor message is the minimum of the data validity deadline and the start time of the consuming task. Formally, if a sensor value is produced at time t, and its validity is v time units, then the effective deadline of sensor message m is:

$$ed(m) = \min(t + v, \delta) \tag{1}$$

where δ is the start time of the consuming task that consumes m. Suppose that the message needs to traverse h hops from the source to the destination and let m^i denote the transmission of message m at the i^{th} hop. Further, let the path delay $pd(m^i)$ denote the total propagation delay and transmission time that will be incurred on the remaining hops to the destination. Then the latest transmission start time (LST) of the message at the i^{th} hop is:

$$l(m^i) = ed(m) - pd(m^i) \tag{2}$$

The LST denotes the *latest time* by which the i^{th} hop must start transmitting the message in order for it to reach the destination by its effective deadline. If there are multiple messages queued up at a robot awaiting transmissions, then their LSTs can be used to derive a transmission schedule.

2.3 Problem Formulation

Given a team of robots equipped with sensors, suppose that each sensor produces a new value periodically, once every p_i time units. Suppose that this sensor value needs to be sent to from the source robot R_s to a destination robot R_d for processing before its effective deadline *ed.*

Since the robots form an ad-hoc network, the sensor message will need to traverse multiple hops to reach its destination. We exploit a specific characteristic of our robotic application to route messages over the ad-hoc network. Specifically, the leader robot is aware of the precise location of each robot for purposes of path planning. Since the location of each robot is known, the leader has a complete view of the ad-hoc network and can compute an optimal route from each sensor to the destination robot during planning. Computed routes are conveyed to the robots for forwarding messages. Since planner has full knowledge of the topology and the movement of robots, it can recompute routes before the underlying topology changes. By exploiting this specific characteristic of the robotic application, the need for a dynamic route discovery protocol can be avoided [13].

In this paper, we will discuss the difficulties and solutions for the last two phases of a communication plan, and leave out the broadcast phase since it has been thoroughly studied in networking community. Given a set of sensor messages that are generated periodically, their associated deadlines and their routes, we study the problem of scheduling message transmissions at each hop so that end-to-end deadline violations are minimized. A desirable scheduler should not only be cognizant of message deadlines when making scheduling decision, it should also effectively use the channel bandwidth by (1) avoiding collisions to minimize the useless bandwidth utilization, and (2) exploiting spatial reuse to maximize bandwidth utilization. Since packet collisions cause a node to back-off and retransmit, they increase the end-to-end delay for a message transmission. Avoiding collisions—two simultaneous transmissions will not collide if and only if the two receivers are mutually outside the other sender's interference range [1]—is crucial for providing timeliness guarantees. Similarly, exploiting spatial reuse enables multiple simultaneous transmissions, enabling the application to maximize its use of the channel capacity.

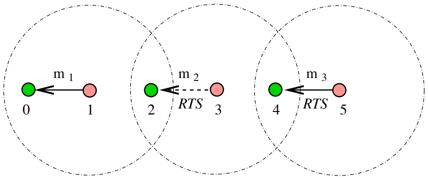

Figure 2. Example of false blocking and channel reuse

3. Design Considerations

Wireless networks based on the 802.11 family use CSMA/CA algorithms at the MAC layer. In this section, we argue that CSMA/CA networks have certain drawbacks that can hamper timeliness guarantees that are crucial in real-time sensor applications.

First, despite the use of collision avoidance techniques, CSMA/CA networks do not completely eliminate the possibility of a collision. Second, senders can back-off exponentially when they sense ongoing transmissions on the channel. Third, CSMA/CA networks use a clear-to-send (CTS)/request-to-send (RTS) protocol for collision avoidance. In such networks, a node must explicitly request permission to transmit by sending a "request-to-send" (RTS) and must receive a "clear-to-send" (CTS) acknowledgment before sending data. Further, all nodes in the vicinity that receive these messages must avoid transmission for the transmission duration, and are thus *blocked*. This can lead to the false blocking problem [14] which reduces spatial channel reuse, as illustrated next.

3.1 Message Contention and Channel Reuse

Consider a simple one hop transmission example depicted in Figure 2. The location of each node and transmission range of each sender are shown in the figure. For simplicity, we assume the interference range is equal to the transmission range. As shown in Figure 2, three single hop messages m_1, m_2 and m_3 need to be scheduled for transmission. Table 1 depicts the arrival time (AT), the transmission duration (TT)(the difference between the instants when the first bit is sent out and the last bit is received by the receiver), and the effective deadline (ED), for each message. The latest start time (LST), calculated using Equation 2, is also shown in the table.

Since m_1 arrives at time 0 and is the only message in the system, node 1 transmits a RTS and receives a CTS acknowledgment from node 0. Since node 2 also receives this RTS, it is blocked during the transmission of m_1. When message m_2 arrives at time 1, node 3 sends a RTS to node 2 but does not receive a response. Node 4 receives this RTS message as well and is also blocked. Consequently, when

---

[1] Transmission range of a sender is defined to be the radial range where all nodes inside it can successfully receive a transmission from the sender; interference range is the distance where the sender's transmissions interfere with a node that is receiving a message sent by a third node.

Message	AT	TT	ED	LST
m_1	0	2	6	4
m_2	1	5	8	3
m_3	1	2	8	6

Table 1. Message characteristics

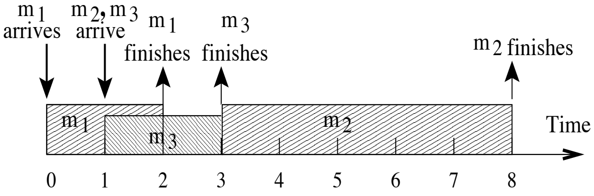

Figure 3. Parallel transmissions reduce deadline misses.

node 5 sends a RTS to node 4 for message m_3, it does not receive a CTS and is unable to transmit m_3. Observe, from Figure 2, that m_1 and m_3 do not interfere with one another and it is possible to transmit them simultaneously. This is referred to as *false blocking*. In this example, false blocking causes the messages to be transmitted sequentially in the order m_1, m_2 and m_3, resulting in poor spatial reuse. More importantly, doing so causes m_3 to miss its deadline.

However, if the scheduler were to exploit spatial reuse and transmit m_3 in parallel with m_1, followed by m_2, then all messages are able to meet their deadlines (see Figure 3).

3.2 Why Simple Channel Reuse is Not Sufficient?

Although the previous example demonstrated the benefits of exploiting spatial channel reuse for meeting deadline constraints, surprisingly, parallelizing transmissions via spatial reuse can sometimes *increase* deadline misses. Consider the same scenario depicted in Figure 2 but with the parameters listed in Table 2. Like before, the sensor message m_1 is transmitted first at time $t = 0$. Since m_1 and m_2 interfere with one another, m_2 is blocked. However, when m_3 arrives at time $t = 1$, it can be transmitted in parallel with m_1 since the two transmissions do not interfere. Assuming this is done, m_2 can not be transmitted until m_3 finishes at time $t = 3$. Since message m_2 requires a transmission duration of 6 time units, m_2 will finish only at $t = 9$, causing it to miss its deadline (see Figure 4). In this scenario, the only schedule that satisfies all deadlines is to transmit the messages sequentially: m_1, m_2 and m_3. The example shows that naively maximizing spatial reuse can sometime be detrimental in meeting deadline guarantees. Thus, the message scheduler should consider the potential impact of scheduling a message on future message transmissions.

Message	AT	TT	ED	LST
m_1	0	2	6	4
m_2	1	6	8	2
m_3	1	2	10	8

Table 2. Message characteristics

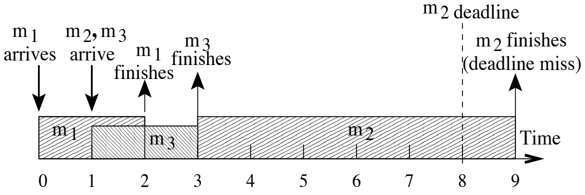

Figure 4. Deadline misses caused by parallel transmissions.

4 Scheduling Messages With Deadlines

The problem of scheduling parallel messages with deadlines over wireless channel can be shown to be NP-hard. The problem is NP-hard even when messages traverse a single hop, having unit transmission durations and identical effective deadlines. This can be proved by reducing the graph k-colorability problem [2] to it, where a contention graph $G = (V, E)$ is used to represent conflicts among transmissions. In the graph, each vertex represents a transmission, and an edge exists between two vertices iff the two transmissions (vertices) cannot be scheduled simultaneously. If the graph is k-colorable, all transmissions can be completed within k time units.

Due to the NP-hard nature, we must resort to heuristics to schedule multi-hop messages through the network. In this section, we present two such heuristics.

4.1 Per-Hop Smallest LST First (PH-SLF)

Per-hop Smallest LST First (PH-SLF) is a distributed scheduler, where each node makes local scheduling decisions independent of other nodes. In this approach, given a set of messages that are queued up at a node, the node schedules the message with the smallest LST for transmission. Observe that the latest start time (LST), as defined in Equation 2, is the deadline by which the node must start transmitting the message in order for it to meet its effective (end-to-end) deadline. The underlying MAC protocol is vanilla CSMA/CA. As a result, collisions and the resulting back-offs, and false blocking can not be eliminated in this approach. The advantage of this approach is that it can be used in conjunction with vanilla 802.11 networks, since PH-SLF can be implemented in software in the OS driver.

We use PH-SLF as our baseline algorithm. In the rest of this section, we present an approach that explicitly avoids collisions and maximizes spatial reuse.

4.2 Channel Reuse-based Smallest LST First (CR-SLF)

The goal of our Channel Reuse-based Smallest LST First (CR-SLF) approach is to be cognizant of message deadlines at each hop, while avoiding collisions and exploiting spatial reuse. Before presenting the approach, we define some terminology. Table 3 lists the key attributes for a message and the associated terminology.

Notation	Meaning
m_x^i	message m_x at the i^{th} hop
$T(m_x^i)$	the transmission of m_x^i
$a(m_x^i)$	the arrival time of m_x at the i^{th} hop
$d(m_x^i)$	the deadline of m_x^i, $d(m_x^i) = ed(m_x)$
$l(m_x^i)$	the latest start time *(LST)* of $T(m_x^i)$, see Eq. 2
$s(m_x^i)$	the transmission start time of $T(m_x^i)$
$f(m_x^i)$	finish time of m_x^i—the time m_x reaches the next hop
$e(m_x)$	the execution time for m_x (same for all per-hop transmissions of m_x)

Table 3. Message attributes

Since sensor updates are generated periodically, the arrival time of a message at the source (first hop) is the time at which the sensor data is produced. The arrival time at an intermediate node is the time the last bit of the message arrives at that hop enroute to the destination. Observe that the arrival time at an intermediate node depends on when the message is scheduled for transmission at the previous hop. The start time is the time when message is scheduled for transmission and the finish time is the time when the message is completely received by the next hop node (and the channel becomes idle again). The execution time is the time for which the channel is busy and is the sum of the transmission delay and the propagation delay (the difference between the instants when the first bit is sent out and the last bit is received by the next hop). Observe that,

$$a(m_x^{i+1}) = f(m_x^i) = s(m_x^i) + e(m_x^i)$$

In the rest of this paper, transmission refers to a per-hop transmission unless specified otherwise. With this background, we present the intuition behind our approach, followed by the details.

4.2.1 Overview

Given a set of nodes, their locations and transmission ranges as well as a set of messages queued up at these nodes, their destinations, effective deadlines, and the associated routes, our scheduler derives a schedule to meet these deadlines. By a schedule, we mean that the transmission start time of each message is computed for all hops from the source to the destination.

The scheduler exploits the following characteristics in deriving this schedule:

- It maximizes spatial reuse by scheduling non-interfering message transmissions in parallel.
- It considers message transmissions in the order of their LSTs. The LST is the "local" deadline of a transmission at a hop, since it is the latest time by which the message must be scheduled to meet its end-to-end effective deadline.

The basic idea is to partition the set of message transmissions into *disjoint sets* such that transmissions within each set do not interfere with one another and can be executed in parallel. These sets are ordered sequentially, and all transmissions within a set must finish before transmissions in the next set can begin.

To construct these sets, the scheduler considers the transmissions in order of their LSTs. At each step, the transmission with the smallest LST is chosen and the scheduler checks if it is feasible to assign this transmission to an existing set. It is feasible to do so only if (i) the transmission does not interfere with existing message transmissions in that set, (ii) the message can be scheduled for transmission so that its finish time is no later than its deadline, and (iii) inserting this transmission into the set does not cause deadline violations for currently scheduled transmissions in other sets. If no existing set is feasible, a new set is created with that message transmission so long as the deadline is met. Once a message is scheduled at hop i, it can be considered for scheduling at hop $i+1$. Observe that a message needs to be transmitted hop by hop, since the arrival time at the next hop is not known until it is scheduled for transmission at the previous hop. The above process continues until all queued up messages are scheduled along all hops from their sources to their destinations (i.e., the message transmission on each hop is assigned to a feasible set). The constructed sets define the transmission schedule for these multi-hop messages.

The scheduler that executes such an algorithm is designed to be a centralized scheduler. A centralized model is reasonable for a team of robots since a centralized path planner is used to determine the movement plan for each robot. Because the planner has complete knowledge of the location of all nodes and the messages they need to send, the communication scheduler can run in conjunction with the planner as depicted in Figure 1.

4.2.2 Details of the Algorithm

Initially, the schedule $\mathcal{S}$ is empty: $\mathcal{S} = \phi$. The goal is to construct a set of sets $\mathcal{S} = \{S_1, S_2, ..., S_n\}$ where elements are disjoint and message transmissions in each set are non-interfering. We define a *start time* and a *finish time* for each set S_j. The *start time* of set S_j, $s(S_j)$, denotes the instant where all its transmissions can start using the channel; the *finish time* of S_j, $f(S_j)$, denotes the instant where all transmissions have reached the their respective receivers. In general, $s(S_{j+1}) = f(S_j)$; that is, transmissions in a set can start to transmit when those in the prior set have finished.

The algorithm proceeds in the following steps.

Step 1 : *Select a transmission to schedule.* From the list of yet to be scheduled message transmissions, the scheduler chooses the one with the *smallest* LST. This enables the scheduler to consider the most urgent transmission first. Other strategies are possible, such as choosing transmissions based on the earliest data validity first or earliest start time first, although we do not consider such alternatives in this paper.

Step 2: *Assign this message transmission to a set.* Suppose that n sets have been constructed in the partial schedule thus far: $S_1, S_2, ..., S_n$, and $T(m_x^i)$ has been selected. The scheduler attempts to assign this transmission to the *first feasible set* in the set list. If no existing set is feasible (or the schedule is empty), then a new set S_{n+1} is added to the list and $T(m_x^i)$ is inserted into this set so long as the deadline constraint is not violated.

A set S_j to be feasible for a message transmission $T(m_x^i)$ iff the following conditions are satisfied.

1. The finish time of the set $f(S_j)$ is later than the arrival time of the message $a(m_x^i)$. This indicates that a message transmission should be added to a set only if there is some temporal overlap with existing transmissions so that parallelism can be exploited.
2. The finish time of m_x^i is no later than its effective deadline, i.e. $f(m_x^i) = max(s(S_j), a(m_x^i)) + e(m_x^i) \le d(m_x^i)$.
3. $T(m_x^i)$ does not interfere with any existing message transmissions in S_j.
4. The insertion of the transmission $T(m_x^i)$ into S_j does not violate deadlines of messages in subsequent sets $S_k, j < k \le n$.

The first three conditions are easy to understand, so we elaborate on the fourth condition. If the current message m_x^i happens to be the longest message or its transmission finishes last in S_j, then the duration for which S_j occupies the wireless channel is increased. As a result, the transmission start times of messages in subsequent sets will need to be pushed forward. Since a later start time may violate their deadlines, the scheduler needs to verify that inserting this transmission of m_x^i in S_j does not impact the deadline meeting of subsequent message transmissions. To do so, we first need to compute the new finish time of S_j. If inserted in S_j, the start time of m_x^i is $s(m_x^i) = \max(s(S_j), a(m_x^i))$. Then, its finish time is $f(m_x^i) = s(m_x^i) + e(m_x^i)$. The new finish time of the set is the maximum finish time of all messages in the set: $f_{new}(S_j) = max_{\forall x}(f(m_x))$.

Next, we can compute the amount by which all subsequent transmissions are pushed forward. This is done by computing the new start time of each set S_k, $j < k \le n$, which is simply the finish time of the previous set: $s(S_k)) = f(S_{k-1})$. The new start time of each message transmission $T(m_y)$ in the set is recomputed as $s(m_y) = \max(s(S_k), a(m_y))$. Then, the finish time is $f(m_y) = s(m_y) + e(m_y)$. The new finish time of the set is the maximum finish time of all messages in the set: $f_{new}(S_k) = max_{\forall y}(f(m_y))$, $T(m_y) \in S_k$. Given the new start and finish times of the affected messages, the scheduler needs to verify that the finish time of each message is not later than its deadline: $f(m_y) \le d(m_y)$. If no deadlines are violated, then it is possible to insert the current selected message transmission, $T(m_x^i)$, in S_j.

The scheduler searches for the first set in the list $S_1, S_2, ..S_n$ that is feasible and inserts the selected transmission into that set. If no existing set is feasible and inserting the transmission into a new set S_{n+1} violates its deadline, then the algorithm can not meet the deadline for this message. In this scenario, the message is removed from consideration and all scheduled transmissions for hops 1 to $i - 1$ are removed from the corresponding sets. And then the start/finish times of those sets are adjusted accordingly.

Step 3 : *Update the finish time of the feasible set and insert a new transmission for the next hop.* If a feasible set S_j is found, then the transmission $T(m_x^i)$ is inserted into the set, and the new finish time is updated as discussed above. The transmission $T(m_x^i)$ is deleted from the list of yet to be scheduled transmissions, and the next hop transmission $T(m_x^{i+1})$ is inserted into the list, assuming that hop i is not already the destination for the message. All these steps are repeated until the list of unscheduled transmissions becomes empty.

4.2.3 An example

In this section, we will use a simple one-hop transmission example to illustrate the basic procedure of the algorithm. Consider the example shown in Figure 2 with message characteristics in Table 1. Because $T(m_1)$ has the smallest LST, it is selected first. Initially, the schedule set is empty, a new set S_1 is created for the transmission of m_1, and we have: $s(S_1) = 0$, $f(S_1) = f(m_1) = 2$. Then, $T(m_2)$ is considered since it has the smallest LST. Since $T(m_2)$ interferes with $T(m_1)$, set S_1 is not feasible and a new set is needed for $T(m_2)$. Thus, we have two sets: $S_1 = \{T(m_1)\}$, $S_2 = \{T(m_2)\}$, and $s(S_2) = f(S_1) = 2, f(S_2) = max(s(S_2), a(m_2)) + e(m_2) = 2 + 5 = 7$. Finally, let us consider the last transmission of m_3. Consider S_1 first with the four conditions: 1) The finish time of S_1 is 2 which is later than $a(m_3)$,

2) $f(m_3) = max(s(S_1), a(m_3)) + e(m_3) = 1 + 2 = 3 < d(m_3)$, 3) Transmission of m_3 will not interfere with the existing transmission of m_1 in the set, 4) Since $f(m_3) = 3 > f(S_1)$, we have the new possible finish time for S_1: $f_{new}(S_1) = f(m_3) = 3$; so the new start time for S_2 is $s_{new}(S_2) = f_{new}(S_1) = 3$ and we have $f_{new}(m_2) = 3+5 = 8 \leq d(m_2) = 8$. Therefore, S_1 is feasible for $T(m_3)$, transmission of m_3 is assigned to S_1. The final schedule is : $S_1 = \{T(m_1), T(m_3)\}$, $S_2 = \{T(m_2)\}$, which indicates that m_1 and m_3 are transmitted in parallel, followed by the transmission of m_2.

5 Performance Evaluation

We have designed an event-driven simulator to simulate a team of robots that exchange multi-hop sensor messages and move in the environment. We compare the performance of our proposed algorithm CR-SLF with the PH-SLF scheduling using our simulator. We evaluate the impact of sensor period, deadline and message size, based on specific transmission topology. Further, we investigate how the different interference ranges may affect the properties of the algorithms.

The metrics used to measure performance is the *Deadline Miss Ratio* which is defined to be:

$$\frac{number\ of\ unsuccessful\ end\text{--}to\text{--}end\ transmissions}{total\ end\text{--}to\text{--}end\ transmissions}$$

Note here, a successful message transmission is one where the message is transmitted from the source to the destination before the effective deadline.

5.1 Experimental Settings

The wireless card on each robot and its radio parameter are based on the existing commercial product (e.g., Lucent Wavelan card) with a 2 Mbps data rate and a transmission range of 250 meters. Unless specified otherwise, the interference range is set to be equal to the transmission range. In our simulations, each pair of nodes (robots) are separated by the distance of 200 meters which is likely to yield close to the maximum capacity possible [7]. All simulated data packets are preceded by an RTS/CTS exchange, and all packet sizes include a header with the payload.

Unless specified otherwise, Table 4 gives the default parameter settings used in our simulations. The effective deadlines are specified in relative terms, relative to the start times of the respective transmissions at the source.

Message Size	Sensor Period	Relative Deadline
512 byte	10ms	50ms

Table 4. Default Settings

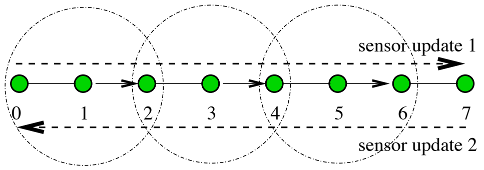

Figure 5. Scenario 1: transmissions over a robot chain

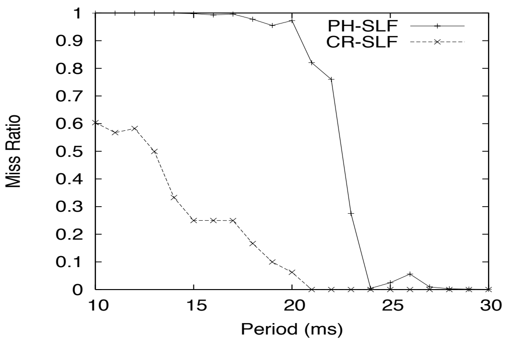

Figure 6. Impact of sensor period

5.2 Impact of the Sensor Period and Deadline

In this experiment, the robotic group is based on a chain topology with 8 robots, as illustrated in Figure 5. Two sensor messages periodically travel through intermediate nodes from two end robots to the opposite end on both directions. We use this scenario to demonstrate the effectiveness of our proposed algorithm, CR-SLF, for different sensor periods and deadlines, and compare it to PH-SLF scheduling.

The impact of varying the sensor periods is shown in Figure 6. In this simulation, the period to generate a new message is depicted on the x-axis. We can see that CR-SLF has fewer deadline misses than the PH-SLF scheduling, especially when the period is between 10ms and 20ms and where collisions are likely unless the scheduler is careful. Beyond 20ms, since the transmission duration per hop is approximately around 0.002s. When the next update is generated, the previous one has reached the destination (without considering the update from the opposite direction). Hence, the probability of collisions is very low. Therefore, both algorithms are able to schedule messages with few collisions or deadline misses.

The impact of message deadlines is shown in Figure 7. Here, the sensor period is set to 19ms. As the deadline of each sensor update goes from 50ms to 150ms, we have 40% improvement for CR-SLF and 87% improvement for PH-SLF. The reasons we have less improvement for CR-SLF are: 1) although the deadlines are different, the utilization and the probability of collisions are the same; CR-SLF has already explicitly taken these issues into account, and the deadline will only affect the order of transmissions considered, 2) for the PH-SLF scheduling, the larger deadline ac-

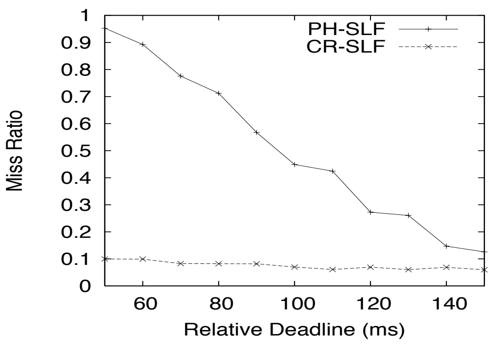

Figure 7. Impact of message deadline

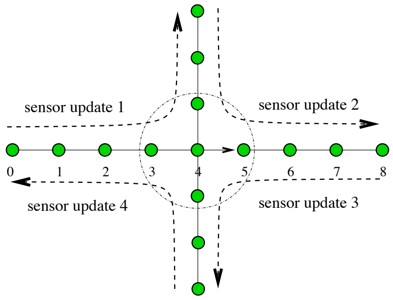

Figure 8. Scenario 2: transmissions over a cross topology

tually gives it more chances to back off and retransmit, increasing the number of eventual successful transmissions.

5.3 Impact of Message Size

In this example, we study the effect of varying the message size. We use a cross topology as shown in Figure 8. There are four periodic sensor updates traveling through the network. The period of each sensor is 30ms and the relative deadline is 100ms.

We vary the message size from 256 bytes to 1024 bytes for each respective transmission and measure the miss ratio. The results are shown in Table 5. As we know, the packet size reflects the transmission duration. Since small packet sizes have a smaller probability of collisions, both algorithms work very well. However, as the packet size increases, the per hop transmission duration increases. This increases the duration for which the channel is busy, since many transmissions may be traversing intermediate hops at the same time. Consequently, collisions cause exponential back-offs and increase the queue delay at each hop. As a result, the performance of PH-SLF scheduling degrades a lot. On the other hand, since CR-SLF explicitly avoids collisions, it is able to meet the deadlines for larger message sizes, so long as the messages are feasible.

Size (byte)	PH-SLF MissRatio	CR-SLF MissRatio
256	0	0
512	0.725	0.124
1024	0.996	0.394

Table 5. Impact of message size

5.4 Impact of Interference Range

Note that one node can interfere with message reception at another node even when they are too far apart for successful transmission [7]. In this simulation, we use a tree topology, shown in Figure 9, to study the impact of the interference range. Observe that in a tree, the interference increases for nodes closer with the root, due to the larger node density. We use the parameters in [7] to vary the interference range. We consider two different interference ranges, namely 250 meters and 550 meters, and two different sensor periods, namely 20ms and 25ms. Sensor messages from two leaf nodes, node 19 and 22, are assumed to be sent out periodically to the root (node 0). The third sensor update is sent from a leaf, node 20, to another leaf on the other side, node 21. The relative deadline for each message update is set to 200ms.

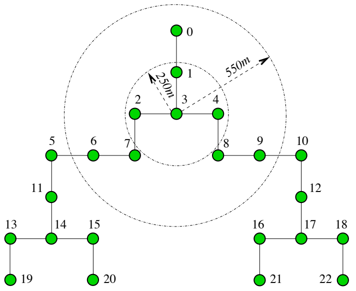

Figure 9. Scenario 3: tree topology with various interference ranges

We measure the miss ratio for all four combinations of the interference ranges and the sensor periods. Figure 10 shows the results and we have several observations. First, CR-SLF performs better than the PH-SLF scheduling for all different parameters. Second, for a fixed sensor period, when the interference range increases, the miss ratio increases for both algorithms, but the difference between the two algorithms gets smaller. The reason is that if the interference range is small, the probability of collisions is dominated by the sensor period. However, when the period is larger, the occurrence of collisions is dominated by both of the sensor period and the interference range.

5.5 Impact of Routing and Node Mobility

In this section, we consider the impact of routing and node mobility. Figure 11 depicts the sequence of the moves in a robot chain, where the leader (node 7) leads the team to make a U-turn. Two sensor messages traverse the chain, one from node 0 to 7 and vice versa, with sensor period of

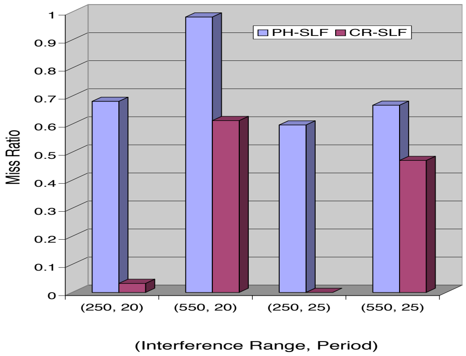

Figure 10. Impact of interference range

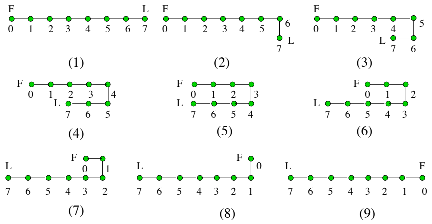

Figure 11. Scenario 4: movement of a robot team

15ms. Two scenarios are considered here: (i) overlay routing, where each update traverses through all intermediate nodes of the overlay chain, and (ii) shortest path routing, where transmissions use the shortest path from the source to destination.

The results of CR-SLF are shown in Figure 12. For the overlay routing, since the update has to travel through every intermediate node, as the team moves from position (1) to (5), the interference increases due to the increase of the node density. Hence, the miss ratio increases. But when the team moves from (5) to (9), the team becomes a chain again, the miss ratio decreases since the interference decreases.

For the shortest path routing, the algorithm will always find the shortest path for message transmissions. For instance, in position (5), messages are directly sent out from node 0 to 7 without passing through any other nodes. So when the team moves from position (1) to (5), the number of intermediate routers involved actually decreases. This causes the miss ratio to decrease since the probability of collisions also decreases.

5.6 Summary of Our Results

In summary, CR-SLF performs much better than the PH-SLF algorithm both at high and low utilization in term of the measurement for miss ratio, especially when the probability of collisions is high. The major reason is that CR-SLF explicitly avoids collisions while PH-SLF incurs exponential back-off at each collision. Also, CR-SLF doesn't inject infeasible packets into the network once it finds out the packet cannot meet the deadline; hence, the infeasible packet will not affect other feasible packets.

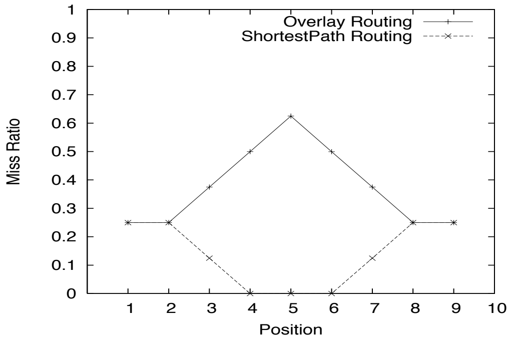

Figure 12. Impact of mobility

6 Related Work

Real-time research challenges for wireless sensor work are addressed in [17, 16]. The delay performance of policed traffic to provide statistical real-time guarantees over wireless networks was studied in [19]. Velocity monotonic scheduling for scheduling packets in real-time multi-hop sensor network was proposed in [8]. Although the selection of packet takes the real-time constraint into consideration, no conflicts or interference effects are taken into account, and thus no guarantees are given. SPEED [4] uses feedback control technique to support soft real-time communication service with a desired delivery speed across the sensor network, so that the end-to-end delay is proportional to the distance between the source and destination. SPEED focuses on routing service whereas our focus is on scheduling. Additional differences between their technique and ours are: 1) maximizing useful bandwidth utilization is explicitly taken into account for scheduling message transmissions to meet the deadlines in our method, and 2) we only inject those messages that can meet deadlines into the network. [6] studies the problem of minimizing the total packet transmission time in wireless sensor network. However, in that problem, only single hop is considered and there is no deadline constraint.

Recently, there has been work on the relationship between the delay and capacity or throughput in wireless networks [3, 15, 1, 11, 12]. All of these efforts focus on computing asymptotic performance bounds. By modeling neighboring interference as conflict graph, lower and upper bounds on the maximum throughput for a given network and workload are computed in [5].

7 Concluding Remarks

In this paper, we presented two heuristics for scheduling message transmissions with validity and processing constraints in multi-hop robotic sensor networks. Our results show that CR-SLF outperforms PH-SLF since it not only takes the deadline into account, but also attempts to schedule parallel per-hop transmissions as many as possible, so that the end-to-end effective deadlines can be met. We now discuss two other issues.

Scalability In our proposal, we have a leader node responsible for routing and scheduling decisions. This is a realistic model for small robotic teams. When group becomes large, the whole group can be split to multiple smaller groups. A hierarchical communication infrastructure is then used, and our algorithm can be used for communications at each level.

Mobility We handle the mobility by phasing the execution of each plan. In fact, in robotic applications, the speed of a robot is usually much slower than message transmission. Hence, robot can transmit messages while moving. However, we must make sure that the transmission can succeed during the movement, which means that network topology must not change even though two nodes may move in opposite directions. Due to the issues of speed, signal strength and routing, this is a problem that needs further study.

References

[1] A. E. Gamal, J. Mammen, B. Prabhakar, and D. Shah. Throughput-Delay Trade-Off in Wireless Networks. In *IEEE INFOCOM04*, Mar 2004.

[2] M.R. Garey and D.S. Johnson. *Computers and Intractability - A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company, 1979.

[3] M. Gastpar and M. Vetterli. On the Capacity of Wireless Networks: the Relay Case. In *IEEE INFOCOM'02*, pages 1577–1856, Jun 2002.

[4] Tian He, John A. Stankovic, Chenyang Lu, and Tarek F. Abdelzaher. SPEED: A Staleless Protocol for Real-Time Communication in Sensor Networks. In *International Conference on Distributed Computing Systems (ICDCS)*, May 2003.

[5] Kamal Jain, Jitendra Padhye, Venkata N. Padmanabhan, and Lili Qiu. Impact of interference on multi-hop wireless network performance. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 66–80, San Diego, CA, USA, 2003.

[6] Huan Li, Prashant Shenoy, and Krithi Ramamritham. Scheduling Communication in Real-Time Sensor Applications. In *IEEE Real Time Technology and Applications Symposium (RTAS)*. IEEE, May 2004.

[7] Jinyang Li, Charles Blake, Douglas S. J. De Couto, Hu Imm Lee, and Robert Morris. Capacity of Ad Hoc Wireless Networks. In *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, pages 61–69, Rome, Italy, July 2001.

[8] Chenyang Lu, Brian M. Blum, Tarek F. Abdelzaher, John A. Stankovic, and Tian He. RAP: A Real-Time Communication Architecture for Large-Scale Wireless Sensor Networks. In *IEEE Real Time Technology and Applications Symposium (RTAS)*. IEEE, September 2002.

[9] Haiyun Luo, Jerry Cheng, and Songwu Lu. Self-coordinating Localized Fair Queueing in Wireless Ad-Hoc Networks. *IEEE Transactions on Mobile Computing*, 3(1), January-March 2004.

[10] Haiyun Luo, Songwu Lu, Vaduvur Bharghavan, Jerry Cheng, and Gary Zhong. A Packet Scheduling Approach to QoS Support in Multihop Wireless Networks. *ACM Journal of Mobile Networks and Applications (MONET), Special Issue on QoS in Heterogeneous Wireless Networks*, 9(3), June 2004.

[11] R. Moraes, H. Sadjadpour, and J. J. Garcia-Luna-Aceves. Reducing Delay while Maintaining Capacity in Mobile Adhoc Networks Using Multiple Random Routes. In *Asilomar Conference on Signals, Systems, and Computers*, Pacific Grove, CA, Nov 2004.

[12] Michael Neely and Eytan Modiano. Capacity and Delay Tradeoffs in Mobile Ad Hoc Networks. In *(Invited Paper)IEEE BroadNets*, San Jose, CA, October 2004.

[13] Charles E. Perkins and Elizabeth M. Royer. Ad hoc On-Demand Distance Vector Routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, New Orleans, LA, February 1999.

[14] Saikat Ray, Jeffrey B. Carruthers, and David Starobinski. RTS/CTS-Induced Congestion in Ad Hoc Wireless LANs. In *Wireless Communications and Networking Conference (WCNC)*, March 2003.

[15] G. Sharma and R Mazumdar. On Achievable Delay/Capacity Trade-offs in Mobile Ad Hoc Networks. In *Conference on Wireless Networks and Optimization (WiOpt)*, Cambridge, UK, March 2004.

[16] John A. Stankovic. Research Challenges for Wireless Sensor Networks. *SIGBED Review: Special Issue on Embedded Sensor Networks and Wireless Computing*, 1(2), July 2004.

[17] John A. Stankovic, Tarek F. Abdelzaher, Chenyang Lu, Lui Sha, and Jennifer C. Hou. Real-time communication and coordination in embedded sensor networks. *Proceedings of the IEEE*, 91(7):1002–1022, 2003.

[18] J. Sweeney, H. Li, R. Grupen, and K. Ramamritham. Scalability and Schedulability in Large, Coordinated, Distributed Robot Systems. In *Proceedings of the 2003 IEEE Conference on Robotics and Automation, Taipei, Taiwan*, May 2003.

[19] S. Wang, R. Nathuji, R. Bettati, and Wei Zhao. Real-Time Guarantees over Wireless Links. In *Proceedings of the 24th IEEE International Conference on Distributed Computing Systems (ICDCS 2004)*, March 2004.