

# RushNet: Practical Traffic Prioritization for Saturated Wireless Sensor Networks

Chieh-Jan Mike Liang<sup>‡</sup>, Kaifei Chen<sup>\*</sup>, Nissanka Bodhi Priyantha<sup>‡</sup>, Jie Liu<sup>‡</sup>, Feng Zhao<sup>‡</sup>

<sup>‡</sup>Microsoft Research   <sup>\*</sup>University of California, Berkeley

{liang.mike, bodhip, jie.liu, zhao}@microsoft.com   kaifei@berkeley.edu

## Abstract

Network traffic prioritization is gaining attention in the WSN community, as more and more features are being integrated into sensor networks. Real-world deployment experience suggests that WSN brings new challenges to existing problems, such as resource constraints, low data-rate radios, and diverse application scenarios. We present the RushNet framework that prioritizes two common traffic patterns in multi-hop sensor networks: *low-priority (LP)* traffic that is large-volume but delay-tolerant, and *high-priority (HP)* traffic that is sporadic but latency-sensitive. RushNet achieves schedule-free and coordination-free delivery differentiations with the following features. First, RushNet works with most data collection protocols to deliver LP traffic. Second, RushNet leverages transmission power difference and radio capture effect to implement on-demand HP packet delivery with low overhead. Third, RushNet proposes a retrodiction technique to help nodes minimize the overhead of recovering LP packet loss due to concurrent HP traffic. We evaluate RushNet performance with micro-benchmarks and a crowd-sourced office comfort monitoring deployment. The deployment results suggest RushNet can achieve a throughput close to network capacity, and deliver 98% of the HP packets with a latency of less than four seconds.

## Categories and Subject Descriptors

C.2.1 [Computer-Communications Networks]: Wireless Communication

## General Terms

Design, Experimentation, Measurement, Performance

## Keywords

Traffic Prioritization, Capture Effect, Wireless Sensor Networks

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

SenSys'14, November 3–5, 2014, Memphis, TN, USA.  
Copyright is held by the owner/author(s). Publication rights licensed to ACM.  
ACM 978-1-4503-3143-2/14/11 ...\$15.00  
<http://dx.doi.org/10.1145/2668332.2668341>

## 1 Introduction

Wireless sensor networks (WSN) often have to support multiple application semantics with the same network resource. For example, in a structural monitoring sensor network [7, 11, 36], regular data collection interleaves with detected event reports. In cyber-physical systems such as building HVAC control, a normal air flow control cycle can be up to 15 minutes, while user inputs and alarms need to be handled as soon as possible.

It is common that, in a sensor network, majority data can tolerate large delays or jitters. We refer to them as *low-priority (LP)* data. Within the deadline constraints, they accept an eventual consistency semantics. Some other data, such as alarms, events, or commands, albeit sporadic, require low-latency delivery for near real-time reactions. We refer to them as *high priority (HP)* data.

In this paper, we use latency as the key traffic class differentiator because it implies both spatial and temporal constraints, more stringently than reliability and energy consumption. For example, a deployment demanding low latency mostly implies the use of delivery reliability mechanisms to bound the data arrival time at the gateway. However, a deployment with reliability mechanisms does not necessarily put constraints on the latency.

Mixing qualities of services in routing sensor data is not a big challenge when the network is over provisioned. Since the channel occupancy is low on average, one can regularly provision or statistically allocate resources for delay-sensitive packets. For example, in a Time Division Multiple Access (TDMA) network, this means allocating dedicated time slots for HP traffic. In a Carrier Sense Multiple Access (CSMA) network, this means incorporating different backoff time constants. Specifically, when channel contention happens, a LP data sender on average waits longer than a sender with an HP packet [23, 28]. Within a node, a common solution to prioritize traffic is buffer scheduling [21]. That is, a node will empty its HP buffers before sending any LP data.

However, the situation is different in a “busy” sensor network, where the data generation rate is close to the available network bandwidth. We call this a *saturated* wireless sensor networks. Previous work has shown that, for control applications, it is desirable to increase the sensing frequency to improve control performance and robustness [27], which means that the LP data can be collected towards network capacity. Frequent carrier sensing with a long backoff

time or constantly reserving TDMA slots reduces effective bandwidth significantly, which leads to a reduction in network throughput. In this case, statically allocating TDMA slots for sporadic events leaves chunks of network resources unused and wasted. Similarly for CSMA approaches, when channel occupancy is high, large back off intervals can lower the average network throughput.

Our work addresses the challenges of supporting mixed-priority traffic flows in large-scale and resource-constrained sensor networks that require high average throughput. We design **RushNet**, a schedule-free and coordination-free data delivery framework. It leverages two existing, but never integrated, techniques, namely *bulk transfer* and *capture effect*. Our basic idea is to first implement a high-throughput network backbone that delivers a large volume of LP data. Then, we use an in-the-air preemption mechanism to “squeeze in” sporadic HP packets, without stopping existing flows on the wireless medium. The LP and HP data deliveries are provided by the framework’s LP and HP transport services, respectively. RushNet has the following properties.

First, RushNet does not put strict assumptions on the implementation of the LP transport service. This allows network administrators to select the ones that fit their application requirements. While the popular Collection Tree Protocol (CTP) is a suitable choice, our current version of RushNet provides a LP transport service implementing token-based congestion-avoidance with hop-by-hop block transfers. As token-based data collection protocols can minimize various delays in packet transmissions [17], the network can deliver high-volume of LP data at high throughput. Empirical results show that RushNet can improve the network throughput by a factor of 4.75 over previous sensor network data collection systems.

Second, for the HP transport service to minimize the control overhead in sending an additional traffic class, the challenge is to reduce explicit coordination among potential transmitters. Our approach is to give the HP transport service the freedom to inject low-volume of HP packets at higher transmission power at any time, while leveraging the radio capture effect (c.f. § 3.1). In addition, RushNet goes one step further by proposing preemptive packet train, a technique that encourages the radio capture effect to happen on off-the-shelf IEEE 802.15.4 radio chips. The insight from real testbed measurements is that popular off-the-shelf 802.15.4 radio chips restrict capture effects to certain radio states only, i.e., preamble search. This restriction is also reported in previous work [1]. Preemptive packet train essentially repeats the same HP packet transmission in a certain way to influence the receiver’s radio state. Micro-benchmarks show that our techniques improve the packet reception ratio due to capturing by a factor of 10.

Finally, we explore other design considerations that stem from sending HP packets at a higher transmission power, as these packets essentially overpower concurrent transmissions in the same coverage area. RushNet presents a two-tier caching technique with loss retrodiction (i.e., estimating the loss probability after each transmission) to minimize the overhead of recovering lost LP packets. This technique is designed to run on resource-constrained sensing motes, and

adaptive to the amount of memory resources available on the deployed mote platform.

We perform end-to-end performance evaluations in an office sensor network deployment and show that 98% of the HP packets are successfully delivered, and the latency improved by a factor of 5 comparing to non-prioritization approaches. Although our results are from 802.15.4 radios, literature suggests that they are also applicable to 802.11 radios [12, 13] (c.f. § 9).

In summary, this paper makes the following contributions. (1) We describe a practical WSN traffic prioritization framework, RushNet, that minimizes control overhead by removing explicit node coordination and scheduling. RushNet targets resource-constrained mote platforms. (2) We present an approach that delivers high-priority packets by leveraging both transmission power difference and radio capture effect. In addition, based on empirical data, our approach incorporates a technique called preemptive packet train to improve the success rate of triggering radio capture on off-the-shelf 802.15.4 radio chipsets. (3) We present a technique, two-tier caching with packet loss retrodiction, to speed up the recovery of packets lost due to traffic prioritization.

The rest of the paper starts with a discussion on the design rationales in Section 2, which talks about design requirements and why related work cannot satisfy all requirements. Section 3 discusses how RushNet leverages radio capture effect to transport HP data without explicit node coordination. Section 4 continues the discussion on the HP transport service and discusses additional design considerations. Section 5 presents our current implementation of RushNet including both LP and HP transport services. Section 6 and Section 7 show results from both micro-benchmarks and a crowdsourced indoor WSN deployment. Finally, we survey related work in Section 8, discuss some of the overarching issues related to RushNet design in Section 9, and conclude in Section 10.

## 2 Design Rationales

This section first discusses the requirements of a practical traffic prioritization scheme for wireless sensor networks, then elaborates on why related efforts may not satisfy all these requirements.

### 2.1 Design Requirements

Traffic prioritization aims for high-priority data packets to have a lower end-to-end delivery latency, as compared to low-priority data packets. We believe that WSN brings additional design requirements to this problem.

First, the traffic prioritization scheme should **minimize assumption on the capabilities of resource-constrained sensing motes**. Most real-world sensing networks deploy resource-constrained sensing nodes for their low hardware cost and small footprint. In fact, this trend is one factor that has been pushing sensor networks to scale up. In the context of traffic prioritization, resource constraints imply that some sensing nodes might not be able to perform complex network scheduling or queuing strategies.

Second, the traffic prioritization scheme should **minimize the control overhead in the network**. This requirement is partially driven by the first requirement, in that control over-

head can speed up depletion of the limited energy budget on sensing nodes. Another motivation comes from the observation that radios on sensing nodes typically have a low data rate. As such, the amount of control traffic on the network has a direct and potentially significant consequence on the network throughput, especially in a large sensing network where the data generation rate is close to network capacity.

Third, the traffic prioritization scheme should **impose minimal restrictions and assumptions on the underlying data collection design and operations**. Application scenarios on sensor networks are so diverse that a single vertical solution might not suffice. For example, the expected network congestion level can influence whether congestion avoidance or congestion control should be implemented. Therefore, the traffic prioritization scheme should not impose certain data collection designs, as doing so can impact system practicality.

## 2.2 Existing Traffic Prioritization Approaches

We discuss two categories of common traffic prioritization approaches in WSN: traffic-aware topologies and on-demand coordination. This discussion focuses on why current approaches cannot adequately satisfy the requirements listed previously, and it motivates the need for RushNet.

### 2.2.1 Traffic-Aware topologies

This category of approaches assumes prior knowledge of certain network characteristics (e.g., node locations and traffic volume of each traffic class). This knowledge can be difficult to maintain for reasons such as resource constraints. In addition, the effectiveness of traffic prioritization can degrade if the actual network characteristics deviate from the expected.

Multi-Path and Multi-SPEED Routing Protocol (MMSPEED) [6] provides traffic prioritization in two metrics: reliability and timeliness. MMSPEED implements two separate mechanisms for each metric. For the reliability metric, MMSPEED nodes build a multi-path tree, and they forward data packets by picking the path that best satisfies the specified reliability requirement. For the timeliness metric, MMSPEED protocol assumes that all nodes in the network know their geo-locations, and nodes forward data packets by using a geographic forwarding scheme on local neighborhood information. In addition, MMSPEED proposes dynamic compensation to minimize the impact on the end-to-end routing performance, in case of errors from local routing decisions. We believe that there are concerns on the practicality of MMSPEED in some real-world deployments. First, the requirement of localizing all nodes might not be feasible in many real-world deployments, especially with resource-constrained sensing nodes. Second, it is not clear how errors in local decisions would be amplified in long routing paths, especially as the sensing network scales up.

TEEN [20] implements a different approach to build routing topologies that are aware of traffic prioritization. Specifically, TEEN uses a TDMA schedule to allocate dedicated time slots to each node for the purpose of transmitting high-priority data. The number of these high-priority TDMA slots allocated depends on a complete knowledge of network traffic patterns. However, in many sensing networks, high-

priority data packets (e.g., alerts and alarms) can be generated in unpredictable time intervals. An example is the server removal alert of an asset tracking network at data centers.

### 2.2.2 On-Demand Coordination

This category of approaches reacts to network dynamics, but node coordination can add overhead and delay to data transmissions.

Prioritized Heterogeneous Traffic-oriented Congestion Control Protocol (PHTCCP) [21] solves the problem of traffic prioritization as a rate control problem. Specifically, PHTCCP assumes a tree topology, and it implements a hop-by-hop explicit rate control triggered by link congestion detection based on the rate at which a node's children are uploading packets of different priority. If a child node is not able to flush data from the high-priority queue fast enough, the parent can send out rate adjustment messages to slow down its other children. PHTCCP is not suitable for sensing networks where the occurrence of high-priority traffic is sporadic and unpredictable, as the timely delivery of a high-priority packet might not justify the overhead in on-demand detection and adjustment. In addition, the on-demand detection and adjustment add additional delays before the high-priority packet can be transmitted.

3-Hop-Ahead [9] is an on-demand path reservation scheme. It uses a contention-based MAC for regular data collection traffic, but reserves the path if high-priority traffic is generated. Since 3-Hop-Ahead reserves paths on demand, there is an additional setup delay before the high-priority packets can be transmitted. In addition, the presence of high-priority packets suppresses other concurrent transfers within the 3-hop range to reduce interference, but the radio medium utilization over time might be reduced due to idle gaps on the medium.

## 3 RushNet Basic Design

RushNet exposes an HP transport service and a LP transport service to applications on sensing nodes. As the LP transport service works with most data collection protocols, the challenge is on designing the HP transport service.

High-priority traffic carries latency-sensitive data and alerts. One simple approach is to piggy-back high-priority data onto existing low-priority traffic, but the uncertainty in low-priority data delivery latency would be transitive to high-priority data. RushNet takes a different approach: nodes can send HP data packets at any time, even if the wireless medium is busy. Specifically, we improve the packet reception ratio (PRR) of HP packets by using both transmission power difference and radio capture effect.

This section first motivates and addresses the technical challenges in using the capture effect on popular off-the-shelf 802.15.4 radios, with experimental results from popular radio chips: TI CC2420/CC2520 [32] and Atmel RF230/RF231 [4]. Then, this section describes our solution that implements coordination-free and schedule-free HP transport service.

### 3.1 Background on Radio Capture Effect

Capture effect is a property of the radio transceiver, which can decode a stronger signal from one transmitter in the presence of a weaker signal from another transmitter. The cap-

ture effect was first discussed in the context of Frequency Modulation (FM) radios [15], and WSN and wireless communities have observed a similar effect on other radios.

Previous work has demonstrated the radio capture effect on certain low-power radios [25, 34] – TI CC1000/1100 radio chip [31]. We note that TI CC1000/1100 has a software radio stack, which exposes rich information about each incoming data bit to the upper layers. However, for reasons of performance and system integration simplicity, most 802.15.4 radios have a hardware stack. As we explain next, such a hardware stack complicates systems that attempt to leverage the radio capture effect.

### 3.2 Challenges with 802.15.4 Radio Chipsets

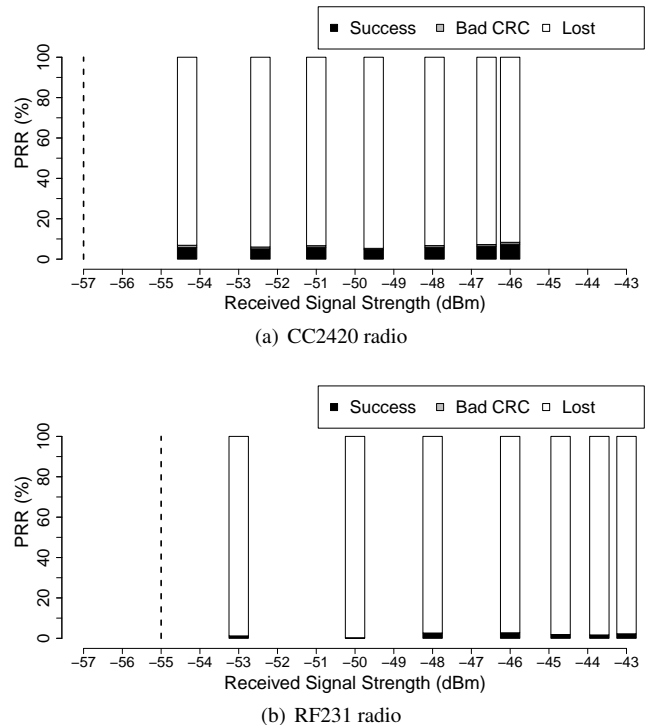
Our discussion focuses on TI CC2420/CC2520 [32] and Atmel RF230/RF231 [4], the two most popular off-the-shelf 802.15.4 chipsets in the WSN community. The community has shown that if the colliding packet appears after the preambles of the current packet transmission, then the receiver does not receive the colliding packet, irrespective of the received signal strength (RSS) [1]<sup>1</sup>. This is different from observations on non-802.15.4 radios reported by the wireless community, and there are two main reasons for this observation.

First, the hardware stack of CC2420 implements a state machine, with distinct modes of synchronization header decoding and payload reception. During the synchronization header decoding mode, CC2420 searches for preambles and start frame delimiter (SFD) with the strongest RSS. Preambles allow transmission timing synchronization on both ends, and the 1-byte SFD marks the start of a frame. At this point, CC2420 generates an interrupt and switches to payload reception mode. Specifically, CC2420 treats all subsequently demodulated bits as payload bits, and it does not search for synchronization headers.

Second, the 802.15.4 spreading reduces the extent of software-based recovery possible from packet corruptions, due to collisions. Considering that CC2420 treats all bits in the payload reception mode as payload bits, it is theoretically possible for the colliding packet to be embedded in the corrupted packet. In other words, the first half of the corrupted packet would contain the original data, and the second half would be overwritten by the stronger packet. Unfortunately, due to the 802.15.4 bit spreading, this approach is difficult to realize in the real world, as explained next.

The 802.15.4 standard specifies a spreading procedure mapping 4-bit symbols to 32-bit chip sequences [8]. Hence, there are 16 expected chip sequences. The de-spreading process would require the receiver to match each received chip sequence to the closest expected chip sequence. The spreading procedure highlights the importance of chip timing synchronization. If the incoming frame has a different symbol boundary alignment than expected (e.g., the receiver switches to a stronger signal without recalibrating the transmission timing), the receiver will interpret the wrong group of chip bits. In addition, since off-the-shelf 802.15.4 radios

<sup>1</sup>In a quiet environment, there is no external interference. For ease of discussion, we use the term received SINR (signal-to-interference-plus-noise ratio) and RSS interchangeably.



**Figure 1. Observed PRR due to capture effect. X-axis shows the RSS of packets from  $transmitter_2$ , and the dotted line represents the RSS of packets from  $transmitter_1$ . Naively sending packets does not give good radio capture effect, regardless of the power difference.**

do not expose the raw chip bits received, it is difficult to perform packet recovery.

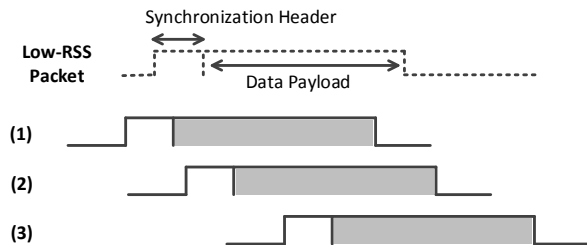
Next, we quantify the impact on capture performance due to the limitations of off-the-shelf 802.15.4 radios.

#### 3.2.1 Empirical Results

We carry out experiments on both CC2420 and RF231 radio chipsets. Testing on more than one radio chip minimizes any hardware-specific artifact. For the former, we use the TelosB platform [24] running TinyOS 2.1 [2]. For the latter, we use the RF231 development board with the Cortex M3 development board from STMicroelectronics, which runs our own drivers and applications in C. In both cases, we try to create transmission collisions by disabling the CSMA and the CCA check prior to the transmission. Furthermore, receivers disable CRC validation and filtering to receive corrupted packets for inspection. The transmitter sends maximum-sized 802.15.4 packets, or 128 bytes.

We should note that, since both transmitters are equidistant from the receiver, the received SINR (signal-to-interference-plus-noise ratio) difference should be close to the transmission power difference in a quiet environment. Therefore, the transmission power is an experiment parameter. In fact, in a quiet environment, the received SINR is equivalent to the received signal strength (RSS).

In the experiment, we instrument  $transmitter_1$  to continuously send packets as fast as possible, while  $transmitter_2$



**Figure 2.** Depending on timing, there are three ways that a high-RSS packet and a low-RSS packet can collide. The high-RSS packet can (1) come in before the low-RSS packet, (2) during the synchronization header searching, or (3) during the reception of the low-RSS packet payload. The dotted and solid lines depict low-RSS and high-RSS packets, respectively.

sends a packet every second. Transmitters are 100 cm apart and on the opposite sides of the receiver. Then, we fix the transmission power of  $transmitter_1$  ( $-11$  dBm and  $-9$  dBm for CC2420 and RF231, respectively) while increasing that of  $transmitter_2$  (from  $-11$  dBm to  $0$  dBm and from  $-9$  dBm to  $3$  dBm for CC2420 and RF231, respectively). The receiver records all packets heard from both transmitters. Ideally, as the output power difference increases, the receiver should see more packets from  $transmitter_2$ .

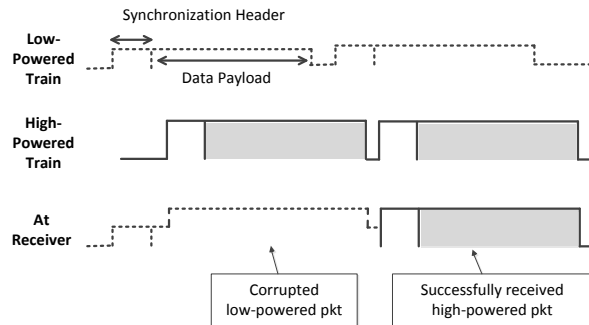
Figure 1(a) and Figure 1(b) show the experiment results. First, both radios have poor capture effect performance, which is at most 10% PRR from  $transmitter_2$ . Second, the capture effect performance does not improve, regardless of the received signal strength differences. These observations suggest that naively sending packets with different output power does not guarantee radio capture on off-the-shelf 802.15.4 radios, which motivated us to explore *preemptive packet train* as the technique for encouraging capture effect.

### 3.3 Preemptive Packet Train

There are two factors in successfully receiving a packet of higher received signal strength (RSS),  $Pkt_{high\_RSS}$ , in the presence of a lower-RSS packet,  $Pkt_{low\_RSS}$ . First, the power difference between  $Pkt_{high\_RSS}$  and  $Pkt_{low\_RSS}$  needs to be above a certain threshold. Second, the transmitter should send  $Pkt_{high\_RSS}$  when the receiver is searching for frame synchronization headers in the air. Since we derive the necessary RSS difference threshold from real-world experiments, we defer the relevant discussion to Section 6.1.1. The rest of this section focuses on the second requirement.

Rather than simply transmitting high-priority packets with higher output power, RushNet transmits the same high-priority packet multiple times with small intervals. The goal is for the receiver to successfully capture at least one packet in this train, even in the presence of lower-RSS low-priority packets. We next discuss the theory behind this idea by considering three different ways that two packets can collide.

In Figure 2, both the first and second case happen before the receiver enters the data payload reception mode. If a  $Pkt_{high\_RSS}$  appears between two consecutive  $Pkt_{low\_RSS}$ , then



**Figure 3.** The first high-RSS packet prevents the receiver from locking to the second low-RSS packet, and forces the receiver to lock onto the second high-RSS packet.

the channel should be sufficiently quiet for the receiver to successfully capture the synchronization header. Even if the receiver is in the process of locking to the preambles of a weaker signal, a stronger signal can force the receiver to resynchronize. However, if the receiver has already started receiving the data payload of  $Pkt_{low\_RSS}$ , then the receiver will not resynchronize to  $Pkt_{high\_RSS}$ . Therefore, we control the timing of when the receiver finishes locking to a synchronization header, as explained next.

For ease of illustration, we will consider a high-priority packet train length of two (c.f. Figure 3). Since  $Pkt_{high\_RSS}$  starts after  $Pkt_{low\_RSS}$ , the tail of  $Pkt_{low\_RSS}$  overlaps with the body of  $Pkt_{high\_RSS}$ . The portion of  $Pkt_{high\_RSS}$  after the tail effectively acts as channel background noise, and it prevents the receiver from locking onto a new 802.15.4 synchronization header. In fact, since the interval between two consecutive  $Pkt_{high\_RSS}$  is small <sup>2</sup>, the receiver cannot fully lock onto the synchronization header of another  $Pkt_{low\_RSS}$ . Experiment results from Section 6 show a 70% improvement in high-RSS packet reception with a train length of four.

**Integration into HP Transport Service.** As the receiver receives the HP packet payload inside the preemptive packet train, it immediately relays the payload upstream with another preemptive packet train. We note that the receiver needs to check the payload sequence number to filter out duplicated data, due to repeated transmissions from the train.

Link-level reliability is implemented with overhearing, by taking advantage of the fact that the receiver would immediately relay HP packets upstream. Specifically, the transmitter overhears on the radio medium for 50 ms to determine whether its parent has relayed the HP packets.

## 4 Additional Design Considerations

As described previously, the basic idea behind RushNet is to leverage the transmission power difference and radio capture effect to improve the delivery ratio of high-priority packets. While the high delivery rate and low coordination over-

<sup>2</sup>The smallest inter-packet interval on TelosB is about  $600 \mu s$ , considering the time to send STXON(CCA) command over the SPI bus and the 12 symbol periods that CC2420 waits before each transmission.

heads are attractive, there are other design considerations in adopting this approach, as elaborated in this section.

## 4.1 Impacts on Concurrent Network Traffic

Transmitting high-priority packets with higher transmission power effectively overpowers any concurrent traffic stream within a radio coverage. We delve into this problem from two perspectives where the co-locating traffic stream is either low-priority or high-priority.

### 4.1.1 On Concurrent Low-Priority Traffic

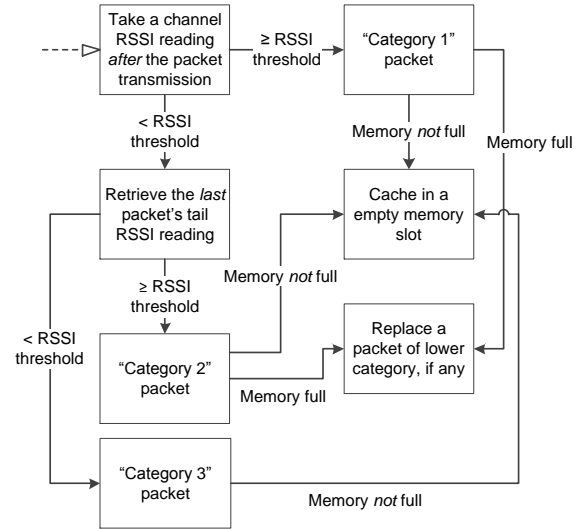
Overpowering current transmissions can increase the interference region, which then causes packet losses for the underpowered traffic streams. Most transport protocols implement collision avoidance mechanisms to resolve transmission interference in a radio coverage area. For example, the sender can explicitly request radio medium access from its neighbors, or the sender can implicitly infer free radio medium access by listening for activities. However, as RushNet aims to minimize control overhead and delivery latency, we propose a **packet recovery scheme with two-tier caching** that is applicable to most LP transport protocols used.

To recover missing packets, packet acknowledgments are the most common approach. In the case of negative acknowledgments (NACKs), since the transmitter streams packets of a data block sequentially, the receiver can determine missing packets by looking at the sequence number of all received packets. At the end of the entire data block transport, the receiver immediately sends one NACK with a bit map indicating missing packets. Then, the sender retransmits according to the NACK bit map. However, resource constraints on sensor nodes pose challenges in implementing retransmissions efficiently.

Since sensor nodes have a limited amount of memory (RAM), data most likely reside on the external flash<sup>3</sup>. However, retrieving a piece of data might involve multiple reads from the flash over the SPI bus, depending on the storage abstraction (e.g., the Log abstraction in TinyOS 2). This bus access overhead in time grows with the amount of data to retrieve. One way to mitigate this problem is caching packets of the current transfer session in memory. However, with the memory constraint, nodes need to intelligently decide which packets to cache.

RushNet implements a two-tier cache hierarchy, in which memory sits on top of the flash and caches outgoing packets that might be lost during the current block transmission session<sup>4</sup>. In other words, packet transmissions that are likely to have been lost should be cached in memory to speed up the recovery later. The challenge is for the sender to estimate this packet loss probability for each outgoing packet on-line (or called *loss retrodiction*).

We note that concurrent HP transmissions are the most probable cause of packet losses for RushNet’s LP traffic. Since most data packets have the same size, the colliding HP transmission would overlap with either the head or the tail of



**Figure 4. Packet loss retrodiction algorithm based on packet tail RSSI readings. The input is the channel RSSI sampled immediately after each packet reception.**

the current transmission. Therefore, the sender samples the channel noise level, or the received signal strength indicator (RSSI), *after* each packet transmission. We call these readings the “tail” RSSI. The sender then infers loss probability from these readings, as discussed next. Our empirical results show a retrodiction accuracy of > 95% with this approach (c.f. § 6).

All outgoing packets are classified according to their probability of being successfully received by the receiver. *category 1* packets are very likely to be lost, and *category 3* packets are unlikely to be lost during the transmission. The classification algorithm works as follows (c.f. Figure 4).

First, if a packet’s tail RSSI is above a threshold, then an overlapping transmission might have occurred. So, the packet is classified as category 1. Then, for packets failing the first test, the sender checks whether the previous packet’s tail RSSI was above a threshold. In which case, a concurrent transmission might have occurred at the head of the current packet. So, the packet is classified as category 2. RushNet caches both category 1 and 2 packets if there is still space available on the memory, and exercises replacement policy otherwise. The replacement policy specifies that a category 1 packet can replace a category 2 packet, but not vice versa.

The main challenge is that, false positives and false negatives can occur if the tail RSSI threshold is too low or too high, respectively. The general guideline is to consider the network density, and setting a higher threshold if the density is higher. The reason is that, with a shorter inter-node distance, the signal attenuates slower. We empirically determine this threshold for our set up in Section 6.2. In addition, we leave it as future work for nodes to learn this threshold on-line based on the HP packets that they overhear. We also note that this packet classification also works in a noisy environment, as the RSSI of overlapping traffic should still be

<sup>3</sup>Most sensor nodes have external flash. For example, TelosB has a 1 MB external flash.

<sup>4</sup>Our current implementation on TelosB allocates 2 KB of RAM to first-level caching.

higher than otherwise.

#### 4.1.2 On Concurrent High-Priority Traffic

Another interesting design consideration is the impacts on concurrent high-priority traffic. While HP traffic is low-volume and sporadic, it is possible for two nodes in a neighborhood to initiate HP packet transmissions at the same time. In addition, the problem is worse because RushNet nodes do not perform any coordination before transmitting.

To prevent competing HP flows, RushNet dynamically adjusts the CCA threshold as follows. Depending on the traffic class of the outgoing packet, the sender increases or lowers the CCA (clear channel assessment) threshold accordingly. With a higher CCA threshold, nodes detect existing HP packets in the air, but not LP packets. In the case of a busy medium, nodes defer pending HP transmissions following exponential backoffs. Therefore, similar to other CSMA-based protocols, the amount of HP traffic that a network can support depends on both the network density and the frequency of HP transmissions.

To detect failed HP transmissions, the sender leverages the fact that the receiver would immediately relay successful transmissions. Specifically, after the transmission, the sender overhears on the radio medium for 50 ms to determine whether its parent has relayed the HP packets.

### 4.2 Effectiveness in the Presence of Cross Radio Technology Interference

From our experience with real-world WSN deployments, we have observed 2.4 GHz Wi-Fi (or 802.11) networks co-locating with many indoor deployments such as office buildings. The 802.11 network in most cases does not saturate the entire 2.4 GHz spectrum, except places such as crowded libraries. An interesting consideration is whether RushNet can still adequately perform traffic prioritization in the presence of such Cross Technology Interference (CTI) in the real-world.

First, since packets of higher priority are transmitted with a higher transmission power, the probabilities of successful delivery of high-priority packets is still higher than that of low-priority packets. The main reason is that high-priority packets still have a higher received SINR (signal-to-interference-plus-noise ratio). This property is important as it satisfies the fundamental goal of prioritizing traffic.

Second, we argue that the preemptive packet train from 802.15.4 radios can trigger the same back-off effect on 802.11 radios in some cases. Following Liang et al. [18], we experimentally showed that when a 802.15.4 node is close to an 802.11 transmitter, an 802.15.4 packet can actually cause the 802.11 transmitter to back off, due to the elevated channel energy. When this happens, 802.11 corrupts only the first packet of the 802.15.4 packet train, and leaves the rest of train unaffected.

### 4.3 Impacts from Co-locating 802.15.4 Networks

It is possible that another 802.15.4-based WSN deployment co-locates with the RushNet network. Intuitively, transmissions of the *same* power on the *same* channel from the co-locating deployment can limit the occurrence of radio capture (hence the PRR). Fortunately, sending HP trains naturally

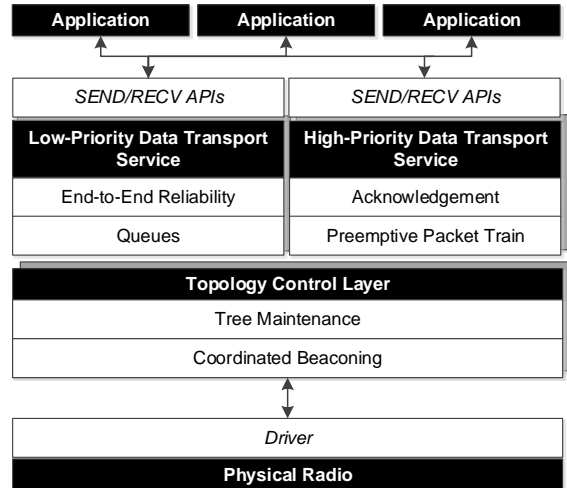


Figure 5. RushNet framework.

addresses this concern. Specifically, since a packet train lasts longer than a single packet transmission, only some packets of the train are corrupted. Other packets of the train will trigger CSMA backoff prior to the next packet transmission from the co-locating network.

We note that one related concern is the hidden terminal problem, especially its potential impact on the accuracy of packet loss retrodiction. In addition to assigning different channels to co-locating networks, we are exploring ways to improve RushNet’s resilience to this problem.

## 5 Implementation

Figure 5 illustrates the RushNet framework. The lowest layer is the tree topology control that maintains bi-directional routing trees for data transfers between nodes and the gateway at root. Two data transport services sit above the topology control, and they address the requirement of the two traffic classes separately: low-priority (LP) and high-priority (HP) data traffic. Both services expose the SEND/RCV API to applications running on the sensing node. We implement the current system with TinyOS 2.1.

We note that RushNet does not impose restrictions nor assumptions on the implementation of the tree topology control and the LP transport services. Section 5.1 and Section 5.2 describe their current implementation in our system. Then, Section 5.3 presents our implementation of the high-priority data transport service.

### 5.1 Topology Control

The topology control module maintains robust data routing trees rooted at gateways. Before any data transfer is possible, a (non-tree) node needs to join a tree and establish the parent-child relationship with a tree node.

**Parent Selection.** Gateways initiate routing tree construction on their channels by broadcasting HEARTBEAT messages. HEARTBEATs advertise the node’s status, including the distance and path quality to the root, parent node ID, and the number of children. Upon receiving a HEARTBEAT

message, a non-tree node computes the *path expected transmission count* (PETX) as the *end-to-end path* quality via the sender. PETX is the sum of expected transmission counts of all links on the end-to-end path, and it is based on the Link Quality Indicator (LQI) available on all 802.15.4 radio chips [26]. RushNet nodes select the upstream neighbor with the smallest PETX as their parent.

To reduce collisions due to broadcast storms in dense networks, RushNet employs a reduced contention mechanism. Specifically, a time slot of length  $T$  starts immediately after a node  $P$  broadcasts its HEARTBEAT message.  $T$  is further divided into two uneven sections proportional to the number of children that  $P$  already has and the number of additional children that  $P$  can support. Nodes that receive  $P$ 's HEARTBEAT randomly select a time within the appropriate section to broadcast their HEARTBEAT.

**Routing Tree Construction.** A non-tree node starts by scanning channels for HEARTBEATs from either the gateway or tree nodes. Then, it selects the upstream neighbor with the smallest PETX as its potential parent and initiates a TREEJOIN request. Since the routing tree is for bi-directional traffic flows, the parent also estimates the link quality of this non-tree node in the upstream direction before replying with a GRANT message. This is to eliminate asymmetric wireless links. With this two-way handshake, both ends explicitly agree to forward traffic for each other.

If a node fails to deliver a packet to its parent after some attempts, it enters the quick recovery phase, by repeating the protocol described to find another parent (without disregarding its children). To prevent routing loops, the new parent should have a hop count equal to or less than the previous parent. If there is no suitable parent (under the hop count constraint), the node enters the full recovery phase where it disconnects from its children and evaluates any tree node for potential parent.

## 5.2 Low-Priority Data Transport Service

After the topology control module constructs the bi-directional tree, the LP transport service forwards low-priority traffic on this tree.

Low-priority traffic has the characteristics of being delay-tolerant and high-volume, which suggest that network throughput is the most important metric. To this end, we aim to improve network throughput with two mechanisms: a *token-passing mechanism* to reduce network contentions and prevent congestion, and a *hop-by-hop transmission mechanism* to reduce inter-packet transmission delays.

**Token-Passing Mechanism.** Since a node can transmit only if it holds a token, token-passing effectively allocates the full radio medium access to a single transmitter. While a TDMA approach can also achieve the same goal, it incurs overhead from setting up the schedule, and inefficiency from idle windows due to the owner having no pending packets.

The token-passing mechanism does not require gateways to have prior knowledge of the network. Rather, it relies on the network to determine the next node that should hold the token. This property removes the overhead of having a

separate operation mode to learn the network topology. The protocol works as follows.

Gateways initiate a data collection round on their channels by passing the token to the first node on their child list. Tokens traverse the tree in a depth-first order, which results in fewer packet passes than the breadth-first order. Upon receiving the token, the node immediately passes it sequentially to all of its children in the tree. After all children finish transmitting their measurements, the node streams sensor data accumulated since the last data collection round to the gateway.

Even with aggressive link-level retransmissions, the network can still lose the token for reasons such as node failure. Since nodes stream data only when they hold the token, the gateway infers token loss with a timer on data reception. The gateway recovers by regenerating a token with the same 32-bit ID. In case of false assumption, the duplicated token does not cause problems to the network, as nodes immediately release a token if they have already held one with the same ID. Finally, to prevent the gateway from timing out, nodes send an empty packet if they do not have pending data.

**Hop-by-Hop Block Transport.** When a node holds the token, it streams to the parent all data accumulated (either from child nodes or onboard sensors) since the last streaming.

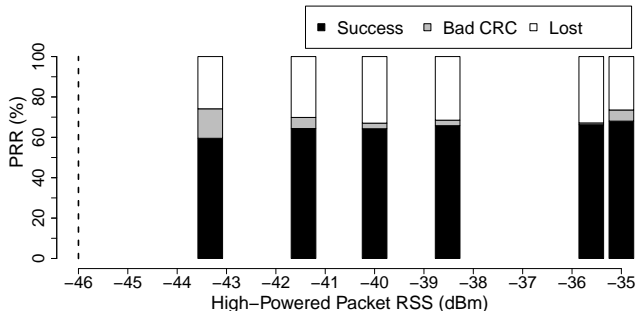
RushNet optimizes the link throughput by minimizing inter-packet delays and idle time. First, from a high-level view, senders execute the following steps: fetching data from the flash to the MCU, loading data to the radio TX-FIFO buffer, transmitting the data in the air. Osterlind et al. identified the SPI bus access as the bottleneck on the TelosB platform [22]. So, RushNet overlaps radio transmissions with flash reads over the SPI bus. Second, by limiting the number of concurrent transmitters, RushNet nodes do not perform carrier sensing and back offs, except for the first packet of a stream.

## 5.3 High-Priority Data Transport Service

Upon receiving data packets from applications via the SEND API, the HP transport service queries the topology control layer to get the next hop node ID. Each transmitted packet carries an ID that allows the receiver to filter out duplicates, due to repeated packet transmissions by the preemptive train.

We currently maintain a transmission power difference of about 5 dBm for sending high-priority and low-priority packets, which is more than the 3 dBm minimum threshold experimentally obtained (c.f. § 6.1.1). To minimize the gap between packets in a preemptive train, we “short-cut” the transmission process. The normal process consists of loading the pending packet to the 802.15.4 radio chip via a bus (e.g., SPI bus), and strobing the radio chip to send the loaded packet content. Fortunately, as the preemptive train carries multiple copies of the same packet content, we can avoid the slow bus by simply strobing the radio chip multiple times. This behavior is supported by most 802.15.4 radio chips such as TI CC2420.





**Figure 6. A minimum RSS difference of 3 dBm is necessary for the receiver to capture an HP preemptive train with a length of two, in the presence of background LP traffic (dotted line). Since the results are collected in a clean lab environment, RSS is equivalent to SINR.**

## 6 Design Evaluation

In this section, we first study system parameters and then verify the system design on a 5-node lab testbed. The testbed gives us full realism that simulators cannot match.

### 6.1 Parameters

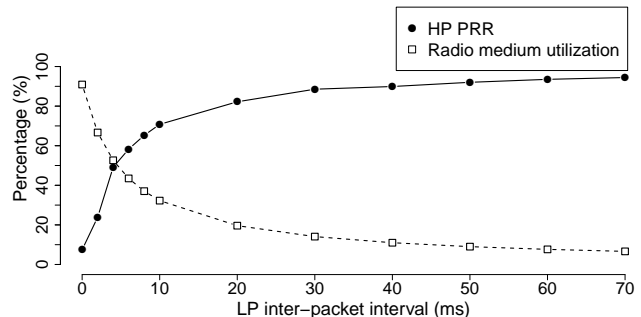
There are three main system parameters: (1) *packet transmission power*, (2) *inter-packet interval of the LP traffic*, and (3) *length of the HP preemptive packet train*. The experiment is repeated for both TI CC2420 and Atmel RF231 radio chipsets. In both cases, we instrument three radio nodes: one receiver and two transmitters for the two traffic classes. Both transmitters do not exercise CSMA, and they are 25 cm away from the receiver to minimize the signal path loss.

#### 6.1.1 Packet Transmission Power

The received SINR is a deciding factor of whether a packet reception will be successful. In a clean lab environment, SINR is equivalent to the received signal strength (RSS). In the context of radio capture, previous work reported that a minimum RSS difference of 3 dBm is necessary for the capture effect to happen on 802.15.4 radios [1]. Our empirical results align with this conclusion, as presented next.

Figure 6 presents the packet reception ratio (PRR) of the HP packet flow in the presence of continuous LP background traffic. The continuous LP traffic is made possible by repeatedly strobing the radio chipset to send. As mentioned before, both the LP and HP transmitter are placed near the receiver 25 cm away, to minimize the signal path loss. The HP transmitter randomly injects a preemptive train length of two at an average interval of 10 seconds. As the RSS power difference reaches 3 dBm, the receiver starts to receive HP packets. Based on this result, our deployment in Section 7 transmits HP data traffic at 0 dBm (the highest power level on CC2420), and  $-5$  dBm for LP data traffic.

Delving into Figure 6, we notice that the PRR does not improve further as the power difference increases beyond 3 dBm. As discussed previously, while a sufficient SINR difference is a prerequisite, the hardware limitation of the 802.15.4 radio chips is also a deciding factor. This insight motivated us to explore additional network parameters to lift



**Figure 7. Increasing the LP inter-packet interval improves the PRR of HP packets, but it also lowers the radio medium utilization. This experiment fixes the HP preemptive train length to be 1.**

this hardware limitation, as discussed in the next section.

#### 6.1.2 Data Transmission Parameters

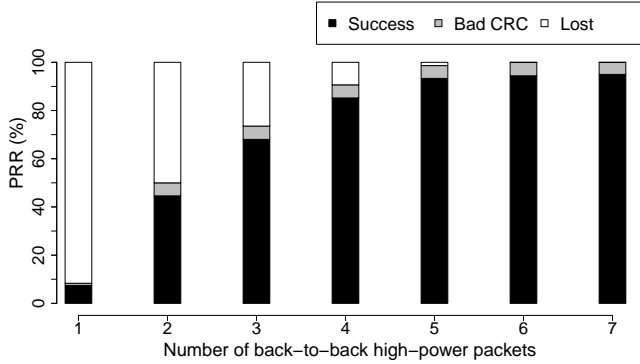
We start by discussing impacts of two train parameters: the inter-packet interval of the low-powered train and the length of the preemptive train. Then, we use empirical results to drive the parameter value selection in real-world deployments.

**Inter-Packet Interval of the LP Traffic.** This interval specifies the duration of idle space between two consecutive LP packets. Increasing this interval reduces the channel medium utilization, thus more opportunities for the HP packets’ synchronization header to be successfully detected by the receiver. We reuse the previous experiment setup and randomly transmit one HP packet at an average interval of 10 seconds. At the same time, we vary the LP inter-packet interval between 0 ms and 70 ms.

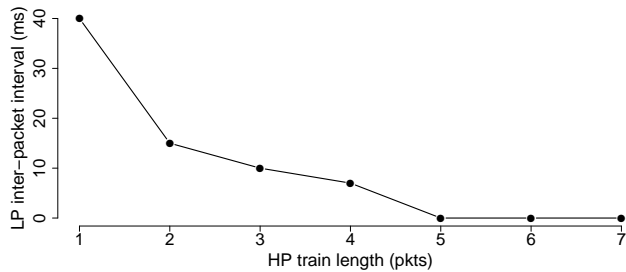
Figure 7 presents results that confirm our intuition: slowing down LP traffic improves the HP packet reception ratio. However, naively increasing this parameter can impact the overall network performance for the following two reasons. First, there is a diminishing return on the HP packet reception ratio as the inter-packet interval grows (in this case, 50 ms). Second, as Figure 7 shows, the throughput reduces as the interval grows. In fact, since high-priority packets are sporadic, having a large LP inter-packet interval is not always beneficial. These observations motivated us to investigate the impact of varying HP preemptive train length, as presented next.

**Length of the HP Preemptive Packet Train.** HP train length specifies the number of times that RushNet repeatedly transmits a particular packet. As discussed previously, the purpose of this repetition is to control when the receiver finishes locking on a synchronization header. Therefore, the longer the HP train length, the more opportunities for the receiver to detect the transmission. We use the same experiment setup as before, and we fix the LP inter-packet interval to be minimum, or  $600 \mu\text{s}$  on the CC2420 radio. Then, we vary the HP train length from one to five.

Figure 8 shows a rapid increase in the HP packet reception ratio as the train length increases. To match the packet



**Figure 8. Impacts of the HP preemptive train length on its packet reception ratio. We fix the LP inter-packet interval to be minimum, or 600  $\mu$ s on CC2420 radio.**



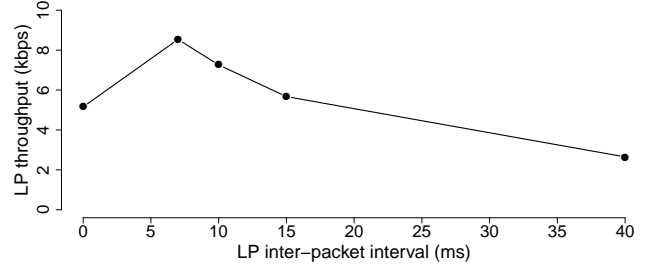
**Figure 9. The minimum LP train interval necessary to achieve at least 95% PRR of HP packets, for different HP train lengths.**

reception ratio of a high-powered train with a length of four, the LP traffic needs to slow down by a factor of 10. Picking an HP train length of four suggests a PRR close to 90% PRR, as redundant packets (1) temporarily stop neighboring nodes from sending and (2) offer multiple chances for receiver to hear the transmission. Finally, Figure 8 shows the diminishing return at a length of five.

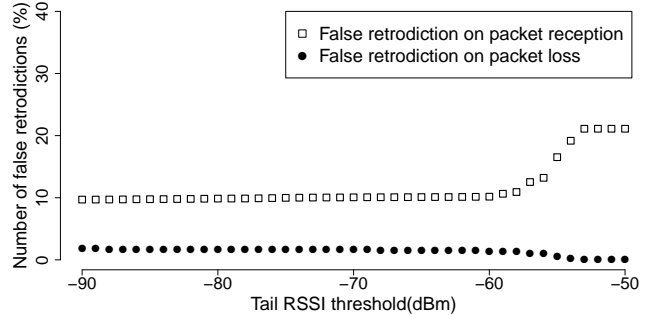
**Optimal Deployment Settings.** Setting the LP packet interval and the HP preemptive train length is not trivial, especially since there is a strong correlation between them. For example, lowering the LP inter-packet interval implies increasing the HP train length to maintain the same HP packet reception ratio.

Figure 9 shows the minimum LP train inter-packet interval necessary to achieve at least 95% PRR of HP packets, for different HP lengths. For reference, Figure 10 plots the LP network throughput for each corresponding LP inter-packet interval. We note that the peak of LP throughput (before the HP train length of four) is due to the receiver being unable to keep up with the transmission, at a low LP inter-packet interval.

For our deployments with the TelosB platform, the sweet spot is the intersection of an HP train length of four and an LP inter-packet interval of 8 ms. In addition, this 8 ms transmission interval is close to the time for MCU to load packets over SPI bus access.



**Figure 10. The LP throughput for different LP inter-packet interval.**



**Figure 11. The packet loss retrodiction confidence for each tail RSSI threshold in terms of false positives and false negatives.**

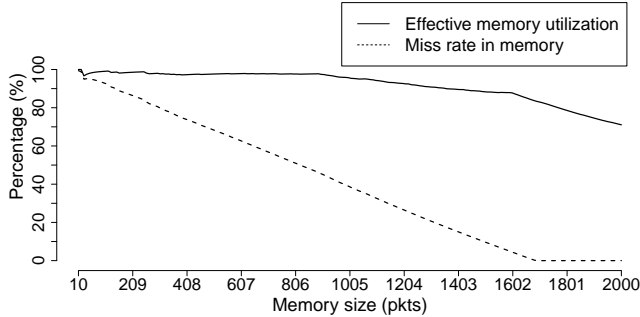
## 6.2 Two-Tier Packet Caching

RushNet’s two-tier packet caching aims to improve the overall packet recovery time. Packets cached in memory can be immediately transmitted without the overhead of flash reads over the SPI bus. Since the caching strategy performs loss retrodiction from tail RSSI readings to infer the packet SINR, this section first determines the optimal threshold. Then, we discuss the impact of available memory to caching performance.

Our experiment setup consisted of five nodes: three receivers, and two transmitters for the two traffic classes. Transmitters did not exercise CSMA, and they were 50 cm away from receivers. The LP transmitter sent one packet every 8 ms, while the HP transmitter randomly sends a preemptive train with a length of four at an average interval of 10 seconds. Without the HP transmitter, all three receivers observed near-perfect PRR from the LP transmitter.

Figure 11 illustrates the impacts of the different tail RSSI thresholds, in terms of false negatives and false positives. Intuitively, if the threshold is too high, nodes effectively underestimate the background interference. The figure shows that false retrodiction on successful reception increases as the tail RSSI threshold increases beyond  $-60$  dBm. Interestingly, the figure also shows that false retrodiction on loss decreases as the threshold increases beyond the same point. Therefore, our deployments adopt this threshold for the retrodiction algorithm.

Next, we evaluate the impact of available memory with trace-driven simulations on the data collected in the previous experiment. We envision RushNet running on a wide range of wireless platforms, from resource-rich 802.11 plat-



**Figure 12. Trace-driven simulations suggest high effective memory utilization during the recovery phase.**

forms to resource-constrained 802.15.4 platforms. We make two observations from Figure 12. First, the effective memory utilization is high, or packets cached in memory have a high probability of being requested during the recovery phase. These packets represent speedup opportunities where the sender does not need to issue flash reads over the SPI bus. Second, while increasing the available memory space decreases the cache miss rate, there is a diminishing return. Specifically, the effective memory utilization decreases as RushNet starts to include packets with any probability of being lost.

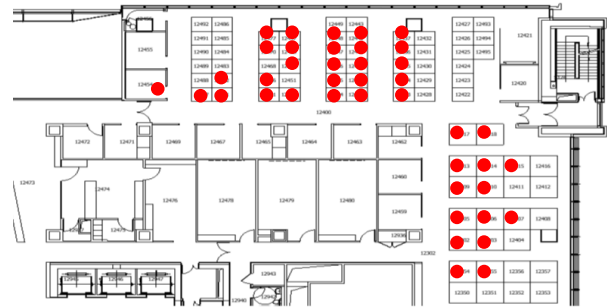
## 7 Deployment Results

RushNet has enabled several indoor applications: Asset tracking at production data centers, and occupant-driven comfort monitoring in offices. This section presents the application-level performance benchmarks from an office deployment.

### 7.1 Deployment Setup

Modern office buildings are heavily instrumented with centralized control systems such as HVAC (Heating, Ventilation, and Air Conditioning) and lighting. While these centralized systems simplify administrative tasks, they do not always provide optimal settings for occupants. In fact, the lack of visibility on the building environmental dynamics is one root cause. Specifically, most existing office sensing infrastructures have limitations, in terms of granularity and coverage. Our *Comfort Monitoring (ComMon)* system explores the use of crowdsourced sensing to monitor indoor environment dynamics such as temperature and humidity.

Our office deployment has two groups of sensing devices for two 802.15.4 radio chipsets: CC2420-based TelosB motes and RF230-based custom MeshID motes. Both groups occupy 802.15.4 channel 26 to minimize external interference on the 2.4 GHz spectrum. TelosB offers several desirable features. First, TelosB has all the sensors that we need: temperature, humidity and ambient light intensity. Second, TelosB can be powered through the built-in USB jack. Powered USB ports are ubiquitous in current office environments (e.g., computers, monitors, and AC-to-USB converters). Finally, TelosB includes a battery pack that can be a secondary power source when the USB port is powered off. We use results from the MeshID group to verify the generality of RushNet.



**Figure 13. Floor map labeling the deployment location of 41 nodes in the TelosB group.**

	Throughput (KBps)
<b>Baseline solution</b>	0.4
<b>RushNet</b>	1.9
<b>Theoretical upper bound</b>	2.3

**Table 1. LP traffic throughput from the office deployment.**

The deployment starts by distributing pre-programmed TelosB motes to participants, who then install motes at their preferred locations. We currently rely on participants to register the mote location on a web site.

ComMon uses RushNet to build a data collection tree and perform end-to-end data transport. The system treats periodic sensor readings as low-priority data. With four sensors and mote health counters, each sensor mote generates 52 bytes of data every 15 seconds. As the network scales up, this seemingly-infrequent sampling frequency can saturate the radio medium. There are two defined high-priority alert: the first is triggered by any sensing data above a predefined threshold, and the second is triggered by the USB port being powered off. The latter provides insights on how occupants power-cycle their equipment on a daily basis.

### 7.2 Application-Level Results

We now present results from the TelosB group of the deployment<sup>5</sup>. Figure 13 marks the location of all 41 nodes on the floor quadrant map. We conducted experiments and collected deployment statistics to evaluate two system metrics: *low-priority traffic throughput*, and *high-priority traffic delivery ratio*.

**LP Traffic Throughput.** We used the popular end-to-end streaming technique from Flush [11] and WRAP [17] as the comparison baseline. Since RushNet limits the number of concurrent transmitters with the token-passing mechanism, the network performance is essentially the sum of all branches. Therefore, we focus on a five-hop branch with relatively low packet reception ratios ( $\sim 90\%$ ). The leaf node sent 4,000 packets. To minimize interference among concurrent packets, the baseline solution estimated its own transmission interval to ensure the current packet leaves a radio interference range before sending the next one. This interval

<sup>5</sup>The co-locating MeshID deployment group gives similar observations.

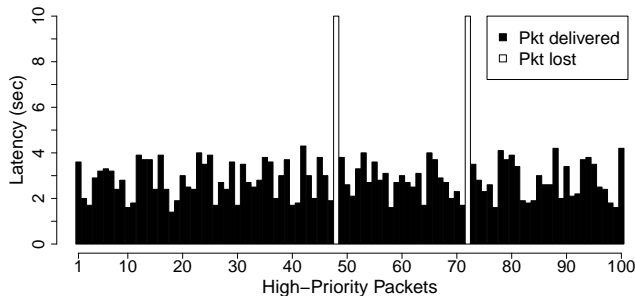


Figure 14. HP packet delivery latency in the deployment.

estimation is 30 ms. On the other hand, without this concern of interference, RushNet adopted a smaller interval of 10 ms.

Table 1 shows that end-to-end throughput of RushNet is close to the theoretical upper bound. Our theoretical upper bound assumes a five-hop path (with a two-hop radio interference range), 250 kbit/s 802.15.4 links, and 4MHz SPI bus. Statistics reveal that about 15% of the traffic are packet retransmissions. On the other hand, the baseline solution observed a slow down of a factor of 4.75, which is partially due to the inter-packet delay of 30 ms.

**HP Traffic Delivery Ratio.** We compared the number of alerts generated in the network, and received by the gateway. We randomly generated 100 alerts over a course of 10 minutes on the same branch above. Figure 14 shows that 98% of the alerts were delivered with a latency of less than four seconds. This is within the requirement from our building administrators. In contrast, since a node holds the token once every 20 seconds, a local packet queue scheduling approach would observe a latency of at least 20 seconds, or an increase by a factor of five. Finally, without any recovery mechanism, the impact on the branch’s LP traffic PRR is only 2%.

### 7.3 Delivery Timing Analysis

The HP delivery latency depends on both hardware choices and software implementation. To guide the design of future systems with preemption in the air, we provide the timing analysis as a max-size 802.15.4 packet travels in our TelosB group deployment.

**Single-Hop Analysis.** We formally define the HP transmission path as the time between the application layer generates an HP packet and the last bit of the packet is in the air. Then, the reception path is the opposite flow. Our measurements show that the transmission and reception paths for an HP train of one packet take about 10.358 ms and 14.124 ms, respectively. As we explain below, most delays are due to the I/Os.

First, most sensor platforms have a separate MCU and radio chip, which requires a bus for data exchange. The bus type and speed play an important role. In our case of 2 MHz SPI bus, the delay is about 5.576 ms, or half the transmission path time. Second, the physical radio type and speed are also important. As the 802.15.4 radio is restricted to 250 Kbps at 2.4 GHz, transmitting a max-sized packet of 128 bytes takes about 3.969 ms. We note that this number increases with the HP train size.

In fact, the HP packet scheduling and processing delay are relatively negligible. Nodes maintain a separate packet queue for HP packets, which simplifies the check for pending HP packets. In addition, we disable unnecessary radio features such as Clear Channel Assessment (CCA) and encryption.

**Multi-Hop Analysis.** Multi-hop forwarding incurs additional steps, such as payload processing, next-hop lookup and queuing operations. Our measurements show that the forwarding delay between the end of a reception path and the beginning of a transmission path is about 485.541 ms. The result is a trade-off between programming modularity and efficiency. TinyOS introduces modularity through the notion of software components and tasks; the former divides an application into individual pieces, while the latter breaks down a long execution path into independent pieces (with state variables). An optimized code with minimal abstractions and redirections can reduce multi-hop delay.

## 8 Related Work

### 8.1 Traffic Prioritization

Most previous efforts reframe the traffic prioritization problem as a scheduling problem. One can divide these works according to their level of service guaranteed.

One category of work centers around flow-based resource reservation. Both MRSVP [30] and HMRSVP [33] address the challenge of adapting Resource Reservation Protocol (RSVP) to wireless networks. INSIGNIA [14] piggybacks resource requests in the packet header to minimize the overhead due to control traffic. However, the drawback of resource reservation is the limited flexibility in dynamically changing the QoS requirement.

Multi-Path and Multi-SPEED Routing Protocol (MM-SPEED) [6] provides traffic prioritization in two metrics: reliability and timeliness. For the reliability metric, MM-SPEED nodes build a multi-path tree, and they forward data packets by picking the path that best satisfies the specified reliability requirement. For the timeliness metric, the MM-SPEED protocol assumes that all nodes in the network know their geo-locations, and nodes forward data packets by using geographic forwarding scheme on local neighborhood information. In addition, MMSPEED proposes dynamic compensation to minimize the impact on the end-to-end routing performance, in case of errors from local routing decisions. The requirement of localizing all nodes might not be feasible in many real-world deployments, especially with resource-constrained sensing nodes. Second, it is not clear how errors in local decisions would be amplified in long routing paths, especially as the sensing network scales up.

Sequential Assignment Routing (SAR) [29] offers a higher QoS guarantee by influencing the end-to-end routing decision, instead of simply being a MAC-layer solution. SAR builds multi-path trees with considerations to multiple QoS metrics: energy consumption and application-assigned data priority. However, the protocol incurs control overhead in maintaining and reacting to network dynamics. In fact, this overhead grows with the network size.

TEEN [20] allocates dedicated TDMA time slots to each

node for the purpose of transmitting high-priority data. The number of these high-priority TDMA slots allocated depends on the complete knowledge of network traffic pattern. However, in many sensing networks, high-priority data packets (e.g., alerts and alarms) can be generated in unpredictable time intervals.

Prioritized Heterogeneous Traffic-oriented Congestion Control Protocol (PHTCCP) [21] assumes a tree topology, and it implements hop-by-hop explicit rate control triggered by link congestion detection, which are based on the rate at which a node's children are uploading packets of different priorities. PHTCCP is not suitable for sensing networks where the occurrence of high-priority traffic is sporadic and unpredictable, as the timely delivery of a high-priority packet might not justify the overhead in on-demand detections and adjustments. In addition, the on-demand detections and adjustments add additional delays before the high-priority packet can be transmitted.

Wu et al. [35] propose an algorithm for traffic-aware channel assignment, with the assumption that current and future traffic patterns are known.

3-Hop-Ahead [9] uses a contention-based MAC for regular data collection traffic, but reserves the path if high-priority traffic is generated. Since 3-Hop-Ahead reserves path on demand, there is an additional setup delay before the high-priority packets can be transmitted. In addition, the presence of high-priority packets suppresses other concurrent transfers within the 3-hop range to reduce interference, but the radio medium utilization over time might be reduced due to idle gaps on the medium.

## 8.2 Leveraging Radio Capture Effect

After radio capture was first studied in the context of FM radio, the wireless community has used capture to improve throughput and lower latency in Aloha [37] and 802.11 networks [10].

One of the first works on capture effect in the WSN community is [34], where Whitehouse et al. leverage capture to differentiate packet collisions and losses. The technique identifies whether any synchronization header is embedded in the expected payload. If so, then the packet collision is assumed to have occurred.

Lu et al. propose the Flash flooding protocol that improves the network-wide dissemination performance [19]. Flash flooding makes the observation that all payload in a flooding session are essentially the same. Therefore, it allows concurrent transmission to occur in a radio coverage area, and have the receiver capture one of the signals.

Most low-power radio capture characterizations are performed on radios with software stack, such as TI CC1000. Priyantha et al. presented results from experiments on the Cricket platform [25]. However, packet-based radios such as CC2420 have a different behavior [1]: if the colliding packet appears after the preambles of another concurrent packet transmission, then the receiver does not receive the colliding packet irrespective of the transmission output power. Our work delves into the capture effect characterization and proposes novel techniques to encourage capture.

Flashback [5] uses short high-powered transmissions (or flashes) to send high-priority control messages concurrently

with other data transmissions. Unlike RushNet, Flashback uses the location of the bits corrupted by flashes as a way to modulate bits representing the high-priority data. This approach is difficult to do on off-the-shelf low-power radios (e.g., 802.15.4), which do not expose low-level radio information.

## 8.3 Bulk Data Transport

Ahn et al. proposed Funneling-MAC that uses a hybrid (schedule-based) TDMA and (contention-based) CSMA/CA scheme to address the funneling effect in a data collection tree [3]. Such funneling effect is caused by multiple tree nodes trying to upload data to the sink at the same time, and RushNet's WRAP-based [17] approach for LP traffic does not have this problem.

Osterlind et al. identified slow SPI bus access as the bottleneck on the TelosB platform [22], and proposed a preloading technique to overlap the PHY radio transmission and SPI data transfer. Flush [11] looks at the interference problem that arises from the end-to-end multi-hop forwarding. Specifically, the source node should wait until the current packet being forwarded leaves its radio interference range.

Hop [16] advocates hop-by-hop data block transfer, rather than end-to-end packet transfer. Specifically, at each iteration, the network focuses on pushing data one hop closer to the destination. Hop provides speed improvement over Flush because it can eliminate several packet transmission delays. RushNet differs from Hop in that it addresses challenges that resource-constrained platforms bring to hop-by-hop block transfer.

Like WRAP [17], RushNet provides congestion avoidance through a token-passing mechanism. However, RushNet provides further optimizations to data transport, such as block transfer, packet loss retrodiction, and traffic prioritization.

## 9 Discussion

### 9.1 Applicability to Other Radio Setups

An interesting discussion is the applicability of our results obtained from 802.15.4 radios to other radios such as 802.11. Our experiments used two popular off-the-shelf 802.15.4 radios: TI CC2420 and Atmel RF230. Literature suggests that 802.11 radios observe similar preamble search behavior as 802.15.4 radios. Kochut et al. report that capture can occur on 802.11b radios if the high-RSS packet arrives within the preamble time of the low-RSS packet [12]. Lee et al. report a similar result for 802.11a radios [13]. In addition, based on the fact that 802.11 radios use a different modulation scheme and bit rate for the packet header and payload, we can infer that the radios implement a state machine. This suggests that our techniques should be able to encourage radio capture as well.

A related discussion is the applicability to duty-cycling radio setups. In theory, this is possible because RushNet nodes can still send packets with different transmission power. However, a change is needed to ensure receivers stay up for at least the train length.

### 9.2 Extending to Multiple Traffic Classes

Our current effort with RushNet focus on the problem of supporting two traffic classes: high-priority and low-priority.

This support can satisfy most sensing deployments (e.g., monitoring networks that perform periodic data collection and generate alerts). As an extension to RushNet, we also look at support for multiple traffic classes. To this end, we are experimenting with approaches that use queuing and radio timing techniques. In addition, we will use our experience with real-world deployments to drive and test the system design.

## 10 Conclusion

RushNet offers practical traffic differentiation in saturated wireless sensor networks. To achieve coordination-free and schedule-free, we implement "preemption in the air" with both transmission power difference and radio capture effect. The advantage is that a node can transmit high-priority packets at any time. Evaluation results show that RushNet is able to deliver low-priority traffic with a throughput above existing data collection protocols. At the same time, RushNet reduces the high-priority packets delivery latency by a factor of five.

## 11 Acknowledgments

We would like to thank the anonymous reviewers for constructive reviews and Prof. Bhaskar Krishnamachari for shepherding this paper.

## 12 References

- [1] A Measurement Study of Low-Power Wireless. <http://www.stanford.edu/class/cs244e/papers/15-4-study.pdf>.
- [2] TinyOS. <http://www.tinyos.net>.
- [3] G.-S. Ahn, E. Miluzzo, A. T. Campbell, S. G. Hong, and F. Cuomo. Funneling-MAC: A Localized, Sink-Oriented MAC For Boosting Fidelity in Sensor Networks. In *Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2006.
- [4] Atmel, inc. AT86RF231 Product Page. <http://www.atmel.com/devices/at86rf231.aspx>.
- [5] A. Cidon, K. Nagaraj, S. Katti, and P. Viswanath. Flashback: Decoupled Lightweight Wireless Control. In *Special Interest Group on Data Communication (SIGCOMM)*. ACM, 2012.
- [6] E. Felemban, C.-G. Lee, and E. Ekici. MMSPEED: Multipath Multi-SPEED Protocol for QoS Guarantee of Reliability and Timeliness in Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, 2006.
- [7] G. Hackmann, F. Sun, N. Castaneda, C. Lu, and S. Dykec. A Holistic Approach to Decentralized Structural Damage Localization Using Wireless Sensor Networks. In *RTSS*, 2008.
- [8] IEEE. 802.15.4 MAC and PHY. <http://standards.ieee.org/getieee802/download/802.15.4-2003.pdf>.
- [9] S. ik Cha, J. H. Ko, S. Kwon, and C. Kim. 3-Hop-Ahead Path Reservation Scheme for Expedite Traffic in Wireless Sensor Networks. In *International Conference on Information Networking (ICOIN)*, 2009.
- [10] G. Kai, H. Honglin, W. Guo, and Z. Jinkang. A capture-based MAC Protocol and Its Performance Analysis. In *VTC-Fall*, 2003.
- [11] S. Kim, R. Fonseca, P. Dutta, A. Tavakoli, D. Culler, P. Levis, S. Shenker, and I. Stoica. Flush: A Reliable Bulk Transport Protocol for Multihop Wireless Networks. In *Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2007.
- [12] A. Kochut, A. Vasani, A. U. Shankar, and A. Agrawala. Sniffing Out the Correct Physical Layer Capture Model in 802.11b. In *International Conference on Network Protocols (ICNP)*. IEEE, 2004.
- [13] J. Lee, W. Kim, S.-J. Lee, D. Jo, J. Ryu, T. Kwon, and Y. Choi. An Experimental Study on the Capture Effect in 802.11a Networks. In *International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WinTECH)*. ACM, 2007.
- [14] S.-B. Lee and A. Campbell. INSIGNIA: In-band Signaling Support for QoS in Mobile Ad-Hoc Networks. In *International Workshop on Mobile Multimedia Communication (MoMuC)*, 1998.
- [15] K. Leentvaar and J. H. Flint. The Capture Effect in FM Receivers. *Communications, IEEE Transactions on*, 1976.
- [16] M. Li, D. Agrawal, D. Ganesan, and A. Venkataramani. Block-switched Networks: A New Paradigm for Wireless Transport. In *Symposium on Networked Systems Design and Implementation (NSDI)*. USENIX, 2009.
- [17] C.-J. M. Liang, J. Liu, L. Luo, A. Terzis, and F. Zhao. RACNet: A High-Fidelity Data Center Sensing Network. In *Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2009.
- [18] C.-J. M. Liang, N. B. P. J. Liu, and A. Terzis. Surviving Wi-Fi Interference in Low Power ZigBee Networks. In *Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2010.
- [19] J. Lu and K. Whitehouse. Flash flooding: Exploiting the capture effect for rapid flooding in wireless sensor networks. In *International Conference on Computer Communications (INFOCOM)*. IEEE, 2009.
- [20] A. Manjeshwar and D. P. Agrawal. TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. In *International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*. IEEE, 2001.
- [21] M. Monowar, M. O. Rahman, A.-S. K. Pathan, and C. S. Hong. Congestion Control Protocol for Wireless Sensor Networks Handling Prioritized Heterogeneous Traffic. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services (MobiQuitous)*, 2008.
- [22] F. Osterlind and A. Dunkels. Approaching the Maximum 802.15.4 Multihop Throughput. In *HotEmNets*, 2008.
- [23] L. Pan, H. Wu, and N.-F. Tzeng. An Efficient and Scalable Prioritized MAC Protocol (PMAC) for Backbone Communication in Wireless Sensor Networks. In *Conference on Sensor Technologies and Applications (SENSORCOMM)*, 2009.
- [24] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling Ultra-Low Power Wireless Research. In *International Conference on Information Processing in Sensor Networks (IPSN)*. ACM/IEEE, 2005.
- [25] N. B. Priyantha. *The Cricket Indoor Location System*. PhD thesis, Massachusetts Institute of Technology, 2005.
- [26] B. rong Chen, K.-K. Muniswamy-Reddy, and M. Welsh. Ad-Hoc Multicast Routing on Resource-Limited Sensor Nodes. In *International Workshop on Multi-hop Ad Hoc Networks: from Theory to Reality (REALMAN)*. ACM, 2006.
- [27] A. Saifullah, C. Wu, P. B. Tiwari, Y. Xu, Y. Fu, C. Lu, and Y. Chen. Near Optimal Rate Selection for Wireless Control Systems. In *Real-Time and Embedded Technology and Applications Symposium (RTAS)*. IEEE, 2012.
- [28] G. Smith, G. Venkatesan, E. Rouss, O. Aboul-Magd, A. Ashley, N. Kakani, and B. Hart. 802.11 QoS Tutorial. <http://www.ieee802.org/1/files/public/docs2008/avb-gs-802-11-qos-tutorial-1108.pdf>.
- [29] K. Sohrali, J. Gao, V. Ailawadhi, and G. Pottie. Protocols for Self-Organization of a Wireless Sensor Network. *IEEE Personal Communications*, 2000.
- [30] A. K. Talukdar, B. R. Badrinath, and A. Acharya. MRSVP: Resource Reservation Protocol for an Integrated Service Network With Mobile Hosts. *Wireless Network*, 2001.
- [31] Texas Instruments, inc. CC1000 Product Page. <http://www.ti.com/product/cc1000>.
- [32] Texas Instruments, inc. CC2420 Product Page. <http://www.ti.com/product/cc2420>.
- [33] C.-C. Tseng, G.-C. Lee, R.-S. Liu, and T.-P. Wang. HMRSVP: a hierarchical mobile RSVP protocol. *Wireless Networks*, 2003.
- [34] K. Whitehouse, A. Woo, X. Jiang, J. Polastre, and D. Culler. Exploiting the Capture Effect for Collision Detection and Recovery. In *Workshop on Embedded Networked Sensors (EmNets)*, 2005.
- [35] Y. Wu, M. Keally, G. Zhou, and W. Mao. Traffic-aware channel assignment in wireless sensor networks. In *Wireless Algorithms, Systems, and Applications (WASA)*, 2009.
- [36] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. A Wireless Sensor Network For Structural Monitoring. In *Conference on Embedded Networked Sensor Systems (SenSys)*. ACM, 2004.
- [37] W. Yue. The Effect of Capture on Performance of Multichannel Slotted ALOHA Systems. *Transactions on Communications*, 1991.