RACNet: A High-Fidelity Data Center Sensing Network

Chieh-Jan Mike Liang[†], Jie Liu[‡], Liqian Luo[⋆], Andreas Terzis[†], Feng Zhao[‡]

[†]Johns Hopkins University [‡]Microsoft Research [⋆]Google

{cliang4, terzis}@cs.jhu.edu {jie.liu, zhao}@microsoft.com liqianluo@gmail.com

Abstract

RACNet is a sensor network for monitoring a data center's environmental conditions. The high spatial and temporal fidelity measurements that RACNet provides can be used to improve the data center's safety and energy efficiency. RACNet overcomes the network's large scale and density and the data center's harsh RF environment to achieve data yields of 99% or higher over a wide range of network sizes and sampling frequencies. It does so through a novel *Wireless Reliable Acquisition Protocol* (WRAP). WRAP decouples topology control from data collection and implements a token passing mechanism to provide network-wide arbitration. This congestion avoidance philosophy is conceptually different from existing congestion control algorithms that retroactively respond to congestion. Furthermore, WRAP adaptively distributes nodes among multiple frequency channels to balance load and lower data latency. Results from two testbeds and an ongoing production data center deployment indicate that RACNet outperforms previous data collection systems, especially as network load increases.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems—*Distributed applications*; C.2.2 [**Computer-Communication Networks**]: Network Protocols—*Routing protocols*

General Terms

Design, Performance, Experimentation, Measurement

Keywords

Sensor Networks, Mote, Sensing, Architecture, Network Protocol, Congestion Avoidance Protocol, Data Center, Energy, Green

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. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

SenSys'09, November 4–6, 2009, Berkeley, CA, USA.

Copyright 2009 ACM 978-1-60558-748-6 ...$5.00

1 Introduction

Data center energy consumption has attracted global attention due to the fast growth of the IT industry and increasing concerns about carbon footprints and climate change. While advances in component design continue to decrease the power consumption of computer servers, one cannot overlook the energy consumed by the hosting facilities, considering that only 30% to 60% of the total energy that a typical data center consumes powers its IT equipment. The rest is either lost during the power delivery and conversion process, or used by environmental control systems such as Computer Room Air Conditioning (CRAC) units, water chillers, and (de)humidifiers [1, 35].

Lack of visibility into the data center's operating conditions is one of the root causes for this low energy efficiency. As conventional wisdom dictates that IT equipment needs abundant cooling to operate reliably, the CRAC systems in many data centers are set to very low temperatures. Furthermore, data center operators tend to further decrease the CRAC's temperature settings when servers issue thermal alarms because they lack the information to accurately diagnose the problem. Thereby, high-fidelity (i.e., with rack-level spatial granularity and sub-minute sampling rate) historical and real-time data about the environmental conditions inside a data center are invaluable not only for diagnosing problems but for improving the data center's efficiency [4, 24].

Traditional solutions for data center environmental monitoring use wired sensors [24, 27], but the high installation and configuration costs prevent the wide adoption of these systems. Using the motherboards' temperature sensors is also problematic, because these sensors reflect the servers' activities rather than the data center's environmental conditions, as the results from Section 2 show. On the other hand, wireless sensor networks are ideal for the data center monitoring task as they offer low-cost, non-intrusive, and flexible in-situ sensing.

At the same time, the application's requirements in terms of latency and reliability coupled with the data centers' environment pose unique challenges to wireless sensing. As our site survey results show (§ 2.3), even a single data center room requires hundreds of wireless sensors, 50% to 65% of which can interfere with each other. Uncoordinated wireless transmissions in this environment can lead to congestion collapse drastically reducing the network's usable capacity. Furthermore, packet losses are frequent, despite the high node density, due to interference from collocated networks (e.g., WiFi) and the large number of metallic obstacles. Previous sensor networks that did not implement end-to-end reliability exhibited data yields of 20-60% [9, 32, 38] and thus fail to meet the requirements of data center control and troubleshooting applications. More recent reliable collection protocols (e.g., [11, 22, 23, 37]) employ local data caching and end-to-end retransmissions to improve data yields but do not scale to the network sizes and densities required by data center sensing.

Contributions: This paper presents the design, implementation, and evaluation of *RACNet*, a large-scale sensor network for high-fidelity data center environmental monitoring. RACNet uses custom-made *Genomote* sensor nodes that employ a combination of wired and wireless communications to scale. As a key technical contribution of RACNet, we developed a Wireless Reliable Acquisition Protocol (*WRAP*) for scalable data collection (§3). To tackle the challenge of self interference caused by contention in dense wireless networks, WRAP follows a simple yet effective *congestion avoidance* philosophy, leveraging frequency and time multiplexing that is conceptually different from previous congestion control approaches (e.g. [11, 23, 26]). In particular, WRAP uses multiple IEEE 802.15.4 frequency channels simultaneously and adaptively balances the number of nodes on each channel based on traffic load. WRAP also implements coordinated data collection through a *token passing* protocol that provides an implicit network arbitration mechanism, allowing only one active packet flow per frequency channel. We note that WRAP is intrinsically different from congestion control protocols, such as RCRT [23] and IFRC [26]. While the later ones detect and react to congestion, WRAP proactively prevents the contention that generates congestion in the first place.

Using results from two testbeds (§4) and an ongoing deployment at a production data center (§5) we show that WRAP outperforms protocols that use rate-based congestion control (RCRT) and uncoordinated transmissions (CTP), especially at high network loads.

In the remainder of the paper we first present high-level requirements for data center monitoring and outline the challenges of using IEEE 802.15.4 wireless communications in these environments through a site survey in Section 2. Section 3 elaborates on the design of the RACNet reliable data collection system. We present our evaluation results in Section 4, while Section 5 outlines results from a production data center deployment of RACNet. Section 6 provides examples of how data collected from RACNet are used in improving data center operations. Section 7 reviews related work and we conclude in Section 8 with a summary and discussion about future work.

2 System Design Rationale

A data center monitoring and control system requires a low-cost data acquisition system that offers wide coverage and is easy to install and own. In this section, we motivate our choice of using wireless sensor networks for data center sensing and describe the challenges of reliable data collection in this environment.

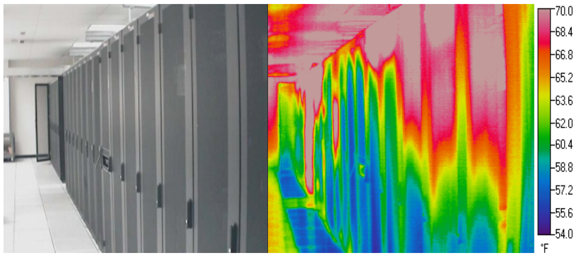

Figure 1. A row of computer racks inside a data center (left) and the corresponding infrared image representing the spatial temperature distribution (right).

2.1 Application Requirements

Thermal and air dynamics in data centers can be complex. Figure 1 allows us to gain an understanding of the underlying spatial variability through a thermal image captured by an infrared camera. This picture exposes the temperature variations that exist over the air intakes of multiple server racks. One can observe temperature differences larger than 10 °F across various heights of the same rack, as well as significant differences in the temperature distribution patterns across different racks.

In order to rapidly detect hot spots created by complex air and thermal dynamics, the system needs to provide sensing with high temporal as well as spatial fidelity. As an example of temporal variation speed, we recorded temperature changes of 10 °C within the five minutes immediately following server and CRAC actions. Based on these observations, we selected a 30-second sampling rate for RACNet, to promptly detect and mitigate abnormal thermal conditions. The sampling rate can be even higher when troubleshooting hot spots or when sampling different sensors (e.g., monitoring the server's power consumption).

In order to support effective cooling control and dynamic workload distribution, RACNet must provide data yields of 95%, or higher. Ideally, these measurements need to be collected before the next samples are generated. When this is not feasible, we still need to archive the data so they can be used for long-term decision making.

2.2 The Need for Wireless Sensors

There are seemingly several options for measuring the temperature and humidity distributions inside a data center. For one, thermal images such as the one shown in Figure 1 visualize temperature variations over the camera's view frame. However, continuously capturing thermal images throughout the data center is prohibitively expensive. Alternatively, modern servers have several onboard sensors that monitor the thermal conditions near key server components, such as the CPUs, disks, and I/O controllers. These sensors are used to detect and prevent hardware failures due to overheating rather than sense the data center's ambient environment. Some recent servers also have temperature sensors at the air intake, and administrators can estimate room conditions from these sensors. However, for servers that do not have sensors at the air intake, it is difficult to accurately estimate the room temperature and humidity from other onboard sensors. Figure 2 plots the temperature measured at vari-

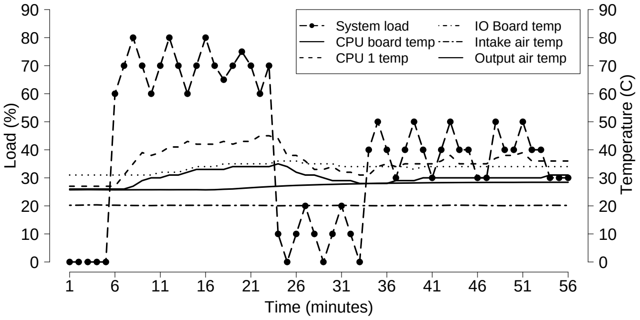

Figure 2. Temperature measured at different locations in and around an HP DL360 server. Also shown is the server's CPU load. Internal sensors reflect the server's workload instead of ambient conditions.

ous points along with the CPU utilization for an HP DL360 server with two CPUs. Air intake and output temperatures are measured with external sensors near the server's front grill and its back cover. It is evident from this figure that internal sensors are quickly affected by changes in the server's workload, rather than reflecting ambient conditions.

Intake air temperature (IAT) is important also because it can be used for auditing purposes. Server manufacturers and data center facility management contracts usually specify server operation conditions in terms of IAT. For example, the HP ProLiant DL360 (G3) servers require IAT to range from 50° to 95°F (10° to 35°C). It is therefore necessary to place external sensors at regular intervals across the servers' air intake grills to monitor IAT.

The communication mechanism used to retrieve the collected measurements is the other crucial aspect in the system design. Options in this case are divided in two categories: in-band vs. out-of-band. In-band data collection routes measurements through the server's operating system (OS) to the data center's (wired) IP network. The advantage of this approach is that the network infrastructure is (in theory) available and the only additional hardware necessary are relatively inexpensive USB-based sensors. However, data center networks are in reality complex and fragile. They can be divided into several independent domains not connected by gateways. Traversing across network boundaries can lead to serious security violations. Finally, the in-band approach requires the host OS to be always on to perform continuous monitoring. Doing so however would prevent turning off unused servers to save energy.

Out-of-band solutions use separate devices to perform the measurements and a separate network to collect them. Self contained devices provide higher flexibility in terms of sensor placement, while a separate network does not interfere with data center operations. However, deploying a wired network connecting each sensing point is undesirable as it would add thousands of network endpoints and miles of cables to an already cramped data center.

For this reason, wireless networks are the only feasible option. Moreover, networks based on IEEE 802.15.4 radios [10] (or 15.4 for short) are more attractive compared to Bluetooth or WiFi radios. The key advantage is that a 15.4 network has a simpler network stack compared to alternative solutions. This simplicity has multiple implications. First, sensing devices need only a low-end MCU such as the MSP430 [34] thus reducing the total cost of ownership and implementation complexity. Second, the combination of low-power 15.4 radios and low-power MCUs leads to lower overall power consumption. The need for low-power consumption will become apparent when we present the mechanism used to power multiple sensing devices from the same power source.

At the same time, there are significant challenges when using 15.4 networks for data center sensing, due to low data throughput and high packet loss rate. The maximum transmission rate of a 15.4 link is 250 Kbps while effective data rates are usually much lower due to MAC overhead and multi-hop forwarding. Furthermore, the lower transmission power[1] can lead to high bit error rates especially in RF-challenging environments such as data centers. To quantitatively understand these challenges, we survey the RF environment in a data center.

2.3 Data Center RF Environment

Data centers present a radio environment different from the ones that previous sensor network deployments faced. This is intuitively true as metals are the dominant materials in a data center. In addition to switches, servers, racks, and cables, other metallic obstacles include cooling ducts, power distribution systems, and cable rails. Given this departure from RF environments studied in the past (e.g., [28, 42]), characterizing this environment is crucial to understanding the challenges it poses to reliable data collection protocols.

For this reason we performed a site survey by uniformly distributing 52 Genomotes (§2.4) in an production data center spanning an area of approximately 12,000 sq-ft. The motes were placed on the top of the racks, following a regular grid pattern with adjacent nodes approximately 35 feet from each other. During the experiment, all nodes took turns broadcasting 1,000 128-byte packets with an inter-packet interval of 50 ms. All nodes used the 802.15.4 frequency channel 26 and transmitted their packets without performing any link-layer backoffs. Upon receiving a packet, each receiver logged the Received Signal Strength Indication (RSSI), the Link Quality Indicator (LQI), the packet sequence number, and whether the packet passed the CRC check.

We summarize the results from this survey below:

Neighborhood Size. We found that *on average 50% of all the nodes are within a node's communication range* and that a node's neighborhood can include as many as 65% of the network's nodes. Moreover, the neighborhood size in the production deployments will be significantly higher as they consist of hundreds of nodes deployed over the same space. It is thereby imperative to devise mechanisms that minimize packet losses due to contention and interference.

Packet Loss Rate. Figure 3 illustrates the distribution of packet reception ratios (PRR) over all the network links. While the majority of the links have low loss rate (i.e., $< 10\%$), *a significant percentage of links experience high*

[1] The TI CC2420 802.15.4 radio we use, transmits at 0 dBm, or 1 mW [33].

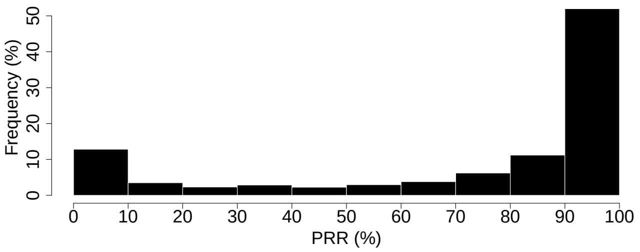

Figure 3. Distribution of packet reception ratios (PRR) across all the links from a 52-node data center site survey. A large percentage of the network's links exhibit non-trivial loss rates.

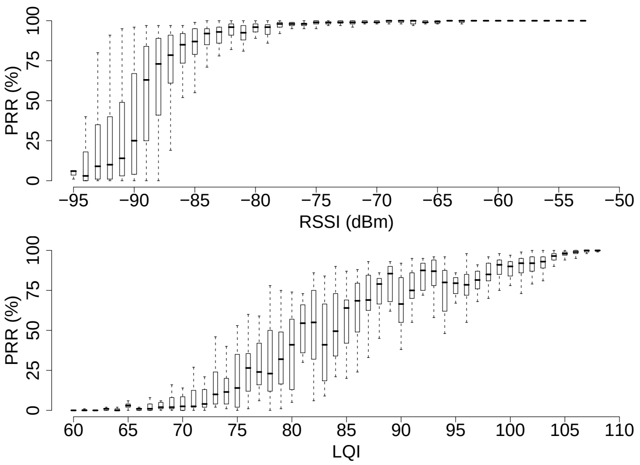

Figure 4. Boxplots of link PRR as a function of RSSI and LQI values. Boxplots show the sample minimum, first quantile, median, third quantile, and sample maximum. Links with RSSI > -75 dBm and LQI > 90 have persistently low PRR.

number of losses. This observation suggests that even in dense networks data collection protocols must discover high-quality links and avoid low quality links in order to build end-to-end paths with low loss rates.

Link Qualities. Both RSSI and LQI measurements have been used to estimate link qualities [29, 36]. RSSI measures the signal power for received packets, while LQI is related to the chip error rate over the packet's first eight symbols (802.15.4 radios use a Direct Sequence Spread Spectrum encoding scheme). Indeed, the results shown in Figure 4 indicate that *there is a strong correlation between RSSI/LQI and packet reception rates.* Based on these results, one can use an RSSI threshold of -75 dBm to filter out potential weak links. Selecting this conservative threshold removes a large number of links. Fortunately, the network remains connected because each node has many neighbors with high RSSI links.

Background RF Interference. Figure 5 shows the background noise distribution measured on each of the sixteen 802.15.4 frequency channels available on the 2.4 GHz ISM band. The measurements were collected by a mote that sampled its RSSI register at a frequency of 1 KHz while no other 802.15.4 radios were active. A total of 60,000 samples were collected on each channel. Because the data center in which the measurements were taken has considerable levels of 802.11 traffic, *802.15.4 channels that overlap with 802.11 channels experienced higher noise levels.* On the other hand, 15.4 channels 15, 20, 25, and 26 are relatively quiet. This motivates us to takes advantage of all the quiet channels simultaneously by dynamically partitioning the network's nodes over multiple collection trees, each operating at a different frequency channel.

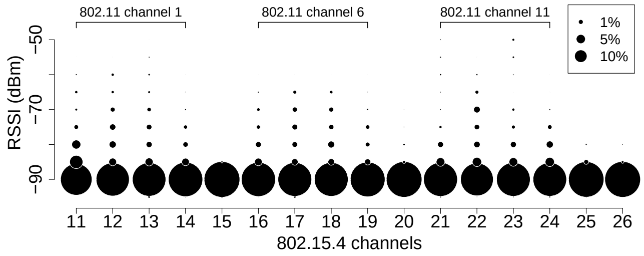

Figure 5. Background noise distribution across all 802.15.4 frequency channels in the 2.4 GHz ISM band. Each of the circumferences is proportional to the occurrence frequency of the corresponding RSSI level. Channels 15, 20, 25, and 26 are relatively quiet compared to other channels.

2.4 Genomotes

Some of the aforementioned challenges are partially addressed by RACNet's hardware design. Figure 6 presents a pair of Genomotes, which are sensor devices specifically designed for RACNet [17].

The wireless master node (shown on the left) and several wired sensors (one example shown on the right) form a (wired) daisy chain to cover one side of a rack, collecting data at different heights. This design increases sensing coverage and reduces the number of contending radios in the same space, without sacrificing deployment flexibility. However, even with the chain design, there are easily several hundred wireless master nodes within a data center colocation facility. For the remainder of this paper, we only consider the network among the wireless master nodes, treating the whole chain as a single node with multiple sensors.

The master node also has a flash memory chip that caches data locally to mitigate temporary connectivity variations. The whole chain is powered by a USB port connected to a server or a wall charger. Using a USB connection to power the whole mote chain means that unlike many previous sensor networks, power is not a critical concern in RACNet. At the same time, the maximum current that one can draw from a USB port by a foreign device is 100 mA. This limitation means that it would impossible to use a server's USB port to power multiple (or even a single) WiFi-based sensing devices. Finally, we note that using the same USB port to carry measurements is not an option because it requires the installation of additional software on the servers – something that is not administratively possible in our environment.

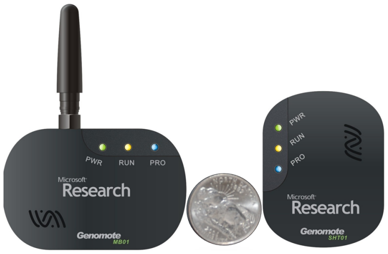

Figure 6. Two types of Genomotes designed for RACNet. The wireless node (on the left) controls several wired nodes (on the right) to reduce the number of wireless sensors within the same broadcast domain.

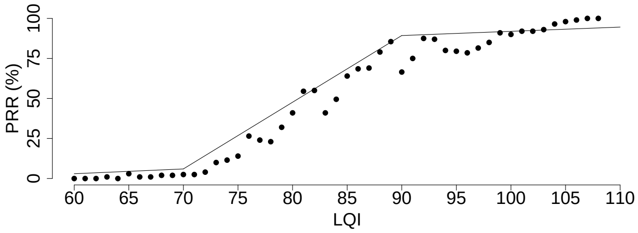

Figure 7. Piece-wise linear approximation of PRR from LQI. The dots are the average PRR for each LQI obtained from the site survey results.

3 Wireless Reliable Acquisition Protocol

The *Wireless Reliable Acquisition Protocol* (WRAP) lies at the center of RACNet. Like many data collection protocols, WRAP has a network layer that controls the topology and a transport layer for data retrieval. Nevertheless, WRAP is unique in the way it combines centralized and distributed decision making to achieve scalability and responsiveness. Specifically, the network layer (§3.2) constructs collection trees across multiple channels in a distributed way. On the other hand, the transport layer (§3.4) relies on a centralized token passing mechanism to prevent network congestion and reliably retrieve data from each of the network's nodes. Note that WRAP takes advantage of the energy supply from server USB ports, and it does not currently exercise duty-cycling.

3.1 Protocol Design Overview

From an architectural perspective, WRAP's design is at the center of the spectrum between distributed and centralized data collection. At one end of this spectrum, the nodes participating in a distributed data collection protocol collaborate to construct a common routing tree and independently forward data as soon as possible [8, 39]. The derived network topology can quickly adapt to link quality changes or node failures. However, the lack of coordination can lead to channel contention and eventually packet losses especially under high network load. At the other end of the design space lies the centralized approach, in which the gateway controls the operation of the entire network leveraging its ample computational resources and complete knowledge of the network topology [22, 30]. Nodes simply report their local channel conditions to the gateway which in turn determines the routing paths and orchestrates data downloads. While centralized approaches achieve high reliability and manageability, the control traffic to and from the gateway can add significant overhead. For example, in order to compensate for link and node failures, neighborhood information must be collected frequently. However, such information scales with the number of network links, which for dense networks can grow with the square of the number of nodes.

WRAP follows a hybrid approach whereby nodes determine the routing topology in a distributed way while the gateway coordinates data transport using a centralized token passing mechanism. Specifically, the gateway periodically generates a token that traverses the derived routing tree in a depth-first manner. Only the tree node that holds the token can transmit one or more packet before passing the token to the next node. By allowing only a single node to transmit at any point in time, token passing bypasses the inter-flow contention that can lead to congestion and packet loss. This is especially important close to the root of a dense network whereby concurrent flows are very likely to interfere with each other. In this respect WRAP is a congestion avoidance mechanism, unlike existing centralized [23] or distributed [26] congestion control protocols. Moreover, by eliminating congestion as a possible cause of packet loss, WRAP removes the ambiguity that complicates the response of congestion control protocols to missing data.

This division of responsibilities ensures timely adaptation to link quality variations and at the same time gives every network node a fair share of the network's resources without contention. RCRT is another example of a hybrid protocol that adds centralized coordination –the rate control information– to an otherwise distributed protocol [23]. However, imposing a single transmission rate for the entire network is inevitably biased towards the weakest node (i.e. the one behind the most lossy link) and artificially degrades the overall network throughput.

3.2 Topology Control

The network layer maintains robust data collection trees rooted at the network's gateways. The mechanism's distributed nature allows nodes to independently react to topology changes including degraded link qualities and node failures. Since WRAP also uses the tree to deliver downstream traffic such as requests for lost data packets, we focus on building bi-directional trees (BiTrees) with high quality bi-directional links.

3.2.1 Parent Selection

Gateways initiate BiTree construction by broadcasting HEARTBEAT messages. HEARTBEATs include fields that represent the node's status, including its hop distance from the root, its parent node ID, the number of children, and the path quality metric to the root. We describe the importance of communicating the number of children in Section 3.2.2.

Upon receiving a HEARTBEAT message, a node takes the following steps: first, the incoming HEARTBEAT message needs to have a RSSI above a threshold to avoid links with high loss rates (§2.3). Next, the node checks whether the potential parent has already reached its maximum number of children. If not, the next step is to evaluate the *path* quality to the gateway via this potential parent by computing the *path expected transmission count* (PETX) as follows:

$$PETX_j = \sum_{l \in P} \frac{1}{PRR_l} = PETX_i + \frac{1}{PRR_{i,j}}$$

where j is the current node, i is its potential parent, and P is the path from j to the gateway via i.

To compute $PETX_j$ recursively, the $PETX_i$ is included in the HEARTBEAT messages. However, estimating $PRR_{i,j}$ directly from HEARTBEATs would require multiple message rounds. Instead, we take advantage of the Link Quality Indicator (LQI) available from radio chips such as the TI CC2420 [33], to reduce control message overhead. Specifically, we use the piece-wise linear approximation shown Figure 7 to estimate a link's PRR based on its LQI. We note that while the curve shown in Figure 7 was derived from the site survey data, it resembles the approximation used in [3].

The node selects the upstream neighbor with the smallest PETX as its *potential* parent and initiates a TREE_JOIN request. The parent also estimates the link quality from this potential child in the upstream direction before replying with a GRANT message. Otherwise, the TREE_JOIN operation times out. This two-way handshake has two benefits. First, it serves as an explicit agreement between the parent and the child node that both have the resources to relay messages for each other. Second, since we require a BiTree for data downloading, it is important to ensure the link quality in both directions, as wireless links can be asymmetric [42].

3.2.2 *Coordinated Beaconing*

As described above, nodes broadcast HEARTBEAT messages to construct and maintain BiTrees. It is therefore desirable to transmit multiple HEARTBEATs in a short amount of time, to accelerate the tree construction process. However, in large and dense networks, this can lead to broadcast storms and severe collisions, eventually affecting the quality and stability of the resulting tree.

A simple and low-maintenance solution would be to adopt a contention-based approach, in which nodes contend for the radio medium. However, this approach is ill-suited for dense networks because the large number of HEARTBEATs is likely to cause collisions and large delays. At the same time, a TDMA-based protocol that assigns exclusive time slots to each node within the same interference range is cumbersome as it requires tight time synchronization and additional control traffic to set up the schedule.

Instead, WRAP uses a reduced contention mechanism to regulate the broadcast of HEARTBEAT messages. Specifically, WRAP defines a time slot of length T that starts immediately after a node P broadcasts its HEARTBEAT message. The time slot is further divided into two uneven sections according to the number of children that P already has and the number of additional children that P can support. The first section is reserved for the HEARTBEAT messages sent by P's children, while the second is used by nodes that are not part of the tree to initiate the handshake process with P. Nodes that receive P's HEARTBEAT randomly select a time within the appropriate section to transmit their message.

While this mechanism reduces contention, it does not guarantee a collision-free network. Specifically, we do not coordinate among nodes within the same broadcast domain that connect to different parents. Instead, we let them contend for the medium.

3.3 Channel Diversity

A RACNet system may consist of many hundreds of nodes within one data center. One way to increase data throughput and reduce data latency is by using multiple gateways. To do so, we take advantage of channel diversity to build multiple BiTrees rooted at different gateways, each on a different channel frequency. Previous work has shown that simultaneous communications over two-channel-apart 802.15.4 channels do not interfere with each other [40]. This section addresses the challenges of building multi-hop BiTrees over multiple channels in a distributed way.

3.3.1 *Construction of Multiple BiTrees*

Every RACNet gateway has a fixed channel assigned by the operator. Non-gateway nodes start by scanning channels sequentially and looking for a tree to join. Since gateways continuously perform data collection, a node can first overhear the network traffic and decide whether the channel potentially has a tree that it can join. In addition, a node can bound its wait time on each channel to (little over) one HEARTBEAT time interval because gateways periodically initiate new rounds of HEARTBEAT beaconing. A node joins the first tree using the two-way handshake mechanism described above. However, the node joins any subsequent trees only if the estimated quality of the new path is better than the one on the current tree.

WRAP follows a transaction model when constructing BiTrees across different channels. It is possible that a node (temporarily) joins multiple trees as it actively scans all available channels. However, nodes in this state do not broadcast HEARTBEAT messages to recruit children. This is to limit further disturbance in the candidate trees that the node later decides not to join. When the scanning phase ends, the node switches to and stays in the last tree it joined. Finally, a node's parent takes its HEARTBEAT transmissions as an indication of its commitment to the tree. Other candidate parents eventually time out and remove the node from their children lists.

Nodes can significantly reduce their channel scanning time with the gateways' help. Specifically, gateways maintain the list of all channels they collectively occupy and include this information in their HEARTBEAT messages. Therefore, after receiving one HEARTBEAT message, nodes immediately know all available channels.

3.3.2 *Balancing Multiple BiTrees*

As nodes join and leave the network or link qualities change, the sizes of different BiTrees can become unbalanced. We quantify the size of a tree by its sum of hops Δ, or the total path length from each node in the tree to the root. As we will show in Section 4.1, Δ largely determines the overall time necessary to finish a data collection round. For this reason WRAP uses Δ to balance the load among all the network's trees.

WRAP implements a distributed algorithm for balancing BiTrees. WRAP periodically checks the Δ's of different trees, and it initiates the channel-balancing process by sending a START_BAL message that propagates through the tree with the largest Δ. WRAP utilizes two mechanisms to avoid network instability: (i) it restricts the channel-balancing process to the gateway with the largest Δ, and (ii) it tolerates certain amount of imbalance in Δ. Let Δ_{avg} be the average among all trees. A gateway b^* starts the channel-balancing process only under the following conditions:

$$\Delta_{b^*} - \Delta_{avg} > \delta, \text{and}$$
$$b^* = argmax_{b \in B}(\Delta_b)$$

where B is the set of all gateways and δ is a threshold parameter that controls the amount of tolerable imbalance.

The START_BAL message contains the probabilities for switching to each of the other channels. Switching probabilities are defined to be higher for more under-utilized channels.

Specifically, a node connected to the tree rooted at b^* will decide to switch out with probability $P_{out} = \frac{\Delta_{b^*} - \Delta_{avg}}{\Delta_{b^*}}$. Once the node decides to leave b^*'s tree, the probability that it switches to another tree $B_i \neq b^*$ is set as follows:

$$P_i = 0, \text{ if } \Delta_i \geq \Delta_{avg}$$
$$P_i = \frac{\Delta_{avg} - \Delta_i}{\sum_{b \in B \text{ and } \Delta_b < \Delta_{avg}} (\Delta_{avg} - \Delta_b)}, \text{ if } \Delta_i < \Delta_{avg}$$

Intuitively, we attempt to migrate the extra nodes from gateway b^* to underloaded gateways, based on their degrees of under-utilization. In other words, more nodes will attempt to join the tree with fewer nodes. Finally, if the node can not find a parent in the target channel, it returns to its original channel.

3.4 Data Collection

The transport layer reliably collects data to RACNet gateways along the network's BiTrees. Rather than having nodes initiate data uploads asynchronously, WRAP coordinates the network traffic to reduce radio contention. At the same time, pull-based approaches in which gateways initiate data collection by sending requests to individual network nodes can incur significant overhead including the cost of one downstream message per node and the round-trip delay for transmitting each node's measurements. WRAP addresses these two sources of overhead by adopting a token passing approach.

3.4.1 Token Passing

The token passing mechanism does not require the gateways to have a priori knowledge of the network topology. Rather, it relies on the network to determine the next node that should hold the token. Since gateways continuously retrieve data from the network, this property also removes the overhead of having a separate phase for collecting neighborhood information from all nodes in the network. The basic protocol works as follows.

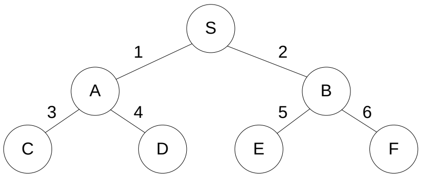

Figure 8. An sample two-level binary tree.

Gateways initiate a data collection round by passing the token to the first node on their list of children (§3.2.2). Each token contains an unique 32-bit token ID so that nodes know when a new round of data collection has started. WRAP tokens traverse the tree in a depth-first order. After receiving the token, the node passes it sequentially to all of its children in the tree. Once all of its children have finished transmitting their measurements the node streams the measurements it has accumulated since the last data collection round to the gateway. To minimize the number of packet transmissions, nodes aggregate as many measurements as possible in one packet. In practice, up to five such measurements fit in one maximum-size, 128-byte 15.4 frame. This ability to aggregate multiple measurements to a single packet is a side benefit of the architectural decision to decouple data collection from data generation. For reasons explained later in the section, nodes send an empty packet if they have no new measurements. When the gateway eventually receives the token back from the network, it scans the measurements received and recovers lost packets by requesting any missing sequence numbers.

Passing the token in a depth-first fashion ensures that the token travels each network edge exactly twice. For example, in the two level binary tree shown in Figure 8, the edge visiting order is 1, 3, 3, 4, 4, 1, 2, 5, 5, 6, 6, and 2 (12 edges in total). If breadth-first traversal was used instead, the token would travel each edge at least twice, because the token has to travel back to the gateway. In the case of Figure 8 this would lead to 16 edge traversals compared to 12. WRAP further reduces the token passing overhead through inference. First, since a parent forwards all the measurements from its children, it has the opportunity to inspect the MORE_DATA field in their packets and determine when the current child has sent its last packet. When this happens, the parent assumes that the child has released the token. Second, since children can overhear packets sent by the parent, the parent piggy-backs the next child node ID when it is ready to pass the token.

Although WRAP aggressively performs link-level retransmissions, the network can still lose the token for various reasons such as node failure. WRAP puts the burden of token recovery on the gateway. Since nodes stream data only when they hold the token, if the gateway's idle timer expires while waiting for incoming data, it assumes that the token has been lost and regenerates a token with the same ID. Since each token carries an unique ID, nodes that have held a token with the same ID will immediately release it.

3.4.2 End-to-end Reliability

WRAP implements a NACK-based, end-to-end data recovery scheme, whereby gateways request end-to-end retransmissions for missing sequence numbers. To amortize the round trip time incurred in the data retransmission process, WRAP accumulates multiple data retransmission requests destined to the same node.

WRAP encapsulates downstream data requests inside source-routed packets. Doing so, requires gateways to have knowledge of their tree topology. To do so, nodes in the tree piggy-back their parent node ID to the end of the data stream that they send to their gateway. Based on this process, a gateway can rebuild the complete tree topology at the end of a data collection round.

3.4.3 Self-Paced Data Streaming

To stream data efficiently, the source node must determine the inter-packet transmission interval that minimizes self-interference and end-to-end delay. WRAP adapts a technique similar to the one proposed in [11] whereby a node estimates the inter-packet delay by measuring the time between transmitting the last packet of a batch and the last time it overhears the same packet forwarded by nodes upstream. To take into account the whole path, parents propagate the local estimates downstream via the HEARTBEAT message. Then, each child node updates its local inter-packet value to the maximum of the previous local value and the one in the HEARTBEAT message.

3.4.4 Data Time Stamping

RACNet relies on a large number of sensors to perform high fidelity data center sensing. To better correlate the measurements at different locations and generate useful results such as heat maps, nodes must be time synchronized and sample their sensors at the same time. WRAP synchronizes the nodes' clocks through a mechanism that adapts techniques proposed in the Flooding Time-Synchronization Protocol (FTSP) [20].

In more detail, WRAP assumes that the gateways maintain globally synchronized clocks. This is a reasonable assumption as protocols such as the Network Time Protocol (NTP) are readily available in the data center. Gateways timestamp each HEARTBEAT message with the current global time immediately before they transmit them. Upon receiving the HEARTBEAT message, nodes create a synchronization point (i.e., a pair of global and local time stamps). Since different nodes have different clock frequencies and drifts [20], WRAP takes multiple synchronization points on each node and applies linear regression on these data points to model the relation between the local and the global clock. Finally, when a sensor takes measurements, it uses the computed model to convert its local time to the global time.

4 Protocol Design Evaluation

We evaluate WRAP's design using experiments conducted on two separate testbeds. While simulators such as TOSSIM [16] are readily available, they cannot match the full realism that testbeds provide and can therefore lead to inaccurate or misleading conclusions.

The first testbed consists of 62 TelosB motes [25] deployed over a single floor of an office building. While different from the Genomote used in the data center, the TelosB mote shares the same TI CC2420 radio [33] and MSP430 microcontroller [34] with the Genomote. The motes are connected to the building's wired IP network through Ethernet-equipped USB bridges. This configuration allows us to use Ethernet as the management channel to capture detailed timing information, collect management data, and reprogram the devices.

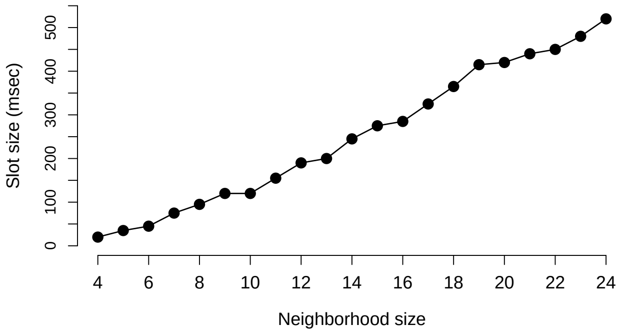

Figure 9. Minimum WRAP backoff slot size necessary to hear HEARTBEAT messages from a node's neighbors. The curve grows linearly with the neighborhood size.

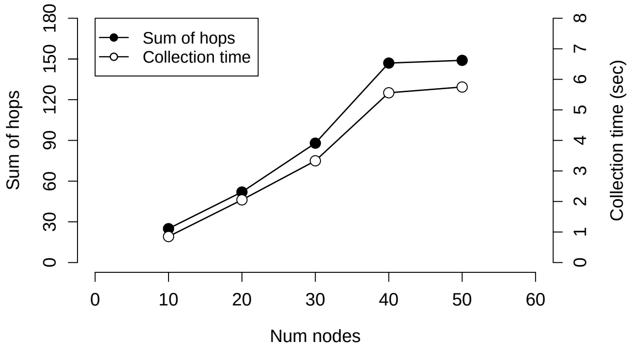

Figure 10. Average one-round collection time and the collection tree's sum of hops as a function of the testbed network size.

The second testbed resides inside a data center environment of approximately 11,000 sq-ft. While the lab testbed allows us to quickly evaluate different protocol parameters (i.e., sampling rate), the data center testbed allows us to test WRAP's scalability at the system's target environment. We placed various number of nodes in the data center following a grid pattern similar to the one used in the RF survey from Section 2.3 and the production deployment described in Section 5.

4.1 Protocol Parameters

Topology Maintenance. *Settling time* is defined as the time necessary for a node to select a stable parent during the system initialization or channel-switching phases. Only after the node has selected a stable parent can the gateway download data from it. Furthermore, the settling time also affects when a node's descendants join the collection tree. It is therefore desirable to reduce this settling time while generating stable and high-quality trees.

WRAP regulates HEARTBEAT messages by having children nodes transmit during a random time selected from their parent's local slot of size T (§3.2). Therefore a larger T reduces the probability of collisions and allows nodes to evaluate more potential parents. On the other hand, larger T values lead to longer delays until the HEARTBEAT messages propagate throughout the tree.

We perform the following experiment to determine a lower bound on T as a function of the neighborhood size. We place a variable number of nodes within one-hop distance from a receiver. For each neighborhood size, we vary T and count the number of HEARTBEAT messages received when each of the transmitters randomly select their HEARTBEAT transmission time from $[0,T]$. The receiver follows the procedure used by a node joining the tree. Namely, for each received HEARTBEAT, it updates its current estimate of the maximum RSSI and LQI seen thus far and the potential parent that transmitted that message.

Figure 9 plots the minimum value of T necessary to receive all the transmitted HEARTBEAT messages as a function of the neighborhood size. It it evident that T grows linearly with the number of potential parents. Administrators can set T according to the expected neighborhood size of the deployment. Considering the size and the density of the production network, we estimate a neighborhood of size 80. Extrapolating from the results in Figure 9, we would need to set T to be around 1600 ms to hear all neighboring nodes (in reality, we set $T = 800$ ms to allow nodes to receive HEARTBEAT messages from 50% of their neighbors).

Data Retrieval. RACNet gateways sequentially collect data via token passing. Thereby the period of a token passing cycle is equal to the total tree collection time. RACNet tries to minimize the collection time over the whole network by evenly distributing nodes over available channels. It does so by using the sum of tree hops as its load balancing metric.

In this experiment we test the hypothesis that the sum of tree hops can be used as a valid proxy for the tree collection time and thus can be used to balance the network's nodes among the multiple trees. We estimate the duration of the data collection round, by running WRAP on our lab testbed while varying the number of nodes from 10 to 50. Each node generates one packet every 30 seconds. Figure 10 illustrates the network size, defined as the sum of all tree hops, and the average data collection time for a single data collection round across the whole network. It is clear that the the sum of tree hops closely follows the collection time and therefore can be used to balance the load over the different trees.

4.2 Application-Level Performance

Data center monitoring and control impose stringent data latency and yield requirements on the data collection process. We evaluate how effectively WRAP meets these requirements using two metrics: *data yield*, defined as the percentage of a node's measurements that successfully arrive at the gateway (including those that use retransmissions) and the average *inter-packet interval (IPI)*, defined as the time interval between the reception of two packets with consecutive sequence numbers from the same node at the gateway. The inverse of the inter-packet interval is a node's *goodput* which is the rate by which unique data (i.e., not including retransmissions) arrive at the gateway.

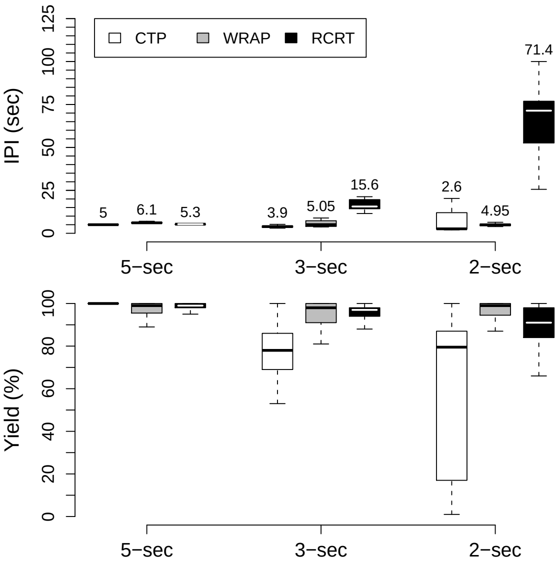

Figure 11. Boxplots of the inter-packet interval (IPI) and data yields under various sampling intervals on the 62-node lab testbed. The number on top of each IPI plot shows the average.

The data yield should ideally be 100% and the inter-packet interval (goodput) should be equal to the node's sampling interval (rate). Note that since WRAP can aggregate multiple sensor measurements in one packet (§3.4), having an inter-packet interval that is higher than the node sampling interval does not mean that the network is becoming persistently backlogged. For example, if nodes generate one packet every 10 seconds and the inter-packet interval is 12 seconds, then a node would have to aggregate two measurements every other five rounds.

Furthermore, yields can fall below 100% due to lost packets. Nevertheless, achieving a yield of 100% in the presence of network losses is still feasible if the gateway persistently issues retransmission requests until it successfully receives all the data. Doing so however can be detrimental to a node's goodput since retransmissions consume network resources. At the same time, transmitting too fast or not recovering from losses can lead to high inter-packet intervals and low goodput. The challenge then for a data collection protocol is to achieve both high data yields and low inter-packet intervals.

We compare WRAP to the Collection Tree Protocol (CTP) [8] and Rate-Controlled Reliable Transport (RCRT) [23]. CTP is a best-effort data collection protocol that does not implement end-to-end retransmissions but rather relies on hop-by-hop retransmissions to reduce packet loss. RCRT implements end-to-end reliability as well as congestion control by controlling the senders' transmission rates. We implemented the same application that periodically samples a set of sensors on top of all three protocols. All of our code is written in TinyOS 2.1 [15]. We use the default CTP version included with the TinyOS 2.1 distribution. We also ported RCRT to TinyOS 2.1 for fair comparison. In all cases, we use a single frequency channel (26) because CTP

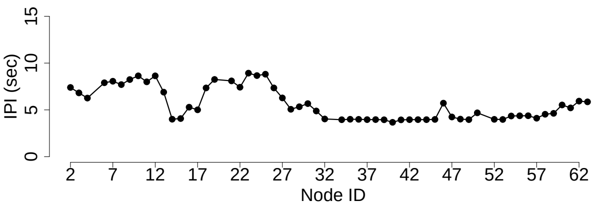

Figure 12. Per-node inter-packet interval achieved by WRAP on the 62-node testbed. Nodes transmit one packet every three seconds. Nodes are labeled according to their location on the testbed.

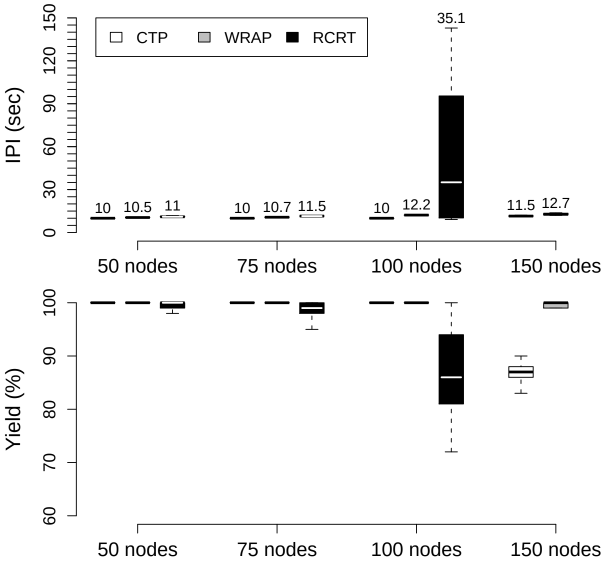

Figure 13. Boxplots of the inter-packet interval (IPI) and data yields under various network sizes on the data center testbed. Each node generates one packet every 10 seconds. The number on top of each IPI plot shows the average.

and RCRT do not have a channel balancing capability.

Sampling Intervals. We first stress the three protocols by increasing the application's sampling rate. Figure 11 presents the three protocols' behavior as we vary the application's sampling interval on the 62-node lab testbed. In each case, the experiment ran for at least 2 hours.

While CTP achieves low packet inter-packet intervals at low network loads, packet losses increase drastically as the network becomes congested. RCRT reacts to this congestion by lowering nodes' transmission rate. We noticed that the RCRT gateway instructed the nodes to reduce their rate to the minimum configured rate (i.e., one packet per 60 seconds) when a 2-sec sampling interval was used. This dramatic reaction was due to the fact that the gateway experienced multiple timeouts while waiting for nodes to acknowledge its requests to lower their rates. Because RCRT treats such timeouts as further signs of congestion, the gateway reacts by lowering the nodes' rates even further. However, even this drastic reaction did not achieve perfect yield in the case of 2-sec sampling.

While CTP ignores congestion, leading to lower yields, and RCRT reactively lowers the nodes' transmission rate, leading to higher inter-packet intervals, WRAP prevents congestion from occurring in the first place by allowing only one network flow at any point in time. As the results from Figure 11 suggest this strategy achieves high data yields and low inter-packet intervals across all sampling rates.

Figure 12 illustrates the conditional fairness property of WRAP. Different parts of the network can have different link quality, especially when the network physically spans a large area. As mentioned before, WRAP tries to increase the packet reception rate with link-layer retransmissions. However, since WRAP bounds the number of retransmissions, it is still possible for nodes with low link quality to miss packets such as the token. The end result is that more network capacity is allocated to nodes with better link quality. This property is desirable as nodes with low link quality do not deteriorate the performance of the entire network. Half of the testbed's nodes (node 1 to 33) have lower link qualities compared to the other half (possibly due to the building structure), and figure 12 shows that these nodes have relatively higher inter-packet intervals.

Network Density. Next, we stress the protocols by increasing the network's density. We do so by uniformly arranging an increasing number of nodes, following a grid pattern, over the same physical area in the data center testbed. Having more nodes in the same space not only increases the amount of traffic that the network must deliver, but also increases contention when node communications are not coordinated, as in the case of CTP and RCRT. To evaluate the effects of network size on performance we fix the application sampling rate to one packet every 10 seconds and increase the number of nodes from 50 to 150. We did not perform the RCRT experiment with 150 nodes because the performance of the protocol (inter-packet interval) degraded appreciably even with a network of 100 nodes. In each case, the experiment ran for at least 2 hours.

As Figure 13 shows, the inter-packet interval increases slightly with the network size. Interestingly, this increase was due to packet loss in the case of CTP and to the longer time necessary to service the whole network in the case of WRAP which achieved 100% yields for all network sizes. As in the previous set of experiments, RCRT reacted to the increased levels of contention by reducing the nodes' transmission rate. Specifically, for the 100-node network the gateway set the nodes' rate to minimum, or one packet every 60 seconds. In practice however the inter-packet interval was even higher because even this decreased rate was not able to prevent network losses and decreases data yields.

5 Production Deployment Results

RACNet has been deployed in several data centers. Next, we present results from a production deployment at a 12,000 sq-ft. facility comprising 696 Genomotes, including 174 wireless master Genomotes. The network uses up to four 802.15.4 channels. The system has been running since mid 2008, collecting more than 2.5 million measurement records per day. Each Genomote chain collects the following measurements every 30 seconds: three temperature readings col-

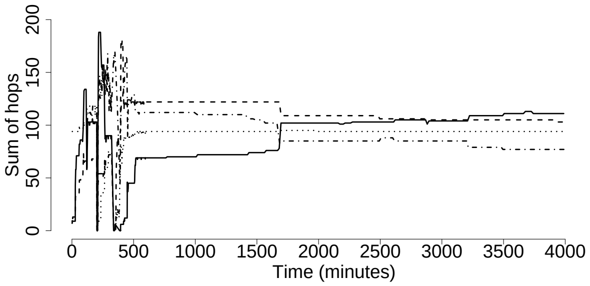

Figure 14. Sum of hops of the four WRAP trees in the production deployment as a function of time. All trees stabilize after a few hours.

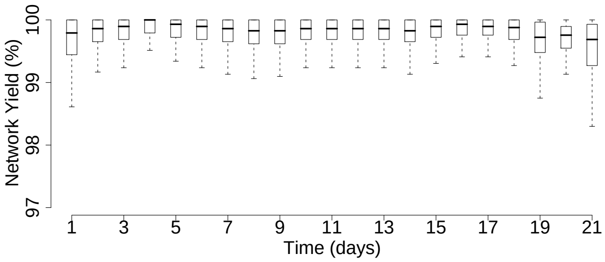

Figure 15. Boxplots of daily network yield from the production deployment over a period of 21 days.

lected at different rack heights, one humidity measurement, and a measurement indicating the availability of the USB power for network monitoring purposes.

Channel Balancing. Figure 14 illustrates WRAP's channel-balancing behavior, using the sum of hops metric during the first three days of the deployment. The first part of the figure ($t < 500$ min) shows significant fluctuations as the network was incrementally deployed and tested. The second part of the figure ($t > 500$ min) corresponds to the phase during which the gateways balanced the load across all four available channels. This phase ended when the difference between the expected load across all channels and the actual load on each channel is within 20% (cf. Sec.3.3), and we did not observe significant variation after this phase.

Data Yield. Figure 15 presents the per-node data yield over a period of three weeks in the production data center deployment. The median yield across all days is above 99.5%, while the lowest yield is always above 98%. This small packet loss is due to the fact that the administrator limits the number of end-to-end retransmission requests to five before WRAP stops the attempt to recover the packet.

Data Collection Latency. We computed the end-to-end latency as the difference between the time the data were timestamped by the node and the time they were inserted into the back-end database. When the network was using three channels, we observed an average of 16 seconds latency.

6 Insights from Sensor Data

The data collected from RACNet are used to improve data center operations and safety. In this section, we give some such examples.

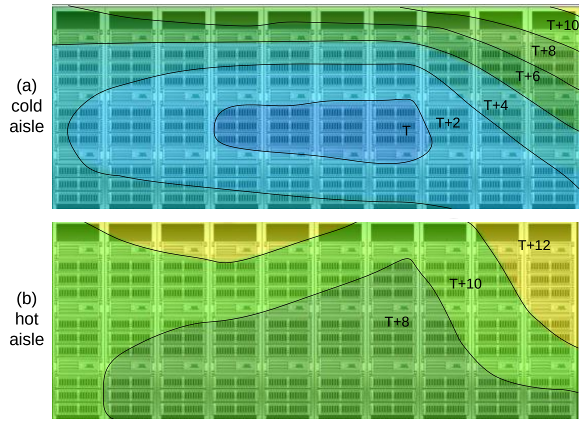

Figure 16. Temperature distribution over the front (cold aisle) and back (hot aisle) of a row of 10 racks. Significant spacial variation demands dense networks of temperature sensors.

6.1 Heat Distribution

Server intake air temperatures across racks are not even. In fact, they vary significantly depending on the relative distances between the racks and the AC units, the types and locations of servers mounted on the racks, and the existence of air blockers that separate cold aisles from hot aisles. Figure 16 presents heat maps generated from 24 sensors located at the front and back of a row of 10 racks supports this argument. In the cold aisle, the temperature difference between the hottest and coldest spots is as high as 10°C. First, hot spots near the bottom of racks is driven by Bernoulli's principle, and fast moving cold air near the floor creates low pressure pockets which draw warm air from the back of the rack [17]. Second, the high temperature at the top right corner is due to uneven air flow which prevents cool air from reaching that area. As a consequence, hot air from the back of the rack flows to the front.

Heat maps such as the one in Figure 16 can be useful in many ways. For example, if cool air can reach the top right corner, then the temperature set point of the supplied air can be increased, leading to energy savings. Furthermore, if an administrator receives an overheating alarm from a server located near the bottom of a rack, the correct course of action is to *decrease* the airflow speed to prevent hot air being drawn from the back. Moreover, these data can be used to control the CRAC system. Instead of using the temperature at the CRAC's return air point to control the cooling level, we can adjust it based on the maximum air intake from all active servers. Nonetheless, designing optimal control laws remains a challenging future task, as changes at the single cooling supply point can affect different data center locations disproportionally.

6.2 Thermal Runaway

Thermal runaway is a critical operation parameter which refers to the temperature changes when a data center loses its cool air supply. However, it is difficult to predict thermal runaway transients through simulations since their accuracy depends on the difficult to obtain thermal properties of IT

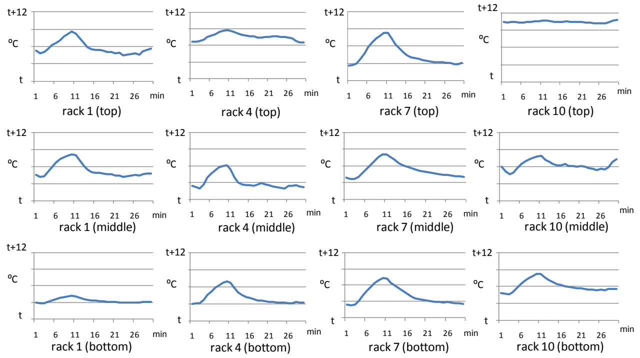

Figure 17. Temperatures collected from locations across a row of ten racks during a CRAC shutdown. Each rack has sensors at the top, middle, and bottom, respectively. Temperature change gradients depend on locations.

equipment. On the other hand, RACNet allowed us to collect actual thermal runaway data during a time period that a CRAC was temporarily shut down for maintenance.

Figure 17 plots the temperature evolution at various locations across a row of ten racks during that maintenance interval. The CRAC was turned off for 12 minutes starting at time 0 in Figure 17. Notice that the mid sections – normally the coolest regions – experienced rapid temperature increases when the CRAC stopped. In contrast, temperature changed moderately at the two ends of the row, especially at the top and bottom of the rack. This is because those racks have better access to room air, which serves as a cooling reserve. This is an important finding because large temperature changes in a short period of time can be fatal to hard drives. For example, the maximum safe rate of temperature change specified for the Seagate SAS 300GB 15K RPM hard drive is 20°C/hr. However, notice that in the middle of rack 7, the rate of temperature change is almost 40°C/hr in the first 15 minutes after the CRAC shutdown. This implies that storage intensive servers need to be placed carefully if the data center has a high risk of losing cooling supply.

6.3 Cooling Effectiveness Analysis

The purpose of a cooling effectiveness analysis is to build dynamic models that link CRAC actions (e.g., increasing or decreasing the chilled water valve opening) to the air intake temperatures at particular servers. Such analyses are important to understanding the cooling capacity at different locations in a data center.

To build such a model, we use the measurements that RACNet collects and the values of the chilled water valve opening to build a discrete-time regression model. The model's input u_k is the opening of the chilled water valve at time k, while the temperature y_k at a particular location is its output. In this example, the CRAC fan speed is constant. We discovered that no linear model can approximate the measurements to a satisfactory degree. Instead, we use a nonlinear model of the form:

$$y_{k+1} = \sum_{i=0}^{N-1} a_i y_{k-i} + \sum_{j=d}^{M+d-1} b_j \sqrt[3]{u_{k-j} - u_0} \qquad (1)$$

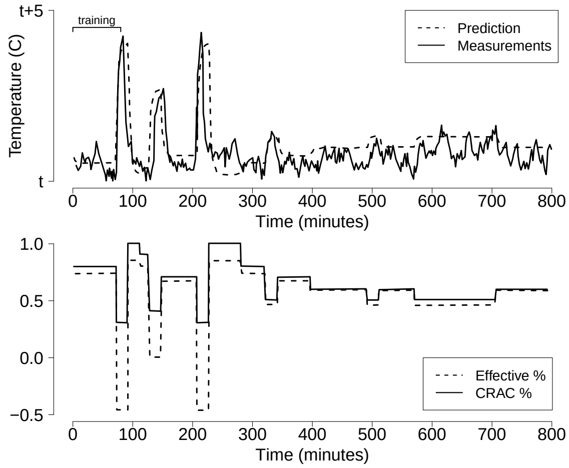

Figure 18. Predicting rack temperatures from CRAC valve openings using a nonlinear regression model.

where d is the *pure delay* in the model, indicating that a change in CRAC valve opening does not have any effect until d time units later. The value of d depends on the distance between the CRAC and the location where the temperature is measured. Moreover, N and M are the orders of the linear regression model and their values also depend on the sensor's location. The lower the sensor is, the smaller the orders are. This indicates that the model captures the air flow dynamics well. The nonlinear term $\sqrt[3]{u_{k-j} - u_0}$ is the so-called *effective input* that captures the nonlinear relationship between valve opening to water flow volume. The values of the a_i and b_j parameters are estimated by performing a least squares fit over a set of collected measurements.

Figure 18 presents the predictive quality of the model from Eq.(1). We build the regression model using the first 160 data points (80 minutes) as a training set and test it on 1,440 data points (12 hours). The top plot compares the predicted temperature to the actual temperature. One can see that the model can accurately capture the temperature trends as a function of valve opening. The bottom panel of Figure 18 plots the input u_k and the effective input (cf. Eq.(1)) as a function of time. It is evident that small decreases in the valve opening u_k near $t = 100, 150, 200$ lead to large changes in the effective input due to the cubic root factor and consequently large temperature increases. This insight derived from the system's measurements can be used to design more efficient CRAC control mechanisms.

The same models can also be used for fault diagnosis. For example, when air dynamics change significantly (e.g., due to objects blocking the air paths) the measurements will deviate from the model's predictions, allowing the operators to take corrective actions.

7 Related Work

Data Gathering Sensor Networks. Sensor networks have been used in several data gathering applications, including environmental [9, 38], habitat [18, 32], and structural monitoring [12, 41]. However, most prior work focuses on outdoor deployments, in which sensors are sparsely deployed and power is the primary concern. On the other hand, RACNet has distinctly different trade-offs. First, power consumption is no longer a determining factor. Instead, performance issues such as data yields and latency become critical. Second, to monitor large data centers at fine spatial granularities, large and dense sensor deployments are necessary. In turn, this dramatic increase in scale leads to solutions that are qualitatively different from those employed in past small-scale, sparse deployments.

In the last year or so, several companies have started to offer wireless sensor networks for data center monitoring. Among them, Federspiel Controls [6] uses OEM sensors from Dust Networks, which incorporate a frequency-hopping protocol called Time Synchronized Mesh Protocol (TSMP) [5]. TSMP network can support up to 255 nodes with a fixed TDMA schedule. Unfortunately, no results on the performance of TSMP are publicly available. SynapSense [31] provides the LiveImaging solution for monitoring data center environment conditions. Little information is known on the networking details of LiveImaging. To the best of our knowledge, LiveImaging supports only five minute sampling intervals (i.e. ten times slower data rate) and does not support multiple frequency channels. Both solutions use battery powered sensors, which limit their sampling rate and system lifetime.

Data Collection Protocols. Data collection has been addressed at length in the sensor network literature. A large portion of the existing work focuses on the power aspect of the problem, aiming to minimize energy consumption through data aggregation (e.g., [19]), ultra-low duty cycles (e.g., [2, 22]), or optimal sensor placement (e.g., [7]). In general, these systems are designed for low data rate applications with no delay requirements. WRAP faces new challenges from the large and dense network configuration and the stringent reliability requirements.

Werner-Allen et al. proposed a request-reply collection protocol called Fetch in the context of their volcano monitoring project [38]. The base station first floods the network with the request, which triggers the target node to return the data. Since data collection occurs infrequently, Fetch does not maintain a dissemination topology. Rather than using per-destination requests, WRAP uses an efficient token passing mechanism to collect data from every node in the network. Lance is a data driven collection protocol that schedules downloads based on the value of the data and the cost of delivery (e.g., energy) [37]. WRAP is a general-purpose data collection protocol, focusing on reliably retrieving all the data to the gateway in a timely manner. Flush is a reliable, single-flow transport protocol for bulk downloads in sensor networks [11]. WRAP adopts the source rate control algorithm from Flush that minimizes the intra-flow interference while streaming data to the gateway. However, WRAP also implements a mechanism for maintaining a data collection tree and utilizes multiple frequency channel to adaptively balance the load among multiple collection trees.

Meliou et al. introduced the concept of data gathering tours, whereby a network's gateway gathers data from a subset of the network's nodes [21]. To do so the gateway calculates a source route that visits all the nodes in the tour. While superficially similar to WRAP's token passing mechanism, data gathering tours are fundamentally different. First, while tours are centrally planed, WRAP is a fully distributed protocol. Second, while data gathering focuses on retrieving data from a subset of the network's nodes, WRAP allows the collection of all the data from very large networks. The work closest to ours is the Rate-Controlled Reliable Transport (RCRT) [23]. However, as the results in Section 4 suggest, RCRT cannot scale to the size or the application data rates necessary by data centering monitoring applications.

A number of multi-channel protocols have been proposed to address the challenges associated with high densities in sensor networks. First, several general multi-channel MAC protocols [14, 43] assign nearby nodes to different channels to improve spatial reuse. The frequent channel switching required in such node-based channel assignment protocols can generate large overhead. Considering the data collection traffic pattern in our application, we decided to adopt a more lightweight alternative: tree-based channel assignment. Instead of assigning different channels to individual nodes, we assign one channel to each spanning tree rooted at a gateway. Channel switches occur only occasionally to re-balance the network's load (e.g., when a gateway joins the network).

Recent work from Le et al. [13] and Wu et al. [40], uses channel assignment strategies that are similar to ours. However, one relies on a centralized algorithm to assign channels [40], while the other achieves load balancing among different trees based on a control theory approach [13]. Both mechanisms do not offer reliable data delivery. In comparison, our approach is both distributed and reliable.

8 Conclusions

The RACNet system presented in this paper is among the first attempts to provide visibility into a data center's cooling behavior, a problem of increasing importance as cooling comprises a large percentage of a data center's energy consumption. At the same time this compelling application challenges wireless sensor network technology in terms of reliability and scalability. The WRAP protocol tackles these challenges by combining three mechanisms: channel diversity, decoupling of tree maintenance from data gathering, and congestion avoidance via a token passing mechanism.

Evaluation results from a medium-size testbed and pilot deployments at a data center suggest that WRAP compares favorably to existing data collection protocols. Specifically, as the aggregate amount of traffic grows, WRAP achieves higher data yields than open-loop protocols such as CTP and higher total throughput than rate control protocols such as RCRT [23]. Furthermore, results from a large-scale production deployment show that WRAP offers stable performance with data yields consistently higher than 99%.

9 Acknowledgments

We would like to thank Microsoft Data Center Services (DCS) team, especially Mike Manos, Amaya Souares, Jeff O'Reilly, Kelly Roark, and Sean James for their support and collaboration on the DC Genome project. We would also like to thank the anonymous reviewers for their useful reviews and Prof. John Regehr for shepherding this paper.

10 References

[1] C. L. Belady. In the data center, power and cooling costs more than the it equipment it supports. *ElectronicsCooling magazine*, 3(1), February 2007.

[2] N. Burri, P. von Rickenbach, and R. Wattenhofer. Dozer: ultra-low power data gathering in sensor networks. In *IPSN '07*, 2007.

[3] B.-R. Chen, K.-K. Muniswamy-Reddy, and M. Welsh. Ad-hoc multicast routing on resource-limited sensor nodes. In *REALMAN '06*, 2006.

[4] G. Chen, W. He, J. Liu, S. Nath, L. Rigas, L. Xiao, and F. Zhao. Energy-aware server provisioning and load dispatching for connection-intensive internet services. In *NSDI '08*, 2008.

[5] Time Synchronized Mesh Protocol. Available at http://www.dustnetworks.com/docs/TSMP_Whitepaper.pdf, 2006.

[6] Federspiel Controls. http://www.federspielcontrols.com.

[7] D. Ganesan, R. Cristescu, and B. Beferull-Lozano. Power-efficient sensor placement and transmission structure for data gathering under distortion constraints. In *IPSN '04*, 2004.

[8] O. Gnawali, R. Fonseca, K. Jamieson, and P. Levis. Robust and efficient collection through control and data plane integration. Technical Report SING-08-02, 2008.

[9] C. Hartung, R. Han, C. Seielstad, and S. Holbrook. Firewxnet: a multi-tiered portable wireless system for monitoring weather conditions in wildland fire environments. In *MobiSys '06*, 2006.

[10] IEEE Standard for Information technology – Telecommunications and information exchange between systems – Local and metropolitan area networks. Specific requirements – Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs). Available at http://www.ieee802.org/15/pub/TG4.html, May 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 *SenSys '07*, 2007.

[12] S. Kim, S. Pakzad, D. Culler, J. Demmel, G. Fenves, S. Glaser, and M. Turon. Wireless sensor networks for structural health monitoring. In *SenSys '06*, 2006.

[13] H. K. Le, D. Henriksson, and T. Abdelzaher. A control theory approach to throughput optimization in multi-channel collection sensor networks. In *IPSN '07*, 2007.

[14] H. K. Le, D. Henriksson, and T. Abdelzaher. A practical multi-channel medium access control protocol for wireless sensor networks. In *IPSN '08*, 2008.

[15] P. Levis, D. Gay, V. Handziski, J.-H. Hauer, B. Greenstein, M. Turon, J. Hui, K. Klues, R. S. Cory Sharp, J. Polastre, P. Buonadonna, L. Nachman, G. Tolle, D. Culler, and A. Wolisz. T2: A Second Generation OS For Embedded Sensor Networks. Technical Report TKN-05-007, Telecommunication Networks Group, Technische Universitat Berlin, 2005.

[16] P. Levis, N. Lee, M. Welsh, and D. Culler. Tossim: accurate and scalable simulation of entire tinyos applications. In *SenSys '03*, 2003.

[17] J. Liu, B. Priyantha, F. Zhao, C.-J. M. Liang, Q. Wang, and S. James. Towards discovering data center genome using sensor nets. In *HotEmNets'08*, 2008.

[18] T. Liu, C. M. Sadler, P. Zhang, and M. Martonosi. Implementing software on resource-constrained mobile sensors: experiences with impala and zebranet. In *MobiSys '04*, 2004.

[19] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. In *OSDI '02*, 2002.

[20] M. Maróti, B. Kusy, G. Simon, and A. Lédeczi. The Flooding Time Synchronization Protocol. In *SenSys '04*, 2004.

[21] A. Meliou, D. Chu, C. Guestrin, J. Hellerstein, and W.Hong. Data Gathering Tours in Sensor Networks. In *Proceedings of IPSN*, 2006.

[22] R. Musaloiu-E., C.-J. M. Liang, and A. Terzis. Koala: Ultra-low power data retrieval in wireless sensor networks. In *IPSN '08*, 2008.

[23] J. Paek and R. Govindan. Rate-Controlled Reliable Transport for Sensor Networks. In *Sensys '07*, 2007.

[24] C. D. Patel, C. E. Bash, R. Sharma, M. Beitelmal, and R. Friedrich. Smart cooling of data centers. In *InterPACK '03*, Maui, Hawaii, June 2003.

[25] J. Polastre, R. Szewczyk, and D. Culler. Telos: Enabling Ultra-Low Power Wireless Research. In *IPSN '05*, 2005.

[26] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis. Interference-aware fair rate control in wireless sensor networks. In *SIGCOMM '06*, Sept. 2006.

[27] Smart Works. http://www.smart-works.com.

[28] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis. Some implications of low power wireless to ip networking. In *HotNets '06*, Nov. 2006.

[29] K. Srinivasan and P. Levis. RSSI is Under Appreciated. In *EmNets '06*, May 2006.

[30] T. Stathopoulos, L. Girod, J. Heidemann, and D. Estrin. Mote herding for tiered wireless sensor networks. Technical Report CENS-TR-58, University of California, Los Angeles, Center for Embedded Networked Computing, December 2005.

[31] SynapSense Corporation. LiveImaging: Wireless Instrumentation Solutions. Available from: http://www.synapsense.com/, 2008.

[32] R. Szewczyk, A. Mainwaring, J. Polastre, J. Anderson, and D. Culler. An analysis of a large scale habitat monitoring application. In *SenSys '04*, 2004.

[33] Texas Instruments. 2.4 GHz IEEE 802.15.4 / ZigBee-ready RF Transceiver. Available at http://www.chipcon.com/files/CC2420_Data_Sheet_1_3.pdf, 2006.

[34] Texas Instruments. MSP430x1xx Family User's Guide (Rev. F). Available at http://www.ti.com/litv/pdf/slau049f, 2006.

[35] The Green Grid. The green grid data center power efficiency metrics: PUE and DCiE. Available at http://www.thegreengrid.org/gg_content/TGG_Data_Center_Power_Efficiency_Metrics_PUE_and_DCiE.pdf, 2007.

[36] TinyOS. MultiHopLQI. Available from: http://www.tinyos.net/tinyos-1.x/tos/lib/MultiHopLQI, 2004.

[37] G. Werner-Allen, S. Dawson-Haggerty, and M. Welsh. Lance: optimizing high-resolution signal collection in wireless sensor networks. In *SenSys '08*, 2008.

[38] G. Werner-Allen, K. Lorincz, J. Johnson, J. Lees, and M. Welsh. Fidelity and yield in a volcano monitoring sensor network. In *OSDI '06*, 2006.

[39] A. Woo, T. Tong, and D. Culler. Taming the underlying challenges of reliable multihop routing in sensor networks. In *SenSys '03*, 2003.

[40] Y. Wu, J. Stankovic, T. He, and S. Lin. Realistic and efficient multichannel communications in dense sensor networks. In *INFOCOM '08*, 2008.

[41] N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin. A wireless sensor network for structural monitoring. In *SenSys '04*, 2004.

[42] J. Zhao and R. Govindan. Understanding Packet Delivery Performance In Dense Wireless Sensor Networks. In *Sensys '03*, 2003.

[43] G. Zhou, C. Huang, T. Yan, T. He, J. A. Stankovic, and T. F. Abdelzaher. Mmsn: Multi-frequency media access control for wireless sensor networks. In *INFOCOM '06*, 2006.