Wireless Data Center Management

IMAGE LICENSED BY GRAPHIC STOCK

*Rahul Khanna, Huaping Liu,
and Thanunathan Rangarajan*

The modern data centers (DCs) are essential to fulfilling ever-evolving computational demands around cloud computing, big data, and IT infrastructure. These DCs are facilities (Figure 1) that house computer systems and associated components such as networking and storage systems. To operate a DC, power supplies, network connections, environmental controls (e.g., air conditioning, humidity), and security infrastructure are needed. Technology and business challenges such as virtualization, load consolidation, real-time troubleshooting, and service-level guarantees require a robust and adaptive server management plan for enterprise. The majority of DC issues are related to overutilization of resources, application failures, data security, power usage effectiveness (PUE), and infrastructure costs. This requires proactive solutions that are business intelligent and built over a network of sense points that are guaranteed to deliver reliable trends and measurements in a reliable and timely fashion. Since it is expensive to build new DCs, the best option is to improve usage of an existing facility through lower infrastructure overhead to deliver better resource management. An optimal sensor network would perform real-time sensor-data collection and deliver a) improved server rack utilization, b) improved DC cooling, and c) improved load-balancing through dynamic capping of thermally constrained systems.

On the infrastructure front, DCs face considerable challenges in seamless integration of telemetry and control functions. These functions are essential to management

Rahul Khanna (rahul.khanna@intel.com) and Thanunathan Rangarajan (thanunathan.rangarajan@intel.com) are with Intel Corporation, 2111 NE 25th Ave., Hillsboro, Oregon 97124, United States. Huaping Liu (huaping.liu@oregonstate.edu) is with the School of EECS, Oregon State University, Corvallis, 97331, United States.

*Digital Object Identifier 10.1109/MMM.2014.2356151
Date of publication: 12 November 2014*

1527-3342/14©2014IEEE

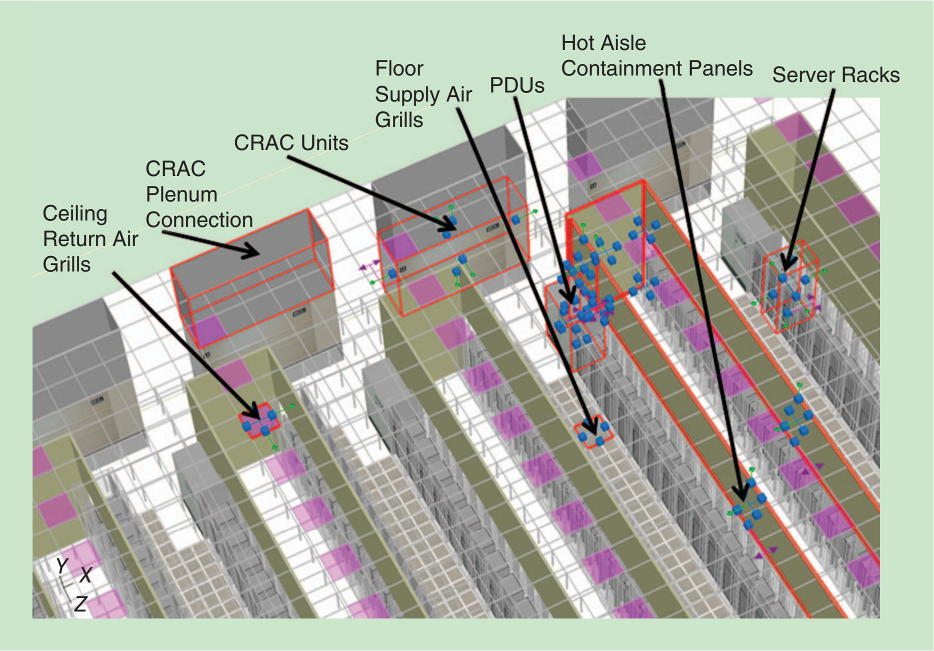


Figure 1. *The essential components of a DC.*

tasks related to power capping, cooling, reliability, predictability, survivability, and adaptability control. It is therefore essential to create an infrastructure of sensors that monitors the physical properties of the dynamically changing environment. The conventional approaches to support distributed sensor data collection and control using wired solutions are static, expensive, and nonscalable. Sensors and control agents supporting this telemetry are a part of a dense and noisy network that are scattered across the DCs. An alternative approach for this unique environment is to use a wireless sensor network (WSN) to improve data efficiency and real-time delivery.

DCs can have anywhere from 10,000 to 100,000 nodes, and each node can generate up to several kilobits of data burst per sampling period. Accurate assessments and analysis of energy efficiency opportunities in a DC requires monitoring multiple environmental parameters (such as temperature, dew point, and pressure in the DC at many locations and elevations), metering of electrical power, and utilization characteristics of each compute node from the electrical substation to its end use. The sense-points data and environmental data is delivered to the supervisory control and data acquisition (SCADA) system in a guaranteed duration to monitor, consolidate, and analyze real-time process control data. The SCADA system uses the sense-point data to produce models and tools for facility management and performance optimization (Figure 2). Monitoring so many parameters is expensive and logistically difficult through a conventional wired monitoring system. According to a recent study conducted by the U.S. Department of Energy, the cost of hard-wired systems ranges from US$1,000 to US$1,500 per sensor node. WSNs achieve equivalent performance at a projected cost of US$100–150 per node (ten times savings). Moreover, wireless systems eliminate the key logistical barrier of placing additional wiring in

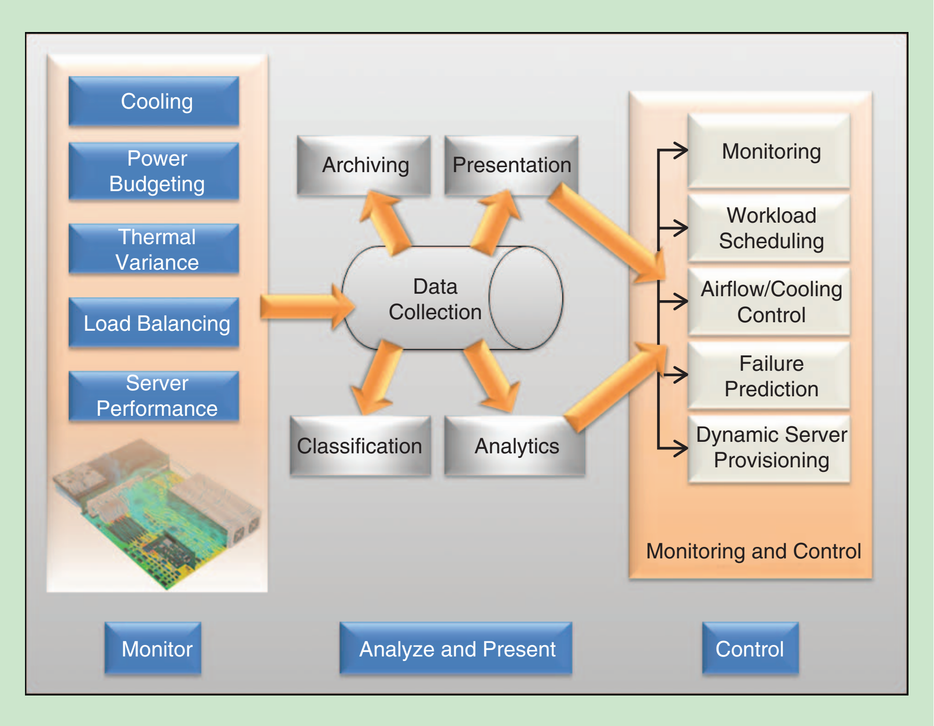


Figure 2. *Technology and business challenges such as virtualization, load distribution, real-time troubleshooting, SLA guarantees, and efficient cooling require a robust and adaptive server management plan for enterprise through efficient and timely monitoring.*

overcrowded racks. Because they are easily expandable and relocatable, wireless systems also provide flexibility to grow and adapt as a DC evolves over time.

Data Center: Thermal Monitoring in Dynamic Environment

A DC is a highly dynamic environment. In this environment, hot spots can be created as a result of temporal events [e.g., increased workload on a set of servers) or spatial events (inefficiency of the computer room air conditioner (CRAC) units in delivering requisite cooling to a particular region in the DC]. In particular, the dynamic nature of workloads means that a bad decision with regard to thermal management could severely impact operational costs, as side effects like hysteresis can cause both increased energy consumption as well as unwanted workload movement. Hence, timely analysis of sensor events is vital to the successful operation of the optimization algorithms. As illustrated in Figure 3, wireless sensors and gateways build a WSN that is capable of measuring, processing, and delivering the sensor data to a central collection point. There are three primary reasons why a naive brute-force decision-making approach would prove inadequate:

1) the dynamic nature of workloads
2) precision and timeliness of sensing physical phenomena such as heat and air flow
3) interplay of DC environs and running workloads.

A close examination of the server platform is necessary to address the above points in the right perspective. At the heart of the platform is the microprocessor chip or the system-on-chip (SOC), which is the primary source of heat generation in the platform. On-die sensors measure heat production in the chip and platform cooling devices such as fans then calibrate their air flow accordingly to provide cooling. As such, if the die temperature sensor is sampled periodically, one would find a net accumulation of heat due to the workload and a net dissipation of heat due to cooling action. Predicting the heat dynamics of the DC therefore involves understanding several dynamic factors at once, assimilating sensor data, and then constructing an instantaneous thermal snapshot of the DC, as shown in Figure 4. It can then be used to predict future thermal behavior and effect control decisions that can minimize hot spots or optimize cooling. For example, an optimum decision is one that produces successive thermal snapshots with progressive diminishment of hot spots. Most prediction algorithms are based on periodic sampling of the states of several entities at once: the DC's sensor network, sensors within servers, the DC's cooling infrastructure, cooling devices (e.g., fans) within servers, and, finally, the set of workloads running on the servers. The prediction logic is hence a discrete-time system, and it predicts the temperature rise in the ith platform at a given sampling instant T as

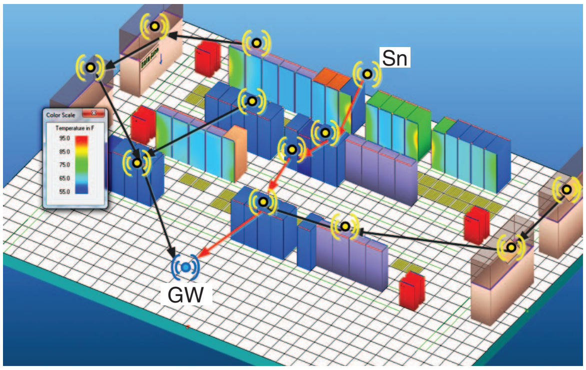


Figure 3. *WSN deployment in a DC: the gateway performs fitness calculations and acts as a data aggregation and WSN topology control agent. It can scan multiple channels to gather data on various subnetworks.*

$$\hat{\varphi}(T) = \sum_{t=T-d}^{T} \mu_t P(t) + \sum_{t=T-d}^{T} \omega_t T(t) - \sum_{t=T-d}^{T-1} \lambda_t C(t). \quad (1)$$

This prediction is based on the observation of sensor and workload data (that we propose to transmit over WSN) over the last d samples for a reasonable insight into the workload and environmental dynamics. The first term on the right pertains to the accumulated power consumed by the running workload (P), which causes a rise in the junction temperature of the component, and is eventually dissipated as heat. The second term pertains to the effects of the local environment (T) in which the server is running, e.g., the effect of a hot spot caused by other racks or other equipment in the vicinity of the server or the redundant cooling delivered by an over-calibrated CRAC unit. The third term refers to the cooling performed purely by the server's cooling system (C), e.g., convective cooling or liquid cooling. The constants μ_t, ω_t, and λ_t must be evaluated at each sampling period and adjusted to reflect the latest thermal snapshot of the system. Equation (1) must be repeated for each server in the DC to yield a prediction vector and its associated coefficient matrices, which is the basis for implementation of an accurate thermal prediction model in the DC management system that can forecast net cumulative temperature rise across equipment in the DC at each sampling instant and make decisions related to workload placement and facilities adjustment to eliminate hot-spot conditions and balance workload to match the delivered cooling by the facilities infrastructure. At the beginning of each sampling period, the prediction model must assess its previous decision relative to the current thermal snapshot. A bad decision must be penalized and corrected using a machine learning approach such as reinforcement learning. The objective in such algorithms would be to reduce the error

$$E_t(T) = [\hat{\varphi}(T) - \varphi(T)]^2. \quad (2)$$

A key observation here is that the availability of timely sensor data from the WSN is vital to ensure accurate decision making. Stale data could produce large deviations from the ideal conditions, resulting in severe hysteresis and complete loss of control, adversely affecting overall total cost of ownership (TCO). Dense WSN installations with highly parallelized subsets enable management software with real-time data for forecasting environmental trends (e.g., direction of flow of heat from hot spots in Figure 4) and perform thermal-aware workload placement.

Building WSN Infrastructure

Through a study conducted by Lawrence Berkeley National Laboratory with SynapSense, it was demonstrated that a WSN could be installed rapidly and at low cost to facilitate delivery of the projected savings. In the modern DCs, a WSN can act as low-cost candidate (ten times) for monitoring tasks as it is nonintrusive, can provide wide coverage, and can be easily repurposed. Within a DC, a WSN system clusters a network of sensor devices that enables real-time monitoring to observe and manage energy, thermal, and performance constraints. A WSN can also be useful as a debugging tool to monitor hot spots, benchmarking, and system forensics, including alerts and alarms.

The general idea is to develop a wireless monitoring infrastructure that can fulfill the following characteristics:

- ability to reduce interference and noise while operating in a dense wireless network
- ability to optimize sensor data flow for efficient battery utilization using network load balancing and route-delay optimization.

Data Center: Wireless Sensor Network Usage

The noninvasive wireless sensors can measure and synthesize historical trends for air temperature, humidity, air particle count, current, power, node utilization, workload performance, service-level agreements (SLAs), and other sense data. They help to model organizational and technological choices to avoid competition for limited controls from multiple applications. Various attributes of the WSN are related to auto-discovery, addressability, event signaling, uniqueness, abstraction model, grouping, and ubiquity.

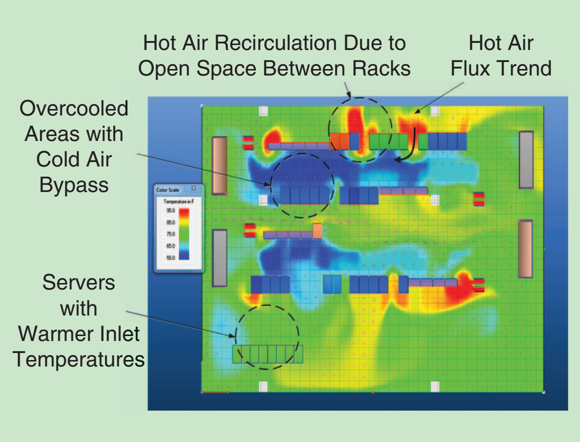

Hot Air Recirculation Due to Open Space Between Racks
Hot Air Flux Trend
Overcooled Areas with Cold Air Bypass
Servers with Warmer Inlet Temperatures
Color Scale
95.0
85.0
75.0
65.0
55.0

Figure 4. *A thermal snapshot of the DC. Efficient monitoring and sense data yield is essential for capturing the accurate heat map.*

The wireless sensor technology in a DC comprises sensor nodes, gateways, routers, server platforms, and software applications (similar to DC Manager). Once the WSN infrastructure is provisioned, it allows DC operator to perform the following functions:

1) accurately measure real-time energy consumption and calculate PUE
2) interpret temperature, humidity, and subfloor pressure differential data from various sensor nodes using live-imaging maps
3) accurately measure server specific performance characteristics and trends for developing statistical models that can forecast resource utilization and energy consumption
4) model relationship between server performance characteristics, energy consumption, and environmental parameters (temperature, humidity, subfloor pressure, etc.)
5) establish baseline energy consumption and identify improvement opportunities by efficient provisioning and loading of server resources
6) using monitoring infrastructure, develop automation strategy that performs adaptive workload provisioning (loading, offloading, migrating, consolidating, etc.), air-flow control, and air-conditioning control
7) monitor environmental conditions to ensure compliance as per the American Society of Heating, Refrigerating, and Air-Conditioning Engineers and provide alerts if the ranges are exceeded.

Deployment and Monitoring Challenges

Unlike traditional WSNs, DC WSN operation is constrained by performance issues related to the facilitie's attributes and placement of sensors in a dense wireless environment. In general, DCs comprise a large number of wireless sensors that are densely deployed and data efficiency and delivery are primary concerns. These concerns can be summarize as:

- *Sensor density.* Unlike sparse distribution as in outdoor sensors, indoor placement of DC sensors pose a challenging problem as they are densely deployed within one-hop communication length from neighboring sensors. This creates interference and collisions that can delay the data packet delivery.
- *DC noise.* Metals are the dominant composition in DCs along with server nodes, server racks, ducts, cables and power-distribution system.

This creates disruptive conditions for reliable and latency-free data delivery.

- *Data delivery.* Sensor data from various nodes and server racks originate in bursts and amounts of several kilobits that needs to be delivered in a guaranteed time. The unique nature of DC wireless networks needs to fulfill certain requirements for effective data collection.
  - The wireless network should be able to operate in an industrial environment that has a large amount of radio-frequency (RF) noise that originates from servers, inverters, WiFi devices, building systems, etc.
  - Time, frequency, and physical diversity should be incorporated to assure reliability, scalability, power source flexibility, and ease of use.
  - The sensor nodes should be ultralow-power wireless transceivers that transfer data to and from integrated sensors or controllers. These transceivers should be able to play a coordinated optimization role with neighboring nodes to eliminate operational interference.
  - Data latency should be minimized for optimal yield and reliability of sensor data.
  - The wireless network should be able to monitor packet throughput, collisions statistics, optimal routing, channel isolation, and feedback the optimal connection to sensor nodes.

Sensors and Gateways

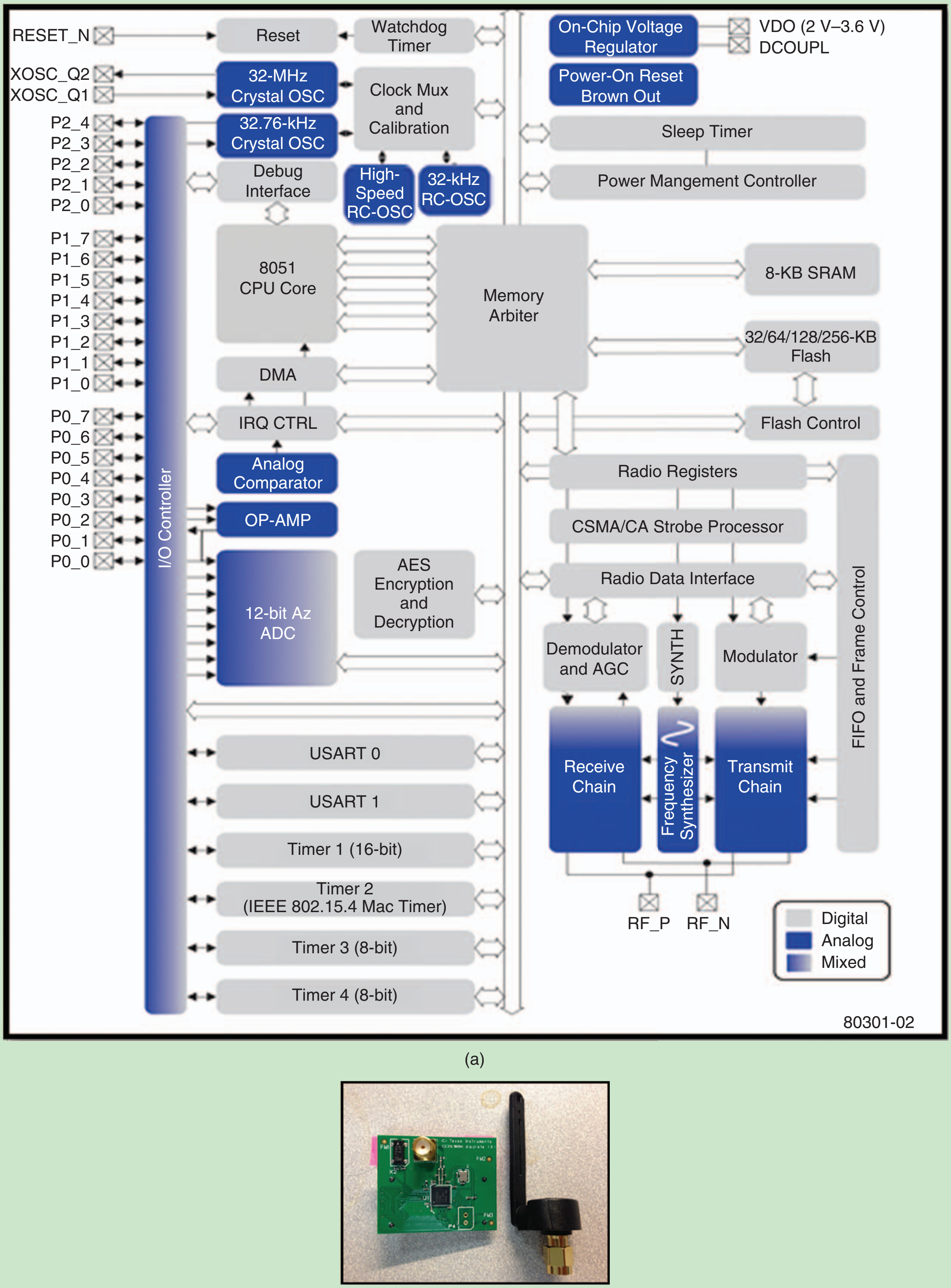
A DC WSN comprises a hierarchy of sensors, gateways, data sink, and data analyzers. Sensors are edge devices that collect the data related to thermal, power, performance, locality, and airflow information and transmit that to the data sink reliably. The sensor data is retained in its local memory until that data is acknowledged by the receiver of the data. Figure 5 illustrates one such widely used SoC, TI CC2530 from Texas Instruments for collecting and transmitting sensor information across WSN. It supports IEEE 802.15.4 [15], Zigbee [16], and RF4CE [17] protocol over 2.4 GHz. The CC2530 combines the performance of an RF transceiver with an industry-standard enhanced 8051 microcontroller (MCU), in-system programmable flash memory, 8-KB RAM, and many other powerful features. The CC2530 power efficiency is supported by various operating modes and short transition times between those operating modes that ensures low energy consumption. It supports RF frequency range from 2,394 to 2,507 MHz, programmable in 1-MHz steps with 5 MHz between channels. This dynamic range facilitates channel diversity that allows the sensor node to select the best channel with low noise and high data throughput. As proposed in the “Data Center Sensor Network Synthesis” section, cooperative information processing by all sensor nodes in a cluster results in an optimally configured WSN.

A gateway acts as an intermediary between a sensor/router and a data sink. Gateways are employed to improve the data throughput, eliminate information redundancy, and exploit locality information for information compression. Figure 6 illustrates a Galileo platform that consists of an Intel Quark SoC X1000 application processor, a 400-MHz, 32-bit Intel Pentium-class SoC. It is capable of providing back-end support for collecting and compressing the sensor information from multiple subnetworks and transmitting that information to the data sink. In addition to the Quark SoC, it supports 256-MB DRAM, 512-Kb embedded SRAM, and a 100-Mb Ethernet port, which is sufficient to execute a high-speed data collection and dynamic evaluation of WSN subnetwork hierarchy.

Handling Sensor Data: Building Collection Trees

A DC sensor network is densely populated with aggressive latency constraints and sampling requirements for optimal cooling and workload distribution. As illustrated in Figure 7, a real-time sensor network is vital to analyze constraints, control load variations, and fulfill dynamic resource demands. Infrastructure is needed to allow reconfiguration of sensors within a server node for optimal coverage, connectivity, and bandwidth to improve the efficiency of functional control. Collection trees forms the basic building blocks of the sensor networks and the related applications. But collection trees working through traditional network protocols suffer from low delivery ratio. Couto et al. [7] proposed an expected transmission count measure to find high-throughput paths on multihop wireless networks, which minimizes the expected number of packet transmissions required to successfully deliver a packet to its ultimate destination. Burri et al. [8] proposed the protocol that coordinates media access control (MAC)-layer, topology control, and routing to construct energy efficient communication subsystem. Madden et al. [9] proposed protocol that enables simple, declarative queries for efficient distribution and execution in low-power WSNs. Koala [10] proposed low duty cycles architecture that exploits the sensor-node idle periods to allow longer sleep times and proactively wakes them up upon bulk data download. Ganesan et al. [12] proposed a joint optimization scheme for sensor placement and transmission structure for data gathering. Sensor nodes are placed in a field such that they aid in minimizing communication energy while reconstructing sensed data at a sink within specified distortion bounds. Jie et al. [11] describe RACNet innovative reliable data collection protocol (rDCP) data collection protocol for high throughput and high reliability data collection using similar concepts of channel diversity and bidirectional collection trees.

We describe a general scheme that uses a machine learning approach for channel allocation using fitness function that incorporates attributes related to



(a)

(b)

Figure 5. *The TI CC2530 SoC [14] solution for an IEEE 802.15.4-based network. (a) The CC2530 SoC block diagram and (b) the CC2530-based IEEE 802.15.4 SoC platform with antenna. (Source: Texas Instruments, CC2330 Data sheet, SWRS081B-April 2009-Revised February 2011.)*

uniformity in allocations, number of hops, route balancing, router density, congestion aware reallocation, data patterns, proximity patterns, and sampling uniformity. A machine learning approach can facilitate sensor network provisioning and reorganization to reduce single-hop sensor node density through synthesizing interference free subnetworks for real-time data collection with latency constraints. This will require an approach to building optimal number of subnetworks for sensor data collection, and data-collection protocol for an individual subtree using time-slot allocation.

Data Center Sensor Network Synthesis

Learning methods such as a genetic algorithm (GA) can facilitate allocation of each wireless sensor node to one of the N subnetwork containers by minimizing the overall interference between neighboring nodes while improving the packet delivery. Furthermore, parallel subnetworks can be synthesized that can operate independently without any interference from other trees. This allows thinning of the dense sensor network and reduction of the average number of nodes that are within one-hop communication range. A GA [5] is one such stochastic search technique that resembles the natural evolution. It supports dynamic reconfigurability and fulfills the need for necessary ingredients required for accurate data acquisition, better data-flow rates, distributed and cooperative management, multiobjective goals, and long-term observability. They enhance real-time usage with parallel solutions that aid in searching multiple points simultaneously and, therefore, avoids being caught in a local optimum.

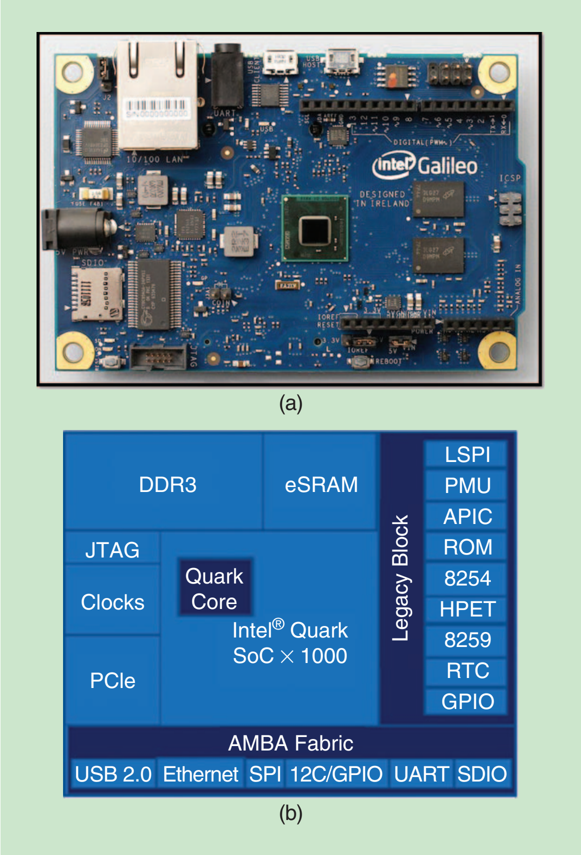


Figure 6. *A WSN gateway: Intel Quark SoC X1000 application processor. The gateway is responsible for sense-data gathering, fitness calculation, topology control, and gateway–gateway communication.*

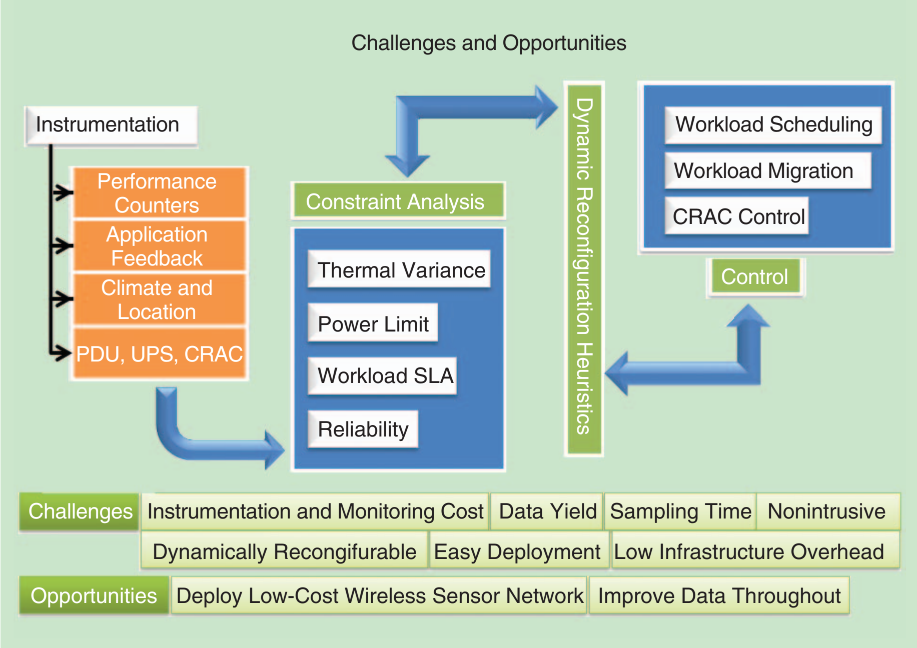


Figure 7. *Real-time data is analyzed for daily trends, load variations, and dynamic resource demands. This along with sensor data from each DC equipment [server nodes, chillers, uninterrupted power supply (UPS) generators, etc.] is transmitted to a DC manager that performs statistical processing and constraint analysis related to power, thermal, workload, and reliability objectives.*

In a DC with large number of sensors placed in close proximity, single-channel communication can drastically reduce the overall throughput due to collisions. In this unique environment with dense sensor-network, limited number of reusable wireless channels can cause co/adj-channel interference. This can degrade the signal-to-noise ratio (SNR) of received packets and consequently the throughput of a DC sensor network. For example, any degradation of data traffic can amount to overcooling and creation of hot spots which can ultimately result in high operational cost of cooling. Therefore in our GA approach, we evolve optimal number of subnetworks through multichannel node allocation in a manner that minimizes the interference while meeting the data latency guarantees, thereby improving the data collection efficiency. We discuss an evolutionary approach to synthesize orthogonal subnetworks through channel diversity to enhance communication performance. We propose a multichannel scheme for dense sensor network, which allocates channels to maximize the parallel transmissions among multiple sensor paths. In the proposed scheme we identify maximum number of noninterfering orthogonal channels (N) that can divide the sensor network (K) into subnetwork represented by K_N.

$$K = (K_1, K_2 \cdot\cdot K_N); \;\; K_i = (A_1^i, A_2^i \cdot\cdot A_n^i), \tag{3}$$

where, A_n^i represents the sensor node n that has been allocated to the i-th subnetwork. Subnetworks can contain multiple trees, each leading to the gateway to achieve maximum throughput. GAs are employed to achieve optimal noninterfering trees. The rest of the section discusses the proposed steps in building the proposed solution

Subnetwork Synthesis: A Machine Learning Approach

We describe a GA-based approach [5] to configure the randomly deployed sensors in a DC into an optimal number of noninterfering independent subnetworks with optimal routes and sensor membership. As discussed later in the "Operational Fitness (F_P): Reconfiguration of Subnetworks" section, each subnetwork parallelize the data collection from its member sensors and sends them to the target in a compressed manner via the most cost-effective route.

As illustrated in Figure 8(c), GAs follow the principle of natural selection, where each individual solution is represented as a binary string (chromosomes) and an associated fitness measure. Successive solutions are built as a part of the evolutionary process where one set of selected individual solutions gives rise to another set for the next generation. Individuals with a high fitness measure are more likely to be selected into the mating pool with an assumption that they will produce a fitter solution in the next generation (next run). Solutions with the weaker fitness measures are naturally discarded. We use roulette-wheel selection to simulate natural selection, where elimination of solutions with a higher functional fitness is, although possible, less likely. There also exists a small likelihood that some weaker solutions may survive the selection process as it may include some component (genes) that may prove useful following the crossover process. Mathematically, the likelihood of selecting a potential solution is given by

$$P_i(\text{Selection Likelihood}) = \frac{F_i}{\sum_{j=0}^{N} F_j}, \tag{4}$$

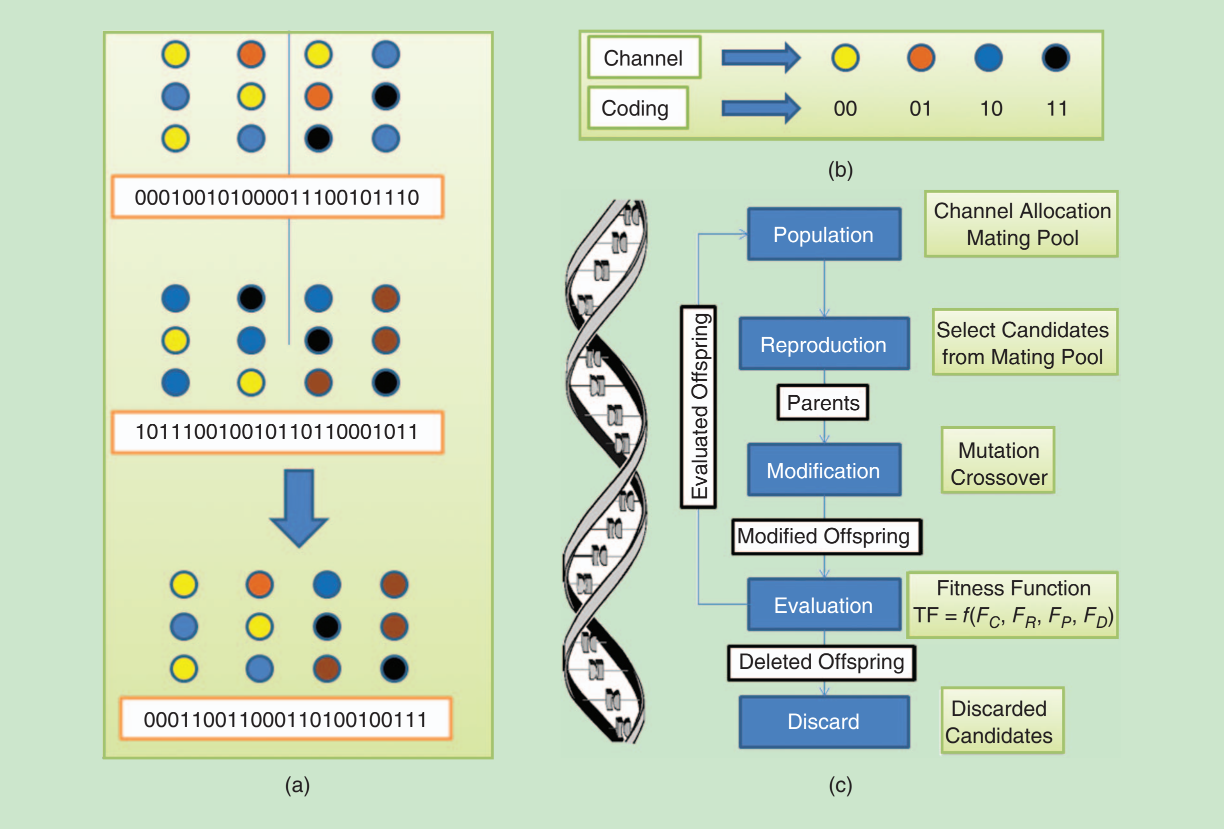


Figure 8. *(a) Crossover process of two-channel-allocation candidates producing an offspring candidate for new channel allocation, (b) the channel allocation 2-bit coding scheme, and (c) a GA for producing channel-allocation candidates.*

where P_i represents the likelihood of a solution to be selected for mating pool, F_i represents the operating fitness of an individual solution, and N is the total number of solution elements in a population. The GA has proved useful in solving complex problems with large search space that are less understood with little domain knowledge.

DC WSNs may use the coding scheme where each individual sensor node channel-code is represented by a 2-bit binary number called a "gene" [Figure 8(b)]. These 2-bit genes define the subnetwork to which the node belongs are called "allele." The chromosome of the GA represents the building blocks (allele) to a solution of the problem that is suitable for the genetic operators (crossover and mutation) and the fitness function. As illustrated in Figure 8(a), two candidate solutions undergo a modification using a crossover function and results in a new candidate solution that undergoes an evaluation for candidacy in a new mating pool. The evaluation process of the candidate solution uses weighted sum of the individual objectives, as defined in the section "Measuring the Quality of Subnetwork," to calculate the quality of overall fitness (also called "cumulative fitness"). Each individual objective functions measure the quality of a specific goal related to a) reduced channel interference, b) balanced routing, c) operational efficiencies, and d) data collection efficiency.

In the modern DCs, a WSN can act as low-cost candidate (ten times) for monitoring tasks as it is nonintrusive, can provide wide coverage, and can be easily repurposed.

Initialization: Characterization of Sensor Nodes

First step in constructing subnetwork in DC is to discover the proximity patterns between sensor nodes (Figure 9) originating from the WSN Gateway. While increasing the transmission power reduces the effect of noninterfering channels, reducing the transmission power increases the number of hops to reach the gateway. Either condition can degrade throughput.

As a part of initialization step, we use configurable RSSI threshold to filter out the weak links, thereby eliminating them from evaluation. As illustrated in Figure 10, RSSI data of all the nodes (For Example, Node N in the figure) is received by the Gateway either directly or through an intermediate proxy. Upon receiving the RSSI information, Gateway executes a *Multilateration Localization* [6] scheme that maps the location coordinates of all the nodes. Each node on the map evaluates the link quality of neighboring nodes using RSSI information. Multilateration is a range-based, decentralized localization algorithm that uses intersecting circles centered on the reference nodes and having radius equal to the estimated distance between reference nodes and blind node. We use the reference nodes with the shortest estimated distance relative to the Blind Node. Accuracy can be enhanced by using more than four reference nodes. RSSI can be evaluated by the following equation:

$$\text{RSSI} = 10 \cdot \log \frac{P_{\text{RX}}}{P_{\text{ref}}}; \quad P_{\text{RX}} \propto P_{\text{TX}} \cdot \left(\frac{\lambda}{4\pi d}\right)^2 \tag{5}$$

where, λ is the wavelength of operation, P_{TX}, P_{RX} are the transmitted and received power respectively, d is the distance between sender and receiver sensor and P_{ref} is the reference power typically set to 1 mW. Transmission power (P_{TX}) acts as a control parameter to isolate two interfering clusters. Each node advertizes its P_{TX} during the initialization phase. Receiving nodes measures the P_{TX} of the received signal. P_{RX} and P_{TX} helps in evaluating the relative distance between two nodes using the equation 5. RSSI information that is based on P_{RX} signifies the quality of communication link between the transmit-receive pair nodes.

To build subnetworks using GA approach, we need to identify the effect of neighboring nodes on each node. We start by broadcasting an advertisement (ADV) message from the gateway. A generic ADV message comprises its node identification (ID), parent-ID, children-list, and transmission power (P_{TX}). Upon hearing the ADV messages, all the sensors within the one-hop distance of the gateway tag themselves as L1 nodes, assign parent-ID, and record the received signal strength indicator (RSSI) into its local storage. This information, along with its node-ID is transmitted back to the parent (in this case Gateway) through contention-based approach. This Gateway-L1 process continues till all L1 nodes have responded to the Gateway. Upon completion, Gateway sends ADV_C message that identifies completion and selects child-ID that should send it's own ADV message. Upon receiving ADV message from L1 node, a new population of L2 nodes is generated that is within one-hop distance of L1 node selected by Gateway. Source-L1 nodes can also be overheard by subset of peer L1 nodes that are within one hop communication distance. L2 nodes update the corresponding information (RSSI, Self-ID, etc) to the source-L1 node. Peer L1 nodes also updates the information to Gateway. This process continues in breadth-first manner till all nodes are accounted for. The node-provisioning process is terminated after one or more of the following conditions are met:

- Timeout: Each L_x enumeration stage is bounded by a timeout, after which the enumeration moves to the nest stage (L_{x+1})
- Count: Since finite number of sensors are used in a DC, the total number of nodes checking-in with the gateway should match with node-count.
- Coverage: Nodes not responding from within a specific coverage area can be identified.

At the end of process, we will have n-levels of nodes distribution. Figure 9 illustrates this process which

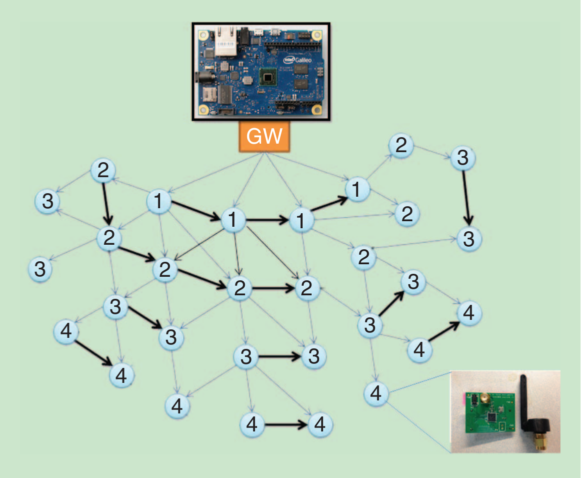


Figure 9. *Illustration of sensor nodes characterization wrt. Gateway (GW). Thick lines represent peer-level communication, other lines represent communication between nodes in different levels of communication hierarchy. Each node-level is represented by number 1, 2, 3, ...*

builds four levels of distribution. Each node is characterized by its ability to communicate with its neighbors with a measured RSSI. Next step in this process is to extract N subnetworks, where each network is allocated by a nonconflicting channel. The average density of network is reduced by a factor of N. Each network can further be divided into multiple clusters that are separated by noninterfering characteristics.

Sensor-Data Collection: Protocol

Data collection protocol is initiated by the gateway by traversing the DATA_SEND message to all the nodes sequentially on the parallel tree (trees on separate channels). Data is collected in depth-first manner and cached into the parent before transmitting up-stream. Periodically gateway traverses ADV_CALIB message through each single tree and ADV_SENSE through the rest of the trees. This allows a candidate node to broadcast ADV message (CALIB) so that the rest of the nodes can listen (SENSE) to that message by switching to the ADV channel, thereby calibrating there measurements with respect to each other. These measurements are delivered up-stream to the gateway as training data for executing GA function. This process is distributed over time to avoid long periods of inactive sensor measurements. This protocol maximizes the data collection by parallelizing the data flow through multiple subnetworks.

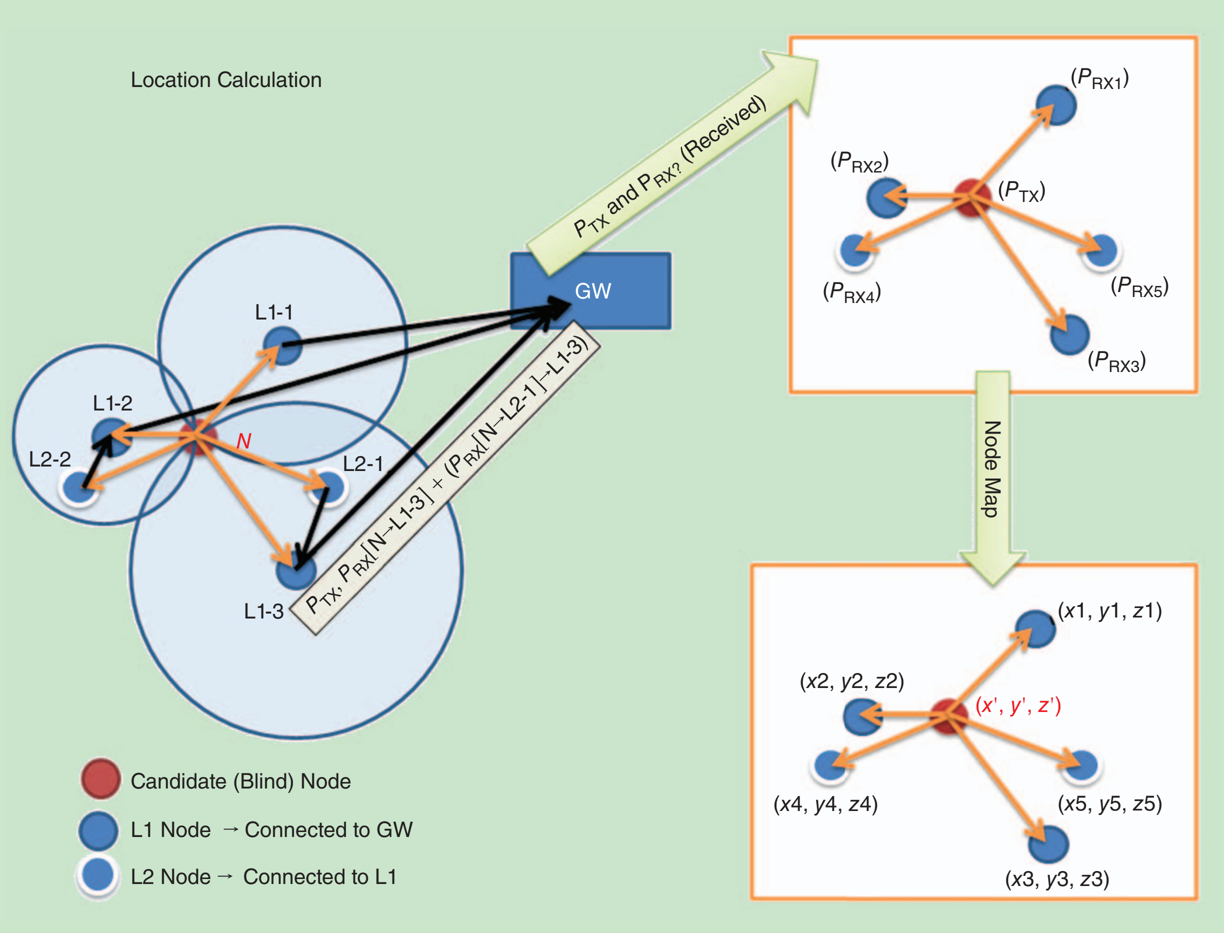


Figure 10. *Cooperative Location Mapping using RSSI. The location of a Blind- Node (N) is dynamically evaluated against the nodes (L1-1, L1-2, L1-3, L2-1, L2-2) with pre-evaluated coordinates. This evaluation propagates through the entire network.*

DATA_SEND Protocol

DATA_SEND Protocol minimizes the amount of messages that need to traverse through the tree. We employ WAIT_SLOT attributes to all member nodes throughout the subtrees that parallelize the data collection though the single tree. Each WAIT_SLOT represents earliest time a node can transmit a collection of all down-stream sensor node packets to its parent.

Assumptions

- Each node is aware of its weight (number of all the nodes downstream) and the weight of its children.
- Data collection can proceed in parallel on subnetworks that communicate on different channels.

Protocol

- Parent nodes enumerates all the children nodes according to number of nodes downstream.
- Parent traverses through enumerated children to send WAIT_SLOT threshold and sequence number through DATA_SEND message. This threshold identifies the earliest time-slot the children nodes can communicate with the parent (with measurement data).
- Parent continues this process till all children on the subtree are enumerated.
- This process continues till all leaf-nodes are enumerated.

Measuring the Quality of Subnetwork

In this section we develop the fitness criteria that executes in the Gateway and measures the quality of solution synthesized by recursive execution of a GA [5]. The solution attempts to allocate each sensor-node to one of the N subnetwork containers by attempting to minimize the overall interference between neighboring nodes. Each subnetwork is a local cluster of nodes that are on the same channel and separated from other cluster by either channel diversity or by maximizing the distance that separates closest members of two clusters on same channel. Each individual candidate solution is measured using the cumulative fitness criteria (TF) comprising of weighted sum of four performance attributes: a) reduced channel interference, b) balanced routing, c) operational efficiencies, and d) data collection efficiency. By improving the quality of subnetwork, we achieve a better packet yield at the collection point or improved packet-receive-ratio (PRR).

Channel Fitness (F_C): Evolution of Subnetworks

Channel Fitness is the *first weighted component* of cumulative fitness. Evaluation step of GA measures the quality or performance of channel distribution that leads to reduced interference between sensor-nodes.

It is essential to create an infrastructure of sensors in a data center that monitors the physical properties of the dynamically changing environment.

We construct channel selection fitness function which is a weighted component that measures the quality or performance of a solution, in this case reduced interference between sensor-nodes. From the previous step ("Initialization: Characterization of Sensor Nodes"), each node has set of neighbors defined by $A_i = (a_1^i, a_2^i \cdot\cdot a_n^i)$. Fitness function rewards the following conditions that helps to construct the initial network:

- Each node makes best effort to share channel with at least one L_{n-1} node. If such connection is not found, then the node can share channel with it's peer level node (L_n). It may be possible for the peer node to connect to L_{n-1}.
- Nodes in each hierarchy are rewarded if they can reduce the interference from L_n & L_{n-1} nodes. Although, eliminating it completely will violate the previous condition.
- If two nodes share single channel at level L_n, nodes are rewarded if they choose same parent at level L_{n-1}.
- An exclusive channel (ADV Channel) is reserved for control information. This information is related to broad-casting data sink messages as well as broadcast messages from newly added sensor. Sensors switch to this channel proactively when idle.

Once the channels are allocated, individual nodes undergo channel characterization to discover multiple subnetworks and the corresponding topology details. This process is similar to *First Step* as defined in the "Initialization: Characterization of Sensor Nodes" section, except that now it is performed at the granularity of a channel (or subnetwork). End result of this process is to create multiple subnetwork clusters for each noninterfering channel, where each cluster on similar channel is separated by more than one-hop distance.

Channel fitness function that guides the construction of subnetworks according to these conditions can be summarized as:

$$F_C = 1 - \frac{1}{3 \cdot K} \cdot \sum_{i=0}^{L} \sum_{j=0}^{N} \left(R_{ij} + \frac{I_{ij} - 1}{M_{ij}} + \frac{P_{ij} - 1}{\hat{I}_{ij}}\right), \qquad (6)$$

where, $R_{ij} = 0$, if there exists a node k that shares channel with node j in the hierarchy $i-1$, M_{ij} represents the number of neighboring nodes to j at level i or below, I_{ij} represents number of shared channels with node j and level i or below, $K = (L \cdot N)$ represents total number of nodes and $\hat{I}_{ij}$ represents number of nodes sharing the channel with node j at level i and P_{ij} represents number of parents that catering to all nodes represented by $\hat{I}_{ij}$. If the parent doesn't exist for the node, it is still represented by NULL parent. In a hierarchical structure, there exists at least one candidate node in $j-1$ level that

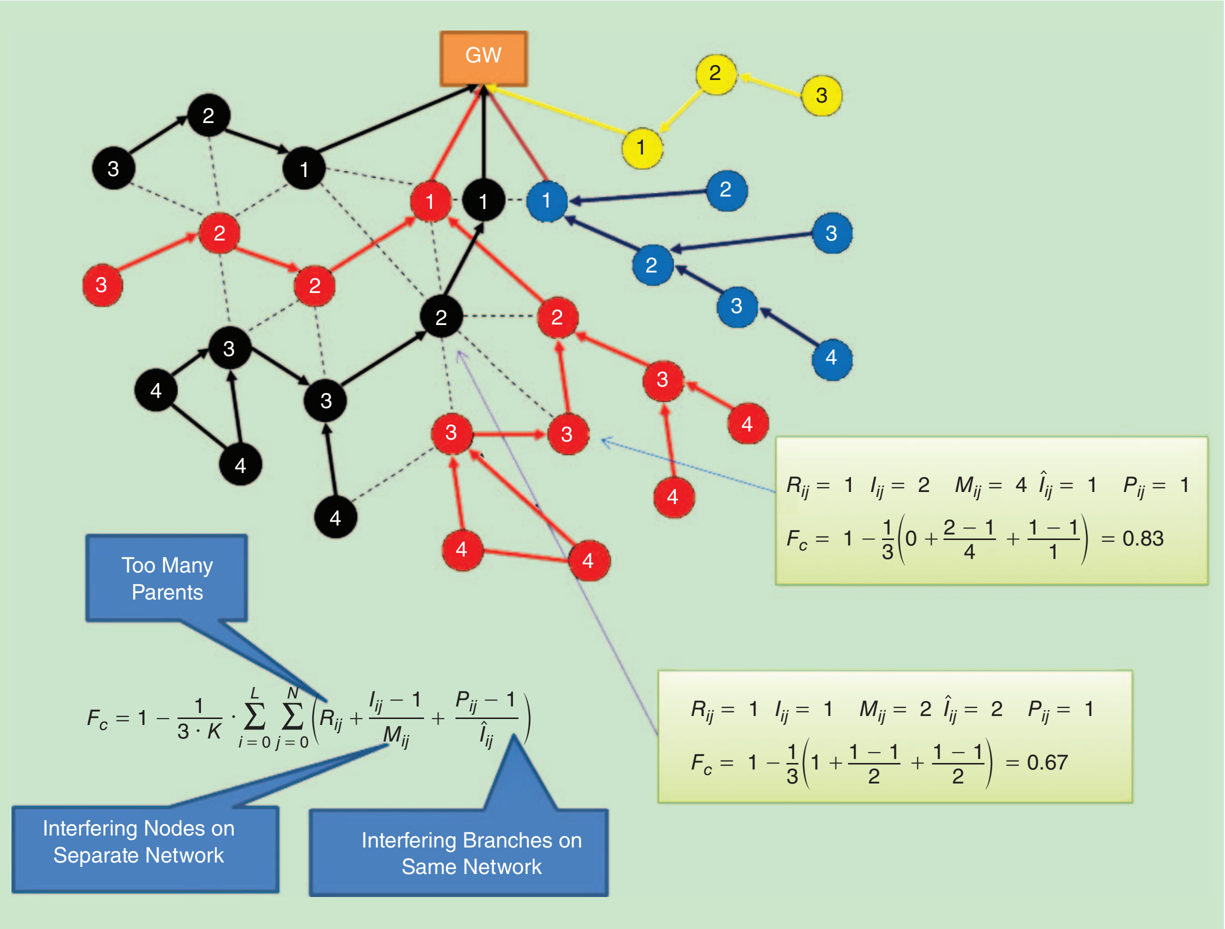

GW
$R_{ij} = 1 \quad I_{ij} = 2 \quad M_{ij} = 4 \quad \hat{I}_{ij} = 1 \quad P_{ij} = 1$
$F_c = 1 - \frac{1}{3}\left(0 + \frac{2-1}{4} + \frac{1-1}{1}\right) = 0.83$
Too Many Parents
$F_c = 1 - \frac{1}{3 \cdot K} \cdot \sum_{i=0}^{L} \sum_{j=0}^{N} \left(R_{ij} + \frac{I_{ij}-1}{M_{ij}} + \frac{P_{ij}-1}{\hat{I}_{ij}}\right)$
$R_{ij} = 1 \quad I_{ij} = 1 \quad M_{ij} = 2 \quad \hat{I}_{ij} = 2 \quad P_{ij} = 1$
$F_c = 1 - \frac{1}{3}\left(1 + \frac{1-1}{2} + \frac{1-1}{2}\right) = 0.67$
Interfering Nodes on Separate Network
Interfering Branches on Same Network

Figure 11. *Channel Fitness (F_C): Illustration of channel distribution (represented by different colors) that builds multiple subnetworks. Black node (Marked) in the center of graph at level 2 has a potential to interfere with another black subnetwork illustrated with dotted line at level 1 (we cannot eliminate this situation completely).*

can act as parent. Figure 11 illustrates the distribution of available channels (four) among sensor nodes and channel fitness of two arbitrary nodes. In general, fitness of all nodes is evaluated and then averaged to find the fitness of the overall solution. The best effort methodology builds subnetworks to reduce the interference between neighboring nodes. Although collisions cannot be ruled out completely, they can be minimized. Nodes that are at single hop communication distance and share the same channel at the same level can identify the parent that can schedule the data delivery by using pull protocol or time-scheduling. Although, using this technique the channel distribution may be optimized for minimizing interference, it may not be optimized for load balancing. The next section enhances the fitness measure to allow balance loading through the network to minimize data-transmission latencies.

Route Fitness (FR): Synthesizing Balanced Routes

Route Fitness is the second weighted component of cumulative fitness. Evaluation step of the GA measures the sensor-data trip-delay performance that leads to channel-reallocation to construct balanced routes. As discussed earlier, DC sensor-nodes generate several kilo-bits of burst data in one sampling period. This data along with rest of the sensor data has to traverse several hops before reaching the gateway. Once the channel allocation solution is applied (Figure 11) among sensor nodes, the number of possible routes are reduced as well. A bad solution would result in elimination of efficient routes that could have been possible with an alternate channel distribution. We define route fitness (F_R) that rewards the channel allocation resulting in optimal delay paths according to the target requirements. Unexpected delays on routes connecting sense-points and gateway can reduce the effectiveness of the received sensor data. Potential routes are evaluated using route fitness F_R (7) that depends on a) congestion—missed or delayed packets and b) average latency—round trip time (RTT) of sensor data transaction.

$$F_R = 1 - \frac{1}{2}\left[\frac{1}{N}\sum_{n}^{N}\min\left(1, \frac{|\tau_n - \bar{\tau}_n|}{\bar{\tau}_n}\right) + \frac{A}{R}\sum_{r}^{R}\frac{T_r}{\hat{T}_r}\right], \quad (7)$$

where τ_n and $\bar{\tau}_n$ are the measured delay and expected delay between end-to-end traffic serviced by connection n respectively, T_r is the number of packets either missed or delayed by intermediate nodes (proxy routers) $r, \hat{T}_r$ is the total packets serviced by intermediate nodes (proxy routers) r and A represents the amplification factor. A connection n is represented by all the routes terminating at L1 level nodes. Router node r is an intermediate node that fulfills the function of sensor as well as router (for at least one node).

Route fitness F_R factor rewards the reconstruction of routes that meets delay requirements to rebalance the overloaded nodes. An interesting property of a GA is that every node seeks to attain the shortest path that is optimized for low interference and low latency to the base-station. This property comes from the fact that the algorithm first identifies interference patterns for all nodes and builds a L-level hierarchy that represents nodes hop-distance from the base-station (or Gateway). Algorithm rewards a hierarchy where each node connects to parents that uses lesser hops (one-hop less) to the base-station. First, each node builds profile structure that contains information regarding all other nodes that can overhear it and its position in the L-level hierarchy. This is a centralized operation that is performed once during the deployment and very infrequently during maintenance cycles. Deployment phase extracts parallel subnetworks that can operate independently without any interference from other trees. This allows thinning of the dense sensor network and reduce the average number of nodes that are within one-hop communication range. Furthermore, deployment phase also involves calibration process where sensor-nodes are activated and characterized for latency and other effects by running GA that combines the effect of channel allocation (equation 6) and route balancing (equation 7).

Operational Fitness (F_P): Reconfiguration of Subnetworks

Operational Fitness is the third weighted component of cumulative fitness. Evaluation step of the GA measures the data transmission efficiency that leads to optimal tradeoffs between data compression opportunities, channel interference and balanced routes. Once the WSNs are deployed for real-time monitoring it is not practical to make changes to the WSN infrastructure too often. Dynamic conditions in the DC such as hardware provisioning, data traffic variations and noise conditions require reconfiguration of WSN over time. To avoid disruption, the process of reconfiguration requires an adaptation mechanism that evolves over time with least amount of intrusion to the existing configuration. Evolutionary mechanisms utilize passive measurements of the characteristic behavior of sensor-nodes over time. These measurements are used to evaluate potential modifications in the sensor network which can be summarized as:

a) *Proximity Patterns:* As defined in the "Initialization: Characterization of Sensor Nodes" section, these patterns define one-hop neighbors of each sensor. As sensors are added, replaced or removed, new patterns emerge and are recorded using control channel.

A key observation here is that the availability of sensor data using WSN in a data center is a cost effective solution to achieve accurate decision making.

b) *Data Patterns:* Sensor data from multiple sensor nodes demonstrate certain patterns that are typical of a local context (e.g., cooling devices). Sensors sharing similar context can have spatial data redundancy which can be exploited using common models. This can reduce the amount of data flows through the network. Nodes that demonstrate steady-state sense-data can exploit temporal redundancies and transmit fraction of data that changed during the sampling interval. Data patterns characterizes the data flow and evaluate the burst patterns.

c) *Router Density:* Reassigning sensor node to an alternate channel will trigger migration of all down-stream nodes to the same channel. This can increase the probability of uncovering co-channel interference between newly-configured nodes and disrupt sense-data delivery.

Measuring the cumulative effects of these dynamic variations allow us to analyze the link quality using a Fitness function (F_P) that incorporates all the dynamic conditions and can be represented as:

$$F_P = 1 - \frac{1}{N}\sum_{i=0}^{N} \delta_i \left[\chi\eta_i + \mathrm{MIN}\left(1, \frac{\mathrm{MAX}(0, (\bar{\lambda}_i - \lambda_i))}{\lambda_i}\right)\right], \quad (8)$$

where χ is the Amplification factor, κ_i counts downstream nodes relative to node i, $\delta_i = 1$ if node i changed it's parent, λ_i and $\bar{\lambda}_i$ are the average data rate at node i's parent before and after the node changed its parent. A change in the parent can result in variation in the data compression ratio and alter the data rates. Optimal route would exploit spatial correlation between sensors that share similar behaviors and trends.

Collection Fitness (F_D): Efficient Data Gathering

Collection Fitness is the fourth weighted component of cumulative fitness. Evaluation step of GA measures the data collection efficiency quantified by sense-data collection on all subnetworks in shortest possible time.

Figure 12 illustrate the process of isolating data collection among multiple tree. Data parallelism is maximized by employing channel-separation and time-slot reservation. Nodes on the same channel communicate with the parent based on time-slot reservation that is adjusted and communicated up-stream according to data-collection delay statistics. Initially, static time-slots are reserved

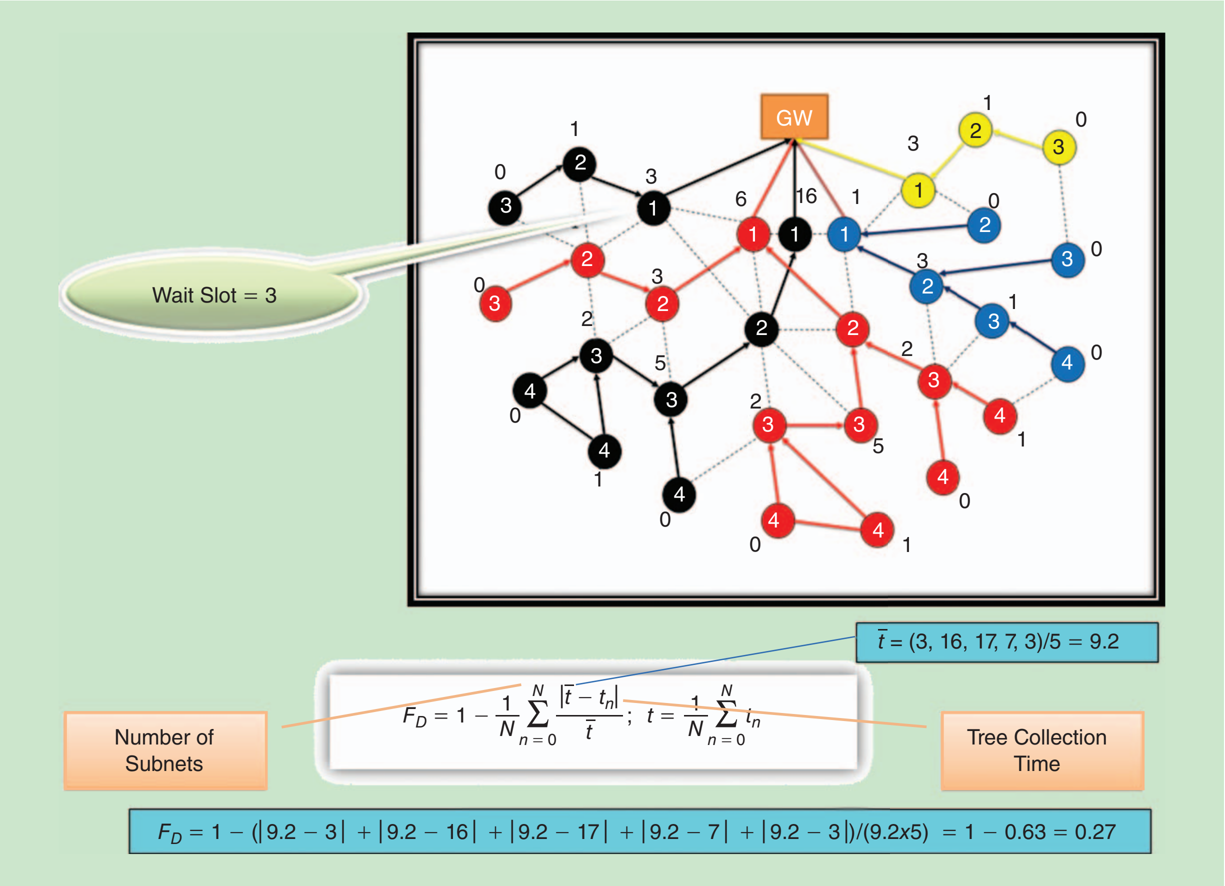

GW
Wait Slot = 3
$\bar{t}$ = (3, 16, 17, 7, 3)/5 = 9.2
$F_D = 1 - \frac{1}{N}\sum_{n=0}^{N}\frac{|\bar{t}-t_n|}{\bar{t}}$; $\bar{t} = \frac{1}{N}\sum_{n=0}^{N} t_n$
Number of Subnets
Tree Collection Time
F_D = 1 − (|9.2 − 3| + |9.2 − 16| + |9.2 − 17| + |9.2 − 7| + |9.2 − 3|)/(9.2x5) = 1 − 0.63 = 0.27

Figure 12. *An illustration of parallel data flow through multiple subnetworks. All nodes in different colors represent member of corresponding subnetwork operating on different channels and can operate in parallel.*

according to maximum packet size to avoid hidden-Node collision. For example, in case the black color channel at level-1, the weight is set as 3. This illustrates the waiting channel at level-2 need to wait at least three waiting slots before transmitting its sense-data. Data collection effectiveness is quantified by the fitness that enables the sense-data collection on all subnetworks in shortest possible time. Fitness function is summarized as:

$$F_D = 1 - \frac{1}{N}\sum_{n=0}^{N}\frac{|\,\bar{t}-t_n\,|}{\bar{t}};\ \bar{t} = \frac{1}{N}\sum_{n=0}^{N} t_n, \qquad (9)$$

where, N is the number of subnetworks, t_n is the average sampling duration of subtree n to collect sense-data of all downstream nodes. Nonuniform sampling duration results in delayed collection of sense data at the central collection server.

Total Fitness

Total Fitness (TF) is the weighted sum of individual fitness and represents optimal subtree construction. Optimized tree would support channel allocation that is free of interference and route congestion. Furthermore, optimal tree would aid spatial compression, scalability and low latency routes. The TF function can be represented as:

$$\text{TF} = \alpha_1 F_C + \alpha_2 F_R + \alpha_3 F_P + \alpha_4 F_D; \qquad (10)$$

where α_n is the relative weight of the fitness components.

Sensor Network Performance

In this architecture, a GA executes and trains on the Gateway Server (Intel Quark SoC X1000 Application Processor) where it gathers the sensor data from corresponding nodes and forwards it to the DC management agent. Additionally, the gateway monitors and analyzes data collection trends and patterns from each individual nodes as well as established routes. The analysis of these trends constructs a feedback loop that influences the configuration of the WSN topology and sensor allocation through machine learning. These trends relate to inter-sensor interference patterns, Node specific packet delivery behavior (sensor data rate, burst patterns) and route specific packet behavior (packets lost, delayed). Although individual sensors do not participate autonomously in the training process, they configure (or reconfigure) themselves to new roles and assignments as a part of continuously evolving solution (or network topology). Gateway server coordinates the sensor topology and provisioning. It monitors trend, pattern, channel-interference, throughput, sensor additions/deletions, link-quality/

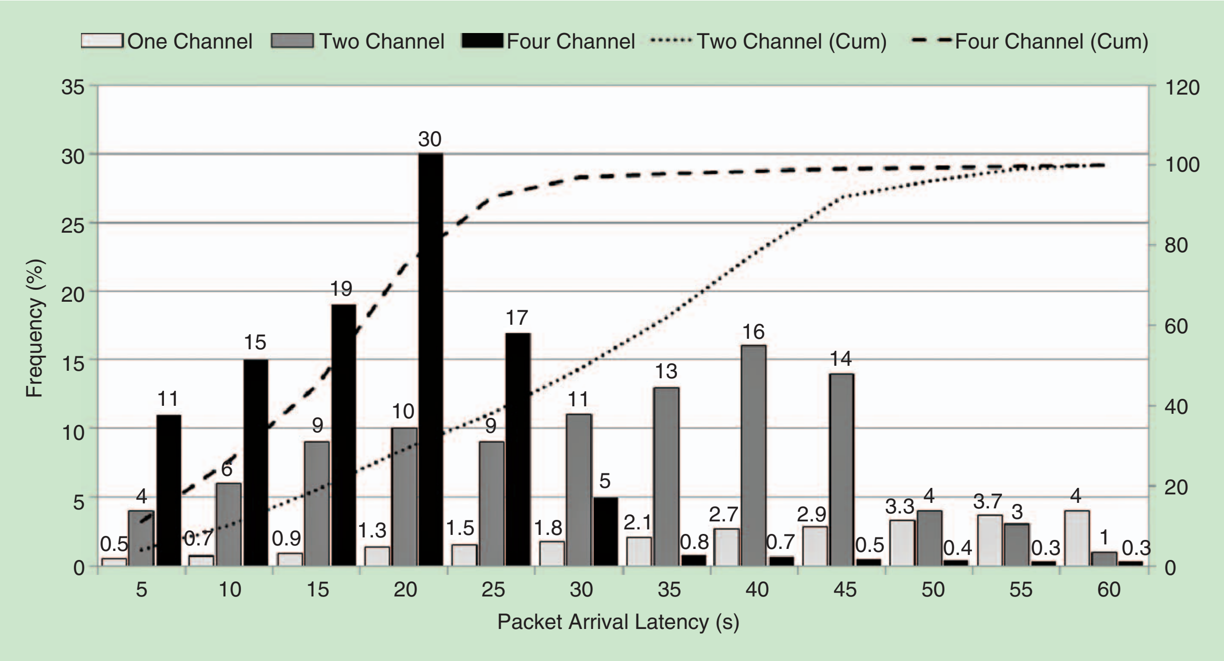


Figure 13. *Packet collection distribution using a GA for channel diversity. For four-channel diversity, 92% of the packets are received with a 25-s deadline.*

reliability, route utilization and analyzes that data in the form of a Fitness Function (10). A GA periodically reevaluates the emerging solution based on fitness criteria and extracts a new solution that replaces the existing solution. This results in incremental reconfiguration (channel reallocation) of WSN.

As described in the "Initialization: Characterization of Sensor Nodes" section, large number of nodes are within one-hop distance of each sensor. A GA synthesizes a solution that assists in maximizing the average single hop distance by optimally allocating nonadjacent channel to sensor nodes. Once the thinning process is concluded, Gateway allocates time-slots for interference free data collection for nodes that share same channel and are within one hop distance. Figure 13 illustrates the improvements in data collection latency due to GA assisted channel diversity. Sensor Network with four-Channel diversity achieves 92% packet receive ratio (PRR) in 25 seconds, compared to two-channel case which achieves the same PRR in 45 seconds. Due to optimal sensor-sensor distance, interference between neighboring sensors is minimized, resulting in lesser collisions and improved efficiency.

Although channel diversity evolves by observing interference patterns and the link quality, other perturbations resulting from network-congestion like packet delays and route-bottlenecks due to heterogeneous nature of sensor activity needs to be handled. Augmenting the channel fitness with the route fitness (F_R) improves the network topology by additionally rewarding the solution that creates balanced routes (see the section "Route Fitness: Synthesizing Balanced Routes"). In addition to optimal channel selection for lower interference and balanced routes, the (F_D) fitness feed-back of a GA improves the efficiency of route reconfiguration that results in producing large number of noninterfering subnetworks with almost identical latency characteristics. Figure 14 emphasizes the benefits of using subnetwork balancing criteria (F_D) within the GA fitness function as described in equation 10. This methodology achieves an average performance boost of 30.65% with a standard deviation of 4.3 over a wide and scalable range of sensor population. Figure 15 illustrates improvements of 20–48 % in the amount of lost and delayed packets for different sensor densities. This improvement can be attributed to reduced contention on the delivery routes resulting from channel allocation that rewards contention-free routes generation.

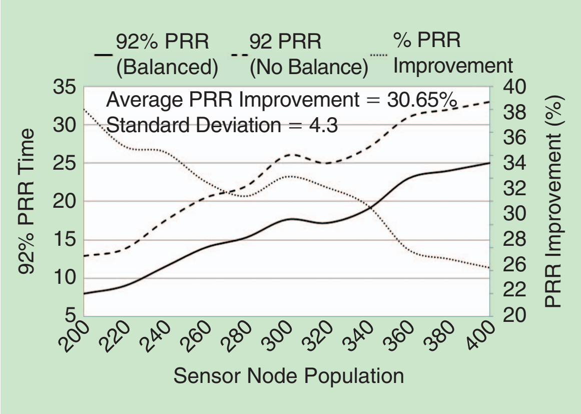


Figure 14. *A 92% PRR latency 1) using GA assisted balanced tree fitness criteria ($\alpha_4 = 0.2$) and 2) not using balanced tree criteria ($\alpha_4 = 0$) (10).*

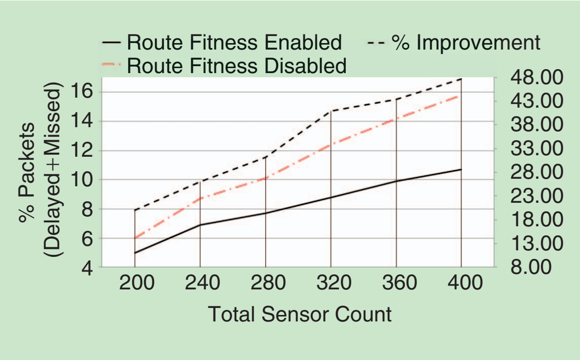


Figure 15. *The percent of improvement in lost and delayed packets if balanced route fitness (F_R) is enabled (compared to disabled). F_R influences the channel allocation for optimal delay.*

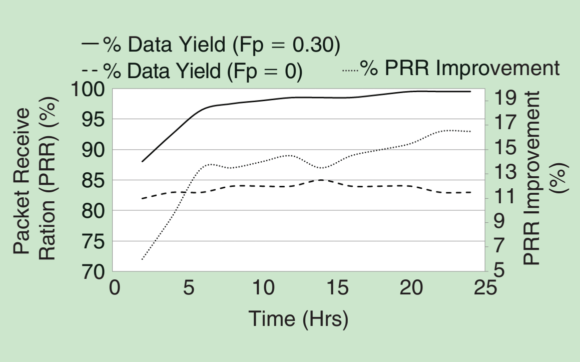


Figure 16. *Illustration of total data yield as PRR over a 24 hour collection period with 400 sensor nodes.*

The ultimate goal of a DC WSN is to collect sense-data packets with variable degrees of spatio-temporal correlation in a shortest period of time. Such variations results in variable length packets due to run-time compression. Figure 16 illustrates sense-data yield graph over a 24 hour period where Y-Axis represents PRR based on sampling period of 25 seconds. With route balancing and congestion control fitness (F_R) included in the cumulative fitness calculation (Equation 7), target PRR compliance (92% PRR over sampling period of 25 seconds) achieves an average of 97.125%. Ignoring this fitness component ($\lambda_2 = 0$) will degrade the PRR compliance by an average of 13.55%. Route fitness factor optimizes the delay paths by reconstructing routes to rebalance the overloaded nodes to meet delay requirements.

Conclusion

DCs face considerable challenges in seamless integration of telemetry and control functions. Real time monitoring and control of DC resources (cooling, power and workload resources) demands high fidelity monitoring of workload and environmental trends. WSNs present a nonintrusive and cheaper alternative to traditional wired networks. This article proposes an efficient approach to time-bound sense-data collection for DC by evolving WSN network topology using evolutionary algorithms. Evolutionary Learning Techniques (like the GA approach in this article) are capable of synthesizing parallel subnetworks by intelligently allocating noninterfering channels to sensor nodes. To improve data yield, channel allocation can be geared towards building noninterfering balanced routes such that each subnetwork within the WSN delivers uniform data collection timings with minimum contention and channel-interference. Machine learning assisted tools construct a fitness function that acts as a feedback loop to continually improve the solution over a finite period of time. Channel diversity through optimal channel allocation solution can deliver exponential performance boost to data collection through eliminating route congestion, parallelizing data collection and minimizing channel contention.

References

[1] I. Hong, D. Kirovski, G. Qu, M. Potkonjak, and M. B. Srivastava, "Power optimization of variable voltage core-based systems," in *Proc. 35th Annu. Design Automation Conf.*, San Francisco, CA, June 15–19, 1998, pp. 176–181.

[2] J. Nejedlo and R. Khanna, "Intel IBIST, the full vision realized," in *Proc. Int. Test Conf.*, Austin, TX, Nov. 1–6, 2009, pp. 1–11.

[3] R. Khanna, H. Liu, and H. H. Chen, "Self-organization of sensor networks using genetic algorithms," in *Proc. IEEE Int. Conf. Communications*, Istanbul, Turkey, June 2006, vol. 8, pp. 3377–3382.

[4] R. Khanna, H. Liu, and H. H. Chen, "Dynamic optimization of secure mobile sensor networks: A genetic algorithm," in *Proc. IEEE Int. Conf. Communications*, June 2007, pp. 3413–3418.

[5] D. Goldberg, *Genetic Algorithm in Search, Optimization and Machine Learning*. Boston, MA: Addison-Wesley, 1989.

[6] K. Langendoen and N. Reijers, "Distributed localization in wireless sensor networks: A quantitative comparison," *Comput. Netw.*, vol. 43, no. 4, pp. 499–518, Nov. 2003.

[7] D. S. J. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hopwireless routing," in *Proc. 9th Annu. Int. Conf. Mobile computing networking*, San Diego, CA, Sept. 2003, pp. 134–146.

[8] N. Burri, P. von Rickenbach, and R. Wattenhofer, "Dozer: Ultra-low power data gathering in sensor networks," in *Proc. 6th Int. Conf. Information processing sensor networks, ACM*, New York, 2007, pp. 450–459.

[9] S. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: A Tiny AGgregation service for ad-hoc sensor networks," in *Proc. 5th symp. ACM SIGOPS Operating Systems Review*, Dec. 2002, vol. 36, pp. 131–146.

[10] R.-E. Musaloiu, C.-J. M. Liang, and A. Terzis, "Koala: Ultra-low power data retrieval in wireless sensor networks," in *Proc. Int. Conf. Information Processing Sensor Networks*, St. Louis, MO, 2008, pp. 421–432.

[11] J. Liu, F. Zhao, J. O'Reilly, A. Souarez, M. Manos, C.-J. M. Liang, and A. Terzis. Project Genome: Wireless sensor network for data center cooling. [Online]. Available: http://msdn.microsoft.com/en-us/library/dd393313.aspx

[12] D. Ganesan, R. Cristescu, and B. Beferull-Lozano, "Power-efficient sensor placement and transmission structure for data gathering underdistortion constraints," *J. ACM Trans. Senor Netw.*, vol. 2, no. 2, pp. 155–181, May 2006.

[13] R. Khanna, H. Liu, and H. H. Chen, "Self-organization of wireless sensor network for autonomous control in an it server platform," in *Proc. IEEE Int. Conf. Communications*, Cape Town, South Africa, May 2010, pp. 1–5.

[14] *A True System-on-Chip Solution for 2.4-GHz IEEE 802.15.4 and ZigBee Applications*, IEEE Standard 802.15.4.

[15] J. A. Gutierrez, L. Winkel, E. H. Callaway, Jr., and R. L. Barrett, Jr., *Low-Rate Wireless Personal Area Networks: Enabling Wireless Sensors with IEEE 802.15. 4*. Piscataway, NJ: IEEE Standards Assoc., 2011.

[16] ZigBee Alliance, "Zigbee specification," ZigBee Document 053474r20, Revision 5, Sept. 12, 2012.

[17] ZigBee Alliance, "ZigBee RF4CE specification," ZigBee Document 094945r00ZB, Version 1.01, Jan. 2010.