This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2008 proceedings.

Proactive Power Optimization of Sensor Networks

Rahul Khanna*, Huaping Liu†, and Hsiao-Hwa Chen‡

*Intel Corporation, 2111 NE 25th Ave., Hillsboro, OR 97124,USA. E-mail: rahul.khanna@intel.com
†School of EECS, Oregon State University, Corvallis, OR 97331 USA. E-mail: hliu@eecs.oregonstate.edu, tel: +1 541 737 2973
‡Department of Engineering Science, National Cheng Kung University, Taiwan, E-mail: hshwchen@ieee.org

***Abstract*—We propose a reduced-complexity genetic algorithm for dynamic deployment of resource constrained multi-hop mobile sensor networks. The goal of this paper is to achieve optimal coverage and improved battery life using dynamic power scaling (DPS) and improved fitness function. DPS exploits idle times, packet delay guarantees, performance and workload data using additional controls related to sensor power states and transmission power. The dynamic power scaling in conjunction with genetic algorithm jointly optimizes power states and topologies by dynamically monitoring workloads, packet arrivals, utilization data and quality-of-service compliance. This results in minimization of the power consumption of the sensor system while maximizing the sensor objectives.**

I. Introduction

Low-cost integration and small-size micro-sensors [1]–[5] have generated significant interest in the area of disposable sensors. These are motion capable, randomly deployed, infrastructure-less, data-centric sensors equipped with data processing capabilities and sensory circuits that cannot be charged (or rarely charged) or replaced. These sensors are constrained in energy, bandwidth, storage, and processing-capabilities and find their uses in the areas of homeland-security, disaster-recovery, target-identification, reconnaissance, medical applications, defense applications [6], and intrusion-detection, etc. Each sensors process the sensory data and transmit to the target (sink) in a secure manner.

This paper extends previous work that used an evolutionary algorithm [7] to divide and position the randomly deployed mobile sensors into an optimal number of independent clusters with cluster-head and optimal route [8]. Once deployed, these sensors maximize their coverage by moving (or re-orienting) themselves at the expense of battery life and develop a long-lasting secure sensor network with variable security attributes [9]. Cluster-head collects data from its member sensors and sends them to the sink in a compressed and secure manner via the most cost-effective router. The energy dissipation of the sensor node is the sum of sensor transceiver and micro computations. As an extension to previous approach, we introduce *Dynamic Power Scaling (DPS)* and *Dynamic Transmission Scaling (DTS)* that uses the mix of proactive mechanisms and tuning parameters derived from workloads, security attributes and idle periods to optimize battery power.

Genetic Algorithm (GA) is a stochastic search technique that mimics the natural evolution proposed by Charles Darwin in 1858. GA has been successfully applied to a wide range of combination problems. They are particularly useful in applications involving design and optimization, where there are large numbers of variables and where procedural algorithms are either non-existent or extremely complicated.

Dynamic Voltage and Frequency Scaling (DVFS) [10], [12] is key technique in exploiting the hardware characteristics of processors to reduce energy dissipation by lowering the supply voltage and operating frequency. Since performance is needed only for a small fraction of the time, the DVFS algorithms enables energy savings while providing the peak computation power in general-purpose systems by optimizing performance and battery life. The proactive scheme predicts the future work requirements and sets up the power states according to the dynamic policy with parameters related to minimum work, and maximum deferment. These policies support data bursts, runtime constraints and optimal power states.

II. Related Work and Motivation

Mobile sensor networks consist of randomly deployed disposable sensors where configurable objectives cooperate with one another to maximize coverage and battery life. In this paper we use DPS and DTS in conjunction with four competing objectives [8], [9] that create an energy-efficient sensor network: (i) dynamic cluster membership, (ii) dynamic routing, (iii) dynamic sensor positioning, and (iv) dynamic sensor security attributes.

Related work includes dynamic voltage scaling (DVS) methodology that inserts additional information into the communication channel to guide the selection of proper voltages for data decryption (encryption) and processing in order to reduce the total computational energy consumption [11], real-time DVS (RT-DVS) that modifies the OS's real-time scheduler, and task management service to provide significant energy savings while maintaining real-time deadline guarantees [12], PowerTOSSIM that proposes efficient emulation of the sensor node hardware platform coupled with careful instrumentation of the power states which generates an event-driven simulator directly from TinyOS code and emits power state transitions [13], Bult *et al.* present advances in low-power systems spanning network design, through power management, low power mixed signal circuits, and highly integrated RF network interfaces [14], Xing *et al.* present that significant energy reduction can be achieved by jointly optimizing the transmission power and sleep time of nodes based on the network workload [15]. DVFS is an important power optimization feature in Intel and AMD class of micro-processors that provide multiple performance states using voltage and frequency scaling. ARM's Intelligent Energy Manager (IEM) voltage and frequency scaling reduces system-level power and energy consumption by as much as 15 to 20%.

III. Sensor Representation and GA Approach

In our previous work on sensor network optimization [8], each sensor node is allocated a functional assignment using using genetic algorithm. These functions are represented as (a) inactive node (powered off), (b) cluster-head (CH), (c) inter-cluster router (ICR), and (d) sensor node (NS). Each cluster is represented by a cluster-head, and cluster-members are represented by inactive/active node sensors and ICRs. Cluster-head performs data-fusion from various node-sensors while

inter-cluster router routes cluster data (from cluster-head) to the sink. Algorithmic details regarding clustering, naming, routing using GA can be found in [8]. Sensor Network implements a multi-objective genetic algorithm (MOGA) with the following fitness functions defined as follows [8], [9]:
(1) **Total Node Fitness (TNF)** forms weighted sum of Coverage Fitness (CF), Cluster-Head Fitness (CHF), Node Communication Fitness (NCF), Battery Status Fitness (BF), Router Load Fitness (RLF), and Sensor Effector Fitness (SEF)

$$\text{TNF} = \alpha_1\text{CHF} + \alpha_2\text{NCF} + \alpha_3\text{BF} + \alpha_4\text{RFL} + \alpha_5\text{SEF} + \alpha_6\text{CF} \quad (1)$$

(2) **Route Selection Fitness Function (RSFF)** generates balanced routes based on node allocation using GA based on node fitness function. During setup operation, both CH and ICR start sending data on the most cost effective routers.
(3) **Total Node Motion Fitness (TNMF)** is weighted sum of Coverage Uniformity Fitness (CUF), Cluster-Node Migration Fitness (CNMF), Cluster-Head Migration Fitness (CHMF), Node Motion Fitness (NMF) and Sensor Data Fitness (SDF). Fitness associated with node motion is given by

$$\text{TNMF} = \alpha_1\text{CUF} + \alpha_2\text{CNMF} + \alpha_3\text{NMF} + \alpha_4\text{CHMF} + \alpha_5\text{SDF} \quad (2)$$

(4) **Secure Node Fitness (SNF)** rewards energy efficient enablement of security attributes that are measured against battery quantization levels and rate of battery usage. While routes (CH→sink) are penalized for carrying malformed and retried packets, they are rewarded for enabling authentication on routers (ICR) and encryption on cluster-heads. Additional penalty is awarded if the authentication is enabled disproportional to threat level quantized to M levels.

IV. Power Scaling Approach

Power scaling optimizes the functional blocks of the sensors for a given Quality-of-Service (QoS) as perceived by the sink. QoS policy is defined as a function of (i) security attributes, (ii) importance and accuracy of sensor data (packet Priority), and (iii) maximum node-sink delay (sensor data, control data, etc.). QoS policies along with quantifiable observations (workloads, data arrival patterns, battery levels, node stability) adjusts the power-scaling parameters of the functional blocks of the node to enhance QoS compliance. For a given functional block, these parameters assume static conditions within a tunable observation period (T_{obs}). Following sections define various functional blocks and corresponding controls.

A. Memory Buffer Subsystem

Memory subsystem (Fig. 1) comprises of Transmit/Receive Buffer (TRB), Transmission Request Queue (TRQ), Performance Counters (PC), and Tunable Registers (TR). Memory buffer is divided into multiple blocks with independent power control applied according to anticipated demand. Data received by the receive buffer is processed for further action (authentication, header manipulation, etc.) before committing to the TRQ. Once in TRQ, data is transmitted to the next hop upon inactivity timer expiration or reaching burst threshold. Number of active memory blocks, observation timer, inactivity timer and burst thresholds are estimated based on QoS policies, sink feedback, and activity trends as measured by performance counters. Performance counters measure mean number of packet arrivals/serviced and IDC for packet arrivals [16] (index of dispersion for count). IDC is defined as the variance of the number of packet arrivals divided by the mean number of packet arrivals in an interval of length t

$$\text{IDC} = \text{var}\left(\sum_{k=0}^{n}\lambda_k\right) \Big/ E\left(\sum_{k=0}^{n}\lambda_k\right) \quad (3)$$

where λ_k is the number of packet arrivals between time interval τ_k and τ_{k+1}. It provides a measure of fluctuation of the receiving rate over a given interval, which reflects considerable burstness in the received packets. Burstness is a direct indication of packet loss and buffer occupancy.

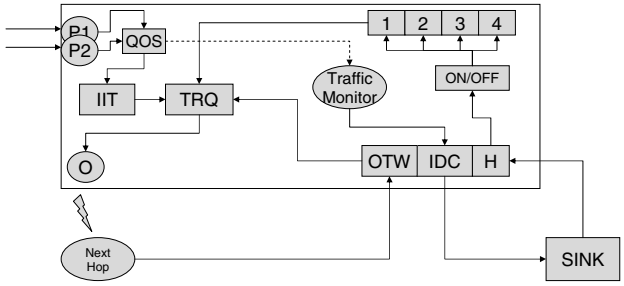

Fig. 1. Memory buffer subsystem: it contains four memory blocks that are activated based on buffer requirements per HURST (H) parameter (updated by sink). TRQ controls the number of fragments transmitted in a single burst based on IIT and QTW parameters. P_1 and P_2 represent input ports, O represents output port (next hop). IDC value is updated upon $P_1 + P_2$ traffic.

IDC slope is further used by sink to measure the HURST parameter (H) which is the measure of the persistence of a statistical phenomenon, or the measure of the long-range dependence of a stochastic process. Buffer requirements are much higher at lower levels of utilization for higher degrees of self-similarity (higher H). Sink updates the H value periodically which is then used by nodes to estimate the number of active memory blocks.

$$\text{Buffer (B)} = \rho^{1/[2(1-H)]}/(1-\rho)^{H/(1-H)} \quad (4)$$

where ρ is the utilization factor given as

$$\rho(\text{utilization}) = E\left(\sum_{k=0}^{n}\lambda_k\right) \Big/ E\left(\sum_{k=0}^{n}\mu_k\right) \quad (5)$$

where μ_k is the number of packets serviced between times τ_k and τ_{k+1}. Adequate buffer activation saves power by avoiding excessive allocation or switching between ON/OFF states.

Additionally, TRQ schedule transmit requests according to maximum transmission rate, Optimal Transmission Window (OTW) (burst size), and QoS requirements relative to delay tolerances. For burst size less than the OTW, TRQ defers the transmit request for a duration equal to the inactivity interval threshold (η_k^j) for n^{th} node at t^{th} time. IIT is programmed according to QoS requirements of the sensor data.

$$\beta_i^j(x) = \max\left(-1, \min\left(1, \frac{T_i^j(x) - D_i^j(x)}{D_i^j(x)}\right)\right) \quad (6)$$

$$\eta_t^j = \eta_{t-1}^j + \frac{\eta_{\min}}{N}\sum_i \beta_i^j(\text{mem}) \quad (7)$$

where $D_i^j(\text{mem})$ and $T_i^j(\text{mem})$ represent the expected delay and measured delay, respectively, for i-th packet (route) on j-th node. While $D_i^j(\text{mem})$ is updated using sink's QoS feedback (Section V-A), OTW is updated as a result of control message sent by target node.

B. Micro Controller (μC) Subsystem

Micro controller is an integral component of the sensor node. In this section we consider the micro-controller and its effect on sensor node power consumption. We review the factors influencing the power consumption and calculate the expected performance based on tunable parameters. Micro controllers are optimized for functions related to:
(1) *Message Handling* - Operations related to message parsing, data fragmentation, handling TRQ, data/header manipulation, and buffer management heuristics.
(2) *Security Protocols* - Operations related to data encryption/decryption and authentication protocols to support confidentiality, tamper protection, authenticity, replay prevention. The computation load (and hence execution time and energy consumption) for encryption and decryption provides an opportunity to optimize the power states.
(3) *Event Handling* - Operations related to events that are routed to a PIN assertions or triggers due to threshold crossings, periodic timers or hysteresis effects.
(4) *Performance Monitoring* - Operations related to synthesis of performance data specific to the sub-system it is monitored for. Performance data is polled at optimal sampling granularity subject to sampling variances.

Processor's dynamic power dissipation is proportional to capacitance, clock frequency, and the square of supply voltage ($P \propto C_L.V_{dd}^2.f$). This implies that to accumulate the same amount of computation, using lower voltage will consume less energy in longer time because the power level is much lower [11]. We use discrete performance states using voltage scaling represented by P_i, where i is the state number. For X discrete states, P_0 is the highest-power/least-latency state, whereas P_X is the performance state.

Since, upon packet arrival, we are uncertain about computational requirements, any unused cycles allotted would eventually be wasted due to idling for extra processor cycles. DVS algorithm avoids wasting cycles by reducing the operating frequency and ensuring that deadline guarantees are not violated by doing so [12]. Based on the QoS requirements of each message and its respective security attributes, P-State ceiling is set for the period of tunable interval (T_{obs}). The ceiling is adjusted according to the delay targets of the ICR or CH nodes.

$$P_t^j = min(X, P_{t-1}^j) + (\gamma/N) \sum_i \beta_i^j(\text{cpu}) \tag{8}$$

where X is the maximum number of discrete performance states (P-States), γ is the scaling factor ranging between 1 and 2 and β_i^j(cpu) is the QoS compliance factor. CPU state of node j is incremented or decremented according to QoS compliance of all CH packets passing through it.

C. Wireless Link Subsystem

Transmission time decreases as direct consequence of increasing bit-rate which, without increasing the data transmission decreases the radio duty-cycle. But high turn-on-to-receive exit latency can make it impossible to achieve the required duty-cycle ($< 1\%$). Furthermore, when the radio switches from sleep mode to transmit mode to send a packet, a significant amount of power is consumed for starting up the transmitter itself [17]. Therefore optimal tuning is needed to avoid reactive response to an idle slot during transmission. Power savings are realized by running micro-controller at the optimal P-State and radio at optimal frequency which spreads the computations in time and transmits the data in a quick burst. This requires decoupling between computational and transmission rate, where each can run at its optimal point using rate-matching between computational processing and data transmission. Since the instantaneous traffic load is mostly lower than the peak value, transmissions can be slowed down, to the optimal operating point. Similar to DVS [Section IV-B], which uses voltage adaptation for an effective CPU power management, Dynamic Modulation Scaling adapts the modulation level to match the instantaneous traffic.

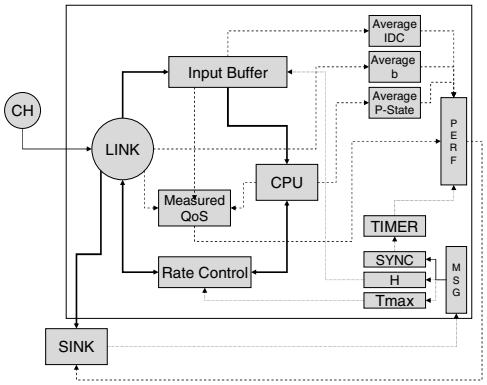

Fig. 2. Performance counters monitor CPU power, modulation power and QoS compliance. It transmits the performance data upon timer trigger. Sink uses this data to generate performance targets for nodes. Rate controller uses this data to distribute delay targets to CPU, link and buffer control.

In this paper we will use M-QAM as the adaptive multilevel modulation scheme because of its lower symbol error probability and bit energy consumption than M-PSK for a given SNR. Sensor node adjusts constellation size (b) and symbol rate (R_s) to reduce the overall energy. Schurgers *et. al* [18] define the following expression for minimizing energy required to transmit one bit by choosing the correct values of b and R_s:

$$E_{\text{bit}} = \left[C_S \cdot (2^b - 1) + C_E + C_R \cdot \frac{R_{S\_\max}}{R_S} \right] \frac{1}{b} \tag{9}$$

where C_E and C_R are functional components that incorporate electronic circuitry for filtering, up-converting and modulating. Parts of the circuitry operates at frequencies proportional to instantaneous symbol rate (R_S), while other parts operate at frequencies proportional to maximum symbol rate ($R_{S\_\max}$). C_S represents the function of target performance that is weakly dependent upon b. Since high value of R_S results in minimizing delay and energy transmitted per bit, it is logical to maximize this value to $R_{S\_\max}$. Hence, the constellation size b is the only option required to trade off energy versus delay. b_t^i is initially set to the maximum for the i-th node at time t and tuned dynamically according to the sink feedback and QoS targets. Similar to η [eq. (6)], this parameter is tuned as a function of target and measured QoS.

V. Parameter Tuning and Fitness Function

In this section we will discuss the aspects of coordinated tuning and fitness function (Fig. 2) that uses the parameters defined in Section IV. While tuning is necessary to maximize the performance/energy ratio of given set of nodes for a given topology (clustering, routing, etc.) and QoS, fitness function is required to optimize the sensor network topology to achieve the target performance/energy ratio for the entire network with optimal battery utilization. GA utilizes the optimal operating

points and performance feedback of the nodes of the instantaneous topology to calculate the output of the fitness function which influences the cluster formation/placement, membership, functional attributes, and routing decisions [8].

A. Coordinated Tuning

Coordinated tuning is necessary for identifying the operating point in order to maximize the performance of the node with respect to energy consumed. Performance is measured using a QoS function that is dependent upon communication delay, messaging priority, etc. Router node j estimates the QoS error for cluster head packet i (ξ_i^j) using:

$$\xi_i^j = \frac{T_i^j - \max(\epsilon, (1 - Qp_i)) \cdot T_{i_{\max}}^j}{\max(\epsilon, (1 - Qp_i)) \cdot T_{i_{\max}}^j}, \quad 0 \leq Qp_i \leq 1 \quad (10)$$

where T_i^j represents the processing delay for packet i on ICR j, Qp_i is the priority of i-th packet, and $T_{i_{\max}}^j$ is the maximum delay allowed for i-th packet on j-th ICR. Periodically, sink sends the SYNC-sink message containing new parameters ($T_{i_{\max}}^j$, Qp_i and Hurst (H)) for all member nodes $(R(i))$ of route catering CH i. Each ICR and CH node implements the closed-loop-control function that minimizes the ξ_i^j by tuning inactivity interval threshold (η^j) [eq. (6)], CPU P-state ceiling (P^j) [eq. (8)] and modulation scaling parameter (b^j) [eq. (9)]. Each of these components contribute partially to the maximum allowed delay $T_{i\max}^j$ thereby operate within its QoS bounds. QoS contribution of each tunable component can be expressed using eq. (12):

$$D_i^j(x) = \max(\epsilon, (1 - Qp_i)).T_{i_{\max}}^j(x) \quad (11)$$

$$\xi_i^j(x) = (T_i^j(x) - D_i^j(x))/D_i^j(x) \quad (12)$$

$$T_{i_{\max}}^j(x) = \phi_x T_{i_{\max}}^j \quad (13)$$

where ϕ_x is the delay contribution factor due to processing element x =(Memory, CPU, Link). Each processing element measures QoS compliance by calculating the differential between measured ($T_i^j(x)$) and expected delay ($D_i^j(x)$). It scales the parameters accordingly to reduce the QoS error.

While early arrivals can cause a bursty traffic and memory pressures, late arrivals can cause performance issues. To avoid that situation, ϕ_x is tuned by monitoring the transmit buffer for a period of programmed interval (T_{obs}). For high bandwidth case we limit the transmit buffer utilization to 70%. This is done by first reducing the modulation delay factor ϕ_b to minimum bounds, then increasing the CPU delay factor ϕ_{cpu} and finally increasing memory delay factor ϕ_{IIT}. For low utilization case, we reduce the CPU delay factor first, followed by modulation delay factor and finally memory.

B. Power Scaling Fitness Function (PSFF)

In this section we define a new fitness function that rewards the uniformity of the power states and QoS compliance within an established route. This function also penalizes a disproportionate allocation of power-states with respect to other routes as well as non-optimal provisioning of memory buffer. Elements of fitness function are described as follows: (1) **QoS Fitness (ω_i^1)** is defined as degree to which routes are compliant with respect to allocated delay budget. A route i is considered compliant (β_i) if the measured delay (D_i) falls within $\pm\delta\%$ of target delay (T_i). Over-compliance as well as under-compliance are both penalized though with a non-uniform penalty factor ($\widetilde{\lambda}$, λ).

$$\omega_i^1 = 1 - \frac{\sum_j (\widetilde{\lambda}|\max(0, \beta_i^j - \delta)| + \lambda|\min(0, \beta_i^j + \delta)|)}{N}. \quad (14)$$

This parameter reflects the burst variability of multiple arrivals multiplexed locally. Variability exists due to variable message sizes, arrival rates, priorities and non-uniform compute requirements per security attributes [9].

(2) **Buffer Optimization (ω_i^2)** is the measure of effective memory utilization. Packets from different sources (CH and ICR) arrive at different times and are statistically multiplexed into the common buffer. These packets can have variable rates depending upon sampling variances and QoS. Buffer occupancy is dependent upon service times of the CPU and wireless subsystem which is optimized as a part of tuning process. It penalizes non-optimal provisioning of the memory blocks that can cause over allocation or reactive switching between memory power-states. Furthermore, inadequate buffer between CPU and transmit logic can cause coupling between CPU processing and transmission rates.

$$\omega_i^2 = 1 - \frac{\sum_j \min(1, |B_{i\_\text{prid}}^j / B_{i\_\text{act}}^j - 1|)}{N}. \quad (15)$$

(3) **Uniform Power-State Distribution (ω_i^3)** penalizes the routes that consume disproportionate amount of power as compared to average power utilization by other routes. It uses average P-state residency ($\overline{P_i^j}$) [eq. (8)] and average modulation scaling ($\overline{b_i^j}$) [eq. (9)] as the measure of power consumption by a node i. As described above, one of the many reasons for non-uniform distribution is heterogeneous message priorities, variable rates and security attributes.

$$\omega_i^3 = 1 - \frac{\sum_j \min\left(2, \left(|\overline{P} - \overline{P_i^j}| + |\overline{b} - \overline{b_i^j}|\right)\right)}{2N}. \quad (16)$$

Overall PSFF fitness function of route i is the weighted sum of all contributing elements. While TNF (Section III) [8] incorporates communication energy as a part of NCF, it lacks the energy contribution due to other components like microcontroller and wireless logic. We modify that equation by adding PSFF contribution:

$$\text{PSFF}_i = \mu_1\omega_i^1 + \mu_2\omega_i^2 + \mu_3\omega_i^3 \quad (17)$$

$$\text{TNF} = \alpha_1\text{CHF} + \alpha_2\text{NCF} + \alpha_3\text{BF} + \alpha_4\text{RFL} + \alpha_5\text{SEF} + \alpha_6\text{CF} + \alpha_7\text{PSFF} \quad (18)$$

where $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + \alpha_5 + \alpha_6 + \alpha_7 = 1$ and α_i depends upon the relative significance of the component. These values can be made adaptive using an external heuristics. Details of GA is described in [8] [9].

VI. Results And Discussion

Experimental setup consists of 100 nodes at random positions in a 30×30 space. Individual node picks up a random coordinate between $(0, 0)$ and $(30, 30)$ and assigns itself an UUID and a random battery capacity between 0 and 15. For simplicity, each node is given a coverage area of 3×3 and assumes line-of-sight propagation. After nodes placement in the listen mode, GA is run with the cross-over rate of 60% and an initial mutation of 6%. The software simulates the sink operation in conjunction with NS-2 software that simulates the network traffic. It executes the GA that generates re-clustering/re-assignment tasks. It

also calculates the fitness parameters based on network traffic, battery usage and power-performance parameters (PSFF). A separate process in the sink simulator runs a predictive algorithm that estimates the traffic and data patterns (sampling rates, data Redundancies, Self-Similarity etc.) into the future using past hysteresis. This is in conjunction with the closed-loop self-optimization of the nodes itself that run optimization heuristics to distribute the delay budget between buffer, CPU and wireless link infrastructure using route-feedback (H, delay budget) from sink. While each GA objective tends to compete with others to converge at the system equilibrium, the end result is to maximize the network life for optimal coverage. The experimental scheme involves local optimizers that tunes themselves and GA optimizers that optimizes the complete topology and node assignments. Guaranteed delays are maintained by adjusting delay times between CPU and wireless link (Section V-A).

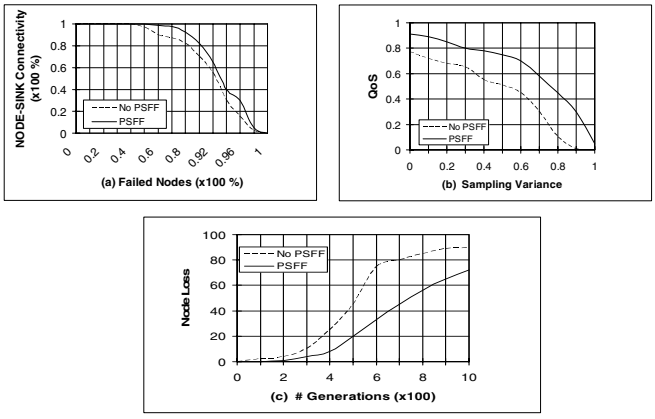

Fig. 3. (a) Percentage of nodes connected to sink in the event of node failure; (b) QoS measure (of Route 1) as a function of variable sampling rates of the neighboring clusters; (c) Percentage of nodes lost (due to battery) as a function of n-th generation.

In a power scaling case (PSFF), delay guarantees benefit between 10-20% over non-PSFF case (Fig. 3b). Due of high sampling variances, we incur frequent buffer-overflows or empty buffer queues in non-PSFF case. Furthermore, because high power-states remain static (due to reactive ON/OFF) for longer duration, it processes the traffic faster that make it burstier towards the sink and causes packet drops. PSFF not only improves the QoS guarantees, but also reduces average power consumption/node by about 27% due to proactive state determination between CPU and Wireless Link subsystem. Fig. 3(c) shows the node loss versus of the number of generations. It is found the Power-Scaling (PSFF) case significantly outperforms the static one with about 25% at 600-th generation. On an average it shows about 12-15% reduction in the number of nodes lost. Main power savings are realized due to voltage and modulation scaling for given delay guarantees (QoS) as well as re-clustering triggers due to disproportionate allocation of packet priorities, multiplexing of heterogeneous traffic patterns with variable rates and security attributes. Fig. 3(a) shows 5-8% improvement in the number of nodes with a valid route to sink because residual energy saved (using PSFF and local power scaling) promotes nodes to act as ICR(s).

VII. Conclusion

We presented dynamic energy-efficient sensor deployment using a multiple-objective genetic algorithm in conjunction with local tuning of CPU, memory and wireless link power states. This approach maximizes coverage, quality-of-service and network life by exploiting dynamic power scaling methods and re-clustering based on those methods. We observe incremental improvement over traditional approach [8] due to proactive mechanisms that predicts the future power states and buffer requirements to achieve the projected delay guarantees (guaranteed QoS). As a result of power conservation due to scaling, we also observe reduced clustering triggers as lesser nodes reach there functional thresholds. In a multi-hop network, heterogeneous traffic flows are multiplexed into an ICR which causes variations in the resource consumption. Re-clustering with power scaling fitness function bias reduce such variations and enhances the QoS compliance (up to 20%) with guaranteed delays. PSFF also prevents GA to converge to a local optimum due to uniform load requirement by biasing PSFF to about 25%. Future work include tuning the optimal bias for various components of TNF [eq. (17)] and an ability to optimize delay guarantees to maximize the network life.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks*, vol. 38, pp. 393–422, Mar. 2002.

[2] K. Akkaya, M. Younis, "A Survey on routing protocols for wirelsss sensor networks," *Ad Hoc Networks*, vol. 3, pp. 325–349, May. 2005.

[3] R. Min, M. Bhardwaj, S.-H. Cho, E. Shih, A. Sinha, A. Wang, and A. Chandrakasan, "Low power wireless sensor networks," in *Proc. Int. Conf. VLSI Design*, Bangalore, India, Jan. 2001.

[4] J. M. Rabaey, M. J. Ammer, J. L. da Silva, Jr., D. Patel, and S. Roundy, "PicoRadio supports ad hoc ultra low power wireless networking," *IEEE Computer*, vol. 33, pp. 42–48, July 2000.

[5] R. H. Katz, J. M. Kahn, and K. S. J. Pister, "Mobile networking for smart dust," in *Proc. 5th ACM/IEEE MobiCom*, Seattle, WA, Aug. 1999.

[6] H. O. Marcy, *et al.* "Wireless sensor networks for area monitoring and integrated vehicle health management applications," in *Proc. AIAA Conf. Guidance, Navigation, and Control*, Portland, OR, USA, Aug. 1999.

[7] D. Goldberg, *Genetic algorithm in search, optimization and machine learning*, Addison-Wesley Publishing Company, Inc., 1989.

[8] R. Khanna, H. Liu and H. H. Chen,"Self-organization of sensor networks using genetic algorithms," in *Proc. IEEE ICC*, Istanbul, June 2006.

[9] R. Khanna, H. Liu, H. H. Chen, "Dynamic optimization of secure mobile sensor networks: a genetic algorithm," in *Proc. IEEE ICC'07*, Jun. 24-28, 2007.

[10] K. Govil, E. Chan, H. Wassermann, "Comparing algorithms for dynamic speed-setting of a low-power CPU," in *Proc. Int. Conf. MobiCom'95*, Berkeley, California, USA, Nov. 13-15, 1995.

[11] L. Yuan and G. Qu, "Design space exploration for energy-efficient secure sensor network," in *Proc. IEEE Int. Conf. Application-Specific Systems, Architectures, and Processors*, Jul. 17-19, 2002.

[12] P. Pillai and K. G. Shin, "Real-time dynamic voltage scaling for low-power embedded operating systems," in *Proc. 18th ACM Symp. Operating Systems Principles*, Banff, Alberta, Canada, Oct. 21-24, 2001.

[13] V. Shnayder, M. Hempstead, B. Chen, G. W. Allen, M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in *Proc. Int. Conf. Embedded Networked Sensor Systems*, Baltimore, MD, USA, Nov. 3-5, 2004.

[14] K. Bult, A. Burstein, *et al.*, "Low power systems for wireless microsensors," in *Proc. 1996 Int. Symp. Low Power Electronics and Design*, Monterey, California, USA, Aug. 12-14, 1996.

[15] G. Xing, C. Lu, Y. Zhang, Q. Huang, and R. Pless, "Minimum power configuration in wireless sensor networks," in *Proc. 6th ACM Int. Symp. Mobile Ad Hoc Networking and Computing*, Urbana-Champaign, IL, USA, May 25-27, 2005.

[16] Gusella R., "Characterizing the variability of arrival processes with indexes of dispersion," *IEEE J. Select. Areas Commun.*, vol.9, no.2, pp. 203-211, Feb. 1991.

[17] Wang A., Cho S., Sodini C., and Chandrakasan A., "Energy-efficient modulation and MAC for asymmetric microsensor systems," in *Proc. ISLPED, 2001*.

[18] C. Schurgers, O. Aberthorne, and M. B. Srivastava, "Modulation scaling for energy aware communication systems," in *Proc. Int. Symp. Low Power Electronics and Design*, Huntington Beach, CA, USA, 2001, pp. 96-99.