Optimal Sampling Rate Assignment with Dynamic Route Selection for Real-Time Wireless Sensor Networks

Weihuan Shu*, Xue Liu*, Zonghua Gu† and Sathish Gopalakrishnan‡
*School of Computer Science, McGill University, Montreal, Canada
†Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong, China
‡Department of Electrical and Computer Engineering, University of British Columbia, Vancouver, Canada

***Abstract*—The allocation of computation and communication resources in a manner that optimizes aggregate system performance is a crucial aspect of system management. Wireless sensor network poses new challenges due to the resource constraints and real-time requirements. Existing work has dealt with the real-time sampling rate assignment problem, under single processor case and network case with static routing environment. For wireless sensor networks, in order to achieve better overall network performance, routing should be considered together with the rate assignments of individual flows. In this paper, we address the problem of optimizing sampling rates with dynamic route selection for wireless sensor networks. We model the problem as a constrained optimization problem and solve it under the Network Utility Maximization framework. Based on the primal-dual method and dual decomposition technique, we design a distributed algorithm that achieves the optimal global network utility considering both dynamic route decision and rate assignment. Extensive simulations have been conducted to demonstrate the efficiency and efficacy of our proposed solutions.**

I. Introduction and Related Work

A Wireless Sensor Network (WSN) is a network consisting of spatially distributed smart sensors cooperatively monitoring physical (or environmental) conditions and connected to each other via wireless links. WSNs have many military and civilian applications including battlefield surveillance, environment and habitat monitoring, healthcare, home automation, and traffic control. Many of those applications have real-time requirements. A Real-Time Wireless Sensor Network (RTWSN) is a wireless sensor network that can make real-time guarantees. In a RTWSN, the sampling rate is a very important parameter, as it is closely related to the Quality of Service (QoS) of applications. For example, in a RTWSN used for video surveillance, the sampling rate refers to how many frames a video source captures and sends out to the monitoring center per second. For most applications, the higher the sampling rate is, the better the QoS becomes. However, typical WSNs usually face many practical constraints, e.g., computation limit of the sensors, bandwidth of the routes and delay of the network, which restrict the achievable sampling rates. How to allocate system resources in a way to maximize the aggregate performance of the network subject to these constraints is an important research topic for RTWSN. In this paper, we address the problem of real-time sampling rate allocation with dynamic route selection, with the goal of optimizing the global network performance while maintaining real-time schedulability in RTWSN.

Resource allocation has been an active research area for computing systems [18], [27], [2]. Most of them do not take real-time requirements into consideration and hence cannot be directly applied to real-time systems. Kelly and Low et al. first studied the problem of resource allocation for congestion control in computer networks [17], [16], [22]. These work formed the foundation of Network Utility Maximization (NUM), but again the real-time constraints were not considered. Later, researchers extended the work into resource allocation in wireless networks in general [8], [6] and in WSNs in particular [10].

Finding the optimal task execution rates subject to the schedulability constraints was first studied by Seto et al. and by Sha et al. for analog and digital controllers, respectively [24], [26]. They presented offline optimization techniques based on the Kuhn-Tucker conditions, but the schedulability constraint considered is only for a single processor. Rajkumar et al. developed QoS-based Resource Allocation Model (Q-RAM), which is capable of handling multiple quality dimensions [23], but the solution can only be used in a single constraint case. Lee and Ghosh et al. studied the scenario under multiple constraints [19], [11], but the problem they addressed is an integer programming problem, which is different from the problem discussed in this paper. In [19], the integer programming problem proved to be NP-hard, and several sub-optimal algorithms were proposed. According to [11], Hierarchical Q-RAM is the technique with the best scalability. However, that algorithm requires the division of multiple constraints into independent groups, which is impractical for multi-hop RTWSN. Lately, the work of Chen et al. made it possible to achieve the maximum system utility [7], but it considered discrete task rates, and the employed model as well as the focused problem are different from ours.

RTWSN presents new challenges for real-time resource allocation. Since routes in a RTWSN may intersect with each other at the routers, sampling rate optimization for real-time flows must take into consideration the traffic contention at each router. Liu et al. first transformed the real-time sampling rate assignment problem in a WSN to a constrained optimization problem [21], which explicitly captures the real-time requirements of the WSN as optimization constraints. They also proposed a distributed algorithm based on the Internet pricing schemes [22]. However, this work assumed that packet routing decision is made independent of rate selection, and routes stay unchanged during the process of sampling rate optimization. As we will show later in Subsection VI-A, this assumption of static route selection may limit the global network performance, which is also referred to as *network utility*.

Recently, based on the NUM framework, extensive research has been conducted towards a systematic understanding of "layering" as "optimization decomposition", where the overall communication network is modeled by a generalized NUM problem: each layer corresponds to a decomposed sub-problem, and the interfaces among layers are quantified as functions of optimization variables coordinating the sub-problems [8]. Chen, Lin, Wang, and He et al. studied the problem of joint optimization of congestion control and routing [6], [20], [28], [12]. Their approaches showed how a joint optimization problem can be decoupled and separated into different network layers.

In this paper, we systematically study the problem of optimal sampling rate assignment together with dynamic route selection for real-time wireless sensor networks. In contrast to the work by Liu et al. [21] where static routing is assumed, we allow *dynamic routing*. In our model, each sensor source has one or more paths leading to its corresponding destination (data sink), but only one path at a time is selected for data transmission. The set of candidate paths between a source and a destination can be chosen offline based on existing routing algorithms for wireless sensor networks such as SPIN [14], GPSR [15], GEAR [29], Rumor Routing [3], SPEED [13] or RPAR [9]. Instead of using the "optimal" route determined by a specific routing algorithm, we keep all the feasible ones as candidate routes according to application requirements and select route to maximize the overall network utility.

We first show that the data transmission scheme employed in [21] is not efficient when the data blocks to be transmitted are relatively large, then propose a new scheme that can enhance the network utility. The new scheme also facilitates schedulability analysis for each router, as well as the implementation of the distributed algorithm. The optimization problem with dynamic route selection is then formulated and transformed into an optimization problem with nonlinear objective function and linear constraints. Finally, a distributed algorithm based on the primal-dual method and dual decomposition technique will be given for the joint optimization problem. The algorithm is able to find the optimal sampling rates and the optimal routing, while maintaining the real-time schedulability in a dynamic routing environment.

The rest of this paper is organized as follows. Section II introduces the system architecture for multi-hop RTWSN used in this paper. In Section III, we give the real-time schedulability analysis for the system, and show how to improve the utilization of routers comparing to the data transmission scheme used in previous work. The improvement on utilization of individual routers increases network throughput, and this in turn leads to better network utility. In Section IV, we model the optimal sampling rate assignment with dynamic route selection problem as a nonlinear optimization problem. The centralized algorithm and the distributed algorithm are both given in Section V. In Section VI, we first show that the distributed algorithm is efficient by analyzing the simulation results on convergence, and then prove that our proposed data transmission scheme is better by comparing it with the original scheme in [21]. Finally, Section VII presents conclusions and future work.

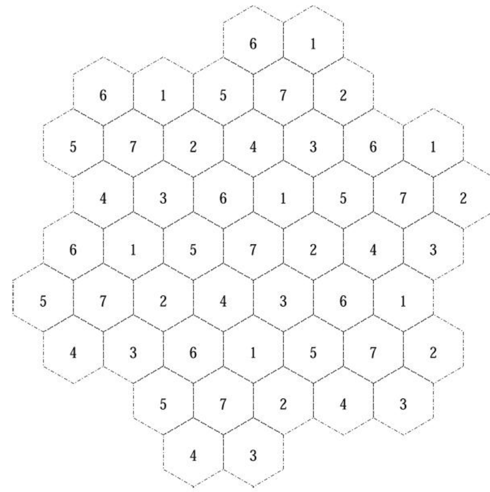

Fig. 1. The Mixed FDMA-TDMA Base Station Backbone Layout

II. System Architecture

A. RICH Architecture

Caccamo et al. first provided real-time support for multi-hop RTWSN [5], where a cellular base station layout is deployed as the backbone for the underlying RTWSN, as shown in Fig. 1. In this architecture, each base station functions as a router at the center of each cell. The base stations use seven non-overlapping Radio Frequency (RF) bands, and all RF broadcasts are one-hop. The inter-cell communication in the wireless sensor network uses a globally synchronized TDMA scheme, where a period is divided into six slots, each corresponding to the data transmission towards one of the six directions. Therefore, the inter-base-station communication is a mixed FDMA-TDMA scheme.

Based on the cellular base station backbone layout in [5], Liu et al. proposed the Real-time Independent CHannels (RICH) architecture [21]. RICH architecture employs the mixed FDMA-CDMA scheme instead of the mixed FDMA-TDMA scheme [5], in order to achieve better flexibility and simpler schedulability analysis. Fig. 2 shows the internal architecture of the RICH base station. Due to the employment of DSSS-CDMA technique, transmissions can be carried out independently, and there is no synchronization requirement needed between any pair of transmissions. Furthermore, the

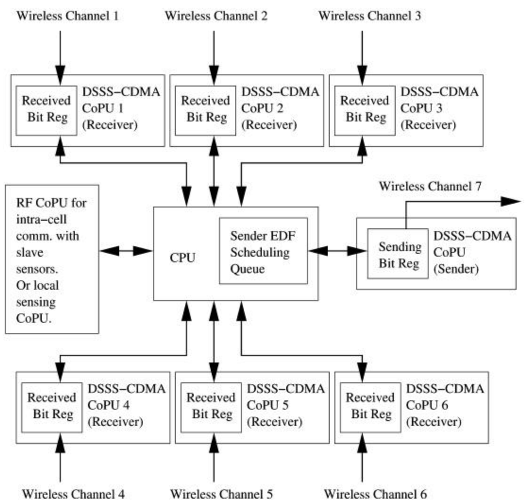

Fig. 2. The Internal Architecture of a RICH Base Station

scheduling can be independently adjusted to be specific to each base station.

In this paper, we adopt the RICH architecture for RTWSN deployment. Note that under the RICH architecture, there can be multiple wireless sensors (slaves) deployed inside each cell. These sensors usually perform the actual sensing and also communicate with the base station (head sensor) of the corresponding cell at different RF bands that do not interfere with the inter-cell communications. This paper only focuses on the inter-cell communication. Since intra-cell communication is local to the cell, it is not the focus of this paper, and is therefore not addressed explicitly.

B. Schedulability Modeling

For a given RICH base station n, the available bandwidths to its six neighboring RICH base stations are $B_n^1, B_n^2, \ldots, B_n^6$ respectively, which may be different from each other due to irregularity of the wireless medium [30]. We set the data transmission bandwidth of base station n to be $B_n = \min_i B_n^i, 1 \leq i \leq 6$, hence the broadcast of n can be reliably received by all its six neighbors. In the rest of this paper, we use *node* to refer to the base station that acts as a router, and *source* to refer to the base station that has data originated within the corresponding cell. Note that a source is also a router, since it also forwards data received from the sensors within the cell.

Let $\mathcal{S}$ be the set of sources in the network, and let $\mathcal{S}_n$ be the set of sources for which node n forwards data. Assume a source $s \in \mathcal{S}_n$ has a sampling/reporting rate of f_s, and each report packet has a length of l_s. The corresponding transmission time for a packet from source s on node n is therefore $c_s = l_s/B_n$. As the sampling/reporting rate is fixed, the data transmission for $\mathcal{S}_n$ on node n can be viewed as a periodic task set, and existing scheduling algorithms can be used to determine the schedulability, such as Rate Monotonic (RM) and Earliest Deadline First (EDF) scheduling algorithms. In this paper, we choose non-preemptive EDF scheduling algorithm because of the simple analysis and the fact that packet transmissions are usually non-preemptive. The detailed schedulability analysis is given in the next section.

III. Schedulability Analysis

A. Preemptive EDF

EDF is a dynamic priority scheduling algorithm that always selects the task with the shortest absolute deadline to execute, in other words, the task with the earliest deadline has highest priority. If all tasks are periodic, preemptive, and have deadlines equal to their periods, EDF is able to achieve 100% processor utilization. Since $f(frequency) \times T(period) = 1$, we can write the schedulability condition as

$$\sum_{\tau \in \mathcal{T}} c_\tau f_\tau \leq 1,$$

where τ is a periodic task in task set $\mathcal{T}$, c_τ is the computation time of task τ and f_τ is the frequency of task τ.

B. Non-preemptive EDF

Although preemptive EDF scheduling is optimal, data transmission is a task that cannot be interrupted once it begins, so non-preemptive scheduling must be used for practical applications. A network data packet typically consists of a header section with the sequence number, type and routing information, and a data section with the actual data payload. It is well-known that transmission of a data packet should be atomic and not be interrupted in the middle, since missing any bits from the header section will influence data transmission and missing any bits from the data section will violate data integrity. The schedulability condition for non-preemptive EDF [4] is

$$\sum_{\tau \in \mathcal{T}} c_\tau f_\tau + C_i f_i \leq 1, \text{ for each } i \in \mathcal{T}, \tag{1}$$

where $C_i = \max_{\{\tau \in \mathcal{T} \text{ and } \tau \neq i\}}\{c_\tau\}$ is the maximum blocking time for the task i.

C. EDF Scheduler for Data Transmission

The constraint (1) for node n can be transformed into

$$\sum_{s \in \mathcal{S}_n} \frac{l_s}{B_n} f_s + \frac{L_i}{B_n} f_i \leq 1, \text{ for each } i \in \mathcal{S}_n,$$

where l_s is the packet length for source s and $L_i = \max_{\{s \in \mathcal{S}_n \text{ and } s \neq i\}}\{l_s\}$ is the maximum packet length among all the packets that may block the data transmission for source i. Then we have the following schedulability condition for data transmission:

$$\sum_{s \in \mathcal{S}_n} l_s f_s + L_i f_i \leq B_n \text{, for each } i \in \mathcal{S}_n. \tag{2}$$

By taking into account the header of each packet, we have $l_i\,(PacketLength) = h_i(HeaderLength) + d_i(DataLength)$. In [21], the data from each source are directly encapsulated into packets with different sizes that are application-specific. This scheme is appropriate for small packet sizes, but may adversely affect system performance with large packet sizes due to the blocking time term in inequality (2), which is related to the maximum length of packets from other sources. We give an example to illustrate this issue.

Assume that node n with a total bandwidth of $1.92Mbps$ acts as a router for two sources s_1 and s_2, whose parameters are given in Table I.

TABLE I
PARAMETERS OF THE EXAMPLE.

	f Hz	l Mb
s_1	5	0.2
s_2	10	0.01

We define *achievable utilization* of a node n to be the maximum utilization of a node, i.e. $\sum_{s\in\mathcal{S}_n} l_s f_s / B_n$, while satisfying the schedulability condition. We define the *leftover bandwidth* as the extra bandwidth capacity of node n to transmit more data, defined as:

$$\min_{i\in\mathcal{S}_n} B_n - \Big(\sum_{s\in\mathcal{S}_n} l_s f_s + L_i f_i\Big).$$

If node n only forwards packets for s_1, then the leftover bandwidth of node n is $1.92 - 0.2 \times 5 = 0.92Mbps$ which seems to be sufficient to forward additional data from source s_2. However, it turns out that node n cannot forward data from both s_1 and s_2 simultaneously due to blocking. The schedulability condition (2) for source s_1 is $0.2 \times 5 \le 1.92Mbps$, which is satisfied; but that for sources s_1 and s_2 includes two inequalities, in which $0.2\times5+0.01\times10+0.2\times10 \le 1.92Mbps$ does not hold. Although the packet size of s_2 is very small, and half of the bandwidth of node n has not been utilized, it is still impossible to meet the schedulability condition. In fact, no matter how small the packet size of s_2 is, the term due to blocking time from packets of s_1 is $0.2 \times 10 = 2Mbps$, which will make the condition (2) false. We call this phenomenon *utilization jump*. It greatly limits the achievable utilization of the nodes and therefore limits the overall network performance. This problem is especially severe for video sensor networks, where nodes send high-resolution video frames periodically at high sampling rates, but is less of a concern for other application, where nodes send scalar measurement values such as temperature, pressure, humidity, etc.

In this paper, we propose a solution for this problem by dividing a large data block from a given data source into multiple smaller fixed-size packets with the same deadline as the original data block. Packets with the same deadline have the same priority, and are processed in FIFO order. We call this new scheme *packet transformation*, in analogy with the technique of *period transformation* in prioritized preemptive scheduling [25]. Since the maximum packet length that can block a transmission task is l, the fixed packet length, we can re-write inequality (2) as

$$\sum_{s\in\mathcal{S}_n} l\, k_s f_s + l f_i \le B_n \text{, for each } i \in \mathcal{S}_n, \tag{3}$$

where $k_s = \lceil \frac{p_s}{d} \rceil = \lceil \frac{p_s}{l-h} \rceil$ is the number of packets resulting from dividing the data block of source s with size p_s.

Compared to the scheme in [21], *packet transformation* increases the achievable utilization by reducing the utilization jump, since the length of a blocking packet is bounded in the new schedulability condition (3). However, it may increase system overhead due to an increased number of packet headers. Therefore, we should choose packet sizes judiciously to strike a balance between the utilization waste due to utilization jump and overhead due to packet headers, as we show in Subsection VI-B.

IV. Mathematical Formulation

In this section, we present the formal problem formulation of the optimal joint sampling rate assignment and dynamic route selection problem for real-time wireless sensor networks. Our formulation models the problem as a nonlinear convex optimization problem with linear constraints as follows:

$$\min_{\boldsymbol{f},\, \mathbf{R}\in\mathcal{R}_s} \sum_{s\in\mathcal{S}} U_s(f_s) \tag{4}$$

subject to

$$\boldsymbol{f} \le \boldsymbol{f}^{max} \tag{5}$$

$$\boldsymbol{f} \ge \boldsymbol{f}^{min} \tag{6}$$

$$\mathbf{A}\boldsymbol{f} \le \boldsymbol{b}. \tag{7}$$

In the above formulation, $\boldsymbol{f}$ and $\mathbf{R}$ are the decision variables representing the flow rates and routes to be decided. $U_s(f_s)$ is a function measuring the utility loss of the real-time flow originated from source s, and $\mathcal{S}$ represents the set of sources in the network. The formulation is discussed in detail in the following subsections.

A. Network Utility Loss Index

For most applications, performance (QoS) improves with increasing sampling rate. Ideally, the best performance is achieved with infinite sampling rate, i.e., continuous sampling, which is obviously not achievable in reality. We use the *Utility Loss Index* (*ULI*) to capture the performance loss using a discrete sampling rate compared to the case when using continuous sampling [24]. For control applications, Seto et al. showed that the ULI is in the following general form:

$$U_s(f_s) = \omega_s \alpha_s e^{-\beta_s f_s},$$

where f_s is the sampling rate of source s, and non-negative values ω_s, α_s and β_s are application-specific parameters, which can be determined through curve fitting using measurement data. In this paper, we generalize the form of ULI function to strictly decreasing differentiable convex function with regard to rate f_s. The sum of ULI over all the sources in the network is defined as *network ULI*, which is the objective function of our formulated optimization problem. The network utility maximization can be achieved by minimizing the network ULI.

B. Single-path Routing

We first introduce some notations used to model the network and routing. They will be encapsulated in the constraints formulation.

B_n Broadcast bandwidth of node n.
$\boldsymbol{b}$ Broadcast bandwidth vector.
K^s Number of acyclic paths from source s to its destination.
$\mathcal{S}$ The set of sources in the network.
S Number of sources in the network.
$\mathcal{N}$ The set of nodes in the network.
N Number of nodes in the network.
$\mathcal{L}$ The set of directional links in the network.
L Number of directional links in the network.

Suppose that there are K^s acyclic paths from source s to its destination, represented by an $L \times K^s$ 0–1 matrix $\mathbf{H}^s$ where

$$H^s_{lj} = \begin{cases} 1, & \text{if path } j \text{ of source } s \text{ uses link } l; \\ 0, & \text{otherwise.} \end{cases}$$

Let $\mathcal{H}^s$ be the set of all columns of $\mathbf{H}^s$ that represents all the available paths to s. Define the $L \times K$ matrix $\mathbf{H}$ as

$$\mathbf{H} = [\mathbf{H}^1 \ldots \mathbf{H}^S],$$

where $K = \sum_s K^s$ is the total number of paths existing in the network, and $\mathbf{H}$ defines the physical topology of the network.

Let $\boldsymbol{w}^s$ be a $K^s \times 1$ vector where the jth entry represents the fraction of flow from s on its jth path such that

$$w^s_j \geq 0 \text{ and } \mathbf{1}^T \boldsymbol{w}^s = 1,$$

where $\mathbf{1}$ is a vector of an appropriate dimension with the value 1 in every entry. We require $w^s_j \in \{0, 1\}$ for single-path routing. Collecting the vectors $\boldsymbol{w}^s$ for $s = 1, \ldots, S$, we get a $K \times S$ block-diagonal matrix $\mathbf{W}$. Let $\mathcal{W}_s$ be the set of all such matrices corresponding to single-path routing defined as

$$\mathcal{W}_s = \{ \ \mathbf{W} \mid \mathbf{W} = diag(\boldsymbol{w}^1, \ldots, \boldsymbol{w}^S) \in \{0, 1\}^{K \times S}, \mathbf{1}^T \boldsymbol{w}^s = 1, \forall s \ \}.$$

As mentioned above, $\mathbf{H}$ defines the set of acyclic paths available to each source, and $\mathbf{W}$ defines how the sources load balance across these paths. Their product defines an $L \times S$ routing matrix $\mathbf{R} = \mathbf{HW}$ that specifies the fraction of the flow of s at each link l. The set of all single-path routing matrices is

$$\mathcal{R}_s = \{ \ \mathbf{R} \mid \mathbf{R} = \mathbf{HW}, \ \mathbf{W} \in \mathcal{W}_s \ \},$$

where

$$R_{ls} = \begin{cases} 1, & \text{if link } l \text{ is in the path of souce } s; \\ 0, & \text{otherwise.} \end{cases}$$

We also define an $N \times S$ traffic matrix $\mathbf{T}$ to specify the relationship between routers and sources, where

$$T_{ns} = \begin{cases} 1, & \text{if node } n \text{ is a router for source } s; \\ 0, & \text{otherwise.} \end{cases}$$

In other words, $T_{ns} = 1$ indicates $s \in \mathcal{S}_n$.

A node n is a router for source s if and only if n forwards the data for s. So we define an $N \times L$ matrix $\mathbf{L}_{out}$ to be the out-link matrix which specifies whether a link $l \in \mathcal{L}$ is an out-link of node $n \in \mathcal{N}$, that is

$$L^{out}_{nl} = \begin{cases} 1, & \text{if link } l \text{ is an out-link of node } n; \\ 0, & \text{otherwise.} \end{cases}$$

Therefore, the traffic matrix $\mathbf{T}$ can be calculated as

$$\mathbf{T} = \mathbf{L}^{out}\mathbf{R}.$$

C. Constraints

1) Maximum Device Limit: A sensor device may impose a limit on sampling rate due to its physical limitations. We use $\boldsymbol{f}^{max}$ to refer to this limit:

$$\boldsymbol{f} \leq \boldsymbol{f}^{max}.$$

This corresponds to Constraint (5).

2) Minimum Application Requirement: An application may impose a minimum sampling rate to maintain its minimum performance level:

$$\boldsymbol{f} \geq \boldsymbol{f}^{min}.$$

This corresponds to Constraint (6).

3) Schedulability Constraint: Based on the analysis in Subsection III-C, we can derive the following constraint from the schedulability condition (3):

$$\mathbf{A}\boldsymbol{f} \leq \boldsymbol{b}.$$

This corresponds to Constraint (7).

D. An Example

To help the readers understand our modeling better, we give a practical example in this subsection. Consider the wireless sensor network shown in Fig. 3, where nodes 1, 3, 4, 11 and 14 are sources (numbered $s_1, \ldots, s_5$ respectively) that send data to their corresponding destinations 15, 16, 1, 13 and 7 (numbered $d_1, \ldots, d_5$ respectively). Suppose the following candidate routes between the sources and the corresponding destinations are obtained with an existing routing algorithm:

$$\mathcal{H}^1 = \begin{cases} 1 \rightarrow 2 \rightarrow 5 \rightarrow 10 \rightarrow 15 \\ 1 \rightarrow 2 \rightarrow 5 \rightarrow 11 \rightarrow 15 \end{cases}$$

$$\mathcal{H}^2 = \begin{cases} 3 \rightarrow 5 \rightarrow 11 \rightarrow 16 \\ 3 \rightarrow 6 \rightarrow 11 \rightarrow 16 \\ 3 \rightarrow 6 \rightarrow 12 \rightarrow 16 \end{cases}$$

$$\mathcal{H}^3 = \begin{cases} 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \end{cases}$$

$$\mathcal{H}^4 = \begin{cases} 11 \rightarrow 10 \rightarrow 9 \rightarrow 13 \\ 11 \rightarrow 10 \rightarrow 14 \rightarrow 13 \\ 11 \rightarrow 15 \rightarrow 14 \rightarrow 13 \end{cases}$$

$$\mathcal{H}^5 = \begin{cases} 14 \rightarrow 10 \rightarrow 5 \rightarrow 6 \rightarrow 7 \\ 14 \rightarrow 10 \rightarrow 11 \rightarrow 6 \rightarrow 7 \\ 14 \rightarrow 10 \rightarrow 11 \rightarrow 12 \rightarrow 7 \\ 14 \rightarrow 15 \rightarrow 11 \rightarrow 6 \rightarrow 7 \\ 14 \rightarrow 15 \rightarrow 11 \rightarrow 12 \rightarrow 7 \\ 14 \rightarrow 15 \rightarrow 16 \rightarrow 12 \rightarrow 7 \end{cases}.$$

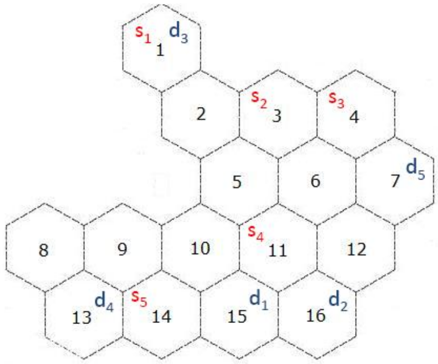

Fig. 3. The Network Deployment of the Example.

TABLE II
PARAMETERS OF THE DATA SOURCES OF THE EXAMPLE.

s	α_s	β_s	ω_s	p_s	f_s^{max}	f_s^{min}
1	0.66	0.3	1	0.01	11	30
2	0.66	1.0	2	0.015	2.5	25
3	0.66	0.5	3	0.02	5	30
4	0.66	0.7	4	0.025	1	40
5	0.66	0.3	5	0.03	2	30

TABLE III
PARAMETERS OF THE NODES OF THE EXAMPLE.

n	1	2	3	4	5	6	7	8
B_n	0.25	0.6	0.4	0.25	0.25	0.7	0.2	0.15
n	9	10	11	12	13	14	15	16
B_n	0.3	0.4	1	0.5	0.25	0.3	0.75	0.3

If every source s selects the first path in $\mathcal{H}^s$ as its route, then we have $\mathbf{W} = diag(\boldsymbol{w}^1, \ldots, \boldsymbol{w}^5)$ where

$$\begin{aligned} \boldsymbol{w}^1 &= (1,0)^T \\ \boldsymbol{w}^2 &= (1,0,0)^T \\ \boldsymbol{w}^3 &= (1)^T \\ \boldsymbol{w}^4 &= (1,0,0)^T \\ \boldsymbol{w}^5 &= (1,0,0,0,0,0)^T. \end{aligned}$$

The routing matrix $\mathbf{R}$ is very large and is therefore not shown here, but we can write the corresponding routing set as follows:

$$\mathcal{R} = \left\{ \begin{array}{ccccccccc} 1 & \rightarrow & 2 & \rightarrow & 5 & \rightarrow & 10 & \rightarrow & 15 \\ 3 & \rightarrow & 5 & \rightarrow & 11 & \rightarrow & 16 & & \\ 4 & \rightarrow & 3 & \rightarrow & 2 & \rightarrow & 1 & & \\ 11 & \rightarrow & 10 & \rightarrow & 9 & \rightarrow & 13 & & \\ 14 & \rightarrow & 10 & \rightarrow & 5 & \rightarrow & 6 & \rightarrow & 7 \end{array} \right. .$$

And the corresponding traffic matrix is:

$$\mathbf{T} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

The nth row in $\mathbf{T}$ shows which sources use node n as a router, and it is related to the schedulability constraints of node n.

The parameters of sources are defined in Table II and the parameters of nodes are defined in Table III. In this paper, the unit for sampling rate is Hz, the unit for data block and packet size is Mb, and the unit for bandwidth is $Mbps$, if unspecified.

Consider node 5 as an example. The 5th row of traffic matrix $\mathbf{T}$ is $(1,1,0,0,1)$, indicating that node 5 forwards data for sources s_1, s_2 and s_5. We can write the local schedulability constraints at node n as $\mathbf{A}_n\boldsymbol{f} \leq \boldsymbol{b}_n$, where $\mathbf{A}_n$ and $\boldsymbol{b}_n$ define the rows relevant to node n in $\mathbf{A}$ and $\boldsymbol{b}$, respectively. Assume the fixed packet length l is $1kb$, and for simplicity, we assume there is no overhead for the header in this example, i.e., $h = 0$. So, a data block of source s with size p_s will be divided into $k_s = \lceil \frac{p_s}{l} \rceil$ packets. According to the schedulability condition (3), we have three inequalities, one for each source. The inequalities can be shown in the form of $\mathbf{A}_n\boldsymbol{f} \leq \boldsymbol{b}_n$ with

$$\mathbf{A}_n = \begin{pmatrix} 0.01 + 0.001 & 0.015 & 0 & 0 & 0.03 \\ 0.01 & 0.015 + 0.001 & 0 & 0 & 0.03 \\ 0.01 & 0.015 & 0 & 0 & 0.03 + 0.001 \end{pmatrix},$$

$$\boldsymbol{b}_n = (0.25, 0.25, 0.25)^T.$$

V. OPTIMIZING SAMPLING RATES WITH DYNAMIC ROUTE SELECTION

In this section, we present solutions for the joint optimal sampling rates and route selection problem.

Our solutions draw upon ideas from the recent research of Network Utility Maximization (NUM), which formulates network system design problem as maximization of the aggregate utility of all the nodes subject to physical or economic constraints. Since the publication of the seminal work [17] by Kelly et al. in 1998, the NUM framework has enabled many applications in network research. Compared to the traditional linear network flow problem, the NUM framework takes advantages of many advances in nonlinear optimization theory and distributed algorithms. Specifically, the design of the distributed solution algorithm in this paper is based on the primal-dual method and dual decomposition technique.

A. Primal and Dual Problems

The primal problem of the optimal sampling rate assignment with dynamic route selection for real-time wireless sensor network problem is described by (4)–(7).

A centralized algorithm can be directly derived by solving the primal problem. However, a centralized algorithm requires collecting data from each source and node. This will generate a lot of traffic and create traffic bottlenecks around the central computing nodes. To avoid this problem, a distributed algorithm is usually more desirable for solving the optimization problem in sensor networks.

An alternative solution for our optimization problem is based on solving the Lagrangian dual problem corresponding to the primal problem (4)–(7):

$$\min_{\lambda\geq 0}\sum_{s}\min_{f^{min}\leq f\leq f^{max}}\left(U_s(f_s)+f_s\min_{\mathbf{R}^s\in\mathcal{H}^s}\sum_{m}A_{ms}\lambda_m\right)-\sum_{m}b_m\lambda_m, \tag{8}$$

where $\boldsymbol{\lambda}$ can be interpreted as the *prices* (or *schedulability prices*) for the schedulability constraints in (7). The dual problem (8) finds the optimal sampling rates and routes in an iterative manner such that the network ULI is minimized and the *schedulability cost* is minimized, as shown later in (10) and in (11) respectively.

Let $f_s(t)$ be the updated sampling rate proposal for source s at iteration t. Let $\mathbf{R}^s(t)$ be the updated routing vector for source s at iteration t. The dual problem can be solved using dual decomposition, in a manner of gradually approaching:

$$\boldsymbol{\lambda}(t)=\arg\min_{\lambda\geq 0}\sum_{s}\left(U_s(f_s)+f_s\sum_{m}A_{ms}\lambda_m\right)-\sum_{m}b_m\lambda_m, \tag{9}$$

$$f_s(t)=\arg\min_{f^{min}\leq f\leq f^{max}}U_s(f_s)+f_s\sum_{m}A_{ms}\lambda_m,\ \forall s\in\mathcal{S}, \tag{10}$$

$$\mathbf{R}^s(t)=\arg\min_{\mathbf{R}^s\in\mathcal{H}^s}\sum_{m}A_{ms}\lambda_m,\ \forall s\in\mathcal{S}. \tag{11}$$

An iterative subgradient method can be used to update the dual variable $\boldsymbol{\lambda}$:

$$\lambda_m(t)=\left[\lambda_m(t-1)+\gamma\left(\sum_{s}f_sA_{ms}-b_m\right)\right]^{+},\ 1\leq m\leq M \tag{12}$$

where function $[\bullet]^+$ is defined as $[x]^+=\max\{x,0\}$, and M is the number of schedulability constraints in constraint set (7). Therefore, Equation (9) can be replaced by Equation (12), and Equations (10)–(12) form a solution for the dual problem (8) by solving the three decomposed sub-problems: price updates, sampling rate assignment and route selection. The sub-problems are coordinated by the prices $\boldsymbol{\lambda}$.

Define the Lagrangian

$$L(\mathbf{R},\boldsymbol{f},\lambda)=\sum_{s}\left(U_s(f_s)+f_s\sum_{m}A_{ms}\lambda_m\right)-\sum_{m}b_m\lambda_m. \tag{13}$$

The primal problem (4)–(7) and the dual problem (8) can be expressed respectively as

$$V_{sp}=\min_{\boldsymbol{f},\mathbf{R}\in\mathcal{R}_s}\min_{\lambda}L(\mathbf{R},\boldsymbol{f},\lambda),$$

$$V_{sd}=\min_{\lambda}\min_{\boldsymbol{f},\mathbf{R}\in\mathcal{R}_s}L(\mathbf{R},\boldsymbol{f},\lambda).$$

Theorem 1:

$$V_{sp}\geq V_{sd}.$$

Proof: Due to the page limit, we omit the proof for *Theorem 1* here. The proof can be found in the extended version of this paper, available at http://www.cs.mcgill.ca/~wshu/public/. ■

Definition 1 (Duality Gap): Duality Gap refers to the difference between the optimal value of the primal problem V_{sp} and that of the corresponding dual problem V_{sd}.

The solution based on Lagrangian dual and dual decomposition works only if there is no duality gap.

B. Centralized Algorithm

A centralized solution can be obtained by solving the sampling rate optimization problem with static routing as shown below for **all** possible route configurations:

$$\min_{f}\sum_{s\in\mathcal{S}}U_s(f_s)$$

subject to

$$\boldsymbol{f}\leq\boldsymbol{f}^{max}$$

$$\boldsymbol{f}\geq\boldsymbol{f}^{min}$$

$$\mathbf{A}\boldsymbol{f}\leq\boldsymbol{b}.$$

Therefore, the optimal routing is the one that minimizes the network ULI, while the solution gives the optimal sampling rates. This optimization problem can be solved with many commercial optimization packages such as the MatLab Optimization Toolbox.[1]

C. Distributed Algorithm

In this section, we present our design of the distributed algorithm for solving the optimal sampling rates with dynamic route selection problem. The algorithm is based on the recent research on cross-layer optimization in TCP/IP networks [28], where each constraint in (7) is given a schedulability price, and each source tries to select the sampling rate and route that minimize the network ULI and the schedulability cost.

The distributed algorithm has two main attributes:

- It converges to the optimal solution of the optimization problem.
- Each update computation is only based on local information of a node or a source.

First we list some notations that will be used in the distributed algorithm:

γ	Step size of updating.
d_n^{out}	The worst-case out-degree of a node n, i.e., the data from how many sources are possible to pass through node n, and use n as a router;
b_n^i	The right side of the ith schedulability constraint at node n, i.e., $b_n^i=B_n$.
$\boldsymbol{b}_n$	A $d_n^{out}\times 1$ vector with all the elements set to B_n.

[1]The *fconmin* function from the MatLab Optimization Toolbox can be employed for optimization problems with nonlinear objective function and linear constraints, and one can select the specific solver from the several options coming with the library, including Interior-Point Algorithm, Active Set Algorithm, and Line Search Algorithm. In addition, if an algorithm is not specified manually, MatLab will automatically choose an optimal one based on application characteristics. For our example, the Line Search Algorithm is chosen automatically by MatLab.

λ_n^i The price for the ith constraint at node n.

$\boldsymbol{\lambda}_n$ The vector of schedulability prices at node n.

A $\mathbf{A} := \mathbf{A}(R, l)$. That is, **A** depends on the routing and the pre-defined packet length l, as shown in Subsection IV-D.

$\mathbf{A}_{ns}\boldsymbol{\lambda}_n$ $\mathbf{A}_{ns}\boldsymbol{\lambda}_n := \sum_{1\le i\le d_n^{out}} A_{ns}^i \lambda_n^i$.

To facilitate the distributed computation, we divide the M constraints in (7) into N sets, each $\boldsymbol{\lambda}_n$ corresponding to a node n. That is, each node only keeps the d_n^{out} schedulability constraints and prices relevant to itself.

The distributed algorithm is made up of an initialization section and an iteration section. In each iteration step, the prices, sampling rates and routes are updated based on the latest information until convergence.

1) Initialization:

a. Each node n sets all the d_n^{out} relevant prices to 1.
b. Each source s sets the sampling rate f_s to f_s^{min}.
c. Each source s selects the first candidate route from $\mathcal{H}^s$.

2) Update Prices at Iteration t:

a. Each source sends out a *RP* (Rate Proposal) packet with the latest rate proposal to its destination along the currently selected route.
b. Upon receiving the *RP* packets from all the relevant sources, each node n computes new prices for the constraints with the following price updating equation:

$$\lambda_n^i(t) = \left[\lambda_n^i(t-1) + \gamma\Big(\sum_s f_s A_{ns}^i - b_n^i\Big)\right]^+, \ 1 \le i \le d_n^{out}.$$

3) Update Sampling Rates at Iteration t:

a. Each destination sends an *SRU* (Sampling Rate Update) packet with value 0 along the reversed path of the current route to the source.
b. Upon receiving an *SRU* packet, each node adds $\mathbf{A}_{ns}\boldsymbol{\lambda}_n$ to the value in the packet, and forwards it along the reversed path.
c. Upon receiving an *SRU* packet, each source s updates its rate proposal according to local optimization as follows:

$$f_s(t) = \arg\min_{f_s^{min}\le f_s\le f_s^{max}} U_s(f_s) + f_s \sum_n \mathbf{A}_{ns}\boldsymbol{\lambda}_n.$$

4) Update Routing at Iteration t:

a. Each destination sends a *RU* (Routing Update) packet with value 0 along the reversed path of every possible route to the source.
b. Upon receiving a *RU* packet, each node adds $\mathbf{A}_{ns}\boldsymbol{\lambda}_n$ to the value in the packet assuming it is a router for current source s, and forwards the packet along the reversed path.
c. Upon receiving all the *RU* packets from all possible routes for a source-destination pair, each source s updates the routing according to local optimization

$$\mathbf{R}^s(t) = \arg\min_{\mathbf{R}^s\in\mathcal{H}^s} \sum_n \mathbf{A}_{ns}\boldsymbol{\lambda}_n.$$

D. Convergence Criteria

Definition 2 (Equilibrium): We say that $(\tilde{\mathbf{R}}, \tilde{\boldsymbol{f}}, \tilde{\boldsymbol{\lambda}})$ is an equilibrium if it is a fixed point of the above algorithm. That is, starting from routing $\tilde{\mathbf{R}}$, sampling rates $\tilde{\boldsymbol{f}}$ and the associated prices $\tilde{\boldsymbol{\lambda}}$, the algorithm yields $(\tilde{\mathbf{R}}, \tilde{\boldsymbol{f}}, \tilde{\boldsymbol{\lambda}})$ for the next iteration.

Theorem 2: An equilibrium $(\tilde{\mathbf{R}}, \tilde{\boldsymbol{f}}, \tilde{\boldsymbol{\lambda}})$ exists if and only if there is no duality gap between the primal problem (4)–(7) and the dual problem (8). In this case, the equilibrium $(\tilde{\mathbf{R}}, \tilde{\boldsymbol{f}}, \tilde{\boldsymbol{\lambda}})$ is a solution for both the primal and dual problems.

Proof: Please refer to Appendix-A. ■

According to *Theorem 2*, the distributed algorithm has an equilibrium exactly when there is no duality gap in the network utility optimization, i.e., when $V_{sp} = V_{sd}$. In other words, when the distributed algorithm converges, the equilibrium found is the solution of the optimization problem (4)–(7).

In practical applications, the distributed algorithm terminates when an equilibrium is found, which is characterized by the following convergence criteria:

$$||\boldsymbol{\lambda}(t) - \boldsymbol{\lambda}(t-1)||_n \le \varepsilon_\lambda, \tag{14}$$

$$||\boldsymbol{f}(t) - \boldsymbol{f}(t-1)||_n \le \varepsilon_f, \tag{15}$$

$$\mathbf{R}(t) = \mathbf{R}(t-1), \tag{16}$$

where $\varepsilon_\lambda > 0$ and $\varepsilon_f > 0$ are sufficiently small real numbers. $||v||_n$ denotes the nth-norm of vector $v = (v_1, \ldots, v_k)$. That is, $||v||_1 = \max_i v_i$ and $||v||_n = (\sum_{i=1}^k v_i^n)^{\frac{1}{n}}$ when $n \in \mathbb{Z}^+$ and $n \neq 1$.

VI. Performance Evaluation

Extensive simulation experiments have been conducted with MatLab to demonstrate the efficacy of our solutions. The results and some further analysis are presented in the section.

A. Convergence

This simulation is based on the parameters of the example in Subsection IV-D.

First, we solved the optimization problem with the centralized algorithm described in Subsection V-B. The optimal network ULI is 0.1877, with $\boldsymbol{f}^* = (22.7273, 10.0000, 11.9048, 11.5385, 9.6775)^T$ and $\boldsymbol{r}^* = (2, 2, 1, 1, 4)$ where the sth element in $\boldsymbol{r}^*$ indicates the optimal route for source s. For example, the 5th element of $\boldsymbol{r}^*$ is 4, meaning that s_5 should select the 4th route from its candidate route set $\mathcal{H}^5$.

Another simulation experiment was conducted using the distributed algorithm proposed in Subsection V-C, with the step size γ set to 0.3. The convergence criteria are described by conditions (14)–(16), with 2nd-norm, and with ε_λ and ε_f both set to 1×10^{-9}. The algorithm converges within 1000 iterations and the result is exactly the same as the one obtained from the centralized solution. The convergence of the distributed algorithm is shown in Fig.4 and Fig.5, where Fig.4 shows the convergence of the global network ULI and Fig.5 shows the convergence of the sampling rate of each source.

Fig. 6 shows the leftover bandwidth of each node after the convergence. According to the figure, it is non-negative for every node, indicating that all the nodes are schedulable, since

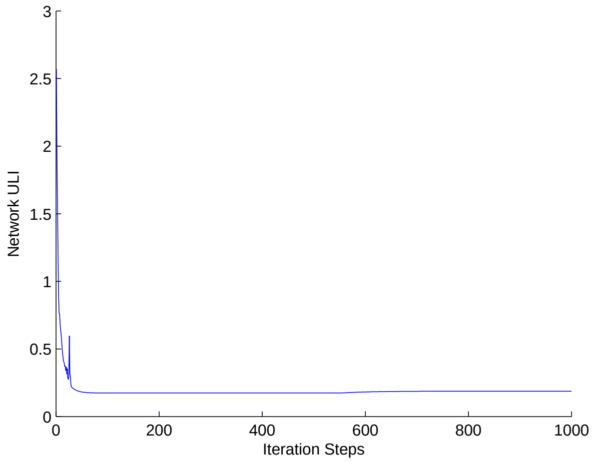

Fig. 4. The Convergence of Network ULI.

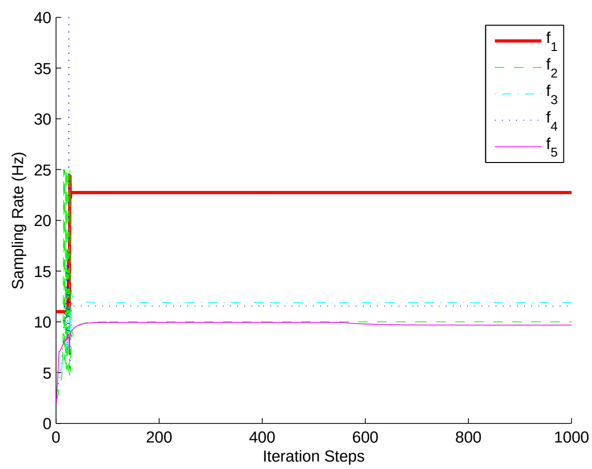

Fig. 5. The Convergence of Sampling Rates.

the schedulability condition (3) for data transmission model is satisfied.

To demonstrate the advantage of optimizing sampling rates with dynamic route selection, we enumerated all possible routings, solved each corresponding optimization problem with static routing, and then calculated the average network ULI. The average value of network ULI for this experiment is 3.1866, and this shows the importance of dynamic routing: the network utility obtained by optimizing sampling rates with static routing may be very pessimistic for a given static routing.

B. Effect of Varied Packet Sizes

We also conducted simulation experiments to show how different packet sizes affect the optimal network utility. Assume the header of a packet takes 96 bits, which is a reasonable

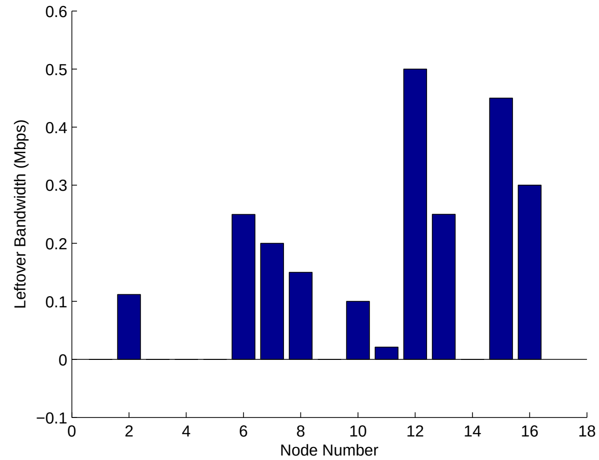

Fig. 6. The Leftover Bandwidths after Convergence.

TABLE IV
PARAMETERS OF THE DATA SOURCES FOR THE SIMULATION.

s	α_s	β_s	ω_s	p_s	f_s^{max}	f_s^{min}
1	0.66	0.3	1	1	∞	0
2	0.66	1.0	2	1.5	∞	0
3	0.66	0.5	3	2	∞	0
4	0.66	0.7	4	2.5	∞	0
5	0.66	0.3	5	3	∞	0

value in real applications. Tables IV and V show the simulation parameters.

The optimal network ULI is 0.8433 if data blocks are not divided into packets, even if the optimization considers the dynamic routing. Based on our packet transformation technique, we varied the packet size that the data blocks are divided into, and the result is shown in Fig. 7. The achievable optimal network ULI is around 0.5, which is much better than the result of 0.8433 obtained without using the packet transformation technique, i.e., the method in [21].

Another interesting observation from Fig. 7 is that, the result forms a U-shape: the minimum network ULI is achieved with a medium packet size that is neither too small nor too large. The reason is that, there is a tradeoff between the overhead of transmitting the header information and the utilization waste in schedulability analysis, caused by the worst-case blocking time for sending a packet. A more detailed discussion about the tradeoff can be found in Subsection III-C.

TABLE V
PARAMETERS OF THE NODES FOR THE SIMULATION.

n	1	2	3	4	5	6	7	8
B_n	16.0	30.0	24.0	16.0	16.0	42.0	12.0	10.0
	9	10	11	12	13	14	15	16
	20.0	24.0	54.0	28.0	16.0	20.0	48.0	14.0

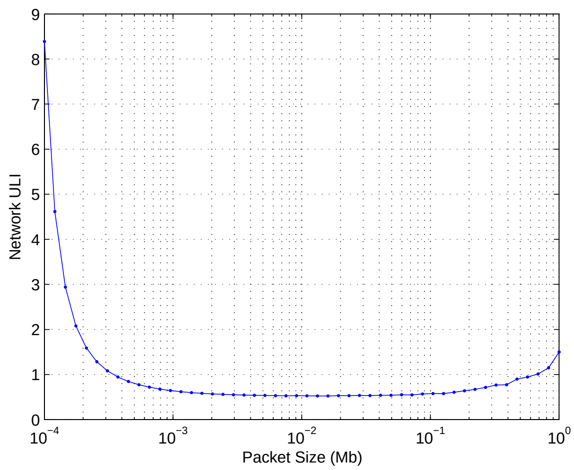

Fig. 7. The Optimal Network ULI Changes with the Packet Length.

VII. Conclusion and Future Work

In this paper, we address the problem of optimizing sampling rate assignment with dynamic route selection for real-time wireless sensor networks. We show that the packet transformation scheme, which involves splitting large data block into smaller packets, can achieve better network utility, especially for applications with large data block sizes. This scheme facilitates schedulability analysis, as well as the implementation of the distributed algorithm for real-time wireless sensor networks. Furthermore, we model the problem as a holistic optimization problem with nonlinear objective function and linear constraints, and present an efficient distributed algorithm based on the primal-dual method. Leveraging the dual decomposition technique, the holistic optimization problem is solved by decomposing the original problem into three sub-problems, which can be solved separately and iteratively. We also demonstrate the efficacy of the distributed algorithm, and the superiority of dynamic routing compared to static routing, via extensive simulations.

There are a number of future research directions. First, the convergence speed of the distributed algorithm is dependent upon some design parameters, e.g., the initial prices λ and the step size of updating γ, it will be interesting to study how to set these values to maximize the convergence speed. Further, we plan to implement our algorithm in real-life sensor network testbed with video surveillance or real-time control applications.

Acknowledgment

This work is supported in part by NSERC Discovery Grants awarded to Xue Liu and Sathish Gopalakrishnan, and by Hong Kong RGC CERG #613506 awarded to Zonghua Gu.

References

[1] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, 1 edition, 1995.

[2] J. Bolot, T. Turletti, and I. Wakeman. Scalable feedback control for multicast video distribution in the internet. In *Proceedings of ACM SIGCOMM*, pages 58–67, 1994.

[3] D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. In *Proceedings of the 1st Workshop on Sensor Networks and Applications (WSNA)*, pages 22–31, 2002.

[4] G. C. Butazzo. *Hard Real-Time Computing Systems*. Springer, 2 edition, 2005.

[5] M. Caccamo, L. Y. Zhang, L. Sha, and G. Buttazzo. An implicit prioritized access protocol for wireless sensor networks. In *Proceedings of the 23rd IEEE Real-Time Systems Symposium (RTSS)*, pages 39–48, 2002.

[6] L. Chen, S. H. Low, M. Chiang, and J. C. Doyle. Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks. In *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–13, 2006.

[7] Y. Chen, C. Lu, and X. Koutsoukos. Optimal discrete rate adaptation for distributed real-time systems. In *Proceedings of the 28th IEEE Real-Time Systems Symposium (RTSS)*, pages 181–192, 2007.

[8] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle. Layering as optimization decomposition: A mathematical theory of network architectures. In *Proceedings of the IEEE*, volume 95, pages 255–312, 2007.

[9] O. Chipara, Z. He, G. Xing, Q. Chen, X. Wang, C. Lu, J. Stankovic, and T. Abdelzaher. Real-time power-aware routing in sensor networks. In *Proceedings of the 14th IEEE International Workshop on Quality of Service (IWQoS)*, pages 83–92, 2006.

[10] Q. Gao, J. Zhang, X. Shen, and B. Larish. A cross-layer optimization approach for energy efficient wireless sensor networks: coalition-aided data aggregation, cooperative communication, and energy balancing. *Adv. MultiMedia*, 2007(1), 2007.

[11] S. Ghosh, R. Rajkumar, J. Hansen, and J. Lehoczky. Scalable resource allocation for multi-processor qos optimization. In *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS)*, pages 174–183, 2003.

[12] J. He, M. Chiang, and J. Rexford. TCP/IP interaction based on congestion price: Stability and optimality. In *IEEE International Conference on Communications (ICC)*, volume 3, pages 1032–1039, 2006.

[13] T. He, J. A. Stankovic, C. Lu, and T. Abdelzaher. SPEED: A stateless protocol for real-time communication in sensor networks. In *Proceedings of the 23rd International Conference on Distributed Computing Systems (ICDCS)*, pages 46–55, 2003.

[14] W. Heinzelman, J. Kulik, and H. Balakrishnan. Adaptive protocols for information dissemination in wireless sensor networks. In *Proceedings of the 5th International Conference on Mobile Computing and Networking (MobiCom)*, pages 174–185, 1999.

[15] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom)*, pages 243–254, 2000.

[16] F. Kelly. Charging and rate control for elastic traffic. *European Transactions on Telecommunications*, 8:33–37, 1997.

[17] F. Kelly, A. Maulloo, and D. Tan. Rate control in communication networks: Shadow prices, proportional fairness and stability. *Journal of the Operational Research Society*, 49, 1998.

[18] J. F. Kurose and R. Simha. A microeconomic approach to optimal resource allocation in distributed computer systems. *IEEE Transactions on Computers*, 38(5):705–717, 1989.

[19] C. Lee, J. Lehoczky, D. Siewiorek, R. Rajkumar, and J. Hansen. A scalable solution to the multi-resource qos problem. In *Proceedings of the 20th IEEE Real-Time Systems Symposium (RTSS)*, pages 315–326, 1999.

[20] X. Lin and N. B. Shroff. The impact of imperfect scheduling on cross-layer congestion control in wireless networks. *IEEE/ACM Transactions on Networking*, 14(2):1804–1814, 2006.

[21] X. Liu, Q. Wang, W. He, M. Caccamo, and L. Sha. Optimal real-time sampling rate assignment for wireless sensor networks. *ACM Transactions on Sensor Networks*, 2(2):263–295, 2006.

[22] S. H. Low and D. E. Lapsley. Optimization flow control — I: Basic algorithm and convergence. *IEEE/ACM Transactions on Networking*, 7(6):861–874, 1999.

[23] R. Rajkumar, C. Lee, J. Lehoczky, and D. Siewiorek. A resource allocation model for qos management. In *Proceedings of the 18th IEEE Real-Time Systems Symposium (RTSS)*, pages 298–307, 1997.

[24] D. Seto, J. P. Lehoczky, L. Sha, and K. G. Shin. On task schedulability in real-time control system. In *Proceedings of the 17th IEEE Real-Time Systems Symposium (RTSS)*, pages 13–21, 1996.

[25] L. Sha, J. Lehoczky, and R. Rajkumar. Solutions for some practical problems in prioritized preemptive scheduling. In *Proceedings of the 7th IEEE Real-Time Systems Symposium (RTSS)*, 1986.

[26] L. Sha, X. Liu, M. Caccamo, and G. Buttazzo. Online control optimization using load driven scheduling. In *Proceedings of the 39th IEEE Conference on Decision and Control (CDC)*, volume 5, pages 4877–4882, 2000.

[27] C. A. Waldspurger and W. E. Weihl. Lottery scheduling: Flexible proportional-share resource management. In *Proceedings of the 1st USENIX Conference on Operating Systems Design and Implementation (OSDI)*, pages 1–11, 1994.

[28] J. Wang, L. Li, S. H. Low, and J. C. Doyle. Cross-layer optimization in TCP/IP networks. *IEEE/ACM Transactions on Networking*, 13(3):582–595, 2005.

[29] Y. Xu, J. S. Heidemann, and D. Estrin. Geography-informed energy conservation for ad hoc routing. In *Proceedings of the 7th International Conference on Mobile Computing and Networking (MobiCom)*, pages 70–84, 2001.

[30] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic. Impact of radio irregularity on wireless sensor networks. In *Proceedings of the 2nd International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 125–138, 2004.

Appendix

A. Proof of Theorem 2

1) Necessity: Let $(\tilde{\mathbf{R}}, \tilde{\boldsymbol{f}}, \tilde{\boldsymbol{\lambda}})$ be an equilibrium of the distributed algorithm, then we have

$$\sum_m \tilde{A}_{ms}\tilde{\lambda}_m = \min_{\mathbf{R}^s \in \mathcal{H}^s} \sum_m A_{ms}\tilde{\lambda}_m, \ \forall s \in \mathcal{S}, \tag{17}$$

$$(\tilde{\boldsymbol{f}}, \tilde{\boldsymbol{\lambda}}) = \arg\min_{\boldsymbol{\lambda}} \min_{\boldsymbol{f}} \sum_s \Big(U_s(f_s) + f_s \sum_m \tilde{A}_{ms}\lambda_m\Big) - \sum_m b_m \lambda_m. \tag{18}$$

We will show that $(\tilde{\mathbf{R}}, \tilde{\boldsymbol{f}}, \tilde{\boldsymbol{\lambda}})$ solves the dual problem (8). Then, since the dual problem lower bounds the primal problem (4)–(7) (*Theorem 1*), and $\tilde{\mathbf{R}} \in \mathcal{R}_s$ is a single-path routing and hence primal feasible, $(\tilde{\mathbf{R}}, \tilde{\boldsymbol{f}}, \tilde{\boldsymbol{\lambda}})$ also solves the primal problem.

We assume $(\mathbf{R}^*, \boldsymbol{f}^*, \boldsymbol{\lambda}^*)$ is the optimal solution for the dual problem (8). That is,

$$\begin{aligned} & (\mathbf{R}^*, \boldsymbol{f}^*, \boldsymbol{\lambda}^*) \\ = \ & \arg\min_{\boldsymbol{\lambda}} \min_{\boldsymbol{f}} \sum_s \Big(U_s(f_s) + f_s \min_{\mathbf{R}^s \in \mathcal{H}^s} \sum_m A_{ms}\lambda_m\Big) - \sum_m b_m \lambda_m. \end{aligned} \tag{19}$$

Let

$$g_1(\boldsymbol{\lambda}) := \min_{\boldsymbol{f}} \sum_s \Big(U_s(f_s) + f_s \sum_m \tilde{A}_{ms}\lambda_m\Big) - \sum_m b_m \lambda_m,$$

$$g_2(\boldsymbol{\lambda}) := \min_{\boldsymbol{f}} \sum_s \Big(U_s(f_s) + f_s \min_{\mathbf{R}^s \in \mathcal{H}^s} \sum_m A_{ms}\lambda_m\Big) - \sum_m b_m \lambda_m.$$

Then (18) implies $g_1(\tilde{\boldsymbol{\lambda}}) = \min_{\boldsymbol{\lambda}} g_1(\boldsymbol{\lambda})$, and (19) implies $g_2(\boldsymbol{\lambda}^*) = \min_{\boldsymbol{\lambda}} g_2(\boldsymbol{\lambda})$. Since $\tilde{\mathbf{R}} \in \mathcal{R}_s$, we have

$$g_1(\boldsymbol{\lambda}) \geq g_2(\boldsymbol{\lambda}), \ \forall \boldsymbol{\lambda}$$

and hence

$$g_1(\tilde{\boldsymbol{\lambda}}) = \min_{\boldsymbol{\lambda}} g_1(\boldsymbol{\lambda}) \geq \min_{\boldsymbol{\lambda}} g_2(\boldsymbol{\lambda}) = g_2(\boldsymbol{\lambda}^*).$$

On the other hand

$$\begin{aligned} g_1(\tilde{\boldsymbol{\lambda}}) &= \min_{\boldsymbol{f}} \sum_s \Big(U_s(f_s) + f_s \sum_m \tilde{A}_{ms}\tilde{\lambda}_m\Big) - \sum_m b_m \tilde{\lambda}_m \\ &= \min_{\boldsymbol{f}} \sum_s \Big(U_s(f_s) + f_s \min_{\mathbf{R}^s \in \mathcal{H}^s} \sum_m A_{ms}\tilde{\lambda}_m\Big) - \sum_m b_m \tilde{\lambda}_m \\ &= g_2(\tilde{\boldsymbol{\lambda}}) \\ &\leq g_2(\boldsymbol{\lambda}^*) \end{aligned}$$

where the second equality follows from (17). Therefore $g_1(\tilde{\boldsymbol{\lambda}}) = g_2(\boldsymbol{\lambda}^*) = g_2(\tilde{\boldsymbol{\lambda}})$ and $L(\tilde{\mathbf{R}}, \tilde{\boldsymbol{f}}, \tilde{\boldsymbol{\lambda}}) = L(\mathbf{R}^*, \boldsymbol{f}^*, \boldsymbol{\lambda}^*)$. Moreover, $(\tilde{\mathbf{R}}, \tilde{\boldsymbol{f}}, \tilde{\boldsymbol{\lambda}})$ is an optimal solution of the dual problem.

2) Sufficiency: Assume that there is no duality gap and $(\mathbf{R}^*, \boldsymbol{f}^*, \boldsymbol{\lambda}^*)$ is an optimal solution for both the primal problem and the dual problem. We claim that it is also an equilibrium of the distributed algorithm. That is, we need to show that

$$\sum_m A^*_{ms}\lambda^*_m = \min_{\mathbf{R}^s \in \mathcal{H}^s} \sum_m A_{ms}\lambda^*_m, \ \forall s \in \mathcal{S} \tag{20}$$

and

$$\begin{aligned} (\boldsymbol{f}^*, \boldsymbol{\lambda}^*) &= \arg\min_{\boldsymbol{\lambda}} \min_{\boldsymbol{f}} L(\mathbf{R}^*, \boldsymbol{f}, \boldsymbol{\lambda}) \\ &= \arg\min_{\boldsymbol{f}} \min_{\boldsymbol{\lambda}} L(\mathbf{R}^*, \boldsymbol{f}, \boldsymbol{\lambda}) \end{aligned} \tag{21}$$

where the equality (21) follows from the assumption that there is no duality gap.

Since $(\mathbf{R}^*, \boldsymbol{f}^*, \boldsymbol{\lambda}^*)$ solves the dual problem (8), the optimal routing matrix $\mathbf{R}^*$ satisfies (20) by the Saddle Point Theorem [1]. $(\mathbf{R}^*, \boldsymbol{f}^*, \boldsymbol{\lambda}^*)$ also solves the primal problem (4)–(7). In particular, $(\boldsymbol{f}^*, \boldsymbol{\lambda}^*)$ solves the utility optimization problem over sampling rates and its Lagrangian dual, with $\mathbf{R}^*$ as the routing matrix, i.e., $(\boldsymbol{f}^*, \boldsymbol{\lambda}^*)$ satisfies (21).