Cross-Layer Interactions in Multihop Wireless Sensor Networks: A Constrained Queueing Model

YANG SONG
University of Florida
and
YUGUANG FANG
University of Florida
Xidian University

In this article, we propose a constrained queueing model to investigate the performance of multihop wireless sensor networks. Specifically, the cross-layer interactions of rate admission control, traffic engineering, dynamic routing, and adaptive link scheduling are studied jointly with the proposed queueing model. In addition, the stochastic network utility maximization problem in wireless sensor networks is addressed within this framework. We propose an adaptive network resource allocation scheme, called the ANRA algorithm, which provides a joint solution to the multiple-layer components of the stochastic network utility maximization problem. We show that the proposed ANRA algorithm achieves a near-optimal solution, that is, $(1 - \epsilon)$ of the global optimum network utility where ϵ can be arbitrarily small, with a trade-off with the average delay experienced in the network. The proposed ANRA algorithm enjoys the merit of self-adaptability through its online nature and thus is of particular interest for time-varying scenarios such as multihop wireless sensor networks.

Categories and Subject Descriptors: C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communications*

General Terms: Algorithms, Design, Performance, Theory

Additional Key Words and Phrases: Cross-layer design, online algorithms, stochastic network optimization, stochastic utility maximization

4

Y. Fang is also a Changjiang Scholar Chair Professor with the National Key Laboratory of Integrated Services Networks, Xidian University, Xi'an, China.

This work was supported in part by the U.S. National Science Foundation under Grants CNS-0916391, CNS-0721744 and CNS-0626881. The work of Y. Fang was also partially supported by the National Natural Science Foundation of China order Grant 61003300, the Fundamental Research Funds for the Central Universities under Grant JY1000090102, and the 111 Project under Grant B08038.

Authors' addresses: Y. Song, Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611; email: yangsong@ufl.edu; Y. Fang, Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL 32611; email: fang@ece.ufl.edu.

DOI 10.1145/1870085.1870089 http://doi.acm.org/10.1145/1870085.1870089

ACM Reference Format:
Song, Y. and Fang, Y. 2010. Cross-Layer interactions in multihop wireless sensor networks: A constrained queueing model. ACM Trans. Model. Comput. Simul. 21, 1, Article 4 (December 2010), 26 pages. DOI = 10.1145/1870085.1870089 http://doi.acm.org/10.1145/1870085.1870089

1. INTRODUCTION

Wireless sensor networks have attracted significant attention in both industrial and academic communities in the past few years, especially with the advances in low-power circuit design and small-size energy supplies which significantly reduce the cost of deploying large-scale wireless sensor networks. The sensor networks can sense and measure the physical environment, for example, temperature, speed, sound, radiation, and the movement of the object, etc. In addition, wireless sensor networks have become an important solution for military applications such as information gathering and intrusion detections. Other implementations of the wireless sensor networks include the healthcare body sensor networks, vehicular-to-roadside communication networks, multimedia sensor networks, and underwater communication networks. For more discussions on the wireless sensor networks, refer to the survey papers such as Akyildiz and Kasimoglu [2004] and Yick et al. [2007].

Since the sensor nodes in the network are usually deployed in places where traditional wired networking solutions are not feasbile, wireless transmissions among sensor nodes are strongly preferred. In addition, due to the restrained size of wireless sensor nodes, the computational capability of a single node is limited. Therefore, the measured information is usually transmitted to a remote Data Processing Center (DPC) for further data analysis. Furthermore, due to the unreliable wireless links, multiple data sinks may exist in the network which collect the measured data and transmit the packets to the DPC node securely and reliably, possibly through the Internet.

Before the wide deployment of wireless sensor networks, a systematic understanding on the performance of the multihop wireless sensor networks is desired. However, finding a suitable and accurate analytical model for wireless sensor networks is particularly challenging. First, the time-varying channel conditions among wireless links significantly complicate the analysis for the network performance in terms of throughput and experienced delay, even in an average sense [Stolyar 2006; Lin 2006; Gupta and Shroff 2009]. Secondly, due to the unpredictability of the behavior of the monitored object, the exogenous traffic arrival to the network, that is, the number of newly generated packets, is a stochastic process. Therefore, to ensure the stability of the network, that is, to keep the queues in the network constantly finite, the analytical model of wireless sensor networks should comprise a rate admission control mechanism which can dynamically adjust the number of admitted packets into the network. Thirdly, due to the hostile wireless communication links, a dynamic routing scheme should be included in the analytical model. Moreover, the model should capture the complex issue of wireless link scheduling which is significantly challenging due to the mutual interference of wireless transmissions. Lastly, in order to fully explore the network resource and to mitigate

the network congestion, an appropriate analytical network model should be able to dynamically deliver packets through multiple data sinks and thus an automatic load balancing solution can be achieved.

In the existing literature, most of the proposed models for wireless sensor networks rely on the fluid model [Kelly et al. 1998], where a flow is characterized by a source node and a specific destination node, for example, Kelly et al. [1998], Chiang [2005], Chiang et al. [2007], Low and Lapsley [1999]. However, this model is not applicable to the cases where the generated packets can be delivered to *any* of the sink nodes, that is, the destination node is one of the sinks and is selected dynamically. Moreover, this fluid model neglects the actual queue interactions within the wireless sensor network. In this article, to study the cross-layer interactions of the multihop wireless sensor networks, we propose a *constrained queueing model* where a packet needs to wait for service in a data queue. More specifically, we investigate the joint rate admission control, dynamic routing, adaptive link scheduling, and automatic load balancing solution to the wireless sensor network through a set of interconnected queues. Due to the wireless interference and the underlying scheduling constraints, at a particular time slot, only a subset of queues can be scheduled for transmissions simultaneously. To demonstrate the effectiveness of the proposed constrained queueing model, we investigate the Stochastic Network Utility Maximization (SNUM) problem in multihop wireless sensor networks. Based on the proposed queueing model, we develop an Adaptive Network Resource Allocation (ANRA) scheme which is a cross-layer solution to the SNUM problem and yields a $(1-\epsilon)$ near-optimal solution to the global optimum network utility where $\epsilon > 0$ can be arbitrarily small. The proposed ANRA scheme consists of multiple-layer components such as joint rate admission control, traffic splitting, dynamic routing, as well as adaptive link scheduling. In addition, the ANRA scheme is essentially an online algorithm which only requires the instantaneous information of the current time slot and hence significantly reduces the computational complexity.

The rest of the article is organized as follows. Section 2 briefly summarizes the related work in the literature. The constrained queueing model for the cross-layer interactions of wireless sensor networks is proposed in Section 3. The stochastic network utility maximization problem of the wireless sensor network is investigated in Section 4, where a cross-layer solution, called the ANRA scheme, is developed. The performance analysis of the ANRA scheme is provided in Section 5. An example which demonstrates the effectiveness of the ANRA scheme is given in Section 6 and Section 7 concludes this article.

2. RELATED WORK

To capture the cross-layer interactions of multihop wireless sensor networks, several analytical models have been proposed in the literature. For example, in Chiang [2005], Chiang et al. [2007], Eryilmaz and Srikant [2005], Kelly et al. [1998], Low and Lapsley [1999], Song and Fang [2007], the multihop network resource allocation problem has been studied through a fluid model. Each flow, or session, is characterized by a source and a destination node where single path routing or multipath routing schemes are implemented. Most of the

work rely on the dual optimization framework which decomposes the complex cross-layer interactions into separate sublayer problems by introducing dual variables. For example, the flow injection rate, controlled by the source node of the flow, is calculated by solving an optimization problem with the knowledge of the dual variables, called shadow prices [Kelly et al. 1998; Low and Lapsley 1999], of all the links that are utilized. However, there are several drawbacks for the fluid-based model. First, to calculate the optimum flow injection rate, the information along all paths should be collected in order to implement the rate admission control mechanism. In a dynamic environment such as wireless sensor networks, this process of information collection may take a significant amount of time which inevitably prolongs the network delay. Secondly, the optimization-based solutions usually pursue fixed operating points which are hardly optimal in dynamic wireless settings with stochastic traffic arrivals and time-varying channel conditions. Thirdly, the fluid model usually assumes that the changes of the flow injection rates are “perceived" by all the nodes along its paths instantaneously. The actual queue dynamics and interactions are neglected.

In contrast, following the seminal paper of Tassiulas and Ephremides [1992], many solutions have been focused on the queueing model for studying the complex interactions of communication networks. Neely et al. extend the results of Tassiulas and Ephremides [1992] into wireless networks with time-varying channel conditions [Neely 2003]. For a more complete survey of this area, refer to Georgiadis et al. [2006]. The key component of the queue-based solutions in these papers is the MaxWeight scheduling algorithm [Tassiulas and Ephremides 1992; Neely 2003]. Intuitively, at a time slot, the network picks the set of queues which: (1) can be active simultaneously and (2) have the maximum overall weight. It is well-known that the MaxWeight algorithm is throughput-optimal in the sense that any arrival rate vector that can be supported by the network can be stabilized under the MaxWeight scheduling algorithm. In addition, the MaxWeight algorithm is an online policy which requires only the information about current queue sizes and channel conditions. However, one notorious drawback of the MaxWeight algorithm is the delay performance. The reason is that in order to achieve the throughput-optimality, the MaxWeight algorithm explores a dynamic routing solution where long paths are utilized even under a light traffic load. This phenomenon is substantiated via simulations by a recent work of Ying et al. [2009]. In Ying et al. [2009], the authors propose a variant of the MaxWeight algorithm where the average number of hops of transmissions is minimized. Therefore, when the traffic is light, the proposed solution provides a much lower delay than the traditional MaxWeight algorithm. However, as a trade-off, the induced network capacity region in Ying et al. [2009] is noticeably smaller than that of the original MaxWeight algorithm. Consequently, it is difficult to provide a minimum rate guarantee on all the sessions in the network. Our work is inspired by Ying et al. [2009]. With respect to Ying et al. [2009], however, our article innovates in the following ways. First, we focus on a heavy-loaded wireless sensor network. Therefore, our solution incorporates a rate admission control mechanism which is not considered in Ying et al. [2009]. Secondly, rather than minimizing the overall number of hops, we

maximize the overall network utility which can also ensure the fairness among competitive traffic sessions. Thirdly, we specifically provide a minimum average rate guarantee for every session to ensure the QoS requirement. Fourthly, instead of a single destination scenario as considered in Ying et al. [2009], we extend the model to cases where multiple data sink nodes are available. Each source node can deliver the packets to any of the sinks. Moreover, the dynamic routing and the issue of automatic load balancing is realized by the network on-the-fly. Finally, while Ying et al. [2009] treats different sessions equally when minimizing the overall number of hops, our model prioritizes all the sessions with different QoS requirements. Therefore, a more flexible solution with service differentiations can be achieved. We will present the constrained queueing model in the next section.

3. A CONSTRAINED QUEUEING MODEL FOR WIRELESS SENSOR NETWORKS

3.1 Network Model

We consider a multihop wireless sensor network represented by a directed graph $\mathcal{G} = \{N, L\}$ where N and L denote the set of vertices and the set of links, respectively. We will use the notation of $|A|$ to represent the cardinality of set A, for example, the number of nodes in the network is $|N|$ and $|L|$ is the number of links. Time is slotted as $t = 0, 1, \ldots$ and at a particular time slot t, the instantaneous channel data rate of link $(m, n) \in L$ is denoted by $\mu_{m,n}(t)$. In other words, link (m, n) can transmit a number of $\mu_{m,n}(t)$ packets during time slot t. We assume that during one time slot, the channel conditions of links will remain constant. However, the value of $\mu_{m,n}(t)$ is subject to changes at the boundaries of time slots. Denote $\boldsymbol{\mu}(t)$ as the *network link rate vector* at time slot t. In this article, we assume that $\boldsymbol{\mu}(t)$ remains constant within one time slot but is subject to changes at time slot boundaries. The value of $\boldsymbol{\mu}(t)$ is assumed to be evolving following an irreducible and aperiodic Markovian chain with arbitrarily large yet finite number of states.[1] However, the steady state distributions are unknown to the network.

At time slot t, the network selects a feasible *link schedule*, denoted by $I(t) = \{I_1(t), I_2(t), \cdots, I_{|L|}(t)\}$ where $I_l(t) = 1$ if link l is selected to be active and $I_l(t) = 0$ otherwise. The set of all feasible link schedules is denoted by $\Omega(t)$ which is determined by the underlying scheduling constraints such as interference models and duplex constraints. Therefore, selecting an interference-free link schedule in the network graph $\mathcal{G}$ is equivalent to the process of attaining an independent set in the associated conflict graph $\tilde{\mathcal{G}}$, where the vertices are the links in $\mathcal{G}$ and a link exists in $\tilde{\mathcal{G}}$ if the two original links in $\mathcal{G}$ cannot transmit simultaneously.

[1]It should be noted that the Markovian assumption is for the ease of analysis. Our proposed model can be extended to more general scenarios where the time average of an arbitrary link rate state is well defined, as in Neely et al. [2005].

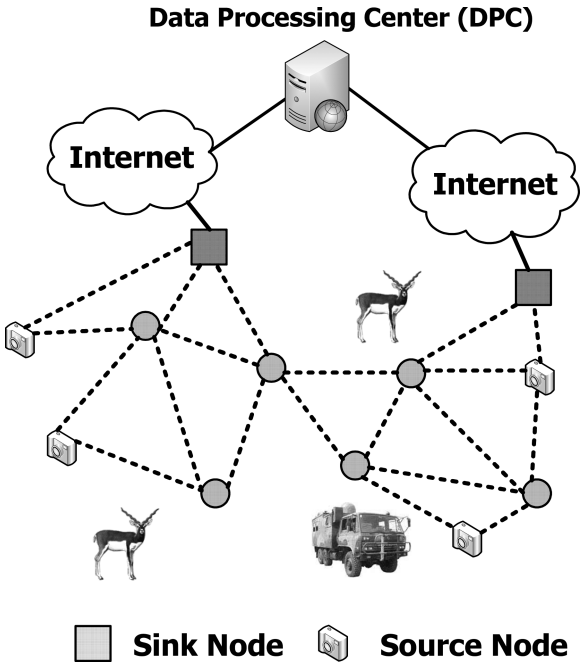

Fig. 1. Topology of wireless sensor networks.

3.2 Traffic Model

There are a number of $|S|$ *source nodes* in the wireless sensor network which consistently monitor the surroundings and inject exogenous traffic to the network. For example, in a wildlife monitoring scenario, the sensors, usually placed with cameras, need to measure the animals' movements and behaviors and transmit the generated packets to the remote Data Processing Center (DPC) in a multihop fashion. To simplify analysis, we assume that each source node is associated uniquely with a *session*. The set of source nodes is denoted by $S = \{n_1^0, n_2^0, \ldots, n_{|S|}^0\}$ where $n_s^0, s = 1, \ldots, |S|$ is the source node of session s. It is worth noting that the following analysis can be extended straightforwardly to the scenarios where each source node may generate multiple sessions.

There are $|D|$ number of *sinks* in the network which are connected to the remote data processing center via the Internet. In other words, the sink nodes can be viewed as the gateways of the wireless sensor network. Denote the set of sinks as $D = \{d_1, d_2, \ldots, d_{|D|}\}$. In this article, we consider a general scenario where the data packets from a source node can be delivered to the DPC via *any* of the sink node in D. Therefore, different from the existing literature such as Chiang [2005], Song and Fang [2007], and Ying et al. [2009], the source nodes do not specify the particular destination node for the generated packets. The selection of the destination node is achieved by the network via dynamic routing schemes. The network topology considered in this article is illustrated in Figure 1.

For a particular node in the network, say node n, we denote ϕ_n^d as the number of minimum hops from node n to the dth data sink in set D. Define

$$\widetilde{\phi}_n = \min_d(\phi_n^d), d = 1, \ldots, |D| \tag{1}$$

as the minimum value of ϕ_n^d for node n, that is, the minimum number of hops from node n to a sink node in set D. We assume that node n is aware of the value of $\widetilde{\phi}_n$ as well as those values of the neighboring nodes, which are attainable via precalculations by traditional routing mechanisms such as Dijkstra's algorithm.

At time slot t, the exogenous arrival of session s, that is, the number of new packets[2] generated by the source node of session s, is denoted by $A_s(t)$. We assume that there is an upper bound for the number of new packets within one time slot, that is, $A_s(t) \leq A_{\max}, \forall s, t$. For ease of exposition, we assume that $A_s(t)$ is independently and identically distributed over time slots with an average rate of λ_s. However, the data rates from multiple source nodes can be arbitrarily correlated. For example, if the wireless sensor network is deployed for monitoring purposes, it is very likely that a movement of the object will trigger several concurrent updates of the nearby sensors.

Denote the vector $\boldsymbol{\lambda} = \{\lambda_1, \ldots, \lambda_{|S|}\}$ as the *network arrival rate vector*. The network capacity region Λ is thus defined as all the *feasible*[3] network arrival vectors that can be supported by the network via certain policies, including those with the knowledge of futuristic traffic arrivals and channel rate conditions. In this article, we consider a heavy-loaded traffic scenario where the network arrival vector $\boldsymbol{\lambda}$ is outside of the network capacity region. Therefore, in order to achieve the network stability, a rate admission control mechanism is implemented at the source nodes. More specifically, at time slot t, we only admit a number of $X_s(t)$ packets into the network from the source node of session s, that is, n_s^0. Apparently, we have

$$X_s(t) \leq A_s(t), \forall s, t. \tag{2}$$

In addition, we assume that each session has a continuous, concave, and differentiable utility function, denoted by $U_s(X_s(t))$, which reflects the degree of satisfaction by transmitting $X_s(t)$ number of packets. It is worth noting that by selecting proper utility functions, the fairness among competitive sessions can be achieved. For example, if $U_s(X_s(t)) = \log(X_s(t))$, a *proportional fairness* among multiple sessions can be enforced [Chiang et al. 2007; Srikant 2003; Shakkottai and Srikant 2008].

3.3 Queue Management

For each node n in the network, there are $|N| - \widetilde{\phi}_n$ number of queues that are maintained and updated. The queues are denoted by $Q_{n,h}$, where

[2]We assume that the packets have a fixed length. For scenarios with variable packet lengths, the unit of data transmissions can be changed to bits per slot and the following analysis still holds.

[3]Note that additional constraints may be imposed. For example, the constraints on the minimum average rate and the maximum average power expenditure can be enforced. For more discussions, please refer to Georgiadis et al. [2006].

$h = \widetilde{\phi}_n, \ldots, |N| - 1$. Note that $|N| - 1$ is the maximum number of hops for a loop-free routing path in the network. The packets in the queue of $Q_{n,h}$ are guaranteed to reach one of the sink nodes in set D within h hops, as will be shown in Section 4. It is interesting to observe that for a newly generated packet by session s, the source node, that is, n_s^0, can place it in any of the queues of $Q_{n_s^0,h}$, where $h = \widetilde{\phi}_{n_s^0}, \ldots, |N| - 1$, for further transmission. That is to say, consecutive packets from the source node n_s^0 may traverse through different number of hops before reaching a destination sink node in set D. Therefore, when a new packet is generated, the source node needs to make a decision on which queue the packet should be placed, namely, traffic splitting decision. In addition, the decision should be made promptly on an online basis with low computational complexity.

With a slight abuse of notation, we use $Q_{n,h}$ to denote the queue itself and $Q_{n,h}(t)$ to represent the number of queue backlogs[4] in time slot t. For a single queue, say $Q_{n,h}$, it is stable if [Neely et al. 2005; Neely 2003]

$$\lim_{B\to\infty} g(B) \to 0, \tag{3}$$

where

$$g(B) = \limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \Pr(Q_{n,h}(t) > B),$$

where B is a positive number. The network is stable if all the individual queues in the network are stable.

For a link $(n, j) \in L$, we require that the packets from $Q_{n,h}$ can be only transmitted to $Q_{j,h-1}$, if exists. Therefore, the queue updating dynamic for $Q_{n,h}$ is given by

$$Q_{n,h}(t+1) \le \left[Q_{n,h}(t) - \sum_{(n,j)\in L} u_{n,j}^{n,h}(t) \right]^+ + \sum_{(m,n)\in L} u_{m,n}^{m,h+1}(t) + \sum_s X_s^h(t)\delta_{n=n_s^0}, \tag{4}$$

where $[A]^+$ denotes $\max(A, 0)$ and $u_{n,j}^{n,h}(t)$ represents the allocated data rate for the transmissions of $Q_{n,h} \to Q_{j,h-1}$ on link (n, j), at time slot t, and

$$\sum_{h=\widetilde{\phi}_n}^{N-1} u_{n,j}^{n,h}(t) = u_{n,j}(t),$$

where $u_{n,j}(t) = \mu_{n,j}(t)$ if $I_{n,j}(t) = 1$, that is, link (n, j) is scheduled to be active during time slot t, and $u_{n,j}(t) = 0$ otherwise. The notation of $X_s^h(t)$ denotes the number of packets that are admitted to the network for session s and are stored in queue $Q_{n,h}$ for future transmissions. The indicator function $\delta_A = 1$ if event A is true and $\delta_A = 0$ otherwise. Note that the inequality in (4) incorporates the scenarios where the transmitter of a particular link has less packets in the queue than the allocated data rate. We assume that during one time slot, the numbers of packets that a single queue can transmit and receive are upper

[4]In the unit of packets.

bounded. Mathematically speaking, we have

$$\sum_{(m,n)\in L} u_{m,n}^{m,h+1}(t) \le u_{in}, \forall n, h, t, \tag{5}$$

and

$$\sum_{(n,j)\in L} u_{n,j}^{n,h}(t) \le u_{out}, \forall n, h, t. \tag{6}$$

3.4 Session-Specific Requirements

In this article, we consider a scenario where each session has a specific rate requirement α_s. Therefore, to ensure the minimum average rate, we need to find a policy that

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} X_s(t) \ge \alpha_s, \forall s. \tag{7}$$

In addition, we assume that each session in the network has an average hop requirement β_s. More specifically, define

$$M_s(t) = \sum_{h=\widetilde{\phi}_{n_s^0}}^{|N|-1} h X_s^h(t), \tag{8}$$

where

$$\sum_{h=\widetilde{\phi}_{n_s^0}}^{|N|-1} X_s^h(t) = X_s(t), \forall s, t.$$

We require that for each session s,

$$\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} M_s(t) \le \beta_s, \forall s. \tag{9}$$

Note that the average hop for a particular session s is related to the average delay experienced and the average energy consumed for the packet transmissions of session s. Therefore, by assigning different values of α_s and β_s, a prioritized solution among multiple competitive sessions can be achieved for the network resource allocation problem.

4. STOCHASTIC NETWORK UTILITY MAXIMIZATION IN WIRELESS SENSOR NETWORKS

In the previous section, we propose a constrained queueing model to investigate the performance of multihop wireless sensor networks. The model consists of several important issues from different layers, including the rate admission control problem, the dynamic routing problem, as well as the challenge of adaptive link scheduling. To better understand the proposed constrained queueing model, in this section, we will examine the Stochastic Network Utility Maximization Problem (SNUM) in multihop wireless sensor networks. As a cross-layer solution, an Adaptive Network Resource Allocation (ANRA) scheme is

proposed to solve the SNUM problem jointly. The proposed ANRA scheme is an online algorithm in nature which provably achieves an asymptotically optimal average overall network utility. In other words, the average network utility induced by the ANRA scheme is $(1-\epsilon)$ of the optimum solution, where $\epsilon > 0$ is a positive number that can be arbitrarily small, with a trade-off with the average delay experienced in the network.

4.1 Problem Formulation

Recall that every session s possesses a utility function $U_s(X_s(t))$ which is continuous, concave, and differentiable. Without loss of generality, in the rest of this article, we will assume that $U_s(X_s(t)) = \log(X_s(t))$. Therefore, in light of the stochastic traffic arrival as well as the time-varying channel conditions, our objective is to develop a policy which maximizes

Stochastic Network Utility Maximization (SNUM) Problem

$$\sum_s E(U_s(X_s(t))) \tag{10}$$

such that:

—The network remains stable.
—The average rate requirements of all $|S|$ sessions, denoted by $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_{|S|}\}$, are satisfied.
—The average hop requirements of all $|S|$ sessions, denoted by $\boldsymbol{\beta} = \{\beta_1, \ldots, \beta_{|S|}\}$, are satisfied.

Note that if the underlying statistical characteristics of the stochastic traffic arrivals and the time-varying channel conditions are known, the SNUM problem is inherently a standard optimization problem and thus is easy to solve. However, due to the unawareness of the steady state distributions, the SNUM problem is remarkably challenging. In addition, in wireless sensor networks, dynamic algorithmic solutions with low computational complexity are strongly desired. In the following, we propose an ANRA scheme to solve the SNUM problem asymptotically. The ANRA scheme is a cross-layer solution which consists of joint rate admission control, traffic splitting, dynamic routing, as well as adaptive link scheduling components. Moreover, the ANRA algorithm can achieve an automatic load balancing solution by utilizing different sink nodes corresponding to the variations of the network conditions. The ANRA algorithm is an online algorithm in nature which requires only the state information of the current time slot. We show that the ANRA algorithm achieves a $(1-\epsilon)$ optimal solution where ϵ can be arbitrarily small. Therefore, the proposed ANRA algorithm is of particular interest for dynamic wireless sensor networks with time-varying environments.

4.2 The ANRA Cross-Layer Algorithm

Before presenting the proposed ANRA scheme, we introduce the concept of virtual queues [Neely 2006; Georgiadis et al. 2006; Stolyar 2005] to facilitate our analysis. Specifically, for each session s, we maintain a virtual queue Y_s, which is initially empty, and the queue updating dynamic is defined as

$$Y_s(t+1) = [Y_s(t) - X_s(t)]^+ + \alpha_s, \forall s, t. \tag{11}$$

Similarly, we define another virtual queue for every session s, denoted by Z_s, and the queue dynamic is given by

$$Z_s(t+1) = [Z_s(t) - \beta_s]^+ + M_s(t), \forall s, t, \tag{12}$$

where $M_s(t)$ is defined in (8). Note that the virtual queues are software-based counters which are easy to maintain. For example, the source node of each session can calculate the values of virtual queues $Y_s(t)$ and $Z_s(t)$ and update the values accordingly following (11) and (12). In addition, we introduce a positive parameter J which is tunable as a system parameter. The impact of J on the algorithm performance will be discussed shortly. The proposed ANRA cross-layer algorithm is given as follows.

Adaptive Network Resource Allocation (ANRA) Scheme:

—**Joint Rate Admission Control and Traffic Splitting (at time t):**
For each source node, say n_s^0, there are a number of queues, that is, $Q_{n_s^0,h}$, $h = \widetilde{\phi_{n_s^0}}, \ldots, |N| - 1$. Find the value of h which minimizes

$$Z_s(t)h + Q_{n_s^0,h}(t), \tag{13}$$

where ties are broken arbitrarily. Denote the optimum value of h as h^*. The source node n_s^0 admits a number of new packets as

$$X_s(t) = \min\left(\widetilde{X_s(t)}, A_s(t)\right), \tag{14}$$

where

$$\widetilde{X_s(t)} = \left[\frac{J}{2(Z_s(t)h^* + Q_{n_s^0,h^*}(t)) - 2Y_s(t)}\right]^+. \tag{15}$$

For traffic splitting, the source node n_s^0 will deposit all $X_s(t)$ packets in $Q_{n_s^0,h^*}$.

—**Joint Dynamic Routing and Link Scheduling (at time t):**
For each link $(m, n) \in L$, define a link weight denoted by $W_{m,n}(t)$, which is calculated as

$$W_{m,n}(t) = \left[\max_{h=\widetilde{\phi_m},\ldots,|N|-1} \left(Q_{m,h}(t) - Q_{n,h-1}(t)\right)\right]^+. \tag{16}$$

Note that if $Q_{n,h-1}$ does not exist, the transmissions from queue $Q_{m,h}$ to $Q_{n,h-1}$ are prohibited. At time slot t, the network selects an interference-free link schedule $I(t)$ which solves

$$\max_{I(t)\in\Omega(t)} \mu_{m,n}(t) W_{m,n}(t). \tag{17}$$

If link (m, n) is active, that is, $I_{m,n}(t) = 1$, the queue of $Q_{m,\bar{h}}$ is selected for transmissions where

$$\bar{h} = \text{argmax}_{h=\widetilde{\phi}_m,\ldots,|N|-1} \left(Q_{m,h}(t) - Q_{n,h-1}(t)\right). \tag{18}$$

End

Note that (17) is similar to the original MaxWeight algorithm introduced in Tassiulas and Ephremides [1992] and generalized in Neely et al. [2005], Neely [2003], and Stolyar [2005]. The dynamic routing and link scheduling are addressed jointly by solving (17), which requires centralized computation. However, following Tassiulas and Ephremides [1992], many works have been focused on the distributed solutions of (17). Although the distributed computation issue is not the focus of this article, we emphasize that our proposed ANRA scheme can be approximated well by existing distributed solutions such as Joo [2008], Akyol et al. [2008], Radunovic et al. [2008], Modiano et al. [2006], Jiang and Walrand [2008], Gupta et al. [2007], Stolyar [2008], and Wu et al. [2007]. For example, in Akyol et al. [2008], each node in the network utilizes an IEEE 802.11 MAC protocol where the contention window size, or equivalently, the channel access probability in Stolyar [2008], is adjusted consistently to approximate the link weight. The accuracy of such random-access-based distributed approximations are studied and evaluated extensively in Akyol et al. [2008]. The scheduling component, that is, (17), of our proposed ANRA scheme can be approximated well by the solutions suggested in the aforesaid papers.

For the packets placed at queue $Q_{n_s^0,h}$, at most h hops of transmissions are needed in order to reach one of the sink nodes in set D. This can be verified straightforwardly due to the requirement that a transmission from $Q_{m,h}$ to $Q_{n,h-1}$ can occur if and only if $h - 1 \geq \widetilde{\phi}_n$. Moreover, the joint rate admission control and the optimum traffic splitting components of ANRA can be implemented by the source node in a distributed fashion. Note that in order to calculate the instantaneous admitted rate, the source node of session s needs only to know the local queue backlog information. Moreover, the decision of traffic splitting requires only local queue information as well. Therefore, at every time slot, the joint rate admission control and traffic splitting decision can be made on an online basis in accordance to the time-varying conditions of local queues. Furthermore, we will show that this simple adaptive strategy does not incur any loss of optimality. The achieved network utility induced by the ANRA scheme can be pushed arbitrarily close to the optimum solution. Next, we will characterize the global optimum utility in the network and provide the main performance results of the proposed ANRA scheme.

4.3 Performance of the ANRA Scheme

In this section, we first characterize the global optimum solution of the SNUM problem in (10). Define U^* as the global maximum network utility that any scheme can achieve, that is, the optimum solution of (10). In order to achieve U^*, it is naturally to consider more complicated policies such as those with the knowledge of futuristic arrivals and channel conditions. However, in the following theorem, we show that, somewhat surprisingly, the global optimum

solution of the SNUM problem can be achieved by certain stationary policies, that is, the responsive action is chosen regardless of the current queue sizes in the network and the time slot that the decision is made. Recall that we have assumed that for each session, $A_s(t)$ is independently and identially distributed over time slots. Denote $\mathbf{A}(t)$ as the vector of instantaneous arrival rates of all sessions, at time slot t. Let $\mathcal{A}$ be the set of all possible value of $\mathbf{A}(t)$. Note that for every element in $\mathcal{A}$, that is, $\mathcal{A}_a, a = 1, \ldots, |\mathcal{A}|$, we have $0 \preceq \mathcal{A}_a \preceq A_{\max}$ where $\preceq$ denotes the element-wise comparison. We use π_a to represent the steady state distribution of $\mathcal{A}_a$.

THEOREM 1. *If the constraints in the SNUM problem are satisfied, the maximum network utility, denoted by U^*, can be achieved by a class of stationary randomized policies. Mathematically, the value of U^* is the solution of the following optimization problem, with the auxiliary variables p_a^k and R_a^k, as*

$$\max \sum_a \pi_a \sum_s U_s\left(\sum_{k=1}^{|S|+1} p_a^k R_a^k\right) \tag{19}$$

such that:

—*The constraints in* (10) *are satisfied.*
—$0 \preceq R_a^k \preceq \mathcal{A}_a$.
—$p_a^k \geq 0, \forall a, k$.
—$\sum_{k=1}^{|S|+1} p_a^k = 1, \forall a$.

PROOF. We prove Theorem 1 by showing that for arbitrary policy which satisfies the constraints in the SNUM problem, the overall network utility is at most U^*, which is the optimum utility attained by a class of stationary randomized policies. In other words, we need to show that

$$U^P = \limsup_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \left(\sum_s U_s(X_s(t))\right) \leq U^*, \tag{20}$$

where U^P is the average network utility under a policy P.

For each state in $\mathcal{A}$, say $\mathcal{A}_a$, define R_a as the set of nonnegative rate vectors that are element-wise smaller than $\mathcal{A}_a$. Define $\mathcal{CR}_a$ as the convex hull of set R_a. Therefore, any point in $\mathcal{CR}_a$ can be considered as a feasible network admitted rate vector given that the current arrival rate vector is $\mathcal{A}_a$. Note that every point in $\mathcal{CR}_a$ is a vector with a dimension of $|S|$-by-1. Therefore, it can be represented by a convex combination of at most $|S| + 1$ points, denoted by $R_a^k, k = 1, \ldots, |S| + 1$, according to Caratheodory's theorem. In light of this, we first consider a time interval from 0 to $T - 1$. Denote $N_a(T)$ as the set of time slots that $\mathbf{A}(t) = \mathcal{A}_a$. Therefore, we can rewrite (20) as

$$U^P = \limsup_{T\to\infty} \sum_a \frac{|N_a(T)|}{T} \sum_s U_s\left(\sum_{k=1}^{|S|+1} p_a^k R_a^k\right).$$

Due to the stationary assumption, we have

$$U^P = \sum_a \pi_a \sum_s \limsup_{T\to\infty} U_s\Big(\sum_{k=1}^{|S|+1} p_a^k R_a^k\Big).$$

Note that we also assume that the utility function is continuous and bounded. Therefore, the compactness of the utility functions is assured. Next, we focus on a subsequence of time durations, denoted by $T_i, i = 1, \ldots, \infty$. Denote

$$U_{net}^P(T_i) = \sum_a \pi_a \sum_s U_s\Big(\sum_{k=1}^{|S|+1} p_a^k(T_i) R_a^k\Big).$$

It is straightforward to verify that

$$U^P = \limsup_{i\to\infty} U_{net}^P(T_i).$$

Due to the compactness of the utility functions, following Bolzano-Weierstrass theorem [Trench 2003], we claim that there exists a subsequence of $T_i, i = 1, \cdots, \infty$, such that

$$\lim_{i\to\infty} U_s\Big(\sum_{k=1}^{|S|+1} p_a^k(T_i) R_a^k\Big) \to \widetilde{U}_s^a.$$

Denote $\widetilde{p}_a^k$ as the values which generate $\widetilde{U}_s^a$, that is,

$$\widetilde{U}_s^a = U_s\Big(\sum_{k=1}^{|S|+1} \widetilde{p}_a^k R_a^k\Big).$$

We have

$$U^P = \limsup_{i\to\infty} U_{net}(T_i) = \sum_a \pi_a \sum_s U_s\Big(\sum_{k=1}^{|S|+1} \widetilde{p}_a^k R_a^k\Big).$$

According to the definition of U^* in (19), we conclude that $U^P \le U^*$. □

Intuitively, Theorem 1 indicates that the global maximum network utility can be achieved by certain randomized stationary policies. However, to calculate U^*, the stationary policy needs to know the steady state distributions which are difficult to obtain in practice. In light of this, we propose an adaptive network resource allocation scheme, namely, ANRA, which is an online solution and does not require such statistical information as a priori. For notation succinctness, denote

$$U^A(t) = \sum_s E\left(U_s(X_s(t))\right)$$

as the expected network utility induced by the ANRA scheme. The performance of the ANRA algorithm, with a parameter J, is given as the following theorem.

THEOREM 2. *For a given system parameter J, we have*

$$\liminf_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} U^A(t) \ge U^* - \frac{\bar{B}}{J}, \tag{21}$$

where $\bar{B}$ is a constant and is given by

$$\bar{B} = |N|(|N|-1)((u_{out})^2 + (u_{in} + A_{\max})^2) + \sum_s ((\alpha_s)^2 + (\beta_s)^2) + |S|(A_{\max})^2((|N|-1)^4 + 1).$$

In addition, the constraints in the SNUM problem, that is, (10)*, are satisfied simultaneously.*

PROOF. The proof of Theorem 2 is deferred to Section 5. □

The value of constant $\bar{B}$ is determined by the number of nodes in the network, the number of sessions, and the values of session requirements, etc. It is worth noting that if we let $J \to \infty$, the performance induced by the ANRA algorithm can be arbitrarily close to the global optimum solution U^*. However, as a trade-off, a larger value of J also yields a longer average queue size in the network. According to Little's Law, a larger queue size corresponds to a longer average delay experienced in the network. Therefore, by selecting the value of J properly, a trade-off between the network optimality and the average delay in the network can be achieved. We will discuss more about this issue in the next section.

5. PERFORMANCE ANALYSIS

In this section, we provide a proof to Theorem 2 in the previous section. Recall that in (11) and (12), we introduce two virtual queues, that is, $Y_s(t)$ and $Z_s(t)$ for each session s. Therefore, the average rate and the average hop requirements from all sessions are converted into the stability requirements for the virtual queues. For example, the virtual queue update of $Y_s(t)$ is given by (11). If the virtual queue Y_s is stable, the average arrival rate should be less than the average departure rate of the queue, that is,

$$\alpha_s \le \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} X_s(t),$$

which is exactly the minimum average rate requirement imposed by session s. By the same token, the average hop requirement of session s is converted to the stability problem of the virtual queue Z_s. Define $\overline{\boldsymbol{Q}}(t) = \big(\boldsymbol{Q}(t), \boldsymbol{Y}(t), \boldsymbol{Z}(t)\big)$, namely, all the data queues and the virtual queues in the network. Our objective is to find a policy which stabilizes the network with respect to $\overline{\boldsymbol{Q}}$ while maximizing the overall network utility.

We first take the square of (4) and have

$$\begin{aligned}(Q_{n,h}(t+1))^2 &\le (Q_{n,h}(t))^2 \\ &+ \Bigg(\sum_{(n,j)\in L} u_{n,j}^{n,h}(t)\Bigg)^2 + \Bigg(\sum_{(m,n)\in L} u_{m,n}^{m,h+1}(t) + \sum_s X_s^h(t)\delta_{n=n_s^0}\Bigg)^2 \\ &- 2Q_{n,h}(t)\Bigg(\sum_{(n,j)\in L} u_{n,j}^{n,h}(t) - \sum_{(m,n)\in L} u_{m,n}^{m,h+1}(t) - \sum_s X_s^h(t)\delta_{n=n_s^0}\Bigg).\end{aligned}$$

Since we assume that each node generates at most one session, we have

$$\sum_s X_s^h(t)\delta_{n=n_s^0} \le A_{\max}, \forall t, n.$$

Note that if we allow that a node can initiate multiple sessions, we have

$$\sum_s X_s^h(t)\delta_{n=n_s^0} \le |S|A_{\max}, \forall t, n,$$

where $|S|$ is the number of sessions in the network.

In light of (5) and (6), we have

$$\begin{aligned}(Q_{n,h}(t+1))^2 - (Q_{n,h}(t))^2 &\le (u_{out})^2 + (u_{in} + A_{\max})^2 \\ - \ 2Q_{n,h}(t)&\left(\sum_{(n,j)\in L} u_{n,j}^{n,h}(t) - \sum_{(m,n)\in L} u_{m,n}^{m,h+1}(t) - \sum_s X_s^h(t)\delta_{n=n_s^0} \right). \end{aligned} \tag{22}$$

We next sum (22) over all the data queues in the network, that is, $Q_{n,h}$, and have

$$\begin{aligned}\sum_{n,h}(Q_{n,h}(t+1))^2 - \sum_{n,h}(Q_{n,h}(t))^2 &\le B_1 \\ - \ 2Q_{n,h}(t)&\left(\sum_{(n,j)\in L} u_{n,j}^{n,h}(t) - \sum_{(m,n)\in L} u_{m,n}^{m,h+1}(t) - \sum_s X_s^h(t)\delta_{n=n_s^0} \right), \end{aligned} \tag{23}$$

where

$$B_1 = |N|(|N|-1)((u_{out})^2 + (u_{in} + A_{\max})^2).$$

Note that $M_s(t)$, defined in (8), satisfies

$$M_s(t) \le (|N|-1)^2 A_{\max}.$$

Next, we take the square of (11) and (12) and thus have

$$(Y_s(t+1))^2 \le (Y_s(t))^2 + (X_s(t))^2 + (\alpha_s)^2 - 2Y_s(t)(X_s(t) - \alpha_s)$$

and

$$(Z_s(t+1))^2 \le (Z_s(t))^2 + (M_s(t))^2 + (\beta_s)^2 - 2Z_s(t)(\beta_s - M_s(t)).$$

Similarly, we sum over all the sessions and have

$$\sum_s (Y_s(t+1))^2 - \sum_s (Y_s(t))^2 \le B_2 - 2\sum_s Y_s(t)(X_s(t) - \alpha_s)$$

where

$$B_2 = |S|(A_{\max})^2 + \sum_s (\alpha_s)^2.$$

Also, we obtain

$$\sum_s (Z_s(t+1))^2 - \sum_s (Z_s(t))^2 \le B_3 - 2\sum_s Z_s(t)(\beta_s - M_s(t)),$$

where

$$B_3 = \sum_s (\beta_s)^2 + |S|\,(|N|-1)^4\,(A_{\max})^2.$$

Define the system-wide Lyapunov function as

$$L(\overline{\mathbf{Q}}(t)) = \sum_{n,h} (Q_{n,h}(t))^2 + \sum_s (Y_s(t))^2 + \sum_s (Z_s(t))^2.$$

Next, we define the Lyapunov drift [Neely 2003] of the system as

$$\Delta = E\big(L(\overline{\mathbf{Q}}(t+1)) - L(\overline{\mathbf{Q}}(t)) \mid \overline{\mathbf{Q}}(t)\big). \tag{24}$$

Define

$$\bar{B} = B_1 + B_2 + B_3,$$

we have

$$\begin{aligned}\Delta \;\le\; & \bar{B} - 2\sum_{n,h} Q_{n,h}(t) E\Bigg(\sum_{(n,j)\in L} u_{n,j}^{n,h}(t) - \sum_{(m,n)\in L} u_{m,n}^{m,h}(t) - \sum_s X_s^h(t)\delta_{n=n_S^0} \Big| \overline{\mathbf{Q}}(t)\Bigg) \\ & - 2E\Bigg(\sum_s Y_s(t)(X_s(t)-\alpha_s) \Big| \overline{\mathbf{Q}}(t)\Bigg) - 2E\Bigg(\sum_s Z_s(t)(\beta_s - M_s(t)) \Big| \overline{\mathbf{Q}}(t)\Bigg).\end{aligned}$$

Next, we subtract both sides by $JE(\sum_s U_s(X_s(t))|\overline{\mathbf{Q}}(t))$ and have

$$\begin{aligned}& \Delta - JE\Bigg(\sum_s U_s(X_s(t)) \Big| \overline{\mathbf{Q}}(t)\Bigg) \le \bar{B} \\ & - 2\sum_{n,h} Q_{n,h}(t) E\Bigg(\sum_{(n,j)\in L} u_{n,j}^{n,h}(t) - \sum_{(m,n)\in L} u_{m,n}^{m,h+1}(t) \Big| \overline{\mathbf{Q}}(t)\Bigg) \\ & + 2E\Bigg(\sum_s \sum_h Q_{n_s^0,h}(t) X_s^h(t) \Big| \overline{\mathbf{Q}}(t)\Bigg) - 2E\Bigg(\sum_s Y_s(t)X_s(t) \Big| \overline{\mathbf{Q}}(t)\Bigg) + 2\sum_s Y_s(t)\alpha_s \\ & + 2E\Bigg(\sum_s Z_s(t)M_s(t) \Big| \overline{\mathbf{Q}}(t)\Bigg) - 2\sum_s Z_s(t)\beta_s - JE\Bigg(\sum_s U_s(X_s(t)) \Big| \overline{\mathbf{Q}}(t)\Bigg).\end{aligned} \tag{25}$$

We rewrite the R.H.S. of (25) as

$$\begin{aligned}\text{R.H.S.} = {} & \bar{B} + 2\sum_s Y_s(t)\alpha_s - 2\sum_s Z_s(t)\beta_s \\ & - 2\sum_{n,h} Q_{n,h}(t) E\Bigg(\sum_{(n,j)\in L} u_{n,j}^{n,h}(t) - \sum_{(m,n)\in L} u_{m,n}^{m,h+1}(t) \Big| \overline{\mathbf{Q}}(t)\Bigg) \\ & - E\Bigg(\sum_s 2Y_s(t)X_s(t) - \sum_s 2Z_s(t)M_s(t) - \sum_s \sum_h 2Q_{n_s^0,h}(t)X_s^h(t) \\ & + J\sum_s U_s(X_s(t)) \Big| \overline{\mathbf{Q}}(t)\Bigg).\end{aligned}$$

We observe that the dynamic routing and scheduling component of the ANRA scheme is actually maximizing

$$\sum_{n,h} Q_{n,h}(t) E\Bigg(\sum_{(n,j)\in L} u_{n,j}^{n,h}(t) - \sum_{(m,n)\in L} u_{m,n}^{m,h+1}(t) \Big| \overline{\mathbf{Q}}(t) \Bigg). \tag{26}$$

In addition, the joint rate admission control and traffic splitting component of the ANRA scheme is essentially maximizing

$$E\Bigg(\sum_{s} 2Y_s(t)X_s(t) - \sum_{s} 2Z_s(t)M_s(t) - \sum_{s}\sum_{h} 2Q_{n_s^0,h}(t)X_s^h(t) + J\sum_{s} U_s(X_s(t)) \Big| \overline{\mathbf{Q}}(t) \Bigg) \tag{27}$$

with the constraints of

$$\sum_{h} X_s^h(t) = X_s(t), \forall s, t. \tag{28}$$

To see this, we can decompose (27) to show that each session s only maximizes

$$JU_s\left(X_s(t)\right) + 2Y_s(t)X_s(t) - 2Z_s(t)\sum_{h} hX_s^h(t) - \sum_{h} 2Q_{n_s^0,h}(t)X_s^h(t). \tag{29}$$

Therefore, the proposed ANRA algorithm indeed minimizes the R.H.S. of (25) over all policies.

Consider a reduced network capacity region, denoted by Λ_ϵ, parameterized by $\epsilon > 0$, as

$$\{\boldsymbol{\lambda} | \lambda_{n,h} + \epsilon \in \Lambda\}, \tag{30}$$

where Λ is the original network capacity region and

$$\lambda_{n,h} = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T-1} \sum_{s} X_s^h(t)\delta_{n=n_s^0}. \tag{31}$$

Define U_ϵ^* as the global optimum network utility achieved in the reduced capacity region. Apparently, we have $\lim_{\epsilon\to 0} U_\epsilon^* \to U^*$. In addition, denote $X_{s,\epsilon}^{h*}(0)$, $X_{s,\epsilon}^{h*}(1), \ldots, X_{s,\epsilon}^{h*}(t), \ldots$ as the optimum rate sequence which yields U_ϵ^*. Define $\bar{X}_s^\epsilon$ as the average of the optimum rate sequence of session s, in the reduced capacity region. It is straightforward to verify that $\bar{X}_s^\epsilon + \epsilon$ is in the original network capacity region Λ. Therefore, following a similar analysis as in Neely [2003], Neely et al. [2005], Neely et al. [2008], and Sharma et al. [2009], we claim that there exists a randomized policy, denoted by R, which generates

$$E\Bigg(\sum_{(n,j)\in L} u_{n,j}^{n,h}(t) - \sum_{(m,n)\in L} u_{m,n}^{m,h+1}(t) \Bigg) \geq X_{s,\epsilon}^{h*} + \epsilon \tag{32}$$

if n is one of the source nodes and

$$E\Bigg(\sum_{(n,j)\in L} u_{n,j}^{n,h}(t) - \sum_{(m,n)\in L} u_{m,n}^{m,h+1}(t) \Bigg) \geq \epsilon \tag{33}$$

for other nodes. Furthermore, policy R ensures

$$E\Big(\sum_h X_{s,\epsilon}^{h*}(t) \geq \alpha_c + \epsilon\Big)$$

and

$$E(M_{s,\epsilon}^*(t) + \epsilon \leq \beta_s),$$

where $M_{s,\epsilon}^*(t)$ is generated by $X_{s,\epsilon}^{h*}(t)$. Due to the fact that the proposed ANRA scheme minimizes the R.H.S. of (25) overall all policies, including R, we have

$$\begin{aligned}
&\Delta - JE\Big(\sum_s U_s(X_s(t))\Big|\overline{\mathbf{Q}}(t)\Big)\\
&\leq \bar{B} - 2\epsilon\Big(\sum_{n,h} Q_{n,h}(t) + \sum_s Y_s(t) + \sum_s Z_s(t)\Big) - JE\Big(\sum_s U_s(\sum_h X_{s,\epsilon}^{h*}(t) + \epsilon)\Big|\overline{\mathbf{Q}}(t)\Big).
\end{aligned}$$

We next take the expectation with respect to $\overline{\mathbf{Q}}(t)$ and obtain

$$\begin{aligned}
&L(\overline{\mathbf{Q}}(t+1)) - L(\overline{\mathbf{Q}}(t)) - JE\Big(\sum_s U_s(X_s(t))\Big)\\
&\leq \bar{B} - 2\epsilon E\Big(\sum_{n,h} Q_{n,h}(t) + \sum_s Y_s(t) + \sum_s Z_s(t)\Big)\\
&-JE\Big(\sum_s U_s(\sum_h X_{s,\epsilon}^{h*}(t) + \epsilon)\Big).
\end{aligned} \tag{34}$$

We sum over time slots $0, \ldots, T-1$ and have

$$\begin{aligned}
&L(\overline{\mathbf{Q}}(T)) - L(\overline{\mathbf{Q}}(0)) - \sum_{t=0}^{T-1} JE\Big(\sum_s U_s(X_s(t))\Big)\\
&\leq T\bar{B} - \sum_{t=0}^{T-1} JE\Big(\sum_s U_s\Big(\sum_h X_{s,\epsilon}^{h*}(t) + \epsilon\Big)\Big)
\end{aligned} \tag{35}$$

since

$$E\left(\sum_{n,h} Q_{n,h}(t) + \sum_s Y_s(t) + \sum_s Z_s(t)\right)$$

is always nonnegative. Next, we divide the both sides of (35) by T and rearrange terms to have

$$\frac{1}{T}\sum_{t=0}^{T-1} JE\Big(\sum_s U_s(X_s(t))\Big) \geq \frac{1}{T}\sum_{t=0}^{T-1} JE\Big(\sum_s U_s\Big(\sum_h X_{s,\epsilon}^{h*}(t) + \epsilon\Big)\Big) - \bar{B} - \frac{L(\overline{\mathbf{Q}}(0))}{T},$$

where the nonnegativity of the Lyapunov function is utilized. Since we assume that the initial queue backlogs in the system are finite and the virtual queues are initially empty, taking $\epsilon \to 0$ and $\liminf_{T\to\infty}$ yields the performance result of the ANRA algorithm stated in Theorem 2.

We next show that the constraints of the SNUM problem are also satisfied. To illustrate this, we show that the queues in the network, including real data queues and virtual queues, are stable. Based on (34), we sum over time slots $0, \ldots, T-1$ and have

$$\sum_{t=0}^{T-1} 2\epsilon E\Big(\sum_{n,h} Q_{n,h}(t) + \sum_s Y_s(t) + \sum_s Z_s(t)\Big) \leq L(\overline{\mathbf{Q}}(0)) + \sum_{t=0}^{T-1} JE\Big(\sum_s U_s(X_s(t))\Big) + T\bar{B}. \quad (36)$$

Due to $X_s(t) \leq A_{\max}$ and the assumptions on the utility function, we claim that $U_s(t)$ is upper bounded and denote the maximum utility within one time slot as $U_{\max}$, that is,

$$U_s(t) \leq U_{\max}, \forall s, t. \quad (37)$$

Divide the both sides of (36) by T and we have

$$\frac{1}{T}\sum_{t=0}^{T-1} 2\epsilon E\Big(\sum_{n,h} Q_{n,h}(t) + \sum_s Y_s(t) + \sum_s Z_s(t)\Big) \leq \frac{L(\overline{\mathbf{Q}}(0))}{T} + J|S|U_{\max} + \bar{B}.$$

By taking $\limsup_{T\to\infty}$, we have

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} E\Big(\sum_{n,h} Q_{n,h}(t) + \sum_s Y_s(t) + \sum_s Z_s(t)\Big) \leq \frac{J|S|U_{\max} + \bar{B}}{2\epsilon}. \quad (38)$$

Note that the preceding analysis holds for any feasible value of ϵ. Denote φ as the maximum value of ϵ such that Λ_φ is not empty. Finally, we conclude that

$$\limsup_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} E\Big(\sum_{n,h} Q_{n,h}(t) + \sum_s Y_s(t) + \sum_s Z_s(t)\Big) \leq \frac{J|S|U_{\max} + \bar{B}}{2\varphi} < \infty. \quad (39)$$

The stability of the network follows immediately from Markov's Inequality and thus completes the proof.

It is worth noting that as shown in (39), a large value of J induces a longer average queue size in the network. Therefore, a trade-off between the algorithm performance of the ANRA scheme and the average delay experienced in the network can be controlled effectively by tuning the value of J.

6. CASE STUDY

In this section, we demonstrate the effectiveness of the ANRA algorithm numerically through a simple network shown in Figure 2. We stress that, however, this exemplifying study case reproduces all the challenging problems involved in the complex cross-layer interactions in time-varying environments, such as stochastic traffic arrivals, random channel conditions, and dynamic routing and scheduling, etc. As shown in Figure 2, the source nodes in the network are node A and B whereas the destination sink nodes are denoted by E and F.

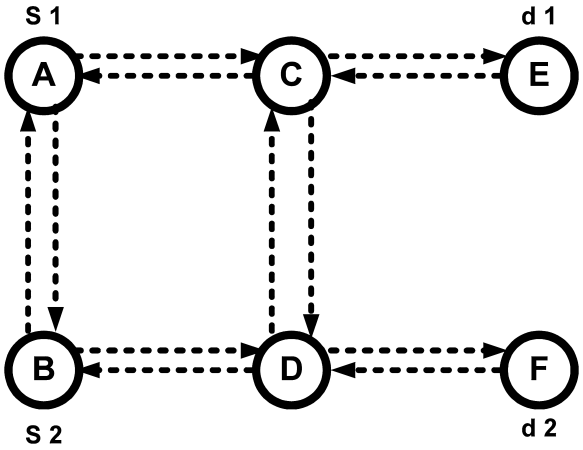

Fig. 2. Example network.

There are six nodes and twelve links in the network. Therefore, node A and B each maintains four queues, from hop 2 to hop 5, and node C and D each maintains five queues, from hop 1 to hop 5, in the buffer. At each time slot, a wireless link is assumed to have three equally possible data rates[5], 2, 8 and 16. The traffic arrivals are independently and identically distributed with three equally possible states, that is, 0, 10, and 20. The minimum rate requirements of the two sessions are 5 and 8 and the average hop requirements of the sessions are 30 and 10. Without loss of generality, we assume that at a given time slot, two links with a common node cannot be active simultaneously. For example, if link $A \rightarrow B$ is active, link $B \rightarrow A$, $A \rightarrow C$, $C \rightarrow A$, $B \rightarrow D$ and $D \rightarrow B$ cannot be selected.

Figure 3 depicts the average network utility achieved by the ANRA scheme for different values of J where each experiment is executed for 50000 time slots. We can observe from Figure 3 that the overall network utility rises as the value of J increases. However, the speed of utility improvement decreases and the achieved network utility converges to the global optimum utility U^* gradually. It is worth noting that in practice, the value of U^* cannot be attained efficiently without knowing the underlying statistical characteristics. However, the proposed ANRA scheme can achieve a solution which is arbitrarily close to the global optimum solution with no such information required. To demonstrate the trade-off of different values of J, in Figure 4, we show the average queue size in the network for $J = 20, 50, 200, 500, 1000, 2000, 5000, 10000$, and 20000. We can see that, as expected, the average queue size increases as the value of J gets larger. Note that the average queue size is related to the average delay in the network. Therefore, a trade-off between the network optimality and the average experienced delay can be achieved by tuning the value of J.

In Figure 5, we illustrate the sample trajectories of the admitted rates of two sessions with $J = 5000$, for the first 50 time slots. We can observe that each session admits different amount of packets into the network adaptively

[5]Note that the unit of data transmissions is packet per slot.

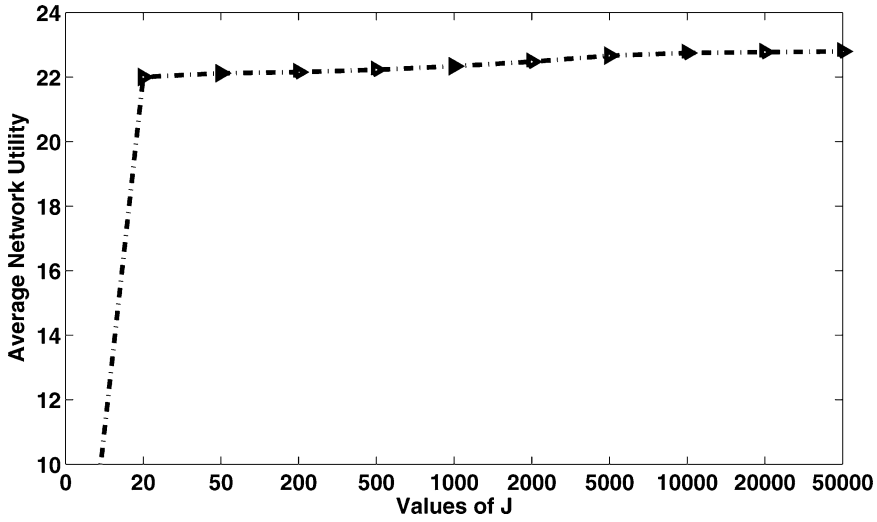

Fig. 3. Average network utility achieved by ANRA for different values of J.

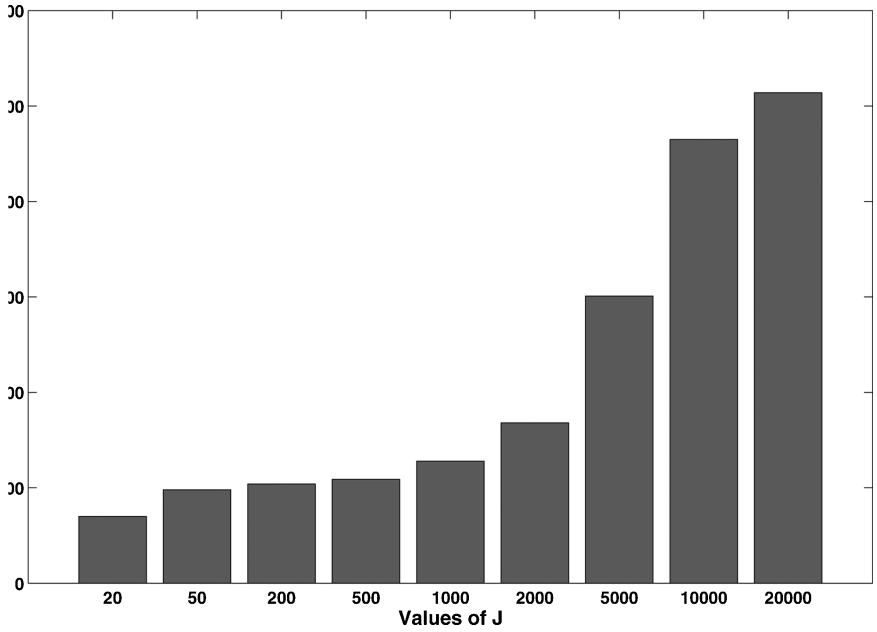

Fig. 4. Average network queue size by ANRA for different values of J.

following the time-varying conditions of the network. In addition, we depict the trajectories of the four virtual queues with the same settings, in Figure 6, for the first 100 time slots. By comparing Figure 5 and Figure 6 jointly, we can observe that for the minimum rate virtual queue, say Y_1, whenever there is the tendency that the virtual queue is accumulating, as depicted in Figure 6, the corresponding admitted rate by session 1 increases in Figure 5.

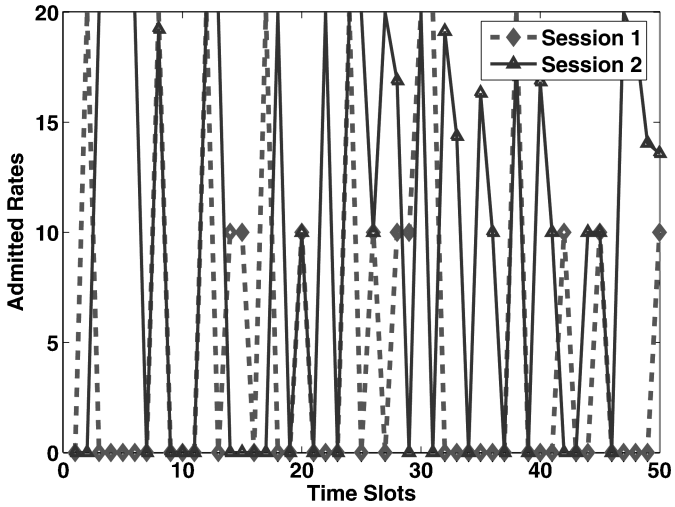

Fig. 5. Sample trajectories of the admitted rates of two sessions for $J = 5000$.

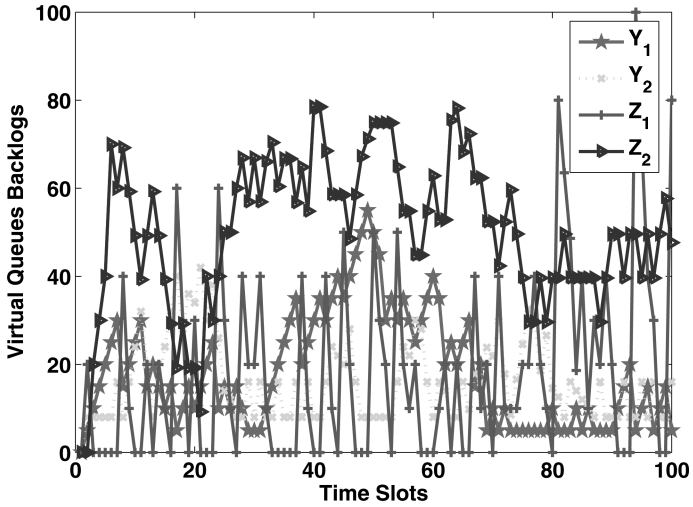

Fig. 6. Sample trajectories of the virtual queues for $J = 5000$.

By the definition of the virtual queue, a larger backlog of Y_1 indicates that the average departure rate of the virtual queue, that is, the average admitted rate, is insufficient. Therefore, the source node of session 1 will attempt to increase the admitted rate and thus the backlog of the virtual queue will decrease accordingly where the stability of the virtual queue can be assured.

In Figure 7, the traffic splitting decisions of the two source nodes, that is, the hop selections of the source nodes, are illustrated. We can observe that both source nodes incline to utilize the queues with the smaller number of hops. The queues with longer hops, for example, $h = 3$ or 4, are used only when the queue backlogs in the queues with smaller hops are overwhelmed. In addition, we can see that on average, session 2 utilizes a smaller number of average hops

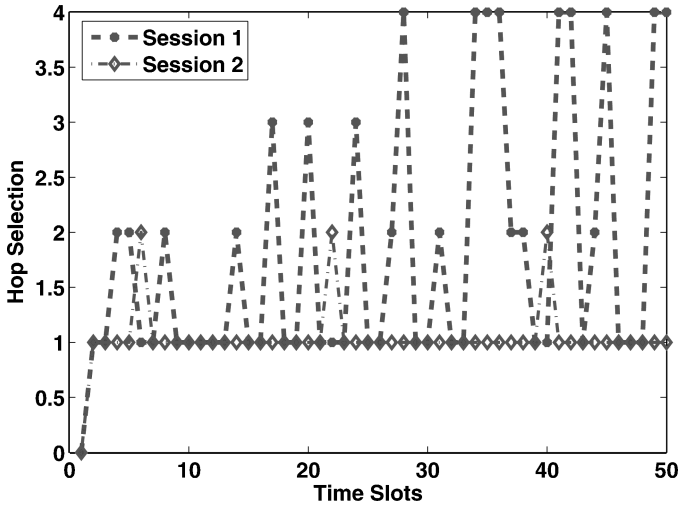

Fig. 7. Sample trajectories of the hop selections for $J = 5000$.

than session 1. Recall that session 2 has a much more stringent constraint on the average number of hops than session 1, that is, 10 versus 30. Therefore, the source node of session 2, that is, n_2^0, inclines to deposit more packets on the queues with smaller hop counts. As a consequence, by assigning different values of rate and hop requirements, a service differentiation solution can be achieved by the ANRA scheme among multiple competitive sessions in the network. In addition, a near-optimal network utility can be attained simultaneously.

7. CONCLUSIONS AND FUTURE WORK

In this article, we propose a constrained queueing model to capture the cross-layer interactions in multihop wireless sensor networks. Our model consists of components from multiple layers such as rate admission control, dynamic routing, and wireless link scheduling. Based on the proposed model, we investigate the stochastic network utility maximization problem in wireless sensor networks. As a cross-layer solution, an adaptive network resource allocation scheme, called the ANRA algorithm, is proposed. The ANRA algorithm is an online mechanism which yields an overall network utility that can be pushed arbitrarily close to the global optimum solution.

As a future work, energy-aware distributed scheduling algorithms are to be studied and evaluated. In addition, the extension of our model to wireless sensor networks with network coding seems interesting and needs further investigation.

REFERENCES

AKYILDIZ, I. F. AND KASIMOGLU, I. H. 2004. Wireless sensor and actor networks: Research challenges. *Elsevier Comput. Networks*.

AKYOL, U., ANDREWS, M., GUPTA, P., HOBBY, J., SANIEE, I., AND STOLYAR, A. 2008. Joint scheduling and congestion control in mobile ad-hoc networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*.

CHIANG, M. 2005. Balancing transport and physical layer in wireless multihop networks: Jointly optimal congestion control and power control. *IEEE J. Select. Areas Comm. 1*, 104–116.

CHIANG, M., LOW, S. H., CALDERBANK, A. R., AND DOYLE, J. C. 2007. Layering as optimization decomposition: A mathematical theory of network architectures. *Proc. IEEE 95*, 255–312.

ERYILMAZ, A. AND SRIKANT, R. 2005. Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*.

GEORGIADIS, L., NEELY, M. J., AND TASSIULAS, L. 2006. *Resource Allocation and Cross-Layer Control in Wireless Networks*. Foundations and Trends in Networking.

GUPTA, A., LIN, X., AND SRIKANT, R. 2007. Low-complexity distributed scheduling algorithms for wireless networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*.

GUPTA, G. R. AND SHROFF, N. 2009. Delay analysis for multi-hop wireless networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*.

JIANG, L. AND WALRAND, J. 2008. A distributed csma algorithm for throughput and utility maximization in wireless networks. In *Proceedings of the Allerton Conference of Communication Control, and Computing*.

JOO, C. 2008. A local greedy scheduling scheme with provable performance guarantee. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*.

KELLY, F., MAULLOO, A., AND TAN, D. 1998. Rate control for communication networks: Shadow prices, proportional fairness and stability. *J. Oper. Res. Soc. 49*, 237–252.

LIN, X. 2006. On characterizing the delay performance of wireless scheduling algorithms. In *Proceedings of the Allerton Conference of Communication Control, and Computing*.

LOW, S. H. AND LAPSLEY, D. E. 1999. Optimization flow control, i: Basic algorithm and convergence. *IEEE/ACM Trans. Netw. 7*, 861–875.

MODIANO, E., SHAH, D., AND ZUSSMAN, G. 2006. Maximizing throughput in wireless networks via gossiping. In *Proceedings of the ACM SIGMETRICS Conference*.

NEELY, M. J. 2003. Dynamic power allocation and routing for satellite and wireless networks with time varying channels. Ph.D. thesis, Masssachusetts Institute of Technology.

NEELY, M. J. 2006. Energy optimal control for time varying wireless networks. *IEEE Trans. Inf. Theory 52*, 2915–2934.

NEELY, M. J., MODIANO, E., AND LI, C.-P. 2008. Fairness and optimal stochastic control for heterogeneous networks. *IEEE/ACM Trans. Netw. 16*, 396–409.

NEELY, M. J., MODIANO, E., AND ROHRS, C. E. 2005. Dynamic power allocation and routing for time-varying wireless networks. *IEEE J. Select. Areas Comm. 23*, 89–103.

RADUNOVIC, B., GKANTSIDIS, C., GUNAWARDENA, D., AND KEY, P. 2008. Horizon: Balancing tcp over multiple paths in wireless mesh network. In *Proceedings of the 14th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiCom)*.

SHAKKOTTAI, S. AND SRIKANT, R. 2008. *Network Optimization and Control*. Foundations and Trends in Networking.

SHARMA, A. B., GOLUBCHIK, L., GOVINDAN, R., AND NEELY, M. J. 2009. Dynamic data compression in multi-hop wireless networks. In *Proceedings of the ACM SIGMETRICS Conference*.

SONG, Y. AND FANG, Y. 2007. Distributed rate control and power control in resource-constrained wireless sensor networks. In *Proceedings of the IEEE MILCOM Conference*.

SRIKANT, R. 2003. *The Mathematics of Internet Congestion Control*. Birkhauser Boston.

STOLYAR, A. 2005. Maximizing queueing network utility subject to stability: Greedy primal-dual algorithm. *Queu. Syst. 50*, 401–457.

STOLYAR, A. 2006. Large deviations of queues under qos scheduling algorithms. In *Proceedings of the Allerton Conference of Communication Control, and Computing*.

STOLYAR, A. 2008. Dynamic distributed scheduling in random access networks. *J. Appl. Probab. 45*, 297–313.

TASSIULAS, L. AND EPHREMIDES, A. 1992. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Trans. Autom. Control 37*, 1936–1949.

TRENCH, W. F. 2003. *Introduction to Real Analysis*. Prentice Hall.

WU, X., SRIKANT, R., AND PERKINS, J. 2007. Scheduling efficiency of distributed greedy scheduling algorithms in wireless networks. *IEEE Trans. Mobile Comput.*.

YICK, J., MUKHERJEE, B., AND GHOSAL, D. 2007. Wireless sensor network survey. *Elsevier Comput. Networks*.

YING, L., SHAKKOTTAI, S., AND REDDY, A. 2009. On combining shortest-path and back-pressure routing over multihop wireless networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (InfoCom)*.

Received May 2009; revised September 2009; accepted October 2009