3-2012

Utility-based Bandwidth Adaptation in Mission-Oriented Wireless Sensor Networks

Sharanya ESWARAN
Telcordia Technologies

Archan MISRA
Singapore Management University, archanm@smu.edu.sg

Flavio BERGAMASCHI
IBM UK

Thomas LA PORTA
Pennsylvania State University

DOI: https://doi.org/10.1145/2140522.2140530

Published in ACM Transactions on Sensor Networks, Volume 8 Issue 2, March 2012, Article 17.
http://doi.org/10.1145/2140522.2140530

Utility-Based Bandwidth Adaptation in Mission-Oriented Wireless Sensor Networks

SHARANYA ESWARAN, Telcordia Technologies
ARCHAN MISRA, Singapore Management University
FLAVIO BERGAMASCHI, IBM, U.K.
THOMAS LA PORTA, Pennsylvania State University

This article develops a utility-based optimization framework for resource sharing by multiple competing missions in a mission-oriented wireless sensor network (WSN) environment. Prior work on network utility maximization (NUM) based optimization has focused on unicast flows with sender-based utilities in either wireline or wireless networks. In this work, we develop a generalized NUM model to consider three key new features observed in mission-centric WSN environments: i) the definition of the utility of an individual mission (receiver) as a joint function of data from multiple sensor sources; ii) the consumption of each sender's (sensor) data by multiple missions; and iii) the multicast-tree-based dissemination of each sensor's data flow, using link-layer broadcasts to exploit the "wireless broadcast advantage" in data forwarding. We show how a price-based, distributed protocol (WSN-NUM) can ensure optimal and proportionally fair rate allocation across multiple missions, without requiring any coordination among missions or sensors. We also discuss techniques to improve the speed of convergence of the protocol, which is essential in an environment as dynamic as the WSN. Further, we analyze the impact of various network and protocol parameters on the bandwidth utilization of the network, using a discrete-event simulation of a stationary wireless network. Finally, we corroborate our simulation-based performance results of the WSN-NUM protocol with an implementation of an 802.11b network.

Categories and Subject Descriptors: C.2.2 [**Computer-Communication Networks**]: Network Protocols

General Terms: Algorithms, Design, Experimentation

Additional Key Words and Phrases: Utility optimization, bandwidth allocation, modeling of systems, network protocols, congestion control

---

This research was sponsored by U.S. Army Research Laboratory and the U.K. Ministry of Defence and was accomplished under Agreement Number W911NF-06-3-0001. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Army Research Laboratory, the U.S. Government, the U.K. Ministry of Defense, or the U.K. Government. The U.S. and U.K. Governments are authorized to reproduce and distribute reprints for Government purposes with copyright attribution.
Author's address: S. Eswaran, Advanced Technology Solutions, Telcordia Technologies, 344 IST Building, University Park, PA 16802; email: eswaran@cse.psu.edu; A. Misra, School of Information Systems, Singapore Management University; F. Bergamaschi, IBM, U.K.; T. La Porta, The Pennsylvania State University.

1. INTRODUCTION

Sensor-generated information flows are becoming critical for improved situational awareness in a variety of battlefield environments. The sensor devices may, themselves, be viewed as a shared resource with a common network substrate transporting the sensed data for consumption by diverse applications. Traditionally, research on WSNs has primarily focused on energy management, especially for long-running passive applications, such as habitat monitoring, etc. However, a class of tactical applications, such as battlefield monitoring, vehicular tracking, and emergency disaster response, requires a WSN to be operational for smaller time periods (i.e., days rather than months), employs more sophisticated, expensive, and higher data rate sensors (e.g., video, short-aperture radar, and acoustic sensors) and is more concerned with managing the limited network bandwidth, rather than low-energy, long-duration operation. It is thus important to develop a scalable resource-sharing framework that optimizes the transport of various types of sensor data (e.g., acoustic, video, thermal) to a concurrent, competing set of missions. In this article, we develop a utility-based, cross-layer optimization framework for maximizing the dissemination of elastic sensor flows over a wireless sensor network (WSN), while taking into account the application-level requirements of the missions.

The network utility maximization (NUM) problem and its distributed implementation have been extensively studied as a resource allocation mechanism for wireline [Kelly 1997; Low and Lapsley 1999; Chiang et al. 2007; La and Anantharam 2002] and ad hoc wireless networks [Chiang 2005; Wang and Kar 2006; Lin and Shroff 2004; Chen et al. 2005, 2006; Eryilmaz and Srikant 2006]. A WSN designed for tactical military missions, however, possesses several distinct characteristics and challenges that preclude the direct applicability of prior source-centric NUM algorithms.

—*Joint Utility Functions*. A mission's utility is often derived from multiple sensor flows rather than a single sensor source. The utility function associated with such a composite-utility model implies that it is not possible to articulate a mission's benefits from a specific sensor independently of the data rates that it simultaneously receives from other sensors. More importantly, this provides the network an additional degree of freedom by allowing it to trade off the rates of one sensor vis-à-vis another without necessarily impacting the resultant application-level utility. As a practical example, an intrusion-detection application may utilize video feeds from two overlapping video sensors and may be more or less sensitive to the resolution of one such camera, depending on the current image quality it is receiving from the other video sensor.

—*Multi-Sink Flows*. The presence of multiple concurrent missions also implies that the data from an individual sensor node may be consumed by multiple receivers (missions), each with a distinct perception of the sensor's utility to the mission. Accordingly, we transform the utility model to become receiver-centric, as it is infeasible to associate a receiver-agnostic utility function with a sender. We shall see that the consequent interplay among the utilities of different receivers makes it trickier to develop a distributed optimization technique that drives the network's collective utility towards the global optimum.

—*High Variability*. The duration of WSN-based military missions can vary widely, ranging from a few seconds in the case of gunfire localization to a few hours or more for perimeter surveillance. Moreover, the wireless network topology itself may be highly dynamic with link capacities and connectivity fluctuating over medium time scales due to mobility. Any practical, distributed adaptation technique must thus exhibit fast convergence, that is, reacting quickly to changes in the operational parameters. Moreover, the adaptation protocol should be relatively lightweight, incurring low communication and computation overhead.

Based on these observations, we first formulate the utility maximization problem for the generalized case of a set of receivers with joint (multidimensional) utility functions, subscribing to feeds from a set of common sensors, with the feeds delivered via link-layer multicast over a wireless network. In particular, the solution to the primal NUM problem requires us to combine several previously isolated features, such as *a)* receiver-based utility specification (similar to Shapiro et al. [2002], Kar et al. [2002], Bui et al. [2007]) and *b)* association of capacity constraints with maximal cliques in *transmission-specific conflict graphs* (defined in Sengupta et al. [2007]), in a comprehensive way. More importantly, we then provide a protocol-level implementation of the optimization technique that addresses several practical challenges to applying the NUM framework in a realistic, distributed WSN environment. Prior work on NUM-based optimization algorithms typically fails to analyze the practical applicability of the protocol. We provide the first 802.11b-based simulation of the WSN-NUM protocol and further address the impact of network and model-related aspects on the protocol performance.[1] We further extend this work by providing an actual implementation of the protocol in an 802.11b ad hoc network and corroborate our observations from simulation using the implementation. Through the implementation, we show the practical applicability of WSN-NUM protocol for bandwidth management by comparing with the simulation results.

Specifically, we make the following contributions in this article.

(1) We develop the generalized NUM model for WSNs with joint utility functions and multi-sink flows and show how a receiver-centric model of marginal utility adaptation, in conjunction with a modified rate adaptation technique at sensor nodes and clique-based congestion pricing, can achieve globally optimal rate control.
(2) We develop a comprehensive protocol that addresses the practical aspects of the NUM framework and evaluate the protocol using discrete-event simulations in an 802.11 environment to establish the practically observed overheads and performance of the theoretical optimization model. We also analyze the impact of various network- and protocol-related parameters on the performance of the protocol.
(3) We provide a modification to the basic gradient-based rate adaptation algorithm and demonstrate how the modified rate adaptation technique can provide rapid convergence under high mission dynamics.
(4) We corroborate and compare our simulation results with a real-time implementation of the protocol and provide insights into additional practical challenges and limitations of the protocol.

The rest of this article is organized as follows. Section 2 discusses related work in wireline, ad hoc wireless, and wireless sensor networks. In Section 3, we introduce the WSN-NUM Model that reflects the features of multicast-based sensor feed dissemination over a WSN and describe the distributed optimization algorithm and protocol. In Section 4, we discuss the techniques for solving several of the practical protocol-level challenges of the generalized NUM technique. Section 5 presents a discrete-event simulation of the protocol and its evaluation. Subsequently, in Section 6, we analyze the impact of network- and model-related parameters on the performance of the protocol and also introduce techniques that provide for faster convergence. In Section 7, we describe the implementation details and present the results of implementation. Finally, Section 8 concludes the article.

[1]Preliminary work on these can be found in Eswaran et al. [2008a, 2008b].

2. RELATED WORK

There is a wealth of literature on optimization-based techniques for rate control in communication networks, an approach first introduced by Kelly [1997] for the case of wired links. In this model, each source node s is associated with a concave, nondecreasing utility function $U_s(x_s)$, which depends on the source's transmission rate x_s. The network is modeled as a set of links, and each flow is a collection of links. Under this model, the NUM problem can be expressed as

SYSTEM(U, A, C) :

$$\text{maximize} \sum_{s \in S} U_s(x_s) \text{ subject to } \quad Ax \leq C \text{ over } x \geq 0, \tag{1}$$

where A is a binary routing matrix (with $A_{jk} = 1$, if source k's flow uses link j); x is the vector of source rates; and C is the vector of link capacities. Strict concavity of the utility functions $U_s(.)$ ensures that the problem $SYSTEM(U, A, C)$ has a unique optimal solution. As shown in Kelly [1997], the centralized *SYSTEM* problem can be decomposed into distinct independent problems by introducing a pricing scheme. In particular, the decentralized optimization approach behaves as if each user is charged a price by the network and consequently obtains a rate proportional to the amount it pays. Each user s has its own 'willingness to pay' w_s and receives a flow rate of x_s in return, proportional to the charge per unit flow, $\lambda_s = w_s/x_s$. The charge per unit flow on a path is the sum of the per-unit charge on each constituent link j, with this charge depending on the total traffic load on that link. In particular, if each source iteratively adjusts its rate according to a gradient-based differential equation,

$$\frac{d}{dt}x_s(t) = k\left(w_s - x_s(t) \sum_{j \in \text{ path of s}} \mu_j(t)\right), \tag{2}$$

$$\text{where } \mu_j(t) = p_j\left(\sum_{n: j \in \text{ path of n}} x_n(t)\right),$$

where k is a scalar constant, p_j is a cost function for link j which depends on the total traffic flowing through j, and μ_j is the 'price' charged by link j; and each source also periodically adjusts its "willingness to pay", according to

$$w_s(t) = x_s(t)U_s'(x_s(t)), \tag{3}$$

then the system will converge to a unique stable point that maximizes a relaxation of the *SYSTEM* problem. The same problem has also been solved using a dual approach in Low and Lapsley [1999], while La and Anantharam [2002] presents an Internet congestion control-based technique that solves the NUM problem by using queuing delay as an implicit congestion indicator.

Subsequently, there has been substantial research in the use of optimization-based techniques for not only rate control but other problems, such as power control and scheduling. See Chiang et al. [2007] for a detailed survey of these optimization techniques. In particular, Kar et al. [2002] addresses bandwidth allocation among competing multicast sessions in a wired network. Similarly, Shapiro et al. [2002] investigated the NUM problem for single-rate multicast traffic and introduced the receiver-centric model of NUM adaptation, again for a wired network.

2.1. Wireless Networks

More recently, price-based resource allocation and congestion control mechanisms have been used for cross-layer optimizations in ad hoc wireless networks, where the capacity of a link between two nodes also depends on the load of other interfering links. Cross-layer optimization in wireless networks has been studied in Chiang [2005], Wang and Kar [2006], Lin and Shroff [2004], Chen et al. [2005], Eryilmaz and Srikant [2006], and [Chen et al. 2006], albeit for unicast flows. More recently, the problem of NUM-based optimization for multicast traffic in wireless networks has been solved in Bui et al. [2007], which extends the wired multicast solutions in Shapiro et al. [2002] and Kar et al. [2002] to consider both single-rate and multi-rate sessions (where each receiver of a session may receive different rates).

Our NUM problem has strong similarities with Bui et al. [2007], as we also use that at both network layer and link-layer wireless multicast for disseminating a sensor's feed to multiple missions. Accordingly, we borrow the technique of receiver-centric adaptation from Bui et al. However, unlike Bui et al., where the focus is on fair sharing among different multicast sessions, we allow for receiver heterogeneity (i.e., different receivers receiving the same data rate may profess different utilities).

Contention-based constraints (on the rates achievable by multi-hop unicast flows) were captured in Xue et al. [2006] by associating shadow costs with maximal cliques. This notion of cliques was generalized to the case of link-layer multicast transmissions in Sengupta et al. [2007], where the conflict graph was defined with individual (node, flow) transmissions as vertices. We shall use this model to capture our wireless capacity constraints.

In Tan et al. [2006], a distributed optimization protocol for systems with coupled utilities is developed. This model can be applied to applications such as wireless power control, DSL spectrum management, etc. The utility of a user is a function of its own rate and also other source rates that may diminish its utility due to competition (e.g., interference). In our work, the utility of a mission is implicitly an increasing function of each of the sensor rates, as a mission's utility only improves as the volume of data from its sensors increases. We also do not require explicit message exchanges among missions, as their utilities are decoupled. Another difference is that we do not require our joint utility function to be linearly decomposable.

In Curescu and Nadjm-Tehrani [2008], a bidding algorithm for bandwidth allocation in wireless ad hoc networks is developed. This work, like ours, is inspired by the seminal works of Kelly [1997], Kelly et al. [1998], Low and Lapsley [1999], and Xue et al. [2006], for optimal, utility-based resource allocation. However, there are several important differences between Curescu and Nadjm-Tehrani [2008] and our work. First, the model developed in Curescu and Nadjm-Tehrani [2008] is for unicast networks and, hence, does not capture joint utilities and many-to-many transmissions. Second, the model makes use of link-based conflict graph, as in Xue et al. [2006], whereas we develop transmission-based conflict graphs to leverage the wireless broadcast advantage. Our protocol is fully distributed, while the protocol in Curescu and Nadjm-Tehrani makes use of clique leaders which collect the bids, which act as a point of centralization and also incur additional overhead for leader election. Furthermore, our protocol allows the rate allocation to be unsynchronized without any additional book-keeping. We also discuss the effects of the various parameters used in the model on the network performance and, additionally, provide techniques to adapt the step-size and congestion-tolerance limit dynamically, as well as a real-time implementation of the protocol.

2.2. Wireless Sensor Networks

The relatively little work on NUM-based optimization techniques specifically for WSNs focuses on unicast flows and does not consider the possibility of joint utility functions.

Table I. Most Common Mathematical Symbols

M	Set of all missions	S	Set of all sources
$sset(m)$	Set of sources subscribed by mission m	$mset(s)$	Set of missions receiving flow s
Q	Set of all maximal cliques in conflict graph	$clique(s)$	Set of cliques containing flow s
(k, s)	Transmission of flow s from node k	c_{ks}	Link rate for transmission (k, s)
w_{ms}	Willingness to pay of mission m for flow s	x_s	Data rate of source (flow) s
μ_q	Shadow cost of congestion at clique q	$U_m(.)$	Utility of mission m
κ	Step-size for rate adjustment	ϵ	Congestion tolerance limit
λ	Shadow charge per unit flow		

For a tree-shaped WSN topology in which all nodes transmit to the root, Sridharan and Krishnamachari [2007] showed how the NUM problem with max-min fair rate allocation can be modeled as two linear subproblems—one that determines the optimal max-min rate allocation and one that maximizes the aggregate rate while meeting the constraint obtained from the first problem. Similarly, Rangwala et al. [2006] developed an AIMD algorithm for interference-aware max-min rate control for a tree-shaped WSN topology that implicitly solves the associated NUM problem.

3. NETWORK MODEL AND OPTIMIZATION MODEL FOR WSN WITH JOINT UTILITIES

We first develop a receiver-centric NUM-based optimization model for mission-oriented WSN with joint utilities, which ensures optimal bandwidth allocation and congestion control when there is no in-network data processing. Accordingly, we consider the new environment of mission-based WSNs, where each mission's utility is a joint function of the rate from multiple sensors. Let the i^{th} mission be denoted as m_i; let M be the set of all missions and S the set of all sensors. Let the utility of a mission m be denoted as $U_m(\{x_s\}_{s\in sset(m)})$, where for any mission m, $sset(m)$ is the set of sensors that are sources for m (i.e., contribute to the utility $U_m(.)$). Thus, $\{x_s\}_{s\in sset(m)}$ represents the vector of rates associated with the set of sensors $sset(m)$. Furthermore, for any sensor s, let $mset(s)$ denote the set of missions subscribing to this sensor's data. Table I provides a list of the symbols commonly used in this article.

3.1. Assumptions

For the analysis in this article, we make the following set of assumptions.

(1) The utility of each mission m is expressed solely in terms of the transmission data rates of the sensors in $sset(m)$, that is, $U_m(.) \overset{s\in sset(m)}{\longleftarrow} x_s$. While specific utilities may be more finely nuanced, it is generally true that a mission's utility is enhanced by an increase in the data volume from each sensor.
(2) Each sensor's flow is completely elastic. Accordingly, a sensor s is able to dynamically adjust its transmission rate x_s by any arbitrary amount (as long as $x_s > 0$). The assumption of a completely elastic and infinitesimally tunable sensor flow may not be completely applicable to sensors with a finite set of discrete settings (e.g., a video sensor may operate at only one of three resolutions); we address these in Eswaran et al. [2009].
(3) Data from a source node is disseminated to the sinks along a well-defined multicast tree precomputed by some routing protocol. All the sinks receive identical rates from a specific sensor, that is, no layered coding is used, nor is any in-network sensor data aggregation employed (we, again, address it in Eswaran et al. [2009]).

3.2. The Wireless Channel Constraint and the NUM Problem

Before expressing the NUM formulation, we first need to modify the constraint $Ax \leq C$ (Equation (1)) to account for the contention-based constraints arising out of the interference among links sharing a common wireless channel. To exploit the wireless

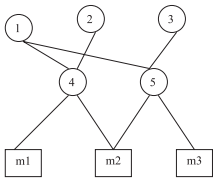


Fig. 1. Connectivity graph of an example network.

broadcast advantage (WBA), we assume (similar to Bui et al. [2007]) that a node on a multicast forwarding tree uses a single link-layer broadcast to reach all its neighboring 'child' nodes. Rates may differ among different broadcast links, but all nodes on a single broadcast link receive at the same rate. Two transmissions interfere with each other if the sender or any receiver of one transmission is in the interference range of the sender or any receiver of the other transmission (referred to as the 'protocol model' of interference in Gupta and Kumar [2000]). To express these interference constraints, we characterize a transmission at a node k by the tuple (k, s), where s is the sensor source node for this flow. A *sensor flow* is thus a series of transmissions by the nodes in the corresponding multicast forwarding tree.

To capture the resulting constraints, we formulate a transmission-based conflict graph (CG) (similar to the approach in Sengupta et al. [2007]), where each vertex of the CG refers to a (k, s) transmission in the network, and an edge between two vertices in the CG implies that the two transmissions interfere with each other. As shown in Chou et al. [2007], the number of nodes in such a multicast CG is given by $\sum_{v_i \in V}(2^{\Delta_i} - 1)$ (where Δ_i denotes the out-degree of node v_i, and V is the set of all vertices). Given the practical impossibility of locally computing independent sets of such a large CG, we instead apply the sufficiency constraint that, in any maximal clique, the sum of the air-time fractions of all the transmissions must not exceed unity. In other words, for each maximal clique $q \in Q$,

$$\sum_{\forall (k,s) \in q} \frac{x_s}{c_{ks}} \leq 1, \tag{4}$$

where x_s is the flow rate, c_{ks} is the transmission rate for transmission (k, s), and Q is the set of all maximal cliques in the CG corresponding to WSN.

The problem of adaptive rate control in such a WSN may then be expressed by the *SENSOR* problem, such that

$$\begin{aligned} &\text{maximize} \sum_{m \in M} U_m(\{x_s\}_{s \in sset(m)}), \\ &\text{subject to} \sum_{\forall (k,s) \in q} \frac{x_s}{c_{ks}} \leq 1, \quad \forall q \in Q. \end{aligned} \tag{5}$$

Figure 1 shows the connectivity graph of an example network consisting of three source nodes, 1, 2, and 3 and three sinks, m_1, m_2, and m_3, where $sset(m_1) = 1, 2$; $sset(m_2) = 1, 2, 3$; and $sset(m_3) = 1, 3$. The intermediate nodes do not sense any data, that is, they do not generate any traffic and are just relay nodes. Figure 3 shows the multicast forwarding trees that the flows follow, and Figure 2 gives the corresponding conflict graph. We can see that in this graph, there are three maximal cliques: $\{T_{11}, T_{22}, T_{41}, T_{42}\}$, $\{T_{11}, T_{33}, T_{53}, T_{51}\}$, and $\{T_{11}, T_{41}, T_{42}, T_{51}, T_{53}\}$. This implies that according to Equation (4), the total air-time fractions of transmissions from 1, 2, and 4 must not exceed 1 (and similarly for the total air-time fractions of transmissions from $\{1, 3, 5\}$ and $\{1, 4, 5\}$). For example, node 1 transmitting (T_{11}) for 50%, node 2 (T_{22}) for

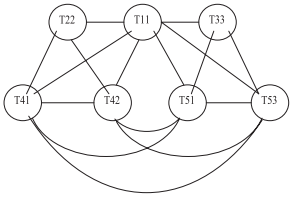


Fig. 2. Conflict graph for the example network.

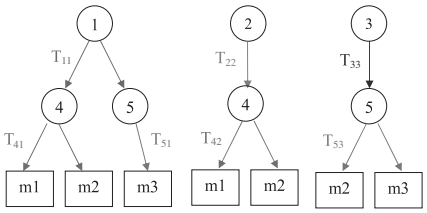


Fig. 3. Multicast forwarding trees and the corresponding transmissions used in the example network.

40%, and node 4 (T_{41} and T_{42}) for 30% of time is not valid, since this would imply that some of these interfering transmissions occurred simultaneously.

3.3. Distributed Optimization Technique

We decompose the *SENSOR* optimization problem into two subproblems *SINK* and *NETWORK*, by introducing a pricing scheme and show that solving these two problems independently solves a relaxation of the *SENSOR* problem.

Suppose a sink (mission) m is charged at a rate λ_{ms} for receiving a rate of x_s from sensor s. The sink m pays an amount w_{ms} per unit time, where $w_{ms} = \lambda_{ms} * x_s$. Thus w_{ms} can be interpreted as the 'willingness to pay' (this is not an actual payment, but instead an intermediate Lagrangian variable that reflects the gradient of the perceived collective utility). Then the utility maximization problem for a becomes ***SINK***$_m$**:**

$$\text{maximize } U_m\left(\frac{\bar{w_m}}{\bar{\lambda_m}}\right) - \left(\sum_{s \in sset(m)} w_{ms}\right) \text{ over } w_{ms} > 0, \tag{6}$$

where $\bar{w_m}$ is a vector of w_{ms}; $\bar{\lambda_m}$ is a vector of λ_{ms}; and element-wise division of $\bar{w_m}$ by $\bar{\lambda_m}$ is assumed.

Similarly, the *NETWORK* problem becomes

$$\begin{array}{ll} \text{maximize} & \sum_{s \in S} \sum_{m \in M} w_{ms}\, log(x_s); \\ \text{subject to} & \sum_{\forall (k,s) \in q} \frac{x_s}{c_{ks}} \leq 1, \text{ for each clique } q \in Q, \text{ over } x_s \geq 0. \end{array} \tag{7}$$

The log function ensures that the objective of the distributed optimization problem is concave and also ensures proportional fairness [Kelly et al. 1998]. The Lagrangian for the problem *NETWORK* is

$$L = \sum_{\forall s \in S} \sum_{\forall m \in M} w_{ms}\, log(x_s) + \sum_{\forall q} \mu_q \left(1 - \sum_{\forall (k,s) \in q} \frac{x_s}{c_{k,s}}\right)$$

where μ_q is the shadow cost corresponding to the bandwidth constraint in clique q. Taking derivatives and using first-order necessary conditions for optimal rate for a

sensor s, we get

$$\frac{dL}{dt} = \frac{\partial L}{\partial x_s}.\frac{dx_s}{dt} = 0$$
$$\Rightarrow$$
$$\frac{d}{dt}x_s(t) = \kappa\left(\sum_{m\in mset(s)} w_{ms}(t) - x_s(t) * \sum_{\forall q\in clique(s)} \mu_q * \sum_{\forall(k,s)\in q}\frac{1}{c_{ks}}\right), \tag{8}$$

where μ_q (a clique's shadow cost) is given by

$$\mu_q\left(\sum_{\forall(k,s)\in q}\frac{x_s(t)}{c_{ks}}\right) = \left(\sum_{\forall(k,s)\in q}\frac{x_s(t)}{c_{ks}} - 1 + \epsilon\right)^+ \Big/ \Delta, \tag{9}$$

where ϵ denotes an over-provisioning factor (a larger ϵ implies that the congestion cost becomes nonzero at a lower utilization factor); Δ is a constant that determines the magnitude of the congestion cost (per bit); and $()^+$ denotes the 'positive part' function. In addition, if each sink (mission) adapts its 'willingness to pay' for sensor s based on the source rates and its own utility function $U_m(.)$, which is based on the partial derivative of its own joint utility according to the equation

$$w_{ms}(t) = x_s(t)\frac{\partial U_m}{\partial x_s}, \tag{10}$$

we can then show that the previous system of $M + S$ equations (one for each of the M *SINK* problems and one for each of the S sensors for optimizing the *NETWORK* problem) satisfies the following property.

THEOREM 1. *The strictly concave function given by*

$$\vartheta(x) = \sum_{m\in M} U_m(X_m) - \sum_{\forall q}\int_0^{\sum_{\forall(k,s)\in q}\frac{x_s}{c_{ks}}} \mu_q(y)dy \tag{11}$$

is a Lyapunov function for the system of differential equations in Equation (8). The unique value x maximizing $\vartheta(x)$ is a stable point of the system.

PROOF (PROOF SKETCH). With a proof similar to that of Kelly et al. [1998], we can easily verify that

$$\frac{d}{dt}\vartheta(x(t)) = \sum_{\forall s\in S}\frac{\partial\vartheta}{\partial x_s}.\frac{d}{dt}x_s(t)$$

is strictly positive when x is not the unique maximizing point of $\vartheta(x)$, and zero otherwise. This proves that the function $\vartheta(x)$ is a Lyapunov function for the system given by Equation (8). □

It follows that the unique solution to Equations (8) and (10) provides a decentralized solution to a relaxation of the optimization problem *SENSOR*, defined by Equation (5). The proof also suggests that the adaptation mechanism will converge to the optimal rate allocation from any initial rate vector, as long as the network conditions do not vary (e.g., in a stationary WSN).

3.4. Observations on the Decomposition for Joint Utility Functions

The NUM-based distributed solution for the mission-centric WSN has the following distinguishing features. First, the rate adaptation performed by a sensor s (according

to Equation (8)) depends on the sum of the w_{ms} of each individual subscribing mission $m \in mset(s)$. The use of partial-derivative-based feedback via w_{ms} effectively decouples each source sensor from the others, even though missions have joint utility functions.

Second, due to the multicast nature of the sensor flows, the adaptation of the w_{ms} parameters is now performed by each of the M sinks. Accordingly, our NUM solution is based on a receiver-centric optimization model.

Finally, the association of shadow costs with each clique (Equations (8) and (9)), rather than with individual links reflects the use of wireless link-layer broadcasts for sensor data transmission. This implies that all contending flows observe the same congestion cost for the shared wireless medium.

4. WSN-NUM PROTOCOL DETAILS

While the joint utility model may be theoretically solved in a distributed manner using Equations (8) and (10), a protocol-level implementation of this solution must address several practical challenges.

4.1. Computing 'Willingness to Pay'

To effectively compute the value of w_{ms} for each sensor in $sset(m)$, a mission m must not only be aware of its utility function $U_m(.)$ but also of the data rates of all the relevant sensors (as the partial derivative term in Equation (10) involves current rates of other sensor nodes, along with the rate of sensor s). Determining the actual transmission data rate of a source is not trivial, as a) the sensor may generate data intermittently or in bursty fashion, and b) packet losses on the forwarding path can cause a receiver to underestimate the true sending rate of the source. Hence, we require the sender to explicitly 'piggyback' its current transmission rate on an appropriate subset of the data packets, so that each subscribing receiver can be directly aware of the sensor's current transmission rate.

4.2. Constructing Local Conflict Graphs and Maximal Cliques

In order to identify all transmissions that interfere with a node, each node computes a conflict graph, whose vertices represent broadcast transmissions. An edge between two vertices means the two transmissions interfere with each other. Maximal cliques in the conflict graph represent sets of contending transmissions. Each node has to compute its cliques locally, and for this, the node requires the transmission and neighbor information of all nodes that are within its interference range. A practical approach to determining nodes within the interference range would be to take into account the ratio between interference range and transmission range, so that the relationship between the distance and the number of hops can be realistically approximated. In general, if the interference range is r times the transmission range, then the transmission-based local conflict graph at that node can be computed with information from neighbors within r hops (with reasonable assumptions on the density of the network). Each node can then construct a local conflict graph by the following method.

(1) Each node exchanges its neighborhood and transmission information with its r-hop neighbors.
(2) Using the received data, each node constructs its local conflict graph.
(3) The local conflict graph of a node changes each time there is a change in mission configuration or in the topology in its neighborhood region. When a new transmission is introduced, the transmitting node initiates the information exchange procedure. When a transmission is removed, it can simply be ignored because, after a timeout period, its airtime fraction is taken as 0.

Each node thus constructs a partial view of the global conflict graph consisting of only the transmissions that belong to the same clique as its own transmissions. The cliques in this graph can be computed locally using the corrected Bierstone algorithm in polynomial time [Mulligan and Corneil 1972].

4.3. Computing and Communicating a Clique's Shadow Cost

In our WSN environment, the total congestion cost for a specific sensor is defined as the sum of the congestion cost of all maximal cliques that it traverses. Obtaining this cost is not trivial for two reasons: (a) the association of multiple child nodes with a specific forwarding node (in the multicast forwarding tree) and (b) the occurrence of multiple distinct transmissions (by different forwarding nodes) of the same flow within a clique.

To observe the first issue, note that the paths of two distinct missions m_i and m_j belonging to $mset(s)$ will not be disjoint but will most likely, have significant overlap (many common upstream nodes). In such a case, if the sensor s were to simply combine (additively) the cumulative costs from each separate path, it would be guilty of overcounting. This problem of inaccurate cumulative cost computation can be remedied by modifying how clique shadow costs are communicated. Unlike the unicast approach, where each link (intermediate node) simply adds its own link cost $p_j(.)$ to the upstream 'cost' field, each intermediate node k with N_k distinct children will divide the upstream cost field by N_k before transmitting this on its 'logical' links. (The term 'logical' is used since the intermediate node, in practice, performs only a single link-layer broadcast to the N_k child nodes). This simple approach, which we refer to as the *Split Shadow Cost* technique, ensures that the source receives an accurate estimate of $\sum_j p_j(.)$, even with multiple receiving missions.

The computation of the shadow costs also requires careful consideration. When two or more nodes of the same multicast forwarding tree belong to the same clique, it is necessary to ensure that only one member of a particular maximal clique performs the splitting of the shadow cost. To ensure this, we associate a unique ID with each clique and piggyback the IDs of all the cliques whose costs have already been appended. A downstream node will thus avoid adding the clique cost again if the clique ID is already present. (The clique ID is computed locally at each node by simply performing a prespecified hash of all the (k,s) pairs in the clique).

Finally, the computation of the shadow cost for a clique also poses some implementation issues. Equation (9) implies that the calculation of the shadow cost by a node requires knowledge of the data rates of flows traversing other neighboring nodes. Accordingly, to compute the clique price, nodes belonging to a particular maximal clique must periodically exchange the current airtime fractions for the corresponding transmissions, that is, $\frac{x_s}{c_{ks}}$. However, the computation of clique cost does not depend entirely on these periodic broadcasts, as each node is continually aware of the current source rates for all flows that transit through it.

4.4. Protocol Steps

In summary, the WSN-NUM protocol follows these steps iteratively, as illustrated in Figure 4.

(1) Each source s transmits data at current rate $x_s(t)$.
(2) Each forwarding node computes the clique cost according to Equation (9) and transmits the split shadow cost along with the data. Each node belonging to a distinct clique cumulatively adds its costs to the received value before forwarding the data, until the mission is reached.

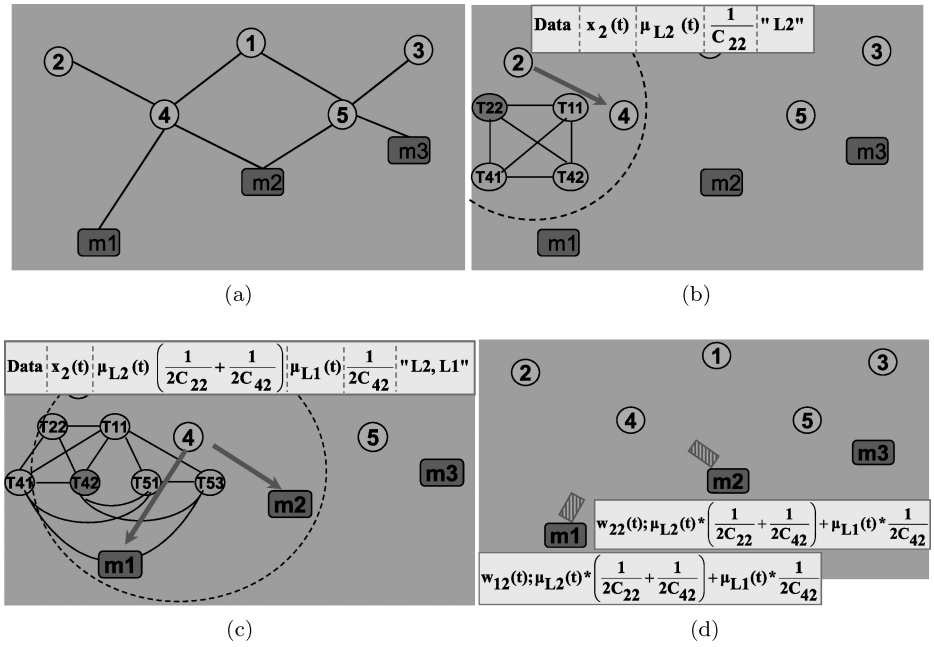


Fig. 4. Illustration of the working of WSN-NUM protocol. (a) Example network with five sensor nodes and three missions. The connectivity among the nodes is shown by the black lines. (b) Source 2 broadcasts its data with additional protocol information, and node 4 receives it. The local conflict graph constructed at node 2 is also shown. (c) Node 4 forwards data along with updated control fields to missions m1 and m2. The local conflict graph at node 4 is also shown. (d) Mission m1 and m2 send their feedback to source 2.

(3) Each mission computes its willingness to pay and forwards this, along with the received costs to the corresponding source, as feedback.
(4) Each source upon receiving this feedback, computes its rate for the next iteration according to Equation (8).

5. SIMULATION-BASED PERFORMANCE EVALUATION

In order to further study the convergence properties, signaling overheads, and other performance metrics of our proposed congestion control framework in a realistic wireless environment, we simulated the WSN-NUM protocol using the discrete-event simulator Qualnet [Qualnet]. We believe that such protocol-level simulations of the NUM techniques have not been reported earlier, and as such, provide a very useful reference of the issues surrounding its practical implementation.

The transmission of data packets is based on the distributed IEEE 802.11b MAC. Moreover, the computation of the cliques and shadow costs are determined (unless otherwise stated) using the default rate-range curves for 802.11b in Qualnet. The simulated wireless ad hoc network consists of 100 stationary nodes placed randomly across a 1,500 m $\times$ 1,500 m field. The link rate is set to 2 Mbps; the flows generate CBR traffic. The inter-feedback interval (the duration between two successive cost feedback signals from a receiving mission) is set to 10 s; the clique-cost update interval is set to 60 s. Based on literature [Yang et al. 2005; Sridharan and Krishnamachari 2007], the congestion-tolerance limit ϵ is set to 0.3, yielding an over-provisioning of 30%. The 95% confidence intervals are presented wherever appropriate, where the values were averaged over 20 runs. The network configuration scripts and WSN-NUM module for Qualnet are available [Scripts].

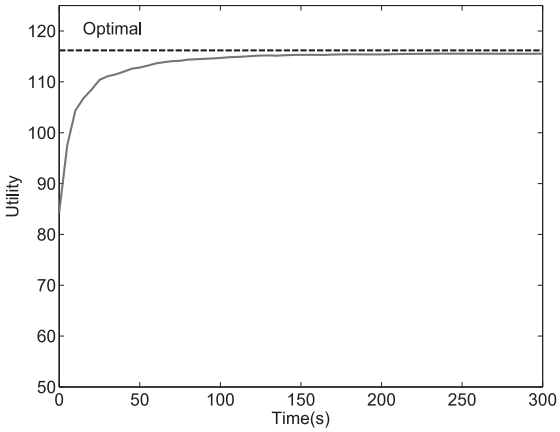


Fig. 5. Evolution of the global utility of the network.

Table II. Signaling Overhead and QoS Metrics

End-to-end Latency (ms)	55.21
Packet Delivery Ratio (%)	72.68
Signaling Overhead (bytes/(node.min))	147.83
Signaling Overhead (% overhead)	0.17

5.1. Utility Variation with Time

Figure 5 shows the evolution of total network utility with time when WSN-NUM is simulated on the previously described network with 20 sources and 25 missions. The optimal utility, computed using the centralized solver GAMS [GAMS], is also shown by the black, dotted line. The load in the network converges to ∼1.4 Mbps. We observe that the WSN-NUM protocol drives the network utility close to the optimal value. It may be noted that this convergence is attained, even though the wireless broadcasts have no link-layer reliability. As a consequence, individual packets may be lost due to both channel errors and collisions, and individual packet transmissions do not always reach the sink nodes (missions). Similarly, as our feedback packets (containing the w_{ms} values) are sent over unreliable UDP, a source may not receive feedback from all the sink nodes. To account for these practical realities, a sensor performs its rate adaptation when the number of feedback messages received exceeds a predesignated threshold value. We have verified that this approach still drives the system close to the optimal, implying that the sink can send its feedback over larger windows of packets, if needed.

5.2. Observed Overhead and QoS Metrics

While the NUM formulation is shown to result in utility quite close to the optimum, it is equally important to study the actual packet-level QoS metrics observed by the receiving nodes. Table II present two important metrics: (i) the *average end-to-end latency* of all packets (averaged across all receivers) and (ii) the *average packet delivery ratio* (averaged across packets received by all receivers). Table II also lists the signaling overhead, which includes the messages exchanged initially for local conflict graph construction and for the periodic feedbacks and clique-cost updates. We can see that the additional signaling required in our protocol takes up only a few bytes per minute at a node.

From these, we observe that the WSN-NUM framework, while somewhat idealized (e.g., it is flow, not packet, -based), does indeed drive the overall system to a stable and

Table III. Impact of Interference Range on Utility

	Wired	Wireless (IR=TR)	Wireless (IR=2TR)
Optimal utility	138.34	116.23	94.82
Mean utility	137.81	115.50	91.06
Std dev	0.11	0.44	0.97

useful equilibrium, where the delays and losses are reasonable. It is perhaps useful to make another observation on the suitability of the clique-based constraint, specified earlier in Equation (4). As mentioned, this constraint is inherently pessimistic, as it potentially disallows some otherwise feasible transmissions. On the other hand, the constraint may also be viewed as optimistic because it is MAC-agnostic, that is, Equation (4) assumes perfect MAC scheduling and fails to factor in the loss in capacity observed at moderate to high loads, due to MAC-layer overheads and contention delays. The simulation-based results suggest that, at least for 802.11b environments, the clique-based formulation provides a reasonable balance between these contending causes of pessimism and optimism. Later, in Section 6.2.1, we shall discuss how dynamically adapting ϵ helps support high packet delivery rates under diverse network conditions and traffic loads.

5.3. Impact of Interference Range on Utility

The initial specification of the algorithm for local computation of conflict graph assumes that the interference range (IR) equals the transmission range (TR) (thus, clique members can be, at most, two hops apart). However, in real 802.11-based environments, the interference range is often higher than the transmission range (e.g., in Qualnet, by default, $IR \approx 1.7 \times TR$). To study the impact of the interference range, we ran our simulation with the same topology and numbers or sources and missions but different link characteristics, effectively simulating different values of the ratio $\frac{IR}{TR}$. Table III compares the system utilities for a wired network (i.e., no interference), wireless network with $IR = TR$, and wireless network with $IR = 2TR$. The mean and standard deviation of the utility was computed after the protocol converged, until the end of simulation runtime (300 s). We see that the optimal rate and the performance of NUM rate control decrease as interference increases. This clearly establishes the importance of accurately capturing the interference constraints in the NUM formulation.

6. FACTORS AFFECTING PERFORMANCE OF WSN-NUM PROTOCOL

There are several factors at the protocol-level, as well as internal to the framework that influence the performance of the WSN-NUM protocol. These factors include:

(1) Step size at which the data rate is adjusted (κ),
(2) Congestion-tolerance limit (ϵ),
(3) Feedback frequency,
(4) Clique-cost update frequency.

In this section, we present the analysis of the impact of these factors on the WSN-NUM protocol performance.

6.1. Step Size

The step size at which the sender adjusts its rate is given by κ in Equation (8). The step size determines the magnitude of increase or decrease in rate during each iteration, and thus influences how soon the protocol converges. It is known [Bertsekas 1999] that the larger the value of step size, the faster the convergence of the protocol. However, if the step size is too large, it gives rise to oscillations in the system. Hence, it is necessary

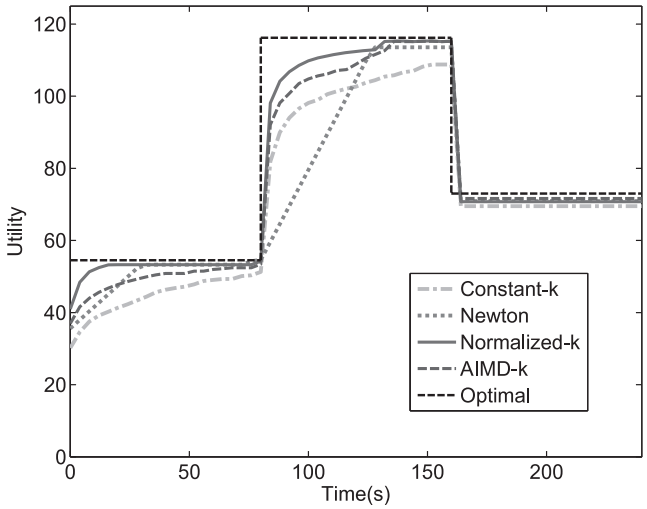


Fig. 6. Comparison of convergence improvement techniques under dynamic mission settings for the network simulated in Section 5: from $t = 0$ to $80s$ there are 10 missions and 10 sources; from $t = 80$ to $160s$ there are 25 missions and 20 sources; from $t = 160$ to $220s$, there are 15 missions and 15 sources.

to set this value such that it neither slows down convergence nor makes the system unstable.

The speed of convergence is a critical issue, because dynamicity is one of the most important characteristics of mission-oriented WSNs, with missions and sensors arriving and departing with varying frequencies. Our protocol must be robust enough to handle this and, at the same time, converge to the optimal rates quickly. The basic NUM framework can be modified in several ways to improve the speed of convergence. For example, alternative decomposition and optimization techniques and their effects on the speed of convergence have been studied in Palomar and Chiang [2007] and Athuraliya and Low [2000]. However, the most simple yet effective way to achieve fast convergence is to dynamically adapt the step size κ. We discuss different methods for intelligently adapting κ, the motivation behind the adaptations, and their effects on convergence, comparing them with a constant value of κ. These methods are simulated on Qualnet for the 802.11b network described in Section 5, under dynamic mission conditions (ten missions and ten sources from $t = 0$ to $80s$; 25 missions and 20 sources from $t = 80$ to $160s$; 15 missions and 15 sources from $t = 160$ to $220s$). It may be noted that the missions are dynamic only in the simulations in Section 6.1. The constant value of κ is tuned to 3,000 after determining, by trial and error, that approximate value that yields quick convergence with no rate oscillations. The results are shown in Figure 6.

6.1.1. Newton's Method. Newton's method is a popular alternative to the gradient-approach because of its fast convergence. The convergence is faster in Newton's method than gradient ascent as a result of using both first- and second-order derivatives for computing the optimum of the objective function, that is, each iteration in the optimization of objective function $f(x(t))$ is defined as

$$\frac{dx}{dt} = \hat{\kappa}[\nabla^2 f(x(t))]^{-1} \nabla f(x(t)), \tag{12}$$

where $\nabla^2 f(x(t))$ is the Hessian, $\nabla f(x(t))$ is the gradient, and $\hat{\kappa}$ is a constant step size.

The applicability of Newton's method in the NUM-framework has been studied before [Athuraliya and Low 2000]. We tested the performance of Newton's method in our problem. The Hessian matrix for our optimization problem is derived as

$$\begin{pmatrix} \frac{-\sum_m w_{m1}}{x_1^2} & 0 & 0 & . \\ 0 & \frac{-\sum_m w_{m2}}{x_2^2} & 0 & . \\ 0 & 0 & \frac{-\sum_m w_{m3}}{x_3^2} & . \\ . & . & . & . \end{pmatrix}.$$

We observe that the Hessian contains independent sensor rate terms and that it is a diagonal matrix, making it easily invertible and distributable. Substituting the Hessian matrix and the gradient from Equation (8) in Equation (12), we get the rate-change equation as

$$\frac{d}{dt}x_s(t) = \hat{\kappa}\frac{x_s}{\sum_{m\in mset(s)} w_{ms}(t)}\left(\sum_{m\in mset(s)} w_{ms}(t) - x_s(t) * \left(\sum_{\forall q\in clique(s)} \mu_q * \sum_{\forall (k,s)\in q} \frac{1}{c_{ks}}\right)\right). \quad (13)$$

We can see that this is similar to Equation (8) but with a variable step size. Figure 6 shows that this method does converge faster than gradient ascent with constant κ, but not as well as the other methods.

6.1.2. Normalized κ. One key observation regarding the gradient-based κ adaptation is that the step size at a source depends on the congestion along that source's flow (to all its receivers). At a conceptual level, the congestion caused by a flow (more accurately, the network resources consumed by the flow) depends on the rate of the source and also on the number of missions receiving data from that flow (a larger number of missions implying, in general, that the flow traverses more links). It is desirable to have a small κ when congestion is high and vice versa. A simple tweak based on this observation is to use a constant κ at each source, normalized by the current number of missions consuming its data, that is, $\kappa_s(t) = \frac{\kappa}{|mset(s)|}$. This approach implies that the source rate changes more conservatively for a flow with more consumers; this is desirable, as a rate change for such a flow causes a larger volume of cascading feedback from the consuming missions. We can see from Figure 6 that this method converges very quickly.

We also experimented with a step size with exponential decay, of the form $\kappa_s(t) = \frac{\kappa}{|mset(s)|} * (\alpha^{-\beta t} + \gamma)$, such that initially the step size is very high and it gradually decays, so that the initial high step size will sufficiently boost the rates. However, our results show that this does not do better than the normalized-κ method. We confirmed that this method performs best when β is zero, which is equivalent to the normalized-κ method.

6.1.3. AIMD κ. Next, we took an Additive Increase Multiplicative Decrease (AIMD)-approach to make the step size adaptive to the congestion level. Accordingly, the step size increases linearly by a constant amount, when $dx_s(t) > 0$, and it is halved when $dx_s(t) < 0$ (which indicates congestion). Once $dx_s(t)$ stabilizes near zero, the step size must be maintained constant. Figure 6 compares the performances of this method with the rest, and we see that this is almost as good as the normalized-κ method. Different functions of dx_s were used as the increase and decrease factors, but none performed significantly better than constant factors.

Thus, by adapting κ according to the congestion in the network, we can significantly improve the convergence time of the NUM protocol, even when the network is dynamic with frequent changes in mission activities.

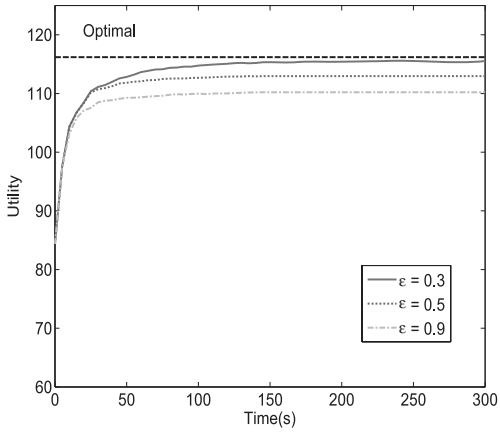


Fig. 7. Evolution in utility for different values of ϵ.

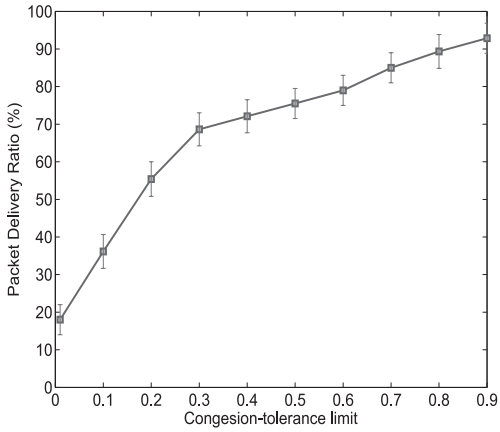


Fig. 8. Effect of congestion-tolerance limit (ϵ) on packet delivery ratio.

6.2. Congestion-Tolerance Limit

Another parameter that is internal to the framework and affects the performance of the protocol is the congestion-tolerance limit ϵ, as seen in Equations (8) and (9). The congestion-tolerance limit ($0 \leq \epsilon \leq 1$) determines how much congestion can be tolerated by the network, or in other words, at what operating point the network is perceived to be congested. This influences how close to the optimal value the system reaches. If ϵ is too big, the bandwidth will be underutilized. For instance, if $\epsilon = 0.6$, the network is perceived to be congested as soon as the bandwidth utilization exceeds 40%. On the other hand, if ϵ is too low (i.e., does not detect congestion at the right operating point), the network will experience high congestion and significant packet losses.

To illustrate these effects, we varied the value of ϵ in the Qualnet simulation of the stationary network and mission configuration described in Section 5. These results, along with the optimal computed using GAMS, are shown in Figure 7. We can see that as ϵ becomes small, the utility approaches the optimal value. In addition to these, we also observe the packet delivery ratio in Figure 8 for different values of ϵ. We see that the packet delivery ratio is lower for lower values of ϵ.

6.2.1. Dynamically Adaptive ϵ. Clearly, the value of ϵ plays an important role in the performance of the network. In the simulation in Section 5, the value of ϵ was set after careful analysis but based on trial and error. In this section, we present two simple algorithms for dynamically adapting ϵ, such that the network converges at a target packet delivery ratio.

(i) *Per-Clique Threshold Adaptation (PCTA).* In this algorithm, each forwarding node monitors the rate at which it receives data over a period of 45 seconds. The packet delivery ratio at each receiving node is computed using the monitored received rate and the source rate piggy backed with data. It is also assumed that the network has a target PDR based on the application- and network-dependent factors. The node sends a two-bit LOSS signal to the upstream node, based on the perceived PDR: *TARGET LOSS* if the PDR is close to the target PDR (e.g., $\pm 3\%$), *HIGH LOSS* if it is lower, and *LOW LOSS* otherwise. The node that computes the congestion cost for its clique sets ϵ based on the LOSS signals observed within the clique (which are included in the clique-cost exchange messages), as follows. If there is at least one HIGH LOSS signal, the ϵ value is doubled; otherwise it is decremented marginally—for example, by 25%—thereby following a 'rapid increase and conservative decrease' policy. If all signals indicate *TARGET LOSS*, there is no change in the ϵ value. Each clique's congestion cost is computed based on its adapted ϵ value.

We simulated this algorithm on the network described in Section 5 with target PDR of 75%. Figure 9(a) shows the evolution of *effective utility* (defined as the utility computed based on the product of the source rate and the average PDR). The resulting average PDR values are shown in Figure 9(b). The evolution of ϵ over time at an intermediate node is shown in Figure 9(c). The convergence in PCTA is slow because each clique uses a different ϵ value, due to it taking several iterations before the congestion costs computed per clique reflect the end-to-end congestion measure.

(ii) *Path Loss-Based Threshold Adaptation (PLTA).* To overcome the slow convergence of PCTA, we modify the previous algorithm to adapt ϵ based on end-to-end congestion. In this approach, only the mission (sink) node monitors the PDR and sends the LOSS signal to the source. The source adapts ϵ based on the 'rapid increase and conservative decrease' policy previously described and piggybacks the new value of ϵ with the data, which is used by the forwarding nodes to compute the congestion cost. We can see from Figures 9(a–c) that this results in rapid convergence of ϵ value.

We compare the performance of PCTA and PLTA with three constant settings of ϵ. $\epsilon = 0.1$ indicates an excessively aggressive threshold, which results in high source rates but low PDR, reducing the effective utility. $\epsilon = 0.7$ indicates an excessively conservative threshold, which results in high PDR but low bandwidth utilization. $\epsilon = 0.3$ is the value used in Section 5 which was selected based on trial and error. We see that the adaptive algorithms also converge close to this value. We will investigate other approaches, such as more fine-grained, weight-based increment and decrement of ϵ, in future work.

6.3. Feedback Frequency

According to the WSN-NUM protocol, the sources adjust the rates based on the cost feedback received from the destinations. The frequency at which this feedback is sent influences the network performance. Clearly, if the feedback is sent in large intervals, the convergence will be slow and the network is likely to operate in a congestion region for a longer time. Conversely, the signaling overhead and packet losses will be high if the feedback signals are sent too frequently.

This trade-off can be observed from the simulation results obtained from Qualnet, again for the network described in Section 5. In Figures 10(a–d), we see how the convergence time, packet delivery ratio, and signaling overhead change, respectively, with different feedback intervals. The signaling overhead is measured from the number

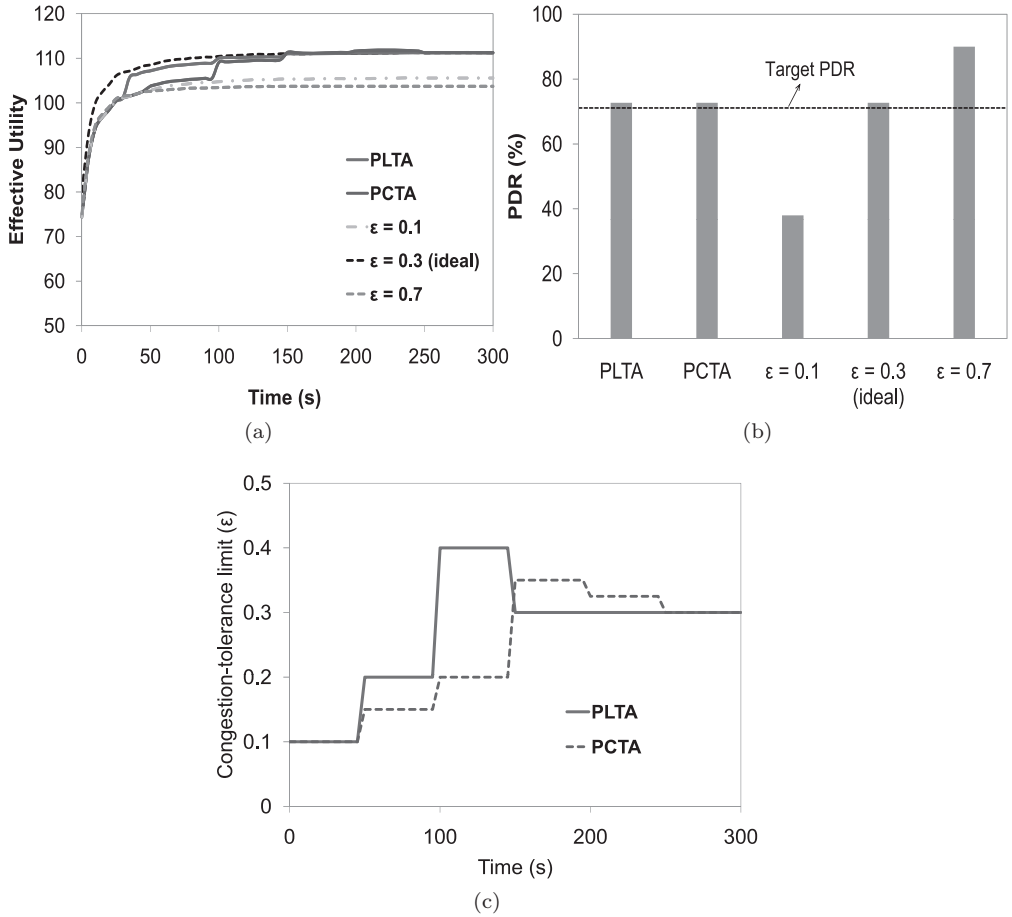
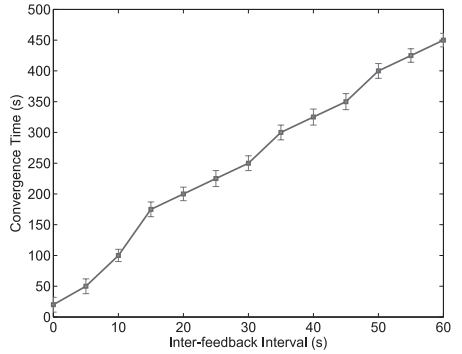
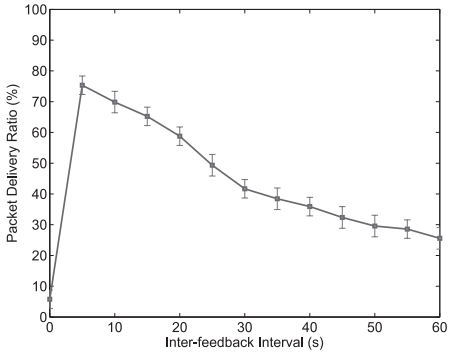


Fig. 9. (a) Evolution of effective utility and (b) average packet delivery ratio after rate convergence, for constant and adaptive values of ϵ. The adaptive methods for dynamically adjusting ϵ include PCTA and PLTA. Three different constant values are studied: $\epsilon = 0.1$ (over-aggressive), $\epsilon = 0.7$ (over-conservative), and $\epsilon = 0.3$ (chosen as ideal by trial and error). (c) The evolution of ϵ over time for PCTA and PLTA at an intermediate node.

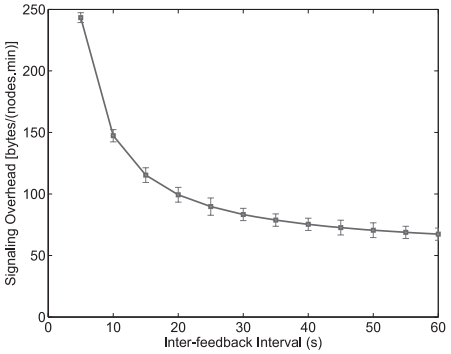
and size of control packets transmitted in the discrete-event simulation. We can see from Figure 10(a) that as the inter-feedback time increases, the time to converge also increases. From Figure 10(b), we see that the packet delivery ratio is low for very small feedback intervals, because the feedback messages increase congestion (feedback interval of 0 corresponds to sending a feedback for every packet received). It increases when this interval increases, because the WSN-NUM protocol regulates the rates to alleviate congestion, but after a threshold interval, the protocol fails to regulate the rates promptly, and the network remains congested for a longer time, resulting in reduced packet delivery ratio. From Figure 10(c), we see that as the feedback interval increases, the overhead decreases significantly. We also observe that the overhead is under 1%. Thus, we observe that the feedback update frequency must be set according to the convergence time desired. For the 100-node network simulated, the best choice is a feedback interval of around 5–10 seconds. At this value, the packet delivery ratio is high, signaling that overhead is low and convergence time is reasonable.



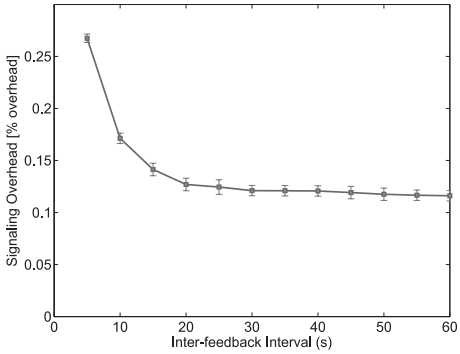
(a) Convergence time vs. inter-feedback interval.



(b) Packet delivery ratio vs. inter-feedback interval.



(c) Signaling overhead vs. inter-feedback interval.



(d) % Signaling overhead vs. inter-feedback interval.

Fig. 10. Effect of inter-feedback interval on convergence time, packet delivery ratio, and signaling overhead.

6.4. Clique-Cost Update Frequency

The airtime fractions of all transmissions in a clique must be exchanged periodically, in order to compute the clique cost. This communication incurs some overhead, so how often this exchange occurs impacts the network. However, it is possible to reduce the need for this exchange by making use of the fact that nodes within transmission ranges can overhear one another. Every node involved in forwarding a flow is always aware about the current source rates of that flow (as this information is embedded in the flow's data packets). All the neighboring nodes that hear the transmission of this packet can also update their cliques with the information of the current airtime fraction. Because all nodes within transmission range of a node are updated, reducing the explicit clique-update frequency does not affect the performance of the protocol.

This is illustrated in Figure 11, which shows the utility of the network when clique-cost exchange takes place every ten seconds and 60 seconds, compared to the case when there is no explicit clique-cost exchange at all. We see that the difference in utility is almost negligible. However, clique-cost update frequency does affect the packet delivery ratio and signaling overhead, as shown in Figures 12 (a and b).

7. IMPLEMENTATION

In addition to simulation studies, we also implemented the WSN-NUM protocol on a real-time network, the details of which are presented in this section.

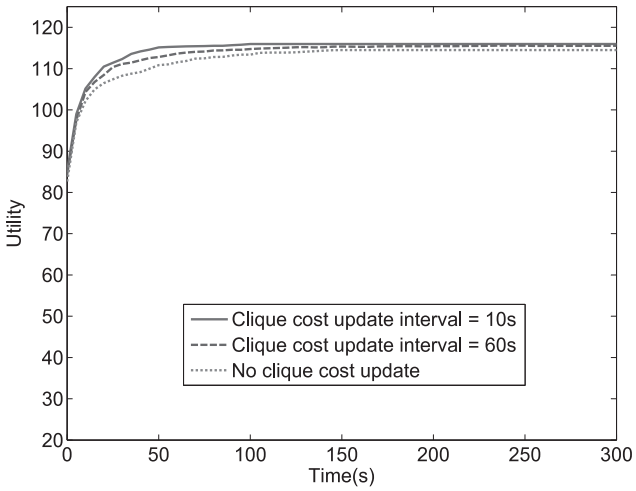
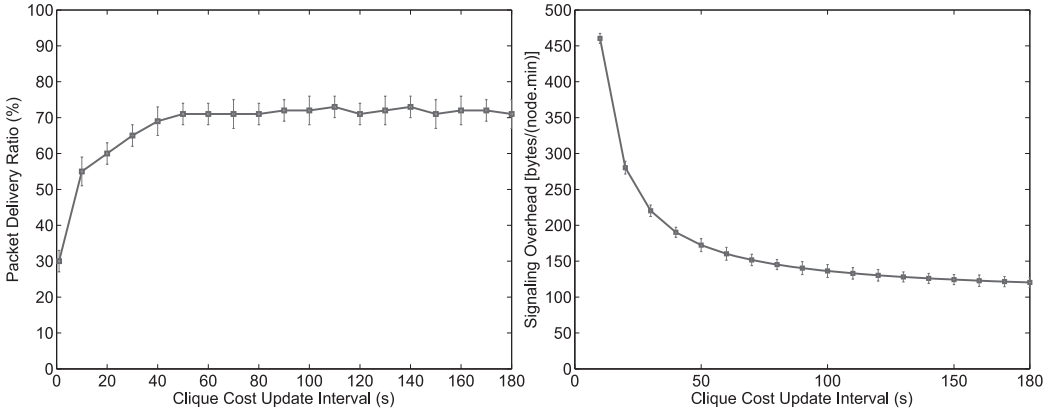


Fig. 11. Variation in network utility with time, for two cases of clique-update interval.



(a) Packet delivery ratio vs. clique-update interval. (b) Signaling overhead vs. clique-update interval.

Fig. 12. Effect of clique-update interval on packet delivery ratio and signaling overhead.

7.1. Implementation Details

The WSN-NUM protocol was implemented on a 10-node 802.11b wireless network. The connectivity graph of the network is shown in Figure 13, where a solid line between two nodes indicates that they are within the transmission range of each other, and a dashed line indicates the flow route. There are three sources (nodes 1, 2, and 3 sending at rates x_1, x_2, and x_3, respectively) and three missions (nodes 8, 9, and 10). Mission 8 receives data flows from 1 and 3, with utility function $log(1+x_1)+log(1+x_3)$; mission 9 receives all three flows, with utility function $log(1+x_1)+log(1+x_2)+log(1+x3)$; mission 10 receives only flow 3, with utility function $log(1+x_3)$. This utility function is concave and also reflects the utility of mission-oriented WSNs, where the missions benefit from higher data rates and, eventually, the gain diminishes beyond sufficiently high data rates. The values of the model parameters are the same as in Section 5. The link transmission rate was fixed at 2 Mbps.

Unlike the Qualnet-based simulations in Section 5, the control and data planes are separated in the implementation. The control messages (such as congestion cost,

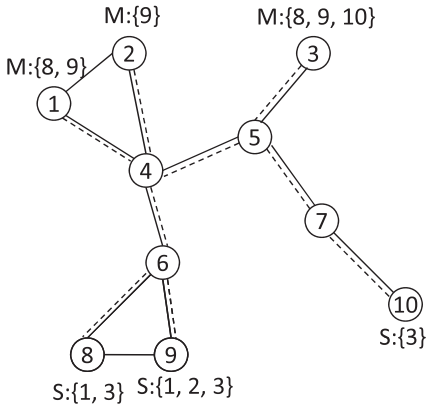


Fig. 13. Topology of implemented network.

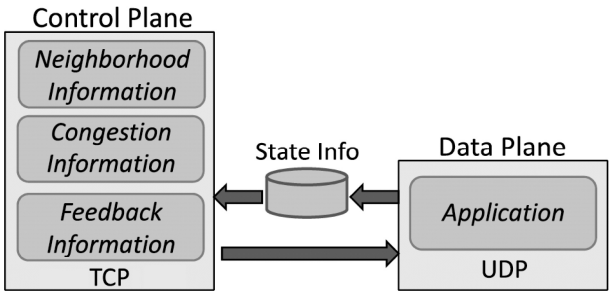
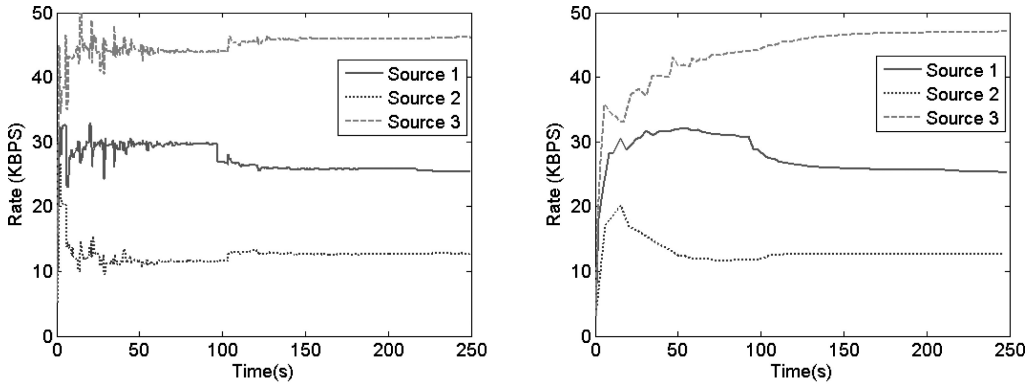


Fig. 14. Architecture of implementation of WSN-NUM protocol.

feedback, and those exchanged during clique construction and update) are reliably communicated using TCP, via a publish-subscribe middleware [Wright et al. 2009]. The application itself resides on a separate data plane and the data packets are transmitted via a separate UDP channel. The data plane periodically interacts with the pub-sub communication middleware and adapts its flow rate according to the congestion in the network.

The implementation (whose architecture is summarized in Figure 14) consists of four main modules.

(1) *Learning about neighborhood*. Each node in the wireless ad hoc network must learn and keep track of its neighbors, its transmissions, and changes in topology. In order to do that, we employ a virtual publisher called the "Neighborhood Sensor" in the pub-sub engine, which periodically publishes and subscribes to "HELLO" messages. In addition, the neighborhood sensor is also responsible for periodically transmitting its current set of neighbors and information about its transmissions. Each node subscribes to this information from its first- and second-hop neighbors to construct its local conflict graph.
(2) *Congestion monitoring*. Each node uses the "Congestion Sensor" to announce its contribution to the network congestion and also learn about the congestion at the other nodes in its vicinity (i.e., interference range), in terms of the fraction of transmission time used. This information is periodically published, and it is used for computing the congestion cost at each clique.



(a) Simulation (b) Implementation

Fig. 15. Adaptation of source rates over time.

(3) *Transmission of feedback*. The control plane is responsible for providing the feedback channel through which each source receives feedback about the congestion cost along its flow path and its contribution to the mission utilities (known as the willingness to pay). This feedback channel proceeds from each mission to all its sources on a hop-by-hop basis, where at each hop, the congestion cost at the clique in that region is computed and cumulatively added to the total cost before sending the feedback to the next upstream node.
(4) *Application-layer source rate adaptation*. The application sends data packets via UDP at a rate that is adapted according to the feedback information on congestion and willingness to pay, received from the control plane.

7.2. Results and Discussion

Figure 15 shows the adaptation of source rates over time in the implementation and compares it with Qualnet-based simulation of the same network. We see that the converged source rates of the implementation and simulation are similar, but there is some difference in the convergence speed. This is because the feedback incurred greater delay in reaching the sources in the implementation because of the TCP-buffering overhead within the pub-sub engine. But the convergence time can be reduced further by either increasing κ or adjusting it, as discussed in Section 6.1.

The most important difference that we observed between the simulation and implementation is the packet loss encountered. Since the data is forwarded using broadcast transmissions, there is significant loss due to collision. Table IV lists the source rate, the packet delivery ratio (PDR), and the net rate, which is source rate times PDR. The further drop in PDR seen in the implementation is due to the external interference in the network. The experiments were conducted late in the night to ensure minimal interference but there was still some effect, as evident from the PDR values.

The WSN-NUM protocol can be modified to use unicast data transmissions instead of broadcast (by changing the number of nodes in conflict graph to include separate transmissions to each receiver, instead of broadcast). This will significantly improve PDR, since collisions will be averted using RTS-CTS messages. If unicast messages are used, there will be additional overhead for forwarding data to multiple receivers via separate transmissions. This will effectively reduce the maximum capacity achievable for each flow, since each source sends data to each receiver in a separate transmission, and thereby fails to utilize the wireless broadcast advantage. We repeated our

Table IV. Comparison of Source Rates for Simulation and Implementation Using Unicast and Broadcast Transmissions

Parameter	Broadcast			Unicast		
	Optimal	Simulation	Impl.	Optimal	Simulation	Impl.
Rate	33.33	25.81	25.35	23.83	19.19	18.70
(KBPS)	16.67	13.08	11.63	11.42	9.58	9.34
	50.00	46.09	46.11	41.67	35.70	34.14
PDR (%)	100	83.30	77.70	100	97.22	96.39
Net	33.33	21.50	19.70	23.83	18.65	18.01
Rate	16.67	10.90	9.03	11.416	9.31	8.99
(KBPS)	50.00	38.39	35.82	41.67	34.7	32.88
Utility	63.01	60.92	60.34	61.41	60.17	59.90

The three numbers in the rate fields indicate the source rates of nodes 1, 2, and 3, respectively.

experiment with unicast transmissions, and these observations are evident from Table IV, where the optimal rates with unicast transmission are much lower than with broadcast, but the PDR is high, making the net rate only slightly lower than the broadcast case.

Depending on the network topology and mission flow subscriptions, the unicast may or may not be better than broadcast in terms of utility, based on the net received rate. The motivation behind opting for broadcast transmission during the initial design of the protocol is because in mission-oriented WSNs there are typically multiple receivers, and using broadcast substantially reduces the number of transmissions, thereby increasing the throughput of the network. If the missions are loss-tolerant, broadcast transmissions are suitable. Otherwise, unicast transmission should be considered. Another possibility is to employ unicast transmissions when there is a single receiver (since RTS-CTS averts collision in this case), and broadcast only when there are more than one receiver. This hybrid mechanism does not require any change to the model or protocol (since multiple receivers still receive only a single broadcast transmission); only the destination address at each single-receiver hop is changed from multicast to unicast. This improved the PDR to $\sim$84% in our implementation.

8. CONCLUSIONS AND FUTURE WORK

We demonstrated a distributed optimization technique for resource sharing in mission-oriented WSNs, which is characterized by joint-utility functions and multicast dissemination of sensor data. Key enhancements include the receiver-centric computation of utilities, the association of congestion costs with transmission-specific cliques (instead of links), and the adaptation of rates by each sensor, based on the sum of feedback parameters. The technique converges to the optimal cumulative utility if the utility functions are jointly concave and can be viewed as a generalization of earlier work for independent utilities.

Protocol-level simulation and real-time implementation over an 802.11b-based WSN provide the first known practical evidence that the suggested NUM technique can indeed be implemented with low overhead, can effectively regulate the rate of the sensor sources, and can also adapt rapidly to variations in network topologies. Based on our performance studies, we find the existence of a trade-off between convergence speed and optimality. In general, smaller values of κ provide solutions closer to the optimal but take longer to reach convergence, with the problem being exacerbated at high loads. Our empirical studies show that, at least in 802.11 environments, the use of link-layer multicast forwarding results in higher source data rates but lower PDR; conversely, using multiple link-layer unicast transmissions result in lower source rates but fairly high PDR.

As future work, we will extend the WSN-NUM model to accommodate node mobility. We will also analyze comprehensive methods to tune ϵ and other parameters, such as feedback frequency, dynamically.

REFERENCES

ATHURALIYA, S. AND LOW, S. 2000. Optimization flow control with newton-like algorithm. *J. Telecomm. Syst. 15*, 345–358.

BERTSEKAS, D. 1999. *Non-Linear Programming*. Athena Scientific.

BUI, L., SRIKANT, R., AND STOLYAR, A. 2007. Optimal resource allocation for multicast flows in multihop wireless networks. In *Proceedings of the IEEE Conference on Decision and Control*. 1134–1139.

CHEN, L., LOW, S., CHIANG, M., AND DOYLE, J. 2006. 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'06)*. 1–13.

CHEN, L., LOW, S., AND DOYLE, J. 2005. Joint congestion control and media access control design for ad hoc wireless networks. In *Proceedings of the 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'05)*. 2212–2222.

CHIANG, M. 2005. Balancing transport and physical layers in wireless multihop networks: jointly optimal congestion control and power control. *IEEE J. Select. Areas Comm. 23,* 1, 104–116.

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

CHOU, C., LIU, B., AND MISRA, A. 2007. Maximizing broadcast and multicast traffic load through link-rate diversity in wireless mesh networks. In *Proceedings of the IEEE International Symposium on World of Wireless, Mobile and Multimedia Networks (WoWMoM'07)*. 1–12.

CURESCU, C. AND NADJM-TEHRANI, S. 2008. A bidding algorithm for optimized utility-based resource allocation in ad hoc networks. *IEEE Trans. Mobile Comput. 7,* 12, 1397–1414.

ERYILMAZ, A. AND SRIKANT, R. 2006. Joint congestion control, routing, and mac for stability and fairness in wireless networks. *IEEE J. Select. Areas Comm. 24,* 8, 1514–1524.

ESWARAN, S., JOHNSON, M., MISRA, A., AND PORTA, T. F. L. 2009. Adaptive in-network processing for bandwidth and energy constrained mission-oriented multi-hop wireless networks. In *Proceedings of the IEEE/ACM International Conference on Distributed Computing in Sensor Systems*.

ESWARAN, S., MISRA, A., AND PORTA, T. L. 2008a. Addressing practical challenges in utility optimization of mobile wireless sensor networks. In *Proceedings of the SPIE Defense and Security Symposium*.

ESWARAN, S., MISRA, A., AND PORTA, T. L. 2008b. Utility-based adaptation in mission-oriented wireless sensor networks. In *Proceedings of the IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*. 278–286.

GAMS. www.gams.com.

GUPTA, P. AND KUMAR, P. 2000. The capacity of wireless networks. *IEEE Trans. Info. Theor. 46,* 2, 388–404.

KAR, K., SARKAR, S., AND TASSIULAS, L. 2002. A scalable low-overhead rate control algorithm for multirate multicast sessions. *IEEE J. Select. Areas Comm. 20,* 8, 1541–1557.

KELLY, F. 1997. Charging and rate control for elastic traffic. *Euro. Trans. Telecomm. 8*, 33–37.

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

LA, R. AND ANANTHARAM, V. 2002. Utility-based rate control in the internet for elastic traffic. *IEEE/ACM Trans. Netw. 10,* 2, 272–286.

LIN, X. AND SHROFF, N. 2004. Joint rate control and scheduling in multihop wireless networks. In *Proceedings of the 43rd IEEE Conference on Decision and Control.* 1484–1489.

LOW, S. AND LAPSLEY, D. 1999. Optimization flow control,i: Basic algorithm and convergence. *IEEE/ACM Trans. Netw. 7*, 861–874.

MULLIGAN, G. AND CORNEIL, D. 1972. Corrections to bierstone's algorithm for generating cliques. *J. ACM 19,* 2, 244–247.

PALOMAR, D. AND CHIANG, M. 2007. Alternative distributed algorithms for network utility maximization: Framework and applications. *IEEE Trans. Autom. Control. 52,* 12, 2254–2269.

QUALNET. www.qualnet.com.

RANGWALA, S., GUMMADI, R., GOVINDAM, R., AND PSOUNIS, K. 2006. Interference-aware fair rate control in wireless sensor networks. In *Proceedings of the ACM SIGCOMM 36*, 63–74.

SCRIPTS. www.cse.psu.edu/ eswaran/WSNNUM.zip.

SENGUPTA, S., RAYANCHU, S., AND BANERJEE, S. 2007. An analysis of wireless network coding for unicast sessions: The case for coding-aware routing. In *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM'07)*. 1028–1036.

SHAPIRO, J., TOWSLEY, D., AND KUROSE, J. 2002. Optimization-based congestion control for multicast communications. *IEEE Comm. Mag. 40,* 9, 90–95.

SRIDHARAN, A. AND KRISHNAMACHARI, B. 2007. Maximizing network utilization with max-min fairness in wireless sensor networks. In *Proceedings of the 5th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks and Workshops (WiOpt'07)*. 1–9.

TAN, C., PALOMAR, D., AND CHIANG, M. 2006. Distributed optimization of coupled systems with applications to network utility maximization. In *Proceedings of the IEEE Conference on Acoustics, Speech and Signal Processing*. Vol. 5.

WANG, X. AND KAR, K. 2006. Cross-layer rate optimization for proportional fairness in multihop wireless networks with random access. *IEEE J. Select. Areas Comm. 24,* 8, 1548–1559.

WRIGHT, J., GIBSON, C., BERGAMASCHI, F., MARCUS, K., PHAM, T., PRESSLEY, R., AND VERMA, G. 2009. Ita sensor fabric. In *Proceedings of the SPIE Defense and Security Symposium*.

XUE, Y., LI, B., AND NAHRSTEDT, K. 2006. Optimal resource allocation in wireless ad hoc networks: a price-based approach. *IEEE Trans. Mobile Comput. 5,* 4, 347–364.

YANG, Y., WANG, J., AND KRAVETS, R. 2005. Interference-aware load balancing for multihop wireless networks. Tech. rep.