Hindawi Publishing Corporation
Journal of Electrical and Computer Engineering
Volume 2016, Article ID 5762851, 10 pages
http://dx.doi.org/10.1155/2016/5762851

Research Article

Cross-Layer Control with Worst Case Delay Guarantees in Multihop Wireless Networks

Shu Fan and Honglin Zhao

Communication Research Center, Harbin Institute of Technology, Harbin 150080, China

Correspondence should be addressed to Honglin Zhao; hlzhao@hit.edu.cn

Received 15 December 2015; Accepted 6 June 2016

Academic Editor: Vinod Sharma

The delay guarantee is a challenge to meet different real-time requirements in applications of backpressure-based wireless multihop networks, and therefore, researchers are interested in the possibility of providing bounded end-to-end delay. In this paper, a new cross-layer control algorithm with worst case delay guarantees is proposed. The utility maximization algorithm is developed using a Lyapunov optimization framework. Virtual queues that ensure the worst case delay of nondropped packets are designed. It is proved through rigorous theoretical analyses and verified by simulations that the time average overall utility achieved by the new algorithm can be arbitrarily close to the optimal solution with finite queue backlogs. The simulation results evaluated with Matlab show that the proposed algorithm achieves higher throughput utility with fewer data dropped compared with the existing work.

1. Introduction

With the exponential increase in wireless multihop networks in the last two decades, increasingly sophisticated approaches that target resource allocation, congestion control, routing, and scheduling have been developed. Among the various policies that have been developed, the backpressure scheduling/routing policy, which was first proposed in the seminal work by Tassiulas and Ephremides [1], is a promising scheme because of its optimal throughput characteristic. Cross-layer algorithms that provide throughput utility optimal operation guarantees for different network structures can be designed by applying the Lyapunov optimization technique and by combining the backpressure scheme with flow control [2]. The flow controller at the transport layer ensures that the admitted rate injected into the network layer lies within the network capacity region. In recent works, spectrum sharing and pricing mechanisms [3], energy management [4], and social selfishness of users [5] have been considered in backpressure-based cross-layer algorithms. Cross-layer algorithms have also been combined with MAC (Media Access Control) layer [6], TCP (Transmission Control Protocol) layer [7], and application layers [8].

Besides throughput utility, end-to-end delay is another important long-term performance metric of the backpressure style algorithms, and it is crucial to many essential applications. As applications with real-time requirements are being developed, it is necessary to design backpressure-based algorithms that provide bounded worst case delay guarantees. Backpressure algorithms usually bear poor delay performance mainly attributed to the following three reasons. First, the slow startup process to form a stable queue backlog gradient from the source to the destination causes large initial end-to-end delay. Second, unnecessarily long or looped paths form owing to the fluctuation of the queue backlog. Finally, the absence of consistent backpressure towards the destination can cause large latency in networks with short-lived or low-rate flows. In [9], average delay bounds are derived for one-hop wireless networks using maximal scheduling. In [10], the delay bounds in wireless ad hoc networks are studied using backpressure scheduling with either one-hop or multihop traffic flows. In [11], the authors propose a cross-layer algorithm providing average end-to-end delay guarantees. These prior works can only provide bounds on the overall average delay via Little's Theorem, except for individual sessions. There are several works aiming to reduce end-to-end delay for individual sessions. In [12], a virtual queue-based gradient is established for nodes. In [13], the authors develop a delay-aware cross-layer algorithm using a novel link-rate allocation strategy and a regulated scheduling policy. A hop-count based queuing structure is used in [14] to provide a worst case hop count to the destination. However, these works fail to provide explicit end-to-end delay guarantees. Deterministic worst case delay guarantees are derived from the algorithm in [15] which uses explicit delay information from the head-of-line packet at each queue in one-hop networks. Considering both one-hop and multihop wireless networks, [16] designs an opportunistic scheduling scheme that guarantees a bounded worst case delay for each session. Our paper is mostly related to the study in [16]. However, different from [16], our algorithm consists of two phases and the persistent service virtual queue [16] is redesigned.

The key contributions of this paper can be summarized as follows.

(i) The paper proposes a two-phase algorithm which can provide a bound on the worst case end-to-end delay of individual sessions by designing a novel virtual delay queue structure.

(ii) By transforming the stochastic control problem into a deterministic optimization problem using the Lyapunov drift-plus-penalty technique, we design a joint congestion control, routing, and scheduling algorithm.

(iii) The performance in terms of utility optimality and network stability of the algorithm is demonstrated with rigorous theoretical analyses. It is shown that the proposed algorithm can achieve a time average throughput utility that can be arbitrarily close to the optimal value, with queue backlogs being bounded by constants.

The remainder of this paper is organized as follows. Section 2 introduces the system model and problem formulation. In Section 3, the algorithm is designed using Lyapunov optimization. The performance analyses of the proposed algorithm are presented in Section 4. The simulation results are given in Section 5. Conclusions are provided in Section 6.

2. Network Model and Problem Formulation

2.1. Network Model. Consider a multihop wireless network consisting of several nodes. Let the network be modeled by a directed connectivity graph $G(N, L)$, where N is the set of nodes and $(i, j) \in L$ represents a unidirectional wireless link between node i and node j which is in the transmission range of i. Let M be the set of unicast sessions m between source-destination pairs in the network. N_s is the set of source nodes s_m and N_d is the set of destination nodes d_m of session m. Packets from the source node traverse multiple wireless hops before arriving at the destination node.

The system is assumed to run in a time-slotted fashion. Nodes in the network communicate using only one channel. $a_{nj}(t) \in \{0, 1\}$ is used to indicate whether link (n, j) is used to transmit packets in time slot t. $a_{nj}(t) = 1$ implies that the link is scheduled. In this model, scheduling is subjected to the following constraints:

$$\sum_{j:(n,j)\in L} \alpha_{nj}(t) + \sum_{i:(i,n)\in L} \alpha_{in}(t) \leq 1, \tag{1}$$

$$\alpha_{nj}(t) + \sum_{k\in N} \sum_{l} \alpha_{kl}(t) \leq 1, \tag{2}$$

where node l is in the transmission range of n and $O(n)$ denotes the set of nodes with $(n, i) \in L$. $I(n)$ denotes the set of nodes with $(j, n) \in L$. Constraint (1) implies that each node is equipped with only one radio, and thus, it can either transmit or receive data at any given time. Constraint (2) states that a node transmitting packets will interfere with the data receptions of the nodes in its transmission range.

2.2. Virtual Queue at the Transport Layer. $A_m(t) \in [0, A_{\max}^{(m)}]$ denotes the arrival rate of session m injected into the transport layer from the application layer at the source node and $A_{\max}^{(m)}$ is the maximum arrival rate of session m. $r_m(t) \in [0, A_m(t)]$ is the admitted rate of session m injected into the network layer. $\eta_m(t) \in [0, A_{\max}^{(m)}]$ is an auxiliary variable known as the virtual input rate. The virtual queue at the transport layer of source node s_m of session m is denoted by Y_m that is updated as follows:

$$Y_m(t+1) = \max\{Y_m(t) - r_m(t), 0\} + \eta_m(t). \tag{3}$$

If each virtual queue Y_m is guaranteed to be stable, according to the necessity and sufficiency for queue stability [17], it is apparent that $\overline{\eta_m} \leq \overline{r_m}$, where the time average value of time-varying variable $x(t)$ is denoted by $\overline{x} = \lim_{t\to\infty}(1/t)\sum_{\tau=0}^{t-1} E(x(\tau))$. Therefore, the lower bound of $\overline{r_m}$ can be derived from $\overline{\eta_m}$ which can be calculated.

2.3. Data Queue at the Network Layer. The data backlog queue for session m at the network layer of node n is denoted by $Q_n^{(m)}(t)$. In each slot t, the queue is updated as

$$\begin{aligned} Q_n^{(m)}&(t+1) \\ &= \max\left\{Q_n^{(m)}(t) - \sum_{i\in O(n)} \mu_{ni}^{(m)}(t) - D_n^{(m)}(t), 0\right\} \\ &\quad + \sum_{j\in I(n)} \mu_{jn}^{(m)}(t) + 1_{\{n=s_m\}} r_m(t), \end{aligned} \tag{4}$$

where $\mu_{ni}^{(m)}(t)$ is the amount of data of session m to be forwarded from node n to i in time slot t. $1_{\{n=s_m\}}$ is an indicator function that denotes 1 if $n = s_m$ and 0 otherwise. In addition, $\sum_{m\in M} \mu_{ni}^{(m)}(t)$ must not be greater than $\mu_n^{\max,\text{out}}$. $D_n^{(m)} \in [0, D_{\max}]$ represents the number of packets of session m that are dropped by node n in slot t. The optimization of $\mu_{ni}^{(m)}(t)$ is the routing decision. As assumed in [18], in this paper, the transmission capacity of any link is set to be 1.

Therefore, $\mu_{ni}^{(m)}(t)$ is either 0 or 1, and it can not be greater than $Q_n^{(m)}(t)$, which is denoted as

$$\mu_{ni}^{(m)}(t) \in \left\{0, \min\left\{Q_n^{(m)}(t), 1\right\}\right\}, \quad \forall (n,i) \in L,\ n \neq d_m,\ m \in M, \tag{5}$$

and $\sum_{m\in M} \mu_{ni}^{(m)}(t) = \alpha_{ni}(t), \forall (n,i) \in L$, can also be derived. The actual amount of packets of session m dropped in slot t can be defined as

$$\widetilde{D_n^{(m)}}(t) = \min\left\{Q_n^{(m)}(t) - \mu_n^{(m)}(t), D_n^{(m)}(t)\right\}. \tag{6}$$

2.4. Persistent Service Virtual Queue. The ϵ-persistent service queue designed in [16] can ensure bounded worst case delay for general types of utility functions. We denote this queue by $G_n^{(m)}$, and in each slot, the queue is updated as

$$\begin{aligned} G_n^{(m)}(t+1) = \max\Bigg\{ & G_n^{(m)}(t) + 1_{\{Q_n^{(m)}(t)>0\}} \\ & \cdot \Bigg(\epsilon - \sum_{i\in O(n)} \mu_{ni}^{(m)}(t)\Bigg) - D_n^{(m)}(t) - 1_{\{Q_n^{(m)}(t)=0\}} \\ & \cdot \mu_n^{\max,\text{out}}, 0\Bigg\}. \end{aligned} \tag{7}$$

From the algorithm in [16] we find that $G_n^{(m)}$ is used in decision of resource allocation and packet dropping. Since in most slots $Q_n^{(m)}(t) > 0$, $G_n^{(m)}$ may increase fast. According to the packet drop decision algorithm, high value of ϵ-persistent service queue leads to serious packets drop. Therefore, the fast increase of $G_n^{(m)}$ leads to dropping of packets and this results in a significant drop in throughput utility.

In this paper, we redesign the ϵ-persistent service queue that is denoted by $Z_n^{(m)}$. In each slot t, the queue is updated as

$$\begin{aligned} Z_n^{(m)}(t+1) = \max\Bigg\{ & Z_n^{(m)}(t) + \epsilon_1 \cdot 1_{\{Q_n^{(m)}(t)>Q_{n,\text{standard}}^{(m)}\}} \\ & + \epsilon_2 \cdot 1_{\{Q_n^{(m)}(t)\le Q_{n,\text{standard}}^{(m)}\}} - D_n^{(m)}(t) \\ & - \sum_{i\in O(n)} \mu_{ni}^{(m)}(t), 0\Bigg\}, \end{aligned} \tag{8}$$

where $\epsilon_1 > \epsilon_2 > 0$. ϵ_1 and ϵ_2 are constants. $Q_{n,\text{standard}}^{(m)}$ is a constant value that is calculated in phase I of the algorithm which will be given in Section 3. Initial backlog $Z_n^{(m)}(0)$ is supposed to be 0.

$Q_{n,\text{standard}}^{(m)}$ is the time average of length of queue of session m in node n. According to (8), $Z_n^{(m)}$ increases fast only when $Q_n^{(m)}(t) > Q_{n,\text{standard}}^{(m)}$, and thus $Z_n^{(m)}$ should grow slower than $G_n^{(m)}$. According to the packet drop decision algorithm, the number of packets dropped in our new algorithm should decrease and throughput should increase, compared with the algorithm in [16].

Any algorithm that maintains bounded $Q_n^{(m)}(t)$ and $Z_n^{(m)}(t)$ ensures persistent service with bounded worst case delay, as shown in Theorem 1.

Theorem 1 (worst case delay). *For all time slots $t \in \{0, 1, 2, \ldots\}$ and all sessions $m \in M$, suppose that the algorithm can ensure*

$$\begin{aligned} Q_n^{(m)}(t) &\le Q_n^{(m),\max}, \\ Z_n^{(m)}(t) &\le Z_n^{(m),\max}, \end{aligned} \tag{9}$$

where $Q_n^{(m),\max}$ and $Z_n^{(m),\max}$ are finite upper bounds for $Q_n^{(m)}(t)$ and $Z_n^{(m)}(t)$, respectively. Assuming First Input First Output (FIFO) service, the worst case delay of the nondropped data at node n is bounded by the constant $W_n^{(m),\max}$, which is given as

$$W_n^{(m),\max} = \left\lceil \frac{\left(Q_n^{(m),\max} + Z_n^{(m),\max}\right)}{\epsilon_2} \right\rceil, \tag{10}$$

where $\lceil x \rceil$ denotes the smallest integer that is greater than or equal to x.

Proof. Fix any slot $t \ge 0$, and let $A_n^{(m)}(t)$ represent the data that arrives at queue $Q_n^{(m)}$ on slot t. As the service is FIFO, the data $A_n^{(m)}(t)$ is placed at the end of the queue $Q_n^{(m)}$ on slot $t+1$. We want to prove that all of the data $A_n^{(m)}(t)$ departs queue $Q_n^{(m)}$ on or before slot $t + W_n^{(m),\max}$. We prove this in three cases.

Case 1. If $Q_n^{(m)}(\tau) > Q_{n,\text{standard}}^{(m)}$ for all $\tau \in \{t+1, \ldots, t + W_n^{(m),\max}\}$, the following can be derived

$$\begin{aligned} Z_n^{(m)}(\tau+1) \ge Z_n^{(m)}(\tau) + \epsilon_1 - \sum_{i\in O(n)} \mu_{ni}^{(m)}(\tau) \\ - D_n^{(m)}(\tau). \end{aligned} \tag{11}$$

Summing the above over $\tau \in \{t+1, \ldots, t + W_n^{(m),\max}\}$ yields

$$\begin{aligned} & Z_n^{(m)}\left(t+1+W_n^{(m),\max}\right) - Z_n^{(m)}(t+1) \\ & \quad \ge \epsilon_1 \cdot W_n^{(m),\max} \\ & \quad - \sum_{\tau=t+1}^{t+W_n^{(m),\max}} \left[\sum_{i\in O(n)} \mu_{ni}^{(m)}(\tau) + D_n^{(m)}(\tau)\right]. \end{aligned} \tag{12}$$

For $Z_n^{(m)}(t+1+W_n^{(m),\max}) \le Z_n^{(m),\max}$, (12) can be rearranged to yield

$$\begin{aligned} & \epsilon_1 \cdot W_n^{(m),\max} - Z_n^{(m),\max} \\ & \quad \le \sum_{\tau=t+1}^{t+W_n^{(m),\max}} \left[\sum_{i\in O(n)} \mu_{ni}^{(m)}(\tau) + D_n^{(m)}(\tau)\right]. \end{aligned} \tag{13}$$

According to (10), $W_n^{(m),\max}$ is the smallest integer that is greater than or equal to $(Q_n^{(m),\max} + Z_n^{(m),\max})/\epsilon_2$. Therefore, we can get $W_n^{(m),\max} \geq (Q_n^{(m),\max} + Z_n^{(m),\max})/\epsilon_2$, and we can derive

$$\begin{aligned}&\epsilon_1 \cdot W_n^{(m),\max} - Z_n^{(m),\max} \\ &\quad \geq Q_n^{(m),\max} + Z_n^{(m),\max} - Z_n^{(m),\max} = Q_n^{(m),\max}.\end{aligned} \tag{14}$$

Then, the following can be derived:

$$\begin{aligned}&\sum_{\tau=t+1}^{t+W_n^{(m),\max}} \left[\sum_{i \in O(n)} \mu_{ni}^{(m)}(\tau) + D_n^{(m)}(\tau) \right] \geq Q_n^{(m),\max} \\ &\quad \geq Q_n^{(m)}(t+1).\end{aligned} \tag{15}$$

Equation (15) means that all the data in queue $Q_n^{(m)}$ on slot $t+1$ (including all of the data $A_n^{(m)}(t)$ which arrives at $Q_n^{(m)}$ on slot t) can depart the queue on or before the slot $t + W_n^{(m),\max}$. Therefore, in the condition of $Q_n^{(m)}(\tau) > Q_{n,\text{standard}}^{(m)}$ for all $\tau \in \{t+1, \ldots, t + W_n^{(m),\max}\}$, the worst case delay of nondropped data at node n is bounded by

$$W_{n,\text{Case 1}}^{(m),\max} = \left\lceil \frac{\left(Q_n^{(m),\max} + Z_n^{(m),\max}\right)}{\epsilon_2} \right\rceil. \tag{16}$$

Case 2. If $Q_n^{(m)}(\tau) \leq Q_{n,\text{standard}}^{(m)}$ for all $\tau \in \{t+1, \ldots, t + W_n^{(m),\max}\}$, the following can be derived:

$$\begin{aligned}&Z_n^{(m)}\left(t+1+W_n^{(m),\max}\right) - Z_n^{(m)}(t+1) \\ &\quad \geq \epsilon_2 \cdot W_n^{(m),\max} \\ &\quad - \sum_{\tau=t+1}^{t+W_n^{(m),\max}} \left[\sum_{i \in O(n)} \mu_{ni}^{(m)}(\tau) + D_n^{(m)}(\tau) \right].\end{aligned} \tag{17}$$

Similar to Case 1, the bound of the worst case delay of nondropped data at node n in Case 2 is derived as

$$W_{n,\text{Case 2}}^{(m),\max} = \left\lceil \frac{\left(Q_n^{(m),\max} + Z_n^{(m),\max}\right)}{\epsilon_2} \right\rceil. \tag{18}$$

Case 3. If $Q_n^{(m)}(\tau) > Q_{n,\text{standard}}^{(m)}$ in n_1 slots of $\tau \in \{t+1, \ldots, t + W_n^{(m),\max}\}$ and $Q_n^{(m)}(\tau) \leq Q_{n,\text{standard}}^{(m)}$ in n_2 slots of $\tau \in \{t+1, \ldots, t + W_n^{(m),\max}\}$ and if we also have $n_1 + n_2 = W_n^{(m),\max}$, the following can be derived:

$$\begin{aligned}&Z_n^{(m)}\left(t+1+W_n^{(m),\max}\right) - Z_n^{(m)}(t+1) \\ &\quad \geq \epsilon_1 \cdot n_1 + \epsilon_2 \cdot n_2 \\ &\quad - \sum_{\tau=t+1}^{t+W_n^{(m),\max}} \left[\sum_{i \in O(n)} \mu_{ni}^{(m)}(\tau) + D_n^{(m)}(\tau) \right].\end{aligned} \tag{19}$$

Let $\epsilon_* = (\epsilon_1 \cdot n_1 + \epsilon_2 \cdot n_2)/ W_n^{(m),\max}$; we can get

$$\begin{aligned}&Z_n^{(m)}\left(t+1+W_n^{(m),\max}\right) - Z_n^{(m)}(t+1) \\ &\quad \geq \epsilon_* \cdot W_n^{(m),\max} \\ &\quad - \sum_{\tau=t+1}^{t+W_n^{(m),\max}} \left[\sum_{i \in O(n)} \mu_{ni}^{(m)}(\tau) + D_n^{(m)}(\tau) \right].\end{aligned} \tag{20}$$

It is easy to derive $\epsilon_2 < \epsilon_* < \epsilon_1$. Similar to Case 1, the bound of the worst case delay of nondropped data at node n in Case 3 is

$$W_{n,\text{Case 3}}^{(m),\max} = \left\lceil \frac{\left(Q_n^{(m),\max} + Z_n^{(m),\max}\right)}{\epsilon_2} \right\rceil. \tag{21}$$

Considering the three cases above, the worst case delay of the nondropped data at node n is bounded by the constant $W_n^{(m),\max}$, which is given as

$$W_n^{(m),\max} = \left\lceil \frac{\left(Q_n^{(m),\max} + Z_n^{(m),\max}\right)}{\epsilon_2} \right\rceil. \tag{22}$$

□

2.5. Throughput Utility Optimization Problem. Similar to the design of the utility function in [16], let $U_m(\cdot)$ be strictly concave, twice differentiable, and nondecreasing utility function with $U_m(0) = 0$. $\overline{r} = (\overline{r_m}, m \in M)$ denotes the throughput of the network. Λ is the capacity region of the network [17]. Then, the throughput utility maximization problem $P1$ can be defined as follows:

$$\max \quad \sum_{m \in M} U_m(\overline{r_m}) - \sum_{m \in M} \sum_{n \in N} \beta_m \overline{d_n^{(m)}}, \tag{23}$$

$$\text{s.t.} \quad \overline{r} \in \Lambda, \tag{24}$$

$$(1), (2), (5), \tag{25}$$

where $\overline{d_n^{(m)}}$ is the time average value of $D_n^{(m)}(t)$. β_m is the maximum slope of the utility function $U_m(x)$. Constraint (24) means that the network stability is guaranteed.

3. Dynamic Algorithm via Lyapunov Optimization

The Lyapunov optimization technique is applied to solve $P1$. $Q_n^{(m)}$ $(\forall n \in N, m \in M)$, Y_m $(\forall m \in M)$, and $Z_n^{(m)}$ $(\forall n \in N, m \in M)$ are used in the dynamic algorithm. Let $\Theta(t) = [Q(t), Y(t), Z(t)]$ be the network state vector in time slot t. Define the Lyapunov function as

$$\begin{aligned}L(\Theta(t)) = \frac{1}{2} \Bigg[&\sum_{m \in M} (Y_m(t))^2 + \sum_{n \in N} \sum_{m \in M} \left(Q_n^{(m)}(t)\right)^2 \\ &+ \sum_{n \in N} \sum_{m \in M} \left(Z_n^{(m)}(t)\right)^2 \Bigg].\end{aligned} \tag{26}$$

The conditional Lyapunov drift in time slot t is

$$\Delta(\Theta(t)) = E\{L(\Theta(t+1)) - L(\Theta(t)) \mid \Theta(t)\}. \quad (27)$$

To maximize a lower bound for $\sum_{m\in M} U_m(\overline{r_m}) - \sum_{m\in M}\sum_{n\in N}\beta_m \overline{d_n^{(m)}}$, the drift-plus-penalty function can be defined as

$$\Delta_V(\Theta(t)) = \Delta(\Theta(t)) - VE\left\{\sum_{m\in M} U_m(\eta_m(t)) - \sum_{m\in M}\sum_{n\in N}\beta_m D_n^{(m)}(t) \mid \Theta(t)\right\}, \quad (28)$$

where V is the weight of the utility defined by the user. The following inequality can be derived:

$$E\{\Delta_V(\Theta(t))\} \le B - \Psi_1(t) - \Psi_2(t) - \Psi_3(t) - \Psi_4(t) + \sum_{n\in N}\sum_{m\in M} Z_n^{(m)} \cdot \epsilon_1, \quad (29)$$

where $\Psi_1(t)$, $\Psi_2(t)$, $\Psi_3(t)$, and $\Psi_4(t)$ can be evaluated as follows:

$$\begin{aligned}
\Psi_1(t) &= \sum_{m\in M}\left[V\cdot U_m(\eta_m(t)) - Y_m(t)\cdot\eta_m(t)\right],\\
\Psi_2(t) &= \sum_{m\in M} r_m(t)\cdot\left[Y_m(t) - Q_n^{(m)}(t)\cdot 1_{\{n=s_m\}}\right],\\
\Psi_3(t) &= \sum_{n\in N}\sum_{m\in M}\sum_{i\in O(n)}\mu_{ni}^{(m)}(t) \cdot\left[Q_n^{(m)}(t) - Q_i^{(m)}(t) + Z_n^{(m)}(t)\right],\\
\Psi_4(t) &= \sum_{n\in N}\sum_{m\in M} D_n^{(m)}(t) \cdot\left[Q_n^{(m)}(t) + Z_n^{(m)}(t) - V\cdot\beta_m\right].
\end{aligned} \quad (30)$$

B is a constant and satisfies

$$\begin{aligned}
B \ge{}& \frac{1}{2}\sum_{m\in M}\left[(\eta_m(t))^2 + (r_m(t))^2\right] + \frac{1}{2} \cdot \sum_{m\in M}\sum_{n\in N}\left[\left(\sum_{j\in I(n)}\mu_{jn}^{(m)}(t) + 1_{\{n=s_m\}}r_m(t)\right)^2 \right.\\
&\left. + \left(\sum_{i\in O(n)}\mu_{ni}^{(m)}(t) + D_n^{(m)}(t)\right)^2\right] + \frac{1}{2}\sum_{m\in M}\sum_{n\in N}\left[\epsilon_1 - \sum_{i\in O(n)}\mu_{ni}^{(m)}(t) - D_n^{(m)}(t)\right]^2,
\end{aligned} \quad (31)$$

according to $0 \le \eta_m(t) \le A_{\max}^{(m)}$, $0 \le r_m(t) \le A_{\max}^{(m)}$, $\mu_{ni}^{(m)}(t) \in \{0,1\}$, and $0 \le D_n^{(m)}(t) \le D_{\max}$, and ϵ_1 is a constant value and the constant B must exist.

The algorithm CCWD is based on the drift-plus-penalty framework [17] and the main design principle of the algorithm is to minimize the right-hand side of (29). The algorithm includes two phases.

Phase I. Choose a sufficiently large T. From time $t = 0, \ldots, T-1$, run the algorithm proposed in [16] using $Q_n^{*(m)}(t)$ as the size of data queues. Set $Q_{n,\text{standard}}^{(m)}$ to be $(1/T)\sum_{\tau=0}^{T-1} Q_n^{*(m)}(\tau) + \rho$, where ρ is a constant.

Phase II. This phase includes five components.

Source Rate Control. For each session $m \in M$ at source node s_m, the admitted rate $r_m(t)$ is chosen to solve

$$\max \quad r_m(t)\cdot\left[Y_m(t) - Q_n^{(m)}(t)\cdot 1_{\{n=s_m\}}\right], \quad (32)$$

$$\text{s.t.} \quad 0 \le r_m(t) \le A_m(t). \quad (33)$$

Problem (32) is a linear optimization problem, and if $Y_m(t) > Q_{s_m}^{(m)}(t)$, $r_m(t)$ is set to be $A_m(t)$; otherwise it is set to be zero.

Virtual Input Rate Control. For each session $m \in M$ at source node s_m, the virtual input rate $\eta_m(t)$ is chosen to solve

$$\max \quad V\cdot U_m(\eta_m(t)) - Y_m(t)\cdot\eta_m(t), \quad (34)$$

$$\text{s.t.} \quad 0 \le \eta_m(t) \le A_{\max}^{(m)}. \quad (35)$$

Since the utility function $U_m(\cdot)$ is strictly concave and twice differentiable, (34) is a concave maximization problem with linear constraint. $\eta_m(t)$ can be chosen by

$$\eta_m(t) = \max\left\{\min\left\{U_m'^{-1}\left(\frac{Y_m(t)}{V}\right), A_{\max}^{(m)}\right\}, 0\right\}, \quad (36)$$

where $U_m'^{-1}(\cdot)$ is the inverse function of $U_m'(\cdot)$ that is the first-order derivative of $U_m(\cdot)$. Since the utility function $U_m(\cdot)$ is strictly concave and twice differentiable, $U_m'(\cdot)$ must be a monotonic function, and therefore, $U_m'^{-1}(\cdot)$ must exist.

Joint Routing and Scheduling. At the node $n \in N$, routing and scheduling decisions $\mu_{ni}^{(m)}(t)$ for each session $m \in M$ can be made by solving the following:

$$\begin{aligned}
\max \quad & \sum_{n\in N}\sum_{m\in M}\sum_{i\in O(n)}\mu_{ni}^{(m)}(t)\cdot\left[Q_n^{(m)}(t) - Q_i^{(m)}(t) + Z_n^{(m)}(t)\right],\\
\text{s.t.} \quad & (1), (2), (5).
\end{aligned} \quad (37)$$

First, for each link (n,i), the session m^* for link (n,i) can be chosen as

$$m^* = \underset{m\in M}{\operatorname{argmax}}\left\{Q_n^{(m)}(t) - Q_i^{(m)}(t) + Z_n^{(m)}(t)\right\}. \quad (38)$$

The weight of link (n,i) is defined as $w_{ni} = Q_n^{(m^*)}(t) - Q_i^{(m^*)}(t) + Z_n^{(m^*)}(t)$. Therefore, the joint routing and scheduling problem can be reduced to the following:

$$\max \quad \sum_{n\in N}\sum_{i\in O(n)}\mu_{ni}^{(m^*)}(t)\cdot w_{ni}, \quad (39)$$

$$\text{s.t.} \quad (1), (2), (5). \quad (40)$$

Transmission rates $\mu_{ni}^{(m^*)}(t)$ are chosen based on (39) which is a tough problem. The solution requires global knowledge and a centralized algorithm.

Packet Drop Decision. For each session $m \in M$ and each node $n \in N$, choose $D_n^{(m)}$ to solve

$$\max \quad \left[Q_n^{(m)}(t) + Z_n^{(m)}(t) - V \cdot \beta_m\right] \cdot D_n^{(m)}(t), \tag{41}$$

$$\text{s.t.} \quad 0 \le D_n^{(m)}(t) \le D_{\max}. \tag{42}$$

Problem (41) is a linear optimization problem, and if $Q_n^{(m)}(t) + Z_n^{(m)}(t) > V \cdot \beta_m$, $D_n^{(m)}(t)$ is set to be $D_{\max}$; otherwise it is set to be zero.

Update of Queues. $Y(t)$, $Q(t)$, and $Z(t)$ are updated using (3), (4), and (8) in each time slot.

4. Performance Analysis

Theorem 2 (bounded queues). *Assume that $D_{\max} \ge \max\{\epsilon_1, A_{\max}^{(m)} + \mu_n^{\max,\text{in}}\}$ holds, where $\mu_n^{\max,\text{in}}$ denotes the maximal amount of packets that node n can receive from other nodes in one slot. Then under the algorithm CCWD, all queues are bounded for all $t \ge 0$ as follows:*

$$\begin{gathered} Q_n^{(m)}(t) \le Q_n^{(m),\max}, \\ Z_n^{(m)}(t) \le Z_n^{(m),\max}, \\ Y_m(t) \le Y^{(m),\max} \end{gathered} \tag{43}$$

provided that these inequalities hold at $t = 0$. The queue bounds are given by

$$\begin{aligned} Y^{(m),\max} &= V \cdot \beta_m + A_{\max}^{(m)}, \\ Q_n^{(m),\max} &= V \cdot \beta_m + \mu_n^{\max,\text{in}} + 1_{\{n=s_m\}} \cdot A_{\max}^{(m)}, \\ Z_n^{(m),\max} &= V \cdot \beta_m + \epsilon_1. \end{aligned} \tag{44}$$

Proof. The theorem is proved by induction.

(1) According to the induction principle, if $Y_m(t) \le Y^{(m),\max}$ holds for all t and supposing that $Y_m(t) \le Y^{(m),\max}$ for time slot t, it should also hold for time slot $t+1$. If $Y_m(t) \le V \cdot \beta_m$, then $Y_m(t+1) \le V \cdot \beta_m + A_{\max}^{(m)} = Y^{(m),\max}$, because Y_m can increase by at most $A_{\max}^{(m)}$ in one slot. If $V \cdot \beta_m < Y_m(t) \le Y^{(m),\max}$, because β_m is the maximum slope of the utility function $U_m(x)$, we have

$$\begin{aligned} & V \cdot U_m(\eta_m(t)) - Y_m(t) \cdot \eta_m(t) \\ & \quad \le V \cdot U_m(0) + V \cdot \beta_m \cdot \eta_m(t) - Y_m(t) \cdot \eta_m(t) \\ & \quad = V \cdot U_m(0) + \eta_m(t) \cdot (V \cdot \beta_m - Y_m(t)) \\ & \quad \le V \cdot U_m(0) = 0 \end{aligned} \tag{45}$$

with equality holding only if $\eta_m(t) = 0$. Because $\eta_m(t) = 0$ when $Y_m(t) > V \cdot \beta_m$, $Y_m(t)$ can not increase in the next slot according to (3). Thus $Y_m(t+1) \le Y_m(t) \le Y^{(m),\max}$. Then $Y_m(t) \le Y^{(m),\max}$ for all t is proved.

(2) According to the induction principle, if $Q_n^{(m)}(t) \le Q_n^{(m),\max}$ holds for all t and supposing that $Q_n^{(m)}(t) \le Q_n^{(m),\max}$ for time slot t, it should also hold for time slot $t+1$. If $Q_n^{(m)}(t) \le V \cdot \beta_m$, then

$$\begin{aligned} Q_n^{(m)}(t+1) &\le Q_n^{(m)}(t) + \sum_{j \in I(n)} \mu_{jn}^{(m)}(t) + 1_{\{n=s_m\}} r_m(t) \\ &\le V \cdot \beta_m + \mu_n^{\max,\text{in}} + 1_{\{n=s_m\}} \cdot A_{\max}^{(m)} \\ &= Q_n^{(m),\max}. \end{aligned} \tag{46}$$

If $V \cdot \beta_m < Q_n^{(m)}(t) \le Q_n^{(m),\max}$, $D_n^{(m)}(t)$ is set to be $D_{\max}$ according to the packet drop decision algorithm. Because $D_{\max} \ge \max\{\epsilon_1, A_{\max}^{(m)} + \mu_n^{\max,\text{in}}\}$, then

$$\begin{aligned} Q_n^{(m)}(t+1) &\le Q_n^{(m)}(t) - D_{\max} + A_{\max}^{(m)} + \mu_n^{\max,\text{in}} \\ &\le Q_n^{(m)}(t) \le Q_n^{(m),\max}. \end{aligned} \tag{47}$$

Then $Q_n^{(m)}(t) \le Q_n^{(m),\max}$ for all t is proved.

(3) According to the induction principle, if $Z_n^{(m)}(t) \le Z_n^{(m),\max}$ holds for all t and supposing that $Z_n^{(m)}(t) \le Z_n^{(m),\max}$ for time slot t, it should also hold for time slot $t+1$. If $Z_n^{(m)}(t) \le V \cdot \beta_m$, then

$$Z_n^{(m)}(t+1) \le Z_n^{(m)}(t) + \epsilon_1 \le V \cdot \beta_m + \epsilon_1 = Z_n^{(m),\max}. \tag{48}$$

If $V \cdot \beta_m < Z_n^{(m)}(t) \le Z_n^{(m),\max}$, $D_n^{(m)}$ is set to be $D_{\max}$ according to the packet drop decision algorithm. Because $D_{\max} \ge \max\{\epsilon_1, A_{\max}^{(m)} + \mu_n^{\max,\text{in}}\}$, then

$$\begin{aligned} Z_n^{(m)}(t+1) &\le Z_n^{(m)}(t) - D_{\max} + \epsilon_1 \le Z_n^{(m)}(t) \\ &\le Z_n^{(m),\max}. \end{aligned} \tag{49}$$

Then $Z_n^{(m)}(t) \le Z_n^{(m),\max}$ for all t is proved. □

Theorem 3 (algorithm performance). *One has $\varphi(t) = \sum_{m \in M} U_m(r_m(t)) - \sum_{m \in M} \sum_{n \in N} \beta_m D_n^{(m)}(t)$. Define $\varphi^*(t) = \sum_{m \in M} U_m(r_m^*) - \sum_{m \in M} \sum_{n \in N} \beta_m D_n^{*(m)}$ to be the optimal value of $\varphi(t)$ subject to constraints (1), (2), and (5), where r_m^* and $D_n^{*(m)}$ are the solutions to maximize $\varphi(t)$. One can have $\sum_{m \in M} U_m(\overline{r_m}) - \sum_{m \in M} \sum_{n \in N} \beta_m \overline{d_n^{(m)}} \ge \varphi^*(t) - B/V$.*

Proof. The drift-plus-penalty function (28) satisfies

$$\begin{aligned}
&\Delta(\Theta(t)) - V\cdot E\left\{\sum_{m\in M} U_m(\eta_m(t)) \right.\\
&\quad\left. - \sum_{m\in M}\sum_{n\in N}\beta_m D_n^{(m)}(t) \mid \Theta(t)\right\} \le B - V\\
&\quad\cdot E\left\{\sum_{m\in M} U_m(\eta_m(t)) \right.\\
&\quad\left. - \sum_{m\in M}\sum_{n\in N}\beta_m D_n^{(m)}(t) \mid \Theta(t)\right\} + \sum_{m\in M} Y_m(t)\\
&\quad\cdot E\{\eta_m(t) - r_m(t) \mid \Theta(t)\} + \sum_{m\in M}\sum_{n\in N} Q_n^{(m)}(t)\\
&\quad\cdot E\left\{\sum_{j\in I(n)}\mu_{jn}^{(m)}(t) - \sum_{i\in O(n)}\mu_{ni}^{(m)}(t) + 1_{\{n=s_m\}}r_m(t)\right.\\
&\quad\left. - D_n^{(m)}(t) \mid \Theta(t)\right\} + \sum_{m\in M}\sum_{n\in N} Z_n^{(m)}(t)\cdot E\left\{\epsilon_1\right.\\
&\quad\left. - \sum_{i\in O(n)}\mu_{ni}^{(m)}(t) - D_n^{(m)}(t) \mid \Theta(t)\right\},
\end{aligned} \tag{50}$$

where B is a constant value. According to Theorem 4.5 in [17] and Lemmas 5.6 and 5.7 in [19], the following inequality can be derived from (50):

$$\begin{aligned}
&\Delta(\Theta(t)) - V\\
&\quad\cdot E\left\{\sum_{m\in M} U_m(\eta_m(t)) - \sum_{m\in M}\sum_{n\in N}\beta_m D_n^{(m)}(t) \mid \Theta(t)\right\}\\
&\le B - V\\
&\quad\cdot E\left\{\sum_{m\in M} U_m(r_m^*) - \sum_{m\in M}\sum_{n\in N}\beta_m D_n^{*(m)} \mid \Theta(t)\right\}\\
&\quad - \delta_1\cdot\sum_{m\in M} Y_m(t) - \delta_2\cdot\sum_{n\in N}\sum_{m\in M} Q_n^{(m)}(t) - \delta_3\\
&\quad\cdot\sum_{n\in N}\sum_{m\in M} Z_n^{(m)}(t),
\end{aligned} \tag{51}$$

where $\delta_1, \delta_2, \delta_3 > 0$. Inequality (51) can be transformed to the exact form specified by Theorem 5.4 in [19]. According to Theorem 5.4 in [19] and the condition $\overline{\eta_m} \le \overline{r_m}$, the following inequality can be derived:

$$\begin{aligned}
&\sum_{m\in M} U_m(\overline{r_m}) - \sum_{m\in M}\sum_{n\in N}\beta_m\overline{d_n^{(m)}}\\
&\quad\ge \sum_{m\in M} U_m(\overline{\eta_m}) - \sum_{m\in M}\sum_{n\in N}\beta_m\overline{d_n^{(m)}}\\
&\quad\ge \sum_{m\in M} U_m(r_m^*) - \sum_{m\in M}\sum_{n\in N}\beta_m D_n^{*(m)} - \frac{B}{V}\\
&\quad= \varphi^*(t) - \frac{B}{V}.
\end{aligned} \tag{52}$$

Inequality (52) implies that the overall throughput utility achieved by the algorithm in this paper is within a constant gap from the optimum value. □

5. Simulation

In the simulations, the commonly used greedy maximal scheduling (GMS) method is used for schedulable link set generation for each algorithm under comparison. This method is widely used for implementing backpressure-based centralized algorithms under sophisticated networks [20].

5.1. Simulation Setup. For simulations, a network with 20 nodes randomly distributed in a square of 1600 m^2 is considered. A transmission is successful if a receiver is within the transmission range of its sender and outside the range of other concurrent senders. The transmission or interference range of a node is 15 m. There are four unicast sessions with randomly chosen sources and destinations. Data of each session is injected into the transport layer with the same rate in each slot at the source nodes. Parameter V is set as $V = [500\ \ 1000\ \ 1500\ \ 2000]$. The throughput utility function is $U(x) = \log(x+1)$. Simulations are run in Matlab R2014a. The simulation time of phase I lasts 30000 time slots. The simulation time of phase II lasts 50000 time slots. All initial queue sizes are set to be 0 and the default values are set as follows: $D_{\max} = 3$, $\beta_m = 1$, $A_{\max}^{(m)} = 2$, $\mu_n^{\max,\text{in}} = \mu_n^{\max,\text{out}} = 1$, $\epsilon_1 = 2$, $\epsilon_2 = 1$, and $\rho = 0$.

5.2. Performance Comparison. In this section, the performance of CCWD is compared with that of an existing method called NeelyOpportunistic, which too can provide bounded worst case delay. NeelyOpportunistic is proposed in [16]. The throughput utilities and the time average number of dropped packets achieved by CCWD and NeelyOpportunistic are compared in Figures 1 and 2, respectively. V is set to be 1000. The data arrival rate is set to be from 0.2 packets to 1 packet per time slot. In Figure 1, the utility achieved by CCWD is higher than that of NeelyOpportunistic. Figure 2 shows that fewer packets get dropped using CCWD than with NeelyOpportunistic. According to the packet drop decision algorithm, high value of ϵ-persistent service queue leads to serious packets drop. As mentioned in Section 2.4, the virtual queue of CCWD being redesigned in this paper grows slower

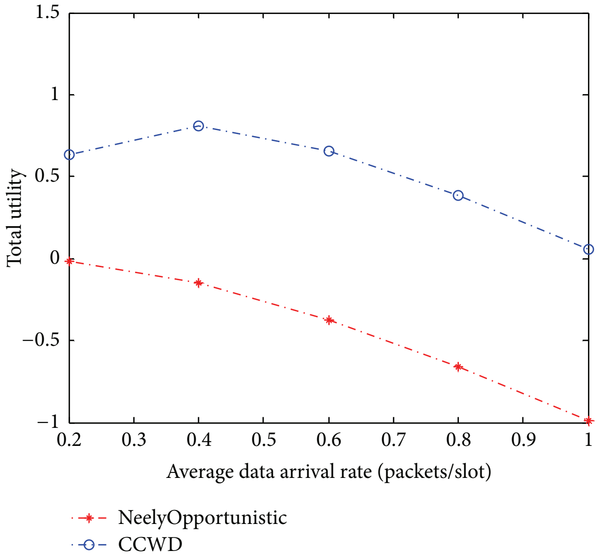


Figure 1: Throughput utility versus average data arrival rate.

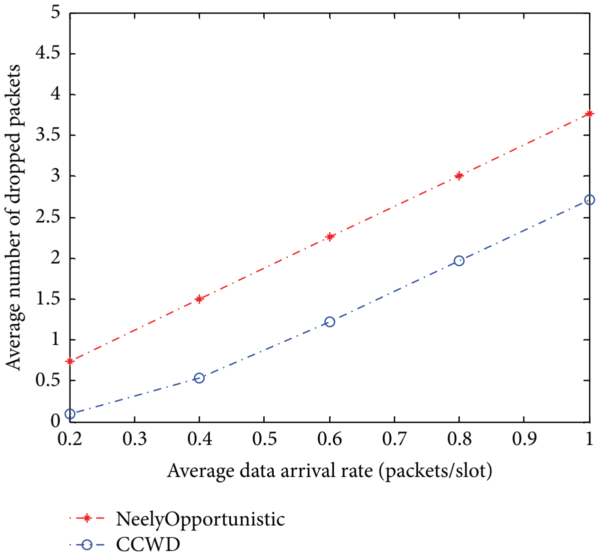


Figure 2: Time average number of dropped packets versus average data arrival rate.

than the virtual queue of NeelyOpportunistic. Therefore, the virtual queue structure in NeelyOpportunistic leads to more serious packet drop and lower throughput utility.

5.3. Impact of V. According to the analyses in Section 4, with the increase of V, the utility achieved by CCWD can be arbitrarily close to the optimal value with an increase in the length of queues that is linear in V. The data arrival rate $A_m(t)$ is set to be 0.4 packets per time slot. Since the utility function is concave and nondecreasing, the optimal value of throughput utility should be $N_{\text{session}} \cdot \log(1 + A_m(t))$, where N_{session} is the number of sessions. N_{session} is 4 in this section. In this simulation, optimal throughput utility should be 1.34. Figure 3 shows that the utility value is increased with an increasing V. According to (44), it is easy to calculate $Y^{(m),\max}$, $Q^{(m),\max}$, and $Z^{(m),\max}$. In this section, since the maximum arrival rate $A_{\max}^{(m)}$ of each session is set to be 2 and the throughput utility function of each session is uniform, $Y_{\max}$, $Q_{\max}$, and $Z_{\max}$ can also be calculated using (44). In Figures 4, 5, and 6, V is increased from 500 to 2000 and on a log base 10 scale. From Figures 4, 5, and 6, we can learn that the time average sizes of Q, Y, and Z all increase approximately proportionally with the increase of V and are not larger than the bounds given in Theorem 2. The simulation results show a match between the simulations and the theoretical analyses.

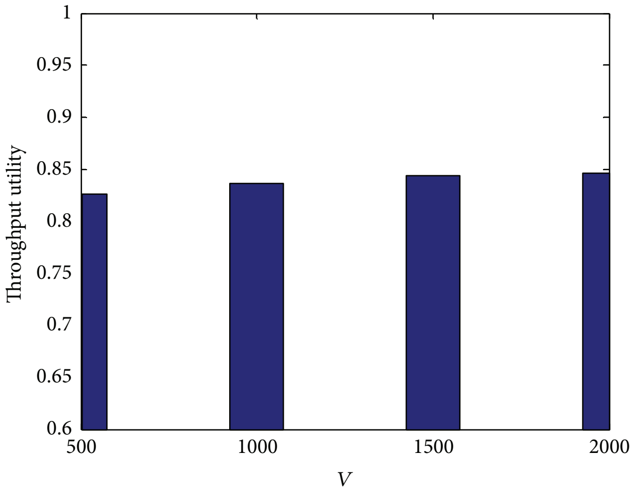


Figure 3: Throughput utility versus V.

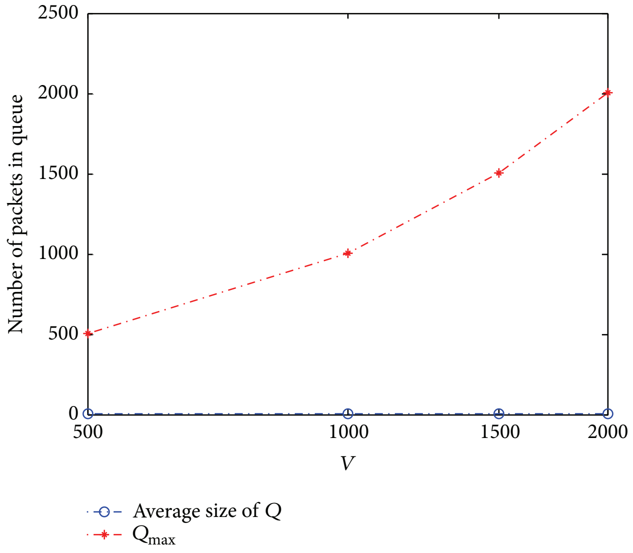


Figure 4: Time average size of Q versus V.

6. Conclusions

This paper proposed a two-phase throughput utility maximization algorithm which provides worst case delay guarantees using a new ϵ-persistent virtual queue for multihop

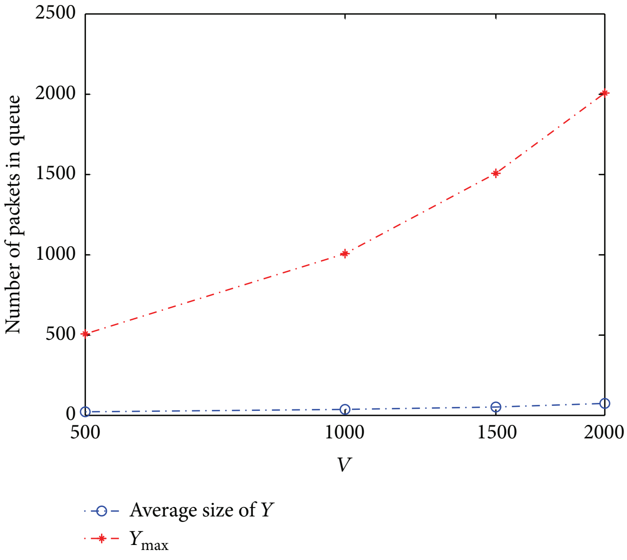


FIGURE 5: Time average size of Y versus V.

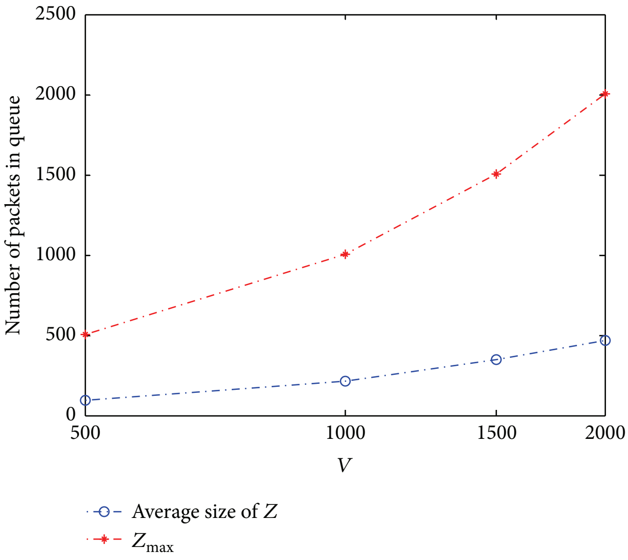


FIGURE 6: Time average size of Z versus V.

wireless networks. Throughput utility optimality of the algorithm is demonstrated with rigorous theoretical analyses. This algorithm ensures that queues are bounded by constants. Compared with existing works, the algorithm presented in this paper performs better in terms of utility and data dropped. The plan for the future work is to combine this proposed algorithm with applications requiring limited delay.

Competing Interests

The authors declare that they have no competing interests.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grant no. 2013CB329003, the National Natural Science Foundation of China under Grant no. 61307016, and Natural Science Foundation of Liaoning Province under Grant no. 2014020193.

References

[1] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Transactions on Automatic Control*, vol. 37, no. 12, pp. 1936–1948, 1992.

[2] M. J. Neely, E. Modiano, and C.-P. Li, "Fairness and optimal stochastic control for heterogeneous networks," *IEEE/ACM Transactions on Networking*, vol. 16, no. 2, pp. 396–409, 2008.

[3] H. X. Li, C. Wu, Z. P. Li, and F. C. M. Lau, "Socially-optimal multi-hop secondary communication under arbitrary primary user mechanisms," in *Proceedings of the IEEE INFOCOM*, pp. 1717–1725, IEEE, Turin, Italy, April 2013.

[4] Z. Liu, X. Y. Yang, P. Zhao, and W. Yu, "Energy-balanced backpressure routing for stochastic energy harvesting WSNs," in *Proceedings of the 10th International Conference on Wireless Algorithm, Systems, and Applications*, pp. 767–777, August 2013.

[5] H. Li, W. Huang, C. Wu, Z. Li, and F. C. M. Lau, "Utility-maximizing data dissemination in socially selfish cognitive radio networks," in *Proceedings of the 8th IEEE International Conference on Mobile Ad-Hoc and Sensor Systems (MASS '11)*, pp. 212–221, Valencia, Spain, October 2011.

[6] L. Jiang and J. Walrand, "A distributed CSMA algorithm for throughput and utility maximization in wireless networks," *IEEE/ACM Transactions on Networking*, vol. 18, no. 3, pp. 960–972, 2010.

[7] H. Seferoglu and E. Modiano, "TCP-aware backpressure routing and scheduling," in *Proceedings of the IEEE Information Theory and Applications Workshop (ITA '14)*, pp. 1–9, San Diego, Calif, USA, February 2014.

[8] E. Anifantis, E. Stai, V. Karyotis, and S. Papavassiliou, "Exploiting social features for improving cognitive radio infrastructures and social services via combined MRF and back pressure cross-layer resource allocation," *Computational Social Networks*, vol. 1, article 4, 2014.

[9] M. J. Neely, "Delay analysis for maximal scheduling with flow control in wireless networks with bursty traffic," *IEEE/ACM Transactions on Networking*, vol. 17, no. 4, pp. 1146–1159, 2009.

[10] L. B. Le, K. Jagannathan, and E. Modiano, "Delay analysis of maximum weight scheduling in wireless ad hoc networks," in *Proceedings of the 43rd Annual Conference on Information Sciences and Systems (CISS '09)*, pp. 389–394, Baltimore, Md, USA, March 2009.

[11] D. Xue and E. Ekici, "Delay-guaranteed cross-layer scheduling in multihop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 21, no. 6, pp. 1696–1707, 2013.

[12] Z. Jiao, B. Zhang, W. Gong, and H. Mouftah, "A virtual queue-based back-pressure scheduling algorithm for wireless sensor networks," *Eurasip Journal on Wireless Communications and Networking*, vol. 2015, article 35, 2015.

[13] H. Z. Xiong, R. Li, A. Eryilmaz, and E. Ekici, "Delay-aware cross-layer design for network utility maximization in multi-hop networks," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 5, pp. 951–959, 2011.

[14] L. Ying, S. Shakkottai, and A. Reddy, "On combining shortest-path and back-pressure routing over multihop wireless networks," in *Proceedings of the 28th Conference on Computer Communications (IEEE INFOCOM '09)*, pp. 1674–1682, Rio de Janeiro, Brazil, April 2009.

[15] M. J. Neely, "Delay-based network utility maximization," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '10)*, San Diego, Calif, USA, March 2010.

[16] M. J. Neely, "Opportunistic scheduling with worst case delay guarantees in single and multi-hop networks," in *Proceedings of the IEEE INFOCOM*, pp. 1728–1736, IEEE, Shanghai, China, April 2011.

[17] M. J. Neely, *Stochastic Network Optimization with Application to Communication and Queueing Systems*, Morgan & Claypool Publishers, 2010.

[18] A. Cammarano, F. L. Presti, G. Maselli, L. Pescosolido, and C. Petrioli, "Throughput-optimal cross-layer design for cognitive radio ad hoc networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 26, no. 9, pp. 2599–2609, 2015.

[19] L. Georgiadis, M. J. Neely, and L. Tassiulas, "Resource allocation and cross-layer control in wireless networks," *Foundations and Trends in Networking*, vol. 1, no. 1, pp. 1–144, 2006.

[20] Z. Jiao, Z. Yao, B. Zhang, and C. Li, "NBP: an efficient network-coding based backpressure algorithm," in *Proceedings of the IEEE International Conference on Communications (ICC '13)*, pp. 1625–1629, Budapest, Hungary, June 2013.

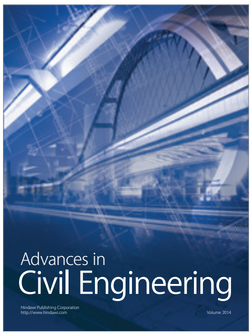
Journal of
Engineering

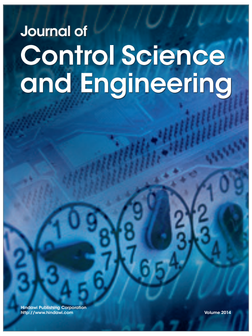
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

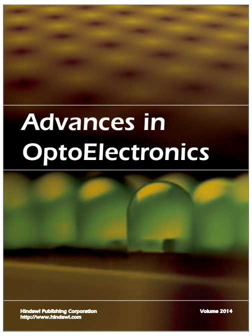
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

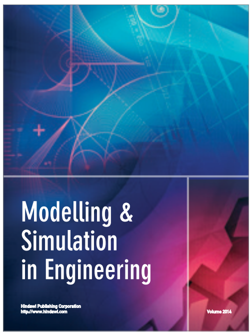
Journal of
Robotics

Journal of
Electrical and Computer Engineering

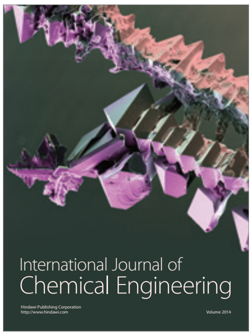

Advances in
OptoElectronics

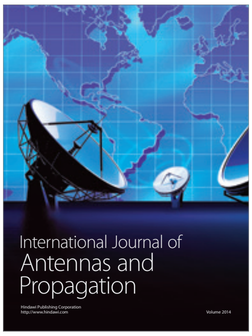
VLSI Design

International Journal of
Navigation and Observation

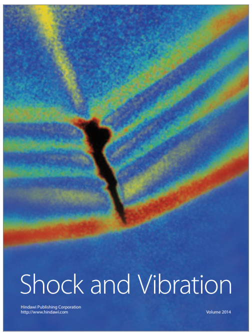

Modelling &
Simulation
in Engineering

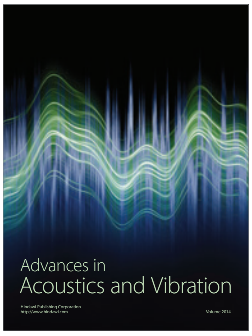
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive
Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration