Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2013, Article ID 376028, 10 pages
http://dx.doi.org/10.1155/2013/376028

Research Article

Throughput-Optimal Scheduling for Cooperative Communications in Wireless Ad Hoc Networks

Thong Huynh,[1] Won-Joo Hwang,[1] and Suk-Hwan Lee[2]

[1] *Department of Information and Communications Engineering, Inje University, Kimhae 621-749, Republic of Korea*
[2] *Department of Information Security, Tongmyong University, Busan 608-711, Republic of Korea*

Correspondence should be addressed to Won-Joo Hwang; ichwang@inje.ac.kr

Received 21 March 2013; Revised 10 July 2013; Accepted 6 August 2013

Academic Editor: Hongli Xu

Cooperative communications that take advantage of the broadcasting nature of wireless environments can help to increase system throughput in wireless ad hoc networks. However, the promised throughput might be lost in presence of the error transmission and the interference caused by relay transmission in cooperative communication. In this paper, we introduced the relay selection schemes that can control the interference at the relay to prevent the relay that may harm other pairs. Then, we proposed the throughput-optimal scheduling that takes into account error probability in decision and maximizes throughput, that is, the amount of packets transmitted without error in network. In addition, we derive a simple and lightweight framework to implement the proposed policy in distributed manner. The simulation results show that the proposed scheduling policy outperforms the noncooperative policy and multihop relay policy.

1. Introduction

Cooperative communications that can take advantage of the broadcasting nature of wireless environments have shown excellent performance in both theoretical aspects and implementations; for example, per-node throughput of the cooperative network is a constant factor, while per-node throughput of the conventional wireless network decreases when node density increases [1]. The basic concept of the cooperative communication comes from the design of multiple-input multiple-output (MIMO) system. In MIMO system, by using the multiple antennas at the transmitter and receiver, the system throughput and reliability were improved significantly [2]. However, in wireless networks, for example, sensor networks and ad hoc networks, due to the space limitation and the implementation cost, it is not reasonable to equip each wireless device with multiple antennas. To overcome this restriction, cooperative communication is proposed as a solution to implement the MIMO technique in the form of single antenna node in wireless networks. Many cooperative communication schemes have been proposed in the general framework of wireless ad hoc network [3, 4] in order to improve the performance of the system.

Scheduling policies are employed to determine a set of active users at a given time regarding the interference constraints between links. Throughput-optimal scheduling policy that provides the highest data rate transmission is desirable in wireless networks. It has been shown that cooperative communication at the network layer can help to increase the throughput of the network [5]. However, in wireless ad hoc networks, the throughput gained by cooperative communication might be lost in the presence of the interference caused by relay transmission. Furthermore, since the wireless links are unreliable and the transmission states are only known by the acknowledgments from receivers, the scheduling decision must consider random packet loss and packet retransmission. Hence, this necessitates a new design for throughput-optimal scheduling for the cooperative communication in ad hoc networks.

In this paper, we focus on designing a throughput-optimal scheduling for the cooperative communication at the network layer. We consider the cooperation between the number of source-destination pairs and relay nodes in wireless ad hoc networks. Each pair can communicate either through direct transmission or cooperative transmission which uses other relay to forward its data to destination.

These relays are selected under the constraint of the interference of relay below a certain threshold to prevent the relays that may harm other pairs. Otherwise, in our scenarios, each source node has a single queue, where packets are backlogged before transmitting to destination. Due to the network stability constraint, the scheduling policy also needs to minimize the sum square of the queue lengths so as to equalize and keep the queue bounded.

The primary contributions of this study are as follows.

(i) We first introduce the relay selection scheme which considers interference from relay node by keeping the interference level below a certain threshold. The effect of the interference is controlled by an interference threshold ε. The simulation results show that the effectively selected relay can help to mitigate interference at the relay and improve the network throughput.

(ii) We propose a scheduling policy to determine which pair is allowed to transmit and which transmission mode is used to transmit data at each time. This scheduling is derived from the suboptimal control principles. At each time slot, the scheduling policy takes an action that maximizes a weighted cost function. This cost function is defined regarding parallel objective of the control strategies: maximization of the throughput, and stabilization of network queues.

(iii) We analyze the sufficient condition for throughput-optimal policy and stability in cooperative communication. A theoretical analysis using the Lyapunov function shows that this scheduling policy obtains the throughput to be near optimal while the simulation results show that it outperforms the noncooperative policy and multihop relay policy.

2. Related Work

There are several works concerning the optimal use of the cooperative communication in wireless ad hoc networks. These works mostly focus on information theory study [6, 7] or design and analysis for cooperation at physical layers with amplify-and-forward, decode-and-forward, selection relaying, and incremental relaying [8]. In [9, 10], the authors focus on the design of optimal algorithms and solve the optimal power allocation between the source and relay nodes to maximize capacity [9] and minimize outage probability [10]. In particular, the research in [11] studied the dynamic resource allocation for delay limited cooperative communication networks and developed dynamic cooperation strategies to achieve a target outage probability. They first consider the optimization problem with objective minimizing the number of packet errors. Then by using the Lyapunov technique, they present a framework to achieve the optimal solution of this problem at each time slot. This approach is similar to that of our study except that our work is performed at network layer. However, the objects of this work are different from the goal of our research, for example, throughput-optimal.

Research on throughput-optimal scheduling for single-hop and multihop wireless networks is addressed in [12, 13]. These results have shown that a scheduling policy based on the current queue length and link rate can achieve the maximum throughput region and stability region. In [12], the backpressure scheduling policy, which is based on the difference of queue length between source and destination nodes of links, can achieve the maximum throughput region and guarantee the stability of the network. The works in [13] extend the scheduling policy to a general framework for scheduling and flow control in multihop wireless networks. Recently, these results have been extended to cooperative networks. In [14], the authors provide an analysis of the throughput region and delay of cooperation in a line-like topology. They focus on characterizing the stability region and the delay of network-level cooperation. However, they assume a line-like topology and TDMA scheduling, whereas our research finds a throughput-optimal scheduling oncooperative ad hoc networks.

3. Problem Formulation

3.1. System Model. We consider a wireless ad hoc network with N source destination pairs and K other nodes which have no own data to transmit. These nodes are installed to increase throughput and reliability of the network. They receive and forward packet to the destination. We assumed a slotted time frame structure that is widely used in implementation of practical system, for example, WiMax, and theoretic analyses [9, 11]. The time is divided into equal units $t = \{0, 1, 2, \ldots\}$ called time slot.

The exogenous traffic stream arrives to each source node at each time slot t according to independent stochastic process. Then, it is stored in separate queues to await transmission. Let $a(t) = \{a_k(t)\}_{k\in N}$ represent the exogenous traffic arriving at the time slot t with the mean $\lambda_c = E\{a_k(t)\}$ and $a_k(t) \le A_{\max}$. The arrival processes are i.i.d over slots, and the second moment $E\{a_k^2(t)\}$ of the arrival process is finite. Let $Q_k(t)$ be the number of packets waiting at queue of source node k at time t.

The wireless channel is considered to be flat fading and all noise components are modeled as additive white Gaussian noise (AWGN) with zero means and variance N_0. We focus on the slow fading channel, where the channel states of all links remain constant during one time slot and might be changed in the next time slot. Moreover, the channel gain between node i and node j is denoted by $h_{i,j}$. Assume that the transmission power at each node is fixed with P. Let $\gamma_{i,j}$ be the instantaneous signal-to-noise ratio (SNR) when node i transmits packet to node j. Thus, the instantaneous SNR $\gamma_{i,j}$ is $\gamma_{i,j} = (P/N_0)|h_{i,j}|^2$, which is modeled as an exponential distribution with means $\delta_{i,r}^2 = (P/N_0)E[|h_{i,j}|^2]$ [15].

In our model, each pair in the network has freedom in selecting the transmission mode at each time slot. They can communicate either through direct transmission or cooperative transmission. In the first case, we say that the pair is in the direct mode, and in the latter case we say that the pair is in the cooperative mode. If the pair is in direct mode, the packet is transmitted directly to the destination by the source node. If the pair is in cooperative mode, the packet is forwarded to the destination by employing another relay node.

3.2. Relay Selection Scheme with Interference Control. In cooperative mode, the transmission of the selected relay may create unacceptable interference for other pairs. To prevent this problem, we proposed that each selected relay must maintain interference constraint, where the interference at each relay must be below a predetermined value γ_{th}. However, since the channel gain at each relay node is changed per time slot in time-varying channel, it is not reasonable to guarantee this constraint in every time slot. Thus, we require the probability of satisfying the interference constraint for selected relay of each pair i above a certain value p_{th}:

$$\Pr\left(\frac{\gamma_{i,r}}{\gamma_{\text{int},r}+1} < \gamma_{th}\right) > p_{th}, \tag{1}$$

where $\gamma_{i,r}$ is the received SNR of the link between source node i and its relay r and $\gamma_{\text{int},r}$ is the interference at the relay node which is caused by multiple transmission of other pairs. Note that, if other pairs keep silent, $\gamma_{\text{int},r} = 0$. In this case, the interference constraint is given by

$$\Pr\left(\gamma_{i,r} < \gamma_{th}\right) = 1 - \exp\left(-\frac{\gamma_{th}}{\delta_{i,r}^2}\right) > p_{th}. \tag{2}$$

Thus, if $1 - \exp(-\gamma_{th}/\delta_{i,r}^2) < p_{th}$, the constraint is not satisfied and the relay is prohibited. Otherwise, the relay will be opportunistically selected. In this case, it is hard to calculate the value of $\gamma_{\text{int},r}$ directly. To overcome this problem, instead of studying the constraint in (1), we study the approximation to the constraint given by

$$(1-\varepsilon)\Pr\left(\gamma_{i,r} < \gamma_{th}\right) > p_{th}. \tag{3}$$

Here, parameter ε indicates the interference control level: if $\varepsilon \rightarrow 1$, the threshold is tight that only few relays are selected to forward the data; otherwise, if $\varepsilon \rightarrow 0$, the threshold is rather loose; most of the relays are selected to forward without restriction. Finally, we obtained the straightforward expression of a set of relays satisfying the interference constraint for each pair i:

$$\mathbf{Re}_i = \left\{r \mid (1-\varepsilon)\Pr\left(\gamma_{i,r} < \gamma_{th}\right) > p_{th}\right\}, \tag{4}$$

or

$$\mathbf{Re}_i = \left\{r \mid \delta_{i,r}^2 \geq \frac{1}{\gamma_{th}}\ln\left(\frac{1-\varepsilon}{1-\varepsilon-p_{th}}\right)\right\}. \tag{5}$$

The average channel gain between relay and source node is determined by $E[|h_{i,r}|^2] = (d_{i,r}/d_0)^{-3}$ [16], in which $d_{i,r}$ is the distance between source node i and relay r and d_0 is the unit distance. Therefore, the relay set for each pair i is circular with radius Re_i as follows: $\mathbf{Re}_i = \{r \mid d_{i,r} \leq \mathrm{Re}_i\}$, where $\mathrm{Re}_i = d_0(N_0\gamma_{th}/P\ln((1-\varepsilon)/(1-\varepsilon-p_{th})))^{1/3}$.

After the set of relays is determined, the source node can opportunistically select from this set one relay to forward the message without considering the effect on other pairs.

3.3. Network Objective and Control Variables. Whenever a pair is communicated, it will receive a certain throughput, which is defined by the number of the successful transmitted packets in one time slot. Let $p_k^d(t), p_k^c(t)$ be the success transmission probabilities over direct link and cooperative link at time t. We use the superscripts d, c to indicate the direct transmission and cooperative transmission in this paper. The success probabilities depend on the channel conditions, modulation, coding type, and power level, as shown in [17]. Using the pilot signal, these probabilities can be estimated at the beginning of each time slot. We assume that these probabilities evolve over slots according to a finite-state Markov chain, and thus the steady state is measurable and is denoted by p^d, p^c. Let $w_k^d(t), w_k^c(t)$ be the number of transmitted packets of pair k in direct mode and cooperative mode at time t. These values can be calculated based on the transmission rate and packet length. Notice that, since different transmission modes have different success probability and number of transmitted packets, the throughput attained by using a different mode will be different. Let $R_k(t) = [R_k^d(t), R_k^c(t)]$ represent the throughput of pair k:

$$R_k^d(t) = w_k^d(t)\,p_k^d(t), \qquad R_k^c(t) = w_k^c(t)\,p_k^c(t). \tag{6}$$

At any time slot t, the source node k can use one of the following transmission modes: direct transmission, cooperative transmission, and no transmission. Define $I_k(t) = [I_k^d(t), I_k^c(t)]$ as the control variable indicating the transmission mode of source node k at time slot t. That is, $I_k(t) = [1,0]$ if node k activated the direct mode, $I_k(t) = [0,1]$ if node k activated the cooperative mode to transmit data, and $I_k(t) = [0,0]$ in other cases. Notice that $I_k(t)$ represents a virtual transmission packet because it describes the possibility of choosing transmission mode independent of the current queue length. In the case when $I_k(t) = [1,0]$ or $[0,1]$ but there is no packet available at the pair k's queue, then no packet is actually transmitted. The vector $\vec{I}(t) = [I_1(t), I_2(t), \ldots, I_N(t)]$ denotes the control vector at time slot t. Let $\Gamma = \{\vec{I}^1(t), \vec{I}^2(t), \ldots, \vec{I}^m(t)\}$ denote the set of all valid control vectors of the network. Clearly, $m \leq 2^{2N}$. In each slot $t > 1$, the scheduling policy Δ decides the control vector $\vec{I}^{\Delta}(t)$ such that $\vec{I}^{\Delta}(t) \in \Gamma$. Now, we present some definitions.

Definition 1 (*network stability*). Queue Q is stable if the time average of the queue length $\lim_{T->\infty}(1/T)\sum_{t=0}^{T-1} E\{Q(t)\} < \infty$. The network is stable if all queues are stable.

Definition 2 (*stability region*). The stability region of a scheduling policy is the set of arrival rates $\{\lambda_m\}_{m\in\{1,\ldots,N\}}$ that stabilize the system under the policy. The stability region of the system Ω is the union of stability region of all scheduling policies.

Definition 3 (*throughput*). For the arrival rate λ, the throughput under the scheduling policy Δ at time slot t, $G^{\Delta}(\lambda)$, is the total successful transmitted packet that is received by all users in the network. Mathematically,

$$G^{\Delta}(\lambda) = \lim_{t\to\infty} \frac{1}{T} \sum_{t=1}^{T} \sum_{k=1}^{N} R_k(t) I_k^{\Delta}(t). \tag{7}$$

Definition 4 (*throughput-optimal*). The scheduling policy is called throughput-optimal for the arrival rate vector $\vec{\lambda} \in \Omega$ if it can obtain the maximum throughput among all scheduling policies that stabilizes the network. We represent the throughput obtained by such policies for arrival rate vector $\vec{\lambda}$ by $G_{\max}(\vec{\lambda})$.

Definition 5 (*ϵ-throughput-optimal*). The scheduling policy is called ϵ-throughput optimal for $\epsilon > 0$ if under this policy the network is stable and $G^{\Delta}(\vec{\lambda}) > G_{\max}(\vec{\lambda}) - \epsilon$. The ϵ-throughput-optimal policy attains a throughput which is at most ϵ less than the throughput of any scheduling.

In this paper, we aim to design the optimal policy that selects the control vector $\vec{I}^i(t) \in \Gamma$ in each time slot to maximize the average throughput, while the stability of the network is guaranteed

$$\max_{\vec{I}^i(t)\in\Gamma} \sum_{k=1}^{N} R_k(t) I_k^i(t) \tag{8}$$

subject to

$$\lim_{t\to\infty} \frac{1}{t} \sum_{k=1}^{t} \sum_{i=1}^{N} Q_i(k) < \infty, \tag{9}$$

where $E(\cdot)$ denote an expectation and I_k^i denote the kth component of control vector $\vec{I}_i$. The constraint in (9) ensures the stability of the network.

4. Throughput-Optimal for Cooperative Communication (TOCC)

4.1. TOCC Policy. We consider a discrete-time linear time-invariant system with dynamics queues, where $\Phi(t) = \{Q_k(t)\}_{c\in N} \geq 0$ are the states and $\vec{I}^k(t) \in \Gamma$ is the control input. Note that only $\vec{I}^k(t)$ is variable here since the scheduling decision is considered in this design. We define a cost function that is comprised of objectives in policy design: maximizing the throughput and stabilizing the queuing. The cost function is defined as follows:

$$J = \lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} E\left\{ \sum_{k=1}^{N} Q_k^2(t) - V \sum_{k=1}^{N} R_k(t) I_k^i(t) \right\}. \tag{10}$$

The first term in function J is the average expectation sum quadratic of the queue length. This quadratic form gives higher cost values to the larger queue length; thus, it tends to equalize the queue length of every pair and reduces queue length of larger queues. The second term of J is the average expectation of total throughput. The minimization of J assures that the queue at each pair is minimized and total throughput is maximized. V is a control parameter that provides tradeoff between the stability of queue and attained throughput.

In general, linear stochastic optimal control problem can be solved effectively with optimal policy in only few special cases [18, 19]. However, there exist different methods to find suboptimal control policy [18]. Regarding computational complexity and its compliance to our objective function, we consider control-Lyapunov feedback method. Suboptimal control policy following the control-Lyapunov feedback method is given as

the TOCC policy:

$$\vec{I}(t) = \arg\max_{\vec{I}\in\Gamma} \left(\sum_{k=1}^{N} \left(Q_k(t) I_k(t) + V R_k(t) I_k(t) \right) \right). \tag{11}$$

The TOCC has a similar form to that of the SNO algorithms [13]; however, the throughput is calculated differently. In TOCC, the throughput takes into account the success probability on each transmission mode and the number of transmitted packets in each mode. The solution of the TOCC uses the value of the current channel states, the queue length $Q(t)$, the success probability, and the achievable rate (or number of transmitted packets) of each mode and does not need a memory of historical states or a required knowledge of the arrival rate.

4.2. Operation of TOCC Policy. In this section, we describe the operation of the TOCC policy: how this policy can stabilize the network while maximizing the throughput, by comparing it with the well-known backpressure policy [12]. We first denote

$$W_{\text{TOCC}}\left(\vec{I}^i\right) = \sum_{k=1}^{N} Q_k(t) I_k^i(t) + V R_k(t) I_k^i(t), \tag{12}$$

$$W_{\text{BP}}\left(\vec{I}^i\right) = \sum_{k=1}^{N} Q_k(t) \cdot I_k^i(t), \tag{13}$$

where $I_k^i(t)$ denote the kth element of vector $\vec{I}^i(t)$. In each time slot t, the TOCC and backpressure policies select the control vector that maximizes $W_{\text{TOCC}}(\vec{I}^i)$ and $W_{\text{BP}}(\vec{I}^i)$. It has been shown in [12] that the backpressure policy stabilizes the network for every arrival rate vector within the stability region. This is because, under this policy, the queue length process $\sum_{k=1}^{N} Q_k^2$ has a negative drift when the total queue length $\sum_{k=1}^{N} Q_k$ is significantly large.

We consider the two cases: large queue length and small queue length. In the first case, when the queue lengths are significantly large, we have $\sum_{k=1}^{N} Q_k(t) \gg V \sum_{k=1}^{N} |R_k(t) I_k|$; then, from (12) and (13) for every $\vec{I}(t)$, we have $W_{\text{TOCC}}(\vec{I}^i) \approx W_{\text{BP}}(\vec{I}^i)$. Thus, TOCC policy and Backpressure policy select similar control vector. It is clear that the queue length process under TOCC policy also has a negative drift when the total queue length is sufficiently large. Therefore, TOCC policy stabilizes network for every arrival rate vector strictly inside the stability region.

In case the queue lengths are small, $W_{\mathrm{TOCC}}(\vec{I}^i)$ and $W_{\mathrm{BP}}(\vec{I}^i)$ differ, and the control vectors which are selected under TOCC policy and backpressure policy are not similar. In this case, we have $Q_k(t) \ll |R_k(t)I_k|$. The TOCC policy prefers to select pair k and control vectors which have high throughput to maximize $W_{\mathrm{TOCC}}(\vec{I}^i)$. Therefore, TOCC policy can guarantee the maximum throughput for every transmission pair.

The following theorem summarizes our discussion above. It is shown that the TOCC is ϵ-throughput-optimal. The TOCC attains a throughput with a gap ϵ to the optimal solution. By controlling parameter V, we can achieve total throughput arbitrary close to the optimal point.

Theorem 6. *Given any arrival rate vector, λ, strictly inside the stability region, the TOCC policy stochastically stabilized all the queue. And then, for every $\epsilon > 0$, there exits V^* such that, for every $V > V^*$, TOCC policy is ϵ-throughput-optimal policy.*

Proof. The proof of Theorem 6 is given in Appendix A. □

4.3. Distributed Implementation Issues. Implementation of the TOCC policy requires two steps: gathering network states information which includes the queue length, success probability, and transmission rate, and computing the optimal control vector. In the first step, it requires centralized implementation and thus leads to increasing delay and bandwidth. In the second step, the control policy TOCC needs to find optimal vector $\vec{I} \in \Gamma$ by solving (11). In the worst case, since the cardinality of Γ can be 2^{2N}, the complexity to compute the optimal control vector $\vec{I}^*$ is also $O(2^{2N})$. Thus, finding the optimal control policy is challenging since it requires the centralized implementation and the complexity of the problem is high.

To address the above challenges, we apply the Randomize Pick and Compare (RPC) framework [20] in order to implement TOCC policy in distributed manner with polynomial computation. The key feature of the RPC framework is that it does not seek to find an optimal control vector in every slot, and hence, it can significantly reduce the computation time. In every time slot, it determines the set $I(t)$ in two steps.

Step 1 (pick). At each time slot, it generates a new control vector $\vec{\bar{I}}(t)$ such that

> $\Pr(W_{\mathrm{TOCC}}(\vec{I}^*(t)) = W_{\mathrm{TOCC}}(\vec{\bar{I}}(t))) > \sigma$, for constant $\sigma > 0$ and $\vec{I}^*(t)$ is the optimal control vector selected by TOCC policy.

Step 2 (compare). Determine the control vector at time slot t by comparing the previous control vector $I(t-1)$ and the new one $\bar{I}(t)$:

$$\vec{I}(t) = \arg\max_{I(t)} \left(W_{\mathrm{TOCC}}\left(\vec{\bar{I}}(t)\right), W_{\mathrm{TOCC}}\left(\vec{I}(t-1)\right) \right). \quad (14)$$

Based on [20, 21], the compare operation can be performed in a distributed fashion with $O(N^3)$ computations. The following theorem shows that RPC framework has the same performance with the TOCC policy.

Theorem 7. *Given any arrival rate vector, λ, strictly inside the stability region, the RPC framework stabilized all the queue. And then, for every $\epsilon > 0$ and $\vec{\lambda} \in \Omega$, there exits V' such that, for every $V > V'$,*

$$G^{\Pi_*}\left(\vec{\lambda}\right) > G_{\max}\left(\vec{\lambda}\right) - \epsilon, \quad (15)$$

where G^{Π_*} is the throughput of the system under RPC framework.

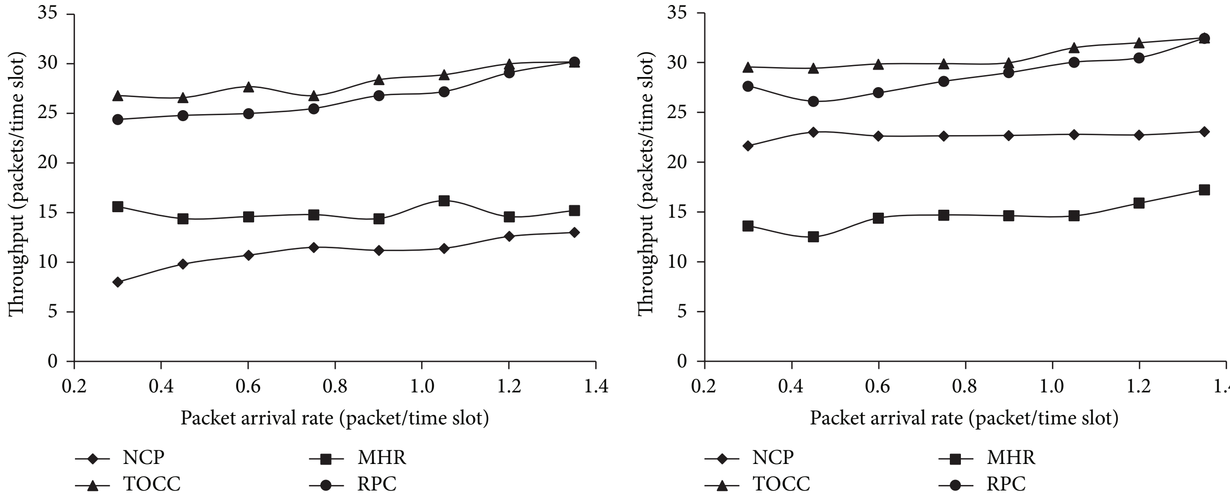
Proof. The proof of Theorem 7 is given in Appendix B. □

5. Performance Evaluation

In this section, we present our simulation to illustrate the theoretical results and to compare the performance of TOCC policy and its implementation with the aforementioned well-known policy.

In our simulation, we generate a random topology with 100 nodes which are uniformly distributed in a rectangular area. We select 20 nodes as the source nodes, and each source node picks one of its neighbor nodes as a destination. A relay node is also picked from the relay set of each pair. In order to determine the relay set of each pair, we let $\gamma_{th} = 1$ and $p_{th} = 0.8$. We set up a dynamic wireless environment where the link qualities and the arrival rate are stationary processes. A 2-hop interference model is used to determine the set of all valid control vectors, Γ. The extended packets arrival to each pair k is assumed to follow the exponential distribution with mean λ_k. These pairs can transmit either directly or cooperatively with a relay node inside their relay set Re_i. At each time slot, our simulation runs the following steps. At the first step, the success probabilities are generated using a random process with given mean quality. At the second step, we find control vector $\vec{I}$ according to the TOCC policy. At the last step, queuing evolution is updated.

5.1. Throughput Comparison. In the first experiment, we compare our performance with two other policies called noncooperative policy (NCP) and multihop relay policy (MHR) which always uses relay node to forward its data. We evaluate the performance of these policies in the term of the throughput. The control parameter V is set to 10. The relay set for each pair is calculated with the interference control level $\varepsilon = 0.8$. The number of source nodes is 20 nodes. Figure 1(a) shows the throughput under the scenario where the direct links between source and destination have very poor quality, $p^d = 0.4$. We can see that the throughput of these policies is decreased in order: TOCC, RPC, MHR, and NCP. Although the multihop relay policy provides better performance than the noncooperative policy in this case, since it does not use the direct link, its performance is far below that of the TOCC and RPC. Similar results are shown in Figure 1(b)



(a) Throughput and arrival rate when average $p^d = 0.4$

(b) Throughput and arrival rate when average $p^d = 0.8$

Figure 1: Throughput and arrival rate under different direct link quality.

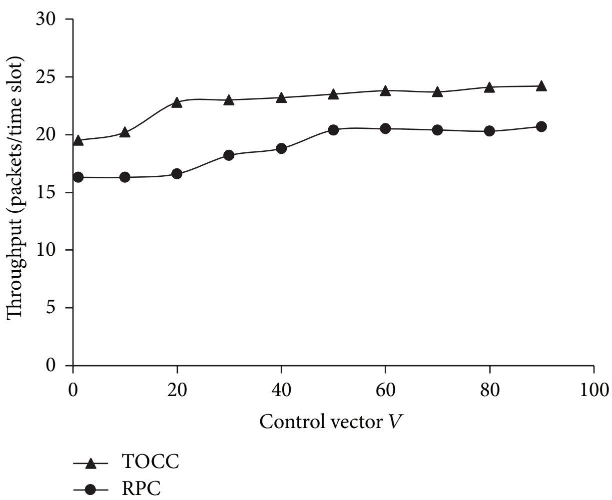


Figure 2: Impact of control parameter V on throughput.

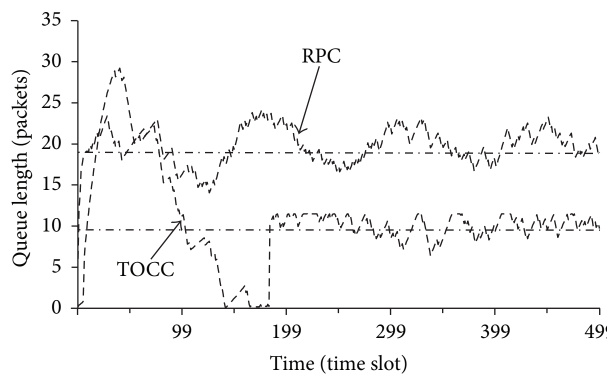
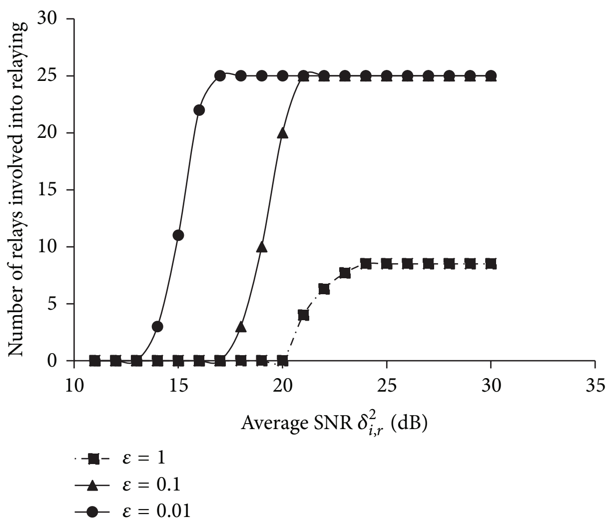


Figure 3: Evolution of the queues under the TOCC and RPC policies.

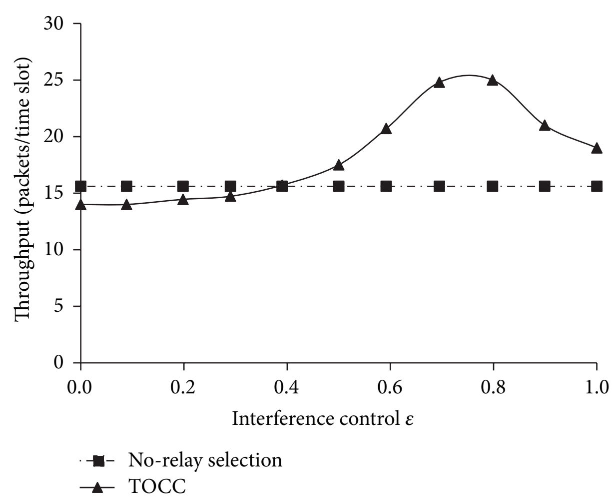
on the contrary case where the direct links between source and destination have good quality, $p^d = 0.8$. These results have two interesting features. First, when the quality of the direct link is good, the noncooperative policy performs better than multihop relay. When the quality of the direct link is low, the two-hop relay performs better. This reflects the nature of the wireless network: the performance of the direct transmission and multihop relay transmission depends on the link qualities. Second, in all cases, TOCC and RPC provide throughput better than what the noncooperative and multihop relays do. The reason behind this result is that the TOCC and RPC have wider degrees of freedom in choosing transmission mode of each pair at every time slot than others. Notice that, contrary to throughput of NCP and MHR policies, the throughput of the TOCC and RPC policies is almost the same in the two cases and its performance does not depend on the link quality of network.

Next, we consider the effect of the parameter V to the performance of throughput. Given an input rate $\lambda_n = 0.6$ and the number of users $k = 20$, simulations were conducted by using control parameters V in the range from 0 to 50, and the results are given in Figure 2. From Figure 2, it is seen that the throughput converges to the optimal throughput level, as the control parameter V is increased. This result has been predicted in Theorems 6 and 7: when $V > V^*$, the throughputs obtained by TOCC and RPC are near optimal.

5.2. Convergence and Network Stability. We verify the stability of the network under TOCC and RPC policies by investigating the queue length over 500 time slots. Figure 3 shows the evolution of the average queue lengths at source nodes. This result shows the queue length property of TOCC and RPC policies, which is specified in Theorems 6 and 7. The queue length fluctuates and coverages at the predefined queue length. Hence, the network is stable under TOCC and RPC policies. Notice that the RPC selects the control vector in an



(a) The number of relays involved into relaying



(b) Throughput under variance of interference control parameter ε

Figure 4: The impact of interference control parameter.

opportunistic way; thus, the convergence of the queue length under TOCC is faster than that of RPC.

5.3. Impact of the Interference Control. We consider the impact of the interference to the selection of the relay. The number of the relays under variation of the parameter ε and the average SNR on the link between source and relay nodes $\delta_{i,r}^2$ are shown in Figure 4(a). The number of the relays is increased with the increase of $\delta_{i,r}^2$. Similarly, the parameter ε also impacts on the permitted number of the relays. This is because small ε means that the interference control is rather loose. Consequently, more relays are permitted to forward message.

The throughput of the TOCC under variance of parameter ε is illustrated in Figure 4(b). The throughput decreases drastically when ε goes to 0. From the interference control effect, it is clear that the throughput is lowered by nearly one and a half times with $\varepsilon = 0$ (control-free case) in contrast to $\varepsilon = 0.7$ case. Furthermore, the parameter ε more than 0.4 is recommended since TOCC can maintain its competitive performance in throughput in comparison with no-relay case.

6. Conclusion

Wireless ad hoc networks with cooperative transmission and direct transmission have been considered in this paper. Aiming at maximizing network throughput, we have investigated the scheduling policy which can guarantee the stability of network and obtain the throughput to be near optimal. In addition, the relay selection has been introduced to deal with interference caused by relay and other pairs. The relay selection scheme considers interference caused by relay transmission, since it required that each relay must maintain an interference constraint. Through simulation, we demonstrate the effectiveness of the proposed TOCC algorithm. The throughput that was obtained by our TOCC policy is very close to the optimum. It is also observed that effectively selecting relay can help to mitigate interference at the relay and improve the network throughput.

Appendices

A. Proof of Theorem 6

First, we prove the supporting lemma.

Supporting Lemmas. Consider the following randomized policy (RD) that finds the probability p_i, with the policy that selects $\vec{I}^i \in \Gamma$ in every time slot by solving the linear programming problem:

$$LP\left(\vec{\lambda}, \delta\right):$$
$$L\left(\vec{\lambda}, \delta\right) = \max_{\vec{p}} \sum_{i=1}^{m} \sum_{k=1}^{N} p_i R_k(t) I_k^i \tag{A.1}$$

subject to

$$\sum_{i=1}^{m} p_i = 1, \tag{A.2}$$

$$\sum_{i=1}^{m} \left| p_i I_{ik} \right| = \lambda_k + \delta, \tag{A.3}$$

where $\delta > 0$ is a parameter to ensure the stability of the network, while each queue is served at the rate higher than the arrival rate. Given any arrival rate vector $\vec{\lambda} \in \Omega$, the RD policy is ϵ-throughput-optimal.

Proof. The constraint (A.3) ensures the stability of the network, while each queue is served at the rate higher than the arrival rate. Then we need to show that

$$G^{RD}\left(\vec{\lambda}\right) > G_{\max}\left(\vec{\lambda}\right) - \epsilon \tag{A.4}$$

for every $\vec{\lambda} \in \Omega$, $\epsilon > 0$. Let $\alpha_i(t) = 1$ if $\vec{I}^i$ is a control vector at time slot t, and $\alpha_i(t) = 0$ otherwise. Then, from (A.1) and (A.3), we have

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\alpha_i(t) = p_i, \tag{A.5}$$

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{m}\alpha_i(t)\, I_k^i(t)\, 1_{\{Q_k>0\}} = \lambda_k, \tag{A.6}$$

$$\lim_{T\to\infty}\frac{1}{T}\sum_{t=1}^{T}\sum_{i=1}^{m}\alpha_i(t)\, I_k^i(t) = \lambda_k + \delta. \tag{A.7}$$

Equation (A.6) comes from the fact that the network is stable; then, the number of data departing the queue is equal to the data arriving at the queue. Now, we have

$$\begin{aligned}
&G^{RD}\left(\vec{\lambda}\right)\\
&= \lim_{T\to\infty}\frac{\sum_{t=1}^{T}\sum_{k=1}^{N}\sum_{i=1}^{m}\alpha_i(t)\, R_k(t)\, I_k^i(t)_{\{Q_k(t)>0\}}}{T}\\
&= \lim_{T\to\infty}\frac{\sum_{t=1}^{T}\sum_{k=1}^{N}\sum_{i=1}^{m}\alpha_i(t)\, R_k(t)\, I_k^i(t)}{T}\\
&\quad - \lim_{T\to\infty}\frac{\sum_{t=1}^{T}\sum_{k=1}^{N}\sum_{i=1}^{m}\alpha_i(t)\, R_k(t)\, I_k^i(t)_{\{Q_k(t)=0\}}}{T}\\
&\geq L\left(\vec{\lambda},\vec{\delta}\right) - \frac{C\sum_{t=1}^{T}\sum_{i=1}^{m}\alpha_i(t)\, I_{k\{Q_k=0\}}^i}{T},
\end{aligned} \tag{A.8}$$

where C is the maximum capacity that a network can support, $C \geq \sum_{k=1}^{N} R_k$. From (A.6), (A.7), and (A.8), it follows that

$$G^{RD}\left(\vec{\lambda}\right) \geq L\left(\vec{\lambda},\delta\right) - C\delta. \tag{A.9}$$

Consider an arbitrary policy Π that stabilizes the network for $\vec{\lambda} \in \Omega$. We denote by $\beta_i(t)$ the cumulative slots that control vector $\vec{I}^i$ is selected under Π until t. Then, we have

$$\begin{aligned}
&\frac{1}{t}\sum_{i=1}^{m}\beta_i(t) = 1,\\
&\lim_{t\to\infty}\frac{1}{t}\sum_{i=1}^{m}\beta_i(t)\, I_k^i = \lambda_k.
\end{aligned} \tag{A.10}$$

Let $\mu_k = \sum_{i=1}^{m}(\beta_i(t)/t) - \lambda_k$ for each k. Then from (A.10), $\beta_i(t)/t$ is in the feasible set of $LP(\vec{\lambda},\vec{\mu})$. Then,

$$\frac{1}{t}\sum_{i=1}^{m}\sum_{k=1}^{N}\beta_i R_k(t)\, I_k^i \leq L\left(\vec{\lambda},\vec{\mu}\right). \tag{A.11}$$

From (A.11) and the continuity of $L(\vec{\lambda},\delta)$, there exits t^* such that for every $t > t^*$ and $\vec{\delta}^* > 0$, we have $-\delta_k^* \leq \mu_k \leq \delta_k^*$. Thus

$$\frac{1}{t}\sum_{i=1}^{m}\sum_{k=1}^{N}\beta_i R_k(t)\, I_k^i \leq L\left(\vec{\lambda},\vec{\mu}\right) \leq \sup_{\{-\delta^*\leq\delta\leq\delta^*\}} L\left(\vec{\lambda},\delta\right). \tag{A.12}$$

Taking the limit of two sides of inequality the previous:

$$G^{\Pi}\left(\vec{\lambda}\right) = \lim_{t\to\infty}\frac{1}{t}\sum_{i=1}^{m}\sum_{k=1}^{N}\beta_i R_k(t)\, I_k^i \leq L\left(\vec{\lambda},0\right), \tag{A.13}$$

where $G^{\Pi}(\vec{\lambda})$ is the throughput obtained by the Π policy. Now, we notice that $\lim_{\delta\to 0}(L(\vec{\lambda},\delta) - \delta C) = L(\vec{\lambda},0)$. Thus, from (A.9), (A.13), and the definition of the previous limitation, there exits δ' such that for every $\epsilon > 0$ and $0 < \delta < \delta'$, it follows that

$$G^{RD} \geq L\left(\vec{\lambda},\delta\right) - \delta C \geq G^{\Pi}\left(\vec{\lambda}\right) - \epsilon. \tag{A.14}$$

Because Π is an arbitrary policy, the result is followed from the previous inequality and Definition 5. □

Proof of Theorem 6. Now, one begins to prove Theorem 6. Let $\vec{I}^{RD}(t)$ and $\vec{I}^*(t)$ be the control vectors at time slot t under RD policy and TOCC policy. Then, G^{RD}, G^* denote the throughput obtained under these policies. Our proof is based on applying the Lyapunov stability techniques. We define the corresponding Lyapunov function as follows:

$$L(Q(t)) = \sum_{i=1}^{N} Q_n^2(t). \tag{A.15}$$

This represents a scalar measure of queue congestion in the network and implies the large queue size. The Lyapunov drift can be represented by $\Delta(Q(t))$:

$$\Delta(Q(t)) = L(Q(t+1)) - L(Q(t)). \tag{A.16}$$

Let $B = N(A_{\max}^2 + 1)$ and using the queue length dynamics,

$$Q_k(t+1) = Q_k(t) + A_k(t) - I_k^*(t). \tag{A.17}$$

The Lyapunov drift can be written as follows:

$$\begin{aligned}
E[\Delta(Q(t))] &= \sum_{k=1}^{N} Q_k^2(t+1) - \sum_{k=1}^{N} Q_k^2(t)\\
&\leq B + 2\sum_{k=1}^{N} Q_k(t)\,\lambda_k\\
&\quad - E\left(\sum_{k=1}^{N} 2Q_k(t)(t)\, I_k^*(t)\right)\\
&= B + 2\sum_{k=1}^{N} Q_k(t)\,\lambda_k\\
&\quad - 2E\left[W_{TOCC}\left(\vec{I}^*(t)\right)\right]\\
&\quad + 2VE\left[\sum_{k=1}^{N} R_k(t)\, I_k^*(t)\right].
\end{aligned} \tag{A.18}$$

Now, for every $\vec{\lambda}$, there exits $\delta > 0$ such that RD is $\epsilon/2$-throughput optimal policy (Supporting Lemma). Hence, from (11), (A.18),

$$\begin{aligned} E(\Delta(Q(t))) \leq{}& B + 2\sum_{k=1}^{N} Q_k(t)\lambda_k \\ &- 2E\left[W_{TOCC}\left(\vec{I}^{\mathrm{RD}}(t)\right)\right] \\ &+ 2VE\left[\sum_{k=1}^{N} R_k(t) I_k^*(t)\right]. \end{aligned} \tag{A.19}$$

Because RD policy selects the control vector $\vec{I}^{\mathrm{RD}}$ with probability $\vec{p}$ independent with queue lengths, thus,

$$E\left(I_k^{\mathrm{RD}}\right) = \sum_{i=1}^{m} p_i I_k^{\mathrm{RD}} = \lambda_k + \delta. \tag{A.20}$$

From (A.19) and (A.20), it follows that

$$\begin{aligned} E(\Delta(Q(t))) \leq{}& B - 2\delta\sum_{k=1}^{N} Q_k(t) \\ &- 2VE\left[\sum_{k=1}^{N} R_k(t) I_k^{\mathrm{RD}}(t)\right] \\ &+ 2VE\left[\sum_{k=1}^{N} R_k(t) I_k^*(t)\right]. \end{aligned} \tag{A.21}$$

(i) Stability of Π_0: from (A.1), (A.21), we have

$$E(\Delta(Q(t))) \leq B - 2\delta\sum_{k=1}^{N} Q_k(t). \tag{A.22}$$

Summing the previous inequalities over $t \in \{1, 2, \ldots T\}$ and noting that $L(Q(0)) = 0$ and $L(Q(T)) > 0$, we have

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{k=1}^{N} Q_k(t) \leq \frac{B}{\delta}. \tag{A.23}$$

Hence, the network is stable under TOCC policy.

(ii) Π_0 is ϵ-throughput-optimal: when all queues converge, we have

$$E(L(Q(t+1))) = E(L(Q(t))), \tag{A.24}$$

$$E\left(I_k^{\mathrm{RD}}\right) = \lambda_k. \tag{A.25}$$

Equation (A.24) comes from the fact that when all queues converge, the arrival rate is equal to the service rate. From the definition of the throughput, it follows that

$$\begin{aligned} E\left[\sum_{k=1}^{N} R_k(t) I_k^{\mathrm{RD}}(t)\right] &= G^{\mathrm{RD}}(\lambda), \\ E\left[\sum_{k=1}^{N} R_k(t) I_k^*(t)\right] &= G^*(\lambda). \end{aligned} \tag{A.26}$$

Thus, from (A.24), (A.25), and (A.26), (A.21) can be written as

$$0 \leq B - 2VG^{\mathrm{RD}}\left(\vec{\lambda}\right) + 2VG^*\left(\vec{\lambda}\right). \tag{A.27}$$

Because RD policy is $\epsilon/2$-throughput-optimal policy, thus,

$$\begin{aligned} G^*\left(\vec{\lambda}\right) &\geq G^{\mathrm{RD}}\left(\vec{\lambda}\right) - \frac{B}{2V} \\ &\geq G_{\max}\left(\vec{\lambda}\right) - \frac{\epsilon}{2} - \frac{B}{2V} \\ &\geq G_{\max}\left(\vec{\lambda}\right) - \epsilon, \end{aligned} \tag{A.28}$$

where $B/V \leq \epsilon$. The result follows if $V > V^* = B/\epsilon$.

□

B. Proof of Theorem 7

Let $\vec{I}^{\Pi_*}(t)$ be the control vector at time slot t under RPC framework. Denote $\{t_k\}_{k=1,2,\ldots}$ as the sequence of time slot at which $W_{\mathrm{TOCC}}(\vec{I}^*(t)) = W_{\mathrm{TOCC}}(\vec{I}^{\Pi_*}(t))$. Thus, we have $E[t_{i+1} - t_i] = 1/\sigma$. Now, with arbitrary $t \in (t_i, t_{i+1})$, we have:

$$\begin{aligned} \sum_{k=1}^{N} Q_k(t) &\leq \sum_{k=1}^{N} Q_k(t-1) + \sum_{k=1}^{N} A_k(t-1) \\ &\leq \sum_{k=1}^{N} Q_k(t_i) + NA_{\max}(t_{i+1} - t_i). \end{aligned} \tag{B.1}$$

From Step 2 of the RPC framework,

$$W_{\mathrm{TOCC}}\left(\vec{I}^{\Pi_*}(t)\right) \geq W_{\mathrm{TOCC}}\left(\vec{I}^*(t_i)\right). \tag{B.2}$$

Now, from (B.1) and (B.2), by rearranging (12), we obtain

$$W_{\mathrm{TOCC}}\left(\vec{I}^{\Pi_*}(t)\right) \geq W_{\mathrm{TOCC}}\left(\vec{I}^*(t)\right) - NA_{\max}\frac{1}{\sigma}. \tag{B.3}$$

Now, given arrival rate vector $\vec{\lambda}$, we use the analysis similar to the proof of Theorem 6 for obtaining (A.18):

$$\begin{aligned} E(\Delta(Q(t))) \leq{}& B + 2\sum_{k=1}^{N} Q_k\lambda_k \\ &- 2E\left[W_{\mathrm{TOCC}}\left(\vec{I}^{\Pi_*}(t)\right)\right] \\ &- 2VE\left[\sum_{k=1}^{N} R_k\left(\vec{I}^{\Pi_*}(t)\right) I_k^{\Pi_*}(t)\right]. \end{aligned} \tag{B.4}$$

Hence, from (B.3) and (B.4),

$$\begin{aligned} E(\Delta(Q(t))) \leq{}& B' + 2\sum_{k=1}^{N} Q_k(t)\lambda_k \\ &- 2E\left[W_{\mathrm{TOCC}}\left(\vec{I}^*(t)\right)\right] \\ &- 2VE\left[\sum_{k=1}^{N} R_k(t) I_k^{\Pi_*}(t)\right], \end{aligned} \tag{B.5}$$

where $B' = B + (NA_{\max}/\sigma)$.

The remaining of the proof is similar to those in the proof of Theorem 6. We omit it here.

Acknowledgment

This work was supported by the 2012 Inje University research grant.

References

[1] A. Özgür, O. Lévêque, and D. N. C. Tse, "Hierarchical cooperation achieves optimal capacity scaling in Ad Hoc networks," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3549–3572, 2007.

[2] M. Shafi, D. Gesbert, D.-S. Shiu, P. J. Smith, and W. H. Tranter, "Guest editorial. MIMO systems and applications: part I," *IEEE Journal on Selected Areas in Communications*, vol. 21, no. 3, pp. 277–280, 2003.

[3] A. Scaglione, D. L. Goeckel, and J. N. Laneman, "Cooperative communications in mobile ad hoc networks," *IEEE Signal Processing Magazine*, vol. 23, no. 5, pp. 18–29, 2006.

[4] A. Nosratinia, T. E. Hunter, and A. Hedayat, "Cooperative communication in wireless networks," *IEEE Communications Magazine*, vol. 42, no. 10, pp. 74–80, 2004.

[5] A. K. Sadek, K. J. R. Liu, and A. Ephremides, "Cognitive multiple access via cooperation: protocol design and performance analysis," *IEEE Transactions on Information Theory*, vol. 53, no. 10, pp. 3677–3696, 2007.

[6] G. Kramer, M. Gastpar, and P. Gupta, "Cooperative strategies and capacity theorems for relay networks," *IEEE Transactions on Information Theory*, vol. 51, no. 9, pp. 3037–3063, 2005.

[7] A. Høst-Madsen, "Capacity bounds for cooperative diversity," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1522–1544, 2006.

[8] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, "Cooperative diversity in wireless networks: efficient protocols and outage behavior," *IEEE Transactions on Information Theory*, vol. 50, no. 12, pp. 3062–3080, 2004.

[9] Y.-W. Hong, W.-J. Huang, F.-H. Chiu, and C.-C. J. Kuo, "Cooperative communications in resource-constrained wireless networks," *IEEE Signal Processing Magazine*, vol. 24, no. 3, pp. 47–57, 2007.

[10] F. S. Tabataba, P. Sadeghi, and M. R. Pakravan, "Outage probability and power allocation of amplify and forward relaying with channel estimation errors," *IEEE Transactions on Wireless Communications*, vol. 10, no. 1, pp. 124–134, 2011.

[11] R. Urgaonkar and M. J. Neely, "Delay-limited cooperative communication with reliability constraints in wireless networks," in *Proceedings of the 28th Conference on Computer Communications, IEEE (INFOCOM '09)*, pp. 2561–2565, Rio de Janeiro, Brazil, April 2009.

[12] 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.

[13] 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.

[14] B. Rong and A. Ephremides, "Protocol-level cooperation in wireless networks: stable throughput and delay analysis," in *Proceedings of the 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '09)*, Seoul, South Korea, June 2009.

[15] J. Si, Z. Li, and Z. Liu, "Outage probability of opportunistic relaying in rayleigh fading channels with multiple interferers," *IEEE Signal Processing Letters*, vol. 17, no. 5, pp. 445–448, 2010.

[16] A. J. Goldsmith, *Wireless Communications*, Cambridge University Press, New York, NY, USA, 2005.

[17] R. C. Daniels and R. W. Heath Jr., "An online learning framework for link adaptation in wireless networks," in *Proceedings of the Information Theory and Applications Workshop (ITA '09)*, pp. 138–140, San Diego, Calif, USA, February 2009.

[18] Y. Wang and S. Boyd, "Performance bounds for linear stochastic control," *Systems and Control Letters*, vol. 58, no. 3, pp. 178–182, 2009.

[19] R. W. H. Sargent, "Optimal control," *Journal of Computational and Applied Mathematics*, vol. 124, no. 1-2, pp. 361–371, 2000.

[20] A. Eryilmaz, A. Ozdaglar, and E. Modiano, "Polynomial complexity algorithms for full utilization of multi-hop wireless networks," in *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM '07)*, pp. 499–507, Anchorage, Alaska, USA, May 2007.

[21] Y. Yi, A. Proutière, and M. Chiang, "Complexity in wireless scheduling: impact and tradeoffs," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc '08)*, pp. 33–42, May 2008.