A Queuing Model-based Incentive Scheme for Optimal Data Transmission in Wireless Networks with Selfish Nodes

Xiaojuan Xie, Hongyi Wu
University of Louisiana at Lafayette
{xxx7680,wu}@cacs.louisiana.edu

Haining Chen
Bethel College, Indiana
chenh@bethelcollege.edu

Abstract

Data transmission in self-organized multi-hop networks heavily depends on the cooperation among nodes. In many applications, however, the autonomous nodes exhibit selfish behaviors, aiming to optimize their own performance without consideration of other nodes in the network. Although a selfish node is interested in transmitting its own data only, part of its resource has to be traded for the cooperation of other nodes in the network, in order to establish a routing path through them to deliver data to its destination. In this paper, we propose a stimulating mechanism to encourage cooperation among the selfish nodes. Specifically, a credit-based Markov chain model is established to analyze the packet dropping probability, with given total bandwidth, bandwidth allocation, buffer space, and the maximum credit of each node. Based on the Markovian model, bandwidth allocation is optimized so that the dropping probability of a node's own packets is minimum. It is a main contribution of this work to address the bandwidth constraint, which is a key resource in wireless networks but has been ignored in all existing incentive schemes. Extensive simulations are carried out to evaluate the proposed incentive scheme, and the simulation results show that it can effectively enable cooperation among selfish nodes and minimize overall packet dropping probability.

1 Introduction

A self-organized multi-hop network consists of a set of autonomous nodes that communicate with each other without the administration of a centralized controller. Each node may serve as a host or a router, or more commonly both. Data transmission in such a network heavily depends on the cooperation among nodes, without which neither routes can be established nor data can be delivered. In many applications, e.g., in a community ad hoc network or wireless mesh network, different nodes have different owners and may freely decide how to make use of their resources such as battery energy and transmission bandwidth. Since relaying data packets for other nodes incurs certain cost, a node may be unwilling to do so given the limited resources it has. In general, there are two types of uncooperative nodes: malicious nodes and selfish nodes. The former are those nodes that intentionally disrupt the network through various attacks. On the other hand, the latter are the nodes that do not perform certain operations in order to preserve their own resources. This paper focuses on a self-organized multi-hop network with selfish nodes that behave rationally. More specifically, each node aims to optimize its own performance only, without wasting its resources to either help or attack other nodes in the network. We also assume that the nodes, though selfish, do not cheat; or alternatively, tamper-proof hardware is adopted to prevent any cheating.

Clearly a mechanism is needed to stimulate the cooperation among selfish nodes. A few stimulation approaches have been proposed in the literature, which can be classified into two categories: the reputation based schemes [1, 5] and the credit (a.k.a virtual currency) based schemes [2, 3, 7, 6, 8, 4, 9]. The reputation based scheme depends on neighbor monitoring in order to dynamically assess the trustworthiness of neighbor nodes and exclude untrustworthy nodes from the network. In the credit based scheme, the sender and/or the receiver needs to pay for the delivery of its data packets. A node receives a certain amount of virtual currency for every packet it relays for other nodes, and the income can be used later to send/receive its own data.

We view the issue of stimulating cooperation as a resource allocation problem. In general, the resources include bandwidth, energy, computing power, etc. In the following discussion, we focus on bandwidth only. But similar idea can be applied to other resources as well. Clearly, a selfish node tends to utilize all of its resource to maximize its benefits. But note that, although it is interested in transmitting its own data only, part of its resource has to be traded for the cooperation of other nodes in the network, in order to establish a routing path through them to deliver data to its destination. In this paper, we propose a incentive scheme that adopts a credit-based queuing analysis approach to en-

978-1-4244-2575-4/08/$20.00

courage cooperation and to find the best deal for each selfish node. In this scheme, we propose a mathematic model, which is employed by each selfish node to continuously analyze its data transmission strategy (i.e., the level of cooperativity) based on a set of up-to-date parameters it obtains, in order to identify the best strategy it will use next. More specifically, we establish a Markov chain, which provides a deep insight into the queuing characteristics of the selfish nodes. Our queuing model takes the bandwidth constraint into consideration, which has been ignored in all studies so far in the literature. Each node can freely allocate a fraction of bandwidth for its own data transmission (which consumes virtual currency) and the remaining to relay traffic of other nodes (which gains virtual currency), with the goal of minimizing dropping probability of its own data packets. Clearly, aggressively allocating as much as possible (or as little as possible) bandwidth for its own traffic does not necessarily achieve this goal. To address this problem, we first analyze the packet dropping probability based on the queuing model, with given total bandwidth, bandwidth allocation, buffer space, and the maximum credit of each node. According to our analytic results, we optimize the bandwidth allocation so that the dropping probability of a node's own packets is minimized. Obviously, the bandwidth allocated to forward other nodes' traffic indicates the degree of cooperativity.

2 A Credit-based Queuing Analysis Approach

In this section, we discuss the incentive scheme, which employs the credit-based mechanism to stimulate cooperations. We first establish a Markov chain model to analyze the packet dropping probability, providing a deep insight into the queuing behavior of the selfish node with limited bandwidth, buffer, and credits. Then, based on our analytic results, we optimize the bandwidth allocation to minimize dropping probability of the self-generated data packets of the selfish node. It is a main contribution of this work to address the bandwidth constraint and optimize bandwidth allocation, which is a key resource in wireless networks but has been ignored in all existing incentive schemes.

2.1 Nodal Model

Since the selfish node is interested in delivering its own data packets only, a credit-based incentive mechanism (similar to [3]) is employed here for stimulating data forwarding. More specifically, each node has a number of nuggets. When the node wants to send one of its own packets, it first estimates the number of intermediate nodes, N, that are needed to reach the destination. If it has at least N nuggets available, the packet is transmitted and the nugget counter is decreased by N. Otherwise, the node cannot send its packet. When a node forwards a packet for the benefit of other nodes, the nugget counter is increased by one. Let C denote the maximum number of nuggets that can be accumulated by a node. Note that this simple mechanism is not necessary to be the best solution. It is adopted here as a basis of our work, to demonstrate the effectiveness of our proposed queuing analysis approach, which can be certainly applied on top of many other incentive mechanisms as well. For example, sophisticated utility and cost functions can be defined to replace the above simple method for nugget management. Additionally, as we have mentioned in the introduction, no cheating or malicious behaviors are assumed in the following discussions.

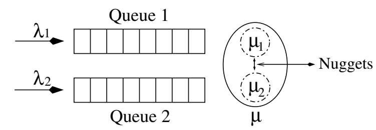


Figure 1. A single node model with two virtual queues.

Without loss of generality, we consider a queuing model as shown in Fig. 1. A node has two (virtual) queues, Queue 1 for the data packets received from all of its neighbors and Queue 2 for the data packets generated by the node itself. We assume that data packets are processed immediately upon arriving at the destination, and thus do not stay in its queue. The maximum lengths of the two queues are M_1 and M_2, respectively. For analytic tractability, we assume that packet arrivals are random and they form a Poisson process at a constant average rate. Let λ_1 and λ_2 denote the average arrival rates for Queue 1 and Queue 2, respectively. Service time is exponentially distributed with an average rate of μ. Generally, μ depends on the available bandwidth, which is in turn affected by the number of neighboring nodes that contend for channel access in the wireless network. For now, we assume that μ is a known constant. The dynamics of μ will be discussed later in this section and investigated in our network simulations to be presented in Sec. 3. Let μ_1 and μ_2 denote the service rates of these two virtual queues, with $\mu_1 + \mu_2 = \mu$. Since μ_1 and μ_2 are allocated separately to the two virtual queues, there are actually two virtual servers, as depicted in Fig. 1. But note that, the two virtual queues are interrelated for earning and consuming the nuggets. Due to the incentive scheme employed here, the actual services of the virtual queues depend on not only the service rates (i.e., μ_1 and μ_2), but also the available nuggets (especially for Queue 2).

Although the node is selfish, it does not maximize its throughput by simply using all available bandwidth for transmitting its own data, because nuggets are required for the transmission of self-generated data packets. Our objective is to optimize the bandwidth allocation for transmitting data packets in Queue 1 and Queue 2 (i.e., to decide optimal μ_1 and μ_2), in order to minimize the dropping rate of the self-generated data packets of the selfish node. Obviously, μ_1 indicates the degree of cooperativity.

2.2 Study of A Simple Case

To facilitate our discussion, we start with a simple case, where $C = M_1 = M_2 = N = 1$. According to the incentive mechanism discussed above, we establish a 3-dimensional Markovian chain model as shown in Fig. 2, where state (i, j, k) indicates that there are i packets in Queue 1, j packets in the Queue 2, and k nuggets available. Denote by $x(i, j, k)$ the steady state probability that the system is at state (i, j, k). Clearly, the state transition diagram describes the data transmission under the incentive mechanism discussed earlier. For example, the transition from $(0, 0, 0)$ to $(1, 0, 0)$ with rate λ_1 represents the reception of a packet from another node; the transition from $(0, 0, 0)$ to $(0, 1, 0)$ with rate λ_2 represents the arrival of a self generated packet; the transition from $(1, 0, 0)$ to $(0, 0, 1)$ with rate μ_1 indicates the transmission of a packet in Queue 1 and the gain of one nugget; and the transition from $(0, 1, 1)$ to $(0, 0, 0)$ with rate μ_2 represents the departure of a self generated packet and the deduction of one nugget. According to the state transition diagram, we can derive the following state equations:

$$
\begin{aligned}
(\lambda_1 + \lambda_2)x(0, 0, 0) &= \mu_2 x(0, 1, 1) && (1)\\
(\lambda_1 + \lambda_2)x(0, 0, 1) &= \mu_1 x(1, 0, 0) && (2)\\
\lambda_1 x(0, 1, 0) &= \lambda_2 x(0, 0, 0) && (3)\\
(\lambda_1 + \mu_2)x(0, 1, 1) &= \lambda_2 x(0, 0, 1) + \mu_1 x(1, 1, 0) && (4)\\
(\lambda_2 + \mu_1)x(1, 0, 0) &= \lambda_1 x(0, 0, 0) + \mu_2 x(1, 1, 1) && (5)\\
\lambda_2 x(1, 0, 1) &= \lambda_1 x(0, 0, 1) && (6)\\
\mu_1 x(1, 1, 0) &= \lambda_1 x(0, 1, 0) + \lambda_2 x(1, 0, 0) && (7)\\
\mu_2 x(1, 1, 1) &= \lambda_1 x(0, 1, 1) + \lambda_2 x(1, 0, 1) && (8)\\
\sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} x(i, j, k) &= 1. && (9)
\end{aligned}
$$

By solving the linear equations, we can also obtain the steady state probability of each state.

Let P_d denote the drop rate of the self generated packets. Since $(0, 1, 0), (0, 1, 1), (1, 1, 0)$, and $(1, 1, 1)$ are the blocking states for Queue 2, we have

$$P_d = x(0, 1, 0) + x(0, 1, 1) + x(1, 1, 0) + x(1, 1, 1) \quad (10)$$

Based on Eq. (10), we can minimize P_d. When $\lambda_1 = \lambda_2 = \hat{\lambda}$, P_d is

$$P_d = \frac{\hat{\lambda}^3}{\Delta}(4\hat{\lambda}^3\mu + 4\hat{\lambda}^2(\mu^2 - \mu_1\mu_2) + 3\hat{\lambda}(\mu_1^2\mu_2 + \mu_1\mu_2^2) + \mu_1^2\mu_2^2). \quad (11)$$

By minimized P_d , we arrive at $\mu_2 = \mu_1 = \frac{\mu}{2}$.

2.3 General Queuing Model

The Markovian model discussed above can be readily extended to the general case with arbitrary M_1,M_2,C, and N. The general Markovian model consists of $M_1 + 1$ planes, each of which has $M_2 + 1$ rows and $C + 1$ columns, summing up to total $(M_1 + 1) \times (M_2 + 1) \times (C + 1)$ states, the state transitions follow several simple patterns as shown below. Let the index for each plane range from 0 to M_1, each row range from 0 to M_2, and each column range from 0 to C. State (i, j, k) is in plane i, row j, and column k. Each state has several transitions to other states. The outgoing transitions from state (i, j, k) are summarized below:

1. For $0 \le i \le M_1 - 1, 0 \le j \le M_2, 0 \le k \le C$, a transition from (i, j, k) to $(i + 1, j, k)$ with a rate of λ_1, which represents the reception of a packet in Queue 1;

2. For $0 \le i \le M_1, 0 \le j \le M_2 - 1, 0 \le k \le C$, a transition from (i, j, k) to $(i, j + 1, k)$ with a rate of λ_2, which represents the arrival of a self generated packet;

3. For $1 \le i \le M_1, 0 \le j \le M_2, 0 \le k \le C - 1$, a transition from (i, j, k) to $(i - 1, j, k + 1)$ with a rate of μ_1, which represents the transmission of a packet in Queue 1 and the gain of one nugget;

4. For $0 \le i \le M_1, 1 \le j \le M_2, N \le k \le C$, a transition from (i, j, k) to $(i, j - 1, k - N)$ with a rate of μ_2, which represents the departure of a self generated packet and the deduction of N nuggets.

The incoming transitions from other states to state (i, j, k) can be derived in a similar way. Once the state diagram is established by following the above rules, the state equations can be readily generated. We solve the linear equations and obtain P_d as a function of μ_1 or μ_2. Then convex optimization is employed to find optimal μ_1 or μ_2 for minimizing P_d.

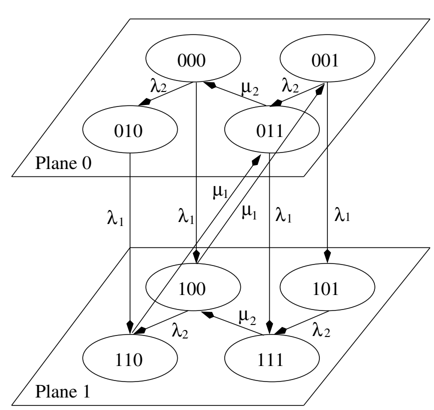
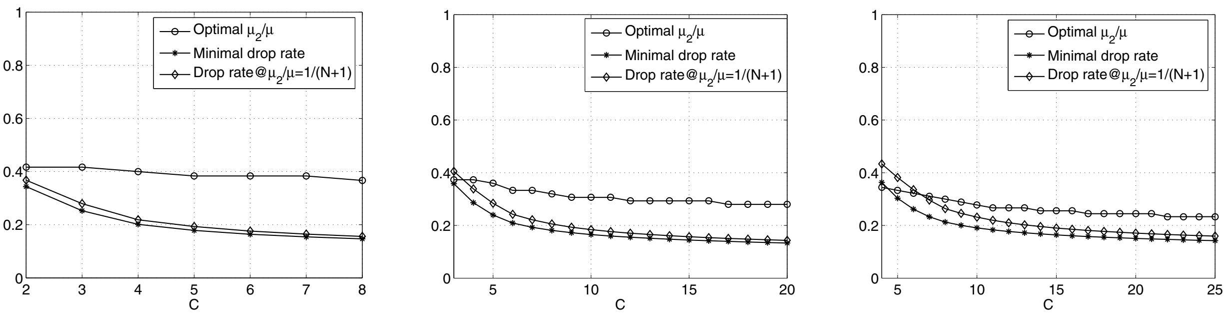


Figure 2. Transition diagram for Markovian chain model.



(a) $N = 2$, $\mu = 12$, and $\lambda_1 = 2\lambda_2 = 6$. (b) $N = 3$, $\mu = 15$, $\lambda_1 = 3\lambda_2 = 9$. (c) $N = 4$, $\mu = 18$, $\lambda_1 = 4\lambda_2 = 12$.

Figure 3. Impact of C on optimal $\frac{\mu_2}{\mu}$, with $M_1 = M_2 = 5$, $\lambda_2 = 3$, $\lambda_1 = N\lambda_2$.

2.4 Numeric Results

In this subsection, we present numeric results obtained through the analytic model discussed above, and study the impact of M_1, M_2, and C on P_d and optimal μ_1 and μ_2. Since $\mu_1 + \mu_2$ equals a constant μ, we focus on optimizing μ_2/μ in the following discussion. For simplicity, we assume $M_1 = M_2 = M$, which is usually true in homogeneous networks. The results are shown in Figs. 3-4.

Figs. 3-4 are under a simulation set up, with N varying from 2 to 4, $\lambda_1 = N\lambda_2$, and $\mu = (N+2)\lambda_2$ (so that μ is always greater than $\lambda_1 + \lambda_2$). We observe the impact of C in Fig. 3 by fixing $M = 5$. Since Fig. 3 shows that the drop rate keeps constant when C is greater than certain values (e.g., $C = 5$, 15, and 25, in Fig. 3 (a), (b) and (c), respectively), such values are used in Fig. 4, when we study the impact of M.

Based on Figs. 3-4 (and other results that are omitted here due to limited space), when M and C are sufficiently large, the dropping rate at $\frac{\mu_2}{\mu} = \frac{1}{N+1}$ is close to the minimal drop rate. This is reasonable because it consumes N nuggets to transmit one self-generated packet. Thus the ratio between μ_1 and μ_2 should be around $N : 1$ in order to achieve optimal, if we ignore the queue dynamics due to random packet arrival and service.

Under small C and/or M, the queuing behavior is more complicated, and the optimal $\frac{\mu_2}{\mu}$ deviates from $\frac{1}{N+1}$ due to the constraints of these two parameters. The larger the N, the bigger the deviation. This is also reasonable as explained below. For example, C, the maximum number of nuggets allowed by a node, limits the incentive of a node for relaying other nodes' data packets. When C is small, a node may quickly accumulates C nuggets, and thus refuses relaying additionally data packets for other nodes. Therefore, it does not need to allocate such high bandwidth as $\frac{N}{N+1}\mu$ to Queue 1. At the same time, it won't have enough nuggets to pay for its own transmission, and thus resulting in higher dropping probability. When C is sufficiently large, the node can make good utilization of the available bandwidth and gain enough nuggets for transmitting its own data packets. Therefore, the optimal $\frac{\mu_2}{\mu}$ converges to $\frac{1}{N+1}$. Similarly, with small M, the the queue is more likely to be built up and overflow. In order to reduce the drop rate of the self-generated data packets, more bandwidth needs to be allocated to Queue 2, thus deviating from $\frac{\mu}{N+1}$. With the increase of M, this problem diminishes.

2.5 Further Discussions

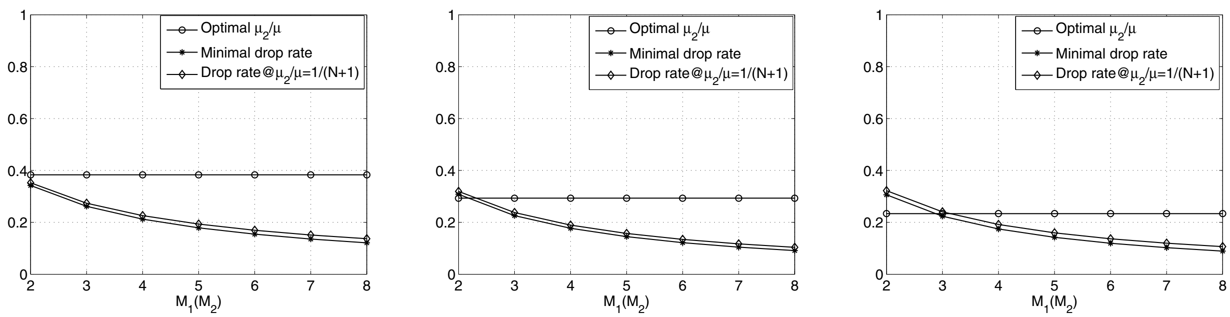
The queuing model discussed above is based on a few known parameters, including λ_1, λ_2, μ, M_1, M_2, C, and N, as inputs. To enable the above queuing analysis in a real network, a node needs to acquire these parameters. Clearly, M_1, M_2, and C are determined by the incentive mechanism. Once a particular stimulation scheme is chosen, they are known by the node. λ_2 is the arrival rate of self-generated traffic, which is also known by the node. The rest three parameters, λ_1, μ, and N are all dynamic and worth elaboration.

λ_1 may dynamically change, depending on several factors, such as the total amount of traffic in the network, the route selection, the data drop rate, etc. The estimation of λ_1 is a typical online estimation problem. Here we apply a sliding window-based linear autoregressive model to calculate the moving average of λ_1. This method is effective and accurate, as shown by the simulation results to be discussed in Sec. 3.

The bandwidth of a node (i.e., μ) in the mobile network is also dynamic. For example, with the change of network topology, μ changes accordingly. Let W denotes the total channel bandwidth of the wireless network interface card and K denote the number of neighbor nodes. Since a node must share the entire bandwidth W with all of its neighbors, we estimate $\mu = \frac{W}{K+1}$.

N depends on the routing path selection as to be discussed in the next section. Once a path is chosen, N is known.

These dynamic parameters are updated in realtime manner, providing up-to-date inputs for the credit-based queuing model, which consequently yields accurate bandwidth



(a) $N = 2$, $\mu = 12$, and $C = 5$. (b) $N = 3$, $\mu = 15$, and $C = 15$. (c) $N = 4$, $\mu = 18$, and $C = 25$.

Figure 4. Impact of $M_1(M_2)$ on optimal $\frac{\mu_2}{\mu}$ with $\lambda_2 = 3$, $\lambda_1 = N\lambda_2$.

allocation of μ_1 and μ_2.

3 Network Simulation

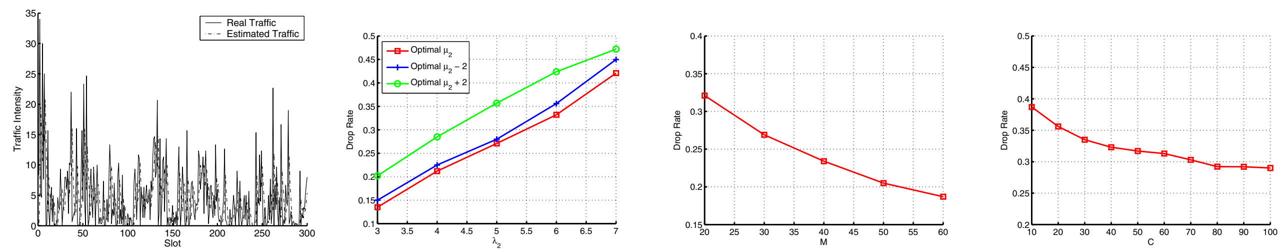
We have implemented the proposed incentive scheme and carried out extensive simulations to evaluate its performance. The default simulation setup is as follows. The network has 30 mobile nodes that are initially placed randomly (uniformly) in a square area of $100m \times 100m$. We divide time into slots. The duration of each time slot is 6 seconds. The nodal mobility model is similar to random waypoint, but the nodes don't move within a given slot. The radio range of every node is $30m$. We assume there is a data link between two nodes if they are within the transmission range of each other. The total bandwidth of the wireless network interface card is $W = 180$ data packets per second, and the size of each packet is $5KB$. At each node, the self-generating data traffic is a Poisson process with data arrival rate of $\lambda_2 = 5$ packets per second. The packet departure of each node is also a Poisson process, and the service rate μ is calculated as $\mu = \frac{W}{K+1}$, where K is the number of neighboring nodes, which varies due to the the nodal mobility. The number of nuggets of each node is initialized to be its maximum, i.e., $C = 80$. The queue length at each node is set to be $M_1 = M_2 = 25$. The simulation results are shown in Fig. 5. All results are the average of multiple simulation runs, and each simulation run lasts for one hour.

The data rate of forwarding traffic (i.e., λ_1) is estimated according to a sliding window approach, with a window size of 3 and the coefficients of 0.5, 0.3, and 0.2. More specifically, $\lambda_1'(n) = 0.5\lambda_1(n-1)+0.3\lambda_1(n-2)+0.2\lambda_1(n-3)$, where n indicates the current time slot, $\lambda_1'(n)$ is the estimated arrive rate in time slot n, while $\lambda_1(n-1)$, $\lambda_1(n-2)$, and $\lambda_1(n-3)$ are the actually measured arrive rates in previous time slots. As can be seen in Fig. 5(a), the estimated forwarding traffic matches the real traffic very well.

Each node randomly chooses the destination for its self-generated data packets. Once the source-destination pair is determined, the source node choose the route with the shortest path according the reputation of the nodes. Due to the limited space here, detail discussion of routing algorithm is omitted here. As a result, the number of intermediate nodes in the chosen route (i.e, N) becomes known by the source node. Then it calculates the optimal bandwidth allocation according to the Markov chain model proposed in Sec. 2. Every node in the network performs the above process. After the bandwidth allocation between self-generating packets and forwarding packets is decided, the node transmits its data packets accordingly.

In our analysis in Sec. 2, we have assumed that the total number of nuggets in the entire network is a constant. N nuggets are deducted from the source and one nugget is gained by each of the N intermediate nodes. With possible packet dropping at the intermediate nodes (e.g., due to queue overflow), however, not every intermediate node can gain the nugget. Thus the total number of nuggets decreases. If we run the simulation for a long period, the number of nuggets owned by each node may drop to zero, and thus no transmission can be initiated. To overcome this problem, we assume that if an intermediate node does not forward the packet successfully, its previous hop can discover this situation and notify the source, which accordingly recycles the nuggets that are not consumed. For example, source A sends packet to destination D along the routing path A→B→C→D. Two nuggets are deducted from A when it sends one packet. If B forwards the packet, it gets one nugget; if C drops the packet, it gets nothing; and the remaining one nugget goes back to A. Note that, this assumption can be valid because we assume the nodes don't cheat, although they are selfish.

Fig. 5(b) shows the network-wide average data drop rate under different traffic load (i.e., λ_2) and different bandwidth allocation schemes at each node. As one may expect, the data drop rate increases with the increase of λ_2. This is reasonable, because the drop rate always increases with the traffic load, under fixed total bandwidth. We also notice that the variation of the service rate μ_2 affects the data drop rate and the lowest drop rate is always achieved when μ_2 equals the optimal value determined by the Markov model



(a) Linear Autoregressive Estimation of forwarding traffic.

(b) Impact of traffic load.

(c) Impact of queue length.

(d) Impact of maximum nuggets.

Figure 5. Network simulation setup and simulation results.

proposed in Sec. 2. This observation testifies the correctness of the Markov model and the effectiveness of the credit-based incentive scheme.

We have also studied the impact of queue length, M. The results are illustrated in Fig. 5(c). When M is small, with the increase of M, there are more buffer space for both self-generated traffic in Queue 1 and forwarding traffic in Queue 2. Thus the probability of dropout due to buffer overflow decreases. When M is sufficiently large, however, further increasing queue size no longer helps, as shown in Fig. 5(c).

The maximum number of nuggets, C, also affects the packet drop rate. Fig. 5(d) shows the results with C ranging from 10 to 100. With the increase of C, the data drop rate decreases. But when C is higher than certain value (e.g., 80 in this figure), the data drop rate keeps constant. This is reasonable because when C is small, there are not enough nuggets circulating in the network, and thus many data packets can not be delivered to the destination. When C is already sufficiently large, data delivery is no longer improved by increasing C, since the consumption and earning of nuggets at a node have reached a dynamic balance, which is not affected by C anymore. Theoretically, we can simply let $C = \infty$, which results in the lowest packet drop rate. But this will lead to high complexity in solving the Markov chain model. So an appropriate small value (e.g., 80 in the above example) is chosen.

4 Conclusion

In this paper, we have proposed a incentive mechanism to stimulate the cooperation among selfish nodes, where a credit-based Markov chain model has been established to analyze the packet dropping probability, with given total bandwidth, bandwidth allocation, buffer space, and the maximum credit of each node. Based on the Markov model, the bandwidth allocation has been optimized so that the dropping probability of a node's own packets is minimum. It is a main contribution of this work to address the bandwidth constraint, which is a key resource in wireless networks but has been ignored in all incentive schemes available in the literatures. Extensive simulations have been carried out to evaluate our proposed incentive scheme. The simulation results have shown that it can effectively enable cooperation among selfish nodes and minimize overall packet dropping probability.

References

[1] S. Buchegger and J. Boudec. Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes-Fairness In Dynamic Ad-hoc NeTworks. In *Proc. of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, pages 226–236, Lausanne, Switzerland, 2002.

[2] L. Buttyan and J. Hubaux. Enforce Service Availability in Mobile Ad-Hoc WANs. In *Proc. of the IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*, pages 87–96, Boston, MA, 2000.

[3] L. Buttyan and J. Hubaux. Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. *ACM/Kluwer Mobile Networks and Applications (MONET)*, 8:579–592, 2003.

[4] S. Eidenbenz and L. Anderegg. Ad hoc-VCG: A Truthful and Cost-Efficient Routing Protocol for Mobile Ad Hoc Networks with Selfish Agents. In *Proc. of ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 245–259, 2003.

[5] Q. He, D. Wu, and P. Khosla. SORI: A Secure and Objective Reputationbased Incentive Scheme for Ad-hoc Networks. In *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6, 2004.

[6] B. Shadpour, S. Valaee, and B. Li. A Self-Organized Approach for Stimulating Cooperation in Mobile Ad hoc Networks. In *Proc. of the Twenty-Second 22nd Biennial Symposium on Communications*, 2004.

[7] V. Srinivasan, P. Nuggehalli, C. Chiasserini, and R. Rao. Energy Efficiency of Ad Hoc Wireless Networks with Selfish User. In *Proc. of European Wireless conference*, Florence, Italy, 2002.

[8] S. Zhong, J. Chen, and Y. Yang. Sprite: A Simple, Cheat-Proof, Credit-Based System for Mobile Ad Networks. In *Proc. of the Twenty-Second Annual Joint Conference of Computer and Communications Societies (INFOCOM)*, volume 3, pages 1987–1997, San Francisco, CA, 2003.

[9] S. Zhong, L. Li, Y. Liu, and Y. Yang. On Designing Incentive-Compatible Routing and Forwarding Protocols in Wireless Ad-Hoc Networks. In *Proc. of ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, pages 117–131, 2005.