Hindawi Publishing Corporation
International Journal of Distributed Sensor Networks
Volume 2012, Article ID 840391, 9 pages
doi:10.1155/2012/840391

Research Article

A Dynamic Pricing Scheme for Congestion Game in Wireless Machine-to-Machine Networks

Zhifei Mao,[1] Guofang Nan,[1] and Minqiang Li[2]

[1] *Institute of System Engineering, Tianjin University, Tianjin 300072, China*
[2] *Department of Information Management and Management Science, Tianjin University, Tianjin 300072, China*

Correspondence should be addressed to Guofang Nan, guofangnan@gmail.com

Received 8 April 2012; Revised 9 July 2012; Accepted 9 July 2012

Academic Editor: Jianhua He

The problem of assigning a set of source nodes to a set of routes in wireless machine-to-machine (M2M) networks is addressed using a game theoretic approach. The objective is to minimize the maximum latency over all source nodes as far as possible while the game achieves a pure Nash Equilibrium (NE). To compute such an NE efficiently, we present a distributed dynamic pricing (DP) scheme, where each source node is assumed to pay for using any route so that the route has incentive to relay data for the source node. A loose upper bound is given for the convergence time of DP, and simulation results show that it performs much faster in practice. The price of anarchy in this game is also investigated by comparing DP with a cost-reducing path method; the results show that DP produces optimum assignment in more than 90% of the simulation runs.

1. Introduction

Features such as self-organization, ease of deployment, low cost, and infrastructurelessness bring to wireless M2M networks the advantages of robustness, easy maintenance, economy, and flexibility. Therefore, M2M networks are envisioned to be the key technology for next generation wireless networks [1]. An M2M network consists of a large number of self-organized and self-configured nodes, the behavior of which cannot be controlled easily without any available centralized scheduling command. Thus, the network can easily be congested when there are a great many nodes that want to transmit data via a common set of routes.

Extensive work has addressed the congestion problem by adjusting the source rate based on a feedback method [2–4], but prevention strategies for the congestion have been ignored. Nevertheless, by joining the congestion control in the routing layer, this problem can be solved via the coordination among nodes [5, 6]. However, in most wireless M2M network applications, each user is its own authority, and unconditional forwarding of packets to other users cannot be assumed directly. Therefore, a scheme that stimulates routes to relay the packets of other nodes is required. Motivated by the general network congestion game [7–9] and pricing scheme [10], our study jointly considers the problems of routing congestion and incentive provision.

In this paper, a distributed dynamic pricing (DP) scheme is developed for wireless M2M networks to minimize the maximum latency over all source nodes. First, routing congestion is formulated as a dynamic game in which pricing is employed in the definition of players' utility function. Each route is encouraged to charge source nodes for delivering their information, and the amount is decided by its latency. Second, dynamics is introduced in the game. The price of each route is changeable rather than fixed so that its load can be adjusted dynamically. We assume that the number of source nodes a route can take has an upper limit, and while the number of source nodes assigned to this route equals its upper limit, we say that this route is supply-demand balanced. Once all routes achieve supply-demand balance, the game terminates and converges to a pure Nash Equilibrium (NE). Finally, simulations are conducted to evaluate the performance of the proposed scheme.

The rest of this paper is organized as follows. Section 2 provides related work in recent years. Section 3 illustrates the system model and Section 4 formulates the network

congestion game. A DP scheme is proposed in Section 5. In Section 6, the simulation results are presented. And we conclude in Section 7.

2. Related Work

In [2], an algorithm jointly addressing congestion control and scheduling in wireless M2M networks is developed, the basis of which is rate allocation among flows in the network. By adjusting backoff min-slot and window size of each flow, the network can achieve high overall throughput and low per-flow delay. However, this algorithm cannot be applied for scenarios with dynamic routing, since route of each flow is assumed to be fixed. EWCCP, a protocol that can be added as a thin layer between IP and TCP, is proposed in [3]. It coordinates flows that compete for common channel according to explicit multibit congestion feedback from routers. Authors in [4] found that the main reason of congestion collapse in wireless M2M networks for streaming services is the severe contention among individual nodes at MAC layer, and thus presented a TCP-friendly congestion control scheme where a contention state estimator is defined as the difference between the number of arriving packets and that of leaving packets during each control interval. Similar to the aforementioned literature, congestion control schemes proposed in [11–13] are all feedback-based. Although these existing feedback-based approaches have achieved some success in congestion control for M2M networks, most of them ignored prevention/avoidance strategies for congestion control.

Congestion avoidance can be achieved by jointly considering congestion and routing. The work in [11] provides a multipath routing algorithm, I2MR, which discovers zone disjoint paths using the concept of path correlation so as to avoid interference in multiple nodes and increase system throughput. Cross-layer design of joint congestion control, routing, and scheduling was presented in [12], which enable each source node to adjust its routes for data transmission in each period according to a local congestion price from its neighbors. Other works such as [14–17] have also jointly considered the problems of congestion control and routing in multihop wireless networks or wireless sensor networks. The major distinction between our work and aforementioned work lies in the formulation of the congestion routing. Specifically, we propose a congestion game to model the route-selection behavior of each node who wants to transmit its data via a route with the lowest price. Moreover, we design a distributed algorithm with low complexity by making use of pricing to implement the congestion game framework.

3. Network Model

Consider a wireless M2M network with a set N of n parallel routes from a set M of m source nodes to a destination node, $M = \{1, 2, \ldots, m\}$ and $N = \{1, 2, \ldots, n\}$. Figure 1 shows an illustration of the network. For each source node i, let the strategy set $S_i \subseteq N$ denote the set of routes to which source node i can possibly be assigned. For each route j, let $B_j \subseteq M$ denote the set of source nodes that can possibly be served by this route. Each source node i intends to send a particular amount of traffic along a fixed route to the destination; thus, their choices form an assignment $\vec{s} = (s_1, s_2, \ldots, s_m)$, where $s_i \in S_i$ is the chosen route of source i. Given an assignment, let q_j be the number of the source nodes choosing route j. Let h_j be the length of route j, which equals the total of nodes contained in this route.

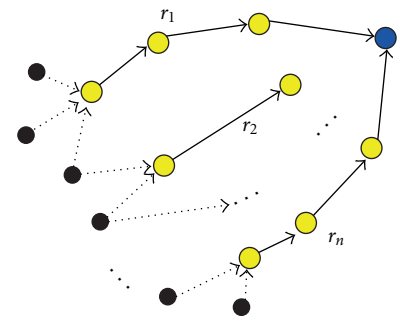

Figure 1: An example of wireless M2M network.

In the network, each route j has a latency d_j. Since the packet of each source node has been generated and processed before its transmission toward one route, the process time will not be included in routes' latency. Furthermore, comparing to queuing time and transmission time, the propagation time can be neglected in most M2M networks except some like satellite networks. Thus, the latency d_j of route j is dependent on the number of its length and current load, which can be expressed as $f(q_j) \oplus g(h_j)$. We assume that all source nodes transmit their packets via a common wireless channel, which means that routes will receive the packets one by one when multiple transmissions arrive simultaneously. Thus, both $f(q_j)$ and $g(h_j)$ can be set as a linear function. Based on the above assumptions, we compute the latency d_j as

$$d_j = q_j \times \frac{L}{R_s} + h_j \times \frac{L}{R_r}, \tag{1}$$

where L is the size of packet generated by source nodes, and R_s is the rate of transmitting one packet from a source node to a route while R_r is the rate of transmitting the same packet between two nodes within the same route. Moreover, we assume that R_s equals R_r. Then, d_j can be rewritten as

$$d_j = \left(q_j + h_j\right) \times D, \tag{2}$$

where $D = L/R_s = L/R_r$.

Our study aims to minimize the maximum latency over all source nodes. Hence, the congestion control problem can be modeled as

$$\min \max_{j \in M} \left(q_j + h_j\right) D. \tag{3}$$

4. Network Congestion Game

The network congestion game deals with the problem of congestion which is fundamental in networks and distributed

systems [7]. Whenever a set of users intend to utilize a much smaller set of resources in a common period, one needs to schedule these users orderly so as to reduce the overall cost and exploit available resources efficiently.

In wireless M2M networks, users and resources are referred to as source nodes and routes, respectively. The network congestion game can be formally defined as a tuple $\Gamma = \langle N, (S_i)_{i\in M}, (u_i)_{i\in M}\rangle$, where M is the set of players, that is, the source nodes. For the remainder of this paper, the terms "source node" and "player" are used interchangeably. S_i is the strategy set of player i. Let $\vec{s} = (s_1, s_2, \ldots, s_m)$ be the strategy profile when each player i chooses s_i, which is equivalent to the assignment in Section 2. $u_i : \Pi = S_1 \times S_2 \times \cdots \times S_m \rightarrow R$ is the utility function of player i. Since each player intends to connect to a route with the lowest latency, the utility function of player i can be formulated as

$$u_i = \{-d_{s_i} \mid s_i \in S_i\}. \tag{4}$$

The problem of incentive provision is managed here by transforming the latency to price which is charged by a route. The source node(s) that chooses this route is the price taker(s). This transformation provides the following benefits: (1) payment can effectively motivate a route to relay the data of a source node, and (2) the route can exploit its price as a lever to adjust its load. The instrument of pricing adopted here is not for maximizing the routes' revenues, but for achieving a socially beneficial objective. The price of route j is set as a function of its latency and denoted as $p_j(d_j)$. The utility function of each player i can then be rewritten as

$$u_i = \{-p_{s_i}(d_{s_i}) \mid s_i \in S_i\}. \tag{5}$$

Suppose the game is played in a one-shot style, which indicates the game experiences only one iteration and then terminates. Since each player has to make a decision without the decisions of other players, the result may be extreme; that is, routes with low prices may share the entire load, whereas those with high prices may share no load at all. Furthermore, a single iteration does not allow routes to adjust their load. Therefore, we treat this game as a dynamic game where players interact with each other by playing the one-shot game numerous times until it converges to an NE. The next section proposes specific rules for the dynamic game.

5. Algorithm

In this section, a distributed DP scheme is presented to implement the dynamic game and enable it to converge in polynomial time. If a source node wants to deliver information via some routes, it first sends a request to all available routes in the strategy set. The game goes by the following specific rules.

(1) Each route j broadcasts an initial price to all source nodes that can communicate with j by wireless connection, and then waits for their responses. The initial price is set as $p_j = \alpha(|B_j| + h_j)$, where α is a price factor.

(2) After receiving the prices from all available routes, each source node i sends a response to the one(s) with minimum price.

(3) After obtaining the response from each source node in B_j, route j decides which node should be accepted. The result is sent to all source nodes in B_j. Whether to accept a source node or not depends on the supply-demand degree of this route, which is formally defined as follows.

Definition 1. The supply-demand degree of a route j is

$$\eta_j^t = \frac{\left|B_j^t\right| + h_j}{p_j^t/\alpha} = \frac{\alpha\left(\left|B_j^t\right| + h_j\right)}{p_j^t}. \tag{6}$$

The above equation means that at iteration t, the maximum number of connections (old and new) that each route can take in is $p_j^t/\alpha - h_j$. Thus, route j balances its supply-demand when the actual number of taken-in source nodes equals the maximum. If newly requested connections plus old connections exceed the maximum, j needs to randomly reject some of the newly requested connections to obtain its supply-demand balance. j then broadcasts the decision.

(4) After receiving the decisions of all routes in S_i, each source node i responds with an acknowledgement to the chosen route. Node i must choose one route randomly if multiple routes intend to accept its request.

(5) Thus far, the interaction between routes and source nodes in a single iteration is finished. Afterwards, each route j needs to determine the price of the next iteration. If the supply-demand is not balanced at the current iteration and there is no new connection accepted or no old connection broken, the price of the next iteration is set to $p_j \leftarrow p_j - \alpha$. Otherwise, the price remains unchanged. If the supply-demand is balanced at the current iteration, the price also remains unchanged and adds a "balanced" mark with the announced price.

The above five steps illustrate the process of each iteration in the dynamic game. The following step decides when the game would be over. First, a definition of the market level supply-demand degree is provided.

Definition 2. The market level supply-demand degree is the average of all the degrees of the routes' supply-demand, that is,

$$N_t = \frac{\sum_{j\in N} \eta_j^t}{n}. \tag{7}$$

Once $N_t = 1$, the whole market is supply-demand balanced, and the game is considered finished (Algorithms 1 and 2).

Two basic metrics for evaluating this algorithm are (1) its ability to converge to a pure NE and (2) the speed it needs to do so. Some significant features of DP are proven from these two aspects.

DP, part 1: executed at each route $j \in N$
Initialization
Set $B_j^0 = \phi$, $\eta_j^0 < 1$, $p_j^1 = \alpha(|B_j| + h_j)$.
repeat
if $\eta_j^{t-1} < 1$ (i.e., unbalanced)
Broadcast its price p_j^t to each source node $i \in B_j$.
else
Broadcast its price p_j^t with a "balanced" signal to each source node $i \in B_j$.
End if
Wait for responses from all source nodes in B_j.
if $|C_j^t| > p_j^t/\alpha - h_j - |B_j^{t-1}| + |L_j^t|$
Randomly choose a subset A_j^t of C_j^t, such that supply-demand is balanced.
Set $B_j^t = B_j^{t-1} \bigcup A_j^t \setminus L_j^t$.
else
Set $B_j^t = B_j^{t-1} \bigcup C_j^t \setminus L_j^t$.
End if
Broadcast its decision to each source node $i \in B_j$.
if $\eta_j^t < 1$ and $|C_j^t| + |L_j^t| == 0$
Set $p_j^{t+1} = p_j^t - \alpha$.
else
Set $p_j^{t+1} = p_j^t$.
$t = t + 1$.
End if
until $N_t = 1$ or $\eta_j^t == 1$ for each $j \in N$(i.e., balance its Supply-demand).
L_j^t: set of source nodes that leave route j during iteration t.
C_j^t: set of source nodes that turn to route j during iteration t.

Algorithm 1

DP, part 2: executed at each source node $i \in M$
repeat
Wait until receiving price from each route $j \in S_i$, then Send a request to route(s) k maximizing u_i^t, then wait for the decision of requested route(s).
if there is more than one route saying "yes" to its request
Choose one randomly to build a formal wireless connection.
End if
$t = t + 1$.
until each route balances its supply-demand.

Algorithm 2

Theorem 3. *The proposed algorithm converges to a pure NE.*

Proof. Let t^* be the number of iterations that the proposed algorithm needs to converge, and let $p_j^{t^*}$ be the final price of any route j.

First, by contradiction, we prove that for every source node $i \in B_j^{t^*}$, $p_j^{t^*} - \alpha \le p_k^{t^*} \le p_j^{t^*} + \alpha$ holds, where $k \in S_i$ is any route available to source node i. Assume $p_k^{t^*} \le p_j^{t^*} - 2\alpha$, and without losing generality let $p_k^{t^*} = p_j^{t^*} - 2\alpha$. Then

(1) suppose the price of route k at iteration $t = 1$ already equals $p_j^{t^*} - 2\alpha$, which means k has balanced its supply-demand from the beginning. Clealy, $p_j^{t^*} - 2\alpha < p_j^{t^*} \le p_j^1$. Thus, the chosen route of player i would be k rather than j, which contradicts the preassumption that i decides to connect to route j;

(2) suppose the price of route k at iteration $t = 1$ is not $p_j^{t^*} - 2\alpha$. There must be some iteration t' between $t = 1$ and $t = t^*$, at which the price of k equals $p_j^{t^*} - \alpha$ and at iteration $t' + 1$, the price equals $p_j^{t^*} - 2\alpha$. Since $p_j^{t^*} - \alpha < p_j^{t^*}$, the chosen route of player i at iteration t' would be k rather than j, which also contradicts the preassumption that i decides to connect to route j.

Considering (1) and (2) jointly, $p_k^{t^*} \le p_j^{t^*} - 2\alpha$ is invalid. Therefore, $p_k^{t^*} \ge p_j^{t^*} - \alpha$. Similarly, we can prove that $p_k^{t^*} \le p_j^{t^*} + \alpha$. Thus, $p_j^{t^*} - \alpha \le p_k^{t^*} \le p_j^{t^*} + \alpha$ holds for every player $i \in B_j^{t^*}$.

Second, we prove that each player i has no incentive to change its choice unilaterally. Without loss of generality, let $p_k^{t^*} = p_j^{t^*} - \alpha$. If i changes its choice from j to k, then k and j need to adjust their prices to $p_j^{t^*}$ and $p_j^{t^*} - \alpha$, respectively, to keep their supply-demand balanced. Note that, the price i has to take is still unchanged. Hence, its choice will not change. □

Lemma 4. *If there is a source node that changes its choice or a route that lowers its price at an iteration t, then we have $N_t > N_{t-1}$ and $\Delta = N_t - N_{t-1} \ge \alpha/n\Lambda(\Lambda - 1)$ where $\Lambda = \max_j(|B_j| + h_j)$.*

Proof. $\Lambda = \max_j(|B_j| + h_j)$ implies that $p_j^t \le \alpha\Lambda$ for each $j \in N$.

(1) Suppose player i changes its choice from k to j, then $p_k^t < p_j^t$. Considering that there may be no alternation in the sets of taken-in source nodes of j and k at iteration $t-1$, their prices may be reduced at iteration t accordingly. Thus, we have $p_j^t \le p_j^{t-1}$ and $p_k^t \le p_k^{t-1}$. Then, the following can be obtained:

$$\begin{aligned}
\Delta &= \frac{1}{n}\left[\left(\eta_j^t - \eta_j^{t-1}\right) + \left(\eta_k^t - \eta_k^{t-1}\right)\right] \\
&= \frac{1}{n}\left[\frac{\alpha\left(\left|B_j^t\right| + h_j\right)}{p_j^t} - \frac{\alpha\left(\left|B_j^{t-1}\right| + h_j\right)}{p_j^{t-1}} + \frac{\alpha\left(\left|B_k^t\right| + h_k\right)}{p_k^t} - \frac{\alpha\left(\left|B_k^{t-1}\right| + h_k\right)}{p_k^{t-1}}\right] \\
&= \frac{\alpha}{n}\left[\frac{\left|B_j^t\right| - 1 + h_j}{p_j^t} - \frac{\left|B_j^{t-1}\right| + h_j}{p_j^{t-1}} + \frac{\left|B_k^t\right| + 1 + h_k}{p_k^t} - \frac{\left|B_k^{t-1}\right| + h_k}{p_k^{t-1}}\right] \\
&\ge \frac{\alpha}{n}\left[\frac{\left|B_j^t\right| - 1 + h_j}{p_j^t} - \frac{\left|B_j^{t-1}\right| + h_j}{p_j^t} + \frac{\left|B_k^t\right| + 1 + h_k}{p_k^t} - \frac{\left|B_k^{t-1}\right| + h_k}{p_k^t}\right] \\
&= \frac{\alpha}{n}\left[\frac{1}{p_k^t} - \frac{1}{p_j^t}\right] \ge \frac{\alpha}{n}\left[\frac{1}{p_j^t - \alpha} - \frac{1}{p_j^t}\right] = \frac{\alpha^2}{np_j^t\left(p_j^t - \alpha\right)} \\
&\ge \frac{\alpha^2}{n\alpha\Lambda(\alpha\Lambda - \alpha)} = \frac{\alpha}{n\Lambda(\Lambda - 1)} > 0.
\end{aligned} \tag{8}$$

This means that $N_{t+1} > N_t$ holds if a source node changes its choice at iteration t.

(2) Suppose route j lowers its price at some iteration t, then $p_j^t = p_j^{t-1} - \alpha$. Furthermore, a lower price may attract extra source nodes to choose j, thus $|B_j^t| \ge |B_j^{t-1}|$. Hence, the following can be obtained:

$$\begin{aligned}
\Delta &= \frac{1}{n}\left(\eta_j^t - \eta_j^{t-1}\right) = \frac{1}{n}\left[\frac{\alpha\left(\left|B_j^t\right| + h_j\right)}{p_j^t} - \frac{\alpha\left(\left|B_j^{t-1}\right| + h_j\right)}{p_j^{t-1}}\right] \\
&\ge \frac{\alpha}{n}\left[\frac{\left|B_j^{t-1}\right| + h_j}{p_j^t} - \frac{\left|B_j^{t-1}\right| + h_j}{p_j^{t-1}}\right] \\
&= \frac{\alpha}{n}\left(\left|B_j^{t-1}\right| + h_j\right)\left[\frac{1}{p_j^{t-1} - \alpha} - \frac{1}{p_j^{t-1}}\right] \\
&\ge \frac{\alpha}{n}\left[\frac{1}{p_j^{t-1} - \alpha} - \frac{1}{p_j^{t-1}}\right] = \frac{\alpha^2}{np_j^{t-1}\left(p_j^{t-1} - \alpha\right)} \\
&\ge \frac{\alpha^2}{n\alpha\Lambda(\alpha\Lambda - \alpha)} = \frac{\alpha}{n\Lambda(\Lambda - 1)} > 0,
\end{aligned} \tag{9}$$

which implies that $N_{t+1} > N_t$ holds if a route changes its price at iteration t.

□

Lemma 5. *$N_t \ge N_{t-1}$ holds and N_t can remain unchanged during two successive iterations at most.*

Proof. Lemma 4 states that $N_{t+1} > N_t$ if there is a source node that changes its choice or if a route lowers its price at an iteration t. Consider the following case: there is neither any source node changes its choice nor any route lowers its price at an iteration t, and some routes are not supply-demand balanced. It can be computed that $\eta_j^t = \eta_j^{t-1}$ according to Definition 1. Thus we have $N_t - N_{t-1} = (1/n)\sum_{j\in N}(\eta_j^t - \eta_j^{t-1}) = 0$. By jointly considering Lemma 4, $N_t \ge N_{t-1}$ can be obtained. N_t can remain unchanged during two successive iterations. Since some routes are still in an unbalanced state and no source node deviates from its initial decision, these routes will reduce their prices at iteration $t+1$ (according to Rule 3). $N_{t+1} > N_t$ can then be obtained according to Lemma 4. This completes the proof. □

Theorem 6. *The number of iterations required to complete the proposed algorithm is $O(n\Lambda^2)$, where $\Lambda = \max_j(|B_j| + h_j)$.*

Proof. Evidently, $N_{t^*} = 1$ and $N_0 \ge 0$. Furthermore, Lemma 5 states that N_t can remain unchanged during two successive iterations at most. The following can then be obtained:

$$\begin{aligned}
t^* &\le 2 \cdot \frac{N_{t^*} - N_0}{\alpha/n\Lambda(\Lambda - 1)} \le 2 \cdot \frac{1}{\alpha/n\Lambda(\Lambda - 1)} \\
&= \frac{2n\Lambda(\Lambda - 1)}{\alpha} = \frac{2}{\alpha}n(\Lambda^2 - \Lambda).
\end{aligned} \tag{10}$$

Table 1: Simulation parameters.

Parameter	Value
Number of source nodes	$20, 40, \ldots, 1000$
Ratio of source nodes to routes	2, 3, 4
Degree of source nodes	$1, 2, \ldots, 10$

Equation (10) indicates that the proposed algorithm terminates in polynomial time, and the upper bound is $O(n\Lambda^2)$. □

6. Simulation Results

We conduct simulations to assess the performance of the proposed algorithm in terms of executed time and to ascertain the optimality of the NE and provide the results in this section. To obtain reliable results, several parameters are investigated, including the number of source nodes, the ratio of source nodes to routes denoted by r, and the degree of source nodes, which is defined as the number of routes that each source node can connect to. The values of these parameters are given in detail in Table 1. Since having assumed that the transmission rate and packet length are the same for all nodes, these two measurements are not included in the variable table. And note that our simulations are dedicated for quite generalized wireless M2M networks and thereby do not rely on specific protocols in MAC layer; it is unnecessary to specify which wireless channel is in use and how to set the window size. A large number of runs are conducted for each simulation and the results are averaged.

6.1. Executed Time. Section 4 shows that the proposed algorithm can be completed in polynomial time and gives an upper bound of the executed time. However, the real speed is essential in practice. Here, we evaluate the practical speed by comparing it with the theoretical time bound T given by (10). $\lambda = P/T$ is denoted as the ratio of practical time to the theoretical time bound (P-T ratio), where P is the practical time that the proposed algorithm takes.

First, we study the effect of the number of source nodes on the proposed algorithm's speed, and the other parameters are set as constants. Figure 2 shows how many iterations the proposed algorithm needs to be completed versus the different number of source nodes when the degree of source nodes is set to an average of 5, and Figure 3 illustrates the value of λ under each corresponding case. It can be seen that with the increase of the number of source nodes, the proposed algorithm requires more iterations to be completed, but the increasing trend is not rapid and it needs only 200 iterations with 1000 source nodes and 250 routes ($r = 4$). Almost in contrast to the trend of absolute iterations, the P-T ratio λ decreases sharply from the beginning with the number of source nodes increasing until around 200, where $\lambda < 0.01$, followed by a gradual decline. Moreover, the value of λ is fairly close to 0 with 1000 source nodes. The reason for this phenomenon may be that the given upper time bound is loose.

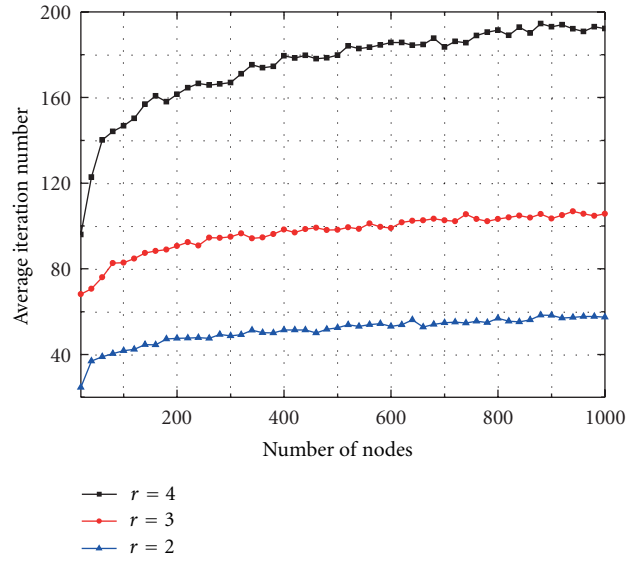

Figure 2: Number of iterations the proposed algorithm takes versus the number of source nodes.

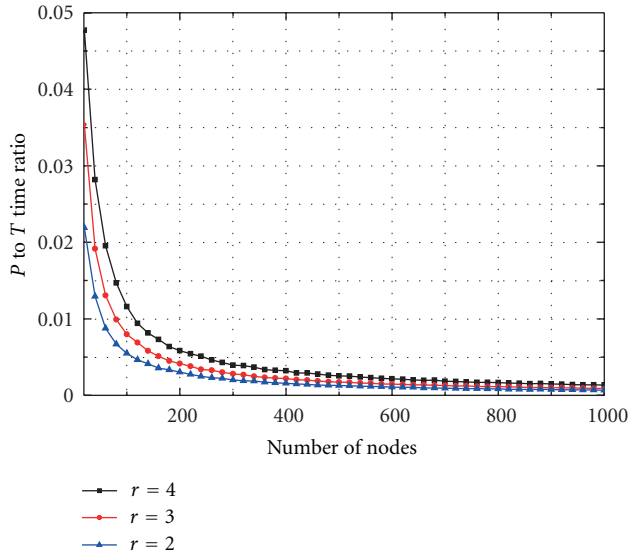

Figure 3: P-T ratio λ versus the number of source nodes.

Second, the degree of source nodes is considered as variable, and 10 groups of the maximum degree of the source nodes are set to $1, 2, \ldots, 10$, respectively. Figure 4 shows that the required average iterations by the proposed algorithm increases almost linearly with the increase in the number of source nodes. In particular, when the maximum degree of each source node is 1, only one iteration is needed because each has only one choice. When there are 400 source nodes with the maximum degree of 10, the proposed algorithm takes nearly 400 iterations to be completed. Figure 5 shows that λ reaches its maximum (slightly below 0.02) when each source node has two available routes on average. Except in a special case where each source node has only one choice,

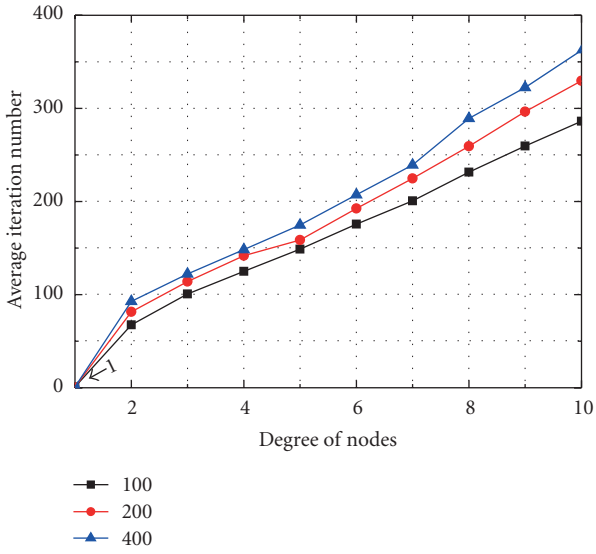

Figure 4: Number of iterations the proposed algorithm takes versus the degree of source nodes.

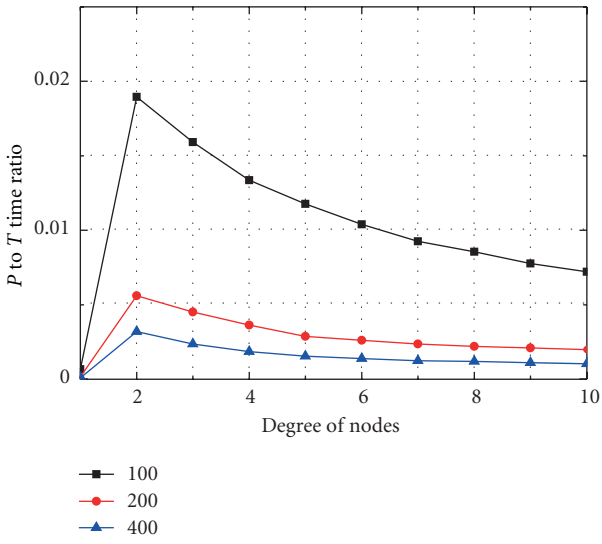

Figure 5: P-T ratio λ versus the degree of source nodes.

the P-T ratio λ decreases slowly as each source node has more choices because each has multiple routes to choose from.

On one hand, the given upper time bound is rather loose. On the other hand, the time bound is polynomial and the P-T ratio λ is extremely small in the given simulation. Therefore, the proposed algorithm has good speed performance.

6.2. Optimality. Section 4 indicates that a one-shot style may render the NE of the game undesirable. This part shows how the dynamic game can improve NE by simulation. Figure 6 provides the comparison results when the degree of source nodes is set to an average of 5 and r to 4. Evidently, the latency under OneShotHop is nearly triple of DynamicPricing. Moreover, OneShotPricing is merely better by one point compared with OneShotHop.

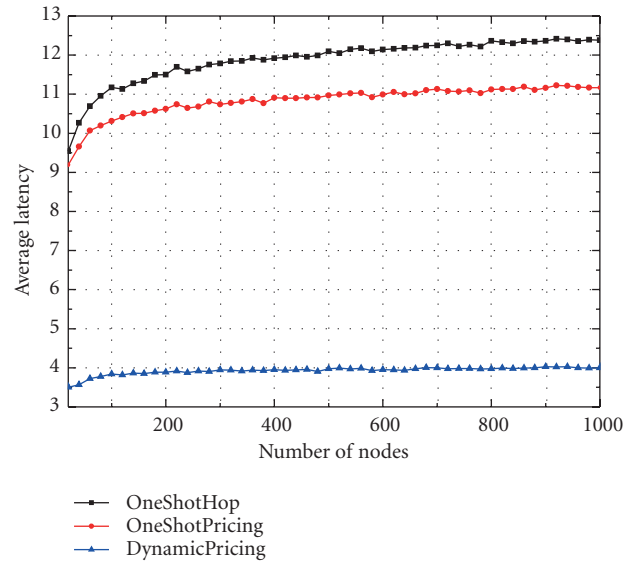

Figure 6: Latency under DynamicPricing, OneShotHop, and OneShotPricing versus the number of source nodes.

Since our proposed algorithm is distributed, there is no guarantee that it can compute the optimal NE every time as a centralized approach can. Therefore, learning the optimality of the NE generated by DP is significant. Here, several experiments are conducted to examine the optimality of the proposed algorithm by comparing it with a well-known centralized approach based on the cost-reducing path and to determine whether it can terminate in polynomial time [18]. Before showing the results, price of anarchy (PoA) [19], which measures the optimality of a game, is introduced as a metric and formally given as follows.

Definition 7 (PoA). For game G, let $\cos t(G)$ denote the social cost of G's NE, and let $\text{opt}(G)$ denote the minimum social cost over all assignments. The price of anarchy is then defined by

$$\text{PoA}(G) = \frac{\cos t(G)}{\text{opt}(G)}. \tag{11}$$

PoA is constantly not smaller than 1 (1 means optimal). The smaller the PoA is, the better the NE becomes.

First, the effect of the number of source nodes on the proposed algorithm's optimality is studied. The degree of source nodes is set to an average of 5. Figure 7 shows that as the number of source nodes increases, the NE's PoA tends to become larger despite small fluctuations. The maximum value of PoA is below 1.40 with 1000 source nodes and 500 routes. The corresponding ratio of the number of runs that generates an optimal NE to that of all runs (optimal-to-all for short) is given in Figure 8. Most of the runs can generate an optimal assignment.

Second, the degree of source nodes is considered to be variable. Ten groups of the maximum degree of the source nodes are set to $1, 2, \ldots, 10$, respectively. When each source

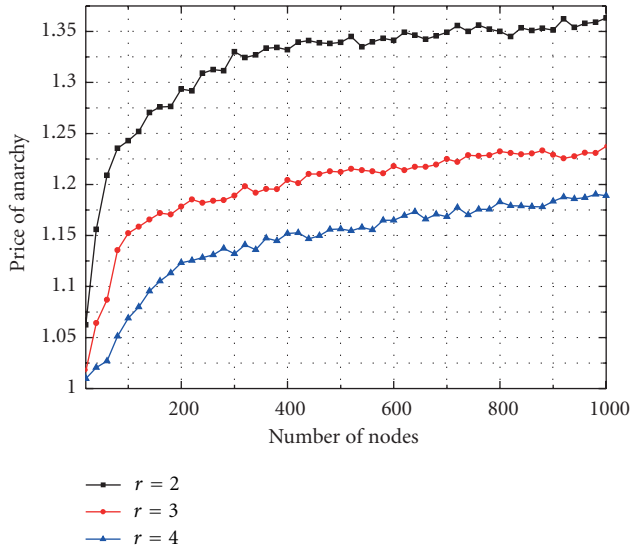

Figure 7: PoA versus the number of source nodes.

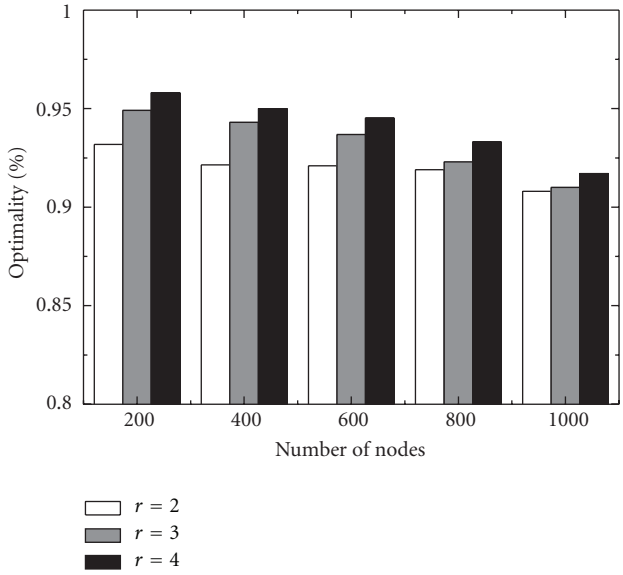

Figure 8: Ratio of optimal-to-all versus the number of source nodes.

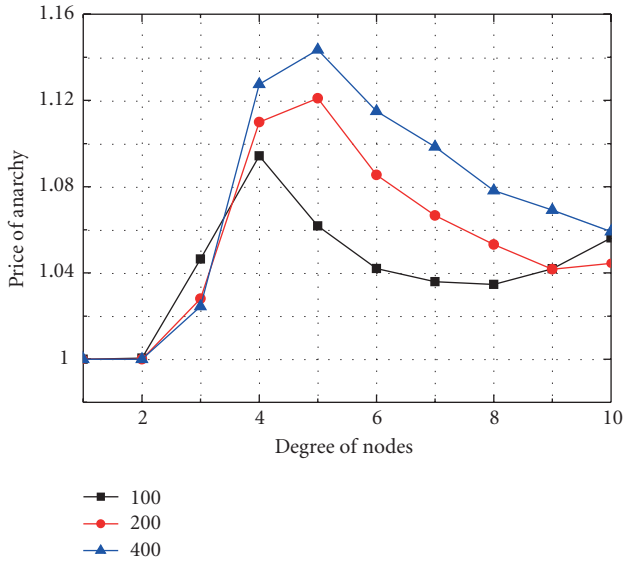

Figure 9: PoA versus the degree of source nodes.

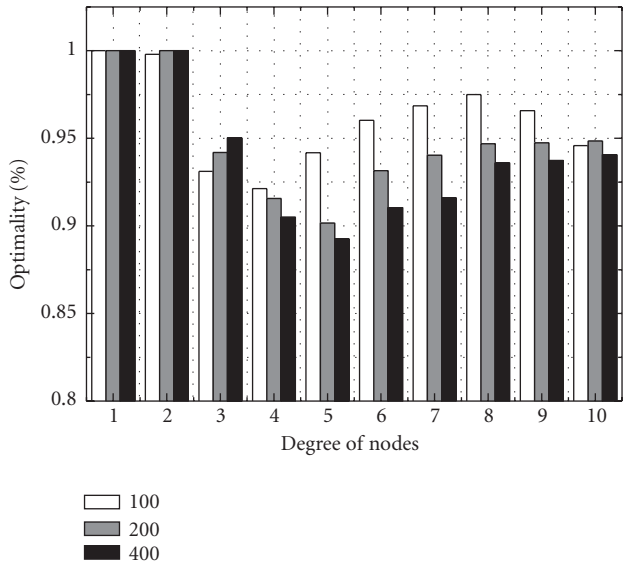

Figure 10: Ratio of optimal-to-all versus the degree of source nodes.

node has only one route to connect to, the assignment is unique, and thus PoA = 1, which is in accordance with that shown in Figure 9. If each source node has multiple choices, the optimality of the proposed algorithm becomes slightly poor. However, the maximum PoA does not exceed 1.16, which is close to 1. Figure 10 provides the information on the ratio of optimal-to-all in each case corresponding to Figure 9. Over 90% of runs can generate an optimal assignment.

7. Conclusion

The routing congestion problem in wireless M2M networks has been investigated using a game theoretic approach. A network congestion game has been formulated, in which many selfish source nodes compete for connecting to a smaller set of routes. Since full cooperation among nodes cannot be assumed in wireless M2M networks, pricing has been applied in the game to motivate cooperative data relay. By setting the pricing scheme dynamic, routes altering their prices and source nodes accordingly altering their connected routes, we have proved that the network congestion game converges to a pure NE in polynomial time. The optimality of the proposed algorithm was evaluated by simulation. The results indicate good practical performance.

Acknowledgments

This research is partially supported by research grants from the National Science Foundation of China under Grant-nos.70701025 and 71071105, the Program for New Century Excellent Talents in Universities of China under Grant no. NCET-08-0396, and a National Science Fund for Distinguished Young Scholars of China under Grant no. 70925005, and the Program for Changjiang Scholars and Innovative Research Team in the university (IRT1028).

References

[1] Q. Zhang and Y. Q. Zheng, "Cross-layer design for qos support in multihop wireless networks," *Proceedings of the IEEE*, vol. 96, no. 1, pp. 64–76, 2008.

[2] P. K. Huang, X. Lin, and C. C. Wang, "A low-complexity congestion control and scheduling algorithm for multihop wireless networks with order-optimal per-flow delay," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '11)*, pp. 2588–2596, April 2011.

[3] K. Tan, F. Jiang, Q. Zhang, and X. Shen, "Congestion control in multihop wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 2, pp. 863–873, 2007.

[4] H. J. Lee and J. T. Lim, "Congestion control for streaming service in IEEE 802.11 multihop networks," *IET Communications*, vol. 4, no. 12, pp. 1415–1422, 2010.

[5] J. Y. Teo, Y. Ha, and C. K. Tham, "Interference-minimized multipath routing with congestion control in wireless sensor network for high-rate streaming," *IEEE Transactions on Mobile Computing*, vol. 7, no. 9, pp. 1124–1137, 2008.

[6] L. Chen, S. H. Low, M. Chiangs, and J. C. Doyle, "Cross-layer congestion control, routing and scheduling design in ad hoc wireless networks," in *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM '06)*, April 2006.

[7] B. Vöcking and R. Aachen, "Congestion games: optimization in competition," in *Proceedings of the Proceedings of the 2nd Algorithms and Complexity in Durham Workshop*, H. Broersma, S. Dantchev, M. Johnson, and S. Szeider, Eds., Kings College Publications, London, UK, 2006.

[8] H. Sperber, "How to find Nash equilibria with extreme total latency in network congestion games?" in *Proceedings of the International Conference on Game Theory for Networks (GameNets '09)*, pp. 158–163, May 2009.

[9] D. Fotakis, S. Kontogiannis, and P. Spirakis, "Symmetry in network congestion games: pure equilibria and anarchy cost," *Approximation and Online Algorithms*, vol. 3879, pp. 161–175, 2006.

[10] C. A. Gizelis and D. D. Vergados, "A survey of pricing schemes in wireless networks," *IEEE Communications Surveys and Tutorials*, vol. 13, no. 1, pp. 126–145, 2011.

[11] X. Zhu, S. Han, and B. Girod, "Congestion-aware rate allocation for multipath video streaming over ad hoc wireless networks," in *Proceedings of the International Conference on Image Processing (ICIP '04)*, pp. 2547–2550, October 2004.

[12] L. Chen, S. H. Low, and J. C. Doyle, "Joint congestion control and media access control design for ad hoc wireless networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '05)*, pp. 2212–2222, March 2005.

[13] M. Chiang, "Balancing transport and physical layers in wireless multihop networks: jointly optimal congestion control and power control," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 104–116, 2005.

[14] C. Wang, B. Li, K. Sohraby, M. Daneshmand, and Y. Hu, "Upstream congestion control in wireless sensor networks through cross-layer optimization," *IEEE Journal on Selected Areas in Communications*, vol. 25, no. 4, pp. 786–795, 2007.

[15] P. Wang and S. Bohacek, "Practical computation of optimal schedules in multihop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 2, pp. 305–318, 2011.

[16] C. Jiang, Y. Shi, Y. T. Hou, and S. Kompella, "On optimal throughput-energy curve for multi-hop wireless networks," in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM '11)*, pp. 1341–1349, April 2011.

[17] Y. P. Hsu and K. T. Feng, "Cross-layer routing for congestion control in wireless sensor networks," in *Proceedings of the IEEE Radio and Wireless Symposium (RWS '08)*, pp. 783–786, January 2008.

[18] N. J. A. Harvey, R. E. Ladner, L. Lovász, and T. Tamir, "Semi-matchings for bipartite graphs and load balancing," *Journal of Algorithms*, vol. 59, no. 1, pp. 53–78, 2006.

[19] C. H. Papadimitriou, "Algorithms, games, and the internet," in *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pp. 749–753, July 2001.