A Q-Learning-based Power-Controlled Routing Protocol in Multihop Wireless Ad Hoc Network

Ke Wang  
NUS Graduate School for Integrative Sciences and Engineering  
National University of Singapore  
Email: wangke@nus.edu.sg

Teck Yoong Chai  
Institute for Infocomm Research  
Agency for Science, Technology and Research  
Email: tychai@i2r.a-star.edu.sg

Wai-Choong Wong  
Department of Electrical and Computer Engineering  
National University of Singapore  
Email:wonglawrence@nus.edu.sg

Abstract—In wireless ad hoc networks, power control has great impact on routing since transmission range is directly determined by a node’s transmission power. Higher power can give higher connectivity and shorter path. However, larger transmission range causes more interference to nearby neighbors and may further impair overall network performance. We propose a Q-Learning-based Power-Controlled Routing (QLPCR) protocol which makes use of Q learning techniques for routing and power control to optimize delay performance of the whole network. A Markov chain CSMA/CA delay model is used to estimate delay of each link in order to determine the optimal power level for all possible routing options.

I. INTRODUCTION

A wireless ad hoc network is a highly distributed network environment. Therefore, many well-addressed problems in wired or mobile networks, such as power control and routing, are still open challenges. Power level, for instance, has significant impact in wireless ad hoc network. It determines a node’s transmission range and therefore its interference range. In wireless ad hoc networks, higher transmission power means fewer hops and better signal-to-interference and noise ratio (SINR). However, to its neighbors, it usually means the opposite.

Lots of research works have been done in the field of wireless ad hoc power control schemes. For example, the Local Mean Algorithm (LMA) [1] tries to maintain each node’s number of neighbors between NodeMinThresh and NodeMaxThresh. The Local Mean of Neighbors algorithm (LMN) [1], on the other hand, does not have a fix threshold. A node tries to keep its number of neighbors as the same as the mean value from its neighbors’ number of neighbors. The Common Power (COMPOW) [2] algorithm, as its name suggests, tries to find a common power level for all nodes such that the whole network maintains connected. One common focus of all these power control schemes is connectivity. It is believed that as long as a whole network maintains connected, smaller power is always preferred since it can prolong the nodes’ life time. This is generally true if routes do not change. However, if a shorter path is used, the overall energy consumption for transmitting a packet may decrease with increased power, although, larger interference is also created at the same time. Therefore, it is quite necessary to consider routing when dealing with power control problems.

There are lots of well-established routing protocols such as AODV [3] and DSR [4]. However, these protocols are all based on minimal-hop algorithms and therefore power-insensitive. GRAdient Cost Establishment (GRACE) [5] uses both link quality and remaining energy level as path cost. Min-Max Battery Cost Routing (MMBCR) [6] tries to find a path with minimum power consumption or maximum remaining energy. The main purpose of these protocols is to ensure that those centrally located nodes will not be drained out fast.

Combining power control and routing can be very challenging. Some researchers try to combine an existing power control and power-aware routing scheme. For example, Dynamic Transmission Power Control (DTPC) [7] tries to add the LMN algorithm into GRACE to avoid network partitioning. Power sensitive power control (PCON) [8] tries to improve the MMBCR by gradually adjusting each node’s transmission power according to its residual energy. In these protocols, the power control scheme does not directly consider its impact on routing. There are some specially designed cross-layer power control routing protocols as well. For example, the Power Control Routing (PCR) [9] protocol uses the number of interfered neighbors as the cost to determine routing paths, and consequently finds a power level to achieve overall minimal path cost. Since it explicitly tries every possible power level, setup time and convergence rate are quite difficult to optimize.

QoS provisioning in wireless ad hoc network is not easy since the topology in a wireless ad hoc network is changing all the time. The Q-learning method [10] is a possible solution [11]-[12] because Q-learning can quickly make use of surrounding information and act accordingly. Hence, we propose a Q-learning-based power-controlled routing (QLPCR) protocol which makes use of the Q-learning method for minimal-delay routing decisions. The power control scheme is based on each node’s Q-value table to optimize overall network delay performance. Unlike other Q-learning methods which uses real rewards from neighbors, QLPCR uses a CSMA/CA delay analysis model to estimate the delays for all traffic.

II. PROTOCOL DESIGN

A. System Overview

In order to optimize QoS performance, a minimal-delay-Q-learning-based routing protocol is used in our system because
it can adaptively find the path of shortest delay to a given destination. However, there are two major drawbacks for Q-learning-based routing. One is selfishness which means each node only cares about its own traffic delay. If there are several flows in a network, the overall performance cannot be optimized because congestions are very likely to form in central areas. This is because Q learning can only give the best solution provided the environment is not changing. If every node is trying to minimize its delay, the whole network can be trapped in a local optimal operation point. Therefore, we introduce a corresponding power-control scheme to avoid such an effect. This power-control scheme can change one node’s transmission power to control its interference level to optimize overall network performance. Besides, by controlling the transmission range, it also can limit the options of the Q-learning routing.

The other drawback of Q-learning-based routing is its convergence rate. For normal Q-learning-based routing protocols such as [11]-[12], they require physical feedbacks/rewards from neighbors. For example, if the purpose of a Q-learning-based routing is to minimize delay, each node needs to know its transmission delay to its neighbors. Theoretically speaking, such information can only be obtained by "try and observe", meaning that sending packets to all neighbors and waiting for their feedbacks. Although the former can be done by broadcasting, the delay feedbacks can only be done by unicasting. Since power control is also embedded in our system, each (Power level, Neighbor) pair needs a delay value in Q-learning.

Since power control is also embedded in our system, each (Power level, Neighbor) pair needs a delay value in Q-learning. Theoretically speaking, such information can only be obtained by "try and observe", meaning that sending packets to all neighbors and waiting for their feedbacks. Although the former can be done by broadcasting, the delay feedbacks can only be done by unicasting. Since power control is also embedded in our system, each (Power level, Neighbor) pair needs a delay value in Q-learning. Therefore, it becomes impractical to implement a real-time feedback system because the action-reward space is too large. Instead, a good delay model is needed to conveniently estimate all delays.

B. CSMA/CA Delay Model

The purpose of this model is to calculate one node’s expected delay given the number of its active transmitting neighbors and its current transmission rate. The delay result is used in both the Q-learning routing and the power-control scheme.

We assume that the network consists of $n$ contending nodes with full load, which means that after completing the transmission of a packet, there is always another packet ready for transmission. Since [13] assumes that each packet has an infinite number of retransmission attempts until it succeeds, the model is changed to have $m$ maximal allowed retransmission attempts and compulsory backoff for a new packet transmission as shown in Fig. 1. The bi-dimensional $\{s(t), bo(t)\}$ is a discrete-time Markov chain with $bo(t)$ as a stochastic process that represents the value of the backoff counter and $s(t)$ as the backoff stage. $p_c$ and $p_b$ are the colliding probability and sensing busy probability, respectively.

The following equations can be obtained directly:

$$b_{i,0} = p_c^i b_{0,0} \text{ for } 0 \leq i \leq m$$  \hspace{1cm} (1)

$$b_{\text{fail}} = p_c^{m+1} b_{0,0}$$  \hspace{1cm} (2)

$$b_{i,k} = \frac{W_i - k}{W_i} \times \frac{1}{1 - p_b} \times b_{i,0} \text{ for } 0 \leq i \leq m, 0 < k < W_i$$  \hspace{1cm} (3)

$$\sum_i \sum_k b_{i,k} + b_{\text{fail}} = 1$$  \hspace{1cm} (4)

where $b_{i,k}$ is the state probability of $\{i, k\}$ and $W_i$ is the contention window for $i$th retransmission. If we define $\tau$ as the probability that one node transmits at a time slot duration, then

$$\tau = \sum b_{i,0} = \sum_i p_c^i b_{0,0} = \frac{1 - p_c^{m+1}}{1 - p_c} b_{0,0}$$  \hspace{1cm} (5)

$p_b$ is the probability of sensing the channel busy which is the probability that among the remaining $n - 1$ nodes, one or more nodes transmit at one time slot duration. $p_c$ is the probability of a transmitting frame colliding given that it is transmitted. Therefore it is also the probability that another one or more nodes transmit at one time slot duration:

$$p_c = p_b = 1 - (1 - \tau)^{n-1}$$  \hspace{1cm} (6)

From (1) - (6), $b_{0,0}$, $\tau$, $p_c$ and $p_b$ can be worked out. Let $T_s$ and $T_c$ be the time for a successful and colliding transmission attempt. Therefore, the expected total backoff delay $E[BD]$ can be calculated according to [13]. The expected transmitting delay $E[TD]$ is:

$$E[TD] = \sum_{k=0}^{m-1} (1 - p_c) p_c^k (kT_c + T_s) + p_c^m mT_c$$  \hspace{1cm} (7)

Overall transmission delay would be $E[D] = E[BD] + E[TD]$. If we consider each node as an $m/m/1/k$ queue, then its departure rate would be $1/E[D]$. Assume its packet arrival rate is $\lambda$ and $\rho = \frac{\lambda}{E[D]}$. So the overall delay is

$$D_{\text{overall}} = \frac{\rho - \rho^{k+1} - k\rho^{k+1} + k\rho^{k+2}}{(1 - \rho)(1 - \rho^k)\lambda}$$  \hspace{1cm} (8)

Since $\rho$ is also a function of packet arrival rate $\lambda$, it can be seen that $D_{\text{overall}}$ can be considered as a function of number of node $n$ and $\lambda$. In short:

$$D_{\text{overall}} = Delay(n, \tau)$$  \hspace{1cm} (9)

This model is computationally complex since (1) - (6) involves high-order equations. However, since $n$ is a discrete variable, a hash table which stores different $E[D]$ values for different values of $n$ can be used to simplify the calculation.

C. Q-Learning based routing

Q learning [10] which consists of state-action pairs is a special form of reinforcement learning. For each pair, a corresponding Q value is assigned to indicate its effect. If delay is used as the Q value, the original formula of Q learning in [14] can be slightly modified as:

$$D_{t+1}(d, \text{next\_hop}) = (1 - \alpha)D_t^x(d, \text{next\_hop}) + \alpha\left(R_{\text{next\_hop}} + \gamma \min_z D_t^{x_{\text{next\_hop}}}(d, z)\right)$$  \hspace{1cm} (10)
where $D_{x(t+1)}^{\text{next hop}}(d)$ is the expected delay for the current node $x$ choosing the next hop $\text{next hop}$ for the destination $d$ at $(t + 1)^{th}$ iteration. $R_{x(t+1)}^{\text{next hop}}$ is the expected delay between node $x$ and $\text{next hop}$. $\alpha$ is the learning rate, $\gamma$ is the discount factor. $D_t^{\text{next hop}}(d, z)$ is the possible minimal future delay for the $\text{next hop}$.

$R_{x(t+1)}^{\text{next hop}}$ is calculated using (9) if the number of neighbors and packet arrival rate are known. $\min D_t^{\text{next hop}}(d, z)$ is obtained from neighbors’ broadcasting information.

### D. Information Sharing System

Information sharing system is critical because all the other subsystems need to make use of their neighbors’ information. To be specific, the Q-learning-based routing needs to obtain minimal delays estimation $\min D_t^{\text{next hop}}(d, z)$ from their neighbors. The CSMA/CA delay model needs to make use of the number of active transmitting neighbors. As for the power control scheme, a special term ‘price’ is used to indicate the loss of being interfered.

Because of the full-load assumption, a threshold is needed to classify nodes as inactive if its transmission rate is smaller than the threshold and vice versa. Therefore, each node’s current transmission rate should be known to its neighbors.

Since both the sender and receiver’s neighbors participate in contention, we need to know the number of both the sender’s and the receiver’s active transmitting neighbors to calculate the delay. However, simply summing them up will double-count those mutual neighbors. In fact, if both the sender’s and the receiver’s transmission range, the number of active transmitting neighbors and the distance are known, a probability model can be used to calculate the expected number of overall contending neighbors.

Since the sender’s and the receiver’s respective number of active transmitting neighbors are known, the problem is reduced to find the sender’s and the receiver’s mutual active transmitting neighbors.

For each active transmitting neighbor of the sender, its probability of being the receiver’s active transmitting neighbor is calculated. By summing all the probabilities up we can get the expected total number of mutual active transmitting neighbors.

Fig. 2 shows how we can calculate the probability. Assume that $\text{Neighbor}$ is one of $\text{Sender}$’s active transmitting neighbors. Since $\text{Receiver}$ and $\text{Neighbor}$’s distance is known to $\text{Sender}$, two circles are drawn to indicate their possible locations. Then the third circle indicates $\text{Receiver}$’s transmission range. The bold arc between two circles is the possible...
locations for Neighbor being a mutual active transmitting neighbor.

The log-distance path loss model is applied in our system. Each node will periodically broadcast information using maximal power level, which is also known to all nodes. Therefore, when a broadcasting packet is received, the distance between the sender and the receiver is known to the receiver immediately. For the same reason, transmission range can be calculated using transmission power as well.

In conclusion, the following information is included in the broadcasting packets:

- Minimal delays: minimal delays to different destinations
- Power level: current transmission power of sender
- Transmission rate: current transmission rate of sender
- Number of active transmitting neighbors: number of neighbors that are within sender’s transmission range with transmission rate greater than the threshold
- Price: the price of being interfered. This will be explained later.

In this paper, we do not consider rate control, therefore, power level is optimal when it is just enough to reach the next hop. Therefore, the problem of power control is reduced to choosing its transmission range in terms of the distance to the furthest neighbor reachable. We define \( \phi(x) \) as the power level cost function for node \( x \)

\[
\phi(x) = \min_{d \in \{n|n \in N_x, \|d\| \leq \rho_x^d \}} (D_t^x(d,l)) \times \alpha^d + \sum_l C_l
\]

where the first term is the overall weighted delay of node \( x \) itself. The second term is the interference level caused to its neighbors. In order to find the optimal power level, we need to find \( z \) which can minimize \( \phi_x(z) \), and then power level is chosen to be \( \rho_x \).

### III. SIMULATION RESULTS

We use Network Simulator version 3.12.1 (ns-3.12.1) for our simulations. Three power-control schemes: LMN, LMA [1] and PCR [9] with combined Q-learning-based routing protocols are compared with QLPCR. The original AODV protocol with maximal transmission power is also included in our simulations. The following is the parameters used in the setup:

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>SIMULATION PARAMETERS</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC layer</td>
<td>802.11 DCF with RTS/CTS</td>
</tr>
<tr>
<td>Simulation area</td>
<td>1500m x 300m</td>
</tr>
<tr>
<td>Maximum speed</td>
<td>1m/s</td>
</tr>
<tr>
<td>Mobility pattern</td>
<td>Random way-point model</td>
</tr>
<tr>
<td>Packet size</td>
<td>1024 bytes</td>
</tr>
<tr>
<td>Flow rate</td>
<td>0.02mbps</td>
</tr>
<tr>
<td>Flow number</td>
<td>5-40</td>
</tr>
<tr>
<td>Radio channel Rate</td>
<td>2Mpbs</td>
</tr>
<tr>
<td>Maximum transmission range</td>
<td>250 m</td>
</tr>
<tr>
<td>Number of runs</td>
<td>20</td>
</tr>
</tbody>
</table>

### A. CSMA/CA Model Verification

First, the CSMA/CA delay model needs to be verified. The simulation area is restricted to 1 meter by 1 meter to make sure that all nodes are competing with each other to access the media. The number of flows is always the same as the number of nodes to ensure full load. Fig. 3a shows that the simulation result is quite close to the calculation result.

### B. Performance Analysis

Flows are generated using random source-destination pairs. LMA’s upper and lower limits are both set to 4. Fig. 3b-3d show the delay, packet delivery ratios (PDR) and throughput respectively. It can be seen that QLPCR has the best performance because each node can dynamically adjust its power
level according to current traffic conditions and the status of the surrounding neighbors.

The major advantage of LMN is that for a given area, each node’s connectivity is proportional to this area’s node density. Therefore, traffic can be well balanced among those centrally located nodes by using Q-learning-based routing. That is also the reason LMN has a good performance in terms of PDR and throughput. However, since LMN does not consider traffic conditions, it is expected that all flows have to go through a longer path, which is not desirable if the network is not congested. Therefore, its delay is twice the delay of QLPCR.

LMA does not perform well because its number of neighbors is fixed to a small value. In central areas, congestion is more likely to happen because the choice of routing is very limited. In fact, it can be seen that even when there are only 5 flows, the network is already congested.

PCR uses the number of interfered neighbors as link weights. As a result, those far away nodes with quite limited number of neighbors are preferred. As a result, congestion is quite severe even when traffic is not that heavy. For the same reason, PCR has the largest delay since it tends to avoid central areas by using extra hops to get to destinations. PCR does not change with traffic load because each node is treated equally. When traffic is light or medium, using paths with more hops is obviously not a good choice.

AODV transmits using the maximal power level. Therefore, when traffic is light, its performance is very good because of the minimal hop algorithm. However, when traffic load increases, the effect of interference begins to dominate. It can be seen that when the number of flows reaches 20, the whole network begins to congest and its performance declines drastically after that.

IV. CONCLUSION

In this paper, we propose a Q-Learning-based Power-Controlled Routing (QLPCR) protocol which makes use of Q-learning techniques to make routing and power-control decisions in order to optimize QoS performance. A CSMA/CA delay model is used to estimate delays between two nodes. The simulation shows that our protocol can perform well in various loading conditions.

ACKNOWLEDGEMENT

This research was carried out at the NUS-ZJU Sensor-Enhanced Social Media (SeSaMe) Centre. It is supported by the Singapore National Research Foundation under its International Research Centre @ Singapore Funding Initiative and administered by the Interactive Digital Media Programme Office.

REFERENCES