A Game Theory Based Congestion Control Protocol for Wireless Personal Area Networks

Jang-Ping Sheu
Department of Computer Science
National Tsing Hua University
Hsinchu, 30013, Taiwan
sheujp@cs.nthu.edu.tw

Chao-Xiang Hsu
Department of Computer Science
National Tsing Hua University
Hsinchu, 30013, Taiwan
alenbos0517@gmail.com

Chuang Ma
School of Computer and Information
Shanghai Second Polytechnic
University
Shanghai, 201209, China
machuang@sspu.edu.cn

***Abstract*—In Wireless Sensor Networks (WSNs), the presence of congestion can increase the ratio of packet loss, energy consumption and reduce of the network throughput. Especially, this situation will be more complex in Internet of Things (IoT) environments, which is composed of thousands of heterogeneous nodes. In this paper, we address the congestion problem between child and parent nodes in RPL-enabled networks, which typically consist of low power and resource constraint devices. We use game theory strategy to design a parent-change procedure which decides how nodes changing their next hop node toward sink to mitigate the effect of network congestion. Comparing to the ContikiRPL implementation, the simulation results show that our protocol can achieve more than twofold improvement in packet loss rate and throughput with similar average hop count.**

Keywords—congestion control*; *game theory*; *IoT*; *IPv6*; *wireless sensor networks

I. INTRODUCTION

In the past few years, we have witnessed a large evolution of Internet of Things (IoT) in our life [1, 2]. IoT has been used to connect industrial devices, hospital instruments, household appliances such as refrigerators, air conditions and TV sets, etc. These systems usually include many end devices that conform to IEEE 802.15.4 standard and are often characterized by short transmission range, low data rate, low cost and low communication power. In order to connect and get access to these devices, the Internet protocol IPv6 is proposed and it serves as the best solution due to its popularity, applicability, available address space, support for stateless address auto-configuration and easiness to access [3].

However, there are some difficulties to implement IPv6 in the wireless personal area networks. For instance, IEEE 802.15.4 supports only 127 bytes as maximum transmission size packet whereas IPv6 needs 1280 bytes as maximum transmission size. Thus, the Internet Engineer Task Force (IETF) working group has standardized an adaptive layer to use in the devices with IEEE 802.15.4 MAC/PHY called IPv6 over Low power Wireless Personal Network (6LoWPAN) [4]. With 6LoWPAN implementation, IEEE 802.15.4 devices will gain the ability to receive, process, and forward IPv6 packets. Based on 6LoWPAN, IETF further proposed IPv6 Routing Protocol for Low-power and Lossy Networks (RPL) [5].

RPL is a tree-like topology routing protocol supporting multiple point to single point (sink node), single point to multiple point and point-to-point traffic which can be used in the applications of Wireless Sensor Networks (WSNs) such as environment monitoring and body sensing. Because of the tree-like topology, if an event occurs in the leaf node, all the nodes in the event region will send packets to sink and may cause network congestion towards sink node. In WSNs, congestion control [6] contains congestion detection and congestion avoidance. In general, congestion detection adopts few metrics such as buffer occupancy, channel loading, and the ratio of packet inter-arrival time to packet service time to detect the presence of congestion. When congestion is detected, we use the congestion avoidance mechanisms to mitigate the presence of congestion.

Currently, there is no explicit mechanism to detect or to avoid congestion in RPL protocol. In fact, RPL protocol uses a simple parent selection mechanism to avoid selecting parents with bad link quality, large hop count, or large expected transmission count [7]. However, these schemes may result in Ping-Pong effect where a node changes its parent frequently. Even, Ping-Pong effect will lead to significant packet loss, decreased throughput, and increased packet delay.

In this paper, we propose a congestion control protocol called Game Theory based Congestion Control Protocol (GTCC) based on game theory over RPL to alleviate the effect of congestion. In our approach, nodes will be informed about the presence of congestion by their parent through control messages. In this case, a child node will decide whether it changes its parent or not based on the game theory strategy [8]. Thus, our scheme can eliminate congestion by changing parent with light load and avoid the ping-pong effect.

The rest of this paper is organized as follows. Section II introduces related work. The detail of our protocol is presented in Section III. The simulation results are described in Section IV, and Section V concludes this paper.

II. RELATED WORK

We divide previous congestion control schemes into two categories: rate adjustment schemes and alternative path selection schemes. In rate adjustment mechanisms [9, 10, 11], the nodes will reduce their sending rate to decrease the number of packets in local and global networks when they detect the congestion. In alternative path selection

0730-3157/15 $31.00 © 2015 IEEE
DOI 10.1109/COMPSAC.2015.21

mechanisms [12, 13], if a node detects congestion, it would try to find another better path to transmit packets. We will review both kind of mechanisms in this section.

A. Rate adjustment mechanisms

In [9], the authors proposed hop-by-hop congestion control and load balancing scheme called CONSEQ in WSNs. It uses special effective queue length (EQL) as metric of congestion degree. CONSEQ dynamically adjusts transmission rate according to degree of congestion of each node in its forwarding set which contains neighbor nodes with smaller hop count to sink. If congestion is not mitigated, each node will use fuzzy logic to reduce the transmission rate.

The authors in [10] proposed a Priority-based Congestion Control Protocol (PCCP). PCCP uses ratio of packet inter-arrival time to packet service time as a metric of congestion. Once the congestion is detected, nodes will use the transmission rate of upward nodes and priority of packet to adjust its transmission rate.

The authors in [11] proposed a priority based congestion control for heterogeneous traffic in multipath WSNs. The congestion is detected by packet service rate. When congestion is detected, it will adjust the transmission rate by considering priority and traffic rate of neighbors in next transmission period to mitigate congestion. This protocol does not take advantage of multipath routing, and it reduces the transmission rate of each node instead of rerouting and bypassing the congested path.

B. Alternative path selection mechanisms

A new concept of routing protocol with congestion alleviation called Traffic-Aware Dynamic Routing (TADR) is proposed in [12]. TADR considers network traffic pattern as a “bowl” with sink residing at the bottom, and all data packets flow down just like water along the surface of the bowl. TADR uses combination of depth field force and queue length potential force to indicate which neighbor should forward next. Although TADR guides node to detour the congestion path, it has high chance to form one or more routing loops and increases the end-to-end delay.

In [13], the authors proposed a new scheme called Siphon. Siphon uses special virtual sinks distributed in the whole network, which have more powerful radio than normal sensor nodes. When congestion is detected, sensor nodes will forward packets to near virtual sink and the virtual sink will forward the packets to real physical sink via other radio network such as Wi-Fi. However, it needs another connected radio network which is infeasible in both low power and low cost consideration WSNs.

III. GAME THEORY BASED CONGESTION CONTROL PROTOCOL (GTCC)

We prefer to mitigate congestion via alternative path selection mechanisms and propose the congestion control protocol GTCC in RPL. When congestion is detected by net packet flow rate, which is packet generation rate subtracted by packet service rate, nodes in congestion region will execute parent-change procedure based on game theory strategy in order to find another better parent to improve network throughput.

A. Congestion in RPL

In RPL, the network topology is a Destination Orientated Direct Acyclic Graph (DODAG). Each node will emit DODAG Information Object (DIO) packet to all its neighbors to maintain the network connectivity. The DIO packets are controlled by a polite gossip policy [14], where each node periodically broadcasts a DIO packet to local neighbors but stays quiet if it has recently heard a DIO packet sent by itself. The DIO packet includes RPLInstanceID which is a unique identity of the network, rank field i.e. the sender’s rank, and the option field which is used to store optional information such as objective function of RPL. The objective function is used to calculate the rank of nodes. In our protocol, we use the first bit of rank field as Congestion Notification bit (CN bit) and we will store the sending node’s children information, including their IP addresses and sending rates, and the sending rates of the sender into the option field.

When a node receives a DIO packet, it will use the objective function and the rank of the DIO sender to calculate an expected rank. If the expected rank is smaller than its current rank, the node will consider changing its parent to the DIO sender.

For most applications in WSNs, traffic flow in a network is light for a long time until one of the predefined events occurs in the sensing region. When the source sensors begin to collect data, sensors in the region will start transmitting a large amount of packets. Once the packets number is large enough to form transient packet burst, it will possibly cause congestion on the path from source nodes towards the sink node. However, mitigating the congestion by reducing the rate of upstream node will violate fidelity level required by applications and decrease the throughput in RPL [12].

B. The Proposed Protocol

Our GTCC protocol will redirect the traffic flow to another path by parent-change procedure. In this procedure, nodes change their parents with maximum benefit such as fewer hop count, smaller buffer occupancy, or higher link quality. After that, the traffic flow will be scattered. It will improve the throughput of communication and reduce the packet loss rate. In GTCC, each node will keep to read the CN bit in DIO packet from parent. On the other hand, we will try to find an alternative path to scatter the traffic flow when congestion is detected.

In RPL network, DIO packets are used to maintain the network connectivity. Upon receiving a DIO packet, a node saves the sending rate and link quality into its neighbor table and checks whether the DIO sender is its parent. If the DIO packet is sent from its parent, the node will check the CN bit first. If the CN bit is clear, it means there is no congestion and the child node will calculate its rank as regular. Otherwise, the child node knows that the congestion occurs on its parent. Therefore, all child nodes associated with the parent will use game theory based strategy to determine their new parents. If congestion cannot be mitigated through the parent-change procedure, each node will notify its children through the DIO packet with CN bit set and the congestion information will be obtained by all reachable child nodes.

a) Congestion detection metrics

In recent congestion control protocols, several congestion metrics were proposed such as queue occupancy [12], channel loading [13], ratio of packet inter-arrival time to packet service time [10, 11]. We use the net packet flow rate which is packet generation rate r_{gen} subtracted by packet service rate r_{out} as metric for detecting the presence of congestion on parent node. We define the congestion metric α as below.

$$\alpha = r_{gen} - r_{out} \tag{1}$$

The value of α can be treated as the buffer occupation growing rate. If α is greater than 0, the probability of congestion is considerable. Conversely, if α is less than or equals to 0, the probability of congestion is low. In RPL standard, DIO packets can be used to disseminate the net packet flow rate. Furthermore, we can use DIO packets to inform the presence of congestion to all the neighbors by adding a CN bit.

b) Rank value of nodes

The most important part of our protocol is to decide the rank value of each node in a DODAG network. The rank value can directly influence the network topology and the performance of network because each node will select a parent to minimize its rank value. Many metrics and constraints were studied to calculate the rank of a node such as energy state, hop count, expected transmission count, delay, and throughput. In our protocol, we use the link quality (LQ) of a candidate parent and the rank of the candidate parent to calculate the rank of a node n.

$$rank(n) = \mathrm{RI} + LQ_p + rank_p \tag{2}$$

RI is a constant which represents a rank increasing between a node and its parent. The rank increasing is used to prevent routing loop and the value is varied by implementations. We set RI to 256 in our protocol (same as Contiki [14]). And $rank_p$ is the rank of the candidate parent p. LQ_p is the link quality from node n to p. Here, we use Link Quality Indicator (LQI) of each received packet to measure the link quality between two communication nodes. We note that, link quality distribution might not be homogeneous. Thus, when a node receives a packet from a sender, it will piggyback LQI of the received packet into its ACK message and send back to the sender. By choosing LQ as the metric of rank calculation, we can get LQI directly from the received packet. Also, LQI cannot only detect congestion but also detect node failure or other unavailable status of node. As a result, the usage of LQI can help node to bypass congested and low performance paths. Considering the link quality and rank are different metrics for rank computation, we use an LQ function to map the LQI to the scale of rank, LQ(LQI).

According to (2), a node's rank depends on the rank of its candidate parent and the LQ of the link from its candidate parent. Let LQI_{ij} denote the link quality indicator between nodes i and j. Considering a simple network topology of three nodes A, B and C, while the expected direction of packet flow is from A to C. Node A can directly be connected to nodes B and C, also, node B has a direct link to node C. In general, node A will choose node C as parent because of smaller hop count to sink than via B to C. However, if the LQI_{AC} is much lower than LQI_{AB} and LQI_{BC}, the two-hop communication from A to C via B may perform better than one-hop direct communication from A to C. So, a threshold of LQI is necessary to estimate the current performance of link quality in real network. In order to find the threshold value of LQI, we execute experiments in a testbed built by Octopus II sensor platform [15].

From the result of experiments, we find three special values, denoted by L^0, L^*, and L^f. We notice that different hardware platforms have different value of L^0, L^*, and L^f. when the LQI value is larger than L^0=140, the transmission time of 1000 packets (127 Bytes each) is about 15 seconds. However, the packet transmission time increases dramatically when LQI is less than L^0. Moreover, the transmission time is double as the LQI is equal to L^*=115. When LQI is smaller than L^f=100, the transmission time is unacceptable. Noting that, different hardware platforms have different value of L^0, L^*, and L^f. According to equation (2), in the example above, if node A wants to select node B as its parent, the following statement must hold:

$$\mathrm{RI} + \mathrm{LQ}(LQI_{AB}) + \mathrm{LQ}(LQI_{BC}) < \mathrm{LQ}(LQI_{AC}) \tag{3}$$

In order to prevent ping-pong effect, we add a hysteresis loop from $L^* - \delta$ to $L^* + \delta$ in LQ function to avoid the link quality oscillating around L^* as shown in Fig. 1. We set $\delta = 5$ in our experiments.

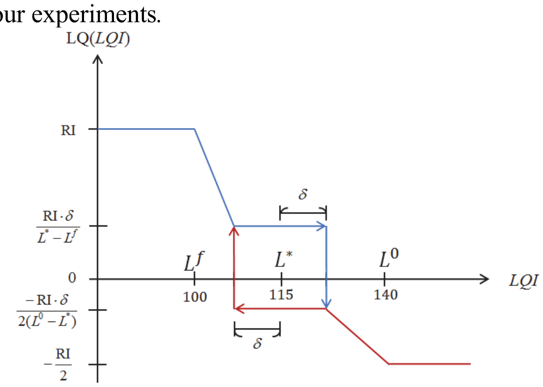

Figure 1. The function of LQ when LQI serves as variable

Therefore, we have equation (4).

$$\mathrm{LQ}(LQI) = \begin{cases} -\dfrac{\mathrm{RI}}{2}\ , & LQI \geq L^0 = 140 \\ \dfrac{-\mathrm{RI}}{2(L^0 - L^*)}(LQI - L^*), & L^* + \delta < LQI < L^0 \\ \dfrac{-\mathrm{RI}\cdot\delta}{2(L^0 - L^*)}, & L^* - \delta < LQI \leq L^* + \delta, \text{ for LQI is decreasing from } L^0 \\ \dfrac{\mathrm{RI}\cdot\delta}{L^* - L^f}, & L^* - \delta \leq LQI < L^* + \delta, \text{ for LQI is increasing from } L^f \\ \dfrac{-\mathrm{RI}}{L^* - L^f}(LQI - L^*), & L^f < LQI \leq L^* - \delta \\ \mathrm{RI}, & LQI \leq L^f = 100 \end{cases} \tag{4}$$

When $LQI \geq L^0$, we set LQ = –RI/2. This is because if there is a two-hop communication with good link quality, we can eliminate a rank increasing RI with the two good links. When LQI is between $L^* + \delta$ and L^0 or between L^f and $L^* - \delta$, we set LQ proportional to the transmission time of link quality as shown in Fig. 1. When LQI is below or equal to the value $L^f = 100$, we set LQ value to RI.

c) Parent-change procedure

When congestion is detected, each node in the congested area will start the parent-change procedure, although it does not mean that every node have to change their current parents. In parent-change procedure, each node uses the potential game theory method [8] to find a better parent to improve the network performance. Based on the potential game theory, we can converge to a stable state called Nash Equilibrium (NE) which is also the best parent allocation for whole nodes in the congested area. As we have discussed before, each node will select a parent which can minimize its rank. However, if too many nodes select the same parent, the load of this parent node will increase significantly to lead to even congestion. Thus, we can treat the behavior of parent selection as a game called *parent-selection game*. Parent-selection game is a game which each player (node) attends a competition of parent selection to minimize its rank. In this game, the action (a node makes its decision of parent selection) of each node will affect other node's utility (i.e. throughput). We will describe the mechanism of parent-selection game as follows.

Our parent-change procedure starts with a parent node that sends a congestion message to its children through the DIO packet. In potential game theory, we can reach NE by restricting only one node from changing its parent with minimal utility at a time. We use a random timer to randomly diffuse the time of change of parent by every node. When node i changes its parent, it will broadcast a new DIO packet to notify other children and increase the rank of old parent by RI to avoid changing back to the old parent. Noting that, we can adjust the timer according to new metrics instead of random timer, such as we can generate the shorter timer for nodes with higher transmission rates to reach NE faster.

d) Parent-selection game

In game theory, each player's action will affect other player's utility. The difficulty in finding an optimal action is that each player only knows about the utility of itself. Thus, nodes cannot know which selection is better for the global interest. Fortunately, as a subset of game theory, potential game theory can be used to deal with this problem. A game is a potential game if the incentive of all players to change their selection can be expressed using a single global function called the *potential function* [8]. With the aid of potential function, each node can determine whether a parent is worth changing or not by only considering its utility function. For the use of potential game, we will transform our parent-selection problem into a game representation and present the parent-change procedure. When congestion is detected by a node, it will set the CN bit in DIO packet, which contains the parent information, children list, corresponding transmission rate, and LQI, and forward to all its children. The children nodes receiving this message will consider changing parents according to the potential function. The potential function is built from information in neighbor table of each node, and the table will be updated upon the node received DIO messages from its neighbors. Each node can use this potential function to find a new parent which can decrease the value of this function in each round towards NE according to two properties of potential game [8]: Property (1) is that each ordinal potential game exists a NE. Property (2) is that if we limit only one node from changing its parent at a time, we can converge to NE.

We define the *parent-selection game* as $\Gamma = \langle N, A, u \rangle$, where N is the set of players, A is the set of actions and u is the utility function set. For each player n_i, we defined the following terms:

1) ***Player set*** *N:* The player set is defined as all children nodes of the DIO sender. We denote player set as $N = \{n_1, n_2, \ldots, n_m\}$. The set N contains m children.

2) ***Parent set*** *P:* The parent set of n_i is defined as all neighbors of player n_i whose ranks are less than n_i. We denote parent set as $P_i = \{p_1, p_2, \ldots, p_g\}$ if there are g parents for player n_i. $P = \cup_{1 \leq k \leq m} P_k$ is the union of parents of the m children, assuming $|P| = q$.

3) ***Action set*** *A:* The action set A is composed of any possible actions of each player. For player n_i, $A_i = \{a_i \mid a_i \in B^q$, there is only one 1 among a_i with the rest are 0$\}$ is used to represent the parent selection decision of n_i, where $B = \{0,1\}$. For instance, $a_i=(1,0,\ldots0)$ represents that node n_i chooses node p_1 as its parent. The action set of this game is defined as $A=\{(a_1, a_2, \ldots, a_m)^T \mid \forall a_i \in A_i\}$. Therefore, each element in A is an $m \times q$ matrix which shows the decision of every player in N. Thus, $a_{i,j} = 1$ means that player i selects parent j as parent in action a_i. Noting that if $p_j \notin P_i$, $a_{i,j}$ will be 0.

4) ***Utility function*** *u:* For player n_i, utility function is defined as $u_i : A \rightarrow \mathbb{R}$. The utility function is used to represent how much node n_i cost to reach sink node for action a_i. Thus, the smaller value of utility is the better one. We define utility function of player n_i with action a as,

$$u_i(a) = \mathrm{RI}+\mathrm{LQ}(LQI(p_k))+rank(p_k)+\sum_{1 \leq j \neq i \leq m,\ a_{j,k}=1} rate(n_j)+N_k \times rate(n_i) \quad (5)$$

where p_k is the parent candidate of player n_i ($a_{i,k} = 1$) and N_k is the number of children of parent p_k. The utility function of player n_i is composed of four terms: rank increase per hop, link quality between nodes n_i and p_k, rank of p_k and the sum of packet transmission rate of all children of p_k. In utility function (5), we consider the rank of candidate parent p_k and transmission rate of each child associating with p_k. Hence, the utility function is able to reflect the load of a candidate parent. Noting that, we multiply N_k to the transmission rate of node i ($rate(n_i)$) is to balance the number of children in each parent node. This is because selecting a parent with high N_k will increase the cost of utility function quickly. The LQ function and *rank* is defined in equation (4) and the transmission rate function is defined in equation (6).

$$rate(n_i)=R\times \mathrm{RI}/M, \quad (6)$$

where R is the packet delivery rate and M is the maximum packet delivery rate of each node. When a node reaches its maximum rate, it will not handle more data packet. So we let the rate function equals to RI when it reaches the maximum packet delivery rate. This will lead nodes to select other parents when the load of the candidate parent is satisfied.

A game Γ is an ordinal potential game if it admits an ordinal potential function. A function $\Phi : A \rightarrow \mathbb{R}$ is an ordinal potential for Γ if for every $i \in N$ and for every $a_{-i} \in A_{-i}$, where $A_{-i} = \{(a_1,a_2,..,a_{i-1},a_{i+1},\ldots,a_q) \mid \forall a_j \in A_j,\ 1 \le j \ne i \le q\}$ and the condition in equation (7) is satisfied, where $a_{-i} = (a_1,a_2,..,a_{i-1},a_{i+1},\ldots,a_q)$ is the action without player i (i.e. $a = (a_1, a_2, \ldots, a_q) = (a_i, a_{-i})$).

$$u_i(a_i', a_{-i}) - u_i(a_i, a_{-i}) < 0 \Leftrightarrow \Phi(a_i', a_{-i}) - \Phi(a_i, a_{-i}) < 0 \quad (7)$$

We define our ordinal potential function Φ as equation (8).

$$\Phi(a) = \sum_{j=1}^{m}\{N_k \times rate(n_j) + \mathrm{LQ}(LQI(p_k)) + rank(p_k)\}, \text{ if } a_{j,k} = 1 \quad (8)$$

where $a \in A$. The potential function is able to reflect the global interest in a network. In equation (8), the potential function contains the rank of each node's parent, link quality between parent and its children, and the packet rate of each child multiplied by the number of children in its parent.

We can prove that our parent-selection game is an ordinal potential game and it reaches NE by each node changing its parent with minimal utility at a time. However, due to the space limitations, we omit the description of proof in this paper.

IV. Performance Evaluation

We compare the performance of our proposed protocol to ContikiRPL with OF0 and OF-ETX denoted as CRPL-OF0 and CRPL-OF-ETX [14]. In CRPL-OF0 and CRPL-OF-ETX the ranks of each node are calculated using the metric of hop count and expected transmission times from source to sink, respectively. We evaluate the performance of the protocols on throughput, packet loss rate, and average hop count by Cooja simulation tool [16] in Contiki system. We evaluate the performance of our protocol by two scenarios. In scenario 1, we deploy the sink node in the center of the sensing area to collect data. Each sensing node has one or two parents. In scenario 2, we deploy the sink node in the top of the sensing area to evaluate the effect of sensing nodes with three or four parents. The area size is 220m×220m, simulation time is 200 seconds, radio range is 50m and the numbers of nodes are 26 and 22 in scenario 1 and 2, respectively. Our metrics for performance comparison are the average packet loss rate, average throughput, and average hop count. We simulate different transmission rate ranging from 2.5 to 18.2 packets per second for the above metrics. In our protocol GTCC, we set $L^0 = 110$, $L^*=91$ and $L^f=87$ according our rank calculation design. In the simulations, we can reach NE within 13 times of parent changing.

In Fig. 2, the packet loss rate of our protocol is less than 25% while the loss rates of CRPL-OF0 and CRPL-OF-ETX are both higher than 57% when the packet transmission rate of each node is equal to 10. When the transmission rate of node grows, the loss rates of CRPL-OF0 and CRPL-OF-ETX are significantly increasing. This is because nodes in CRPL-OF0 and CRPL-OF-ETX tend to select parent with fewer hop count to the sink. As the transmission rate grows, there are too many packets injected into the same candidate parent nodes than they can afford. On the other hand, nodes in our protocol can change their parents to keep the global load balance and avoid congested path, so that the lower loss rate are achieved.

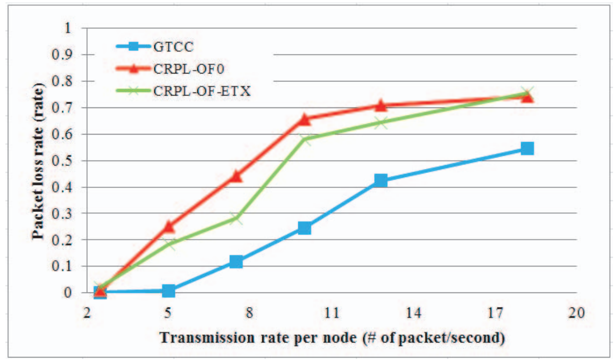

Figure 2. Packet loss rate vs. transmission rate in scenario 1

Fig. 3 shows the throughput with different packet transmission rates. When the transmission rate of each node grows, the throughput of our protocol is two times better than CRPL-OF0 and CRPL-OF-ETX. Moreover, the peak throughput of CRPL-OF0 and CRPL-OF-ETX is around 7.5 packets per second and the peak throughput can extend to 10 packets per second in our protocol.

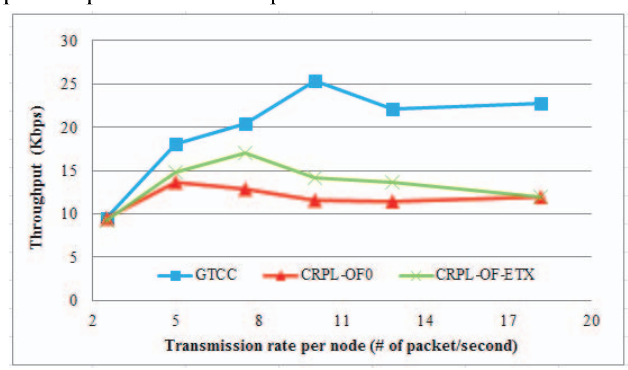

Figure 3. Throughput vs. transmission rate in scenario 1

In Fig. 4, it is shown that GTCC reduces packet loss rate by more than 20% compared to CRPL-OF0 and CRPL-OF-ETX with various transmission rates. For transmission rate being 5 packets per second, the packet loss rates of CRPL-OF0 and CRPL-OF-ETX are both near 50% while GTCC is 20% because the sensing nodes in ContikiRPL protocols are selecting nodes 15, 19 and 23 as parent nodes even there are other available candidate parents. In brief, our protocol takes the advantage of changing parent and mitigates the congestion. In Fig. 5, it is shown that the average throughputs of CRPL-OF0 and CRPL-OF-ETX are almost fixed as the packet transmission rate grows because of packet congestion. On the other hand, GTCC is 2.5 times better than CRPL-OF0 and CRPL-OF-ETX in scenario 2.

The average hop count to the sink in our protocol is 2.6 and 4.6 in scenarios 1 and 2, respectively. It is approximate to them in CRPL-OF0 and CRPL-OF-ETX, 2 and 4.3 in CRPL-OF0, 2.7 and 4.1 in CRPL-OF-ETX, respectively.

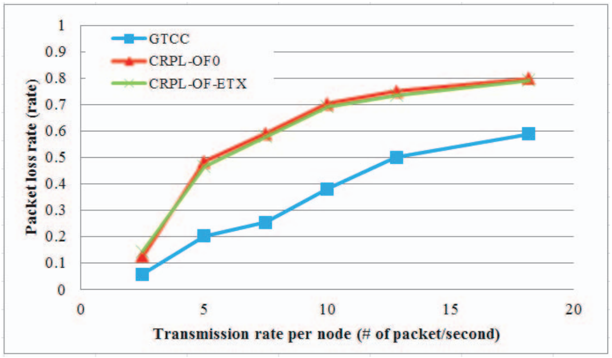

Figure 4. Packet loss rate vs. transmission rate in scenario 2

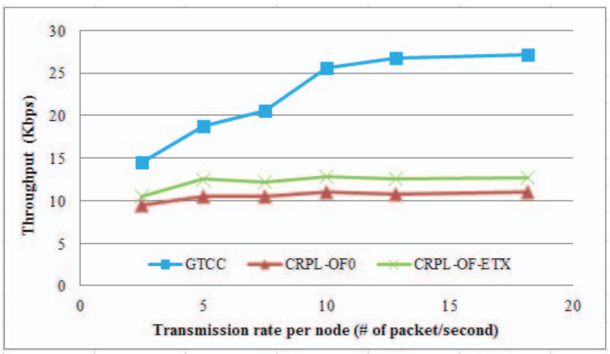

Figure 5. Throughput vs. transmission rate in scenario 2

V. CONCLUSIONS

In this paper, we proposed a novel congestion control protocol based on game theory over RPL to maximize the throughput. Our protocol exploits a parent selection scheme which can improve the throughput of communication. When congestion occurred, nodes will change their parents according to the utility function in game theory to avoid the congested path. We implement our protocol in Contiki OS and evaluate the performance via simulator Cooja. It is shown that our protocol has two times improvement in throughput and less packet loss rate compared to ContikiRPL protocols with approximate average hop count to sink node.

REFERENCES

[1] L. Atzori, A. Iera, and G. Morabito, "From Smart Objects to Social Objects: The Next Evolutionary Step of The Internet of Things." *IEEE Communications Magazine*, pp. 97-105, Jan. 2014.

[2] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, "Internet of Things (IoT): A vision, Architectural Elements, and Future Directions," *Future Generation Computer Systems*, Vol. 29, pp. 1645-1660, 2013.

[3] V. Michopoulos, L. Guan, G. Oikonomou, and I. Phillips, "A Comparative Study of Congestion Control Algorithms in IPv6 Wireless Sensor Networks", *Proceedings of International Conference on Distributed Computing in Sensor Systems and Workshops*, pp. 1-6, Barcelona, Spain, June 2011.

[4] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks", RFC 4944 (Proposed Standard), Internet Engineering Task Force, Sep. 2007. http://tools.ietf.org/html/rfc4944/.

[5] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL)", *RFC 6550 (Proposed Standard), Internet Engineering Task Force*, March 2012. http://tools.ietf.org/html/rfc6550/.

[6] M. A. Kafi, D. Djenouri, J. Ben-Othman, and N. Badache, "Congestion Control Protocols in Wireless Sensor Networks: A Survey," *IEEE Communications Surveys & Tutorials*, Vol. 16, No. 3, pp. 1369-1390, 2014.

[7] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "A High-Throughput Path Metric for Multi-hop Wireless Routing", *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pp. 134-146, San Diego, USA, Sep. 2003.

[8] D. Monderer and L. S. Shapley, "Potential games", *Games and Economics Behavior*, Vol. 14, No. 1, pp. 124-143, Jan. 1996.

[9] C. Basaran, K. D. Kang, and M. H. Suzer, "Hop-by-Hop Congestion Control and Load Balancing in Wireless Sensor Networks", *Proceedings of the 35th Annual IEEE Conference on Local Computer Networks*, pp. 448-455, Denver, USA, Oct. 2010.

[10] D. Patil and S. N. Dhage, "Priority-based Congestion Control Protocol (PCCP) for controlling upstream congestion in Wireless Sensor Network", *Proceedings of International Conference on Communication, Information & Computing Technology*, pp. 1-6, Mumbai, India, Oct. 2012.

[11] S. Sridevi, M. Usha, and G. P. A. Lithurin, "Priority Based Congestion Control for Heterogeneous Traffic in Multipath Wireless Sensor Networks", *Proceedings of IEEE International Conference on Computer Communication and Informatics*, pp. 1-5, Coimbatore, India, Jan. 2012.

[12] F. Ren, T. He, S. K. Das, and C. Lin, "Traffic-Aware Dynamic Routing to Alleviate Congestion in Wireless Sensor Networks", *IEEE Transactions on Parallel and Distributed Systems,* Vol. 22, No. 9, pp. 1585-1598, Sep. 2011.

[13] C. Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, "Siphon: Overload Traffic Management using MultiRadio Virtual Sinks in Sensor Networks", *Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, pp. 116-129, San Diego, USA, Nov. 2005.

[14] N. Tsiftes, J. Eriksson, and A. Dunkels, "Poster Abstract: Low-Power Wireless IPv6 Routing with ContikiRPL", *Proceedings of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, pp. 406-407, Stockholm, Sweden, April. 2010.

[15] J. P. Sheu, C. C. Chang, and W. S. Yang, "A Distributed Wireless Sensor Network Testbed with Energy Consumption Estimation", *International Journal of Ad Hoc and Ubiquitous Computing*, Vol. 6, No. 2, pp. 63-74, Jul. 2010.

[16] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-Level Sensor Network Simulation with COOJA", *Proceedings of the 31st IEEE Conference on Local Computer Networks*, pp. 641-648, Tampa, USA, Nov. 2006.