

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

Chuang Ma,¹ Jang-Ping Sheu,² and Chao-Xiang Hsu²

¹School of Computer and Information, Shanghai Second Polytechnic University, Shanghai 201209, China ²Department of Computer Science, National Tsing Hua University, Hsinchu 30013, Taiwan

Correspondence should be addressed to Chuang Ma; machuang@sspu.edu.cn

Received 21 April 2015; Revised 24 July 2015; Accepted 27 July 2015

Academic Editor: Jian-Nong Cao

Copyright © 2016 Chuang Ma et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In wireless sensor networks (WSNs), the presence of congestion increases the ratio of packet loss and energy consumption and reduces the network throughput. Particularly, this situation will be more complex in Internet of Things (IoT) environment, which is composed of thousands of heterogeneous nodes. RPL is an IPv6 routing protocol in low power and lossy networks standardized by IETF. However, the RPL can induce problems under network congestion, such as frequently parent changing and throughput degradation. In this paper, we address the congestion problem between parent nodes and child nodes in RPL-enabled networks, which typically consist of low power and resource constraint devices. To mitigate the effect of network congestion, we design a parent-change procedure by game theory strategy, by which the child nodes can change next hop neighbors toward the sink. Comparing to the ContikiRPL implementation, the simulation results show that our protocol can achieve more than two times improvement in throughput and reduce packet loss rate with less increasing of average hop count.

1. Introduction

In the past few years, we have witnessed the next big evolution of Internet of Things (IoT) in our life. IoT is used to connect industrial devices, hospital instruments, and household appliances such as refrigerators, air conditions, and TV sets with human applications. With IoT connected, people can control and receive messages from household appliances everywhere by their smart devices such as laptop computers, smart TV sets, and even mobile phones. These systems usually include many end devices conforming the IEEE 802.15.4 standard and they are often characterized by short transmission range, low data rate, low cost, and low communication power. In order to achieve connectable and accessible features of these devices, the Internet protocol IPv6 is proposed and serves as the best solution because of its popularization, applicability, sufficient address space, supporting for stateless address autoconfiguration, and easiness to access [1].

However, there are some difficulties to implement IPv6 in the wireless personal area networks. For instance, IEEE 802.15.4 uses only 127 bytes as maximum transmission size and IPv6 need 1280 bytes as the maximum transmission

size. Thus, the Internet Engineer Task Force (IETF) working group has standardized an adaptive layer used in the devices with IEEE 802.15.4 MAC/PHY called IPv6 over low power wireless personal network (6LoWPAN) [2]. 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) [3]. In low-power and lossy networks (LLNs), the routers are constrained in limited processing power, memory, and energy. Thus, the interconnections between nodes are normally suffered by low data rates, high loss rates, and long delay time. As routing protocol based on LLNs, RPL is a tree-like topology routing protocol supporting multipoint-to-point (nodes to sink), point-topoint, and point-to-multipoint (sink to nodes) 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 nodes, all the nodes in the event region will send packets to sink and may cause network congestion toward sink node. However, the congestion control is more difficult to achieve than other types of network because of the features of lower

power and lower computing abilities in LLNs. In WSNs, congestion control [4] contains congestion detection and congestion avoidance under limited resource environment. In general, congestion detection adopts some metrics such as buffer occupancy, channel loading, and the ratio of packet interarrival time to packet service time to detect the presence of congestion. When congestion is detected, we will use the congestion avoidance mechanisms to mitigate the presence of congestion.

Currently, there is no explicit mechanism to detect or 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 [5]. However, these schemes may result in ping-pong effect which refers to the phenomenon of repeated state switching from one to the other. If ping-pong effect occurs on a node, it will change its parent frequently. For example, a node will select its new parent node when it detects the congestion that occurs on the path including its old parent node. However, when congestion occurs on its new parent node later, even just after the node changes its parent, it must select its old parent node again if there is no available parent in its neighbors. As a result, the node will change its parent very frequently so that it cannot work well for relay and transmission. Even, ping-pong effect will significantly lead to packet loss, decreasing throughput, and increasing packet delay.

In this paper, we mainly focus on tree topology for two reasons. Firstly, we address the congestion problem in RPL-enabled networks routing protocol which are treelike topology routing protocol which can be used in the applications of WSNs. In RPL pattern, if an event occurs in the leaf nodes, all the nodes in the event region will send packets to sink and may cause network congestion toward sink node. Secondly, the congestion control in tree topology is more difficult to achieve than other types of network, so that there are no researches with congestion mechanism in RPL and also they do not aim at tree-like network topology. We propose a Game Theory Based Congestion Control Protocol (GTCC) to alleviate the effect of congestion. In our approach, nodes will be informed about the presence of congestion by their parents through control messages. In this case, a child node will decide whether it changes its parent or not based on the game theory strategy [6]. Thus, our scheme can alleviate congestion by changing parent with light load and avoid the ping-pong effect. We evaluate our protocol with Cooja simulator [7]. Comparing with ContikiRPL implementation [8], our simulations show that GTCC can achieve more than two times improvement in reduction of packet loss rate and increasing of throughput with less average hop count to the sink.

The rest of this paper is organized as follows. Section 2 introduces related work. The detail of our protocol is presented in Section 3. The simulation results are described in Section 4, and Section 5 concludes this paper.

2. Related Work

In order to mitigate and avoid the presence of congestion, there are some proposed schemes aimed at congestion control in traditional WSNs. In [9], the authors reviewed various existing techniques for detecting and controlling congestion in WSNs. The research hotspots and difficulties in this area which need to be further studied were also pointed out in this paper. Depending on the control policy, the congestions control protocols are divided into resource control and traffic control. Resource control protocols are classified according to the type of resource to be tuned. In traffic control mechanisms, the protocols are divided into reactive and preventive (avoiding). Reactive solutions are classified as the reaction scale, while preventive solutions are split up into buffer limitation and interference control.

In our work, we divide previous congestion control schemes into two categories in traffic control: rate adjustment schemes and alternative path selection schemes. In rate adjustment mechanisms, 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 mechanisms, if a node detects congestion, it would try to find another better path to transmit packets. We will review both kinds of mechanisms in this section.

2.1. Rate Adjustment Mechanisms. In COngestion Detection and Avoidance (CODA) protocol [10], the congestion can be detected by queue length and wireless channel load. When congestion is detected, all the source nodes whose source event rates exceed a threshold need to wait for the acknowledge message from sink node after sending the sensing data. If a source node cannot receive acknowledge message from sink, it will reduce its transmission rate. In [1], 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 congestion degree 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 [8] proposed a Priority-Based Congestion Control Protocol (PCCP). PCCP uses ratio of packet interarrival time to packet service time as 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. Also, in [11], the authors proposed a bioinspired congestion control approach for WSNs streaming applications that necessitate controlled performance with graceful degradation. A deterministic population balance equation inspired from biological systems is used as a throughput formula to adjust the sending rate of sensor nodes and to achieve adaptability to changing traffic loads, scalability, and fairness among flows.

In [12], the authors presented HRTC, a hybrid scheme for congestion control in WSNs. This scheme attempts to complement the resource control method with traffic control. The advantage of this hybrid solution lies on the fact that, due to the frequent variations that take place in WSNs topologies and node placements, each node is able to figure out which congestion control method is the most appropriate to apply at any moment, giving priority to resource control that extends network lifetime as well as throughput.

DDR [13] uses buffer occupancy as congestion metric and specifies a high threshold value for this metric. If the buffer occupancy is greater than the high threshold, the congestion notification bit will be set to one and it will use queuing theory to adjust the new transmission rate. Moreover, DDR calculates the hop count between each flow from source to sink and measures the reporting rate. It assigns low reporting rate to nodes which are located closer to sink.

In [14], the authors addressed the rate control and resource allocation problem for heterogeneous wireless sensor networks, which consist of diverse node types or modalities such as sensors and actuators and different tasks or applications. Each node calculates the source rate based on the aggregate price of capacity and energy along its path to the sink and constantly updates the data rate based on both the capacity and energy of the passing flows as well as from a feedback from the sink.

2.2. Alternative Path Selection Mechanisms. In [15], 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 networks such as Wi-Fi. However, it needs another connected radio network which is infeasible in both low power and low cost consideration WSNs.

A new concept of routing protocol with congestion alleviation called Traffic-Aware Dynamic Routing (TADR) is proposed in [16]. 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 be the next forwarder. Although TADR guides node to detour the congestion path, it has big chance to form one or more routing loops and increases the end-to-end delay.

LACAR [17] is a location aided congestion aware routing protocol which provides a proactive way to avoid congestion. LACAR uses buffer occupancy and average number of transmission attempts per successful transmission to detect congestion. It uses knowledge of each neighbor and sink node combined with congestion metric to route a path with less congestion and energy consumption. However, LACAR uses the knowledge of nodes' location which is normally unavailable in WSNs. In [18], cross-layer congestion control optimal algorithms are presented for congestions occurred simultaneously at node-level and link-level in WSNs. Contention channels were reasonably allocated through the maximum spanning tree searching. The algorithms guaranteed data effective transmission, heavily highlight throughput, and maintain the energy consumption at low communication overhead.

Game theory can be used in researches to deal with the problem of congestion in wireless networks. In [19], the authors formulated the problem of congestion control using coalition game theory and propose a centralized assignment algorithm to dynamically assign Cell Access Nodes (CANs) to Access Points (APs) in wireless network and can achieve airtime-balancing Nash equilibrium. Also, they proved that the assignment algorithm terminates in a stable partition which attains optimal grand aggregate utility for the network. In [20], the authors presented a game theoretic model at the power control MAC layer of IEEE 802.11p for wireless transmission in network. For selfish and nonselfish nodes of unknown type, a Bayesian noncooperative game was formulated to model the decision making process of nodes to broadcast a beacon packet for the radio bandwidth in an incomplete information environment. They considered the proposed strategy as Bayesian Nash equilibrium, where the nodes transmit a beacon packet according to a channel gain based threshold based on power control.

3. Game Theory Based Congestion Control Protocol (GTCC)

In this paper, we prefer to mitigate congestion via alternative path selection mechanisms for the following reasons. The first one is that, in rate adjustment scheme, dropping packets are a common way to notify the source to reduce their packets generation rate. But in RPL, packets are usually fragmented so that dropping packets will cause entire packet retransmission. Moreover, the nodes in RPL networks usually have limited resources such as buffer size, which may degrade the performance of congestion control based on rate adjustment mechanism because the buffer of each node will be full quickly as transient packet burst. The second reason is for the consideration of maximizing the throughput and utilization. If we find another path with enough bandwidth toward sink, we can keep original transmission rate and get more throughput and utilization from alternative path scheme.

We 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. To explain our protocol more explicitly, we first describe how RPL works. After that, we propose our congestion control protocol.

3.1. 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 [3], 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 which is 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



FIGURE 1: An alternative path selection example.

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 example, we consider an RPL topology in Figure 1(a). We assume that the objective in this network is to minimize the hop count to the DODAG root. Thus, the rank of a node indicates the hop count between root node (indicated by *R*) and the node. When node *B* receives DIO packets from nodes A, D, and E, their corresponding hop counts (ranks) required to root R are 3, 2, and 3, respectively. Thus, node B will select node D as its parent to get the minimal rank. 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 toward the sink nodes. However, mitigating the congestion by reducing the rate of upstream node will violate fidelity level required by applications and decrease the throughput in RPL [16].

Our 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 less hop count, less buffer occupancy, or higher link quality. After that, the traffic flow will be scattered, this will improve the throughput of communication and reduce the packet dropping rate. For example in Figure 1(a), if node *A* starts a long-term packet burst transmission, node *D* will become a bottleneck for nodes *B* and *C*. In this case, nodes *B* and *C* will change their parents to nodes *E* and *F*, respectively, which can deliver their packets more quickly than node *D* as shown in Figure 1(b).

3.2. 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 continue reading 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. Thus, our protocol combines advantages of both alternative path selection scheme and rate adjustment scheme.

In RPL network, DIO packets are used to maintain the network connectivity. Upon receiving a DIO packet, a node will save the sending rate and link quality into its neighbor table and check if 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 that 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 children 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 broadcasted to all leaf nodes.

3.2.1. Congestion Detection Metrics. In recent congestion control protocols, several congestion detection metrics were proposed such as using dual buffer thresholds and weighted buffer difference for congestion detection [21], congestion detection by evaluation of channel loading [15], and computing different congestion levels based on the queue length evolution [22]. To evaluate the congestion in tree topology WSNs, a metric must be introduced to describe the occurrence and degree level of congestions so that 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 follows:

$$\alpha = r_{\rm gen} - r_{\rm out}.$$
 (1)

In (1), the congestion metric α is defined as the difference of packet generation rate r_{gen} and packet service rate r_{out} . 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 0, the probability of congestion is low. In RPL standard [3], DIO packets can be used to disseminate the net packet flow rate. Furthermore, we assume that the period of DIO packet is small enough for information exchange in congestion control and we can use DIO packets to inform the presence of congestion to all the neighbors by adding a CN bit.

3.2.2. 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 such as the rank of a node network.

$$\operatorname{rank}(n) = \operatorname{RI} + \operatorname{LQ}_p + \operatorname{rank}_p.$$
(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 [3] and the value is varied by implementations. We set RI to 256 in our protocol. And rank, is the rank of the candidate parent p. LQ_p is the link quality from node n to node 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 the packet back to the sender. By choosing LQ as the metric of rank calculation, we can get LQI directly from the received packet, which is computed in MAC layer by Received Signal Strength Indication (RSSI). 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 which 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_{ii} denote the link quality indicator between nodes i and j. Considering a simple network topology of three nodes A, B, and C in Figure 2, the expected direction of packet flow is from A to C. Node A can directly connected to nodes B and C, and also node B has a direct link to node C. In general, node A will choose node C as its parent because of less hop count to sink than via B and C to sink. However, if the LQIAC is much lower than LQIAB and LQIBC, 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 [23]. The experiment result is shown in Figure 3.

From the result of experiments, when the LQI value is larger than $L^0 = 140$, the transmission time of 1000 packets (127 Bytes/packet) 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. We notice that different hardware platforms have different value of L^0 , L^* , and L^f . According to (2), in the example above, if node A wants to select node B as its parent, the following equations must be hold:

$$RI + LQ (LQI_{AB}) + RI + LQ (LQI_{BC}) + rank_{c}$$

$$< RI + LQ (LQI_{AC}) + rank_{c}.$$
(3)

Thus, we have

$$RI + LQ (LQI_{AB}) + LQ (LQI_{BC}) < LQ (LQI_{AC}).$$
(4)

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 oscillates around L^* as shown in Figure 4. We set $\delta =$ 5 in our experiments.

Therefore, we have

$$LQ (LQI) = \begin{cases} -\frac{RI}{2}, & LQI \ge L^{0} = 140 \\ \frac{-RI}{2(L^{0} - L^{*})} (LQI - L^{*}), & L^{*} + \delta < LQI < L^{0} \\ \frac{-RI \cdot \delta}{2(L^{0} - L^{*})}, & L^{*} - \delta < LQI \le L^{*} + \delta, \text{ for LQI is decreasing from } L^{0} \\ \frac{RI \cdot \delta}{L^{*} - L^{f}}, & L^{*} - \delta \le LQI < L^{*} + \delta, \text{ for LQI is increasing from } L^{f} \\ \frac{-RI}{L^{*} - L^{f}} (LQI - L^{*}), & L^{f} < LQI \le L^{*} - \delta \\ RI, & LQI \le L^{f} = 100. \end{cases}$$
(5)



FIGURE 2: An example of one-hop and two-hop communications from node *A* to node *C*.



FIGURE 3: The transmission time of 127 kilobytes data with LQI.

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 value proportional to the transmission time of link quality as shown in Figure 4. When LQI is below or equal to the value $L^f = 100$, we set LQ value to RI. Thus, if the LQI_{AC} is less than L^* , node A can select node B as its parent as long as LQI_{BC} and LQI_{AB} are greater than or equal to L^0 .

3.2.3. Parent-Change Procedure. In the above subsection, we have described our congestion detection method and rank calculation in RPL. When congestion is detected, each node in the congested area will start the parent-change procedure, although it does not mean that every node has to change their current parents. In parent-change procedure, each node uses the potential game theory method [6] 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 in this parent node will significantly increase to lead to congestion also. Thus, we can treat the behavior of parent selection as a game called *parent-selection* game. Parent-selection game is a game such that 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). For example, in Figure 1(a), it is shown that nodes A, B, and C are competing for parents D, E, and F. If every node only considers the state of itself, they will select a same parent node *D* because it has the least hop to the sink if all links have same link quality. However, if nodes A, B, and C have lots of data packets for transmission, node D is



FIGURE 4: The function of LQ when LQI serves as variable.

not capable of dealing with the large amount of packets and the congestion is inevitable. Thus, we will try to prevent this condition and maximize the network throughput by game theory based method. In the following, we will describe the mechanism of parent-selection game.

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 that can change its parent with minimal utility in a time. We use a random timer to randomly diffuse the timing of changing parent of 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. We note that we can adjust the timer according to new metrics instead of random timer for better global effect because any changes on each node will affect the utility function, such that we can generate the shorter timer for nodes with higher transmission rates to reach NE faster.

3.2.4. 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* [6]. 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 it 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 which 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 toward NE according to two properties of potential game [6]. Property (1) is that the NE exists in each ordinal potential game. Property (2) is that if we limit only one node that can change 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 $N = \{n_1, n_2, ..., n_m\}$. 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 $P_i = \{p_1, p_2, \dots, p_g\}$ if there are g parents for player n_i . $P = \bigcup_{1 \le k \le m} P_k$ 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, $a_i = (1, 0, ..., 0)$ represents that node n_i chooses node p_1 as its parent. The action set of this game is defined as $A = \{(a_1, a_2, ..., a_m)^T \mid \forall a_i \in A_i\}$. Therefore, each element in A is $m \times q$ 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 does node n_i cost to reach sink node for action a_i . We use the cost to describe the utility because the cost to reach sink node is most important fact for transmission of relay nodes on one hand, not only for single node but also for network as a whole. We can get better transmission performance by decreasing the cost of transmission. On the other hand, we can use the cost of a node to sink node to evaluate the benefit of a node in networks by game theory. For example, if a node selects it parent by evaluation of cost, it will affect the selection of other nodes by game theory. Thus, the smaller the value of utility is, the lower the cost and the higher the efficiency the transmission is. We define utility function of player n_i with action a as

$$u_{i}(a) = \operatorname{RI} + \operatorname{LQ}\left(\operatorname{LQI}\left(p_{k}\right)\right) + \operatorname{rank}\left(p_{k}\right) + \sum_{1 \leq j \neq i \leq m, a_{i,k}=1} \operatorname{rate}\left(n_{j}\right) + N_{k} \times \operatorname{rate}\left(n_{i}\right),$$
⁽⁶⁾

where p_k is the parent candidate of player n_i ($a_{i,k} = 1$), LQI(p_k) is the link quality between nodes n_i and p_k , 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



FIGURE 5: An example topology of parent-selection game.

of packet transmission rate of all children of p_k . In utility function (6), 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 is defined in (5) and the transmission rate function is defined in

rate
$$(n_i) = R \frac{\mathrm{RI}}{M},$$
 (7)

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 be able to handle more data packet. Thus, we let the rate function be equal 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.

An example is shown in Figure 5. For node *B*, the player set is $N = \{A, B, C, D\}$, $P = \{E, F\}$, and $a_B = \{(1, 0), (0, 1)\}$, where (1, 0) and (0, 1) represent that node *B* can selects nodes *E* and *F* as parents, respectively. We assume that the maximum packet rate of each node is 20, the packet rates of nodes *A*, *B*, *C*, and *D* are 5, 5, 2, and 3, respectively, LQ value of link *B* to *F* is -128, and rank of node *F* is 200. The utility of node *B* when it selects node *F* as parent is $256+(-128)+200+(5+2+3+4\times5)\times 256/20=712$, according to (6) and (7).

A game Γ is an ordinal potential game if it admits an ordinal potential function. A function $\Phi : A \to \mathbb{R}$ is an ordinal potential for Γ if for every $i \in N$ and for every $a_{-i} \in A_{-i}$, where $A_{-i} = \{(a_1, a_2, \dots, a_{i-1}, a_{i+1}, \dots, a_q) \mid \forall a_j \in A_j, 1 \leq j \neq i \leq q\}$ and the condition in (8) is satisfied, where a_{-i} is the action without player i (i.e., $a = (a_1, a_2, \dots, a_q) = (a_i, a_{-i})$):

$$u_{i}\left(a_{i}^{\prime},a_{-i}\right)-u_{i}\left(a_{i},a_{-i}\right)<0$$

$$\longleftrightarrow\Phi\left(a_{i}^{\prime},a_{-i}\right)-\Phi\left(a_{i},a_{-i}\right)<0.$$
(8)

We define our ordinal potential function Φ as the following equation and prove that it will satisfy (8) in Theorem 1:

 $\Phi(a)$

$$= \sum_{j=1}^{m} \left\{ N_k \times \operatorname{rate}(n_j) + \operatorname{LQ}(\operatorname{LQI}(p_k)) + \operatorname{rank}(p_k) \right\}, \quad (9)$$

if $a_{j,k} = 1$,

where $a \in A$. The potential function is able to reflect the global interest in a network. In (9), 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.

Theorem 1. *The parent-selection game is an ordinal potential game with ordinal potential function* Φ *.*

Proof. We need to prove that Φ is a potential function in parent-selection game which satisfies (8). Assuming that a player n_i changes its parent to $p_{l'}$ from p_l , it makes the action changes to a' from a. The difference of utility function is

$$u_{i}(a') - u_{i}(a) = LQ(LQI(p_{l'})) - LQ(LQI(p_{l}))$$

$$+ rank(p_{l'}) - rank(p_{l})$$

$$+ \sum_{1 \le j \ne i \le m, a'_{j,l'}=1} rate(n_{j})$$

$$- \sum_{1 \le j \ne i \le m, a_{j,l}=1} rate(n_{j}) + N'_{l'}$$

$$\times rate(n_{i}) - N_{l} \times rate(n_{i}).$$
(10)

Then the difference of potential function is

/ n

$$\Phi(a') - \Phi(a) = LQ(LQI(p_{l'})) - LQ(LQI(p_{l}))$$

$$+ \operatorname{rank}(p_{l'}) - \operatorname{rank}(p_{l})$$

$$+ \sum_{j=1}^{m} N'_{k} \times \operatorname{rate}(n_{j})$$

$$- \sum_{j=1}^{m} N_{k} \times \operatorname{rate}(n_{j}),$$
(11)

where
$$a_{j,k} = 1$$
, for each *j*.

Let $C = LQ(LQI(p_{l'})) - LQ(LQI(p_l)) + rank(p_{l'}) - rank(p_l)$. Consider

$$\Phi(a') - \Phi(a) = C + \sum_{j=1}^{m} N'_{k} \times \operatorname{rate}(n_{j})$$
$$- \sum_{j=1}^{m} N_{k} \times \operatorname{rate}(n_{j})$$

$$= C + \sum_{j=1, a'_{j,j}=1}^{m} N'_{l} \times \operatorname{rate}(n_{j})$$
$$+ \sum_{j=1, a'_{j,j'}=1}^{m} N'_{l} \times \operatorname{rate}(n_{j})$$
$$- \sum_{j=1, a'_{j,j'}=1}^{m} N_{l} \times \operatorname{rate}(n_{j})$$
$$- \sum_{j=1, a'_{j,j'}=1}^{m} N_{l'} \times \operatorname{rate}(n_{j}).$$
(12)

Because only node *i* changed its parent from p_l to $p_{l'}$, we can only consider the columns *l* and *l'* in *a* and *a'*, respectively. Thus, the number of children of p_l in *a* is one less than that in action *a'* and the number of children of $p_{l'}$ in *a* is one more than that in action *a'*. Thus, we have $N'_l = N_l - 1$ and $N'_{l'} = N_{l'} + 1$, and (12) can be deduced as follows:

$$C + \sum_{1 \le j \ne i \le m, a'_{j,i'} = 1} \operatorname{rate}(n_j) + N'_{l'} \times \operatorname{rate}(n_i)$$
$$- \sum_{1 \le j \ne i \le m, a_{j,i} = 1} \operatorname{rate}(n_j) - N_l \times \operatorname{rate}(n_i) \qquad (13)$$
$$= u_i(a') - u_i(a).$$
So $\Phi(a') - \Phi(a) = u_i(a') - u_i(a).$

Theorem 1 shows that our parent-selection game is an ordinal potential game. Thus, we can reach NE by each node changing its parent with minimal utility in a time.

We will give an example of how the parent-selection game works. Figure 6(a) shows the network topology including rank (numbers besides node ID), link (arrows between two nodes), and packet rate (numbers in the block) of each node before the congestion happened. We assume the link quality of all links is larger than L^0 which means that the RI+LQ(LQI) is a constant of 128. We assume some events detected by node 7 and the packet rate of node 7 increases to 20 as shown in Figure 6(b). Assume the maximum packet rate of each node is 20. The net packet flow rate α of node 3 will be equal to 10 (5 + 20 + 5 - 20) and the congestion is detected.

Thus, node 3 emits DIO with CN bit set. When the children of node 3 receive the DIO packet, they will start changing parent with random timer. We assume that the generated timer for nodes 6, 7, and 8 is 3, 5, and 8, respectively. Node 6 will change its parent first. The original utility of node 6 is 768, $(128 + \operatorname{rank}(n_3) + 3 \times \operatorname{rate}(n_6) + \operatorname{rate}(n_7) + \operatorname{rate}(n_8))$. To minimize the utility of node 6, we will change its parent to node 5 and decrease its utility to 448. After that, node 6 will emit a new DIO to notify its neighbor. Then the utility of node 7 is decreasing to 832 from 1152. Node 7 will select node 9 as its parent which will decrease its utility to 640. After node 7 changed its parent to node 9, node 8 will not change its parent



FIGURE 6: An example of parent-change game with 9 nodes.



FIGURE 7: The final topology of Figure 6 after parent-change procedure.

because it is already in NE and congestion is alleviated. The final topology is shown in Figure 7.

3.2.5. Rate-Reduction Mechanism. When congestion occurs, the congested parents will use DIO packet with CN bit set to notify their children to invoke the parent-change procedure. As every child is already in NE, the congestion will be alleviated. If congestion is not mitigated after the parent-change scheme, the children of the congested parent will start the procedure of rate-reduction to reduce the packet rate. We illustrate the packet flow in Figure 8.

The packet generation rate of a node is composed of the packet rate $r_{\rm src}$ of itself, which the current node serves as a source node and the packet rate $r_{\rm rly}$ from its children, which the current node serves as a relay node. In rate-reduction procedure, a child reduces its packet rate multiplied by the value of reduction rate $\beta = (r_{\rm out} - r_{\rm src})/r_{\rm rly}$ where $r_{\rm out}$ is the packet output rate. If there is no congestion occurs, the output packet is equal to the sum of source packet and the relay packet, in other words, $\beta = 1$. However, in most congestion scenarios, the output packets of a relay node come from relay



FIGURE 8: An illustration of packet flow inside a node.

transmission, so that the transmission packets of output are approximately equal to the relay packets that means $0 < \beta < 1$. In the rate-reduction scheme, we want to reserve the bandwidth of the packet rate $r_{\rm src}$ and reduce the packet rate $r_{\rm rly}$. Thus, we can limit the total packet rate of children to $r_{\rm out}$ - $r_{\rm src}$.

We can give an example by Figure 5. Assume the packet rates of nodes *A*, *B*, *C*, and *D* are all 5 and r_{src} and r_{out} of node *F* are 4 and 20, respectively. We can calculate the reduction rate β as (20-4)/(5+5+5+5) = 0.8 After the rate reduction computation, the packet rates of nodes *A*, *B*, *C*, and *D* are reduced to 4. So we can keep the packet rate r_{src} of node *F*.

4. Performance Evaluation

In the related works, there are no researches with congestion mechanism in RPL and also they do not aim at tree-like network topology. Thus, we compare the performance of our proposed protocol to ContikiRPL with OF0 and OF-ETX denoted as CRPL-OF0 and CRPL-OF-ETX [8]. The rank of each node in CRPL-OF0 and CRPL-OF-ETX is calculated by the metric of hop count and expected transmission times from source to sink, respectively. We evaluate the performance of each protocol on throughput, packet loss rate, and average hop count by Cooja simulation tool [7] in Contiki system. Cooja is a cross-level sensor simulator. Unlike NS-2 and other simulators, a node in Cooja is actually compiled and executing in Contiki. Thus, we can actually simulate every instruction in every node of our implementation.



FIGURE 9: Packet loss rate in different standard derivation of LQ.

4.1. Simulations without Parent Change Procedure. Nodes in simulation are categorized into three types: sink node, relay nodes, and sensing nodes. Sink node is the root node in RPL which collects sensing data, relay nodes are used to relay data to the sink node, and sensing nodes are responsible to sensing the environmental data within its sensing area. To evaluate our proposed protocol more explicit and intuitionistic, especially in our rate-reduction mechanism, we assume that the relay nods relay transmission only but do not generate sensing data. There is no parent changing mechanism in RPL, so we first evaluate the performance of our protocol without parent changing comparing to CRPL-OF0 and CRPL-OF-ETX. In this scenario, we put the sink node in the center of $150 \text{ m} \times 150 \text{ m}$ area and randomly deploy nine relay nodes and ten sensing nodes. We send two packets per second with packet size of 100 bytes. In our protocol GTCC, we set $L^0 = 110$, $L^* = 91$, and $L^f = 87$ according to our rank calculation design. We evaluate our protocol, CRPL-OF0 and CRPL-OF-ETX, by packet loss rate under different standard derivation. The result is shown in Figure 9.

The packet loss rate of CRPL-OF0 increases continuously when the standard derivation of link qualities grows. This is because the large standard derivation of link qualities will lead the packet loss rate of direct link to sink to increase. CRPL-OF0 tends to select the least hop count to the sink so its packet loss rate is greater than others. The packet loss rates in CRPL-OF-ETX and GTCC are both under 0.3 for all standard derivation of link qualities. This is because they try to find another path when link quality is low, but our rank calculation scheme can avoid adding a parent with low quality link whose link quality is less than L^f , by detouring to another parent with link quality greater than L^0 . Thus, our protocol outperforms CRPL-OF0 and CRPL-OF-ETX.

4.2. Simulation with Parent Change Procedure. We evaluate the performance of our protocol by two scenarios in simulations with parent change procedure and their nodes deployments are shown in Figure 10. 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. We use Unit Disk Graph Medium (UDGM) with distance lose module as radio interference. The success rate of transmission in UDGM module decreases as distance increasing. We set our packet size to 100 bytes to avoid packet fragmentation. This is because there is no retransmission mechanism in Contiki and it will cause entire packet retransmission even if only one fragment is lost. The area size is $220 \text{ m} \times 220 \text{ m}$, simulation time is 200 seconds, radio range is 50 m, and the numbers of nodes are 26 and 22 in scenarios 1 and 2, respectively.

Our metrics for performance comparison are the average packet loss rate, average throughput, and average hop count. The average packet loss rate is the ratio of total number of data packets lost to total number of data sent from sensing nodes. To measure the average packet loss rate, we simulate different transmission rates ranged from 2.5 to 18.2 packets per second for the above metrics. In order to evaluate the maximum throughput, we evaluate GTCC with or without Rate Reduction (RR) scheme in our simulations. Moreover, for fairly evaluating the performance of GTCC and ContikiRPL protocols, we set the link quality of all links to values above L^0 in the following simulation. In the following simulations, we can reach NE by no more than 13 times of parent changing.

In Figure 11, 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 rates are achieved. In addition, we enable the rate reduction mechanism to restrict the transmission rate of each node to 7.5 packets per second that will reduce the packet loss rate to below 20%.

Figure 12 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 indicates the maximum packet flow that network can carry. The peak throughput can extend to 10 packets per second in our protocol. Also, the throughput of GTCC with RR is lower than GTCC without RR.

In Figure 10 of scenario 2, sensing nodes (7–10 and 16– 18) are located in the bottom of the sensing area. There are seven nodes (11, 13, 15, 19, and 21–23) that can be chosen as the parents of the sensing nodes. Since nodes 15, 19, and 23 have the less hop count to sink, this will make nodes 15, 19, and 23 become hotspots and congestion will occur in high probability in ContikiRPL protocols. However, our protocol



FIGURE 10: The nodes deployment in scenarios 1 (a) and 2 (b).



30 25 Throughput (kbps) 20 15 10 5 0 2 5 8 11 14 17 20 Transmission rate per node (number of packets/s) GTCC - GTCC with RR - CRPL-OF0

FIGURE 11: Packet loss rate versus transmission rate (scenario 1).

FIGURE 12: Throughput versus packet transmission rate (scenario 1).

can avoid the congested path and outperforms CRPL-OF0 and CRPL-OF-ETX.

In Figure 13, it is shown that GTCC reduces packet loss rate 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. Thus, we can notice that GTCC can take advantage of changing parent and mitigate the congestion. Furthermore, GTCC with RR scheme can restrict the transmission rate to 7.5 packets per second which have lower packet loss rate than other protocols.

In Figure 14, 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. Although GTCC with RR scheme can reduce packet loss rate, it degrades the throughput to some extent too. Also, we can notice that the throughput of GTCC with RR is lowest although the gap is small enough, because, in the scenario 2, the sink is in the most edge far away from center of WSN, in other words, it is a most ultimate condition of implementation. By contrast,



FIGURE 13: Packet loss rate versus transmission rate (scenario 2).



FIGURE 14: Throughput versus packet transmission rate (scenario 2).

the other most ultimate condition is described in scenario 1, where the sink node is the center of all nodes. We can expect and evaluate that the throughput of GTCC with RR in normal scenarios between two most ultimate conditions will be distributed between the lowest one in Figure 14 and better one in Figure 12.

In Figure 15, the average hop count to the sink in our protocol is 2.6 and 4.6 in scenarios 1 and 2, respectively. It is a little higher than CRPL-OF0. But, our protocol can achieve higher throughput and less packet loss rate. Finally, we can see that CRPL-OF-ETX has the largest hop count among all protocols in scenario 1 because nodes are easily changing their parents to the ones which have longer hops than original parents to the sink in case of congestion happened.

5. Conclusions

In traditional congestion control scheme, rate reducing mechanisms are used to mitigate the presence of congestion. However, rate reducing will degrade the throughput and



FIGURE 15: Average hop count to the sink in both scenarios.

lose the fidelity of application's request. Instead of alleviating the congestion only by reduction the rate of transmission, we try to find another scheme based on game theory to improve the throughput. In this paper, we propose 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 communications. When congestion occurred, nodes will change their parents according to the utility function in game theory to avoid the congested path. If congestion is not alleviated, we conduct a rate reduction mechanism to reduce the transmission rate of source nodes. 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 a little average hop count increasing to sink node. Furthermore, GTCC with RR schemes has two times improvement in packet loss rate and 25% degrade in throughput compared by GTCC without RR.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- S. Deering and R. Hinden, "Internet protocol, version 6 (IPv6) specification," Tech. Rep. RFC 2460, Internet Engineering Task Force, 1998, http://www.ietf.org/rfc/rfc2460.txt.
- [2] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 packets over IEEE 802.15.4 networks," RFC 4944, Internet Engineering Task Force, 2007, (Proposed Standard), http://tools.ietf.org/html/rfc4944/.
- [3] T. Winter, P. Thubert, A. Brandt et al., "RPL: IPv6 routing protocol for low-power and lossy networks (RPL)," RFC 6550, Internet Engineering Task Force, 2012, (Proposed Standard), http://tools.ietf.org/html/rfc6550/.
- [4] P.-K. Huang, X.-J. Lin, and C.-C. Wang, "A low-complexity congestion control and scheduling algorithm for multihop wireless networks with order-optimal per-flow delay," *IEEE/ACM Transactions on Networking*, vol. 21, no. 2, pp. 495–508, 2012.

- [5] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, "A high-throughput path metric for multi-hop wireless routing," *Wireless Networks*, vol. 11, no. 4, pp. 419–434, 2005.
- [6] D. Monderer and L. S. Shapley, "Potential games," *Games and Economic Behavior*, vol. 14, no. 1, pp. 124–143, 1996.
- [7] F. Österlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross-level sensor network simulation with COOJA," in *Proceedings of the 31st Annual IEEE Conference on Local Computer Networks*, pp. 641–648, Tampa, Fla, USA, November 2006.
- [8] N. Tsiftes, J. Eriksson, and A. Dunkels, "Poster abstract: lowpower wireless IPv6 routing with ContikiRPL," in *Proceedings* of the 9th ACM/IEEE International Conference on Information Processing in Sensor Networks, pp. 406–407, Stockholm, Sweden, April 2010.
- [9] 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.
- [10] C.-Y. Wan, S. B. Eisenman, and A. T. Campbell, "Energyefficient congestion detection and avoidance in sensor networks," *ACM Transactions on Sensor Networks*, vol. 7, no. 4, article 32, 2011.
- [11] P. Antoniou and A. Pitsillides, "A bio-inspired approach for streaming applications in wireless sensor networks based on the Lotka-Volterra competition model," *Computer Communications*, vol. 33, no. 17, pp. 2039–2047, 2010.
- [12] C. Sergiou and V. Vassiliou, "HRTC: a hybrid algorithm for efficient congestion control in wireless sensor networks," in *Proceedings of the 6th IEEE International Conference on New Technologies, Mobility and Security (NTMS '14)*, pp. 1–5, IEEE, Dubai, United Arab Emirates, March-April 2014.
- [13] V. S. Deshpande, P. P. Chavan, V. M. Wadhai, and J. B. Helonde, "Congestion control in Wireless Sensor Networks by using Differed Reporting Rate," in *Proceedings of the 2nd World Congress* on Information and Communication Technologies (WICT '12), pp. 209–213, IEEE, Trivandrum, India, November 2012.
- [14] J. Jin, M. Palaniswami, and B. Krishnamachari, "Rate control for heterogeneous wireless sensor networks: characterization, algorithms and performance," *Computer Networks*, vol. 56, no. 17, pp. 3783–3794, 2012.
- [15] C.-Y. Wan, S. B. Eisenman, A. T. Campbell, and J. Crowcroft, "Siphon: overload traffic management using multiradio virtual sinks in sensor networks," in *Proceedings of the 3rd ACM International Conference on Embedded Networked Sensor Systems* (SenSys '05), pp. 116–129, San Diego, Calif, USA, November 2005.
- [16] 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, 2011.
- [17] M. M. Bhuiyan, I. Gondal, and J. Kamruzzaman, "LACAR: location aided congestion aware routing in wireless sensor networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '10)*, pp. 1–6, Sydney, Australia, April 2010.
- [18] C. T. Li, J. K. Wang, B. Wang, and Y. H. Han, "An efficient compressed sensing-based cross-layer congestion control scheme for Wireless Sensor Networks," in *Proceedings of the* 26th Chinese Control and Decision Conference (CCDC '14), pp. 637–641, Changsha, China, June 2014.
- [19] D. DebBarma, Q. Wang, I. G. M. M. Niemegeers, S. H. de Groot, and A. Lo, "Coalition game-theory-based congestion control

in hybrid Fi-Wi indoor network," in *Proceedings of the 3rd International Conference on Future Generation Communication Technologies (FGCT '14)*, pp. 25–32, August 2014.

- [20] Y. H. Kwon and B. H. Rhee, "A Bayesian game-theoretic approach for MAC protocol to alleviate beacon collision under IEEE 802.11p WAVE vehicular network," in *Proceedings of the* 6th International Conference on Ubiquitous and Future Networks (ICUFN '14), pp. 489–494, IEEE, Shanghai, China, July 2014.
- [21] L. Q. Tao and F. Q. Yu, "ECODA: enhanced congestion detection and avoidance for multiple class of traffic in sensor networks," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 3, pp. 1387–1394, 2010.
- [22] A. El Masri, A. Sardouk, L. Khoukhi, A. Hafid, and D. Gaiti, "Neighborhood-aware and overhead-free congestion control for IEEE 802.11 wireless mesh networks," *IEEE Transactions on Wireless Communications*, vol. 13, no. 10, pp. 5878–5892, 2014.
- [23] 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, 2010.





World Journal

Rotating Machinery



Journal of Sensors



International Journal of Distributed Sensor Networks



Advances in Civil Engineering





Submit your manuscripts at http://www.hindawi.com









International Journal of Chemical Engineering





International Journal of Antennas and Propagation





Active and Passive Electronic Components





Shock and Vibration





Acoustics and Vibration