JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 35, 737-748 (2019)
DOI: 10.6688/JISE.201907_35(4).0002

A Cooperative Game Theoretic Approach for Congestion Management in 6LoWPAN

G. RAJESH[1] AND R. RANJITHA[2]
Department of Information Technology
Anna University
Chennai, 600044 India
E-mail: [1]raajiimegce@gmail.com; [2]ranjithamercy@gmail.com

The internet of (IoT) is things a computing idea that describes the concept of everyday physical objects being connected to the web and having the ability to spot themselves to alternative devices. The Internet Engineering Task Force (IETF) has developed a set of IPv6-based protocols to overcome the challenges of connecting resource-limited wireless sensor nodes to the Internet. In that 6LoWPAN networks, due to the scalable number of devices, heavy network traffic causes congestion which significantly influences the network performance and affects the network QoS parameters. The proposed system of this paper includes the cooperative game theoretic approach for congestion prediction and buffer sharing algorithm (BSA) to reduce congestion in the 6LoWPAN network due to the buffer overflow. Simulation results of the proposed system imply that this outperforms by an average of 52%, 41.71%, 26.19% in terms of throughput, end-to-end latency, and energy consumption respectively as compared to existing GTCCF (game theory based congestion control framework) algorithm.

Keywords: congestion control, cooperative game theory, buffer sharing algorithm, 6LoWPAN networks, IoT applications

1. INTRODUCTION

IoT is considered to be the current very big opportunity and challenge for the global research, business community, technology users and companies [1]. The Internet of Things (IoT) has a highly complicated topology that consists of many types of heterogeneous sensor networks. In recent years, the number of wearable and mobile devices in IoT has increased exponentially [2]. These things are connected to the Internet where they can collaborate and provide services such as smart city, smart home, and smart healthcare applications [2].

Smart Home applications have become the success full business in the global market and it is predicted that Smart homes will become as common as mobile phones very soon. IoT products give us greater control over home or office door locks, lights, and home appliances and also provides insights into resource consumption habits; inventory and business processes; and better connection to the people, systems and the environments that shape our daily lives. With Smart home companies like Microsoft, Nest, Ecobee, Ring, and August, to name a few, will become household brands and they now started delivering smart appliances.

Received January 5, 2018; accepted September 19, 2018.
Communicated by Sun-Yuan Hsieh.

1.1 Game Theory

Game theory is a mathematical tool which is used to analyze the strategical interaction among multiple decision makers [15]. Game theory is used in the study of decision-making process where different players make their choices. The agents who are playing the game are called players. In every movement of the game, the player takes an action. Their plan of actions to be taken by the player is called strategy such as the request for bandwidth, offered price, new call *etc.* [6]. After taking the decision of all players, each player will get either a positive or a negative return. This return of every player is called its payoff. Each player will choose strategies which can maximize their payoff. This will lead to the concept of equilibrium in a game, which is called the solution of a game. Thus, equilibrium is defined as the combination of best strategies for each player. When every player cannot increase its payoff by changing its strategy while there is no change in another player strategy then the solution of that game is called Nash equilibrium. When the player cannot be further increased its payoff with any strategy, then the game has its Pareto optimal solution. If there is more than one equilibrium candidate then the Pareto optimal solution is preferred [17]. It can be used for decision making between competitive environment and cooperative environment.

Nash equilibrium is the well-known solution in traditional game theory. But in this game, evolutionary equilibrium provides an optimal solution and it gives the stability. In an evolutionary game, initially, random networks are assigned to all the users present in the scenario. Then payoff associated with each user assigned any particular network in the scenario is calculated [23]. The payoff achieved by user choosing network is given by

$$\pi_i = \cup(c_i|n_i) - p_i n_i. \tag{1}$$

Where above formula denotes the capacity associated with a network, is the pricing function of the network and is the total number of users choosing network in that area which is given by

$$n_i = N * x_i. \tag{2}$$

Where N and denotes the total number of users and proportion of users choosing network respectively. The function in Eq. (1) is called utility function and defined by

$$\cup = [u_k(x_k)]\ w_k. \tag{3}$$

Where denotes the number of network attributes, denote the utility associated with each attribute, denotes the attributes vector and denotes the weight given to each attribute and it can be calculated by any method such as AHP method. After calculating individual payoff, the average payoff of the user is calculated which is given by

$$\bar{\pi} = \sum x_i \pi_i. \tag{4}$$

This average payoff is used for selection of the network. If the average payoff is equal to the individual payoff, then the network which has the maximum payoff is selected by the user. This selection technique is particularly used at the user side for selection of network and in this, no cooperation is involved as every time user select the network from the list of the network without consulting any other user present in the selection environment, so it is a user-centric and non-cooperative game theory approach.

This paper is motivated by above contemplations to propose a congestion control algorithm called Cooperative game theory framework to solve the congestion problem and is aware of congestion prediction using utility function and congestion cost function to support the IoT application requirements. The main contributions of this paper include:

Design a congestion control game algorithm to avoid congestion in 6LoWPAN networks. The node's payoff function is formulated to achieve the node demand (preference) on resources for sending high data rate (utility function) and the desirable fairness among leaf nodes according to their preferences (priority cost function), while alleviating and mitigating congestion in the network (congestion cost function).

Here proposed a lightweight buffer sharing algorithm (BSA) which is aware of the idle nodes with more buffer space, to support the network demands. The proposed algorithm is evaluated IoT operating system, Contiki OS [9], through Cooja simulator [10].

2. RELATED WORK

In literature, many algorithms have been proposed and tested for the congestion control for 6LoWPAN networks [6, 7, 11]. However, the most of the existing literature does not consider the unique characteristics of the IEEE 802.15.4 standard, IPv6 and 6LoWPAN protocol stack (*i.e.* RPL routing protocol, the adaptation layer, and IEEE 802.15.4 MAC and PHY layers [1]. However, according to our best knowledge, none of the proposed algorithms in congestion control literature for WSNs and 6LoWPAN networks: (i) uses cooperative game theory for congestion prediction and (ii) buffer sharing algorithm [12, 13] to solve the congestion problem in IoT networks. However, the cooperative game theory provides an analytical framework suited for characterizing the interactions and decision making process among several players with conflicting interests [15].

In [16], Michopoulos *et al.* proposed a new congestion control algorithm called Duty Cycle-Aware Congestion Control (DCCC6) for 6LoWPAN networks. DCCC6 algorithm detects the presence of radio duty cycle and adjusts its operation accordingly. the authors used the dynamic buffer occupancy as a congestion detection method as well as a modified AIMD (Additive-Increase Multiplicative-Decrease) to manage the congestion in the network.

In [17], Castellani *et al.* proposed three different congestion control schemes called Griping, Deaf and Fuse for controlling unidirectional and bidirectional data flows in (Constrained Application Protocol) CoAP/6LoWPAN networks. CoAP is based on distributed back pressure concept. It uses a buffer occupancy strategy (in Griping) and missing acknowledgment packet (in Deaf and Fuse) to detect the congestion as well as AIMD scheme to avoid the congestion by adjusting the transmission rate to reduce the injected packets into the network.

In [18], Hellaoui and Koudil proposed a congestion control solution for CoAP/6Lo-

WPAN networks. It is based on a bird flocking concept to pass packets through uncongested areas and avoid congested ones. Also uses the buffer occupancy strategy to detect congested nodes in the network as well as the resource control method to mitigate the congestion by selecting the least congested routes to deliver packets to the destination (sink node).

In [19, 20], Kim *et al.* proposed an effective queue utilization based RPL algorithm called (QU-RPL). QU-RPL uses the queue utilization factor in parent selection process to satisfy the traffic load balancing. When a node experiences a certain number of consecutive buffer overflows, it broadcasts a DIO (DODAG Information Object) message which contains the congestion information. The node alters its parent on experiencing congestion with one that has less buffer occupancy and lower hop distance to the sink node. In the absence of congestion, the node chooses its best parent based on the same parent selection mechanism of the default RPL.

In [14, 21], the authors proposed a congestion control mechanism called Game Theory Congestion Control (GTCC) for 6LoWPAN networks. GTCC detects congestion by using the network packet flow rate which is the packet generation rate subtracted by the packet service rate. When a parent node identifies the network congestion, it sends a congestion intimation message to its children through a DIO control packet. When the children nodes receive the DIO packet, they start the parent-change procedure.

In [22], Tang *et al.* proposed a congestion avoidance multipath routing algorithm based on RPL called CA-RPL. Also, the authors propose a routing metric for RPL called DELAY ROOT which minimizes the average delay toward the root node. CA-RPL mitigates network congestion by distributing a large amount of traffic to different paths. CA-RPL uses the DELAY ROOT and three other metrics: ETX (Expected Transmission Count), rank and number of received packets for parent selection process

3. PROPOSED SYSTEM

Network performance is also reduced by congestion in the network. When the number of packets sent into the network is more than carrying capacity, the overflowed packets will be dropped in the network. In this proposed system number of packets get reduced by performing load sharing in the buffer. Fig. 1 shows the module diagram of the proposed system.

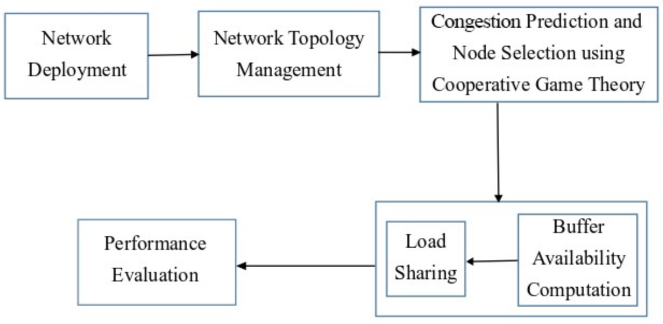

Network Deployment
Network Topology Management
Congestion Prediction and Node Selection using Cooperative Game Theory
Buffer Availability Computation
Load Sharing
Performance Evaluation

Fig. 1. Module diagram of the proposed system (BSA).

3.1 Module Diagram

3.1.1 Network deployment and topology management

In 6LoWPAN networks, the RPL routing protocol [7] is responsible for constructing the network topology. Three types of nodes are defined: (1) sink (root) nodes which act as the hub or interface or access node for other nodes in the network; (2) intermediate nodes which forward packets to the sink; and (3) leaf nodes. The DAG root broadcasts a DIO control message to other nodes in the network. When an intermediate node receives the DIO message, it replies to the sink node with destination advertisement object (DAO) for joining the DODAG. Then, the intermediate node sends a DIO message to all neighbors. This process continues until the DIO message reaches the leaf nodes. When a node receives a DIO message from more than one neighbor, it selects its parent with the best rank. Also, when a node does not receive a DIO message within a specific time, it sends a DODAG information solicitation (DIS) message to solicit DIO messages from its neighbors.

3.1.2 Congestion prediction

A game theoretic approach to predict congestion and also select a node for sharing the load in the 6LoWPAN network

Cooperative game theory Bankruptcy game is the cooperative game theory which is used to model distribution problems [6]. Bankruptcy game is used in the case when the amount is insufficient to satisfy all user condition. A coalition exists in this game as it is a cooperative game. For obtaining better or maximum payoff, players cooperate with each other. After coalition forming, the earned payoff is denoted by the characteristics function.

Players: Group of M players (leaf nodes), $L_1, \ldots, L_k, \ldots, L_m$ where m represents the number of leaf nodes which are associated with the parent.

Strategies: S_k; $\forall k \in M$ represents the feasible action space for player L_k. Each node (player) L_k can send a minimum data rate of zero and a maximum data rate of λmax_k.

Congestion cost function: This function reflects how much the parent node is congested due to the leaf nodes. According to Queuing Theory; if the arrival rate at the parent node's buffer is higher than the service rate from the parent, the buffer starts overflowing the packets and congestion occurs. Thus, one possible method is to choose the congestion cost function as the ratio between the total receiving rate and total forwarding rate at the parent's buffer. As the receiving rate is greater than the forwarding rate the ratio increases. We use $C_k(\lambda_k, \lambda - k)$ to represent the congestion cost of node (player) L_k where $\lambda - k = [\lambda_j]$ $j \in M$; is the vector of sending rates (strategies) of all players except player L_k and $s = (\lambda_k, \lambda - k) \in S$ is referred to as the strategy profile.

Utility function: This function is designed where each player gets more profit by increasing its sending rate. We use $U_k(\lambda_k)$ to represent the utility function of player L_k where λ_k is sending rate (strategy) of player L_k.

$$U_k(\lambda_k) = \log(\lambda_k + 1) \quad (5)$$

Priority cost function: Represent the priority cost function of player L_k; Player L_k has to pay a penalty based on its priority (p_k) and its sending rate (λ_k) to distinguish between high priority nodes and low priority nodes.

3.1.3 Modified RPL

RPL Instance It defines Optimization Objective when forming paths towards roots based on one or more metrics. Metrics may include both Link properties (Reliability, Latency) and Node properties (Powered on not). A network may run multiple instances concurrently with different optimization criteria.

RPL Control Messages RPL defines a new ICMPv6 message with three possible types: DAG Information Object (DIO) – carries information that allows a node to discover an RPL Instance, learn its configuration parameters and select DODAG parents. DAG Information Solicitation (DIS) – solicit a DODAG Information Object from an RPL node. Destination Advertisement Object (DAO) – used to propagate destination information upwards along the DODAG.

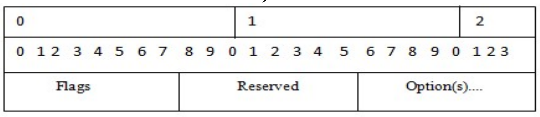
Table 1. RPL control messages.

RPL Control Message	Functions
DODAG Information Object (DIO)	Carries Information and Select DODAG parent
Destination Advertisement Object (DAO)	Propagate destination and Transfer upwards
DODAG Information Solicitation (DIS)	Solicit DIO message from RPL node
DAO Acknowledgment (DAO-ACK)	Send by DAO parent

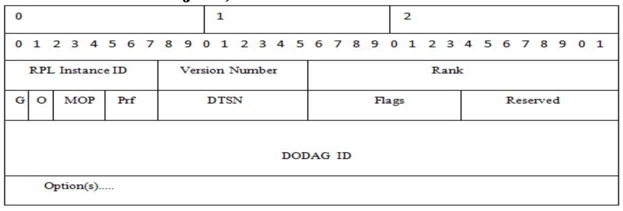
DODAG Construction

Nodes periodically send link-local multicast DIO messages, Stability or detection of routing inconsistencies influence the rate of DIO messages, nodes listen for DIOs and use their information to join a new DODAG, or to maintain an existing DODAG also use a DIS message to solicit a DIO. Based on information in the DIOs the node chooses parents that minimize path cost to the DODAG root.

DIS (Destination Information Solicitation):



DIO (DODAG Information Object):



DAO (Destination Advertisement Object):

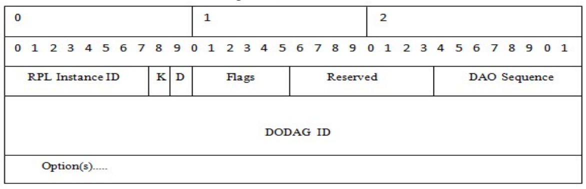


Fig. 2. Format of control messages.

Buffer availability computation

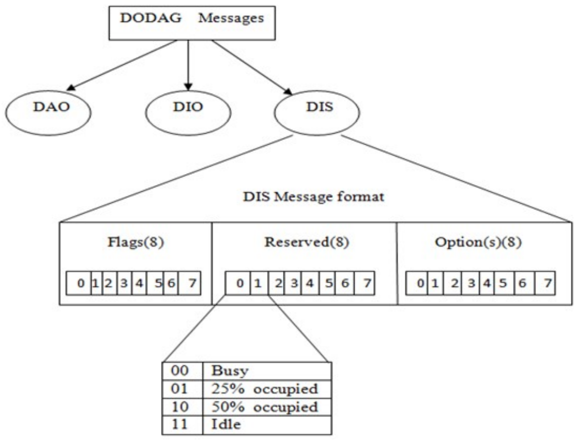


Fig. 3. Buffer availability computation.

In this buffer availability computation performed over the DIS message format which has an unused reserved bit (8 bit), from that two bits are taken for checking buffer status in the receiver node.

Algorithm of buffer availability computation:
Input:
BO – Buffer Occupancy
ETX – Expected Transmission Count
STATUS BIT

Output:
FLAG BIT SET for the node has space in the buffer.

```
init:
BO Value = flag bit
  process: if flag(11)
  Node is idle
  Select(node)
  else if flag(10)
  Node 50% free
  Select(node)
```

```
end:
else Node Busy
```

The parameters used are BO-buffer occupancy, Expected transmission count. The buffer occupancy threshold value is set using a flag bit in DIS message. Whenever flag bit set to 11,10,01 then space is available in the buffer.

3.4 Buffer Sharing Algorithm

In the buffer sharing algorithm, multiple frames are sent by the sender at a time before needing an acknowledgment. Multiple frames sent by the source are acknowledged by the receiver using a single ACK frame. It provides the upper limit on the number of frames that can be transmitted before requiring an acknowledgment. The windows have a specific size in which the frames are numbered modulo-*n*, which means they are numbered from 0 to $n-1$. For *e.g.* if $n = 8$, the frames are numbered 0, 1, 2, 3, 4, 5, 6, 7, 0, 1, 2, 3, 4, 5, 6, 7, 0, 1. The size of the window is $n-1$. For *e.g.* in this case it is 7, therefore, a maximum of $n-1$ frames may be sent before an acknowledgment. For example, in order to acknowledge the group of frames ending in frame 4, the receiver sends an ACK containing the number 5. When sender sees an ACK with number 5, it comes to know that all the frames up to number 4 have been received.

Algorithm of load sharing:

```
N = window size
Rn = request number
Sn = sequence number
Sb = sequence base
Sm = sequence max
Receiver
    $Rn = 0$
    Do the following forever:
If      the packet received = $Rn$ and the packet is error free
        Accept the packet and send it to a higher layer
        $Rn = Rn + 1$
Else
        Refuse packet
        Send a Request for $Rn$
Sender:
        $Sb = 0$
        $Sm = N + 1$
        Repeat the following steps forever:
        1. If you receive a request number where $Rn > Sb$
        $Sm = (Sm - Sb) + Rn$
        $Sb = Rn$
        2. If no packet is in transmission,
        Transmit a packet where $Sb \Leftarrow Sn \Leftarrow Sm$.
        Packets are transmitted in order.
```

4. PERFORMANCE EVALUATION

The proposed congestion control framework has been tested and evaluated on different network scenarios through simulation by using the Contiki 3.0 OS and Cooja simulator.

Table 2. Contiki OS and Cooja set up.

Settings	Value
Wireless channel	UDG model with Distance Loss
Communication range	5m
Number of nodes	1 Sink & 7 Receiver nodes
Mote type	Tmote Sky
Transport layer	UDP
Network layer	IPv6
MAC layer	Contiki MAC

Fig. 4 shows the number of received packets from the leaf nodes at the sink. For BSA, it is clear that the node consumes equal energy due to the sharing of data between nodes to reduce congestion which is occurred due to bulk data transfer by a single node in a network. The energy consumption level of a node at any time of the simulation can be determined by finding the difference between the current energy value and initial energy value. If an energy level of a node reaches zero, it cannot receive or transmit any more packets. The amount of energy consumption in a node can be printed in the trace file. The energy level of a network can be determined by summing the entire node's energy level in the network.

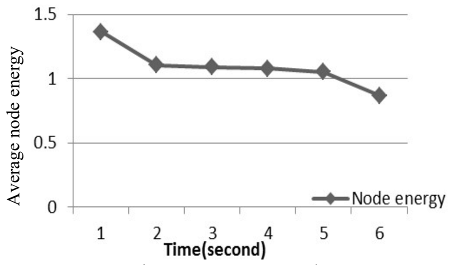


Fig. 4. Average node energy.

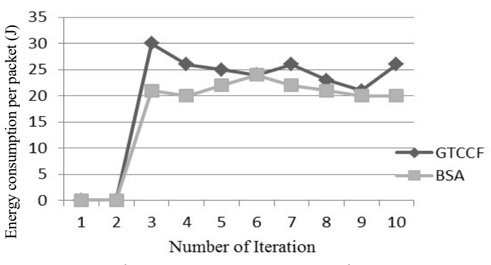


Fig. 5. Energy consumption.

Fig. 5 shows the energy consumption due to transmission and reception in the leaf and intermediate nodes per successfully delivered packet. Energy consumption per packet is calculated by,

$$\frac{\text{Total energy consumption due to } Tx \text{ and } Rx}{\text{Total number of received packets at the sink}}. \tag{6}$$

Here note that with BSA, the energy consumption in the network is less than GTCCF because GTCCF waste energy by transmitting and receiving packets which are then lost due to buffer overflow on the path without successful delivery. Also, the consumed energy per packet in GTCCF is significantly higher than other, as the number of delivered packets to sink is much lower so that BSA works well in the case of the buffer overflow in node due to heavy data to transfer. The performance of BSA in energy consumption is 26.19% increased than GTCCF.

The overall throughput is the total number of received packets every second at the sink node. It is clear that BSA has stable and higher throughput as compared to another algorithm such as GTCCF. Throughput is calculated using the formula,

$$\frac{\text{Number of bits received}}{\text{Number of bits transmitted}}. \tag{7}$$

Here in this Fig. 6 BSA has content and stable one due to transfer data from the receiver and also its performance in throughput is 52% higher than GTCCF.

Fig. 7 shows end-to-end delay which is the time between a packet being generated at the application of the source until its successful reception at the application of the final destination. When congestion occurs, the delay is high because the buffer is full so the packet waiting time in the buffer is high. In BSA algorithm, buffer full is avoided by sharing so delay reduced. BSA algorithm improves performance in end-to-end delay is 41.71% higher than GTCCF algorithm.

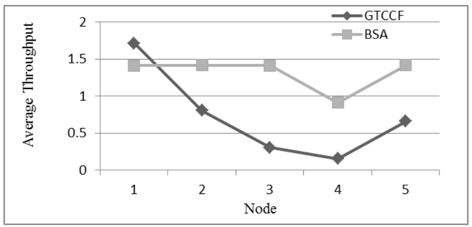


Fig. 6. Average throughput per packet.

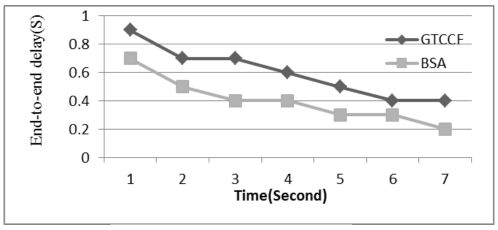


Fig. 7. End-to-end delay.

5. CONCLUSION

In this paper, the congestion problem in 6LoWPAN networks is modeled as a game by using the cooperative game theory. Also, a new and simple congestion control mechanism called a buffer sharing algorithm (BSA) is proposed. To support the IoT application requirements, the proposed framework is aware of buffer overflow due to bulk data transfer. Also, BSA is built and designed on the unique characteristics of IEEE 802.15.4, IPv6 and 6LoWPAN protocol stack. The proposed algorithm is evaluated in Contiki 3.0 OS simulated using cooja simulator and game theory based congestion control framework (GTCCF). Simulation results show that our proposal improves the QoS aspects *e.g.* throughput, end-to-end delay, energy consumption, packet loss ratio as compared to the existing algorithm. Future work will involve investigating the impact of BSA in other Qos parameters like queuing delay and application layer latency.

REFERENCES

1. H. A. A. Al-Kashoash, M. Hafeez, and A. H. Kemp, "Congestion control for 6LoWPAN networks: A game theoretic framework," *IEEE Internet of Things Journal*, Vol. 4, 2009, pp. 760-771.
2. L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Computer Networks*, Vol. 54, 2010, pp. 2787-2805.
3. N. Khalil, M. R. Abid, D. Benhaddou, and M. Gerndt, "Wireless sensors networks for Internet of Things," in *Proceedings of IEEE 9th International Conference on Intelligent Sensors*, *Sensor Networks and Information Processing*, 2014, pp. 1-6.
4. N. Kushalnagar, G. Montenegro, and C. Schumacher, "IPv6 over LowPower wireless personal area networks (6LoWPANs): Overview, assumptions, problem statement, and goals," *Internet Engineering Task Force*, RFC 4919, 2007.
5. M. Weldon, *The Future X Network: A Bell Labs Perspective*, CRC Press, FL, 2016.
6. A. Ghaffari, "Congestion control mechanisms in wireless sensor networks: A survey," *Journal of Network and Computer Applications*, Vol. 52, 2015, pp. 101-115.
7. M. A. Kafi, D. Djenouri, J. Ben-Othman, and N. Badache, "Congestion control protocols in wireless sensor networks: A survey," *Communications Surveys & Tutorials*, Vol. 16, 2014, pp. 1369-1390.
8. T. Winter, P. Thubert, A. Brandt, J. Hui, and R. Kelsey, "RPL: IPv6 routing protocol for low-power and lossy networks," *Internet Engineering Task Force*, RFC 6550, 2012.
9. A. Dunkels, B. Grönvall, and T. Voigt, "Contiki – a lightweight and flexible operating system for tiny networked sensors," in *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, 2004, pp. 455-462.
10. F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, "Cross level sensor network simulation with COOJA," in *Proceedings the 31st IEEE Conference on Local Computer Networks*, 2006, pp. 641-648.
11. C. Sergiou, P. Antoniou, and V. Vassiliou, "A comprehensive survey of congestion control protocols in wireless sensor networks," *Communications Surveys & Tutorials*, Vol. 16, 2014, pp. 1839-1859.
12. H.-Y. Shi, W.-L. Wang, N.-M. Kwok, and S.-Y. Chen, "Game theory for wireless sensor networks: A survey," *Sensors*, Vol. 12, 2012, pp. 9055-9097.
13. R. Machado and S. Tekinay, "A survey of game-theoretic approaches in wireless sensor networks," *Computer Networks*, Vol. 52, 2008, pp. 3047-3061.
14. J. P. Sheu, C. X. Hsu, and C. Ma, "A game theory based congestion control protocol for wireless personal area networks," in *Proceedings of IEEE 39th Annual Computer Software and Applications Conference*, Vol. 2, 2015, pp. 659-664.
15. Z. Han, D. Niyato, W. Saad, T. Bas¸ar, and A. Hjørungnes, *Game Theory in Wireless and Communication Networks: Theory*, *Models*, *and Applications*, Cambridge University Press, UK, 2012.
16. V. Michopoulos, L. Guan, G. Oikonomou, and I. Phillips, "DCCC6: Duty cycle-aware congestion control for 6LoWPAN networks," in *Proceedings of IEEE International Conference on Pervasive Computing and Communications Workshops*, 2012, pp. 278-283.
17. A. P. Castellani, M. Rossi, and M. Zorzi, "Back pressure congestion control for CoAP

/6LoWPAN networks," *Ad Hoc Networks*, Vol. 18, 2014, pp. 71-84.
18. H. Hellaoui and M. Koudil, "Bird flocking congestion control for CoAP/RPL/6LoWPAN Networks," in *Proceedings of ACM Workshop on IoT Challenges in Mobile and Industrial Systems*, 2015, pp. 25-30.
19. H.-S. Kim, J. Paek, and S. Bahk, "QU-RPL: Queue utilization based RPL for load balancing in large scale industrial applications," in *Proceedings of the 12th Annual IEEE International Conference on Sensing*, *Communication*, *and Networking*, 2015, pp. 265-273.
20. H.-S. Kim, H. Kim, J. Paek, and S. Bahk, "Load balancing under heavy traffic in rpl routing protocol for low power and lossy networks," *IEEE Transactions on Mobile Computing*, Vol. 16, 2016, pp. 964-979.
21. C. Ma, J.-P. Sheu, and C.-X. Hsu, "A game theory based congestion control protocol for wireless personal area networks," *Journal of Sensors*, Vol. 2016, 2015, pp. 1-13.
22. W. Tang, X. Ma, J. Huang, and J. Wei, "Toward improved RPL: A congestion avoidance multipath routing protocol with time factor for wireless sensor networks," *Journal of Sensors*, Vol. 2016, 2015, pp. 1-11.
23. H. A. A. Al-Kashoash, Y. Al-Nidawi, and A. H. Kemp, "Congestion aware RPL for 6LoWPAN networks," in *Proceedings of IEEE Wireless Telecommunications Symposium*, 2016, pp. 1-6, doi: 10.1109/WTS.2016.7482026.
24. L. Wang and G.-S. Kuo, "Mathematical modeling for network selection in heterogeneous wireless networks – A tutorial," *Communications Surveys and Tutorials*, Vol. 15, 2013, pp. 271-292.
25. H. A. A. Al-Kashoash, Y. Al-Nidawi, and A. H. Kemp, "Congestion analysis for low power lossy networks," in *Proceedings of IEEE Wireless Telecommunications Symposium*, 2016, pp. 1-6, doi: 10.1109/WTS.2016.7482027.

G. Rajesh received his Ph.D. in the faculty of Information and Communication Engineering from Anna University, Chennai, India. He is presently working as an Assistant Professor in the Department of Information Technology, Anna University. His areas of interest include software engineering, wireless sensor networks, wireless networks, and computational intelligence.

Rajadurai Ranjitha received her B.Tech degree in Information Technology from Prince Shri Venkateshwara Padmavathy Engineering College at Anna University in 2016 and M.Tech degree in Information Technology from MIT Campus at Anna University. She is currently working as an Assistant Professor at Jain University (Inurture Educational Solution). Her research interests include networking, Internet of Things and mobile application development.