Available online at www.sciencedirect.com
ScienceDirect
ICT Express 4 (2018) 209–215
www.elsevier.com/locate/icte

Analytical modelling of congestion for 6LoWPAN networks

Hayder A.A. Al-Kashoash[a,b,*], Fadoua Hassen[a], Harith Kharrufa[a], Andrew H. Kemp[a]

[a] Electronic and Electrical Engineering School, University of Leeds, Leeds LS2 9JT, UK
[b] Technical Institute/Qurna, Southern Technical University, Basra, Iraq

Received 18 November 2016; received in revised form 12 April 2017; accepted 1 November 2017
Available online 2 December 2017

Abstract

The IPv6 over Low-Power Wireless Personal Area Network (6LoWPAN) protocol stack is a key part of the Internet of Things (IoT) where the 6LoWPAN motes will account for the majority of the IoT 'things'. In 6LoWPAN networks, heavy network traffic causes congestion which significantly affects the overall performance and the quality of service metrics. In this paper, a new analytical model of congestion for 6LoWPAN networks is proposed using Markov chain and queuing theory. The derived model calculates the buffer loss probability and the channel loss probability as well as the number of received packets at the final destination in the presence of congestion. Also, we calculate the actual wireless channel capacity of IEEE 802.15.4 with and without collisions based on Contiki OS implementation. The validation of the proposed model is performed with different scenarios through simulation by using Contiki OS and Cooja simulator. Simulation results show that the analytical modelling of congestion has an accurate agreement with simulation.

Keywords: Markov chain; Queuing theory; Congestion; 6LoWPAN network

1. Introduction

The IoT is considered to be the next big challenge for the Internet research community and it has recently drawn significant research attention [1]. The IoT will comprise billions of intelligent communicating things such as wireless sensor nodes, radio frequency identification (RFID) tags and near field communication (NFC) devices that extend the border of the world with physical entities and virtual components [2]. Wireless sensor networks (WSNs) are considered as one of the most important elements in the IoT. 6LoWPAN [3] is used for full integration of WSN with the Internet where sensor nodes implement the Internet Protocol (IP) stack though it was originally designed for wired networks. However, the implementation of the TCP/IP model in WSN and 6LoWPAN networks has many issues due to the limitation of bandwidth, energy and buffer resources. Transmission Control Protocol (TCP) requires extra resources for connection setup and termination before and after the data transmission whilst User Datagram Protocol (UDP) does not provide a congestion control mechanism. Thus, TCP and UDP are not efficient for WSN and 6LoWPAN networks [1]. Therefore, one of the main issues in WSN and 6LoWPAN networks is congestion that causes packet loss, increased energy consumption and degraded throughput.

Congestion occurs when multiple sensor nodes start to send packets concurrently at high data rate or when a node relays many flows across the network. Thus, link collision on the wireless channel and packet overflow at buffer nodes occur in the network [4]. Recently, a few papers have investigated and addressed congestion in 6LoWPAN networks [5–12], but none considered congestion assessment and analysis through analytical modelling. In this paper, we propose an analytical model to study the 6LoWPAN network performance in the presence of congestion (e.g. how many packets are lost due to buffer overflow and the average number of packets received by

* Corresponding author at: Electronic and Electrical Engineering School, University of Leeds, Leeds LS2 9JT, UK.
E-mail addresses: hayderaam@stu.edu.iq (H.A.A. Al-Kashoash), elfha@leeds.ac.uk (F. Hassen), elhdy@leeds.ac.uk (H. Kharrufa), a.h.kemp@leeds.ac.uk (A.H. Kemp).
Peer review under responsibility of The Korean Institute of Communications and Information Sciences (KICS).

https://doi.org/10.1016/j.icte.2017.11.001
2405-9595/

a sink node) using Markov chain analysis and queuing theory. Queuing theory is one of the most important tools for studying and analysing computer network performance [13,14]. Queuing analysis is considered as a special case of Markov chains. It deals with queues (nodes' buffers) where customers (packets) compete to be processed by servers (sensor nodes). Also, we calculate the IEEE 802.15.4 effective channel capacity based on Contiki OS implementation with and without wireless channel collision occurrence. Finally, we validate our modelling with different parameters i.e. number of nodes, buffer sizes and offered loads, through simulation using Contiki OS [15] and Cooja simulator [16].

The reminder of the paper is structured as follows: In Section 2, we introduce the system model. In Section 3 and 4, we derive an analytical model of congestion in 6LoWPAN networks that calculates the buffer loss probability and channel loss probability respectively. Section 5 calculates the actual IEEE 802.15.4 channel capacity based on Contiki OS implementation. The accuracy of the derived model is evaluated through simulation in Section 6. Finally, Section 7 concludes this paper.

2. System model

In 6LoWPAN networks, the IPv6 routing protocol for low-power and lossy networks (RPL) [17] is responsible for constructing the network topology. Three types of nodes are defined: sink (root) nodes which provide connectivity to other networks, intermediate nodes which forward packets to the sink and leaf nodes. Consider a network of M leaf nodes, $L_1, \ldots, L_k, \ldots, L_M$, one intermediate node, I, and one sink node, S. The topology of the network is shown in Fig. 1. In the network, we denote by $l_{k,i}$ the link between node L_k and node I and $l_{i,s}$ the link between node I and node S. For node I, we define a children set, Δ_I, to be all nodes that have node I as a next-hop node. Each node in the network has a buffer of B packet size. We assume that the wireless channel capacity (CC_b) in bit per second is distributed among nodes as the intermediate node has half portion of the leaf node. The reason is that the radio of the intermediate node is receiving and transmitting at the same time whereas the leaf node's radio is just sending traffic. Also, we assume that the sensor nodes run the contention based IEEE 802.15.4 MAC protocol with unslotted CSMA/CA as an access mechanism.

When congestion occurs, the packets are lost either at the sensor node (buffer overflow) or on the wireless channel. Fig. 2 shows the node model for the leaf and intermediate nodes. In Fig. 2, the applications in the leaf nodes, $L_1, L_2, \ldots.., L_M$, generate packets at an average data rate of $\lambda_1^L, \lambda_2^L, \ldots\ldots, \lambda_M^L$ respectively. Then the packets are stored in the MAC's buffer to be transmitted by the MAC protocol to the intermediate node I. We assume that the leaf nodes $L_1, L_2, \ldots.., L_M$, transmit the packets with an average departure rate of $\mu_1^L, \mu_2^L, \ldots\ldots, \mu_M^L$ respectively. Before the packets arrive at the intermediate node I, a number of packets are lost on the wireless channel with a probability $P_{ch-loss}^j$ where $j = 1, 2, \ldots, k, \ldots, M$.

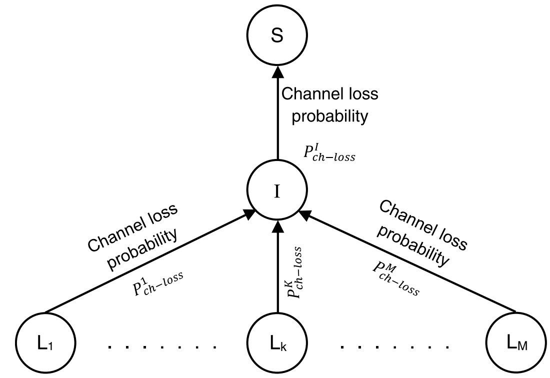


Fig. 1. Network topology.

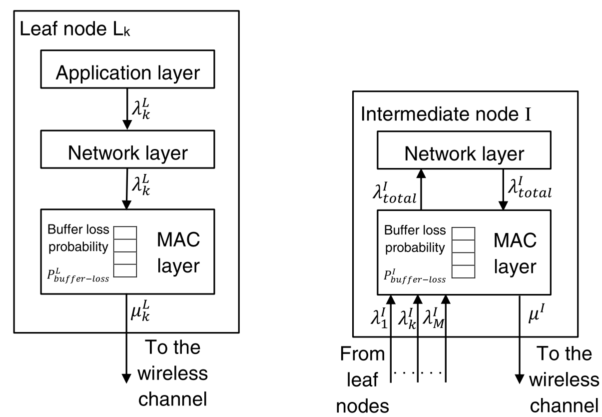


Fig. 2. Leaf and intermediate nodes model.

Then, packets arrive at the node I with an average rate of $\lambda_1^I, \lambda_2^I, \ldots\ldots, \lambda_M^I$ form nodes $L_1, L_2, \ldots.., L_M$ respectively as:

$$\lambda_j^I = (1 - P_{ch-loss}^j)\mu_j^I \tag{1}$$

where $j = 1, 2, \ldots, k, \ldots, M$, and the total arrival packets at node I is $\lambda_{total}^I = \sum_{j=1}^{M} \lambda_j^I$. When node I receives the packets, it stores them in its buffer to forward them later to the sink node S with an average departure rate of μ^I.

In the following sections, we develop a model to calculate the probabilities of packet buffer loss and channel loss in the network.

3. Buffer loss probability

In this section, we perform Markov chain analysis to calculate the buffer loss probability ($P_{buffer-loss}$). The states of the Markov chain represent the number of packets stored in the buffer. Consider the packet arrivals are independent binomial distribution with a mean rate of λ (packet/s) and a mean service time of each packet is assumed $1/\mu$. The buffer can be modelled as an M/M/1/B model where B represents the buffer size. We take the time step of state transitions equals to the inverse of the maximum data rate which is the channel capacity in packet per

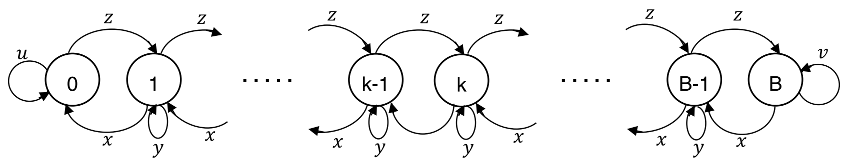


Fig. 3. State transition diagram.

second (CC_p) as follows:

$$T = \frac{1}{CC_p} \tag{2}$$

where $CC_p = CC_b/N$ and N is packet length in bits.

At a given time step a maximum of one packet can arrive at or leave the buffer. We assume that at a certain time step, the probability of packet arrival is P_{arr} and the probability that a packet leaves the queue is P_{dep}. The state transition diagram for the M/M/1/B queue is shown in Fig. 3 where $v = 1 - x$, $w = 1 - z$, $x = (1 - P_{arr})P_{dep}$, $y = P_{arr}P_{dep} + (1 - P_{arr})(1 - P_{dep})$ and $z = P_{arr}(1 - P_{dep})$.

The equilibrium (steady state) distribution vector of the transition matrix is given by:

$$\pi = [\pi_0 \quad \pi_1 \quad \pi_2 \quad \cdots \quad \pi_B].$$

To simplify the analysis, we consider that the applications at the leaf nodes generate packets with equal data rate of an average of λ^L where $\lambda^L = \lambda^L_1 = \lambda^L_2 = \cdots = \lambda^L_M$. For the leaf nodes, the probability of packet arrival at the buffer is $P^L_{arr} = \lambda^L/CC_P$ and the probability of packet departure is $P^L_{dep} = \mu^L_{max}/CC_p$.

As the channel capacity is distributed among nodes with leaf node having twice portion of the intermediate node, the maximum departure rate at a leaf node can be written as: $\mu^L_{max} = 2CC_p/(2M + 1)$. Thus, the average number of lost packets per time step T at each leaf node's buffer is:

$$L^T_{leaf} = \pi^L_B P^L_{arr}(1 - P^L_{dep}) \tag{3}$$

and the average number of lost packets per second at each leaf node's buffer is as follows:

$$L_{leaf} = \pi^L_B P^L_{arr}(1 - P^L_{dep})CC_p \tag{4}$$

Thus, the probability of lost packets at the leaf node's buffer is given by:

$$P^L_{buffer-loss} = \frac{L_{leaf}}{\lambda^L}. \tag{5}$$

For the intermediate node I, the probability of packet arrival at the buffer is $P^I_{arr} = \lambda^I_{total}/CC_p$ and the probability of packet departure is $P^I_{dep} = \mu^I_{max}/CC_p$ where μ^I_{max} is:

$$\mu^I_{max} = \begin{cases} CC_p/(2M + 1) & \text{if } \mu^L = \mu^L_{max} \\ CC_p - M\mu^L & \text{if } \mu^L < \mu^L_{max} \end{cases}$$

and $\mu^L = (1 - P^L_{buffer-loss})\lambda^L$.

Thus, the average number of lost packets per time step T at the intermediate node's buffer is as follows:

$$L^T_{inter.} = \pi^I_B P^I_{arr}(1 - P^I_{dep}) \tag{6}$$

and the average number of lost packets per second at the intermediate node's buffer is given by:

$$L_{inter.} = \pi^I_B P^I_{arr}(1 - P^I_{dep})CC_p. \tag{7}$$

Thus, we can write the probability of lost packets at the intermediate node's buffer as:

$$P^I_{buffer-loss} = \frac{L_{inter.}}{\lambda^L_{total}} \tag{8}$$

We also can write the total average number of lost packets per second at the buffers in the network as:

$$L_{buffer-loss} = ML_{leaf} + L_{inter.} \tag{9}$$

Ultimately, the total probability of lost packets at nodes' buffers in the network is:

$$P_{buffer-loss} = \frac{L_{buffer-loss}}{M\lambda^L}. \tag{10}$$

Substituting equations (4), (7) and (9) in (10), we get:

$$P_{buffer-loss} = \frac{[M\pi^L_B(CC_p - \mu^L)] + [\pi^I_B\lambda^I_{total}(CC_p - \mu^I)]}{M\lambda^L CC_p} \tag{11}$$

The average number of received packets per second at sink node (λ^S) is:

$$\lambda^S = (1 - P^I_{ch-loss})(1 - P^I_{buffer-loss})\lambda^I_{total}. \tag{12}$$

4. Channel loss probability

In the IEEE 802.15.4 CSMA/CA, packets are assumed to be lost in the wireless channel due to two reasons:

(1) Channel access failure: when a node tries to transmit a packet, it performs a CCA to sense the wireless channel. If the channel is idle, then the node begins to transmit. Otherwise, it increments the value of two parameters; the number of backoff (NB) and the backoff exponent (BE). After that, the node waits for a random time in the range $[0, (2^{BE} - 1)]$ backoff unit periods before it does the CCA again. Each backoff unit period equals 20 symbols * 16 µs/symbol [18]. This process continues until the value of BE exceeds the value of *macMaxCSMABackoffs* parameter. Then, the packet is discarded due to channel access failure.

(2) Maximum number of retransmission limit: when the node sends the packets it waits for an ACK packet. If the node does not receive the ACK packet due to a collision or an ACK timeout expires, then, it increments the retransmission count and tries to retransmit the packet. If the number of retransmissions reaches the maximum number of the retransmissions parameter *macMaxFrameRetries*, then, the packet is dropped. Thus, the probability of channel loss for node j ($P^j_{ch-loss}$) is:

$$P^j_{ch-loss} = P^j_{caf} + P^j_{mrl} \tag{13}$$

where P^j_{caf} is the probability of packet loss due to channel access failure, P^j_{mrl} is the probability of packet loss due to the maximum number of retransmissions limit and j is the node ID.

In [19], Di Marco et al. have developed an analytical model to calculate P^j_{caf} and P^j_{mrl} for unslotted IEEE 802.15.4 as follows:

$$P^j_{caf} = \frac{\alpha_j^{m+1}(1-(P_{coll,j}(1-\alpha_j^{m+1}))^{n+1})}{1-P_{coll,j}(1-\alpha_j^{m+1})} \tag{14}$$

$$P^j_{mrl} = (P_{coll,j}(1-\alpha_j^{m+1}))^{n+1} \tag{15}$$

where α_j is the probability that CCA is busy, $P_{coll,j}$ is the probability that a transmitted packet encounters a collision, m is the maximum number of backoffs and n is the maximum number of retransmissions, for node j. For more details about P^j_{caf} and P^j_{mrl}, please refer to [19].

5. Contiki-based IEEE 802.15.4 effective channel capacity

The developed buffer loss probability model depends on a set of parameters; one of them is the actual channel capacity. We do validation of our proposed modelling with Contiki OS and Cooja simulator. In this section, we estimate the actual channel capacity based on Contiki 3.0 OS implementation. The IEEE 802.15.4 standard supports a maximum data rate at the physical layer of 250 kbps in the 2.4 GHz band. In reality, the effective data rate is smaller than 250 kbps and its actual value varies with time due to channel access algorithm operation, overhead of ACK packet transmission, collisions and number of active nodes. Contiki OS uses unslotted CSMA/CA as a channel access mechanism and it implements the data link layer as three sublayers: framer, RDC and medium access control (MAC). In the simulation, these are 802.15.4 (framer), nullrdc (RDC) and CSMA (MAC). When the application generates packets, they are passed down to the MAC layer through the network layer and sicslowpan layer. When the MAC layer receives the packets, it enqueues them into its buffer. After that, the MAC layer sends the packets to the nullrdc layer which calls a function called 'NETSTACK_RADIO.prepare' to prepare the packet with the radio. While preparing, if the radio is currently receiving a packet over air or it has already received a packet that needs to be read before sending an ACK packet, then, the radio returns 'TX_COLLISION'. Otherwise, the nullrdc calls another function called 'NETSTACK_RADIO.transmit' to send the already prepared packet. Next, the nullrdc layer waits for ACK packet with a time called *macAckWaitDuration*. If the ACK is received during the wait time, the nullrdc waits again for a time called 'AFTER _ACK_DETECTED_WAIT_TIME' ($T_{A\_A\_D}$) and then returns 'TX_OK' to the CSMA layer. Otherwise, if *macAckWaitDuration* time ends and ACK is not received, the nullrdc returns 'TX_NOACK'. When the CSMA layer gets 'TX_OK', it dequeues the successful transmitted packet and sends the next packets. Otherwise, if it gets 'TX _COLLISION' or 'TX_NOACK', the CSMA waits for a random backoff time in the range $[time, time{+}2^{BE} \times time]$ where $time$ is channel check interval which equals $1/channel\ check\ rate$. Then, after the backoff time ends, the CSMA layer retransmits

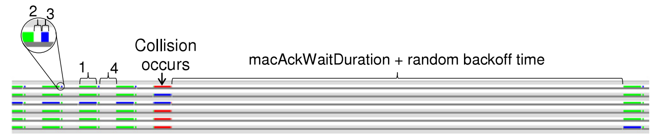


Fig. 4. TimeLine of 6 motes in Contiki OS.

the packet again. The MAC layer makes *macMaxFrameRetries* retransmission attempts and if unsuccessful, the packet is dropped. When the packet is received, it waits for a time called $turnaround\ time$ and then it sends the ACK packet.

Fig. 4 shows the TimeLine of 6 motes where one of them sends packets to others. Clearly, we can see the packet transmission time, turnaround time, ACK transmission time and wait time which includes $T_{A\_A\_D}$ as well as other times. Also, we can see that when a collision occurs (two nodes transmit at the same time), the collided nodes wait for *macAckWaitDuration* plus a random backoff time before they try to transmit again.

Considering that a collision does not occur, the maximum effective data rate, EDR_{max}, that Contiki 3.0 OS can support is as follows:

$$EDR_{max} = \frac{N}{T_{nocoll}} \tag{16}$$

where N is the data packet length (in bits) and T_{nocoll} is the actual time needed to transmit one data packet without collision. The maximum data packet length that IEEE 802.15.4 link can support is 127 bytes and T_{nocoll} is calculated as follows:

$$T_{nocoll} = T_{data} + turnaround\ time + T_{ACK} + T_{wait} \tag{17}$$

where T_{data} and T_{ACK} are the amount of time required to transmit data packet and ACK packet respectively.

$Turnaround\ time$ is the time required to switch between transmit and receive, or vice versa and T_{wait} includes $T_{A\_A\_D}$ and other times as shown in Fig. 4. Thus, EDR_{max} is calculated as:

$$EDR_{max} = \frac{127 \times 8}{(4.256 + 0.192 + 0.288 + 3.7)\ \text{ms}} \approx 120\ \text{kbps}$$

In practice, transmissions typically suffer collisions. When a collision does occur, the actual data rate goes down as the probability of collision increases in the network. When a collision occurs, the mote retransmits the collided packet. This takes extra time which includes transmission time of the collided data packet, *macAckWaitDuration* and random backoff time as shown in Fig. 4. Thus, the actual data rate, ADR, with a probability of collision of $P_{collision}$ is as follows:

$$ADR = \frac{N}{(1-P_{collision})T_{nocoll} + P_{collision}T_{coll}} \tag{18}$$

where T_{coll} is the actual time needed to transmit one data packet within collision and one retransmission attempt and is calculated as:

$$T_{coll} = T_{data} + macAckWaitDuration + T_{backoff} + T_{nocoll} \tag{19}$$

For example, with a probability of collision of 5%, the actual data rate can be calculated as follows:

$$ADR = \frac{127 \times 8}{[0.95 \times 8.436 + 0.05(4.256 + 0.4 + 125 + 8.436)]\ \text{ms}} \approx 68\ \text{kbps}.$$

For more information about the values used in EDR_{max} and ADR equations (e.g. $T_{data} = 4.256$ ms), please refer to [20–22].

When a node enters a backoff period and there are other active nodes within transmission range that have data packets to send. During this backoff period other active nodes still utilize this time by sending their data packets. Thus, as there are active nodes during a backoff period of a collided node, the channel time is utilized as much as possible and therefore the actual channel capacity increases. Also, as the active nodes can detect the idle time of the wireless channel quickly, the channel utilization will be high and therefore, the actual channel capacity increases. Overall, the actual channel capacity is not constant and it varies according to network circumstances. The actual channel capacity is affected by many factors such as probability of collision, number of active nodes and the utilization rate of idle wireless channel time by active nodes. In [20,21], Sun et al. have developed effective channel capacity estimation of IEEE 802.15.4 beaconless mode without taking account of the random backoff time and collision occurrence. Also, in [22], Latré et al. have determined throughput of unslotted IEEE 802.15.4 with unreal assumptions (no losses due to collisions, no packets are lost due to buffer overflow, perfect channel with bit error rate of zero). In our modelling validation, we estimate the actual channel capacity value based on our simulation results.

6. Simulation results

In this section, we present simulation results by using Contiki 3.0 OS and Cooja simulator to validate our buffer loss probability modelling for varying number of leaf nodes, buffer sizes and various offered loads. The protocols and simulation parameters used in the simulation are shown in Table 1 ($macMinBE$ and $macMaxBE$ values were chosen based on the default settings of Contiki OS). The total duration time of each simulation is set to be 60 s and during the simulation time, each leaf node sends data packets periodically to the sink node with an offered load of 32 packet/s.

In the first scenario, we change the number of leaf nodes to 2, 4, 6, 8 and 10 where each leaf node's application generates packets with an average rate of 32 packet/s. Fig. 5 shows the average number of dropped packets per second due to buffer overflow in the leaf node and the intermediate node estimated by simulation and using analytical modelling. From this figure, we can observe a good agreement between simulation and analytical results. Also, we can see that as the number of leaf nodes increases in the network, the number of lost packets due to buffer overflow increases in both leaf node and intermediate node. The reason is that as the number of leaf nodes increases, the portion of channel capacity for each node is reduced and therefore the departure rate of each node becomes lower and the probability of buffer loss becomes high.

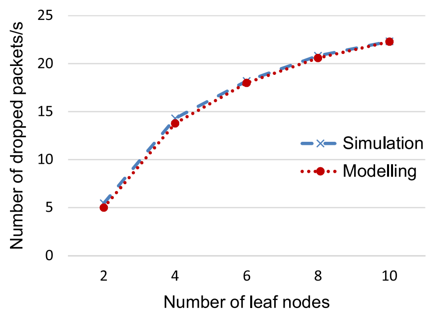
Table 1
Protocol stack and simulation parameters.

Protocol	Parameter value
RPL	Objective function = MHROF
SICSlowpan	Compression method = HC06
CSMA (MAC layer) nullrdc (RDC layer) 802.15.4 (framer)	Buffer size = 10 packets *macMaxFrameRetries* = 3 *channel check rate* = 8 Hz *macAckWaitDuration* = 0.4 ms $T_{A\_A\_D}$ = 0.6667 ms *macMinBE* = 0 *macMaxBE* = 3 Frame size = 127 bytes MAC ACK = enabled
CC2420 RF	*turnaround time* = 0.192 ms

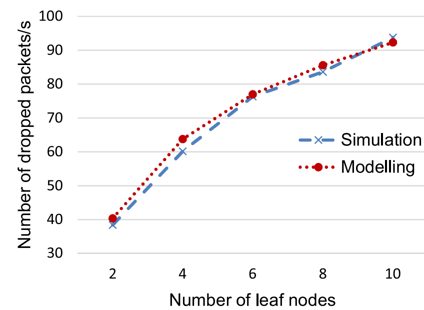
Next, in the second scenario, we set the number of leaf nodes to 5 and we changed the buffer size to 5, 10, 15 and 20 packets. Fig. 6 shows the average number of buffer loss packets in the leaf node and the intermediate node per second. We notice close correlation between simulation and modelling results. It is clear that as the buffer size increases, the average number of lost packets at leaf node's buffer decreases while it increases in the intermediate node. The reason is that when the buffer size is increased in the leaf node, the probability of buffer loss decreases and the leaf node's departure rate increases. As the departure rate of leaf node is increased, the arrival rate at the intermediate node is increased and therefore the probability of buffer loss becomes higher.

In the last scenario, we changed the offered load to 1, 2, 4, 8, 16 and 32 packet/s and set the number of leaf nodes to 5. Fig. 7 shows the number of dropped packets at the buffers of the leaf and intermediate nodes every second with different offered loads. From this figure, we can see matched results between simulation and analytical modelling. Also, we can see that when the offered load is increased, the average number of buffer dropped packets increases in the leaf and intermediate nodes.

Finally, Fig. 8 shows the average number of received packets at the sink node every second for the three scenarios above. From the figure, we notice that the simulation and analytical results have the same trend and a good consistency. Also, we can see that the number of received packets at the sink increases with: decreasing the number of leaf nodes, increasing the buffer size and increasing the offered load until it reaches a certain rate (4 packet/s) after that the number of received packets at the sink starts decreasing. Overall, the scenarios show that the analytical modelling results have a good agreement with the simulation results. Also, the simulation results show that our analytical modelling of congestion accurately models the buffer loss probability and the average number of received packets at the sink node. However, the main difference between analytical model and simulation model is the assumption the leaf node has a double portion of the wireless channel as compared to



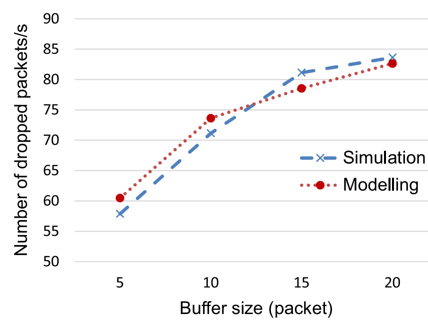
(a) Leaf node.



(b) Intermediate node.

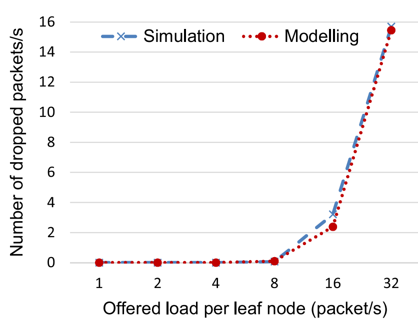
Fig. 5. Dropped packets for varying number of leaf nodes.

(a) Leaf node.

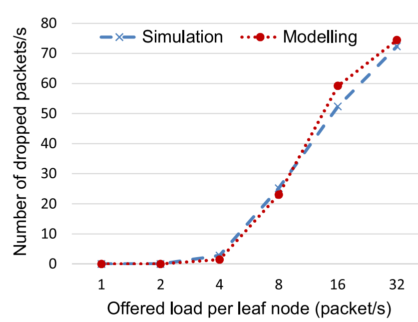


(b) Intermediate node.

Fig. 6. Number of dropped packets with different buffer sizes.



(a) Leaf node.



(b) Intermediate node.

Fig. 7. Number of dropped packets with various offered loads.

the intermediate node which will not always hold with the simulation model.

7. Conclusion

In this paper, we have presented an analytical model for 6LoWPAN network in the presence of congestion using Markov chain analysis and queuing theory. We have derived the expressions for the buffer loss probability and throughput at the sink. Also, we have calculated the IEEE 802.15.4 actual channel capacity under an unslotted CSMA-CA with and without collisions based on Contiki 3.0 implementation. Simulation results show that our analytical modelling of congestion has an accurate and good matching with the simulation for different scenarios and various parameters. Also, simulation results show

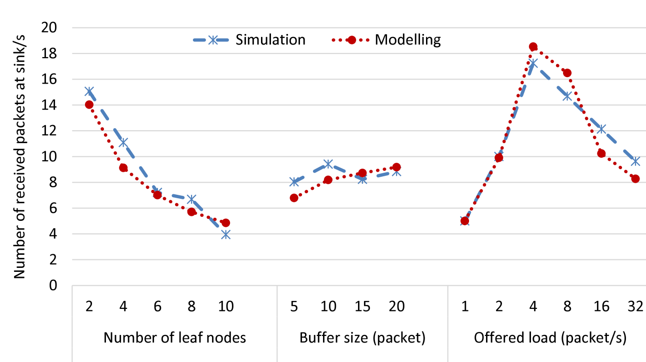


Fig. 8. Average number of received packets at sink node.

that: (i) As the number of leaf nodes increases, buffer overflow increases in the network, (ii) As buffer size is increased, buffer overflow at the leaf node decreases while it increases at the intermediate node, and (iii) As offered load is increased, the number of dropped packets increases in the leaf and intermediate nodes.

Conflict of interest

The authors declare that there is no conflict of interest in this paper.

References

[1] Z. Shelby, C. Bormann, 6LoWPAN: The Wireless Embedded Internet, John Wiley & Sons, 2009.

[2] S. Li, L. Da Xu, S. Zhao, The internet of things: a survey, Inf. Syst. Front. 17 (2) (2015) 243–259.

[3] N. Kushalnagar, G. Montenegro, C. Schumacher, IPv6 over low-power wireless personal area networks (6LoWPANs): Overview, assumptions, problem statement, and goals, in: Internet Engineering Task Force (IETF), RFC 4919, 2007.

[4] A. Ghaffari, Congestion control mechanisms in wireless sensor networks: A survey, J. Netw. Comput. Appl. 52 (2015) 101–115.

[5] H. Hellaoui, M. Koudil, Bird flocking congestion control for CoAP/RPL/6LoWPAN networks, in: Workshop on IoT Challenges in Mobile and Industrial Systems, ACM, 2015, pp. 25–30.

[6] A.P. Castellani, M. Rossi, M. Zorzi, Back pressure congestion control for CoAP/6LoWPAN networks, Ad Hoc Netw. 18 (2014) 71–84.

[7] V. Michopoulos, L. Guan, G. Oikonomou, I. Phillips, DCCC6: Duty cycle-aware congestion control for 6LoWPAN networks, in: International Conference on Pervasive Computing and Communications Workshops, IEEE, 2012, pp. 278–283.

[8] V. Michopoulos, L. Guan, G. Oikonomou, I. Phillips, A comparative study of congestion control algorithms in IPv6 wireless sensor networks, in: International Conference on Distributed Computing in Sensor Systems and Workshops, DCOSS, IEEE, 2011, pp. 1–6.

[9] H.A.A. Al-Kashoash, Y. Al-Nidawi, A.H. Kemp, Congestion-aware RPL for 6L0WPAN networks, in: Proceeding of Wireless Telecommunications Symposium, WTS, IEEE, 2016, pp. 1–6.

[10] H.A.A. Al-Kashoash, Y. Al-Nidawi, A.H. Kemp, Congestion analysis for low power and lossy networks, in: Wireless Telecommunications Symposium, WTS, IEEE, 2016, pp. 1–6.

[11] H.A.A. Al-Kashoash, M. Hafeez, A.H. Kemp, Congestion control for 6LoWPAN networks: A game theoretic framework, IEEE Internet Things J. 4 (3) (2017) 760–771.

[12] H.A. Al-Kashoash, H.M. Amer, L. Mihaylova, A.H. Kemp, Optimization based hybrid congestion alleviation for 6LoWPAN networks, IEEE Internet Things J. (2017).

[13] F. Gebali, Analysis of Computer Networks, Springer, 2015.

[14] S. Keshav, Mathematical Foundations of Computer Networking, Addison-Wesley, 2012.

[15] A. Dunkels, B. Grönvall, T. Voigt, Contiki - a lightweight and flexible operating system for tiny networked sensors, in: 29th Annual International Conference on Local Computer Networks, IEEE, 2004, pp. 455–462.

[16] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, T. Voigt, Cross-level sensor network simulation with COOJA, in: 31st Conference on Local Computer Networks, IEEE, 2006, pp. 641–648.

[17] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, RPL: IPv6 routing protocol for low-power and lossy networks, Internet Engineering Task Force (IETF), RFC 6550, 2012.

[18] IEEE 802.15.4, Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), 2006.

[19] P. Di Marco, P. Park, C. Fischione, K.H. Johansson, Analytical modeling of multi-hop IEEE 802.15. 4 networks, IEEE Trans. Veh. Technol. 61 (7) (2012) 3191–3208.

[20] T. Sun, L.-J. Chen, C.-C. Han, G. Yang, M. Gerla, Measuring effective capacity of IEEE 802.15. 4 beaconless mode, in: WCNC, IEEE, 2006, pp. 493–498.

[21] T. Sun, L.-J. Chen, G. Yang, M. Sanadidi, M. Gerla, Senprobe: Path capacity estimation in wireless sensor networks, in: The Wireless Traffic Measurements and Modeling Workshop (SenMetrics), Citeseer, 2005.

[22] B. Latré, P.D. Mil, I. Moerman, B. Dhoedt, P. Demeester, N.V. Dierdonck, Throughput and delay analysis of unslotted IEEE 802.15. 4, J. Netw. 1 (1) (2006) 20–28.