This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2766783, IEEE Transactions on Industrial Informatics

Novel Power Management Scheme and Effects of Constrained On-node Storage on Performance of MAC Layer for Industrial IoT Networks

M. P. R. S. Kiran, V. Subrahmanyam, and P. Rajalakshmi

***Abstract*—In this paper, we propose a novel IEEE 802.15.4 MAC power management scheme that achieves the user specified reliability with minimal power consumption at the node. Also, we develop an accurate mathematical model to analyze the effects of constrained on-node memory for sensed data storage on the MAC layer performance. We use 3D Markov chain and M/G/1/K queue to model the IEEE 802.15.4 MAC and on-node packet queue, respectively. By formulating the precise packet service time, the reliability, packet queue overflow losses, delay and power consumption of the node are analyzed. When compared with simulations and the real-time test bed, the proposed model achieves an accuracy of 97% and 94%, respectively. Also, the performance analysis shows that the proposed power management scheme provides energy savings of up to 74.82%.**

***Index Terms*—IEEE 802.15.4, 3D Markov chain, M/G/1/K queue, Internet of Things, Performance analysis, MAC layer.**

I. Introduction

THE Internet of Things (IoT) is proliferating into industrial applications at a rapid pace. Many industries such as power plants, manufacturing and so forth, demand minimal manual intervention and are moving towards automation [1]–[3]. Although multiple technologies such as WiFi, IEEE 802.15.4, Bluetooth exist, IEEE 802.15.4 has gained prominence in real-time industrial deployments due to its low power consumption, better range and improved security [4]. IEEE 802.15.4 is one of the widely used IoT communication technologies for industrial deployments such as electrical line monitoring in smart grids [5], fouling detection in ducts system [6], and intelligent load management [7] due to its low power consumption [8]. One primary challenge wireless monitoring or automation applications face when compared to traditional wired systems is to meet the stringent performance requirements. The lack of analytical methods to evaluate accurate performance hinders their adoption in many industrial applications [9]. Models existing in the literature ignore queue overflow losses and the packet waiting time in the queue. These measures play a significant role while analyzing the performance of node with limited storage. Our primary contributions in this paper include:

- Model the constrained on-node storage as M/G/1/K queue to formulate the packet overflow losses.
- Model the slotted IEEE 802.15.4 CSMA/CA using 3D Markov chain and derive the accurate service times for a given packet arrival rate.
- Using the developed analytical model, we propose a novel duty cycle management scheme which determines the amount of active period required by a node within a superframe to achieve the user defined reliability with minimal power consumption.
- Validation of the proposed model using both simulation and real-time test bed.

The authors are with the Department of Electrical Engineering, Indian Institute of Technology Hyderabad, Hyderabad 502285, India (e-mail: ee12m1021@iith.ac.in; ee14mtech11012@iith.ac.in; raji@iith.ac.in).

Conventional modeling approaches derive performance measures such as reliability, channel congestion, and delay for successful transmission using a suitable Markov chain [10]–[13]. Burratti *et. al.* [10] focused on analyzing the performance of unslotted CSMA/CA by formulating a mathematical model. In [11], authors analyzed the performance of slotted CSMA/CA without retransmissions using a 2D Markov chain and Yung *et. al.* [12] analyzed the performance of IEEE 802.15.4 CSMA/CA in the presence of superframe but did not consider the packet and acknowledgment lengths. Park *et. al.* [13] developed a 3D Markov chain for analyzing the slotted CSMA/CA along with retransmissions and packet length. In all these models [10]–[13], the authors assumed instantaneous traffic generation for analysis and did not consider the packet queue. In general, if the service offered by MAC is slow, the on-node storage eventually fills up with sensed data, leading to increased loss of packets. Misic *et. al.* [14] analyzed the performance of slotted CSMA/CA with limited packet queue size but considered infinite retransmissions. Also, the analysis diverges significantly from the simulations. Anwar *et. al.* [15] analyzed the IEEE 802.15.4 TDMA with limited queue size. Hence, to the best of our knowledge, this is the first study to analyze the performance of slotted CSMA/CA with limited on-node storage.

Industrial applications also require better lifetime in addition to reliability. The lifetime of IEEE 802.15.4 node strictly depends on its duty cycle. Multiple mechanisms such as ADCA, AAOD, DSAA, DCA, and AMPE are proposed in the literature to improve the lifetime [16]. Gateway using these mechanisms optimize the duty cycle of the leaf nodes adaptively by analyzing the network traffic and remaining packets in the queue of every node. These techniques levy additional computation burden on gateway and user does not have direct control on the achieved reliability. Hence, we propose a novel duty cycle management mechanism that achieves user specified reliability with minimal duty cycle. We are convinced that this theoretical study, as well as illustrated industrial experiments, can improve the efficiency of IEEE

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2766783, IEEE Transactions on Industrial Informatics

TABLE I: Commercial motes with their memory footprints

Mote	Platform	Memory
CoSeN/Mantaro Blocks	Atmel AT86RF231	4 kB
MicaZ	Chipcon CC2420	4 kB
IITH Mote	Atmel AT86RF230	8 kB
Iris	Atmel AT86RF230	8 kB
BPart	CC2540	8 kB
Shimmer	Chipcon CC2420	10 kB
TMote Sky/TelosB	Chipcon CC2420	10 kB

802.15.4 based industrial applications.

The remainder of this manuscript is organized as follows. Section II describes the motivation for this study and Section III provides the system model. Section IV describes the mathematical formulation of the IEEE 802.15.4 MAC and the analysis on packet queue is performed in Section V. Section VI provides the proposed duty cycle management framework and Section VII describes the developed simulation model. In Section VIII, we provide the details of the test bed implementation and the performance of proposed model using both simulations, and real-time test bed is discussed in Section IX. Finally, Section X concludes this paper by discussing the future scope.

II. Motivation

The IEEE 802.15.4 motes have a constrained on-node storage as shown in Table I and can queue a limited number of packets beyond which it discards the data. With the increase in data traffic, the packets spend more time in the queue and queue overflow losses increase. Analytical models significantly help in understanding the performance of the network and aid in choosing the suitable MAC parameters to reduce data losses. As the existing models did not consider the constrained on-node storage while analyzing the performance, the reliability and latency are inaccurate. Hence, we performed an accurate mathematical formulation to analyze the network performance and proposed a novel power management scheme to achieve the user required reliability with improved lifetime.

III. System model

Fig. 1(a) shows the star network topology considered with gateway aggregating data from all the leaf nodes. A leaf node serves an available packet in its on-node storage at imbedded points that include instances just after a node is in the Idle state and after the completion of CSMA/CA. Serving a packet signifies the attempt to transmit it by accessing the channel using CSMA/CA (can be successful or unsuccessful) after which the node empties the packet from the queue. Fig. 1(c) shows the M/G/1/K queue model resembling finite on-node storage of K packets. Although the data arrival process is dependent on the underlying application, for analysis we model it as a Poisson process, and the service follows General process strictly depending on the CSMA/CA parameters.

Discrete time Markov chain is a very useful mathematical tool for analyzing the stochastic behavior of any discrete time systems with finite number of states [17]. Many models existing in the literature for analyzing the performance of MAC layer made use of Markov chains due to its accuracy [10]–[13]. The behavior of MAC can be characterized using a finite

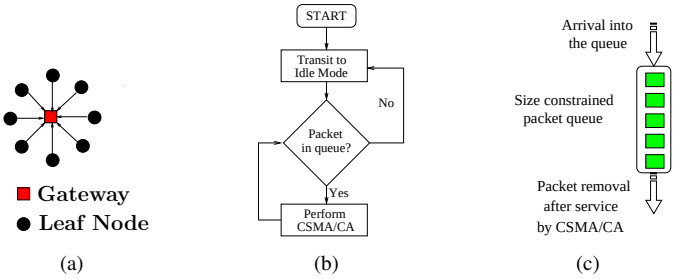


Fig. 1: (a) Network model. (b) State model of a leaf node. (c) Queue model resembling on-node storage

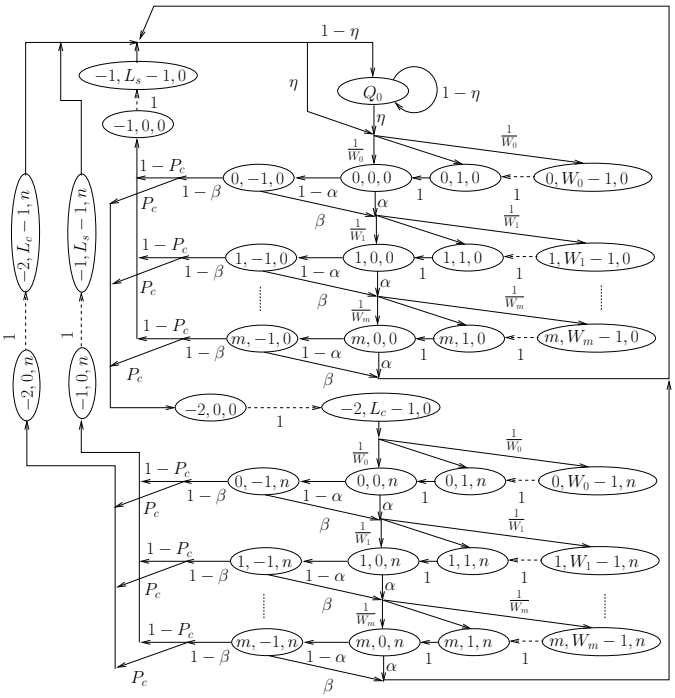


Fig. 2: 3D Markov chain model for leaf node.

number of states. For a given state space $S = s_1, s_2, ..., s_m$, let X_t be a random variable which defines the state of the node at time t. The set of random variables X_t will form a Markov chain, if the probability of the next state being s_j ($X_{t+1} = s_j$), depends only on the current state X_t. For a better understanding on Markov chains, we request the readers to refer [17]. As these state transitions in the slotted CSMA/CA can occur only at the beginning of a backoff period, the time intervals of the Markov chain are discrete resulting in a discrete time Markov chain. Fig. 2 shows the 3D Markov chain model for leaf node where state Q_0 denotes Idle state and rest of the states indicate CSMA/CA. The CSMA/CA states are guided by the three stochastic processes *s(t), c(t)* and *r(t)* which indicate the backoff stage, backoff delay counter and number of retransmissions, respectively at any given time *t*. η indicates the probability of packet availability in the queue. At the beginning of the CSMA/CA, the number of backoff slots is initialized to a value randomly chosen from $[0, W_0 - 1]$ where $W_0 = 2^{macMinBE}$ with $macMinBE$ being minimum backoff

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2766783, IEEE Transactions on Industrial Informatics

TABLE II: Description of Markov chain parameters

Parameter	Description
CSMA/CA Parameters	
L_c	Packet collision duration
L_s	Successful packet duration
L_p	Packet duration
L_{ack}	Acknowledgment duration
$actPeriod$	Active period of superframe
n	Max. no. of retries
m	No. of backoff stages
$macMinBE$	Min. backoff exponent
$macMaxBE$	Max. backoff exponent
P_c	Probability for collision
N	No. of nodes in the network
P_{Q_0}	Probability of node residing in Idle state
$b_{0,0,0}$	Probability of node residing in the backoff state (0,0,0)
τ	Probability of node attempting CCA1 in an arbitrary time
η	Probability of at least one packet availability in the queue
R_{eff}	Effective reliability of the node with CSMA/CA and queue
R	Reliability of the CSMA/CA without queue
P_{dr}	Probability of packet being discarded due to exceeded retry limits
P_{dc}	Probability of packet being discarded due to channel access failure
$b_{i,k,j}$	Probability of the node residing in backoff stage (i,k,j)
$g_{Succ}(t)$	PDF of delay for successful packet transmission
$h_{Coll}(t)$	PDF of delay for packet failure due to exceeded retransmissions
$i_{Backoff}(t)$	PDF of delay for packet failure due to exceeded backoff stages
$T_{Succ}(z)$	PGF of delay for successful packet transmission
$T_{Coll}(z)$	PGF of delay for collision packet transmission
$T_{Backoff}(z)$	PGF of service time for packet failure due to exceeded backoff stages
D_{tot}	Average delay for successful packet transmission with queue delay
D_{Succ}	Average delay for successful packet transmission
D_{Coll}	Average delay for collision packet transmission
$D_{Backoff}$	Average delay for packet failure due to exceeded backoff stages
D_{eff}	Average service time by CSMA/CA
P	Total power consumption of the node
Queuing Parameters	
$M/G/1/K$	M-Arrival process follow Poisson, G-Service follows General process and one server with maximum queue capacity of K elements
P_B	Blocking probability of the queue
ρ	Net offered load for the queue
ρ_c	Carried load for the queue
λ	Packet arrival rate in packets per unit backoff slot
λ_p	Packet arrival rate in packets per second (3125×λ)
$qLim$	On-node storage size in terms of packets
q_k	Probability of imbedded point is due to node in Idle state and finds k packets in the queue
r_k	Probability of imbedded point is due to successful packet transmission and finds k packets in the queue
s_k	Probability of imbedded point is because of packet discard due to exceeded retransmissions and finds k packets in the queue
t_k	Probability of imbedded point is because of packet discard due to exceeded maximum backoff stages and finds k packets in the queue
f_j	Probability of j packet arrivals while node spends in Idle state
g_j	Probability of j packet arrivals during successful packet transmission
h_j	Probability of j packet arrivals during packet discard due to exceeded retransmissions
i_j	Probability of j packet arrivals during packet discard due to exceeded backoff stages

exponent and $macMaxBE$ is maximum backoff exponent. The maximum backoff stages in any given transmission window is limited to $m+1$ (represented using *(0,k,j)* to *(m,k,j)*) and n indicates the maximum number of retransmissions allowed (one normal transmission and n retransmissions = total $n+1$ transmissions) whereas the two Clear Channel Assessments CCA1 and CCA2 are represented using *(i,0,j)* and *(i,-1,j)* respectively. The states *(-1,k,j)* and *(-2,k,j)* indicate successful packet transmission and packet collision slots respectively with L_s and L_c constants indicating the length of successful packet transmission and collision duration respectively. Both L_s and L_c include packet length (L_p), Inter Frame Spacing (IFS, usually 1 unit backoff period) and acknowledgment duration (L_{ack}). Rest of the slots indicate backoff slots where node spends in idle mode. As the node senses channel in CCA1 and CCA2, α and β indicate the respective probabilities for sensing the channel busy. P_c indicates the probability of collision for a transmitted packet. Table II briefly describes the parameters used for modeling.

For the mathematical formulation, we consider below acceptable assumptions which are in agreement with the existing popular studies in the literature [13].

- We assume the traffic overhead due to periodic beacon frame communication from the gateway to be negligible.
- Usually in IEEE 802.15.4 MAC, at the end of every superframe, if the remaining time is insufficient for the frame transmission, the node pauses its backoff countdown timer and resumes in the next superframe. In this study, we assume the superframe is continuous and the error introduced with this assumption is negligible as performed in [13].

IV. Mathematical model

From Fig. 2, the probability of a node residing in Q_0 is

$$P_{Q_0} = P_{Q_0}(1-\eta) + b_{0,0,0}(R + P_{dr} + P_{dc})(1-\eta) \quad (1)$$

where $b_{0,0,0}$ indicates the probability of node residing in $(0,0,0)$. R, P_{dr} and P_{dc} whose relation is given by (2), indicate the probabilities for successful packet transmission, packet being discarded due to exceeded retry limits and discarded due to channel access failure respectively. Using (1) and (2), one can express P_{Q_0} in terms of $b_{0,0,0}$ as given by (3).

$$R + P_{dr} + P_{dc} = 1 \quad (2)$$

$$P_{Q_0} = \left(\tfrac{1-\eta}{\eta}\right) b_{0,0,0} \quad (3)$$

Using the normalization property of Markov chain given by (4), the probability of a node residing in $b_{0,0,0}$ can be derived as shown in (5), where x and y are given by (6). As deriving these probabilities is not our primary contribution in this paper, we advise the readers to refer Appendix I or [13] for more insights into the derivation of (5).

$$\sum_{i=0}^{m}\sum_{k=0}^{W_i-1}\sum_{j=0}^{n} b_{i,k,j} + \sum_{i=0}^{m}\sum_{j=0}^{n} b_{i,-1,j} + \sum_{j=0}^{n}\left(\sum_{k=0}^{L_s-1} b_{-1,k,j} + \sum_{k=0}^{L_c-1} b_{-2,k,j}\right) + P_{S_0} = 1 \quad (4)$$

$$b_{0,0,0} = \left[\frac{1}{2}\left(\frac{1-(2x)^{(m+1)}}{1-2x}W_0 + \frac{1-x^{(m+1)}}{1-x}\right)\frac{1-y^{(n+1)}}{1-y} + (1-\alpha)\frac{1-x^{(m+1)}}{1-x}\frac{1-y^{(n+1)}}{1-y} + (L_s(1-P_c) + L_c(1-P_c))(1-x^{(m+1)})\frac{1-y^{(n+1)}}{1-y} + \frac{1-\eta}{\eta}\left(\frac{x^{(m+1)}(1-y^{(n+1)})}{1-y} + P_c(1-x^{(m+1)})y^n + (1-P_c)\frac{(1-x^{(m+1)})(1-y^{(n+1)})}{1-y}\right)\right]^{-1} \quad (5)$$

$$x = \alpha + (1-\alpha)\beta; \qquad y = P_c(1-x^{m+1}) \quad (6)$$

With the above state probabilities, the reliability of CSMA/CA (R) for a node which determines the fraction of packets that

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2766783, IEEE Transactions on Industrial Informatics

are successfully transmitted can be expressed as

$$R = 1 - P_{dr} - P_{dc} = 1 - \frac{x^{m+1}(1-y^{n+1})}{1-y} - y^{n+1} \tag{7}$$

Now, we model the service time distribution which is one of the primary contributions of this paper. Depending on the service type, service times in IEEE 802.15.4 CSMA/CA can be classified into three categories: (1) delay incurred for a successful packet transmission, (2) delay incurred for a packet failure due to exceeded retransmissions, and (3) delay incurred for a packet failure due to channel access failure. We begin with modeling the Probability Generating Function (PGF) of delay for successful packet transmission which is a power series representation in variable z of the probability mass function for the delay. In CSMA/CA if the channel is free at the end of the first backoff stage, the node immediately starts transmitting the packet. The delay incurred in this case is given by (8). Assuming the node selects the backoff slot $b_{0,k,0}$, it has to wait for $k+1$ slots before accessing the channel. The denominator term $(1-x^{m+1})$ in (8) serves as the condition for successful channel access in the current transmission, and the numerator term $(1-x)$ corresponds to the successful channel access in the first backoff stage.

$$T_b^{(0)}(z) = \sum_{k=1}^{W_0} \frac{1-x}{1-x^{m+1}} \frac{1}{W_0} z^{k+1} \tag{8}$$

$T_{b,i}(z)$ given in (9) represents the PGF for delay incurred in the i^{th} backoff stage alone if the channel access performed by the node is unsuccessful. Assuming the node selected a backoff waiting time of $k-1$ slots, it transits to backoff state $b_{i,k-1,j}$). Then the node will wait for k slots if it incurs channel access failure in CCA1 else for $k+1$ slots if CCA2 is sensed busy before proceeding to the next backoff stage. The term $(\alpha z^k + (1-\alpha)\beta z^{k+1})$ in (9) represents the PGF of node waiting duration in i^{th} backoff stage where the term αz^k represents the event of channel busy in CCA1 resulting in k slots delay and $(1-\alpha)\beta z^{k+1}$ represents the event of channel free in CCA1 and busy in CCA2 resulting in $k+1$ slots delay. The summation includes all possible backoff waiting times in a backoff stage i and the denominator x is for the condition that channel access is unsuccessful and W_i in denominator accounts for random backoff waiting time selection.

$$T_{b,i}(z) = \sum_{k=1}^{W_i} \frac{1}{W_i x} (\alpha z^k + (1-\alpha)\beta z^{k+1}) \tag{9}$$

If the node gets free channel access in i^{th} backoff stage (i.e. $b_{i,0,0}$), the PGF of delay incurred during $[0, i-1]$ backoff stages is given by (10) and delay incurred in the i^{th} backoff stage alone is given by (11).

$$D_{[0,i-1]}(z) = \prod_{l=0}^{i-1} T_{b,l}(z) \tag{10}$$

$$D_i(z) = \sum_{k=1}^{W_i} \frac{z^{k+1}}{W_i} \tag{11}$$

Using (9), (10) and (11), the PGF of delay for successful channel access after the first backoff stage can be obtained using (12). A node can get channel access in any of the m backoff stages available and index i covers all such possibilities. For a successful transmission, the node should access channel within the m backoff stages available, and the term $x^i(1-x)$ gives the probability for a node to access channel in the backoff stage i ($i.e.$ the node obtained channel in $b_{i,-1,0}$).

$$T_b^{(1)}(z) = \sum_{i=1}^{m} \left[\frac{x^i(1-x)}{1-x^{m+1}} D_{[0,i-1]}(z) D_i(z) \right] \tag{12}$$

Equation (13) provides the combined PGF of delay for channel access in a single transmission. The first term in equation (13) represents the PGF for delay if the node achieves channel access in the first backoff stage ($i.e.$ in $b_{0,-1,0}$) and the second summation covers all other possibilities of the node accessing the channel after the first backoff stage.

$$T_b(z) = T_b^{(0)}(z) + T_b^{(1)}(z) \tag{13}$$

Finally, the PGF for the delay due to successful packet transmission is expressed in (14). The index j accounts for the number of retransmissions attempted before the successful transmission. Assuming the successful transmission happened in the j^{th} retransmission, the delay incurred will include total j collision durations and one successful packet transmission duration. The term $y^j(1-y)$ indicates the probability of successful transmission in j^{th} retransmission attempt without any collision and $\left(T_b(z)z^{L_c}\right)^j$ is the PGF for delay incurred during j backoff waiting times including j collisions. Whereas, the term $T_b(z)z^{L_s}$ accounts for delay incurred during the final successful transmission including the backoff waiting time.

$$T_{Succ}(z) = \sum_{j=0}^{n} \left[\frac{y^j(1-y)}{1-y^{n+1}} T_b(z) z^{L_s} \left(T_b(z)z^{L_c}\right)^j \right] \tag{14}$$

Approaching with similar hypothesis, the PGF for service time of packet failure due to exceeded retransmissions is

$$T_{Coll}(z) = \prod_{j=0}^{n} \left(T_b(z)z^{L_c}\right) = \left(T_b(z)z^{L_c}\right)^{n+1} \tag{15}$$

Finally, we derive the PGF for service time of packet failure due to exceeded backoff stages. If the node is unable to access the channel during the first transmission, it discards the packet after the last backoff stage, and the corresponding PGF for delay includes the total delay incurred in every backoff stage as given by (16).

$$T_{Backoff}^{(0)}(z) = \prod_{i=0}^{m} T_{b,i}(z) \tag{16}$$

If the node fails to access the channel in j^{th} retransmission, the total delay includes the delay incurred for first transmission along with subsequent $(j-1)$ retransmissions and the delay incurred in the j^{th} retransmission where the node discards packet due to channel access failure. In any given transmission, the PGF for delay incurred due to channel access and collision is given as $T_b(z)z^{L_c+L_{ack}}$ and the cumulative PGF for one transmission and $(j-1)$ retransmissions is given by (17). Equation (17) is a function of *j*, which can take values from *[1,n]* as the channel access failure can occur in any of the retransmission. The complete PGF of service time for packet failure due to exceeded backoff stages is given by (18). The first term in equation (18) represents the backoff expiry in the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2766783, IEEE Transactions on Industrial Informatics

first transmission window and the second summation covers all possibilities of backoff expiry in any of the retransmission.

$$T_{Backoff}(z,j) = \left(\prod_{u=1}^{j} T_b(z) z^{L_c} \right) \left(\prod_{i=0}^{m} T_{b,i}(z) \right) \quad (17)$$

$$T_{Backoff}(z) = \frac{1-y}{1-y^{n+1}} \left[T^{(0)}_{Backoff}(z) + \sum_{j=1}^{n} y^j T_{Backoff}(z,j) \right] \quad (18)$$

Using (14), the average CSMA/CA delay D_{Succ}, incurred by a successful packet transmission is

$$D_{Succ} = T'_{Succ}(z) \Big|_{z=1} \quad (19)$$

Similarly, using (15) and (18), average delay for the packet discard scenario due to exceeded retransmissions (D_{Coll}) and due to exceeded backoff stages ($D_{Backoff}$) can be found using (20) and (21) respectively.

$$D_{Coll} = T'_{Coll}(z) \Big|_{z=1} \quad (20)$$

$$D_{Backoff} = T'_{Backoff}(z) \Big|_{z=1} \quad (21)$$

V. Packet Queue Model Resembling on-node storage

To analyze the effect of queuing losses, we assume the packet generation follows Poisson arrival process with mean inter-arrival time $\frac{1}{\lambda}$ slots and the maximum queue limit of the node is $qLim$. Since the service times of IEEE 802.15.4 MAC do not follow any standard distribution, it is considered as General distribution and the on-node storage is modeled using $M/G/1/qLim$ queue with multiple vacations. The node spends in Idle state if there are no packets and is referred as a vacation in queuing generality. Our goal from this packet queue modeling is to determine the queue overflow losses and η. Let us denote q_k as the probability that the imbedded point is due to the node checking for packet availability in the Idle state and finds k packets in the queue (*i.e.* node finished a vacation and found *k* packets). Similarly let r_k, s_k and t_k be the probability that the node finds k packets in the queue at the imbedded points after the node's successful packet transmission, packet discard by node due to extended retries and packet discard by node due to extended backoff stages, respectively (*i.e.* the node has finished a service and found *k* packets in the queue). Let f_j be the probability of j packet arrivals while the node spends in the Idle state. Similarly, let g_j, h_j and i_j be the probabilities of j arrivals during the three service times due to node's successful packet transmission, packet discard by node due to extended retries and packet discard by node due to extended backoff stages respectively. Equation (22) gives the probability of j packet arrivals in t slots interval according to Poisson process. f_j, g_j, h_j and i_j can be calculated using equations (23)-(27), where $g_{Succ}(t), h_{Coll}(t)$ and $i_{Backoff}(t)$ are the distributions of service times for all the three scenarios, respectively. Since the packet arrival process is assumed to be following the Poisson distribution, in a given t slots duration, the probability for j packet arrivals can be derived using (22). For the calculation of f_j, we already know the duration of the Idle state, which is one slot (*i.e.* 320μs). Hence, f_j can be directly evaluated using (23). To derive g_j, h_j and i_j, as we do not know the corresponding service times, we make use of the service time distributions.

$$Pr(j) = \frac{(\lambda t)^j e^{-\lambda t}}{j!} \quad (22)$$

$$f_j = \lambda^j e^{-\lambda} \quad (23)$$

$$g_j = \int_0^\infty Pr(j)\ g_{Succ}(t)dt = \frac{1}{j!} \frac{dT^*_{Succ}(\lambda - \lambda s)}{ds} \quad (24)$$

$$h_j = \int_0^\infty Pr(j)\ h_{Coll}(t)dt = \frac{1}{j!} \frac{dT^*_{Coll}(\lambda - \lambda s)}{ds} \quad (25)$$

$$i_j = \int_0^\infty Pr(j)\ i_{Backoff}(t)dt = \frac{1}{j!} \frac{dT^*_{Backoff}(\lambda - \lambda s)}{ds} \quad (26)$$

In (24), $T^*_{Succ}(s)$ denotes the Laplace-Stieltjes transform for $T_{Succ}(z)$. As the fundamentally required probabilities are now derived, we can express the probability q_k as shown in (27) and (28). If a node is in the Idle state during the current imbedded point, the packet buffer of the node should be empty in the previous imbedded point. Hence, (27) formulates the scenario of packet buffer containing k packets in the current imbedded point resulted due to vacation. The term q_0 represents the previous imbedded point is due to vacation and the term $Rr_0 + P_{dr}s_0 + P_{dc}t_0$ represents the scenario where previous imbedded point is due to CSMA/CA service (can be a successful transmission, exceeded retransmissions or exceeded backoff stages) and found no packets in the queue. Although we have three different services namely successful transmission, exceeded retransmissions and exceeded backoff stages, in principle we have only one server which is CSMA/CA. Hence, we fuse these three service effects on the queue performance statistically into a single service (the term $Rr_0+P_{dr}s_0+P_{dc}t_0$) and only single kind of vacation (q_0). In (28), the probability for the queue to be fully occupied with packets while the node spends in the Idle state is given.

$$q_k = (q_0 + Rr_0 + P_{dr}s_0 + P_{dc}t_0) f_k; \quad \forall k \in [0, qLim - 1] \quad (27)$$

$$q_{qLim} = (q_0 + Rr_0 + P_{dr}s_0 + P_{dc}t_0) \sum_{k=qLim}^{\infty} f_k; \quad (28)$$

Similarly r_k, s_k and t_k can be expressed as (29), (30) and (31). In order to ease the formulation of η, let us consider γ_k expressed as shown in (32).

$$r_k = \sum_{j=0}^{k+1} (q_j + Rr_j + P_{dr}s_j + P_{dc}t_j) g_{k-j+1} \quad (29)$$

$$s_k = \sum_{j=0}^{k+1} (q_j + Rr_j + P_{dr}s_j + P_{dc}t_j) h_{k-j+1} \quad (30)$$

$$t_k = \sum_{j=0}^{k+1} (q_j + Rr_j + P_{dr}s_j + P_{dc}t_j) i_{k-j+1} \quad (31)$$

$$\gamma_k = \frac{q_k + Rr_k + P_{dr}s_k + P_{dc}t_k}{q_0 + Rr_0 + P_{dr}s_0 + P_{dc}t_0} \quad (32)$$

Upon solving (32) recursively starting from γ_0, one can generalize the probability γ_{k+1} as shown in equation (33). Using (32) and the normalization criteria given in (34), the probability of queue being empty at any given imbedded point

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2766783, IEEE Transactions on Industrial Informatics

can be derived as (35).

$$\gamma_{k+1} = \frac{\left(\gamma_k - f_k - \sum_{j=1}^{k} \gamma_j (Rr_k + P_{dr}s_k + P_{dc}t_k)\right)}{Rr_0 + P_{dr}s_0 + P_{dc}t_0} \tag{33}$$

$$\sum_{k=0}^{qLim} q_k + \sum_{k=0}^{qLim-1} (Rr_k + P_{dr}s_k + P_{dc}t_k) = 1 \tag{34}$$

$$q_0 + r_0 + s_0 + t_0 = 1 - \eta = \frac{1}{\sum_{k=qLim}^{\infty} f_k + \sum_{k=0}^{qLim-1} \gamma_k} \tag{35}$$

Blocking probability (P_B) or the probability for a generated packet not having space in the queue, can be derived using offered load (ρ) and carried load (ρ_c) as given in (36), where ρ and ρ_c are determined using (37) and (38). Here ρ_c is the probability that the server is busy at an arbitrary time and can be calculated by determining the fraction of time the node spends in service. ρ is a measure of traffic in the queue which can be given by Little's law (arrival rate multiplied by mean service time of the server) and D_{eff} represents the average service time.

$$P_B = \frac{\rho - \rho_c}{\rho} \tag{36}$$

$$\rho = \lambda * D_{eff} = \lambda(RD_{Succ} + P_{dr}D_{Coll} + P_{dc}D_{Backoff}) \tag{37}$$

$$\rho_c = \frac{\eta * D_{eff}}{(1-\eta)+\eta * D_{eff}} = \frac{\eta(RD_{Succ} + P_{dr}D_{Coll} + P_{dc}D_{Backoff})}{(1-\eta)+\eta(RD_{Succ} + P_{dr}D_{Coll} + P_{dc}D_{Backoff})} \tag{38}$$

Using (7) and (36), the total or effective reliability can be formulated as

$$R_{eff} = 1 - P_B - (1 - P_B)(1 - R) \tag{39}$$

In [18], several methods to derive the mean waiting time for a packet in the system are discussed. Using one of the methods in [18], the average number of packets in the queue at equilibrium can be expressed as

$$L_{avg} = \frac{\sum_{k=1}^{qLim-1} k(1-\eta)(\gamma_k - f_k)}{\lambda((1-\eta)+\eta(RD_{Succ} + P_{dr}D_{Coll} + P_{dc}D_{Backoff}))} + qLim\frac{\rho - \rho_c}{\rho} \tag{40}$$

Using (40), the average waiting time for a packet in the system can be given as

$$D_{avg} = \frac{L_{avg}}{\lambda(1 - P_B)} \tag{41}$$

Finally, the total delay for a successful packet transmission is

$$D_{tot} = D_{avg} - D_{eff} + D_{Succ} \tag{42}$$

VI. Duty Cycle management in IEEE 802.15.4

IEEE 802.15.4 when operated under beacon-enabled mode, has a superframe structure in which a node's operation life cycle is divided into active and inactive period. During the active period, the node behavior follows the state process shown in Fig. 2 and during inactive the node spends in sleep mode which we refer here as the Sleep state. Hence, lesser the active period of the node, higher is the energy conservation. The industrial applications usually are classified into different classes with each class having specified reliability requirements [19]. Therefore to achieve better energy savings, we tune the active period of a node depending on the application reliability requirement (R_{req}) provided. Assuming, $actPeriod$ as the fraction of the active period within the total superframe duration, reduction in active period will result in an increase of average service time for the packet. The effect of a decrease in the $actPeriod$ is similar to increase in λ for the node when observed from the perspective of channel congestion and reliability. For a better emphasis, let us consider an example where $actPeriod = 1$ and length of superframe is SD. The net load on the node now ($\rho_n^{actPeriod=1}$) is given as

$$\rho_n^{actPeriod=1} = \frac{\lambda}{SD} \tag{43}$$

If we now reduce the $actPeriod$ to 0.5, the net load ($\rho_n^{actPeriod=0.5}$) is as given below. We can clearly observe that reduction in $actPeriod$ and increase in λ result in a similar hypothesis.

$$\rho_n^{actPeriod=0.5} = \frac{\lambda}{actPeriod \times SD} = \frac{2\lambda}{SD} \tag{44}$$

Now considering λ_{eff} as the effective packet arrival rate with decrease in $actPeriod$, it can be expressed as

$$\lambda_{eff} = \frac{\lambda}{actPeriod} \tag{45}$$

Using (45), one can formulate the $actPeriod$ as

$$actPeriod = \frac{\lambda}{\lambda_{eff}} \tag{46}$$

To determine the λ_{eff} using the user required R_{req}, we make use of the Gradient-Descent algorithm. We define e as the error between R_{eff} and R_{req} as given in (47).

$$e = R_{eff} - R_{req} \tag{47}$$

The update rule for λ_{eff} is given in (48), where δ indicates the step size and l is the iteration.

$$\lambda_{eff}^{l+1} = \lambda_{eff}^{l} + \delta e \tag{48}$$

Fig. 3 depicts the steps involved in determining the $actPeriod$. Firstly, we calculate the maximum R_{eff} feasible using the default packet arrival rate (λ). If R_{req} exceeds R_{eff} the network cannot provide the user required reliability else we proceed with the Gradient-Descent based determination of λ_{eff}. Although we can initialize λ_{eff} to any random value for faster convergence, we initialize λ_{eff} to λ and determine the R_{eff}. We then calculate the error (e) and update the λ_{eff} until R_{eff} converges to R_{req}. Once λ_{eff} is calculated, using (46), we determine the $actPeriod$ which can then be used for network deployment in the field.

A. Power Consumption

With the proposed duty cycle management framework, we now derive the power consumption of the node. Considering P as the power consumption of the node, it consists of power consumed by the node when it is in Sleep (P_{Sleep}), Idle state (P_{Idle}), backoff state ($P_{Backoff}$), CCA (P_{CCA}) and transmission states (P_{Txn}).

$$P = (1 - actPeriod)P_{Sleep} + actPeriod(P_{Idle} + P_{Backoff} + P_{Txn} + P_{CCA}) \tag{49}$$

The power P_{Sleep} and P_{Idle} can be expressed as in (50) and (51) respectively. P_s and P_i indicate the average power consumption of the node in sleep and idle mode respectively.

$$P_{Sleep} = P_s \tag{50}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2766783, IEEE Transactions on Industrial Informatics

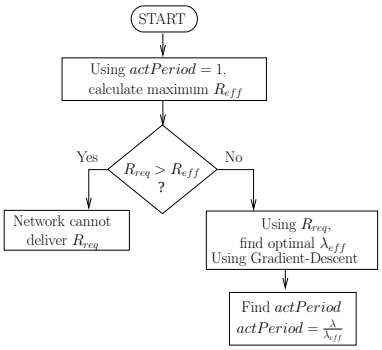


Fig. 3: Procedure to find $actPeriod$.

$$P_{Idle} = P_i Q_0 \tag{51}$$

Equations (52) and (53) expresses $P_{Backoff}$ and P_{Txn}.

$$P_{Backoff} = P_i \sum_{i=0}^{m} \sum_{k=1}^{W_i-1} \sum_{j=0}^{n} b_{i,k,j} \tag{52}$$

$$P_{Txn} = P_t \sum_{i=-2}^{-1} \sum_{k=0}^{L_p-1} \sum_{j=0}^{n} b_{i,k,j} + P_i \sum_{i=-2}^{-1} \sum_{j=0}^{n} b_{i,L_p,j} + \sum_{j=0}^{n} \sum_{k=L_p+1}^{L_p+L_{ack}+1} (P_r b_{-1,k,j} + P_i b_{-2,k,j}) \tag{53}$$

Here, P_i, P_t and P_r represent the average power consumption while the node is in idle, transmitting and receiving, respectively. Power consumed by the node during CCA can be given as (54) where P_{cs} indicates the average power consumption for channel sensing.

$$P_{CCA} = P_{cs} \sum_{i=0}^{m} \sum_{j=0}^{n} (b_{i,0,j} + b_{i,-1,j}) \tag{54}$$

VII. Simulation Framework

For validation of the proposed analytical model, we have developed a simulation model using C programming which simulates the network discussed in Fig. 1(a) with nodes adhering to the state process described in Fig. 1(b). Simulations are an effective way to demonstrate the accuracy of the proposed analytical model [10]–[16]. Also, simulations prove to be an easy technique to analyze the performance of the network under different operating conditions. Hence, in this context we use the simulation model to validate the proposed analytical model. Algorithm 1 serves as the entry point of the simulation framework, where we initialize the necessary parameters such as simulation time in slots (*t*), number of nodes (*N*), minimum backoff exponent (*macMinBE*), maximum backoff exponent (*macMaxBE*) etc. Function *initialize* invoked in *main* procedure is used to initialize the states of nodes at the beginning of the simulation. The functions $leafNode$ and $packetArrival$ are the essential components for CSMA/CA flow and packet arrival process of the node.

Algorithm 2 initializes the state of every node to Q_0 which is indicated by $idleState[i] = 1$ and $csmaState[i] = 0$ for the node i. After initialization, the $leafNode$ routine shown in Algorithm 3 executes for all the nodes present in the network at every time slot. Function $leafNode$ checks if the node is present in Idle state or CSMA/CA flow and if the node is residing in the Idle state, the node checks for packet availability in the buffer associated with the node. If a packet is available in the buffer, the node proceeds to CSMA/CA flow for transmission of the packet and corresponding state transitions can be inferred from $transitToCSMA$ function described in Algorithm 4. In CSMA/CA node checks for the backoff duration indicated by $w[i]$ and will decrement the backoff counter $w[i]$ until it equals 0 as shown in Algorithm 4. Then the node performs the CCA which is described using the $performCCA$ function in Algorithm 4. If both CCA1 and CCA2 are free, then the node proceeds for transmission, else it proceeds to increment backoff exponentially. As the node starts transmission, both successful and collision transmissions are modeled using $transmission$ and $collision$ routines described in Algorithm 5. After completion of successful packet transmission, the node decides upon transiting to Idle state or CSMA/CA depending on the packet availability in the buffer. In the latter case where packet collides, if the number of retransmissions is not exceeded the node will proceed with retransmission else discards the packet and then transits to either Idle or CSMA/CA flow depending on packet availability in the buffer.

Algorithm 1 Simulation framework - **Main procedure**

```
procedure main
    t ← 10^5                                  ▷ Simulation time in slots
    N ← 10                                    ▷ Number of nodes in network
    λ ← 30                                    ▷ Packets per second
    macMaxBE ← 8                              ▷ Maximum BE
    macMinBE ← 2                              ▷ Minimum BE
    maxRetries ← 2                            ▷ Maximum retransmissions
    maxBackoff ← 5                            ▷ Maximum backoff stages
    n ← 2                                     ▷ Maximum retransmissions allowed
    L_s ← 6                                   ▷ Successful packet length in slots
    L_c ← 6                                   ▷ Collision packet length in slots
    L_ack ← 2                                 ▷ Packet length in slots
    initialize()
    for j = 0 to t do
        for i = 0 to N − 1 do
            leafNode(i)                       ▷ Invoke CSMA/CA flow
        end for
        packetArrival()                       ▷ Queuing model
    end for
end procedure
```

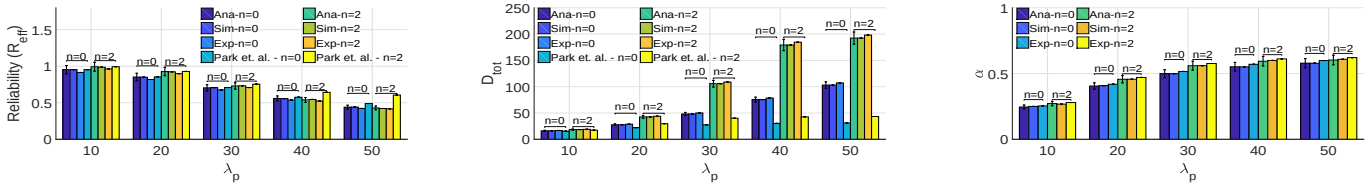
Algorithm 2 Simulation framework - **Initialization**

```
function initialize
    for i = 0 to N do
        idleState[i]←1                        ▷ In Idle state
        csmaState[i]←0                        ▷ Not in CSMA/CA state
        BE[i]←macMinBE                        ▷ Backoff exponent
    end for
end function
function backOff(i,j)
    Generates a random integer between [i,j]
end function
```

VIII. Real-time Experimental test bed

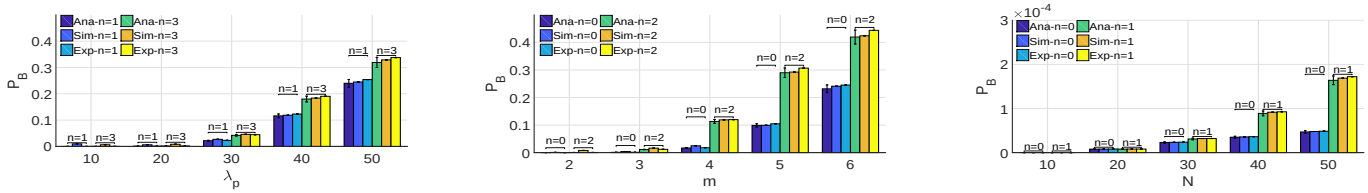
For validating the accuracy of proposed model, we also developed a real-time test bed consisting of varying number of sensor nodes. Commercially available IITH motes which operate in 2.4 GHz ISM band are used for experimentation [20]. IITH motes are clocked at 8 MHz with a physical

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2766783, IEEE Transactions on Industrial Informatics



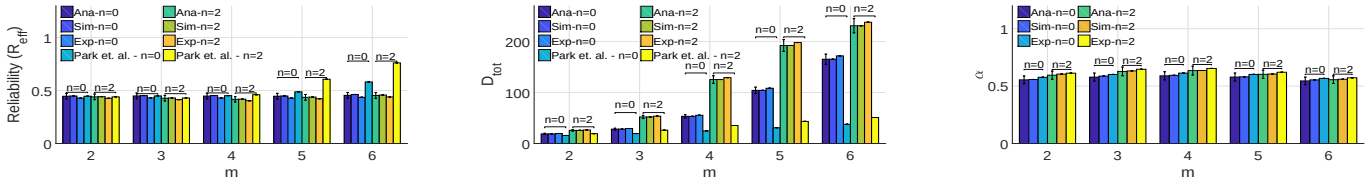
(a) Variation of R_{eff} with variation in λ_p and n. (b) Variation of D_{tot} with variation in λ_p and n. (c) Variation of α with variation in λ_p and n.

Fig. 4: Variation of R_{eff}, D_{tot} and α with λ_p and n (N=10, qLim=5, actPeriod=1, m=5, macMinBE=2, macMaxBE=8).



(a) Variation of P_B with λ_p and n (N=10, qLim=5, actPeriod=1, m=5, macMinBE=2, macMaxBE=8). (b) Variation of P_B with m and n (N=10, qLim=5, actPeriod=1, λ_p=50, macMinBE=2, macMaxBE=8). (c) Variation of P_B with N and n (λ_p=10, qLim=5, actPeriod=1, m=5, macMinBE=2, macMaxBE=8).

Fig. 5: Variation of P_B with λ_p, m, N and n.



(a) Variation of R_{eff} with m and n. (b) Variation of D_{tot} with m and n. (c) Variation of α with m and n.

Fig. 6: Variation of R_{eff}, D_{tot} and α with m and n (N=10, qLim=5, actPeriod=1, λ_p=50, macMinBE=2, macMaxBE=8).

memory of 8 kB and supports Contiki 3.0. The entire test bed is deployed in the Academic Block - A, IIT Hyderabad and the network performance is measured from randomly selected leaf nodes. The beacon interval and superframe duration is chosen to be 1536 symbol durations (BO = 5) and no Contention Free Period (CFP) or Guaranteed Time Slots (GTS) is considered for experimental setup. Each experiment is conducted for a duration of 30 min and the performance measures provided in this paper are average taken over measurements acquired from the randomly selected leaf node every 20 seconds.

IX. Performance Analysis

In this section, we discuss the performance of proposed analytical model by validating with the simulation and experimental outcomes. We considered R_{eff}, D_{tot}, α, P_B and P as the key performance metrics and analyzed with variation in λ, m, N, n and $actPeriod$. For analyzing the impact of λ on the network performance, we introduce a new parameter λ_p which indicates the packet arrival rate in packets per second as opposed to λ which indicates the arrival rate in packets per unit backoff slot. As one unit backoff slot duration is $320\mu s$, the relation between λ and λ_p is

$$\lambda_p = \frac{\lambda}{320 \times 10^{-6}} = 3125 \times \lambda \tag{55}$$

Also, we compare the accuracy of the proposed model with the model proposed by Park *et. al.* [13]. Authors in [13], considered the losses in CSMA/CA and neglected the queue overflow losses. Also, the delay formulated accounts only the CSMA/CA delay and neglects the packet waiting time in the queue. Hence, we compare the R_{eff} and D_{tot} with the reliability and delay achieved using [13]. Although R_{eff} and D_{tot} are different, the channel congestion probability remains same.

Algorithm 3 Simulation framework - **Leaf node functionality**

```
function leafNode(i)
    if idleState[i]==1 then
        if packet available in buffer then transitToCSMA(i)
        end if
    else                                              ▷ Start of CSMA/CA flow
        if m[i]≥0 AND w[i]>0 then decrementBackoff(i)
        else if m[i]≥0 AND (w[i]==0 OR w[i]==-1) then performCCA(i)
        else if m[i]==-1 then transmission(i)
        else if m[i]==-2 then collision(i)
        end if
    end if
end function
function packetArrival
    Generate packets using Poisson arrival process with λ
end function
```

A. Impact of λ_p and n on the performance of network

Fig. 4(a) plots R_{eff} versus λ_p and n. As λ_p increases, traffic and congestion in the channel increases leading to a reduction in effective reliability. Also, the packet waits for longer duration in the queue which leads to increase in P_B and D_{tot}. The same behavior can be observed from Fig.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2766783, IEEE Transactions on Industrial Informatics

Algorithm 4 Simulation framework - **Transitions**

```
function transitToCSMA(i)
    csmaState[i]←1                          ▷ Proceed to CSMA/CA
    idleState[i]←0                          ▷ Node not in Idle state
    w[i]←backOff(0,2^BE[i]-1)               ▷ Random backoff
    m[i]←0                                  ▷ Assign backoff stage
end function
function decrementBackoff(i)
    w[i]←w[i]-1                             ▷ Decrement backoff counter
end function
function performCCA(i)
    if (channel is free) then
        if w[i]==-1 then
            if (collision) then
                m[i]←-2                     ▷ Collision of packet
                w[i] = L_c                  ▷ Collision time
            else
                m[i]←-1                     ▷ Successful transmission
                w[i] = L_s                  ▷ Successful packet time
            end if
        else
            w[i]←w[i]-1
        end if
    else
        m[i]=m[i]+1;                        ▷ Perform exponential backoff procedure
        BE[i]=min(BE[i]+1,macMaxBE)
        w[i] ← backOff(0,2^BE[i]-1)
        if m[i]>maxBackoff then
            Discard packet and remove from queue
            if packet available in buffer then
                Proceed to CSMA/CA
                csmaState[i]←1
                idleState[i]←0
                w[i]←backOff(0,2^BE[i]-1)
                m[i]←0
            else
                idleState[i]=1              ▷ Proceed to Idle state
                csmaState[i]=0
            end if
        end if
    end if
end function
```

Algorithm 5 Simulation framework - **Transmission**

```
function transmission(i)
    w[i]=w[i]-1
    if w[i]==0 then
        Packet transmission completed
        if packet available in buffer then
            csmaState[i]←1                  ▷ Proceed to CSMA/CA
            idleState[i]←0
            w[i]←backOff(0,2^BE[i]-1)
            m[i]←0
        else
            idleState[i]=1                  ▷ Proceed to Idle state
            csmaState[i]=0
        end if
    end if
end function
function collision(i)
    w[i]=w[i]-1
    if w[i]==0 then
        Packet collision completed
        if n<maxRetries then
            n[i]←n[i]+1
            BE←macMinBE
            w[i]←backOff(0,2^BE[i]-1)
            m[i]←0
        else
            if packet available in buffer then
                csmaState[i]←1              ▷ Proceed to CSMA/CA
                idleState[i]←0
                BE←macMinBE
                w[i]←backOff(0,2^BE[i]-1)
                m[i]←0
            else
                idleState[i]=1              ▷ Proceed to Idle state
                csmaState[i]=0
            end if
        end if
    end if
end function
```

TABLE III: Average power consumption of node in different states considered for analysis

$State$	$Avg.\ Power\ Consumption\ (\mu W)$
P_s	0.26
P_i	160
P_t	160
P_{rx}	170
P_{cs}	170

4(b), 4(c) and 5(a), which plots D_{tot}, α and P_B versus λ_p respectively. After increasing n, the node can retransmit more times increasing R_{eff} as shown in Fig. 4(a). Although R_{eff} increases with increase of *n*, it will also increase D_{tot}, α and P_B as the node spends more time in backoff stages. Compared to Park *et. al.*, at lower values of λ_p, the queue overflow losses are negligible and hence, reliability in both the cases are similar. Upon increase in λ_p queue overflow losses increase due to which the R_{eff} achieved is lesser than the reliability achieved using Park *et. al.* Also, for lesser traffic the queue waiting time is negligible due to which D_{tot} is similar to that of Park *et. al.* As the traffic increases, the queue waiting time dominates the delay incurred due to CSMA/CA leading to higher D_{tot} compared to Park *et. al.*

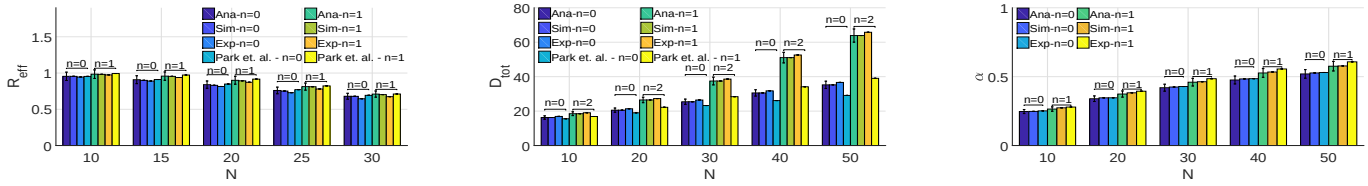
B. Impact of m and n on the performance of network

Fig. 6(a) plots R_{eff} versus m and n. As m increases, R_{eff} initially increases (from m=[2,5]) but later decreases as the node spends mostly in backoff stages leading to increased P_B as shown in Fig. 5(b). Similarly as m increases, D_{tot} will also increase due to increase in backoff waiting time for the node which can be observed from Fig. 6(b). From Fig. 6(c), one can observe the increase in α during m=[2,4] and decreasing behavior in the latter duration of m. During the initial stages of m, the node waits in backoff stages for a smaller duration and packet failures will be dominated by exceeded number of backoff stages. Although packets are discarded, the service rate achieved at the queue will be increased which leads to increase in network traffic thereby increasing α up to a certain value of m (m=4 in this case). After m increases beyond 4, node starts to spend more time in backoff stages and results in a decrease of service rate which leads to increase in P_B significantly. The same can be observed from Fig. 5(b) which plots P_B. As node spends more time in backoff stages, congestion in channel reduces leading to decrease of α (for m>4 in this case). With increase in n, R_{eff} will increase as more retransmissions are allowed and chances for successful packet transmission increases, but also leads to increase in D_{tot}. The same behavior can be observed from Fig. 6(a) and 6(b). With the increase of n, as more transmissions are allowed, it also leads to increase in channel congestion and reduces the service rate leading to an increase in α and P_B which can be observed from Fig. 6(c) and 5(b), respectively. Also, one can observe the significant deviation of R_{eff} and D_{tot} compared to Park. *et. al.* resulted by considering queue overflow losses and packet waiting time in the queue.

C. Impact of N and n on the performance of network

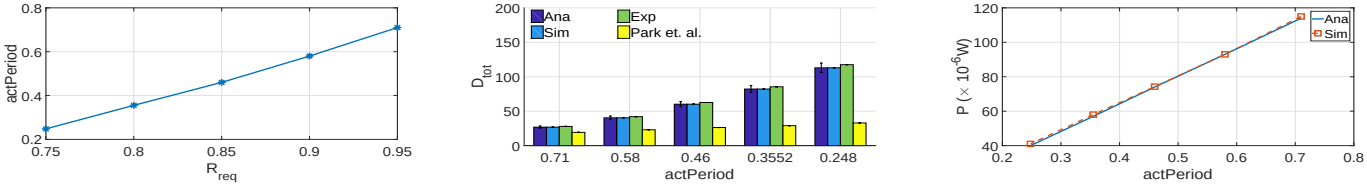
Fig. 7(a), 7(b) and 7(c) plots R_{eff}, D_{tot} and α versus N. As N increases, channel congestion increases leading to loss

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2766783, IEEE Transactions on Industrial Informatics



(a) Variation of R_{eff} with N and n. (b) Variation of D_{tot} with N and n. (c) Variation of α with N and n.

Fig. 7: Variation of R_{eff}, D_{tot} and α with N and n (λ_p=10, qLim=5, actPeriod=1, m=5, macMinBE=2, macMaxBE=8).



(a) Amount of $actPeriod$ required for achieving user specified R_{req}. (b) Variation of D_{tot} with $actPeriod$. (c) Variation of E with $actPeriod$.

Fig. 8: Analysis of duty cycle management framework proposed (N=10, λ_p=5, n=1, m=4, macMinBE=2, macMaxBE=8).

of reliability (R_{eff}) and also increases the packet waiting time which leads to increase in (D_{tot}). As the delay in accessing channel increases, the service rate also decreases leading to queue blocking losses. Hence we can see the increase of P_B in Fig. 5(c) which plots the variation of P_B with N. The impact of n is similar to that of described in earlier scenarios. When compared with Park. *et. al.*, the proposed model accurately models the performance by considering queue blocking losses and queue waiting times.

D. Power conservation using the proposed duty cycle management framework

We will analyze the performance of duty cycle management framework by calculating the required $actPeriod$ for a user specified reliability (R_{req}). For analysis, we consider the power consumption of commercially available IITH mote shown in Table III [20]. Fig. 8(a) plots the amount of $actPeriod$ required for different R_{req}. We can observe that as R_{req} increases, the $actPeriod$ increases because the improvement in successful packet transmission can only be obtained by spending more time in active state. As $actPeriod$ increases, channel availability for a node to transmit also increases leading to less congestion for a constant λ_p. Hence we can observe, the decrease in D_{Succ} from Fig. 8(b). Fig. 8(c) plots P for different $actPeriod$ and lesser the $actPeriod$, lesser is the power consumption. To demonstrate the efficiency of the proposed scheme, we compare with ADCA duty cycling scheme proposed in [16], where authors have proved that ADCA provides the best duty cycle optimization when compared to popular techniques existing in the literature under different traffic scenarios. Table IV compares the proposed model with ADCA. With λ_p varying from [2, 10], one can observe that in all the cases except λ_p = 2 and 4, the proposed model achieves a lesser $actPeriod$ than ADCA. Also, the D_{tot} for the proposed model is significantly lesser than ADCA making the proposed model a better solution for time-critical

TABLE IV: Comparison of proposed duty cycle management scheme with ADCA (N=10, macMinBE=2, macMaxBE=8, m=5, n=1)

	ADCA			Proposed model			
λ_p	R_{eff}	D_{tot}	$actPeriod$	D_{tot}	$actPeriod$	$P(\mu W)$	*Power Saving*
2	0.915	286.99	0.116	96.822	0.155	25	84.3%
4	0.861	568.55	0.194	141.09	0.228	37	76.8%
6	0.79	753.796	0.256	197.15	0.256	42	73.7%
8	0.732	830.8	0.299	235.6	0.288	47	70.6%
10	0.648	932.19	0.336	313.84	0.305	50	68.7%

industrial applications. We have also provided the significant amount of power savings achieved using the proposed model, and our analysis shows that the proposed model produces an average power saving of 74.82% while meeting user specified reliability requirements. Also, using the proposed model, the user has direct control over the achievable reliability, unlike ADCA where the achieved reliability cannot be controlled directly. As the industrial applications are classified into different classes with specified reliability requirements [19], the proposed duty cycle management scheme aids in achieving significant energy savings when used in real-time industrial applications. In terms of computational complexity, as the proposed mechanism decides the amount of duty cycle required even before deploying the network, it does not induce any computational burden on the physical nodes. However, the convergence time strictly depends on the step size chosen for Gradient-Descent optimization, the proposed mechanism when run on an Intel PC with 4 GB RAM takes 350 seconds on an average to converge with $\delta = 0.001$.

E. Blocking probability analysis for motes with on-node storage of 4 kB and 8 kB

For more insight into the effect of P_B, we analyzed the performance of commercially available MicaZ mote with 4 kB and IITH mote with 8 kB on-node storages under high traffic. In ideal conditions, with maximum packet size of 127

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TII.2017.2766783, IEEE Transactions on Industrial Informatics

bytes, MicaZ mote and IITH mote can queue a maximum of 32 and 64 packets. Fig. 9 plots the analytical outcomes of P_B versus λ_p and N, while to avoid congestion we did not include simulation outcomes. In Fig. 9, for the case of 4 kB and λ_p=60, one can observe packet loss due to queue overflow and no losses for 8 kB. With an increase in N from 10 to 20 the blocking probability increased but follows the similar trend when compared with the N=10 scenario. As N increases, the service rate decreases due to larger delays in channel access resulting in higher P_B at low traffic rates. In both the cases of N=10 and 20, we see P_B for on-node storages 4 kB and 8 kB overlapping as λ_p increases which can be better explained using Fig. 10 obtained from simulations. Fig. 10(a) and 10(b) plots the histograms of queue occupancy for a node with λ_p=60, N=10 and on-node storages of 4 kB and 8 kB respectively. At λ_p=60, the service rate is comparatively higher than the packet arrival rate leading to less queue overflow losses and therefore the distribution is biased towards lower queue occupancy. This is what creates the difference in P_B at low traffic regimes for 4 kB and 8 kB on-node storages. Now, as λ_p increases from 60 to 80, the corresponding histograms are shown in Fig. 10(c) and 10(d). One can clearly observe the negative skewness in distribution indicating that the queue is filled up for maximum duration of the network operation. Here as λ_p increases, the service rate is lesser than packet arrival rate leading to the houseful of the queue during initial stages of the operation. Following packets will be lost for most of the time due to queue overflow resulting in the convergence of P_B to the same value for both 4 kB and 8 kB on-node storages.

Each simulation outcome is an average taken over 100 realizations with each realization having a simulation length of 1280 seconds. From the performance analysis it can be observed that the proposed analytical model analyzes the performance of IEEE 802.15.4 MAC with constrained on-node storage with a very good accuracy. When compared with simulation outcomes, the model achieves accuracy of above 97% and compared to experimental results acquired from the real-time test bed achieves an accuracy of above 94%. The increase of error in experimental results is due to: (1) additional traffic induced by periodic super frame transmission, (2) additional delay incurred as the leaf node computes performance metrics, and (3) slight loss of synchronization of backoff slot boundaries across the nodes. We have also indicated the 94% confidence intervals for the analytical results wherever necessary.

X. Conclusion

In this paper, we developed an accurate analytical model and analyzed the IEEE 802.15.4 MAC with size constrained packet queue which is an important aspect for Industrial Internet of Things. We have modeled the distributions for three kinds of service times present in the IEEE 802.15.4 MAC. Also, we proposed a novel power management framework to achieve the user specified reliability requirement with minimal power consumption. For validating accuracy, we have performed extensive validation of the analytical model with simulation outcomes and real-time test bed deployed in the Academic Block

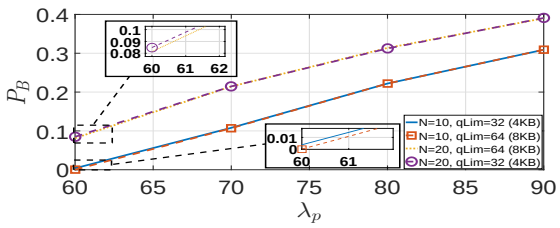


Fig. 9: Effect of λ_p and on-node storage on P_B for nodes with 4 kB and 8 kB on-node storage (macMinBE=2, macMaxBE=5, m=4, n=1, actPeriod=1).

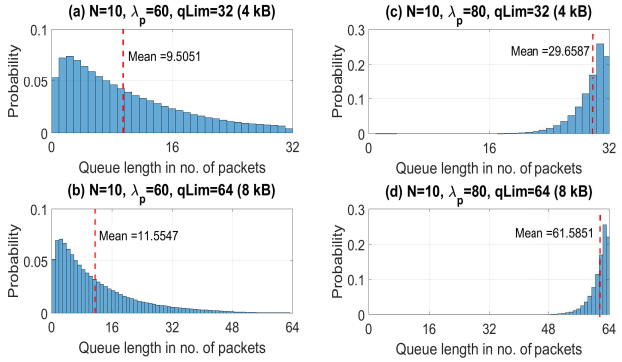


Fig. 10: Effect of λ_p and N on queue occupancy with occupancy histogram for nodes with 4 kB and 8 kB on-node storages (macMinBE=2, macMaxBE=5, m=4, n=1, actPeriod=1).

- A, IIT Hyderabad. It is observed that the proposed analytical model accurately analyzes the performance of IEEE 802.15.4 MAC with constrained on-node storage by achieving less than 3% and 6% error when compared to simulation outcomes and experimental outcomes, respectively. The effectiveness of the mathematical formulation and the power management scheme is also proved by comparing with the popular models existing in the literature. We are convinced that this theoretical study as well as illustrated industrial experiments, can improve the efficiency of IEEE 802.15.4 based industrial applications. Our future scope is to optimize the IEEE 802.15.4 MAC for time-critical industrial applications.

References

[1] L. D. Xu, W. He and S. Li, "Internet of Things in Industries: A Survey," in *IEEE Transactions on Industrial Informatics*, vol. 10, no. 4, pp. 2233-2243, Nov. 2014. doi: 10.1109/TII.2014.2300753

[2] F. Liu, C. Tan, E. Lim, and B. Choi, "Traversing knowledge networks: An algorithmic historiography of extant literature on the Internet of Things (IoT)," *J. Manage. Anal.*, pp. 132, Aug. 2016, doi: 10.1080/23270012.2016.1214540.

[3] C. Zhai, "Delay-aware and reliability-aware contention-free MFTDMA protocol for automated RFID monitoring in industrial IoT", *Journal of Industrial Information Integration*, vol. 3, pp. 819, 2016.

[4] M. Wollschlaeger, T. Sauter and J. Jasperneite, "The Future of Industrial Communication: Automation Networks in the Era of the Internet of Things and Industry 4.0," in *IEEE Industrial Electronics Magazine*, vol. 11, no. 1, pp. 17-27, March 2017. doi: 10.1109/MIE.2017.2649104

[5] K. L. Chen, Y. R. Chen, Y. P. Tsai and N. Chen, "A Novel Wireless Multifunctional Electronic Current Transformer Based on ZigBee-Based Communication," in *IEEE Transactions on Smart Grid,* vol. 8, no. 4, pp. 1888-1897, July 2017. doi: 10.1109/TSG.2015.2510325

[6] L. C. Lemos, J. J. Silva and J. S. R. Neto, "Vibration analysis for fouling detection using hammer impact test and ZigBee based wireless sensor network," in *2016 IEEE 25th International Symposium on Industrial Electronics (ISIE)*, Santa Clara, CA, 2016, pp. 1190-1195. doi: 10.1109/ISIE.2016.7745064

[7] M. S. Kang, Y. L. Ke and J. S. Li, "Implementation of smart loading monitoring and control system with ZigBee wireless network," *2011 6th IEEE Conference on Industrial Electronics and Applications,* Beijing, 2011, pp. 907-912. doi: 10.1109/ICIEA.2011.5975716

[8] IEEE STD 802.15.4-2996, September, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE, 2006. [Online]. Available: http://www.ieee802.org/15

[9] C. Lu et al., "Real-Time Wireless Sensor-Actuator Networks for Industrial Cyber-Physical Systems," in *Proceedings of the IEEE,* vol. 104, no. 5, pp. 1013-1024, May 2016. doi: 10.1109/JPROC.2015.2497161

[10] C. Buratti and R. Verdone, "Performance Analysis of IEEE 802.15.4 Non Beacon-Enabled Mode," in *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3480-3493, Sept. 2009. doi: 10.1109/TVT.2009.2014956

[11] J. He, Z. Tang, H. H. Chen and Q. Zhang, "An accurate and scalable analytical model for IEEE 802.15.4 slotted CSMA/CA networks," in *IEEE Transactions on Wireless Communications*, vol. 8, no. 1, pp. 440-448, Jan. 2009. doi: 10.1109/T-WC.2009.080277

[12] C. Y. Jung, H. Y. Hwang, D. K. Sung and G. U. Hwang, "Enhanced Markov Chain Model and Throughput Analysis of the Slotted CSMA/CA for IEEE 802.15.4 Under Unsaturated Traffic Conditions," in IEEE Transactions on Vehicular Technology, vol. 58, no. 1, pp. 473-478, Jan. 2009. doi: 10.1109/TVT.2008.923669

[13] P. Park, P. Di Marco, C. Fischione and K. H. Johansson, "Modeling and Optimization of the IEEE 802.15.4 Protocol for Reliable and Timely Communications," in *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 3, pp. 550-564, March 2013. doi: 10.1109/TPDS.2012.159

[14] J. Misic, S. Shafi and V. B. Misic, "Maintaining reliability through activity management in an 802.15.4 sensor cluster," in *IEEE Transactions on Vehicular Technology,* vol. 55, no. 3, pp. 779-788, May 2006. doi: 10.1109/TVT.2006.873824

[15] M. Anwar, Y. Xia and Y. Zhan,"TDMA-Based IEEE 802.15.4 for Low-Latency Deterministic Control Applications," in *IEEE Transactions on Industrial Informatics,* vol. 12, no. 1, pp. 338-347, Feb. 2016. doi: 10.1109/TII.2015.2508719

[16] H. Rasouli, Y. S. Kavian and H. F. Rashvand, "ADCA: Adaptive Duty Cycle Algorithm for Energy Efficient IEEE 802.15.4 Beacon-Enabled Wireless Sensor Networks," in *IEEE Sensors Journal*, vol. 14, no. 11, pp. 3893-3902, Nov. 2014. doi: 10.1109/JSEN.2014.2353574

[17] S. Karlin and H. Taylor, *A First Course in Stochastic Processes*, Academic, 2 ed., 1975.

[18] H. Takagi, "M/G/1/K queues with N-policy and setup times," in *Queueing Systems,* vol. 14, no. 1-2, pp. 79-98, Mar. 1993.

[19] M. Li, C. Hua, C. Chen and X. Guan, "Application-driven virtual network embedding for industrial wireless sensor networks," *2017 IEEE International Conference on Communications (ICC)*, Paris, 2017, pp. 1-6. doi: 10.1109/ICC.2017.7996431

[20] M. P. R. Sai Kiran, P. Rajalakshmi, Y. S. Krishna and A. Acharyya, "System Architecture for Low-Power Ubiquitously Connected Remote Health Monitoring Applications With Smart Transmission Mechanism," in *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4532-4543, Aug. 2015. doi: 10.1109/JSEN.2015.2413836

XI. Appendix I

Here we summarize the necessary steps to derive CSMA/CA state probabilities used in this paper. Readers are requested to refer [13] for more detailed description. We will first derive the below important transition probabilities.

$$P(i,k,j/i,k+1,j) = 1, \ for \ k \geq 0 \tag{56}$$

$$P(i,k,j/i-1,0,j) = \tfrac{\alpha+(1-\alpha)\beta}{W_i}, \ for \ \ i \leq m \tag{57}$$

$$P(0,k,j/i,0,j-1) = \tfrac{(1-\alpha)(1-\beta)P_c}{W_0}, \ for \ \ i \leq m \tag{58}$$

From the normalization equation shown in (4), considering the first summation term, can be expressed as

$$\sum_{i=0}^{m}\sum_{k=0}^{W_i-1}\sum_{j=0}^{n} b_{i,k,j} = \sum_{i=0}^{m}\sum_{j=0}^{n} \tfrac{W_i+1}{2}(\alpha+(1-\alpha)\beta)b_{0,0,j} \tag{59}$$

Now by assuming $x = \alpha + (1-\alpha)\beta$ and $y = P_c(1-x^{m+1})$, one can solve the equation (59) to as shown in equation (60).

$$\sum_{i=0}^{m}\sum_{k=0}^{W_i-1}\sum_{j=0}^{n} b_{i,k,j} = \tfrac{b_{0,0,0}}{2}\Big(\tfrac{1-(2x)^{m+1}}{1-2x}W_0 + \tfrac{1-x^{m+1}}{1-x}\Big)\tfrac{1-y^{n+1}}{1-y} \tag{60}$$

Now considering the second summation term, can be expressed as shown below

$$\begin{aligned}\sum_{i=0}^{m}\sum_{j=0}^{n} b_{i,-1,j} &= \sum_{i=0}^{m}\sum_{j=0}^{n}(1-\alpha)(\alpha+(1-\alpha)\beta))^i b_{0,0,j} \\ &= (1-\alpha)\tfrac{1-x^{m+1}}{1-x}\tfrac{1-y^{m+1}}{1-y}b_{0,0,0}\end{aligned} \tag{61}$$

Finally the third summation can be expressed as shown in equation (62)

$$\begin{aligned}\sum_{j=0}^{n}\Big(\sum_{k=0}^{L_s-1} b_{-1,k,j} + \sum_{k=0}^{L_c-1} b_{-2,k,j}\Big) = \\ (L_s(1-P_c)+L_cP_c)(1-x^{m+1})\tfrac{1-y^{n+1}}{1-y}b_{0,0,0}\end{aligned} \tag{62}$$

Using equations (3), (60), (61) and (62) in equation (4), one can derive $b_{0,0,0}$ as shown in equation (5). Now the probability that a node attempts CCA1 (τ) is given by equation (63)

$$\begin{aligned}\tau &= \sum_{i=0}^{m}\sum_{j=0}^{n} b_{i,0,j} = \sum_{i=0}^{m}\sum_{j=0}^{n}(\alpha+(1-\alpha)\beta)^i b_{0,0,j} \\ &= \sum_{i=0}^{m}\sum_{j=0}^{n} x^i(P_c(1-x^{m+1}))^j b_{0,0,0} = \sum_{i=0}^{m}\sum_{j=0}^{n} x^i y^j b_{0,0,0} \\ &= \tfrac{1-x^{m+1}}{1-x}\tfrac{1-y^{n+1}}{1-y}b_{0,0,0}\end{aligned} \tag{63}$$

Using equation (63), we will derive the probability of collision (P_c) which is at least one of the remaining N-1 nodes start transmitting at the same time.

$$P_c = 1-(1-\tau)^{N-1} \tag{64}$$

We will now derive the channel busy probability as below, where α_1 indicates the probability of finding channel busy during CCA1 due to data transmission and α_2 due to ACK transmission. L_p and L_{ack} denote the packet length and acknowledgment length respectively.'

$$\alpha = \alpha_1 + \alpha_2 \tag{65}$$

$$\alpha_1 = L_p(1-(1-\tau)^{N-1})(1-\alpha)(1-\beta) \tag{66}$$

$$\alpha_2 = L_{ack}\tfrac{N\tau(1-\tau)^{N-1}}{1-(1-\tau)^N}(1-(1-\tau)^{N-1})(1-\alpha)(1-\beta) \tag{67}$$

Similarly the probability of finding channel busy in CCA2 is given by equation (68).

$$\beta = \tfrac{1-(1-\tau)^{N-1}+N\tau(1-\tau)^{N-1}}{2-(1-\tau)^N+N\tau(1-\tau)^{N-1}} \tag{68}$$