Chapter 1

Energy Consumption of Key Distribution in 802.15.4 Beacon Enabled Cluster with Sleep Management

Jelena Mišić

University of Manitoba, Winnipeg, Manitoba, Canada

In this Chapter, we analyze performance of the 802.15.4 cluster in beacon enabled mode under the presence of key exchange protocol. We assume that all nodes are applying power management technique based on the constant event sensing reliability required by the coordinator. Power management generates random sleep times by every node which in average fairly distributes the sensing load among the nodes. Key exchange is initiated by cluster coordinator after some given number of sensing packets have been received by the coordinator. We develop analytical model of key exchange integrated into the cluster's sensing function and evaluate the impact of frequency of key exchange on the cluster's energy consumption.

1.1. Introduction

In order to penetrate the market with cost-effective solutions for WSNs we need standardized low-cost, low-power and short-range communication Low Rate Wireless Personal Area Network (LR-WPAN) technology. Important candidate for the application in this area is IEEE 802.15.4 standard.[1]

The 802.15.4 specification outlines some basic security services at the data link layer that can be combined with advanced techniques at the upper layers to implement a comprehensive security solution. For example, the recent ZigBee specification[2] implements a number of protocols—including security-related ones—that can be deployed in an 802.15.4 network. Given that the 802.15.4 devices are typically severely constrained in terms of their communication and computational resources, the implementation of such solutions is likely to impose a significant performance overhead. For the reason of cost effectiveness we assume that Symmetric-Key Key Establishment (SKKE)[2] is implemented over the IEEE 802.15.4 sensor cluster operating in beacon-enabled, slotted CSMA-CA mode.

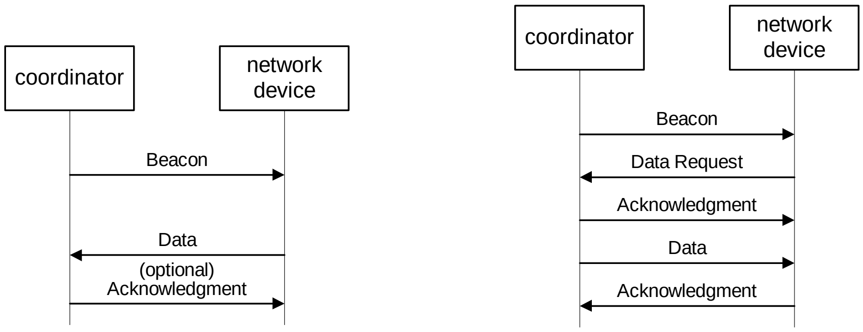
Key update provides an automated mechanism for restricting the amount of data which may be exposed when a link key is compromised. Key update frequency depends on the key update overheads and threat environment under which network is working. Hence controlling the life time of keys and determination of how the key update occurs is a challenging task in any network. In[3] we have reported simulated network behavior without sleep management when the threshold for key update was set to 10 packets. In this Chapter we develop analytical model for the cluster behavior including periodic key exchange (with variable update threshold), power management and sensing data application. Nodes in cluster apply sleep technique in order to deliver only the required number of packets per second (which we will call event sensing reliability) to the coordinator. We use numerical results to evaluate the overhead of key exchange in terms of medium behavior, total number of delivered packets, nodes' utilization and effect on node's life time.

The Chapter is organized as follows. Section 1.2 gives a brief overview of the operation of 802.15.4-compliant networks with star topology in the beacon-enabled, slotted CSMA-CA mode, followed by a review of power management techniques for 802.15.4 and basic security mechanisms provided for by the standard. As the 802.15.4 specification does not prescribe any particular key management approach, we will make use of the SKKE mechanism presented in Section 1.3. Section 1.4 presents derivation of analytical model of the cluster, while Section 1.5 contains derivation of medium behavior and packet service time. Section 1.6 presents numerical results obtained from the analysis. Finally, Section 1.7 concludes the Chapter.

1.2. An overview of 802.15.4 beacon enabled MAC

The 802.15.4 networks with star topology operate in beacon enabled mode where channel time is divided into superframes bounded by beacon transmissions from the PAN coordinator.[1] All communications in the cluster take place during the active portion of the superframe; the (optional) inactive portion may be used to switch to conserve power by switching devices to a low power mode. Standard supports 16 different frequency channels in which clusters can operate within ISM band. Due to interference, physically adjacent clusters must operate in separate channels. Uplink channel access is regulated through the slotted CSMA-CA mechanism.[1]

Data transfers in the downlink direction, from the coordinator to a node, must first be announced by the coordinator. In this case, the beacon frame will contain the list of nodes that have pending downlink packets, as shown in Fig. 1.1(b). When the node learns there is a data packet to be received, it transmits a request. The coordinator acknowledges the successful reception of the request by transmitting an acknowledgement. After receiving the acknowledgement, the node listens for the actual data packet for the period of *aMaxFrameResponseTime*, during which the coordinator must send the data frame.



(a) Uplink transmission.

(b) Downlink transmission.

Fig. 1.1. Data transfers in 802.15.4 PAN in beacon enabled mode.

Power management consists of adjusting the frequency and ratio of active and inactive periods of sensor nodes.[4,5] For 802.15.4 nodes it can be implemented in two ways. In the first one, supported by the standard,[1] the interval between the two beacons is divided into active and inactive parts, and the sensors can switch to low-power mode during the inactive period. Activity management for individual nodes can be accomplished through scheduling of their active and inactive periods. In order to avoid simultaneous activity and collisions by awakened nodes, sleep periods have to be randomized. In order to ensure fairness among the nodes, coordinator has to periodically broadcast required event sensing reliability (number of packets per second needed for reliable event detection) and number of nodes which are alive. Based on that information node can calculate average period of sleep between transmissions. When average sleep period is known, then some discrete random probability distribution can be used to generate individual sleep durations.[6]

The 802.15.4 standard specifies several security suites which consist of a 'set of operations to perform on MAC frames that provide security services'.[1] Specified security services include access control lists, data encryption using pre-stored key, message integrity code generated using the pre-stored key, and message freshness protection. While these services are useful, they are by no means sufficient. In particular, procedures for key management, device authentication, and freshness protection are not specified by the 802.15.4 standard. Hence, they must be implemented on top of 802.15.4 MAC layer.

1.3. Symmetric-Key Key Establishment Protocol

Low cost alternative for this task with possibility to change the symmetric keys between the nodes and the coordinator is the ZigBee protocol suite[2] developed by the ZigBee Alliance, an industry consortium working on developing network and Application Programming Interfaces (API) for wireless ad hoc and sensor networks. The ZigBee APIs include security extensions at different networking layers, using both symmetric and asymmetric key exchange protocols. Asymmetric key exchange protocols, which mainly rely on public key cryptography, are computationally intensive and their application in wireless sensor networks is only possible with devices that are resource rich in computation and power and connected through high bandwidth links.

The application support sub-layer of the ZigBee specification defines the mechanism by which a ZigBee device may derive a shared secret key (Link Key) with another ZigBee device; this mechanism is known as the Symmetric-Key Key Establishment (SKKE) protocol. Key establishment involves coordinator and node, and should be prefaced by a trust provisioning step in which trust information (a Master key) provides a starting point for establishing a link key. The Master key may be pre-installed during manufacturing, may be installed by a trust center, or may be based on user-entered data (PIN, password).

This protocol relies on Keyed-hash message authentication code, or HMAC, which is a message authentication code (MAC) calculated using a cryptographic hash function in conjunction with a secret key. For the cryptographic hash function the 802.15.4 specification supports the AES block cipher in its basic form, while the ZigBee specification suggests the use of a modified AES algorithm with a block size of 128 bits.[7] The hash function of a data block d will be denoted as $H(d)$. The ZigBee specification

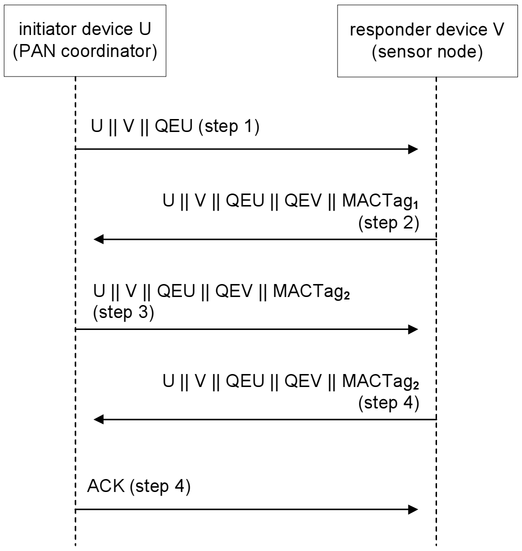


Fig. 1.2. SKKE protocol between the PAN coordinator and the node.

suggests the use of the keyed-hash message authentication code (HMAC):

$$\begin{aligned} &MacTag = HMAC(MacData) \\ &\;= H((MacKey \oplus opad)||H(MacKey \oplus ipad)||MacData) \end{aligned}$$

where *ipad* and *opad* are hexadecimal constants. In this Chapter, we will follow the notation introduced in[2] and present the last equation in the equivalent form $MacTag = MAC_{MacKey} MacData$.

The SKKE protocol is initiated by the PAN coordinator (denoted as initiator device U) by exchanging ephemeral data. The PAN coordinator U will generate the challenge QEU. Upon receiving the challenge QEU, the node (denoted as V) will validate it and also generate its own, different challenge QEV and send it to the PAN coordinator U.

Upon successful validation of challenges, both devices generate a shared secret based on the following steps:

1. Each device generates a $MACData$ value by concatenating their respective identifiers and validated challenges together: $MACData = U||V||QEU||QEV$.
2. Each device calculates the $MACTag$ (i.e., the keyed hash) for $MACData$ using the Master Key $Mkey$ as $MACTag =$

$MAC_{Mkey}MACData$. Note that both devices should obtain the same shared secret $Z = MACTag$ at this time.
3. In order to derive the link key each device generates two cryptographic hashes of the shared secret and hexadecimal numbers, i.e. $Hash_1 = H(Z||01_{16})$ $Hash_2 = H(Z||02_{16})$. The $Hash_2$, will be Link Key among two devices, while $Hash_1$, will be used to confirm that both parties have reached the same Link Key.

1.4. Analytical model of the cluster with SKKE

In this section we will develop Markov chain model for node behavior which includes all phases of SKKE protocol and subsequent sleep and transmission phases. We assume that PAN coordinator maintains a separate counter for the number of transmissions by each node. When the counter value reaches threshold n_k, key update protocol is triggered. Updated keys are used to generate Message Authentication Code. The high level Markov chain which includes key update, sleep periods followed by the transmissions is presented in Fig. 1.3.

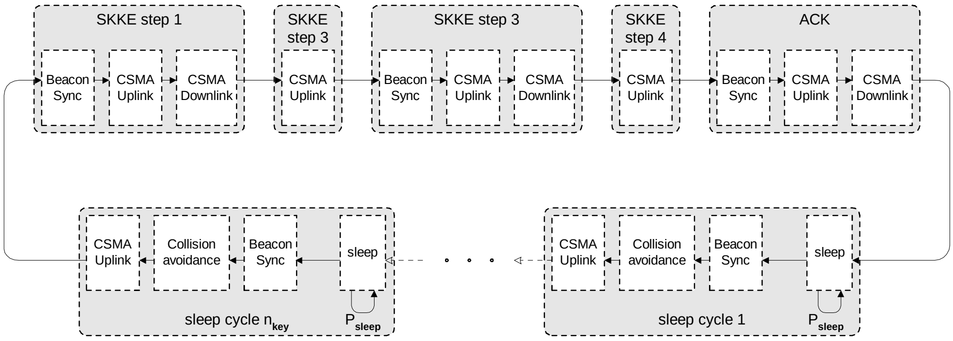


Fig. 1.3. Markov chain for the node behavior under threshold triggered key exchange.

Furthermore, each of the steps which involves downlink transmission requires synchronization with the beacon, transmission of the uplink request packet and transmission of the downlink packet as shown in Fig. 1.1(b). Every transmission is implemented using slotted CSMA-CA specified by the standard.[1] Markov sub-chain for single CSMA-CA transmission (as the component of the Fig. 1.3) is shown in Fig. 1.4. The delay line from Fig. 1.4 models the requirement from the standard that every transmission which can not be fully completed within the current superframe has to be delayed to the beginning of the next superframe and is shown in Fig. 1.5(a). The probability that packet will be delayed is denoted as $P_d = \overline{D_d}/SD$ where SD denotes duration of active superframe part (in backoff periods) and $\overline{D_d} = 2 + \overline{G_p} + 1 + \overline{G_a}$ denotes total packet transmission time including two clear channel assessments, transmission time $\overline{G_p}$, waiting time for the acknowledgement and acknowledgement transmission time $\overline{G_a}$. The block labeled T_r denotes $\overline{D_d}$ linearly connected backoff periods needed for actual transmission. Within the transmission sub-chain, the process $\{i, c, k, d\}$ defines the state of the device at backoff unit boundaries where:

$i \in (0..m)$ is the index of current backoff attempt, where m is a constant defined by MAC with default value 4.

$c \in (0, 1, 2)$ is the index of the current Clear Channel Assessment (CCA) phase.

$k \in (0..W_i - 1)$ is the value of backoff counter, with W_i being the size of backoff window in i-th backoff attempt. The minimum window size is $W_0 = 2^{macMinBE}$, while other values are equal to $W_i = W_0 2^{min(i,\, 5-macMinBE)}$ (by default, $macMinBE = 3$).

$d \in (0..\overline{D_d} - 1)$ denotes the index of the state within the delay line mentioned above; in order to reduce notational complexity, it will be shown only within the delay line and omitted in other cases.

Moreover, we need to include synchronization time from the moment when node wakes-up till the next beacon shown in Fig. 1.5(b) as well as the uniformly distributed time needed to separate potential collisions among the nodes which wake up in the same superframe shown in Fig. 1.5(c). One may argue that this last separation time is not needed since CSMA-CA random backoff times will do the separation but by the standard, both request packets and data packets by awakened nodes will start backoff count immediately after the beacon with backoff window which has the range from 0-7 backoff periods. Due to the small backoff window, collisions will be likely and we think that additional separation is needed. Synchronization with the beacon is also needed to receive the acknowledgement from the coordinator that whole SKKE transaction is completed. We assume that this acknowledgement is sent in downlink packet.

Data and key information packet sizes are assumed to be 12 backoff periods long and therefore we assume that probability to access the medium τ_0 as well as probabilities of transmission without the collision γ, and that

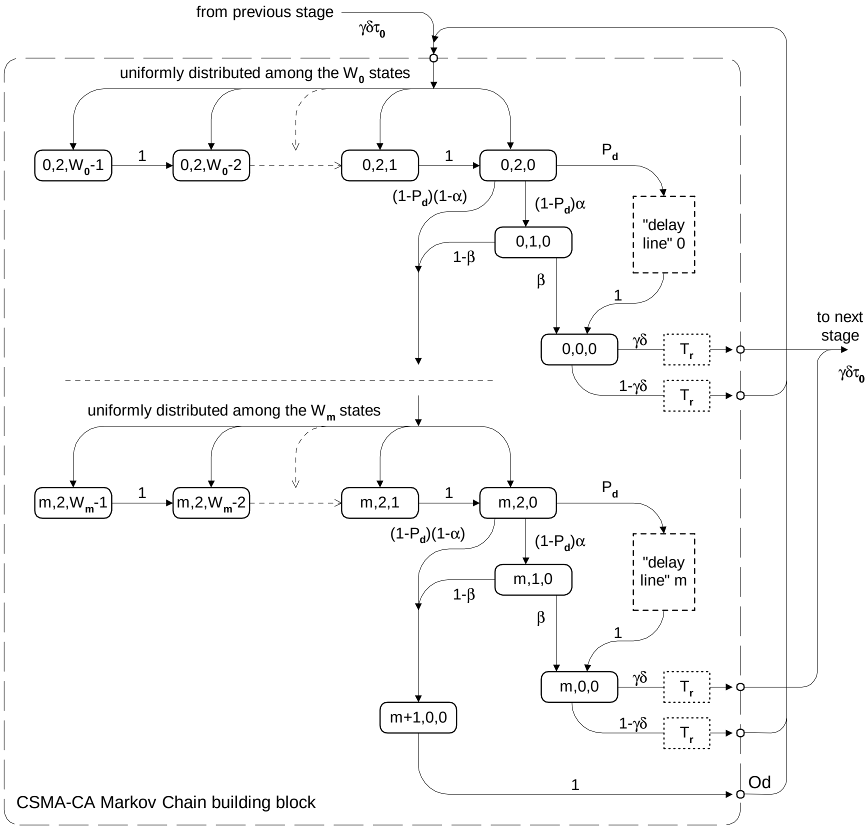


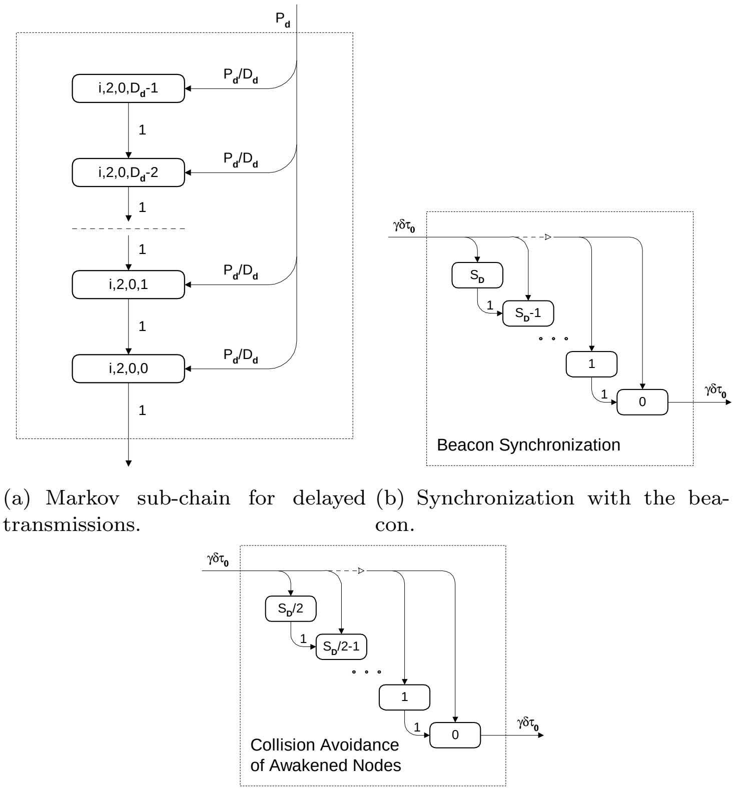
Fig. 1.4. Markov sub-chain for single CSMA-CA transmission.

packet will not be corrupted δ have the stationary value after every transmission attempt. The same assumption holds for probabilities that the medium is idle on first α and second CCA with β respectively.

Let us assume that the input probability to arbitrary transmission block is $\tau_0\gamma\delta$ where $\tau_0 = \sum_{i=0}^{m} x_{0,0,0}$ is medium access probability after each packet transmission. We also assume that Medium Access Control layer is reliable and that it will repeat transmission until the packet is acknowledged.

Therefore the probability of finishing the first backoff phase in transmission block is equal to $x_{0,2,0} = \tau_0\gamma\delta + \tau_0(1-\gamma\delta) = \tau_0$.

Using the transition probabilities indicated in Figs. 1.4 and 1.5(a), we



(a) Markov sub-chain for delayed transmissions.

(b) Synchronization with the beacon.

(c) Time needed to avoid collisions among awakened nodes.

Fig. 1.5. Delay and synchronization lines

can derive the relationships between the state probabilities and solve the Markov chain. For brevity, we will omit l whenever it is zero, and introduce the auxiliary variables C_1, C_2, C_3 and C_4:

$$\begin{aligned}
x_{0,1,0} &= \tau_0(1-P_d)\alpha = \tau_0 C_1 \\
x_{1,2,0} &= \tau_0(1-P_d)(1-\alpha\beta) = \tau_0 C_2 \\
x_{0,0,0} &= \tau_0((1-P_d)\alpha\beta + P_d) = \tau_0 C_3 \\
C_4 &= \frac{1-C_2^{m+1}}{1-C_2}
\end{aligned} \tag{1.1}$$

Using values C_i we obtain the set of equations for transmission sub-chain:

$$\begin{aligned}
x_{i,0,0} &= \tau_0 C_3 C_2^i, \;\text{ for }\; i = 0\,..\,m \\
x_{i,2,k} &= \tau_0 \frac{W_i - k}{W_i} \cdot C_2^i, \;\text{ for }\; i = 1\,..\,m; k = 0\,..\,W_i - 1 \\
x_{i,1,0} &= \tau_0 C_1 C_2^i \text{for }\; i = 0\,..\,m \\
x_{0,2,k} &= \tau_0 \frac{W_0 - k}{W_0} \text{for }\; k = 1\,..\,W_0 - 1 \\
x_{m+1,0,0} &= \tau_0 C_2^{m+1} \\
\sum_{l=0}^{\overline{D_d}-1} x_{i,2,0,l} &= \frac{\tau_0 C_2^i P_d(\overline{D_d} - 1)}{2}
\end{aligned}$$

The sum of probabilities for one transmission sub-chain is:

$$\begin{aligned}
s_t = &\sum_{i=0}^{m} \sum_{k=0}^{W_i - 1} x_{i,2,k} + \sum_{i=0}^{m} x_{i,0,0} + \sum_{i=0}^{m} x_{i,1,0} + x_{m+1,0,0} \\
&+ \sum_{i=0}^{m} \sum_{l=0}^{\overline{D_d}-1} x_{i,2,0,l}
\end{aligned} \tag{1.2}$$

which can further be simplified to:

$$\begin{aligned}
s_t = &\tau_0 C_4 \left(C_3 \overline{D_d} + C_1 + + \frac{P_d(\overline{D_d} - 1)}{2} \right) \\
&+ \sum_{i=0}^{m} \frac{C_2^i (W_i + 1)}{2} + C_2^{m+1}
\end{aligned} \tag{1.3}$$

The sum of probabilities within the beacon synchronization line is equal to $s_b = \tau_0 \gamma \delta \sum_{i=0}^{SD} \frac{i}{SD} = \tau_0 \gamma \delta (SD + 1)/2$ and the sum of probabilities for the collision avoidance line is equal to $s_c = \tau_0 \gamma \delta SD/4 + 1/2$.

In order to model node's sleep time we will assume that sleep time is geometrically distributed with parameter P_{sleep}. Then the sum of probabilities of being in single sleep is equal to $s_{s1} = \tau_0 \gamma \delta / (1 - P_{sleep})$. However, if node wakes up from sleep and finds its buffer empty it will start the new sleep. We will denote the probability of finding empty buffer after sleep as Q_c and derive it later. The sum of of probabilities of being in consecutive sleep then becomes $s_s = \tau_0 \gamma \delta / ((1 - P_{sleep})(1 - Q_c))$. If we denote the threshold value of the number of packets sent using the same key as n_k then the normalization condition for the whole Markov chain becomes:

$$3(s_b + 2s_t) + 2s_t + n_k(s_s + s_t) = 1 \tag{1.4}$$

However, the total access probability by the node is equal to the sum of access probabilities in each transaction i.e.:

$$\tau = (8 + n_k)\tau_0 \tag{1.5}$$

1.4.1. *Analysis of node's packet queue*

In order to find probability Q_c we need to consider node's MAC layer as $M/G/1/K$ queuing model with vacations and set-up time. We assume that when node wakes up it will transmit only one packet and go to sleep again which is known as 1-limited scheduling.[8] Detailed model for the more general sleep policy with Bernoulli scheduling of activity period is derived in.[6] In Bernoulli scheduling, after one packet transmission, node decides to transmit another packet with probability P_{ber} and goes to sleep with probability $1 - P_{ber}$. We can apply this approach to our model using restriction that $P_{ber} = 0$. In the discussion that follows, packets are arriving to each node following the Poisson process with the rate λ.

Consider the Probability Generating Function for one geometrically distributed sleep period (with parameter P_{sleep} as:

$$V(z) = \sum_{k=1}^{\infty}(1 - P_{sleep})P_{sleep}^{k-1}z^k = \frac{(1 - P_{sleep})z}{1 - zP_{sleep}} \tag{1.6}$$

and the mean duration of the vacation is $\overline{V} = V'(1) = 1/(1 - P_{sleep})$. We also note[8] that the PGF for the number of packet arrivals to the sensor buffer during the sleep time is equal to:

$$F(z) = V^*(\lambda - z\lambda) \tag{1.7}$$

where $V^*()$ denotes the Laplace-Stieltjes Transform (LST) of the sleep time which (since sleep time is discrete random variable) can be obtained by substituting the variable z with e^{-s} in the expression for $V(z)$.

A node returning from sleep (i.e., with non-empty buffer) has to synchronize with the next beacon; the synchronization time is uniformly distributed between 0 and $BI - 1$ backoff periods, and its PGF is

$$D_1(z) = \frac{1 - z^{BI}}{BI(1 - z)} \tag{1.8}$$

When awakened node finds the next beacon, then it has to wait for collision separation time before it starts its backoff procedure. The PGF for this collison separtion time is:

$$D_2(z) = \frac{1 - z^{BI/2}}{BI/2(1 - z)} \tag{1.9}$$

The total idle time when node is awakened then has the PGF:

$$D(z) = D_1(z)D_2(z) \tag{1.10}$$

Its LST (Laplace-Stieltjes transform) will be denoted as $D^*(s)$, the corresponding probability distribution function $D(x)$, and the probability density function as $d(x)$. The PGF for packet service time will be denoted as $T_t(z)$ and its probability density function will be denoted as $dt_t(x)$. $T_t(z)$ derived in the Appendix.

Let us now analyze the operation of system, starting from Markov points which include moments of packet departure and moments when the server wakes up (i.e., ends its vacation). Let $V^*(s)$ denote the LST of the vacation time, with the corresponding probability distribution function $V(x)$ and the probability density function $v(x)$.

The PGFs for the number of packet arrivals to the node's buffer during the total idle time and packet service time respectively are:

$$\begin{aligned} S(z) &= \sum_{k=0}^{\infty} s_k z^k = \int_0^{\infty} e^{-x\lambda(1-z)} d(x) = D^*(\lambda - z\lambda) \\ A(z) &= \sum_{k=0}^{\infty} a_k z^k = \int_0^{\infty} e^{-x\lambda(1-z)} dt_t(x) = T_t^*(\lambda - z\lambda) \end{aligned} \tag{1.11}$$

Then, the probabilities of k packet arrivals to the node's buffer during the synchronization time, packet service time and sleep time, denoted with d_k, a_k and f_k, respectively, can be obtained as $s_k = \frac{1}{k!}\frac{d^k S(z)}{dz^k}\Big|_{z=0}$, $a_k = \frac{1}{k!}\frac{d^k A(z)}{dz^k}\Big|_{z=0}$, $f_k = \frac{1}{k!}\frac{d^k F(z)}{dz^k}\Big|_{z=0}$.

Let π_k and q_k denote the steady state probabilities that there are k packets in the device buffer immediately upon a packet departure and after returning from vacation, respectively. Then, the steady state equations for state transitions are

$$
\begin{aligned}
q_0 &= (q_0+\pi_0)f_0, \\
q_k &= (q_0+\pi_0)f_k + \textstyle\sum_{j=1}^{k}\pi_j f_{k-j}, && \text{for } 1 \le k \le L-1 \\
q_L &= (q_0+\pi_0)\sum_{k=L}^{\infty} f_k + \sum_{j=1}^{L-1}\pi_j \sum_{k=L-j}^{\infty} f_k \\
\pi_k &= \sum_{j=1}^{k+1} q_j \sum_{l=0}^{k-j+1}(s_l + a_{k-j+1-l}), && \text{for } 0 \le k \le L-2 \\
\pi_{L-1} &= \sum_{j=1}^{L} q_j \sum_{k=L-j}^{\infty}\sum_{l=0}^{k}(s_l + a_{k-l}) \\
1 &= \sum_{k=0}^{L} q_k + \sum_{k=0}^{L-1}\pi_k
\end{aligned}
\tag{1.12}
$$

The probability distribution of the device queue length at the time of packet departure $\pi_i, i = 0\,..\,L-1$ and return from the sleep $q_i, i = 0\,..\,L$ can be found by solving the system of linear equations (1.12). In this manner, we obtain the probability that the Markov point corresponds to a return from the vacation and the queue is empty at that moment is

$$Q_c = q_0 / \sum_{i=0}^{L} q_i. \tag{1.13}$$

Given that there are n nodes in the cluster the total event sensing reliability is equal to:

$$R = n_k \gamma \delta \tau_0 / t_{boff} \tag{1.14}$$

where $t_{boff} = 0.32$ms corresponds to the duration of one backoff period. Value R has to be set by the sensing application e.g. $R = 10$. Satisfying equation (1.14) will result in minimal energy consumption. However, we have to note that key exchange overhead will result in overhead packet rate of $8\tau_0\delta\gamma/t_{boff}$ packets per second.

1.5. Medium behavior and packet service time

The probability to access the medium when the transmission is deferred to the next superframe due to insufficient time is $\tau_1 = (SD - \overline{D_d} + 1)\dfrac{P_d}{C_3}\tau$, (because this access can occur only in the third backoff period of the superframe and τ is average accross the whole accessible part of the superframe) and the probability to access the medium in the current superframe is $\tau_2 = (1 - P_d/C_3)\,\tau$.

At any moment, $n-1-q$ stations out of $n-1$ are delayed to the start of next superframe due to the insufficient space in the current superframe. Numbers q and $n-1-q$ follow a binomial distribution with probability $P_q = \binom{n-1}{q}(1-P_d)^q P_d^{n-1-q}$.

In order to calculate the probability α that the medium is idle on the first CCA test, we have to find the mean number of busy backoff periods within the superframe; this number will be divided into the total number of backoff periods in the superframe wherein the first CCA can occur. Note that the first CCA will not take place if the remaining time in the superframe is insufficient to complete the transaction, which amounts to $SM = SD - \overline{D_d} + 1$ backoff periods. Then, the probability that any (one or more) packet transmissions will take place at the beginning of the superframe is

$$n_1 = 1 - (1-\tau_1)^{(n-1-q)}(1-\tau_2)^q \tag{1.15}$$

and the number of busy backoff periods due to these transmissions is $n_1(\overline{G_p} + \overline{G_a})$.

The occupancy of the medium after the first transmission time can be found by dividing the superframe into chunks of $\overline{D_d}$ backoff periods and calculating the probability of transmission within each chunk. As the total arrival rate of non-deferred packets is $q\tau_2$, the probability that the number of transmission attempts during the period $\overline{D_d}$ will be non-zero is $n_2 = \overline{D_d}q(\tau_2)$. The total number of backoff periods in which the first CCA can be occur is $SM = SD - \overline{D_d} + 1$. The probability that the medium is idle at the first CCA is

$$\alpha = \sum_{q=0}^{n-1} P_q \left(1 - \left(\frac{n_1\overline{D_d}}{SM} + \frac{n_2(SD - 2\overline{D_d} + 1)}{SM}\right)\right.$$
$$\left. \cdot \frac{(\overline{G_p} + \overline{G_a})}{\overline{D_d}}\right) \tag{1.16}$$

The probability that the medium is idle on the second CCA for a given node is, in fact, equal to the probability that neither one of the remaining $(n-1)$ nodes, nor the coordinator, have started a transmission in that backoff period. The second CCA can be performed in any backoff period from the second backoff period in the superframe, up to the period in which there is no more time for packet transmission, which amounts to SM. The probability in question is

$$\beta = \sum_{q=0}^{n-1} P_q \Bigg(\frac{1}{SM} + \frac{(1-\tau_1)^{(n-1-q)}(1-\tau_2)^q}{SM} + \frac{SM-2}{SM}(1-\tau_2)^q \Bigg) \tag{1.17}$$

Finally, the probability γ that a packet will not collide with other packet(s) that had successful first and second CCAs can be calculated as the probability that there are no accesses to the medium by the other nodes or the coordinator during the period of one complete packet transmission time. (Note that a collision can happen in SM consecutive backoff periods starting from the third backoff period in the superframe.)

$$\gamma = \sum_{q=0}^{n-1} P_q \left(\frac{(1-\tau_1)^{\overline{D_d}(n-1-q)}(1-\tau_2)^{\overline{D_d}q}}{SM} + \frac{SM-1}{SM}(1-\tau_2)^{\overline{D_d}q} \right) \tag{1.18}$$

Regarding the PHY layer, we should expect BER slightly less than 10^{-4}. This is confirmed in the section 6.1.6 of the standard where Packet Error Rate (PER) of 1% is expected on packets which have 20 octets including MAC and PHY level headers. However, in the presence of interference in the ISM band, it is more realistic to expect BER around 10^{-3} and Packet Error Rate equal to $PER = 1 - (1 - BER)^X$ where X is packet length including MAC and physical layer header expressed in bits.

Packet transmission will be corrupted by noise when either data packet is corrupted or acknowledgement packet is corrupted. The probability that data transmission is not corrupted is

$$\delta = (1 - BER)^{X_d + X_a} \tag{1.19}$$

where X_d and X_a are lengths, in bits, of data packet and acknowledgement packet respectively (including all headers).

Packet service time In order to derive this distribution, we begin by modeling the effect of freezing the backoff counter during the inactive period of the superframe. The probability that a backoff period is the last one within the active superframe is $P_{last} = \frac{1}{SD}$, and the PGF for the effective duration of the backoff period, including the duration of the beacon frame, is

$$B_{off}(z) = (1 - P_{last})z + P_{last}z^{(BI-SD+1)}B_{ea}(z) \tag{1.20}$$

The PGF for the duration of i-th backoff attempt is

$$B_i(z) = \sum_{k=0}^{W_i-1} \frac{1}{W_i} B_{off}^k(z) = \frac{B_{off}^{W_i}(z) - 1}{W_i(B_{off}(z) - 1)} \tag{1.21}$$

As noted above, the transmission procedure will not start unless it can be finished within the current superframe. The number of backoff periods wasted due to the insufficient space in the current superframe can be described with the PGF of $B_p(z) = \frac{1}{D_d}\sum_{k=0}^{D_d-1} z^k$, and the PGF of the data packet transmission time for deferred and non-deferred transmissions, respectively, is

$$\begin{aligned} T_{d1}(z) &= B_p(z)z^{(BI-SD)}B_{ea}(z)G_p(z)t_{ack}(z)G_a(z) \\ T_{d2}(z) &= G_p(z)t_{ack}(z)G_a(z) \end{aligned} \tag{1.22}$$

For simplicity, let us denote the probability that a backoff attempt will be unsuccessful as $R_{ud} = 1 - P_d - (1 - P_d)\alpha\beta$. The function that describes the time needed for the backoff countdown and the transmission attempt itself can be presented in the following equation:

$$\begin{aligned} P(z) = &\sum_{i=0}^{m} \prod_{j=0}^{i} \left(B_j(z)R_{ud}\right) z^{2(i+1)} \left(P_d T_{d1}(z) + (1 - P_d)\alpha\beta T_{d2}(z)\right) \\ &+ R_{ud}^{m+1} \prod_{j=0}^{m} B_j(z) z^{2(m+1)} P(z) \end{aligned} \tag{1.23}$$

where R_{ud}^{m+1} denotes the probability that $m+1$ backoff attempts with non-decreasing backoff windows were not successful and the sequence of backoff windows has to be repeated starting from the smallest backoff window. From equation (1.23) we obtain:

$$P(z) = \frac{\sum_{i=0}^{m} \prod_{j=0}^{i} \left(B_j(z)R_{ud}\right) z^{2(i+1)} \left(P_d T_{d1}(z) + (1 - P_d)\alpha\beta T_{d2}(z)\right)}{1 - R_{ud}^{m+1} \prod_{j=0}^{m} B_j(z) z^{2(m+1)}} \tag{1.24}$$

The partial PGF for the time spent in unsuccessful transmissions takes the value:

$$P_2(z) = (1 - \gamma\delta)P(z)T_t(z)$$

where $T_t(z)$, presents PGF of data packet service time. Again, the last function holds only if the bridge uses the CSMA-CA access mode. Then the PGF for the packet service time can be found from the equation:

$$\begin{aligned} T_t(z) &= (P(z) + P_2(z))\left(1 + R_{ep}(z) + R_{ep}^2(z) + R_{ep}^3(z)\ldots\right) \\ T_t(z) &= \frac{P(z)}{1 - R_{ep}(z)} \end{aligned} \tag{1.25}$$

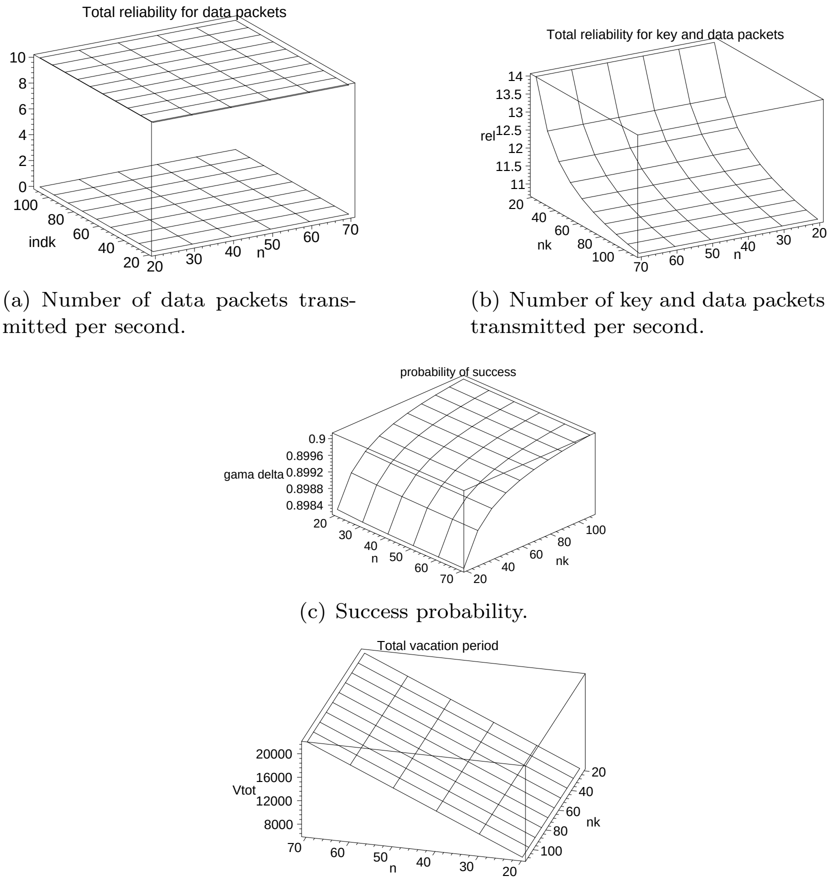
where $R_{ep}(z) = \prod_{j=0}^{m} B_j(z) z^{2(m+1)} R_{ud}^{m+1}$ represents partial PGF of $m+1$ unsuccessful backoff countdown iterations without a transmission attempt.

1.6. Performance evaluation

In this section we present numerical results obtained by solving the system of equations (1.4), (1.5), (1.16), (1.17), (1.18), (1.25), (1.12), (1.13) and (1.14) for system parameters τ_0, τ, P_{sleep}, α, β, γ and Q_c. We have varied the key exchange threshold between 20 and 110 packets while the requested event sensing reliability was kept at $R = 10$ packets per second. Cluster size was varied between 20 and 70 nodes.

We have assumed that the network operates in the ISM band at 2.45GHz, with raw data rate 250kbps and $BER = 10^{-4}$. Superframe size was controlled with SO,BO= 0. The packet size has been fixed at $\overline{G_p} = 12$ backoff periods, while the device buffer had a fixed size of $L = 2$ packets. The packet size includes Message Authentication Code and all physical layer and Medium Access Control protocol sublayer headers, and is expressed as the multiple of the backoff period.[1] We also assume that the physical layer header has 6 bytes, and that the Medium Access Control sublayer header and Frame Check Sequence fields have a total of 9 bytes.

Figure 1.6(b) shows total number of successfully transmitted packets (including key and data information) transmitted per second for requested data reliability of $R = 10$ packets per second (which is shown on Fig. 1.6(a)). We note that the total number of packets hyperbolically grows when the key exchange threshold decreases linearly Fig. 1.6(b). This is intuitive since the frequency of key updates is R/n_k per second and number of overhead packets with key information per second is equal to $8R/n_k$. We note that



(a) Number of data packets transmitted per second.

(b) Number of key and data packets transmitted per second.

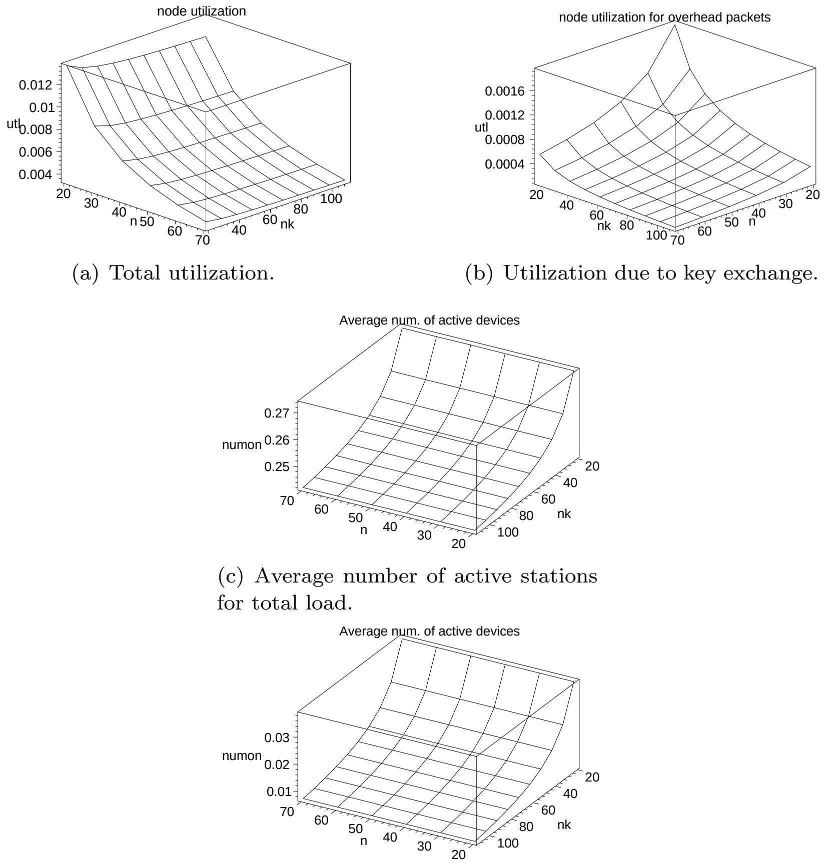
(c) Success probability.

(d) Inactive period.

Fig. 1.6. Event sensing reliability for data and key+data, success probability and inactive period under SKKE

key exchange overhead becomes negligible only for $n_k \geq 90$. Probability that packet will not suffer from collision or noise error sharply drops when threshold for key exchange drops below 40 packets. Both the reliability overhead and success probability depend only on the requested event sensing reliability except for very small key update threshold. Sleep period, on the other hand, depends mostly on the number of alive nodes and impact of key exchange overhead is barely noticeable.

Total node utilization shown in Fig. 1.7(a) depends mostly on the num-



(a) Total utilization.

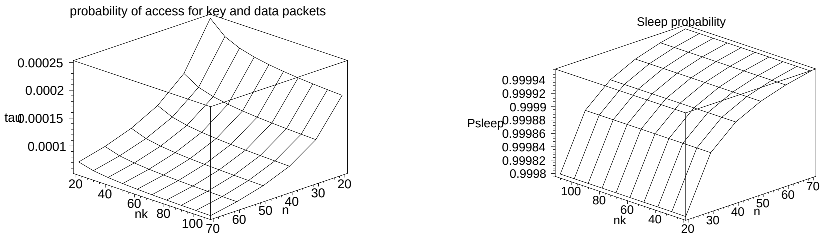
(b) Utilization due to key exchange.

(c) Average number of active stations for total load.

(d) Average number of active stations for overhead load.

Fig. 1.7. Power expenditure indicators.

ber of alive nodes, but it also increases with increase of the number of key exchanges per second and exact impact of the key exchange overhead is shown in Fig.1.7(b). Number of active nodes shown in Fig. 1.7(c) does not depend on the number of alive nodes since R is constant, but it increases with $8R/n_k$ due to increased overhead which is separately shown in Fig. 1.7(d). Finally, medium access probability to the medium and sleep probability for each node are shown in Fig. 1.8. As expected, medium access probability changes as $8R/n_k$ and it also changes approximately with rate $1/n$ when number of nodes n changes. Sleep probability dominantly



(a) Access probability for node including key and data.

(b) Sleep probability.

Fig. 1.8. Medium access probability and sleep probability for a node.

changes with n, while the changes with n_k are much milder.

1.7. Conclusion

In this Chapter we have developed analytical model of key exchange integrated into the sensing function of beacon enabled 802.15.4 cluster. Our results show important impact of the ratio of the event sensing reliability and key update threshold on the cluster's energy consumption. We have evaluated the impact of the threshold for key update on the cluster's descriptors. The results can give useful hints for the choice of frequency of key updates for required event sensing reliability.

References

1. ieee802154. Standard for part 15.4: Wireless MAC and PHY specifications for low rate WPAN. IEEE Std 802.15.4, IEEE, New York, NY (Oct., 2003).
2. ZigBee. ZigBee specification. ZigBee Document 053474r06, ZigBee Alliance, San Ramon,CA, (2005).
3. M. Khan, F. Amini, J. Mišić, and V. Mišić. The cost of security: Performance of zigbee key exchange mechanism in an 802.15.4 beacon enabled cluster. In *Proc. WSNS'06, held in conjunction with IEEE MASS06 2006*, Vancouver, CA, (2006).
4. I. Stojmenović, Ed., *Handbook of Sensor Networks: Algorithms and Architectures.* (John Wiley & Sons, New York, NY, 2005).
5. Y. Sankarasubramaniam, Ö. B. Akan, and I. F. Akyildiz. ESRT: event-to-sink reliable transport in wireless sensor networks. In *Proc. 4th ACM MobiHoc*, pp. 177–188, Annapolis, MD (June, 2003).
6. J. Mišić, S. Shafi, and V. B. Mišić, Maintaining reliability through activity

management in an 802.15.4 sensor cluster, *IEEE Transactions on Vehicular Technology.* **55**(3), 779–788 (May, 2006).

7. A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography.* (CRC Press, 1997).
8. H. Takagi, *Queueing Analysis.* vol. 1: Vacation and Priority Systems, (North-Holland, Amsterdam, The Netherlands, 1991).