CHAPTER 16

IMPACT OF RELIABLE AND SECURE SENSING ON CLUSTER LIFETIME IN IEEE 802.15.4 NETWORKS

JELENA MIŠIĆ

16.1 INTRODUCTION

In order to penetrate the market with cost-effective solutions for wireless sensor networks (WSNs) we need standardized low cost, low power, and short-range AU1 communication [low rate wireless personal area network (LR-WPAN)] technology. An important candidate for the application in this area is the 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 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 carrier sense multiple-access/collision avoidance (CSMA/CA) mode.

In this chapter we address the problem of multicluster sensor network as shown in Fig. 16.1 with integrated node sleep control and key exchange mechanism. The network is formed by three clusters interconnected in a master–slave regime wherein the coordinator of a lower cluster acts as the bridge to the upper one, and the coordinator of the topmost cluster acts as the network sink. In our previous work [3] we analyzed the impact of contention caused by the bridges and ordinary nodes on the cluster lifetimes. In this chapter we include the model of additional traffic caused by key exchanges in

Emerging Wireless LANs, Wireless PANs, and Wireless MANs. Edited by Y. Xiao and Y. Pan

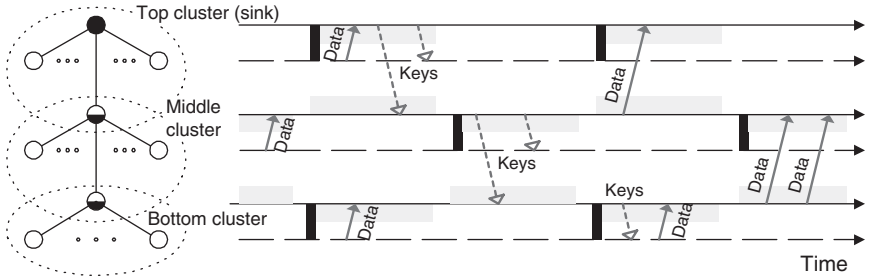

FIGURE 16.1 Network topology.

the network where each cluster has to deliver R packets per second toward the sink. All clusters are equipped with redundant sensors, which enables reduction of individual sensor duty cycle through activity management [4]. In other words, each node spends most of its time in sleep mode and wakes up only to transmit its packets. Since contention between the bridge and ordinary nodes cause nonuniform lifetimes among the clusters, we attempt to find cluster populations that will compensate bridges' activitites due to data delivery and excessive key exchanges.

Individual sensor nodes are battery operated, and their power consumption is modeled according to tmote_sky ultra-low power IEEE 802.15.4–compliant wireless sensor module [5] powered with two AA batteries. Since the coordinators/bridges have to work without ever going to sleep, their power budget is assumed to be infinite; the use of relaying nodes with larger power resources than ordinary sensing nodes has been shown to increase the useful network lifetime [6].

The chapter is organized as follows: Section 16.2 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 16.3. In Section 16.4 we briefly discuss bridge operation. Energy consumption for tmote_sky ultra-low power IEEE 802.15.4–compliant wireless sensor module [5], which we will consider in our modeling, is considered in Section 16.6. Section 16.5 presents derivation of the analytical model of the cluster, while Section 16.7 presents numerical results obtained from the analysis. Finally, Section 16.8 concludes the chapter.

16.2 802.15.4 BEACON-ENABLED MESSAGE AUTHENTICATION CODE (MAC)

AU2

The 802.15.4 networks with star topology are operated in a beacon-enabled mode where channel time is divided into superframes bounded by beacon

AU3 transmissions from the personal area network (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. The standard supports 16 different frequency channels in which clusters can operate within the industrial/scientific/medical (ISM) band. 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. 16.2*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 acknowledgment. After receiving the acknowledgment, the node listens for the actual data packet for the period of *aMaxFrameResponseTime*, during which the coordinator must send the data frame.

Power management consists of adjusting the frequency and ratio of active and inactive periods of sensor nodes [7, 8]. 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, the coordinator has to periodically broadcast required event-sensing reliability (number of packets per second needed for reliable event detection) and the number of nodes that are alive.

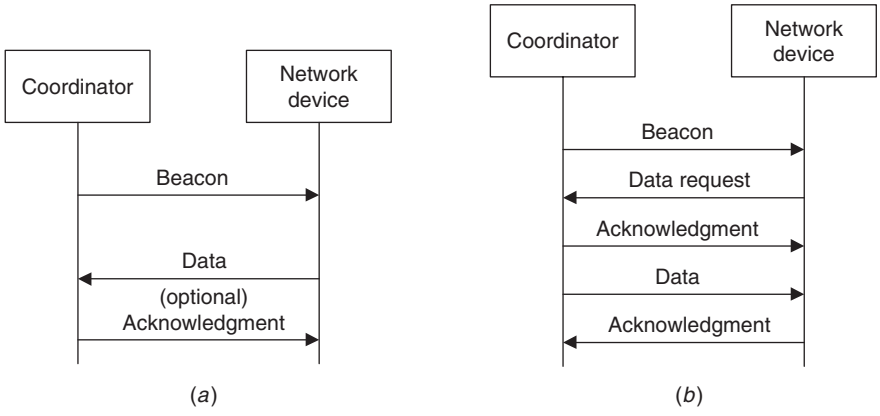

FIGURE 16.2 Data transfers in 802.15.4 PAN in beacon-enabled mode: (*a*) uplink transmission; (*b*) downlink transmission.

AU8 **TABLE 16.1**

Current Consumption		
Radio transmitting at 0 dBm	ω_t	15.8 μJ
Radio transmitting at −1 dBm	ω_t	15.0 μJ
Radio transmitting at −3 dBm	ω_t	13.8 μJ
Radio receiving	ω_r	17.9 μJ
Idle mode, oscillator off	ω_s	18.2 nJ

Based on that information, the node can calculate average period of sleep between transmissions. When the average sleep period is known, then some discrete random probability distribution can be used to generate individual sleep durations [9]. When the node wakes up and has a packet to transmit, it turns its receiver on in order to synchronize with the beacon. If the node's buffer is empty, it will start the new sleep. After receiving the information from the beacon, the node turns the transmitter on and starts backoff count in order to transmit the packet. After packet transmission, the node turns the receiver on in order to receive the acknowledgment. After the positive acknowledgment, the node starts the new sleep period. If the packet was not received correctly, the node has to repeat the transmission. Since the minimal beacon size is two backoff periods, we assume that an additional backoff period (10 bytes) is sufficient for transmitting information about the number of live nodes and requested event-sensing reliability. Let us denote power consumptions as ω_s, ω_r, and ω_t joules per one backoff period during sleep, receiving, and transmitting, respectively. They can be derived from typical operating conditions reported in documentation for ultra-low power IEEE 802.15.4 sensor module operating in ISM band between 2400 and 2483.5 MHz [5] and shown in Table 16.1. According to the specification of the tmote_sky module, two AA batteries are needed in order to supply voltage between 2.1 and 3.6 V.

The 802.15.4 standard specifies several security suites that 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 prestored key, message integrity code generated using the prestored 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 the 802.15.4 MAC layer.

16.3 SKKE PROTOCOL

A low cost alternative for this task with the 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 a 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 sublayer 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 the coordinator and the 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 preinstalled 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 advanced encryption standard (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 [10]. The hash function of a data block d will be denoted as $H(d)$. The ZigBee specification suggests the use of the keyed 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 $MacTag = MAC_{MacKey}MacData$.

AU4

The SKKE protocol is initiated by the PAN coordinator (denoted as initiator device U) by exchanging ephemeral data (Fig. 16.3). 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

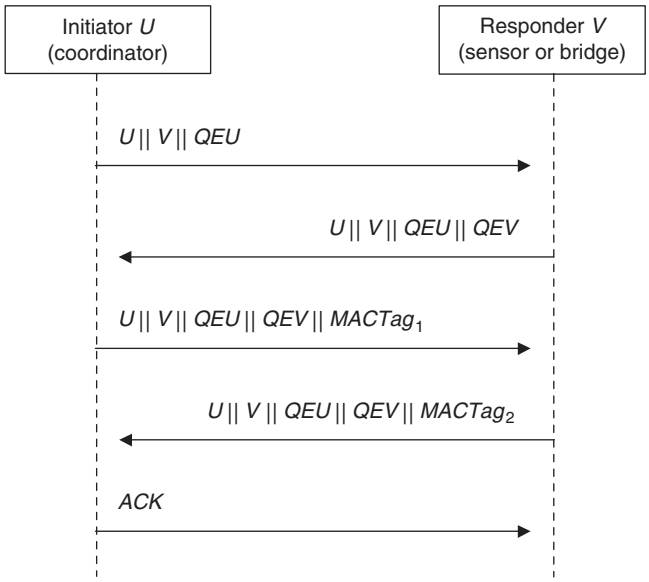

FIGURE 16.3 SKKE protocol between PAN coordinator and node.

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, that is, $Hash_1 = H(Z||01_{16}) Hash_2 = H(Z||02_{16})$. The $Hash_2$ will be the link key among two devices, while $Hash_1$ will be used to confirm that both parties have reached the same link key.

16.4 BRIDGING THE CLUSTERS

Consider the network shown in Fig. 16.1, operating in the ISM band at 2.4 GHz (other bands can be used but we don't consider them here). We assume that all clusters operate in beacon-enabled, slotted CSMA/CA mode under the control of their respective cluster (PAN) coordinators. In each cluster, the channel time is divided into superframes bounded by beacon transmissions from the coordinator [1]. All communications in the cluster take place during the active portion of the superframe, the duration of which is referred to as the superframe duration SD, as shown in Fig. 16.4.

The basic time unit of the MAC protocol is the duration of the so-called backoff period. Access to the channel can occur only at the boundary of the backoff period. The actual duration of the backoff period depends on the

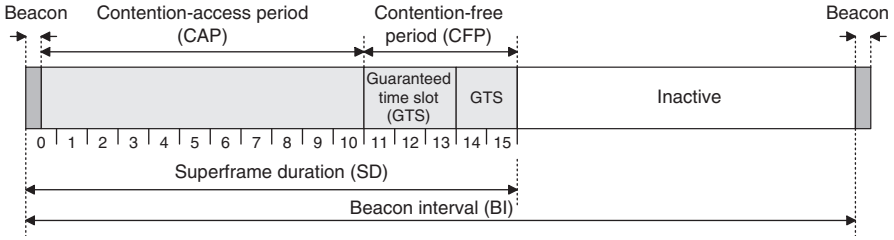

FIGURE 16.4 Composition of superframe under IEEE Std 802.15.4. (Adapted from IEEE 802.15.4-2006, "Wireless MAC and PHY specifications for low rate WPAN," IEEE, New York, 2006.)

frequency band in which the 802.15.4 wireless PAN (WPAN) is operating. Namely, the standard allows the PAN to use one of three frequency bands: 868–868.6, 902–928, or 2400–2483.5 MHz. In the two lower frequency bands, binary phase shift keying (BPSK) modulation is used, giving the data rate of 20 and 40 kbps, respectively. Each data bit represents one modulation symbol, which is further spread with the chipping sequence. In the third band, the AU5 O-QPSK modulation is used before spreading; in this case, four data bits comprise one modulation symbol, which is further spread with the 32-bit speading sequence. Table 16.2 summarizes the basic timing relationships in the MAC sublayer. Note that the constants and attributes of the MAC sublayer, as defined by the standard, are written in italics. Constants have a general prefix of *a*, for example, *aUnitBackoffPeriod*, while attributes have a general prefix of *mac*, for example, *macMinBE*.

AU9 **TABLE 16.2 Basic Timing Relationships in MAC Sublayer**

Type of Time Period	Duration	MAC Constant
Modulation symbol	1 data bit in 860- and 915-MHz bands, 4 data bits in 2.4-GHz band	N/A
Unit backoff period	20 symbols	*aUnitBackoffPeriod*
Basic superframe slot (SO = 0)	Three unit backoff periods (60 symbols)	*aBaseSlotDuration*
Basic superframe length (SO = 0)	16 basic superframe slots (960 symbols)	*aBaseSuperframeDuration* = *NumSuperframeSlots* × *aBaseSlotDuration*
(Extended) superframe duration, SD	*aBaseSuperframeDuration* $\times 2^{SO}$	*macSuperframeOrder*, *SO*
Beacon interval, BI	*aBaseSuperframeDuration* $\times 2^{BO}$	*macBeaconOrder*, BO

As shown in Table 16.2 the superframe is divided into 16 slots of equal size, each of which consists of 3×2^{SO} backoff periods. The variable SO, also known as *macSuperframeOrder*, determines the duration of the superframe; its default value of $SO = 0$ corresponds to the shortest active superframe duration of 48 backoff periods. In the ISM band, the duration of the backoff period is 0.32 ms for a payload of 10 bytes, which results in the maximum data rate of 250 kbps. The time interval between successive beacons is $BI = aBaseSuperframe\ Duration * 2^{BO}$, where $aBaseSuperframeDuration = 48$ backoff periods ($SO = 0$) and BO denotes the so-called *macBeaconOrder*, which can take values between 1 and 14. The duration of the inactive period of the superframe can easily be determined as $I = aBaseSuperframeDuration * (2^{BO} - 2^{SO})$. The default access mode in beacon-enabled operation is slotted CSMA/CA, with some slots optionally reserved for certain nodes.

During the inactive portion of the superframe, any device may enter a low power mode; the cluster coordinator can switch to the upper cluster in order to perform the bridging function—that is, deliver the data to the coordinator of the upper cluster. All *lower* cluster coordinators use this facility to perform the bridging function. As soon as the active part of the superframe is completed in the lower cluster, the coordinator/bridge switches to the *upper* cluster and waits for the beacon so that it can deliver the data from the lower cluster to the upper cluster coordinator/network sink.

All clusters use the CSMA/CA access, which means that the bridge has to compete for medium access with ordinary nodes in the upper cluster. As soon as the data is delivered, the bridge can return to its own cluster. This also means that, should the bridge be unable to transmit its data when the (active portion

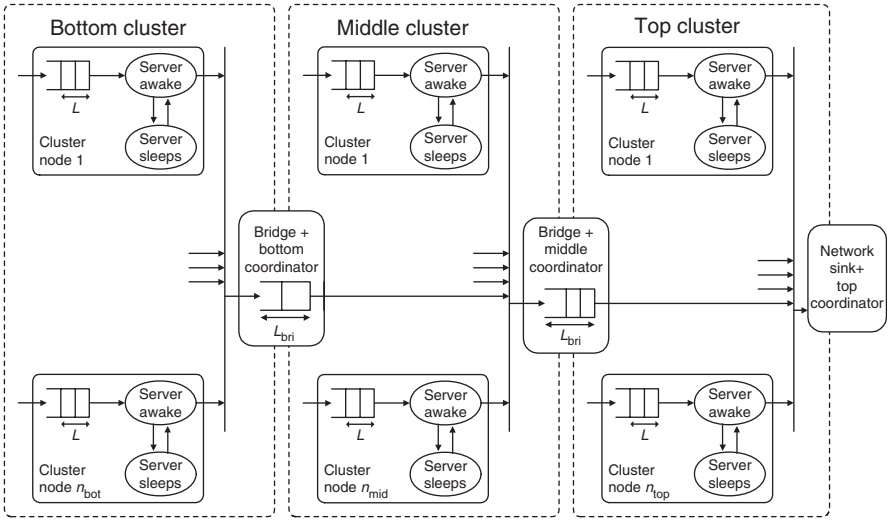

FIGURE 16.5 Queueing model of bridging process among three clusters.

of the) superframe in the upper cluster ends, it will freeze its backoff counter and leave the upper cluster. The backoff countdown will resume when the bridge returns to the upper cluster for the next superframe. Upon returning to the lower cluster, the bridge transmits the beacon, denoting the beginning of the next superframe, and the lower cluster continues to operate.

AU6

Bridge switching is schematically presented in Fig. 16.5. As can be seen, the three clusters have to operate with the same beacon interval, and the time between successive bridge visits to the "upper" cluster is therefore the same as the period between two beacons in its own, "lower" cluster. If the top and bottom clusters are far enough, that is, beyond the transmission range of each other, all three clusters may use the same radio frequency (RF) channel (the 802.15.4 standard uses 16 channels in the ISM band). Note that obtaining an increased area of coverage is the main reason for using a multicluster configuration. If the clusters are closer to each other, the top and bottom cluster may use different channels, and the middle cluster can use either of these.

16.5 ANALYTICAL MODEL FOR ORDINARY NODE IN CLUSTER WITH SKKE

In this section we will develop a Markov chain model for node behavior that includes all phases of the SKKE protocol and subsequent sleep and transmission phases. We assume that the PAN coordinator maintains a separate counter for the number of transmissions by each node. When 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. 16.6.

Furthermore, each of the steps that involves downlink transmission requires synchronization with the beacon, transmission of the uplink request packet, and transmission of the downlink packet as shown in Fig. 16.2*b*. Every transmission is implemented using slotted CSMA/CA specified by the standard [1]. Markov subchain for single CSMA/CA transmission (as the component of the Fig. 16.6) is shown in Fig. 16.7. The delay line from Fig. 16.7 models the requirement from the standard that every transmission that cannot be fully completed within the current superframe has to be delayed to the beginning of the next superframe and is shown in Fig. 16.8*a*. The probability that a 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 acknowledgment, and acknowledgment 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 subchain, 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

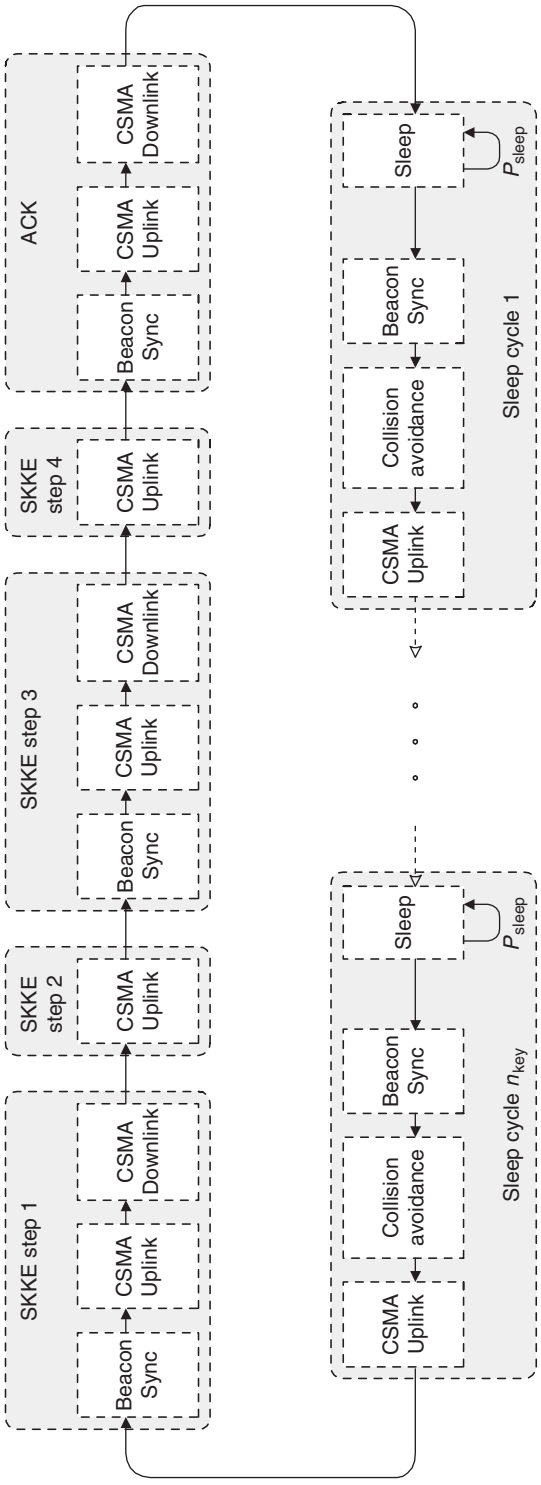

FIGURE 16.6 Markov chain for node behavior under threshold triggered key exchange.

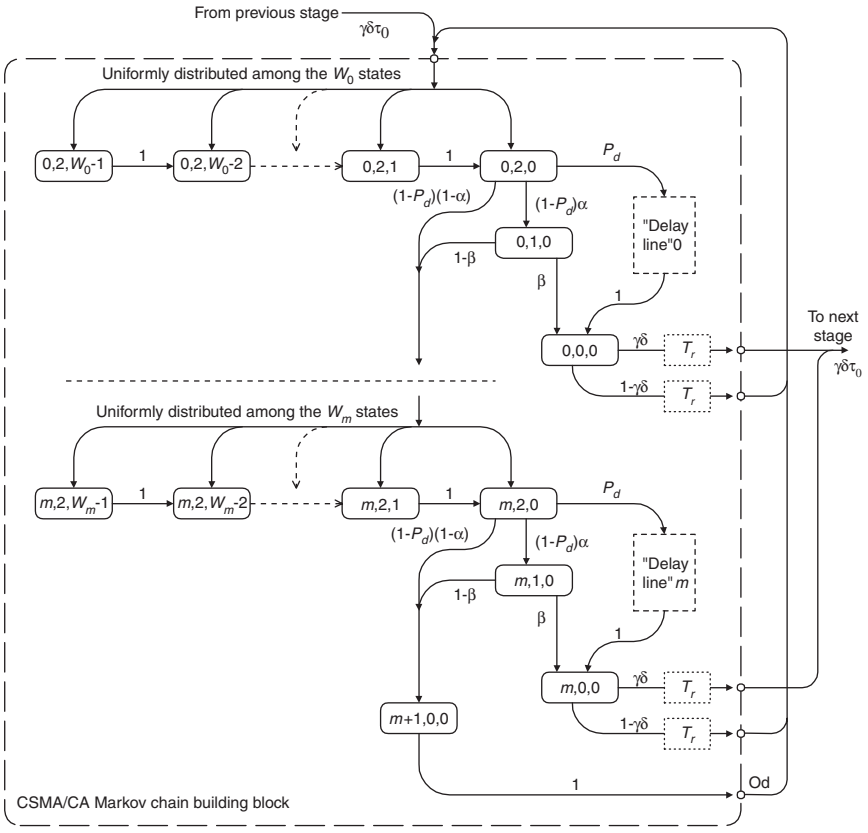

FIGURE 16.7 Markov subchain for single CSMA/CA transmission.

backoff attempt, where m is a constant defined by MAC with default value 4; and $c \in (0, 1, 2)$ is the index of the current clear channel assessment (CCA) phase. The standard prescribes two CCAs after the backoff countdown and if both are successful, trasnmission can start; and $k \in (0..W_i - 1)$ is the value of backoff counter, with W_i being the size of backoff window in ith 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, $macMinBE = 3$); and $d \in (0..\overline{D}_d - 1)$ 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.

We need also to include synchronization time from the moment when the node wakes up till the next beacon, shown in Fig. 16.8*b*, as well as the uniformly distributed time needed to separate potential collisions among the nodes that wake up in the same superframe. One may argue that this separation time is not needed since CSMA/CA random backoff times will do the separation, but both request packets and data packets by awakened nodes will start backoff count

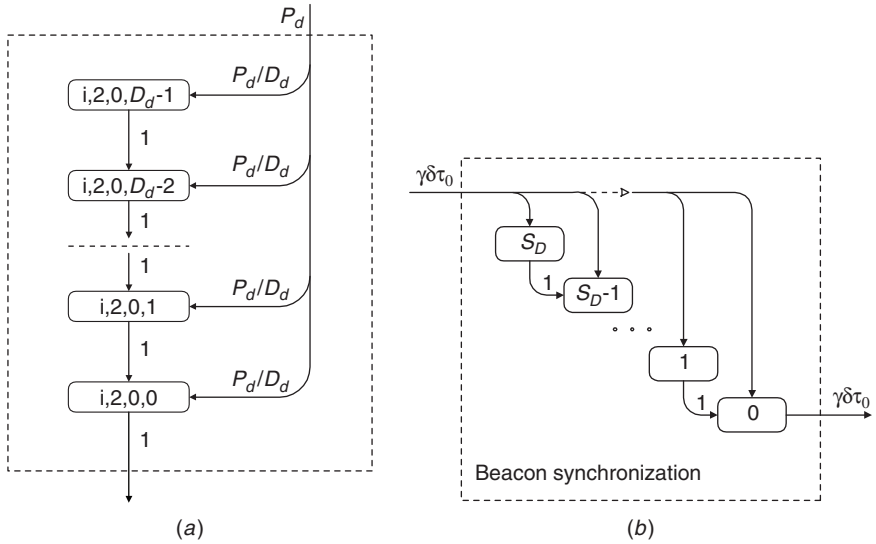

FIGURE 16.8 Delay and synchronization lines: (*a*) Markov subchain for delayed transmissions; (*b*) synchronization with beacon.

immediately after the beacon with backoff window, which has a range from 0 to 7 backoff periods. Due to the small backoff window, collisions will be likely, and we think that additional separation is needed. The collision separation line is similar to the beacon synchronization line except that the delay range is from 0 to $SD/2$ backoff periods. Synchronization with the beacon is also needed to receive the acknowledgment from the coordinator that whole SKKE transaction is completed. We assume that this acknowledgment is sent in downlink packet.

Data and key information packet sizes are assumed to be 12 backoff periods long, and therefore we assume that the probability to access the medium τ_0 as well as probabilities of transmission without the collision γ, and that the 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 and second CCA denoted with α and 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 the medium access probability after each packet transmission. We also assume that the medium access control layer together with the application layer will repeat transmission until thepacket 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. 16.7 and 16.8*a*, we 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:

$$\begin{aligned} x_{0,1,0} &= \tau_0(1-P_d)\alpha = \tau_0 C_1 \\ x_{1,2,0} &= \tau_0(1-P_d)(1-\alpha\beta) = \tau_0 C_2 \\ x_{0,0,0} &= \tau_0[(1-P_d)\alpha\beta + P_d] = \tau_0 C_3 \\ C_4 &= \frac{1-C_2^{m+1}}{1-C_2} \end{aligned} \tag{16.1}$$

Using values C_i we obtain the sum of probabilities for one transmission subchain as:

$$\begin{aligned} s_t = \tau_0 C_4 \left[C_3(\overline{D_d} - 2) + C_1 + \frac{P_d(\overline{D_d} - 1)}{2} \right] \\ + \tau_0 \left[\sum_{i=0}^{m} \frac{C_2^i (W_i + 1)}{2} + C_2^{m+1} \right] \end{aligned} \tag{16.2}$$

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

In order to model the 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 $s_{s1} = \tau_0\gamma\delta/((1-P_{\text{sleep}})$. However, if the node wakes up and finds its buffer empty, it will start the new sleep. We will denote the probability of finding an empty buffer after sleep as Q_c and derive it later. The sum of probabilities of being in consecutive sleep then becomes $s_s = \tau_0\gamma\delta/[(1-P_{\text{sleep}})(1-Q_c)]$. 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

$$3(s_b + 2s_t) + 2s_t + n_k(s_s + s_t + s_b + s_c) = 1 \tag{16.3}$$

However, the total access probability by the node is equal to the sum of access probabilities in each transaction, that is,

$$\tau = (8 + n_k)\tau_0 \tag{16.4}$$

16.5.1 Analysis of Node's Packet Queue

In order to find probability Q_c we need to consider the node's MAC layer as the $M/G/1/K$ queuing model with vacations and setup time. We assume that when

the node wakes up it will transmit only one packet and go to sleep again, which is known as 1-limited scheduling [11]. A detailed model for the more general sleep policy with Bernoulli scheduling of activity period is derived in [9]. In Bernoulli scheduling, after one packet transmission, the node decides to transmit another packet with probability P_{ber} and goes to sleep with probability $1-P_{\text{ber}}$. We can apply this approach to our model using the restriction that $P_{\text{ber}}=0$. In the discussion that follows, packets are arriving to each node following the Poisson process with the rate λ. All nodes have buffers of finite capacity, L packets for an ordinary sensor node and L_{bri} packets for the two bridge/coordinators.

Consider the probability generating function (PGF) for one geometrically distributed sleep period (with parameter P_{sleep}) as

$$V(z)=\sum_{k=1}^{\infty}(1-P_{\text{sleep}})P_{\text{sleep}}^{k-1}z^{k}=\frac{(1-P_{\text{sleep}})z}{1-zP_{\text{sleep}}} \tag{16.5}$$

and the mean duration of the vacation is $\overline{V}=V'(1)=1/(1-P_{\text{sleep}})$. We also note [11] that the PGF for the number of packet arrivals to the sensor buffer during the sleep time is equal to

$$F(z)=V^{*}(\lambda-z\lambda) \tag{16.6}$$

where $V^{*}(\ )$ denotes the Laplace–Stieltjes transform (LST) of the sleep time, which (since sleep time is a 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 nonempty buffer) has to synchronize with the next beacon; the synchronization time is uniformly distributed between 0 and BI-1 backoff periods (where BI is beacon interval), and its PGF is

$$S_1(z)=\frac{1-z^{\text{BI}}}{\text{BI}(1-z)} \tag{16.7}$$

When the awakened node finds the next beacon, then it has to wait for collision separation time before it starts its backoff procedure. We adopt that this time is uniformly distributed between 0 and 7 backoff periods and its PGF has the value

$$S_2(z)=\frac{1-z^{\text{BI}/2}}{8(1-z)} \tag{16.8}$$

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

$$S_t(z)=S_1(z)S_2(z) \tag{16.9}$$

Its LST 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 $dt_t(x)$.

Let us now analyze the operation of the 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

$$\begin{aligned} D(z) &= \sum_{k=0}^{\infty} s_k z^k = \int_0^{\infty} e^{-x\lambda(1-z)} d(x) = S_t^*(\lambda - z\lambda) \\ A(z) &= \sum_{k=0}^{\infty} a_k z^k = \int_0^{\infty} e^{-x\lambda(1-z)}\, dt_t(x) = T_t^*(\lambda - z\lambda) \end{aligned} \tag{16.10}$$

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

$$s_k = \frac{1}{k!} \frac{d^k S(z)}{dz^k}\bigg|_{z=0} \quad a_k = \frac{1}{k!} \frac{d^k A(z)}{dz^k}\bigg|_{z=0} \quad f_k = \frac{1}{k!} \frac{d^k F(z)}{dz^k}\bigg|_{z=0}$$

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

$$\begin{aligned} q_0 &= (q_0 + \pi_0) f_0 \\ q_k &= (q_0 + \pi_0) f_k + \sum_{j=1}^{k} \pi_j f_{k-j} \quad \text{for } 1 \le k \le L-1 \\ q_L &= (q_0 + \pi_0) \sum_{k=L}^{\infty} f_k + \sum_{j=1}^{L-1} \pi_j \sum_{k=L-j}^{\infty} f_k \\ \pi_k &= \sum_{j=1}^{k+1} q_j \sum_{l=0}^{k-j+1} (s_l + a_{k-j+1-l}) \quad \text{for } 0 \le k \le L-2 \\ \pi_{L-1} &= \sum_{j=1}^{L} q_j \sum_{k=L-j}^{\infty} \sum_{l=0}^{k} (s_l + a_{k-l}) \\ 1 &= \sum_{k=0}^{L} q_k + \sum_{k=0}^{L-1} \pi_k \end{aligned} \tag{16.11}$$

The probability distribution of the device queue length at the time of packet departure $\pi_i, i = 0 \cdots L-1$ and return from the sleep $q_i, i = 0 \cdots L$ can be found by solving the system of linear equations (16.11). 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:

$$Q_c = \frac{q_0}{\sum_{i=0}^{L} q_i} \tag{16.12}$$

The probability distribution for the total inactive time of the node has a geometric distribution with the parameter Q_c, applied at the moments when the node returns fom sleep. The corresponding moment-generating function is

$$I^*(s) = \sum_{k=1}^{\infty} (1 - Q_c) Q_c^{k-1} V^*(s)^k = \frac{(1 - Q_c) V^*(s)}{1 - V^*(s) Q_c} \tag{16.13}$$

and the mean value is $\overline{I} = 1/[(1 - Q_c)(1 - P_{\text{sleep}})]$.

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

$$R = \frac{n_k \gamma \delta \tau_0}{t_{\text{boff}}} \tag{16.14}$$

where $t_{\text{boff}} = 0.32$ ms corresponds to the duration of one backoff period. The value R has to be set by the sensing application, for example, $R = 10$. Satisfying Eq. (16.14) will result in minimal energy consumption. However, we have to note that key exchange overhead will result in an overhead packet rate of $8\tau_0 \delta \gamma / t_{\text{boff}}$ packets per second.

16.5.2 Success Probabilities

As we mentioned earlier, we denoted the probabilities that the medium is idle on first and second CCA with α and β, respectively, and the probability that the transmission is successful with γ. Note that the first CCA may fail because a packet transmission from another node is in progress; this particular backoff period may be at any position with respect to that packet. The second CCA, however, will fail only if some other node has *just* started its transmission—that is, the backoff period in which the second CCA is undertaken must be the *first* backoff period of that packet. Note that the first medium access by any node will happen within the first 16 backoff periods of the superframe.

Let the clusters contain n_{bot}, n_{mid}, and n_{top} ordinary sensor nodes, respectively, with the packet arrival rate of λ per node. (References to specific clusters will use the subscripts bot, mid, and top, respectively.) The top cluster coordinator acts as the network sink.

16.5.2.1 Bottom Cluster. We apply the model from Section 16.5 and use expression (16.4) for τ_{bot}. Since τ_{bot} is very small and the number of nodes is large, we may estimate the per-cluster arrival rate of medium access events as

$$\lambda_{c,\text{bot}} = \frac{1}{16}(n_{\text{bot}} - 1)\tau_{\text{bot}}^{(1)} SD \tag{16.15}$$

The probability that the medium is not busy at the first CCA may, then, be approximated with

$$\alpha_{\text{bot}} = \frac{1}{16}\sum_{i=0}^{15} e^{-i\lambda_{c,\text{bot}}} \tag{16.16}$$

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 $n_{\text{bot}}-1$ nodes has started a transmission in that backoff period,

$$\beta_{\text{bot}} = e^{-\lambda_{c,\text{bot}}} \tag{16.17}$$

By the same token, the overall probability of success of a transmission attempt is

$$\gamma_{\text{bot}} = (\beta_{\text{bot}})^{\overline{D_d}} \tag{16.18}$$

16.5.2.2 Middle Cluster. In the middle cluster, besides ordinary nodes, we must account for the presence of the bridge, that is, the coordinator from the bottom cluster. For an ordinary node, we apply the model from Section 16.5 to the environment of middle cluster and use expression (16.4) for τ_{mid}.

The access probability for the bridge coming from the bottom cluster can be modeled as

$$\tau_{\text{bri,mid}} = \frac{1}{16} n_{\text{bot}} \tau_{\text{bot}} SD \tag{16.19}$$

The success probability for bridge transmissions depends on all the nodes in the middle cluster, that is,

$$\gamma_{\text{bri,mid}} = (1 - \tau_{\text{mid}})^{\overline{D_{d^n}}_{\text{mid}}} \tag{16.20}$$

The medium access event rate for a middle cluster node must also account for both the ordinary nodes and the bridge, hence:

$$\lambda_{c,\text{mid}} = \frac{1}{16}(n_{\text{mid}} - 1)\tau_{\text{mid}} SD + \tau_{\text{bri,mid}} \tag{16.21}$$

Parameters α, β, and γ can, then, be calculated in a similar way as their bottom cluster counterparts, that is,

$$\alpha_{\text{mid}} = \frac{1}{16}\sum_{i=0}^{15} e^{-i\lambda_{c,\text{mid}}} \tag{16.22}$$

$$\beta_{\text{mid}} = e^{-\lambda_{c,\text{mid}}} \tag{16.23}$$

$$\gamma_{\text{mid}} = e^{-\lambda_{c,\text{mid}}\overline{D_d}} \tag{16.24}$$

16.5.3 Sink Cluster

Success probabilities α_{top}, β_{top}, and γ_{top} for the top cluster can be found starting from

$$\tau^{(1)}_{\text{bri,top}} = \frac{1}{16}(n_{\text{bot}}\tau_{\text{bot}} + n_{\text{mid}}\tau_{\text{mid}})SD \tag{16.25}$$

16.6 MODEL OF ENERGY CONSUMPTION

While the activity management achieves the extension of the lifetime separately for each cluster, individual cluster lifetimes may differ. If this is the case, the network lifetime is determined by the shortest cluster lifetime; it is maximized if all clusters die at approximately the same time. In order to accomplish that, we have looked into the possibility of modifying cluster parameters so as to equalize their respective lifetimes.

The algorithm to calculate node population considers one cluster at a time in an iterative fashion, starting with the cluster that is farthest away from the sink.

As mentioned above, we assume that all transmissions are acknowledged; if the acknowledgment (ACK) packet is not received within the time prescribed by the standard [1], the transmission will be repeated. Let the PGF of the time interval between the data and subsequent ACK packet be $t_{\text{ack}}(z) = z^2$; actually its value is between *aTurnaroundTime* and *aTurnaroundTime* + *aUnitBackoffPeriod* [1], but we round the exponent to the next higher integer for simplicity.

According to the standard [1], transmission has to be preceded with the backoff procedure and two CCAs during which the radio part is in the receiving mode. Only after successful CCAs, radio module switches to the transmitting mode. The standard allows m (default value is $m = 5$) backoff attempts during which backoff windows take values of $W_0 = 7$, $W_1 = 15$, $W_2 = W_3 = W_4 = 31$ (if the battery saving mode is not turned on). However, under the sleep management regime, all transmissions will complete in one or two backoff attempts,

and battery saving mode is not important. The PGF for the duration of jth backoff time prior to transmission is equal to

$$B_j(z) = \sum_{k=0}^{W_j-1} \frac{1}{W_j} z^k = \frac{z^{W_i} - 1}{W_j(z-1)} \tag{16.26}$$

In order to find energy consumption during the jth backoff attempt, we need to switch to the LST by substitution $z = e^{-s\omega_r}$ (because PGFs don't allow noninteger exponents) and obtain LST:

$$E^*_{B_j}(s) = \frac{e^{-s\omega_r W_i} - 1}{W_j(e^{-s\omega_r} - 1)} \tag{16.27}$$

Let the PGF of the data packet length be $G_p(z) = z^k$, and let $G_a(z) = z$ stand for the PGF of the ACK packet duration. Then the PGF for the total transmission time of the data packet will be denoted with $D_d(z) = z^2 G_p(z) t_{\text{ack}}(z) G_a(z)$; its mean value is $\overline{D_d} = 2 + G'_p(1) + t'_{\text{ack}}(1) + G'_a(1)$. The LST for the energy consumption during pure packet transmission time is $e^{-sk\omega_t}$. The LST for energy consumption during two CCAs is equal to $e^{-s2\omega_r}$. The LST for energy consumption during waiting for and receiving the acknowledgment is $e^{-s3\omega_r}$. The same value has the LST for energy consumption during reception of the beacon frame, which is three backoff periods long.

Then, the PGF for the time needed for one complete transmission attempt including backoffs becomes

$$\mathscr{A}(z) = \frac{\sum_{i=0}^{m} \left[\prod_{j=0}^{i} B_j(z)\right] (1-\alpha\beta)^i z^{2(i+1)} [\alpha\beta G_p(z) t_{\text{ack}}(z) G_a(z)]}{\sum_{i=0}^{m} (1-\alpha\beta)^i \alpha\beta} \tag{16.28}$$

The LST for energy consumption for one transmission attempt then becomes

$$\mathscr{E}^*_{\mathscr{A}}(s) = \frac{\sum_{i=0}^{m} \left[\prod_{j=0}^{i} E^*_{B_j}(z)\right] (1-\alpha\beta)^i e^{-s2\omega_r(i+1)} \left(\alpha\beta e^{-sk\omega_t} e^{-s3\omega_r}\right)}{\sum_{i=0}^{m} (1-\alpha\beta)^i \alpha\beta} \tag{16.29}$$

By taking packet collisions into account, the probability distribution of the packet service time follows the geometric distribution, and its PGF becomes

$$T(z) = \sum_{k=0}^{\infty} [\mathscr{A}(z)(1-\gamma)]^k \mathscr{A}(z)\gamma = \frac{\gamma \mathscr{A}(z)}{1 - \mathscr{A}(z) + \gamma \mathscr{A}(z)} \tag{16.30}$$

In this case, mean packet service time can simply be written as $\overline{T} = T'(1) = \mathscr{A}'(1)/\gamma$.

The LST for the energy spent on a packet service time is then equal to

$$E_T^*(s) = \frac{\gamma \mathscr{E}_A^*(s)}{1 - \mathscr{E}_A^*(s) + \gamma \mathscr{E}_A^*(s)} \tag{16.31}$$

16.6.1 Bottom Cluster

The PGF of the time needed to conduct one transmission attempt is then obtained by substituting α_{bot}, $\beta_{\text{bot}}^{(2)}$, and γ_{bot} in Eq. (16.30). The LST for the energy spent in packet service is obtained by substituting those values in Eq. (16.31). The average value of energy consumed for packet service is obtained as

$$\overline{E_{T,\text{bot}}} = -\frac{d}{ds} E_{T,\text{bot}}^*(s)|_{s=0}$$

The average battery energy consumption per backoff period can be found as

$$u_{\text{bot}} = \frac{\overline{S_1}\omega_r + \overline{S_2}\omega_r + 3\omega_r + \overline{I_{\text{bot}}}\omega_s + \overline{E_{T,\text{bot}}}(1 + 8/n_k)}{\overline{S_1} + \overline{S_2} + 3 + \overline{I_{\text{bot}}} + \overline{T_{\text{bot}}}(1 + 8/n_k)} \tag{16.32}$$

Given the battery budget of b joules, the average number of transmission/sleep cycles in the bottom cluster can be found as

$$n_{c,\text{bot}} = \left\lceil \frac{b}{\overline{S_1}\omega_r + 3\omega_r + \overline{S_2}\omega_r + \overline{E_{T,bot}}(1 + \frac{8}{n_k}) + \overline{I_{\text{bot}}}\omega_s} \right\rceil \tag{16.33}$$

Given the law of large numbers [12], the PGF for total lifetime of the node in bottom cluster becomes

$$L_{\text{bot}}(z) = [S_1(z)S_2(z)T_{\text{bot}}(z)I_{\text{bot}}(z)]^{n_{c,\text{bot}}} \tag{16.34}$$

By differentiating the respective PGFs, we can obtain the standard deviation of the node lifetime as well as the coefficient of skewness, μ, which measures the deviation of a distribution from symmetry [13].

16.6.2 Middle Cluster

By using the appropriate values of α_{mid}, β_{mid}, and γ_{mid} the PGFs for a single transmission attempt and for the overall packet transmission time can be calculated as $\mathscr{A}_{\text{mid}}(z)$ and $T_{\text{mid}}(z)$, respectively. Both PGFs depend on the number of nodes n_{mid} as the parameter. Average battery energy consumption

per backoff period is calculated as

$$u_{\text{mid}} = \frac{\overline{S_1}\omega_r + \overline{S_2}\omega_r + 3\omega_r + \overline{E_{T,\text{mid}}}(1 + 8/n_k) + \overline{I_{\text{mid}}}\omega_s}{\overline{S_1} + \overline{S_2} + 3 + \overline{T_{\text{mid}}}(1 + 8/n_k) + \overline{I_{\text{mid}}}} \tag{16.35}$$

Now, if the lifetime of the middle cluster is to be the same as that of the bottom cluster, the average energy that the node consumes per backoff period in both clusters should have equal values:

$$u_{\text{mid}} = u_{\text{bot}} \tag{16.36}$$

from which we can obtain the initial population of the middle cluster n_{mid}. This equation is necessary only if we want to choose n_{mid} in order to equalize the lifetimes of the bottom and middle clusters. Otherwise, n_{mid} can be chosen using some other policy.

Given the battery budget of b backoff periods, the average number of transmission/sleep cycles in bottom cluster can be found as

$$n_{c,\text{mid}} = \left\lceil \frac{b}{\overline{S_1}\omega_r + 3\omega_r + \overline{S_2}\omega_r + \overline{E_{T,\text{mid}}}(1 + 8/n_k) + \overline{I_{\text{mid}}}\omega_s} \right\rceil \tag{16.37}$$

The PGF for total lifetime of the node in the bottom cluster becomes

$$L_{\text{mid}}(z) = [S_1(z)S_2(z)T_{\text{mid}}(z)I_{\text{mid}}(z)]^{n_{c,\text{mid}}} \tag{16.38}$$

The procedure is then repeated for the top cluster, starting from

$$\tau_{\text{bri,top}}^{(1)} = \frac{1}{16}(n_{\text{bot}}\tau_{\text{bot}} + n_{\text{mid}}\tau_{\text{mid}})SD \tag{16.39}$$

This algorithm is scalable since the overall model can be broken in individual cluster models with input from all clusters at lower level. The condition for the correctness for this approximation is that all clusters are not operating in the saturation condition.

16.7 PERFORMANCE EVALUATION

In this section we present numerical results obtained by solving the system of equations that represent the analytical model of the node's MAC with sleep and key exchange, node's queue behavior, and medium behavior. As a solution we

obtain system parameters τ_0, τ, P_{sleep}, α, β, γ, and Q_c. We have varied the key exchange threshold between 20 and 100 packets, while the requested event sensing reliability per cluster was kept at $R = 10$ packets per second. Source cluster size was varied between 20 and 100 nodes. We assumed that each node is powered with two AA batteries that supply a voltage between 2.1 and 3.6 V and 500 mA-h as required by tmote_sky [5] operating conditions with total energy $b = 10,260$ J.

We have also assumed that the network operates in the ISM band at 2.45 GHz, with a raw data rate of 250 kbps. The 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. Other parameters from the medium access control layer were kept at default values. In Fig. 16.9 we present access probabilities in bottom, middle, and top cluster, respectively, and in Fig. 16.10 we present transmission success probabilities.

Figure 16.11 shows a number of nodes in middle and sink cluster with population in source cluster indicated as n_{bot} and period of key exchange indicated as n_k. We notice significant increase of populations as we move

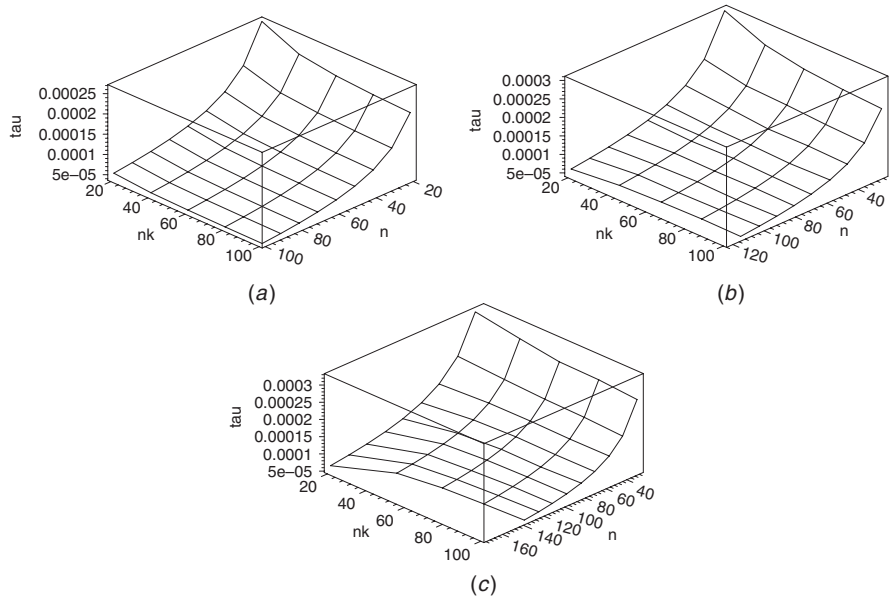

FIGURE 16.9 Access probabilites for node: (*a*) in source cluster, (*b*) in middle cluster, and (*c*) in sink cluster.

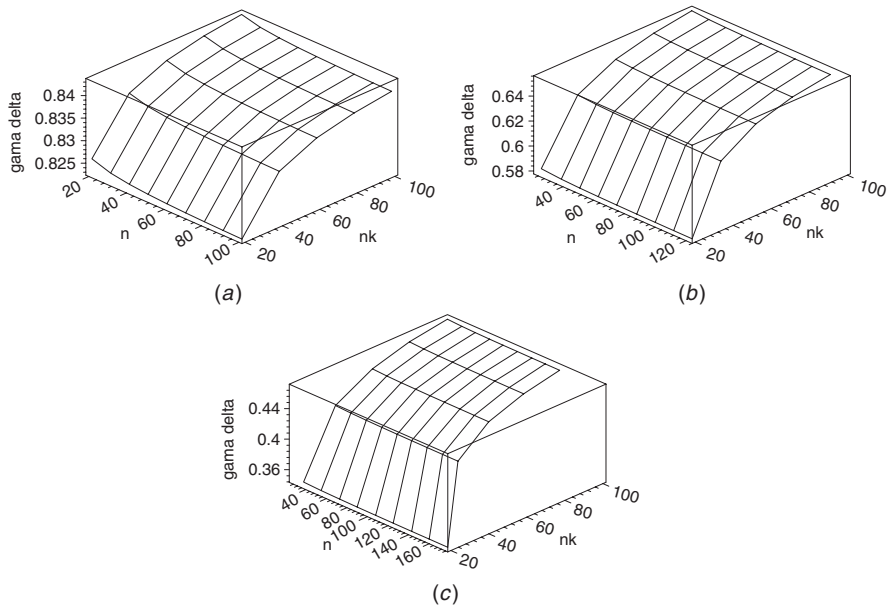

FIGURE 16.10 Medium behavior. Success probability for node: (*a*) in source cluster, (*b*) in middle cluster, and (*c*) in sink cluster.

toward the sink. The task of increase of population in the cluster is mainly to compensate for the drop of transmission success probability and CCA success probabilities, which is caused by the bridges' data and key exchange traffic. Figure 16.12 shows equalized lifetimes of clusters versus recalculated populations and key exchange period.

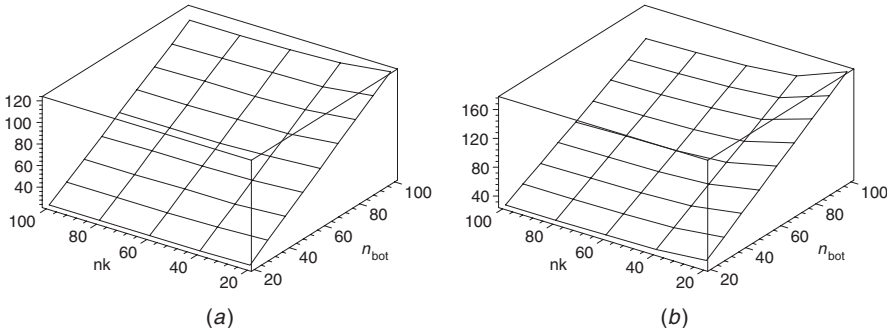

FIGURE 16.11 Node populations: (*a*) in middle cluster; (*b*) in sink cluster.

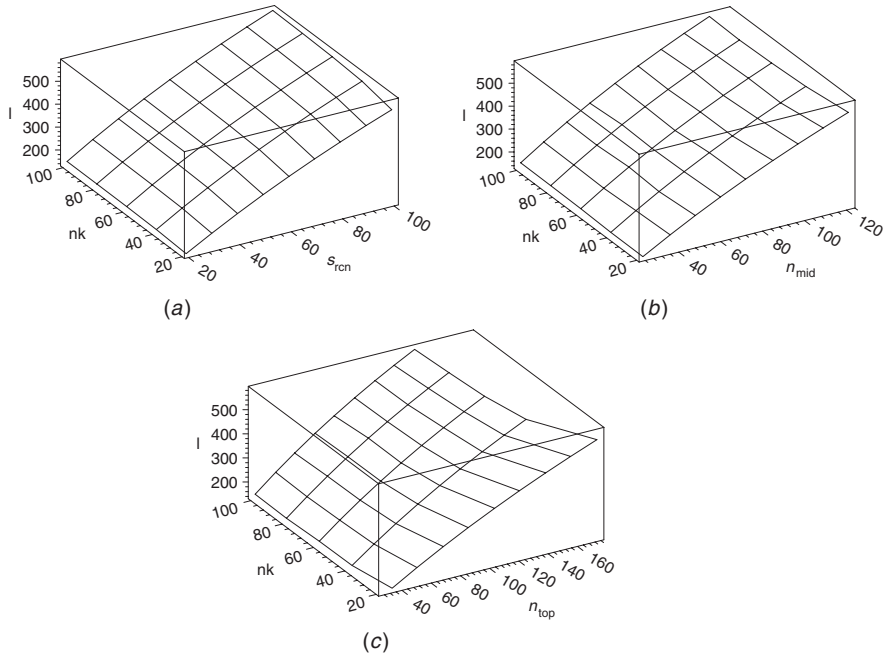

FIGURE 16.12 Cluster lifetimes: (*a*) in source cluster, (*b*) in middle cluster, and (*c*) in sink cluster.

16.8 CONCLUSION AND FUTURE WORK

We have developed an analytical model of the key exchange integrated into the sensing function of the beacon-enabled 802.15.4 cluster. Our results show an 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. In our future work we plan to model more complex key exchange algorithms.

REFERENCES

1. IEEE 802.15.4-2006 "Wireless MAC and PHY specifications for low rate WPAN," revision of IEEE 802.15.4-2003, IEEE, New York, 2006.
2. ZigBee Alliance, ZigBee Specification, Document 053474r06, Version 1.0, ZigBee Alliance, San Ramon, CA, 2004.
3. J. Mišić, C. J. Fung, and V. B. Mišić, "On node population in a multi-level 802.15.4 sensor network," paper presented at Globecom 2006, San Francico, CA, 2006.
4. J. Mišić, S. Shafi, and V. B. Mišić, "Cross-layer activity management in a 802.15.4 sensor network," *IEEE Commun. Mag.* 44, 131–136 (2006).

5. "tmote sky lowpower wireless sensor module," tmote datasheet 802.15.4, Moteiv, San Francisco, CA, available: www.moteiv.com, 2006.
6. M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh, "Exploiting heterogeneity in sensor networks," paper presented at INFOCOM05, Vol. 2, Miami, FL, 2005, pp. 878–890.
7. I. Stojmenović (Ed.), *Handbook of Sensor Networks: Algorithms and Architectures*, Wiley, Hoboken, NJ, 2005.
8. Y. Sankarasubramaniam, Ö. B. Akan, and I. F. Akyildiz, "ESRT: Event-to-sink reliable transport in wireless sensor networks," paper presented at the 4th ACM MobiHoc, Annapolis, MD, 2003, pp. 177–188.
9. J. Mišić, S. Shafi, and V. B. Mišić, "Maintaining reliability through activity management in an 802.15.4 sensor cluster," *IEEE Trans. Vehic. Technol.* 55, 779–788 (2006).
10. A. Menezes, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*, CRC Press, Boca Raton, FL, 1997.
11. H. Takagi, *Queueing Analysis*, Vol. 1: *Vacation and Priority Systems*, North-Holland, Amsterdam, The Netherlands, 1991.
12. G. R. Grimmett, and D. R. Stirzaker, *Probability and Random Processes*, 2nd ed., Oxford University Press, Oxford, 1992.
13. P. Z. Pebbles, Jr., Probability, Random Variables, and Random Signal Principles, McGraw-Hill, New York, 1993.

AU7

Author Query Form

Title: Emerging Wireless LANs, Wireless PANs, and Wireless MANs
Chapter ID: JWUS_ELWPWM_C16 / **Chapter Title:** Impact Of Reliable And Secure Sensing On Cluster Lifetime

Dear Author,
During the preparation of your manuscript for typesetting some questions have arisen. These are listed below. Please check your typeset proof carefully and mark any corrections in the margin of the proof or compile them as a separate list. This form should then be returned with your marked proof/list of corrections to John Wiley.

This form should then be returned with your marked proof/list of corrections to Chayanika, Project Manager, Macmillan India Ltd., Book Division–IPD, Midford Crescent, New No. 159/1 (Old No. 53/1), Richmond Road, Bangalore-560025, India, Tel: 91-80-41237312; +91-80-41237313, Fax: +91-80-41237310, E-mail: d.chayanika@macmillansolutions.com

Queries and/or remarks

AU1	Ok to put in brackets for sense from msp 747?
AU2	Ok as defined?
AU3	Ok as defined?
AU4	cite for F16.3 Ok here?
AU5	define
AU6	Fig. 16.5 meant?
AU7	For Proc., if published, give all facts of publication: full volume title, ed(s), publisher, city & date of publication.
AU8	Provide table title and col. heads
AU9	Title OK?