RESEARCH   **Open Access**

CrossMark

An analytical model for wireless mesh networks with collision-free TDMA and finite queues

Florian Kauer[*] [iD] and Volker Turau

Abstract

Wireless mesh networks are a promising technology for connecting sensors and actuators with high flexibility and low investment costs. In industrial applications, however, reliability is essential. Therefore, two time-slotted medium access methods, DSME and TSCH, were added to the IEEE 802.15.4 standard. They allow collision-free communication in multi-hop networks and provide channel hopping for mitigating external interferences. The slot schedule used in these networks is of high importance for the network performance. This paper supports the development of efficient schedules by providing an analytical model for the assessment of such schedules, focused on TSCH. A Markov chain model for the finite queue on every node is introduced that takes the slot distribution into account. The models of all nodes are interconnected to calculate network metrics such as packet delivery ratio, end-to-end delay, and throughput. An evaluation compares the model with a simulation of the Orchestra schedule. The model is applied to Orchestra as well as to two simple distributed scheduling algorithms to demonstrate the importance of traffic-awareness for achieving high throughput.

Keywords: Wireless mesh networks, Time division multiple access, Scheduling, IEEE 802.15.4, TSCH

1 Introduction

Wireless multi-hop networks are currently on the transition from a popular research topic to the application in real-world scenarios in the industry. Examples include oil refineries [1] and solar tower power plants [2]. However, for such applications, the reliability of state-of-the-art techniques is not sufficient. This can often be traced back to packet collisions due to badly coordinated access to the wireless channel [3]. Therefore, the IEEE has recently extended the IEEE 802.15.4 standard with two approaches for time-division multiple access (TDMA) in multi-hop networks, namely DSME and TSCH [4]. Especially in large networks under heavy load, these approaches are expected to perform much better than the commonly used CSMA/CA. While several implementations of these extensions already exist [5, 6], evaluations of the theoretical performance boundaries have not found much attention in the literature. This paper closes the gap by

providing a framework for analytical evaluation of such networks.

The performance evaluation of techniques for wireless networks is possible in multiple ways. Building real-world testbeds provides good insights into the performance of an actual application. However, it is very cost- and time-intensive. Therefore, event-based simulators are developed that try to replicate the reality as closely as possible, while allowing reproducible experiments without the need of specialized hardware. The disadvantage is a complex and extensive implementation and large execution times.

An alternative is the analysis of mathematical models. An analytical model in the sense of this paper is a system of equations that can be solved by numerical means. While such a model usually only describes a small subset of the behavior of a real system and is therefore only an approximation to the real behavior, the calculations are often faster than a simulation by magnitudes and influences from external sources are avoided. The results can also be verified and reproduced quite easily. Last but

*Correspondence: florian.kauer@tuhh.de
Hamburg University of Technology, Institute of Telematics, Am Schwarzenberg-Campus 3, 21073 Hamburg, Germany

not least, the comparison of a simulator and an analytical model helps to find inaccuracies and bugs in both approaches and gives new insights into the underlying principles.

The main contributions of the paper are as follows:

- A Markov chain model for a node's finite queue that considers the slot schedule.
- A multi-hop model for calculating packet delivery ratio, end-to-end delay, and throughput.
- An open-source implementation of the models [7].
- An extensive evaluation, including a comparison with an event-based simulation and the application to two traffic-aware schedules for TSCH.

After presenting the related work in the next section, the requirements of a model for TSCH are outlined in Section 3 followed by the system model in Section 4. We then take the angle of a single node and present and evaluate the queuing model in Section 5. These models are then concatenated in Section 6 to derive global network metrics, and a comparison with a COOJA simulation is presented. In Section 7, two algorithms for building slot schedules are introduced, evaluated, and compared to the Orchestra SBD schedule. Section 8 gives an outlook to possible extensions, and the paper is finally concluded in Section 9.

2 Related work

Analyzing the performance of wireless networks is a challenging research topic of high value for practical applications [8]. Many evaluations for IEEE 802.15.4 networks are based on simulations [9] or real-world testbeds [10]. Especially for the CSMA/CA technique of IEEE 802.15.4 analytical models exist for single-hop topologies [11] and multi-hop networks [3, 12]. In these publications, packet collisions, especially due to hidden node constellations, are identified as a major reason for bad performance of CSMA/CA.

Since their admission to the standard, TSCH and DSME were analyzed by simulations [13], testbeds [6], and analytical models. The latter includes work focused on special aspects such as network formation [14] or transmission in the contention access period (CAP) [15]. The latter is very similar to the analysis of conventional CSMA/CA because it does not take guaranteed time slots into account. A simple formula for the throughput of DSME networks is given in [16]. Furthermore, much research exists that analyzes TDMA communication on a more general level [17, 18].

A major aspect of network analysis is the influence of the queue filling levels. The fundamentals of queuing theory are well-established [19]. Especially, the M/D/1/K model is of particular interest for TDMA networks, describing Poisson-distributed arrival, deterministic service time, a single transmitter, and a queue of length K. In [20], closed-form expressions are given for the blocking probability and other properties. Many variants were analyzed including service times following a general distribution [21]. In [22], a generalized queuing model for TDMA is developed that is also applicable for traffic that does not follow a Poisson distribution.

In this paper, a queuing model is presented that respects the specific structure of dedicated schedules, in particular, but not limited to, IEEE 802.15.4 TSCH networks. A popular schedule implemented in Contiki [23] is Orchestra [6]. It does not require any management traffic for building up the schedule. Part of the ongoing IETF standardization of 6TiSCH, the link between IPv6 networks and TSCH, is the development of scheduling functions such as SF0 [24] and SF1 [25].

Many other scheduling techniques were proposed that are applicable to TSCH networks or to one of its forerunners, the WirelessHART standard [26]. A centralized schedule for the latter is presented in [27]. Another centralized schedule, this time for TSCH, is the Traffic Aware Scheduling Algorithm (TASA) [28]. Traffic awareness is an important property for improving throughput as we will also see in the evaluation of this paper. However, centralized scheduling algorithms come with a high overhead for building, distributing, and maintaining the schedule, especially in large networks. Therefore, many decentralized approaches were suggested, such as DeTAS [29] and DIS_TSCH [30].

Another promising distributed algorithm to generate a conflict-free schedule is Wave, presented in [31] and extensively analyzed in [32]. DeBraS [33] explicitly targets dense networks where colliding slots are a major problem. All mentioned algorithms implicitly minimize energy consumption by reducing the number of slots assigned to every node. In contrast, [34] explicitly targets minimum energy consumption by formulating an energy efficiency maximization problem.

3 Requirements

The goal of the paper is to provide a tool for estimating the achievable performance of a network using the IEEE 802.15.4 TSCH data link layer with given multi-hop topology and associated schedule. In this section, an analysis of the potential sources for packet loss is presented. These are compared to the features provided by modern TDMA medium access protocols such as IEEE 802.15.4 TSCH and DSME. The most relevant aspects for a model are finally summarized.

The main reasons for packet loss in wireless multi-hop networks can be coarsely categorized as follows.

- Collisions of transmissions of the same network.

- External interferences, for example between IEEE 802.15.4 and IEEE 802.11 transceivers or with another unsynchronized IEEE 802.15.4 network.
- High path loss, for example due to a large distance or fading.
- Queue drops if the rate of packets that are generated or have to be forwarded is higher than the rate at which packets can be transmitted. This can be permanent or during a burst.

To mitigate these losses, the introduced TDMA data link layers TSCH and DSME provide the following features that allow for reliable multi-hop communication.

- A slot structure with predefined timing to arrange the transmissions in the time domain to avoid collisions within the same network.
- Time synchronization to ensure an aligned timing throughout the multi-hop network.
- Channel adaption (only DSME) and channel hopping (both) to arrange transmission in the frequency domain to increase the number of transmissions per time as well as to mitigate external interferences.
- Procedures for setting up time and frequency schedules in a distributed fashion to ensure conflict-free transmissions. This is only integrated in DSME, but the ongoing IETF standardization of 6TiSCH provides related features for TSCH.

To calculate the performance in steady-state, it can be assumed that all nodes are already associated to the network, are properly synchronized to the global notion of time, and have negotiated a conflict-free and valid multi-hop schedule. In such a schedule, a combination of a time slot and a frequency channel is either free or assigned to exactly one transmitter and one receiver in every neighborhood. By this, packet collisions can be avoided, even in hidden node constellations. In DSME networks, this constraint is inherent, given that no failures happen during the negotiation [35]. For TSCH networks, such a fixed assignment is not necessarily required, but sharing will obviously lead to collisions, even if they might not be as severe as with CSMA/CA. Furthermore, the schedule should avoid using links with high path loss or highly fluctuating channel conditions to avoid occasional packet loss on the physical layer by means of an appropriate neighborhood management [36].

In large industrial plants, the frequencies of the used radio components are usually coordinated and monitored to avoid cross-interferences and as a security measure [37]. However, external interferences can often not be avoided completely, so TSCH and DSME can dynamically use multiple frequency channels. Apart from mitigating external interferences, this can also be used to increase the throughput by assigning the same time slot to multiple transceiver pairs on different frequency channel and is therefore also modeled in this paper. It can either be implemented as channel adaption where a fixed frequency channel is assigned at a given time or channel hopping where the channel to be used is iterated over a given sequence. Yet, in this paper, external interferences are not considered, so channel hopping with non-overlapping hopping sequences is equivalent to a fixed assignment as for channel adaption.

It is therefore concluded that in properly constructed TSCH and DSME networks without a lot of external disturbances, the occurrence of queue drops is the main factor that determines the maximum achievable network performance. Yet, in real-world networks, other losses can still occur, so Sect. 8.2 gives an outlook about how to consider these in the model.

As noted in the previous section, the M/D/1/K model is often used to model queues in the context of TDMA networks. However, it has two major weaknesses in the given scenario: Even if the traffic generation itself is modeled as Poisson distributed, this distribution does not necessarily hold for forwarding nodes, so the M/D/1/K model is not suitable for multi-hop networks. Furthermore, the service times are not necessarily deterministic but can diverge significantly, due to two effects: First, the schedule itself might be irregular, for example if multiple subsequent transmission slots are followed by a large idle phase. Secondly, even if the schedule is regular, the service time of packet that arrives at an empty queue is not constant, but depends on the time left until the next transmission slot. This effect is especially relevant in scenarios with low traffic. A proper model must take these effects into account.

4 System model
In the presented system model as illustrated in Fig. 1, the queue of every node is modeled as an instance of a Markov chain and they are linked to form a model of the full network. Packets are either forwarded as received or generated at the node. In both cases, the packets are pushed to a queue of fixed length or dropped if the queue is full. If there is at least one packet in the queue at the beginning of a transmission slot, a transmission attempt takes place.

The calculations are based on fixed multi-hop schedules. A schedule can be calculated offline with algorithms such as the ones presented in Section 7. Alternatively, a schedule can be extracted from a real-world deployment or a simulator as presented in Section 6.4. In the following, the notation for such a schedule as used in the model is introduced.

For the presented model, a network of N nodes $v_0, \ldots, v_{N-1}$ with a fixed schedule is considered. All slots have an equal time duration T_s. For TSCH, it is usually

Fig. 1 Illustration of the proposed model annotated with the notation introduced in the following sections
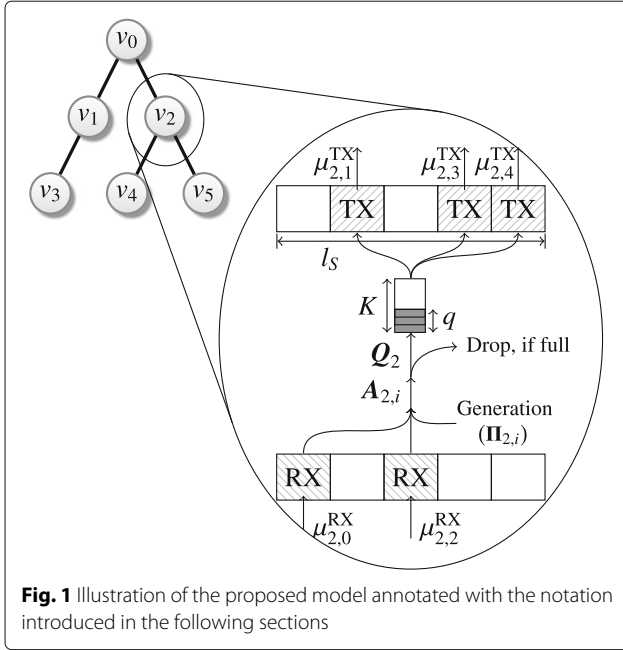
10 ms. A slotframe consists of l_S slots and is repeated with an interval of $T_s \cdot l_S$. The schedule for node v_n is given as a tuple of transmission slots $\mathcal{T}_n = (t_{n,j})_{j=0,\dots,|\mathcal{T}_n|-1}$, sorted in ascending order, where $t_{n,j} \in \mathcal{T}_n$ if and only if the slot with the zero-based index $t_{n,j}$, that is $0 \le t_{n,j} < l_S$, is assigned to n for conflict-free transmission. Correspondingly, $\mathcal{R}_n = (r_{n,j})_{j=0,\dots,|\mathcal{R}_n|-1}$ describes the reception slots. A slot is never in both $\mathcal{T}_n$ and $\mathcal{R}_n$, because a node can not transmit and receive at the same time. Furthermore, $\mathcal{K}_n(i)$ gives the respective reception node if $i \in \mathcal{T}_n$ or the respective transmission node if $i \in \mathcal{R}_n$. Figure 2 presents a simple example for these definitions.

For modeling frequency diversity, $\mathcal{C}_{v_n}(i)$ is introduced that specifies the channel to use in slot i by node v_n. The channel is an element of the set of channels $\mathcal{C}$. For IEEE 802.15.4 in the 2.4 GHz band, it consists of the numbers 11 to 26.

To check that a schedule is conflict-free and valid, the set

$$\mathbf{L}_i = \{(n, \mathcal{K}_n(i)) \mid i \in \mathcal{T}_n, \forall n\} \quad (1)$$
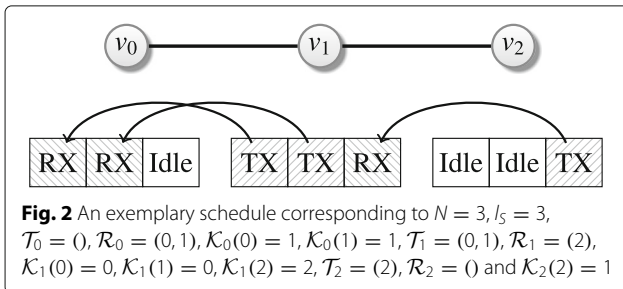
of all links that are active during slot i has to be considered for all slots. Each node can only be transmitter *or* receiver during a slot

$$(i \in \mathcal{T}_n \Rightarrow i \notin \mathcal{R}_n) \land (i \in \mathcal{R}_n \Rightarrow i \notin \mathcal{T}_n). \quad (2)$$

Furthermore, the links have to be unique and consistent, $\forall i \in \mathcal{T}_n$ it must hold

$$i \in \mathcal{R}_{\mathcal{K}_n(i)} \land (\mathcal{K}_n(i) = k \Leftrightarrow \mathcal{K}_k(i) = n). \quad (3)$$

However, this alone is insufficient. Other potential conflicts are shown in Fig. 3. The most obvious constellation is $\mathcal{R}_S$: The reception at w_1 can be disturbed by a transmission from v_2 to w_2 if v_2 is in the neighborhood of w_1. If acknowledgments are used, conflicts between acknowledgments and the data packets have to be taken into account, too, represented by the other constellations. For a link $(v_1, w_1) \in \mathbf{L}_i$, the potentially disturbing links are

$$\begin{aligned}
\mathcal{D}_{i,(v_2,w_2)} = \\
\{(v_2, w_2) \in \mathbf{L}_i \mid D(v_1, v_2) \land v_1 \ne v_2\} \\
\cup \{(v_2, w_2) \in \mathbf{L}_i \mid D(w_1, v_2) \land v_1 \ne v_2\} \\
\cup \{(v_2, w_2) \in \mathbf{L}_i \mid D(v_1, w_2) \land v_1 \ne v_2\} \\
\cup \{(v_2, w_2) \in \mathbf{L}_i \mid D(w_1, w_2) \land v_1 \ne v_2\}.
\end{aligned} \quad (4)$$

Here, $D(v, w)$ denotes that a transmission of v might disturb a reception at w. One possibility to ensure a conflict-free schedule is to have
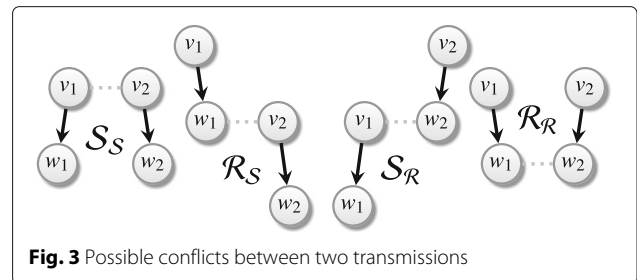
$$\mathcal{D}_{i,\cup} = \bigcup_{l, \in \mathbf{L}_i} \mathcal{D}_{i,l} \quad (5)$$

empty for all slots. Hence, no concurrent transmissions take place in the neighborhood. By exploiting frequency diversity, it is, however, sufficient to make sure that no concurrent transmissions take place on the same frequency channel at the same time, i.e., one has to make sure that for all slots, $0 \le i < l_S$ holds

$$\mathcal{C}_{v_1}(i) \ne \mathcal{C}_{v_2}(i), \forall(v_2, w_2) \in \mathcal{D}_{i,(v_1,w_1)} \quad (6)$$

5 Queue model

Before modeling multi-hop communication, this section considers the viewpoint of a single node v_n and its local behavior. In the multi-hop model, every node of the network will get its own instance of this model. The main focus is the finite queue, so after this section, we will be

Fig. 2 An exemplary schedule corresponding to $N = 3$, $l_S = 3$, $\mathcal{T}_0 = ()$, $\mathcal{R}_0 = (0, 1)$, $\mathcal{K}_0(0) = 1$, $\mathcal{K}_0(1) = 1$, $\mathcal{T}_1 = (0, 1)$, $\mathcal{R}_1 = (2)$, $\mathcal{K}_1(0) = 0$, $\mathcal{K}_1(1) = 0$, $\mathcal{K}_1(2) = 2$, $\mathcal{T}_2 = (2)$, $\mathcal{R}_2 = ()$ and $\mathcal{K}_2(2) = 1$

Fig. 3 Possible conflicts between two transmissions

able to calculate packet drops due to a full queue as well as the expected queuing delay. The inputs into this model are the probabilities for incoming traffic and the distribution of the reception slots $\mathcal{R}_n$ and transmission slots $\mathcal{T}_n$. Beyond the usage in a multi-hop model as introduced in the next section, the model in this section is also applicable in other setups, such as single-hop networks or for evaluating the performance of a single node as presented in Section 5.7.

The following policy is used to model the queue:

- The queue can hold at most K packets.
- The number of packets in the queue at the beginning of a slot is denoted as q.
- New packets can arrive at any time during a slot. At most, $K - q$ packets are accepted during a slot.
- A packet is removed from the queue at the end of a slot if and only if it is a transmission slot and the packet was already in the queue at the beginning of the slot, i.e. $q > 0$, modeling the transmission process itself.

The queue of a node is modeled as a discrete-time Markov chain with the states

$$\Sigma_n = \left\{ \sigma_{q,i}^n \,\middle|\, 0 \leq q \leq K, 0 \leq i < l_S \right\}. \tag{7}$$

In order to account for the irregular schedule and for modeling the service time more accurate, the state of the queue $\sigma_{q,i}^n$ does not only account for the number q of packets in the queue as for usual M/D/1/K models, but also the current position within the slot schedule i. As shown in Fig. 4, the model presented here is a super set of the M/D/1/K model for $l_S = 1$ and $\mathcal{T}_n = (0)$. Figure 5 depicts a more complex schedule, the corresponding states and the transitions as follows.

5.1 Traffic model

The number of arriving packets during a time slot i, that is the generated and received traffic, is described by the random variable $A_{n,i}$. For the purpose of the queuing model, it can follow an arbitrary distribution that can optionally depend on the current state. A popular option is to model the traffic as Poisson distributed with a mean packet rate

Fig. 4 A M/D /1/ K model in the syntax introduced in this paper with $K = 3$, $l_S = 1$ and $\mathcal{T}_n = (0)$
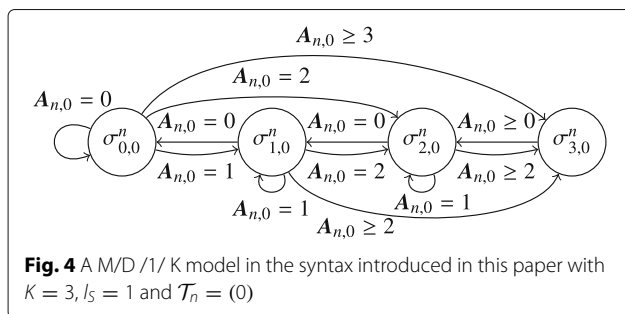
$\lambda_{n,i}$. For this, the random variable $\mathbf{\Pi}_{n,i}$ is introduced with the probability distribution

$$P(\mathbf{\Pi}_{n,i} = k) = \frac{\lambda_{n,i}^k}{k!} e^{-\lambda_{n,i}}. \tag{8}$$

The probability for *at least* k packets is given by

$$P(\mathbf{\Pi}_{n,i} \geq k) = 1 - \sum_{j=0}^{k-1} P(\mathbf{\Pi}_{n,i} = k). \tag{9}$$

Since only a single packet can be received per slot, we extend this model by combining it with a Bernoulli distribution with the probability $\beta_{n,i}$ that a single packet is arriving in addition to the Poisson traffic in slot i, so we get

$$P(\mathbf{A}_{n,i} = k) = \begin{cases} (1 - \beta_{n,i}) \cdot P(\mathbf{\Pi}_{n,i} = 0) & k = 0 \\ (1 - \beta_{n,i}) \cdot P(\mathbf{\Pi}_{n,i} = k) & \text{otherwise} \\ + \beta_{n,i} \cdot P(\mathbf{\Pi}_{n,i} = k - 1) \end{cases} \tag{10}$$

and

$$P(\mathbf{A}_{n,i} \geq k) = \begin{cases} 1 & k = 0 \\ (1 - \beta_{n,i}) \cdot P(\mathbf{\Pi}_{n,i} \geq k) & \text{otherwise.} \\ + \beta_{n,i} \cdot P(\mathbf{\Pi}_{n,i} \geq k - 1) \end{cases} \tag{11}$$

Since all l_S slots occur with the same probability, the expected total number of packets per slotframe is calculated as the sum of the expected number of the $A_{n,i}$ as derived in Appendix A as

$$\mathcal{A}_n = \sum_{i=0}^{l_S - 1} E[\mathbf{A}_{n,i}]$$

$$= \sum_{i=0}^{l_S - 1} (1 - \beta_{n,i}) \cdot \lambda_{n,i} + \beta_{n,i} \cdot (\lambda_{n,i} + 1). \tag{12}$$

5.2 Queuing probability

The random variable $\mathbf{Q}_n$ is the number of accepted packets that are inserted into the queue per slot. Since at most $K - q$ packets can be inserted into the queue, its probability distribution, i.e., the probabilities of queuing k packets, is calculated as

$$P(\mathbf{Q}_n = k \mid \sigma_{q,i}^n) = \begin{cases} P(\mathbf{A}_{n,i} = k) & k < K - q \\ P(\mathbf{A}_{n,i} \geq K - q) & k = K - q \\ 0 & \text{otherwise}. \end{cases} \tag{13}$$

Obviously, the difference of $A_{n,i}$ and $\mathbf{Q}_n$ is the number of packets dropped due to a full queue.

5.3 Transition probabilities

$P(\xi \rightarrow \zeta)$ with $\xi, \zeta \in \Sigma_n$ is the probability of going from ξ to ζ in one step. Transition probabilities not listed are zero, in particular for transitions with non-consecutive

Fig. 5 Schedule for $l_S = 5$, $\mathcal{T}_n = (1, 4)$ and corresponding state diagram for $K = 2$
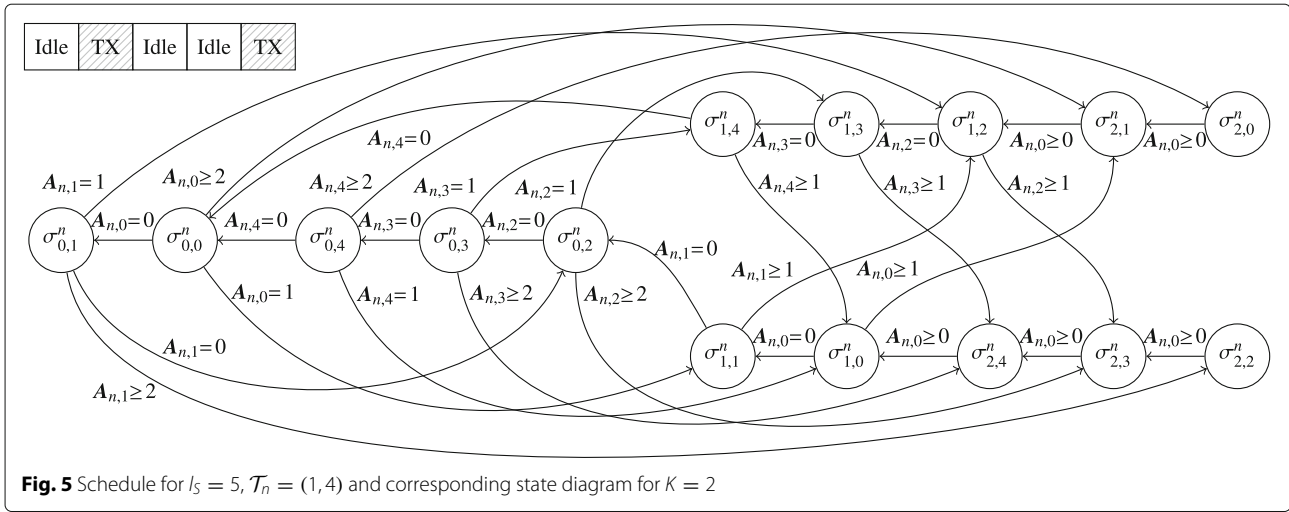
slots. Furthermore, % denotes the modulo operation. So for $0 \leq q \leq K$ and $0 \leq i < l_S$, all possible transitions are specified by

$$P\left(\sigma_{q,i}^n \to \sigma_{\max(q-\tau_{n,i},0)+k,\,(i+1)\%l_S}^n\right)$$
$$= P\left(\mathbf{Q}_n = k \,\middle|\, \sigma_{q,i}^n\right) \quad (14)$$

where

$$\tau_{n,i} = \begin{cases} 1 & i \in \mathcal{T}_n \\ 0 & i \notin \mathcal{T}_n \end{cases} \quad (15)$$

and k is the number of arriving or newly generated packets during the time slot that iterates from 0 to $K - q$. While this gives a complete description of the model, two corner cases should be mentioned explicitly for clarity. With an empty queue and $A_{n,i} = 0$, that is no arriving packets, the model stays in the states with $q = 0$. With a full queue, it is not possible to insert further packets into the queue so $\mathbf{Q}_n = 0$, and thus, the single possible transition out of the states with $q = K$ has the probability

$$P\left(\mathbf{Q}_n = 0 \,\middle|\, \sigma_{K,i}^n\right) = P\left(\mathbf{A}_{n,i} \geq 0\right) = 1, \quad (16)$$

either to a state with $q = K - 1$ if the current slot is a transmission slot or else to a state with $q = K$.

As shown in Appendix B, these probabilities can be used to calculate the stationary distribution of this Markov chain. This results in the probability $c_{n,q,i}$ of being in state $\sigma_{q,i}^n$ in the stationary case.

5.4 Transmission probability
From the $c_{n,q,i}$, the probability that a successful transmission takes place within a given slot is calculated from the complementary probability of being in a state with empty queue ($q = 0$) as

$$\mu_{n,i}^{\mathrm{TX}} = \tau_{n,i} \cdot \left(1 - \frac{c_{n,0,i}}{\sum_{q=0}^K c_{n,q,i}}\right). \quad (17)$$

5.5 Packet acceptance probability
If q packets are in the queue, $K - q$ packets can be inserted into the queue, so the expected number of *accepted* packets in state $\sigma_{q,i}^n$ is

$$E\left[\mathbf{Q}_n \,\middle|\, \sigma_{q,i}^n\right] = P\left(\mathbf{A}_{n,i} \geq K - q\right) \cdot (K - q)$$
$$+ \sum_{k=0}^{K-q-1} P\left(\mathbf{A}_{n,i} = k\right) \cdot k. \quad (18)$$

Since the events of being in a state are mutually exclusive and exhaustive, $E[\mathbf{Q}_n]$ can be calculated from this according to the law of total expectation as

$$E[Q_n] = \sum_{q=0}^{K} \sum_{i=0}^{l_S-1} c_{n,q,i} \cdot E\left[\mathbf{Q}_n \,\middle|\, \sigma_{q,i}^n\right] \quad (19)$$

and the overall expected number of packets per slotframe is thus $l_S \cdot E[\mathbf{Q}_n]$. Together with the overall expected number of packets arriving at the queue, the overall packet acceptance probability is calculated as

$$P_{\mathrm{accept},n} = \frac{l_S \cdot E[\mathbf{Q}_n]}{\mathcal{A}_n}. \quad (20)$$

5.6 Queuing delay
The number of time steps it takes until an arriving packet is transmitted is described by the random variable $\mathbf{D}_n$. For convenience, we define

$$\phi_n(i) = \begin{cases} |\mathcal{T}_n| - 1 & i \leq t_{n,0} \ \vee \\ & i > t_{n,|\mathcal{T}_n|-1} \\ g : t_{n,g} < i \leq t_{n,g+1} & \text{else}, \end{cases} \quad (21)$$

as the index of the transmission slot preceding i and

$$\delta(i,j) = \begin{cases} j - i & j \geq i \\ j - i + l_S & \text{otherwise}, \end{cases} \quad (22)$$

as the number of slots when going from i to j. Then, given the event of being in state $\sigma_{q,i}^n$ after accepting a packet and pushing it in the queue, $\boldsymbol{D}_n$ is calculated as

$$\boldsymbol{D}_n\left(\sigma_{q,i}^n\right) = f_q l_S + 1 + \delta\left(i, \ t_{n,(\phi_n(i)+q)\%|\mathcal{T}_n|}\right) \tag{23}$$

where $f_q = \left\lceil \frac{q}{|\mathcal{T}_n|} - 1 \right\rceil$ is the number of iterations over the full slotframe, the summand 1 accounts for the transmission slot itself and the last summand for the remaining packets after $f_q \cdot |\mathcal{T}_n|$ packets were transmitted. The expected delay for a packet arriving at an arbitrary point in time can be calculated from this as

$$E[\boldsymbol{D}_n] = \sum_{q=0}^{K} \sum_{i=0}^{l_S-1} c_{n,q,i} \cdot \boldsymbol{D}_n\left(\sigma_{g,h}^n\right) \tag{24}$$

with $\sigma_{g,h}^n = \sigma_{\max(q-\tau_n,0)+1,\ (i+1)\%l_S}^n$

considering the additional transition to account for the arriving packet itself.

5.7 Single node evaluation

At first, the results for the queuing model on a single node are analyzed without considering the effects of multi-hop networks. The purpose of this section is to demonstrate the benefit of the refined model in contrast to a simple M/D/1/K model by comparing them with an event-based simulation. For this, an exemplary schedule is shown in Fig. 6 together with the two approximations presented in the following. The schedule is construed in a way that on average the same amount of traffic enters and leaves the system. However, due to a finite queue length of $K = 5$, packet drops can still occur. In Fig. 7, the following four evaluations are compared:

- In the *M/D/1/K* model, there is no option to specify multiple slots and only an overall packet rate rate can be specified.
- In the *Distributed* approximation, the transmission and idle slots are modeled correctly, but incoming traffic is modeled to follow a Poisson distribution and not a Bernoulli distribution. Furthermore, it is distributed over all slots. To get the same $\mathcal{A}_n$ of 1 as in the actual scenario, all $\lambda_{n,i}$ are set to $\frac{1}{l_S} = 0.2$. This scenario allows to assess the deficiency of not including the $\beta_{n,i}$ in the model.
- The *Full Model* scenario additionally models the reception slot correctly as presented.
- The *Simulation* is a discrete event simulation of the queue based on SimPy [38] that implements the policy as presented at the beginning of Section 5. The mean and 95% confidence interval are shown of 10 runs with 10,000 packets.

While the full model matches the simulation very well, the probability of dropping packets due to a full queue is higher in the distributed scenario, since the variance of the packet reception is higher, even though the average reception rate is equal. In fact, for $g_{\text{up}} = 0$ and at least as much TX slots as RX slots, the full model will never indicate a packet loss ($P_{\text{accept},n} = 100\%$), while in the simplified models there is a non-zero probability that more packets are received in a slotframe than sent, eventually leading to packet loss.

The M/D/1/K model gives even less accurate results, because the service time is constant and equal to $l_S \cdot T_s$, while in the distributed and the full model case, packets

Full	TX	RX	Idle	Idle	Idle
$\beta_{n,i}$	0	0.5	0	0	0
$\lambda_{n,i}$	0.1	0.1	0.1	0.1	0.1

Distributed	TX	Idle	Idle	Idle	Idle
$\beta_{n,i}$	0	0	0	0	0
$\lambda_{n,i}$	0.2	0.2	0.2	0.2	0.2

M/D/1/K	TX				
$\beta_{n,i}$	0				
$\lambda_{n,i}$	1				

Fig. 6 Exemplary schedule for evaluating the queuing model together with two approximations, all with the same $\mathcal{A}_n = 1$
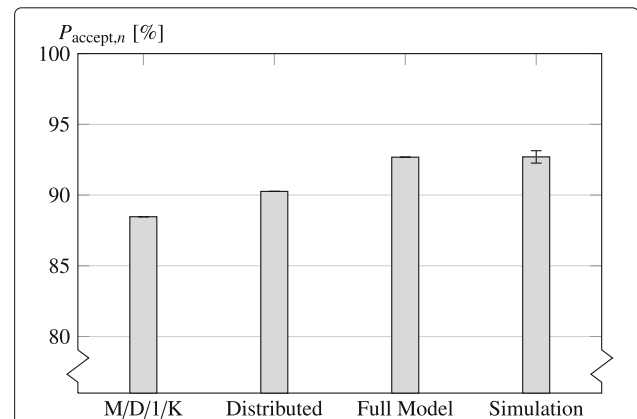

Fig. 7 Comparing the full presented model with a simulation and simpler models: An M/D/1/K model and a model that considers the TX and Idle slots, but distributes the reception over all slots

might be processed faster if the queue is empty and they arrive during the later slots.

5.8 Queue distribution

In this section, we take a deeper look into the distribution of the queue level and its influence on the probability of accepting a packet for different traffic loads.

For this, a scenario with $K = 10$, $l_S = 5$ and a single transmission slot is considered.

In Fig. 8, the probability distribution of the queue level is plotted for different traffic loads $\mathcal{A}_n$. For this, the probability of being in a state with q packets in the queue is calculated as

$$c_{n,q,\star} = \sum_{i=0}^{l_S-1} c_{n,q,i}. \tag{25}$$

In the first scenario, the traffic load is distributed over the $\lambda_{n,i}$, corresponding to a node that does only generate but does not forward traffic, and in the second scenario, it is distributed only over the $\beta_{n,i}$, corresponding to a node that only forwards traffic generated by other nodes.

For a low rate of $\mathcal{A}_n = 0.5$, the queue is empty most of the time with a decaying probability of having more packets in the queue. The probabilities for queue levels larger than 3 are negligible, so the probability of accepting new packets is $P_{\text{accept},n} = 1.00$ for both scenarios.

For a medium rate of $\mathcal{A}_n = 1$, where on average one packet is sent and received every slotframe, the distribution is nearly constant. This leads to a packet acceptance probability of $P_{\text{accept},n} = 0.95$ for the first and $P_{\text{accept},n} =$

Fig. 8 Distribution of queue levels for $K = 10$, $l_S = 5$ and a single transmission slot

0.96 for the second scenario. Interestingly, the probability for $q = K$ is significantly smaller than the others.

This is due to the policy that in a transmission slot with $q = K$, no new packets can be pushed to the queue, but certainly, a packet will be sent, i.e., $q = K - 1$ in the next slot. However, for all other queue levels, it is possible to maintain the same queue level by sending and receiving one packet. That is even more apparent for $\mathcal{A}_n = 1.5$ where a queue level larger than $q = 5$ most of the time leads to a packet acceptance probability of $P_{\text{accept},n} = 0.67$ for both scenarios. Having $q = K$ more often than other queue levels is only possible for very high rates of $\mathcal{A}_n = 2.5$. Here, the node is highly congested and a $P_{\text{accept},n} = 0.40$ is achieved in both scenarios.

When comparing the scenarios, we see that the difference is not very large, but in the second scenario, the probabilities are shifted to the boundaries. This evaluation also shows that often only a small part of the queue is actually utilized. For $\mathcal{A}_n = 0.5$, a maximum queue size of $K = 3$ would be sufficient to prevent packet loss. Also for $\mathcal{A}_n = 2.5$, this would not change the packet acceptance probability. The probability distribution would shift to the left, but does not change its overall shape relative to the upper bound. Of course, the queuing delay would be reduced by this. The bottom line is that the maximum queue length K has the highest influence in scenarios where the amount of incoming and outgoing traffic is similar.

6 Multi-hop model

To model multi-hop communication, every node in the network gets its own instance of the previously presented model. They are then linked to get a full network for allowing to determine network-wide metrics.

6.1 Traffic generation and forwarding

For the purpose of this paper, we assume a data-collection scenario with sink v_0. Every node, except v_0, generates packets with exponentially distributed intervals with mean I_{up} that are to be forwarded to v_0 via a routing tree. The base time unit is the slot length T_s, so the generation rate is

$$g_{\text{up}} = \frac{T_s}{I_{\text{up}}}. \tag{26}$$

In the following, only homogeneous traffic generation is considered, so we set

$$\lambda_{n,i} = \begin{cases} g_{\text{up}} & v_n \neq v_0 \\ 0 & v_n = v_0, \end{cases} \forall 0 \leq i < l_S. \tag{27}$$

but other traffic patterns can be obtained by choosing individual values for $\lambda_{n,i}$.

Besides the traffic generation itself, there is a probability $\mu_{n,i}^{\text{RX}}$ of receiving a packet from a neighbor in a reception slot to be forwarded to the sink. This value is calculated
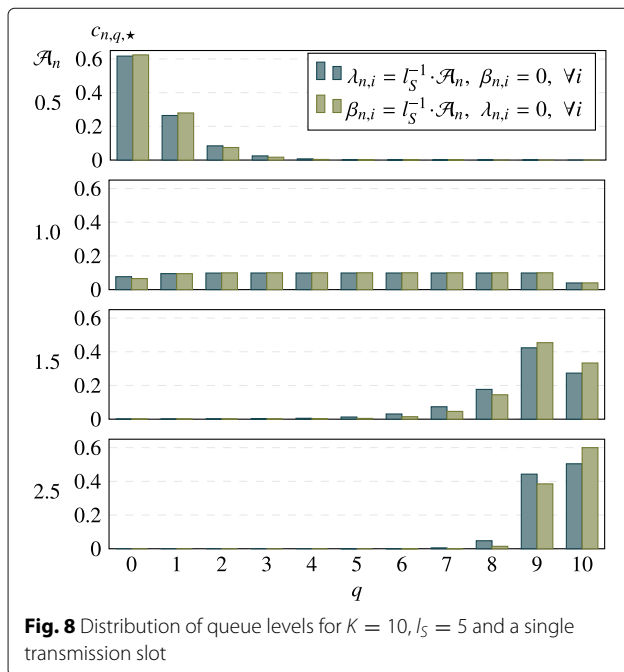
from the probability $\mu^{\mathrm{TX}}_{\mathcal{K}_n(i),i}$ that the neighbor $\mathcal{K}_n(i)$ is transmitting in the given slot as

$$\mu^{\mathrm{RX}}_{n,i} = \begin{cases} \mu^{\mathrm{TX}}_{\mathcal{K}_n(i),i} & i \in \mathcal{R}_n \\ 0 & \text{else.} \end{cases} \quad (28)$$

The forwarding is modeled by the Bernoulli part of the traffic model since at most one packet can be received per slot, so $\beta_{n,i} = \mu^{\mathrm{RX}}_{n,i}$.

6.2 Network throughput
The throughput of the network corresponds to the number of packets arriving at the sink ν_0 per time. Since the sink does not generate traffic itself, the throughput can be calculated as

$$\frac{\mathcal{A}_0}{T_s} = T_s^{-1} \cdot \sum_{i=0}^{l_S-1} \mu^{\mathrm{RX}}_{0,i}. \quad (29)$$

6.3 End-to-end metrics
The packet delivery ratio (PDR) $\mathfrak{R}_{\mathrm{up},n}$, that is the probability that a packet originating from node ν_n is finally received by the sink ν_0, is given by the product of $P_{\mathrm{accept},n}$ over the path

$$\mathfrak{R}_{\mathrm{up},n} = \begin{cases} \mathfrak{R}_{\mathrm{up},p_n} \cdot P_{\mathrm{accept},n} & \nu_n \neq \nu_0 \\ 1 & \nu_n = \nu_0, \end{cases} \quad (30)$$

where p_n is the parent of ν_n in the routing tree.

Similarly, the end-to-end delay $\mathfrak{D}_{\mathrm{up},n}$ is calculated as

$$\mathfrak{D}_{\mathrm{up},n} = \begin{cases} \mathfrak{D}_{\mathrm{up},p_n} + E[D_n] & \nu_n \neq \nu_0 \\ 0 & \nu_n = \nu_0. \end{cases} \quad (31)$$

6.4 Comparing model and simulation
After evaluating the model for a single node only, we now build up a full multi-hop network to analyze its performance. An analytical model is very valuable for appreciating the validity of simulation results. Thus, we use the model to compare its results to the Contiki implementation of the Orchestra scheduling [6] for TSCH running in the COOJA simulator [39] with a maximum queue length of $K = 16$. It is evaluated for a data-collection scenario in a concentric topology with 19 nodes as shown in Fig. 9. Sender-based Dedicated (SBD) Orchestra Slots are used, where every node has a dedicated transmission slot and thus, $\mathcal{D}_{i,\cup} = \emptyset$ and so the schedule is conflict-free and valid. Furthermore, the slotframe length is minimized to maximize the throughput under the given constraint.

The results in Fig. 10 show the end-to-end packet delivery ratio (PDR) averaged over the nodes in the outer circle as well as the end-to-end delay. In the simulation, every node sends packets to the center with exponentially distributed intervals. After a warm-up phase of 15 minutes, 100 packets are monitored if they arrive and how long they take. The resulting mean and the 95% confidence interval over 5 runs are shown in the plot.
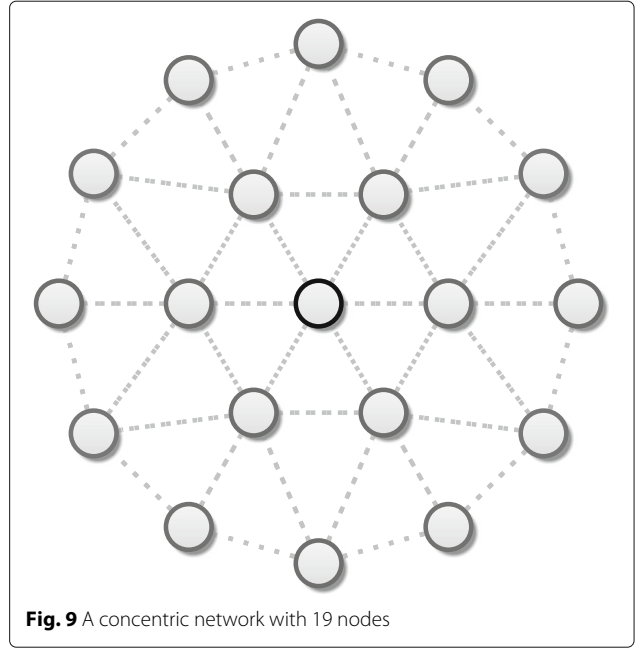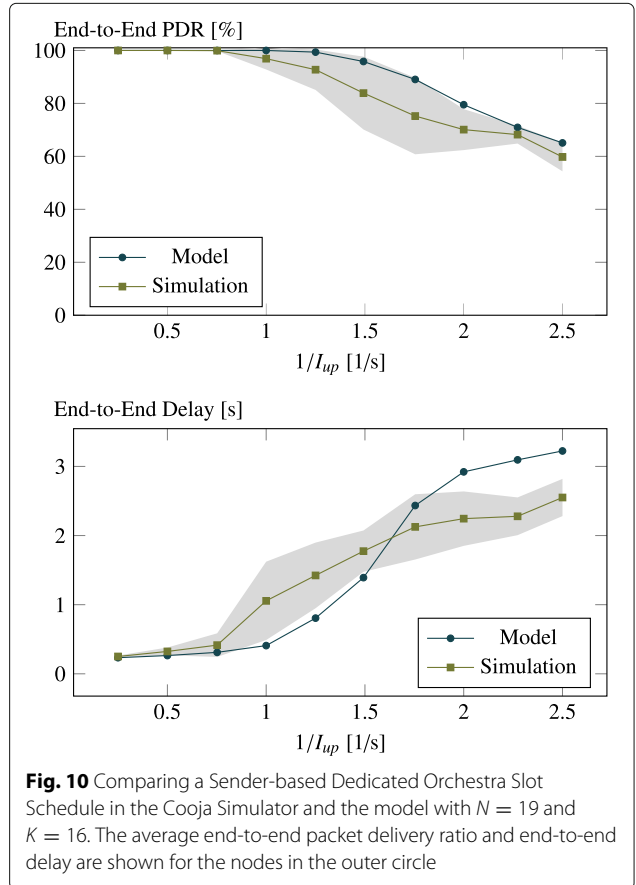

Fig. 9 A concentric network with 19 nodes

Fig. 10 Comparing a Sender-based Dedicated Orchestra Slot Schedule in the Cooja Simulator and the model with $N = 19$ and $K = 16$. The average end-to-end packet delivery ratio and end-to-end delay are shown for the nodes in the outer circle

While the simulation uses the Routing Protocol for Low power and Lossy Networks (RPL) [40], the analytical model uses a predefined tree routing as in [41]. The PDR $\mathfrak{R}_{\mathrm{up},n}$ and the end-to-end delay $\mathfrak{D}_{\mathrm{up},n}$ are shown in the plot. Obviously, no confidence intervals are shown because no randomness is included in the calculation. The results show a good conformance between model and simulation and demonstrate the applicability of the analytical model for multi-hop networks as well as the validity of the Contiki simulation for the given scenario. The existing differences can be mainly traced back to the varying routing trees in the simulation runs.

7   Comparing multi-hop schedules

The analytical model is also very useful to assess new scheduling algorithms without the need to implement them for a full simulation stack. Orchestra is an appropriate schedule for many applications, especially since it requires no management traffic, but its main disadvantage is its inability to handle different traffic loads efficiently. In the given scenario, the inner nodes have to handle the aggregated traffic of all nodes in their sub tree, so they should be able to use more slots than the leafs.

Algorithm 1 Determine proper descendants

1: **initialization**
2:     mark all children as unvisited $\forall\ v_n$

3: **on message** Forward **at** v_n **from** v_s
4:     $v_n.\gamma \leftarrow 0$
5:     HandleNode(v_n)

6: **on message** Backtrack(j) **at** v_n **from** v_s
7:     $v_n.\gamma_s \leftarrow j$ ▷ Store proper descendants of children
8:     $v_n.\gamma \leftarrow v_n.\gamma + j$
9:     HandleNode(v_n)

10: **procedure** HandleNode(v_n)
11:     **if** $\exists$ unvisited child v_u **then**
12:         mark v_u as visited
13:         **send** Forward to v_u
14:     **else if** $v_n \neq v_0$ **then**   ▷ leaf or sub tree fully handled
15:         **send** Backtrack($v_n.\gamma + 1$) to v_{p_n}

In this section, two traffic-aware schedules are introduced for the data collection scenario presented in the previous section for a given tree. The main idea is that every node has enough slots for the traffic generated at that node as well as for forwarding the traffic of its children, assuming equal traffic distribution. More formally, if γ_n denotes the number of proper descendants of node v_n, every node will get $\gamma_n + 1$ transmission slots towards the sink. Algorithm 1 describes a distributed algorithm to calculate γ_n for every node and also to make every node aware of the number of proper descendants of their children. For this, every node keeps the variables $v_n.\gamma$ for the

number of proper descendants of this node n and for each child s its number of proper descendants $v_n.\gamma_s$. As before, v_0 denotes the sink node and p_n the index of the parent of node v_n.

The algorithm is basically a depth-first search. It is started by sending Forward to v_0. After initializing $v_0.\gamma$, Forward messages are sent until a leaf is reached. After this, a Backtrack message is sent back to the parent. The parent increments its counters before continuing the depth-first search. After a subtree is fully handled, the number of nodes in this subtree is sent back to the parent in a Backtrack message.

In the presented distributed algorithms, we assume all messages arrive correctly and in order. Otherwise, the algorithms would get a lot more complex. Associated problems and solutions for unreliable message exchange when building schedules are presented in [35].

The schedules presented in this section are meant to demonstrate the applicability of the analytical model and the advantage of traffic-aware schedules. However, they are not directly applicable to real-world applications due to their inability to cope with changes in the network topology and the traffic load without a complete recalculation. Yet, due to their simplicity, they serve as a starting point for more sophisticated traffic-aware schedules.

7.1   Traffic-aware schedule (single-channel)

The first traffic-aware scheduling algorithm as shown in Algorithm 2 is based on the scheduling algorithm type III from [42]. As for Orchestra's Sender-based Dedicated Slots, only one node in the entire network is sending at every given point in time, so again $\mathcal{D}_{i,\cup} = \emptyset$. It is therefore denoted as single-channel scheduling, though, similar to Orchestra, channel hopping could be used to mitigate external interferences.

Algorithm 2 Traffic-Aware Schedule (Single-Channel)

1: **initialization**
2:     $\mathcal{T}_n \leftarrow ()\ \forall\ v_n$
3:     $\mathcal{R}_n \leftarrow ()\ \forall\ v_n$
4:     mark all children as unvisited $\forall\ v_n$

5: **on message** Track(z) **at** v_n **from** v_s
6:     **if** $\exists$ unvisited child v_u **then**
7:         mark v_u as visited
8:         **send** Track(z) to v_u
9:     **else if** $v_n \neq v_0$ **then**   ▷ leaf or sub tree fully handled
10:         **for** $i \leftarrow z, \ldots, z + v_n.\gamma$ **do**
11:             $\mathcal{T}_n \leftarrow \mathcal{T}_n \frown (i)$ ▷ append to $\mathcal{T}_n$
12:             $\mathcal{K}_n(i) \leftarrow p_n$
13:             **send** AssignRX(i) to v_{p_n}
14:         **send** Track($z + v_n.\gamma + 1$) to v_{p_n}

15: **on message** AssignRX(i) **at** v_n **from** v_s
16:     $\mathcal{R}_n \leftarrow \mathcal{R}_n \frown (i)$
17:     $\mathcal{K}_n(i) \leftarrow s$

The algorithm is started by sending Track(1) to v_0 after every node has performed the initialization. The

algorithm is a depth-first search again, but instead of sending back the number of nodes in the subtree, multiple transmission slots are reserved towards the parent, one for handling the traffic of every proper descendant and one for the traffic generated at that node ($\gamma_n + 1$). The transmission slot is recorded in the local $\mathcal{T}_n$ and a message is sent to the parent v_{p_n} to record the reception slot in its $\mathcal{R}_{p_n}$. When sending the TRACK message back to the parent, the number of already assigned slots is sent along as an offset for the next slot assignment.

The overall number of slots l_S in a slotframe for this schedule is calculated as

$$l_S = 1 + \sum_{n=1}^{N-1} \gamma_n + 1, \qquad (32)$$

since every node (apart from v_0) performs $\gamma_n + 1$ slot allocations. The additional slot is the first slot of the slotframe that is usually left free for shared communication in most TSCH implementations, for example for management traffic. This also holds for the used Orchestra implementation.

7.2 Traffic-aware schedule (multi-channel)

In general, it is not required that only one node is sending at every point in time. Two pairs of nodes that are sufficiently apart can send at the same time without interference. Secondly, nodes can communicate on different channels to avoid interference. This spatial and frequency diversity can be exploited to shorten the slotframe and therefore increase the throughput and lower the latency.

A corresponding distributed algorithm is given in Algorithm 3. In contrast to the previous algorithm, where the child determines the transmission slots towards the parent, in this algorithm, the parent determines the reception slots in which it will expect the child to send. It is started by sending TRACK to v_0. As in the previous algorithm, every node requires $\gamma_n + 1$ transmission slots. This is also calculated in line 9 for a child u when a node is first handled by its parent. Afterwards, $\sigma = v_n.\gamma_u + 1$ slot allocations are performed and the loop is always finished in line 20 if we assume there are always enough channels (see below, otherwise line 14 would fail) and l_S is chosen as follows. For every allocation, a previously unused time slot is searched, recorded at sender and receiver and the used channel is blocked in the 2-hop neighborhood. For this, every node maintains a set of blocked channels for every slot $\mathcal{B}_n(i)$. This search will always be successful if we choose the slotframe length as

---

Algorithm 3 Traffic-Aware Schedule (Multi-Channel)

```
 1: initialization
 2:     T_n ← () ∀ v_n
 3:     R_n ← () ∀ v_n
 4:     B_n(i) ← () ∀ 0 ≤ i < l_S, ∀ v_n
 5:     mark all children as unvisited ∀ v_n

 6: on message TRACK at v_n from v_s
 7:     if ∃ unvisited child v_u then
 8:         mark v_u as visited
 9:         σ ← v_n.γ_u + 1
10:         for i ← 1, …, l_S − 1 do            ▷ slot 0 is reserved
11:             if i ∉ R_n ∪ T_n then                ▷ slot i is idle
12:                 R_n ← R_n ⌢ (i)              ▷ append to R_n
13:                 K_n(i) ← u
14:                 select c ∈ C \ B_n(i)
15:                 C_{v_n}(i) ← c
16:                 send ASSIGNTX(i, c) to v_u
17:                 send BLOCK(i, c, true) to all v ∈ N_n \ {v_u}
18:                 σ ← σ − 1
19:                 if σ = 0 then
20:                     break              ▷ all slots are assigned
21:         send TRACK to v_u
22:     else if v_n ≠ v_0 then        ▷ leaf or sub tree fully handled
23:         send TRACK to v_{p_n}

24: on message ASSIGNTX(i, c) at v_n from v_s
25:     T_n ← T_n ⌢ (i)
26:     K_n(i) ← s
27:     C_{v_n}(i) ← c
28:     send BLOCK(i, c, true) to all v ∈ N_n \ {v_s}

29: on message BLOCK(i, c, forward) at v_n from v_s
30:     B_n(i) ← B_n(i) ∪ c
31:     if forward then
32:         send BLOCK(i, c, false) to p_n
```

$$l_S = 1 + \max_{n=0,\dots,N-1} \begin{cases} 2 \cdot \gamma_n + 1 & v_n \neq v_0 \\ \gamma_n & v_n = v_0 \end{cases}, \qquad (33)$$

because the root requires that the slotframe has at least one slot for receiving (potentially forwarded) traffic from every proper descendant. For all other nodes, this holds, too, but in addition, the same number of slots is required for forwarding and one slot for transmitting the traffic generated at that node. Again, the additional slot is the shared first slot in the slotframe. Since we assume enough channels are available to avoid conflicts, the slotframe length corresponds to the requirement of the node with the largest required slotframe length.

In contrast to the other algorithms, transmissions can take place simultaneously without or with very little interference by assigning a dedicated channel or a non-conflicting channel hopping sequence to every conflicting link in a neighborhood. In the presented algorithm, this is ensured by signaling every slot assignment to all neighbors $\mathbf{N}_n$ of the sender and the receiver as well as the parents of the neighbors. Thereby, the four interference constellations in Fig. 3 are avoided. This idea is similar to the slot allocation handshake of DSME as well as the DeBraS scheduling algorithm [33] and is especially important for dense networks.

In general, this is a graph coloring problem where $\mathbf{L}_i$ is the set of vertices and $\mathcal{D}_{i,\cup}$ is the set of edges. Executing the presented heuristic algorithm for the first slot results in the coloring shown in Fig. 11. Here, only three channels are required, so the 16 channels available for IEEE 802.15.4 in the 2.4 GHz band are more than sufficient. In general, however, no upper bound can be given for the number of required channels, because the conflict graph is not necessarily planar. Figure 12 shows an example that requires five channels and could be extended to an arbitrary number of channels. While in general finding a valid coloring with at most 16 channels is not ensured, it is usually more than sufficient in real-world applications.

7.3 Evaluation

In Fig. 13, the achievable throughput for different schedules is compared for multiple scenarios, i.e., for a network of $N = 19$ and $N = 37$ nodes in a concentric topology and for maximum queue sizes of $K = 6$ (dashed) and $K = 16$ (solid). For this, the number of packets received by the sink v_0 is plotted over the packet generation rate of every node. For low rates, the network is able to handle the complete traffic, so the throughput rises linearly. For high rates, the networks is saturated and increasing the rate does not increase the throughput anymore.

In general, the traffic-aware schedules provide significantly more throughput than Orchestra, while the multi-channel schedule has an even higher throughput than the single-channel schedule. The higher throughput can be explained by considering that nodes higher in the tree that need to handle more traffic have more slots compared to nodes with low traffic demand.
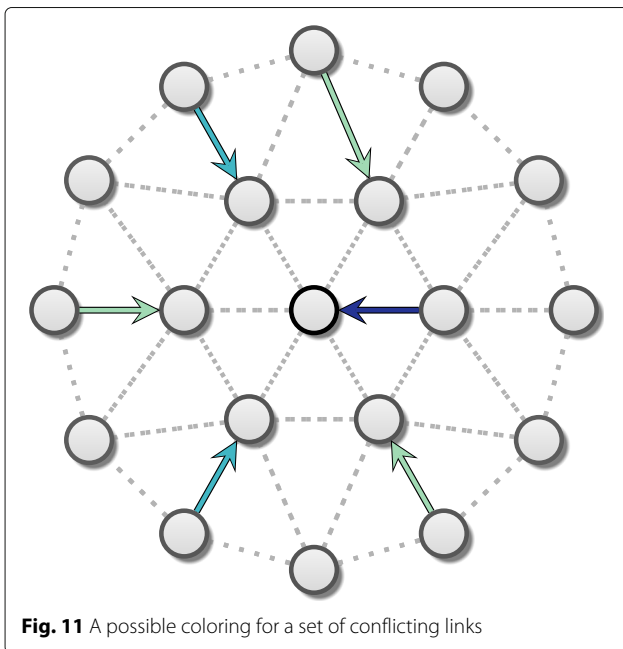

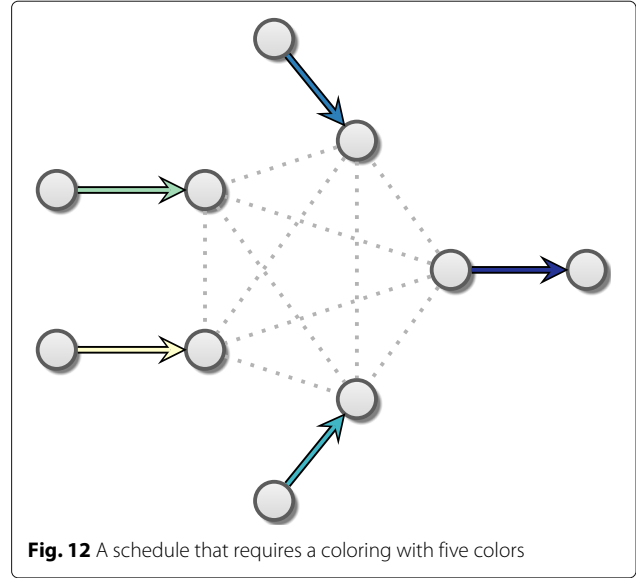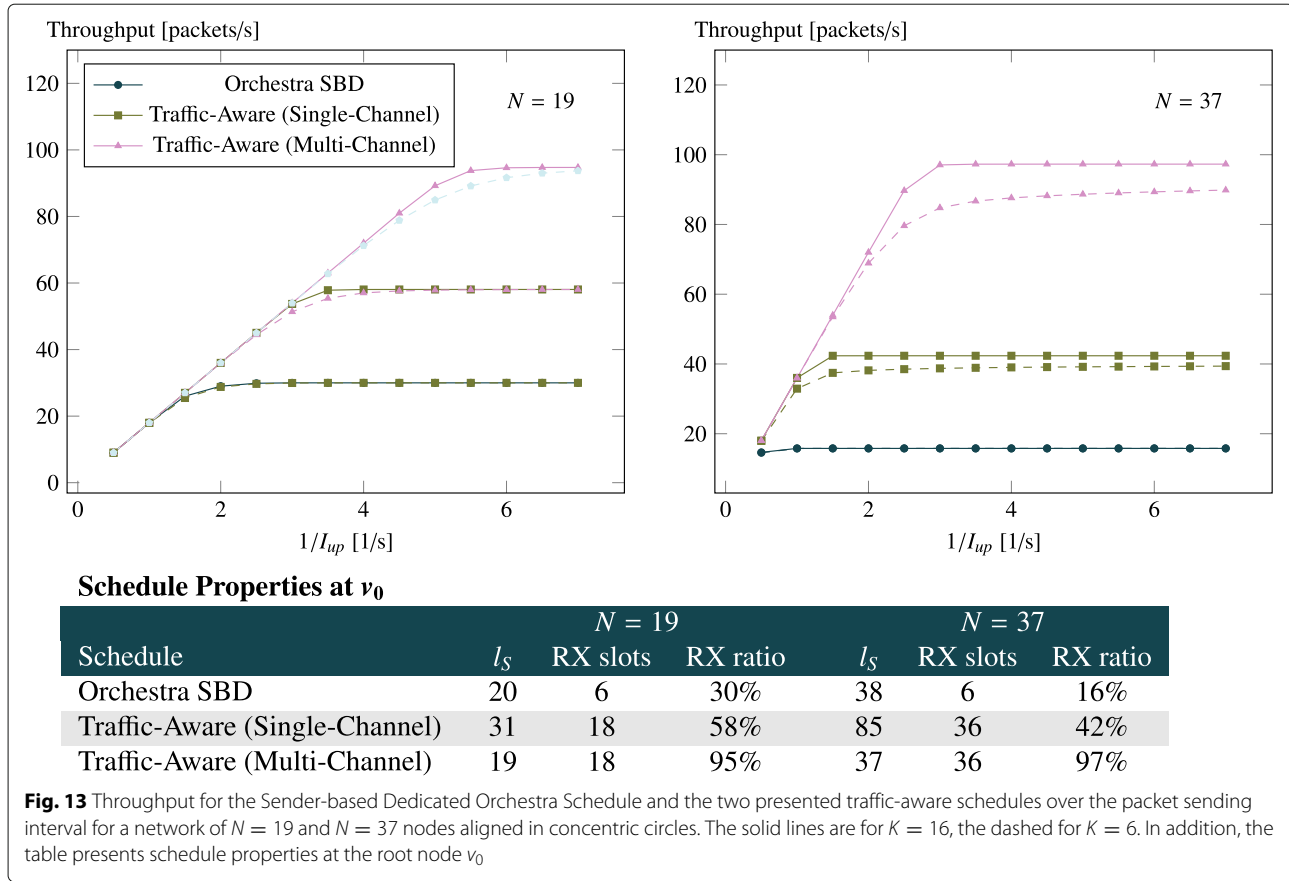**Fig. 11** A possible coloring for a set of conflicting links

Fig. 12 A schedule that requires a coloring with five colors

It is also apparent in Fig. 13 that the maximum queue length K is most significant in the transition section, in conformance to Section 5.8. Compared to the applied schedule, it has a lower impact on the throughput, but the difference is larger for $N = 37$ than for $N = 19$ due to the higher number of hops.

Also, when comparing different network sizes, we see that the saturation is reached for smaller rates. This is expected, because the same packet generation rate applied at more nodes leads to more overall traffic. Secondly, the single-channel schedule and even more Orchestra SBD show a significantly lower throughput, because having more nodes requires a higher number of slots during which the sink is idle. This does not hold for the multi-channel schedule, because spatial reuse is possible.

In the table of Fig. 13, the slotframe lengths are given together with the number and ratio of reception slots at the sink v_0. For Orchestra SBD, every neighbor of v_0 has only one slot, so v_0 has a long phase of inactivity. For the traffic-aware schedules, the inner nodes get more slots because of their higher traffic load. Thus, the sink can receive more packets per slotframe resulting in a higher throughput. The missing slot in the last row is due to the shared slot. In the multi-channel schedule, the sink can even potentially receive data traffic in every slot, apart from the first one. This comparison again explains the difference in throughput for the different network sizes, because comparative to the network with 19 nodes, the RX ratio is significantly lower for $N = 37$ and the Orchestra SBD and single-channel schedules, while it is even larger for the multi-channel schedule where the shared slot has a lower impact due to the longer slotframe length.

Schedule Properties at v_0

Schedule	l_S	N = 19 RX slots	RX ratio	l_S	N = 37 RX slots	RX ratio
Orchestra SBD	20	6	30%	38	6	16%
Traffic-Aware (Single-Channel)	31	18	58%	85	36	42%
Traffic-Aware (Multi-Channel)	19	18	95%	37	36	97%

Fig. 13 Throughput for the Sender-based Dedicated Orchestra Schedule and the two presented traffic-aware schedules over the packet sending interval for a network of $N = 19$ and $N = 37$ nodes aligned in concentric circles. The solid lines are for $K = 16$, the dashed for $K = 6$. In addition, the table presents schedule properties at the root node v_0

8   Possible extensions

8.1   IEEE 802.15.4 DSME

While the focus of this paper is TSCH due to its higher flexibility and broader usage, the model can easily be applied to DSME, too. The main difference between TSCH and DSME is the availability of distributed management procedures for setting up schedules in the latter, but since the presented model analyzes schedules in a steady state, the only major difference is the slot structure. In TSCH, the time slots can be arbitrarily dedicated as contention-free and contention-access slots. DSME has a less flexible structure. It consists of one beacon slot, 8 contention-access slots, and 7 contention-free slots aligned in a fixed, yet configurable, repeated sequence. The contention-access phase is usually used for management, similar to, but longer than the extra slot used in this paper. Secondly, the slots are usually shorter, because TSCH requires some extra time in every slot due to its time synchronization procedure, while DSME uses a dedicated beacon slot (see [43] for details). Therefore, the timing of the model has to be adapted and the schedule has to account for the slots not used for contention free communication.

Furthermore, due to the less flexible structure, a scheduling algorithm such as Algorithm 3 would lead to

excess slots that are not utilized. Therefore, a more elaborate algorithm is required to achieve optimal performance by evenly distributing the excess slots.

8.2   Other sources of packet loss

As outlined in Section 3, most packet losses in TSCH and DSME networks are queue drops. However, in practice, other sources of packet loss are inevitable. The consideration of packet loss during the transmission can be integrated in the presented model by replacing Eq. (28) with

$$\mu_{n,i}^{\text{RX}} = \begin{cases} \mu_{\mathcal{K}_n(i),i}^{\text{TX}} \left(1 - \text{PER}_{b,(\mathcal{K}_n(i),n)}(t)\right) & i \in \mathcal{R}_n \\ 0 & \text{else,} \end{cases} \quad (34)$$

where $\text{PER}_{b,(\mathcal{K}_n(i),n)}(t)$ is the probability that a packet transmission with b byte fails at time t.

An example for a time-independent calculation of $\text{PER}_{b,(m,n)}$ can be found in [41] and in [44] a model for Rayleigh-lognormal fading in IEEE 802.15.4 networks is given. Time dependence is required for accurate modeling of external interferences or fading channels, but that is out of the scope of this paper.

Furthermore, since the model assumes no losses on the physical layer, retransmissions are not considered. If other

sources of packet loss are considered, retransmissions have to be integrated in the model by adding transitions in the Markov chain for maintaining the queue level after a failed transmission. For the considered scenario, these transitions would never be taken and thus not change the results.

9 Conclusions

The paper presents an analytical approach for the assessment of wireless mesh networks that use a collision-free TDMA. A queuing model based on a Markov chain is proposed that models forwarding traffic and irregular slot schedules accurately, in contrast to the well-known M/D/1/K model. These models are linked together to build up a multi-hop model of the whole network for calculating packet delivery ratio, end-to-end delay, and throughput.

The results demonstrate the increased accuracy compared to the M/D/1/K model and illustrate the effect of a finite queue by showing the queue level distribution. For evaluating the multi-hop model, a data-collection scenario is applied. The analytical model is compared to a simulation of the Orchestra schedule, showing good conformance. Finally, two distributed traffic-aware scheduling algorithms are presented. The higher throughput achieved by traffic-awareness is demonstrated and the influence of the maximum queue length is shown.

The calculations were conducted by means of the Portable, Extensible Toolkit for Scientific Computation (PETSc) [45, 46], the simulations with COOJA [39] and SimPy [38]. The open source implementation of the models can be accessed at [7].

Overall, the proposed analytical model, together with its software implementation, is a useful tool for testing new ideas while developing new slot schedules. It is also very helpful for practitioners who want to estimate the performance of wireless mesh networks.

Appendix A: Expected value of $A_{n,i}$
The expected value of $A_{n,i}$ is calculated as

$$
\begin{aligned}
E\big[A_{n,i}\big] &= \sum_{k=0}^{\infty} k \cdot P\big(A_{n,i}=k\big) \\
&= \sum_{k=1}^{\infty} k\Big(\big(1-\beta_{n,i}\big)\cdot P\big(\Pi_{n,i}=k\big) \\
&\qquad + \beta_{n,i}\cdot P\big(\Pi_{n,i}=k-1\big)\Big) \\
&= \big(1-\beta_{n,i}\big)\cdot\left(\sum_{k=1}^{\infty}k\cdot P\big(\Pi_{n,i}=k\big)\right) \\
&\quad + \beta_{n,i}\cdot\sum_{k=1}^{\infty}k\cdot P\big(\Pi_{n,i}=k-1\big)
\end{aligned}
\tag{35}
$$

so with

$$
\begin{aligned}
\sum_{k=1}^{\infty}k\cdot P\big(\Pi_{n,i}=k\big) &= \sum_{k=1}^{\infty}k\cdot\frac{\lambda_{n,i}^{k}}{k!}e^{-\lambda_{n,i}} \\
&= \lambda_{n,i}\cdot e^{-\lambda_{n,i}}\sum_{k=1}^{\infty}\frac{\lambda_{n,i}^{k-1}}{(k-1)!} \\
&= \lambda_{n,i}\cdot e^{-\lambda_{n,i}}\sum_{j=0}^{\infty}\frac{\lambda_{n,i}^{j}}{j!} \\
&= \lambda_{n,i}\cdot e^{-\lambda_{n,i}}e^{\lambda_{n,i}} = \lambda_{n,i}
\end{aligned}
\tag{36}
$$

and

$$
\begin{aligned}
\sum_{k=1}^{\infty}k\cdot P\big(\Pi_{n,i}=k-1\big) &= \sum_{k=1}^{\infty}k\cdot\frac{\lambda_{n,i}^{k-1}}{(k-1)!}e^{-\lambda_{n,i}} \\
&= e^{-\lambda_{n,i}}\left(\left(\sum_{k=1}^{\infty}\frac{\lambda_{n,i}^{k-1}}{(k-1)!}\right)+\sum_{k=1}^{\infty}(k-1)\frac{\lambda_{n,i}^{k-1}}{(k-1)!}\right) \\
&= e^{-\lambda_{n,i}}\left(\left(\sum_{k=1}^{\infty}\frac{\lambda_{n,i}^{k-1}}{(k-1)!}\right)+\lambda_{n,i}\sum_{k=2}^{\infty}\frac{\lambda_{n,i}^{k-2}}{(k-2)!}\right) \\
&= e^{-\lambda_{n,i}}\left(\left(\sum_{j=0}^{\infty}\frac{\lambda_{n,i}^{j}}{j!}\right)+\lambda_{n,i}\sum_{j=0}^{\infty}\frac{\lambda_{n,i}^{j}}{j!}\right) \\
&= e^{-\lambda_{n,i}}\big(e^{\lambda_{n,i}}+\lambda_{n,i}\cdot e^{\lambda_{n,i}}\big)=1+\lambda_{n,i}
\end{aligned}
\tag{37}
$$

we finally get

$$
E\big[A_{n,i}\big] = \big(1-\beta_{n,i}\big)\lambda_{n,i}+\beta_{n,i}\big(\lambda_{n,i}+1\big).
\tag{38}
$$

Appendix B: Stationary distribution
The stationary distribution of the presented Markov chain for node v_n is denoted as

$$
c = \left(c_{n,\left\lfloor\frac{j}{l_S}\right\rfloor,j\%l_S}\right)_{j=0,\dots,(K+1)\cdot l_S-1}
\tag{39}
$$

where all $0\le c_{n,q,i}\le 1$ and $c\cdot e=1$ with

$$
e=\big(1\ 1\ \dots\ 1\big)^{T},
\tag{40}
$$

that is the normalization criterion that the probabilities have to sum up to 1. The stationary distribution is calculated as the solution of $cP=c$, where P is the transition probability matrix

$$
\begin{aligned}
P &= \big[p_{j,k}\big]_{((K+1)\cdot l_S)\times((K+1)\cdot l_S)} \\
p_{j,k} &= P\left(\sigma_{\left\lfloor\frac{j}{l_S}\right\rfloor,j\%l_S}^{n}\rightarrow\sigma_{\left\lfloor\frac{k}{l_S}\right\rfloor,k\%l_S}^{n}\right).
\end{aligned}
\tag{41}
$$

This can be rewritten as

$$
c(I-P)=0\Leftrightarrow(I-P)^{T}c^{T}=0,
\tag{42}
$$

with the identity matrix I. This is a homogeneous system of $(K+1)\cdot l_S$ linear equations and the same number of unknowns.

Fig. 14 A schedule with unreachable states $\sigma^n_{1,2}$
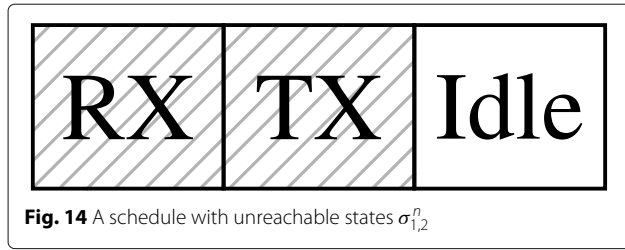
Irreducibility

Consider the example in Fig. 14 with $l_S = 3$, $K = 1$, $g_{up} = 0$ and $\mu^{RX}_{n,0} > 0$. Since nothing is generated and every packet received in slot 0 is immediately sent out again, the state $\sigma^n_{1,2}$ is never visited when starting with $q = 0$. It is also possible to construct more complex examples where blocks of states exist that are linked together, but are not reachable from any state with $q = 0$. This property makes the Markov chain reducible and thus a unique c is not guaranteed by the following proof. Finding these states corresponds to finding the states that are not in the strongly connected set that contains $\sigma^n_{0,0}$.

Since it is meaningless for the application to assign any $c_{n,q,i} > 0$ to those states, they are set to zero and the corresponding columns and rows are removed from P, making the Markov chain and P irreducible, that is there is no permutation of the rows and columns of P resulting in

$$\begin{pmatrix} A & B \\ 0 & D \end{pmatrix}, \tag{43}$$

with the square matrices A and D and the matrix 0 with all elements zero.

Rank of $I - P$

In the following we prove that the matrix $Q = I - P$ has rank $n - 1$ if P is irreducible. The proof goes along the lines of [47]. Since the outgoing transitions of a state have to sum up to one, that is

$$\forall \sigma^n_{q_1,i_1} \in \Sigma_n : \sum_{\sigma^n_{q_2,i_2} \in \Sigma_n} P\left(\sigma^n_{q_1,i_1} \to \sigma^n_{q_2,i_2}\right) = 1, \tag{44}$$

it holds that $Pe = 1$; thus $(I - P)e = 0$, so the matrix Q has at least one zero eigenvalue, is therefore singular and its rank is at most $n - 1$.

Assuming a rank of $n - 2$ or less, then there is at least one vector x with $Qx = 0$ that is orthogonal to e, i.e. $x^T e = 0$. Thus, all linear combinations of x and e are also in the null space of Q. In particular, this holds for $d = x - m \cdot e$ where m is the minimum entry of x. Then at least one, but not all, elements of d are 0 and all others are larger than 0. Note that this would not hold for x parallel to e. However, this is not possible since $x^T e = 0$.

Since permutations do not change the rank, we assume without loss of generality the first $u > 0$ elements of d are

positive and the remaining $n - u$ elements are zero. It holds

$$Qd = 0 \Leftrightarrow (I - P)d = Id - Pd = 0$$
$$\Leftrightarrow Pd = Id \Leftrightarrow Pd = d. \tag{45}$$

This can be partitioned as

$$Pd = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \begin{pmatrix} d_1 \\ \vdots \\ d_u \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} d_1 \\ \vdots \\ d_u \\ 0 \\ \vdots \\ 0 \end{pmatrix}, \tag{46}$$

with C of dimension $(n - u) \times u$. Therefore, it holds

$$C \begin{pmatrix} d_1 \\ \vdots \\ d_u \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{47}$$

For $d_i > 0 \ \forall 1 \le i \le u$ this is only possible if all entries of C are zero or there are negative entries in C. The first one contradicts the irreducibility of P and the second one would require the existence of negative probabilities in P. Therefore, $I - P$ has rank $n - 1$. Since the rank is maintained by transposition, the matrix $(I - P)^T$ has rank $n - 1$, too.

Solution of $cP = c$

Since $(I - P)^T$ has rank $n - 1$, the homogeneous system of linear equations $(I - P)^T c^T = 0$ has a solution space of dimension one, so since $c \cdot e = 1$, there is a unique stationary distribution. Furthermore, if we find any vector $y \neq 0$ with $(I - P)^T y^T = 0$, we can get c by normalization.

Though most methods for the numerical solution of systems of linear equations are tailored to the handling of regular matrices, many can be adapted for the singular case as presented in [48]. For our application, the implementation of the generalized minimal residual method (GMRES) in PETSc [46] with SOR preconditioning and initial guess $0.5e$ turned out to work very well. For more background about why and when GMRES is applicable to Markov chains, see Section 4.4.4 in [48].

Nomenclature

$\mathcal{A}_n$ Total expected number of generated or received packets per slotframe

$A_{n,i}$ Random variable for the number of arriving, i.e., generated and received, packets in slot i at node n

$\beta_{n,i}$ Packet probability in the Bernoulli part of the traffic model

$\mathcal{B}_n$ Relation of slots to blocked channels

$\mathcal{C}$ Channels

c Vector of the $c_{n,q,i}$

$\mathcal{C}_n$ Relation of slots to channel

$c_{n,q,i}$	Probability of being in state $\sigma_{q,i}^{n}$ in the stationary distribution
$D(v,w)$	Predicate that describes potential collisions between nodes
$\mathcal{D}_{i,l}$	The set of possibly disturbing links during slot i
$\mathbf{D}_n$	Random variable for the queuing delay
$\mathfrak{D}_{\mathrm{up},n}$	End-to-end packet delay
γ_n	Number of proper descendants of node n
g_{up}	Packet generation rate in upstream
I_{up}	Mean packet generation interval
$\mathcal{K}_n$	Relation of slots to counterpart
$\mathbf{L}_i$	The set of active links during slot i
$\lambda_{n,i}$	Packet rate in the Poisson part of the traffic model
l_S	Overall number of slots in the slotframe
$\mu_{n,i}^{\mathrm{RX}}$	Probability of receiving a packet in slot i
$\mu_{n,i}^{\mathrm{TX}}$	Probability of sending a packet in slot i
$\mathbf{N}_n$	Neighbors of node n
$P_{\mathrm{accept},n}$	Packet accepting probability
$\mathrm{PER}_{b,l}$	Packet error rate for a transmission of b bytes
p_n	Parent of node n
$\mathbf{\Pi}_{n,i}$	Random variable for the number of arriving packets according to the Poisson part of the traffic model
$\mathbf{Q}_n$	Random variable for the number packets inserted into the queue
$\mathcal{R}_n$	Reception slots
$\mathfrak{R}_{\mathrm{up},n}$	Reliability that a packet sent by n arrives at the sink
$r_{n,i}$	Reception slot.
$|\mathcal{T}_n|$	Number of transmission slots in the schedule for node n
$\mathbf{\Sigma}_n$	The set of states for the queuing model
$\sigma_{q,i}^{n}$	State in the queuing model
$\mathcal{T}_n$	Transmission slots
T_s	Duration of a slot
$t_{n,i}$	Transmission slot
v_0	Root node
v_n	Node with index n

Abbreviations

CAP: Contention access period; CSMA/CA: Carrier sense multiple access/collision avoidance; DSME: Deterministic and synchronous multi channel extension; GMRES: Generalized minimal residual method; PDR: Packet delivery ratio; PER: Packet error ratio; PETSc: Portable, extensible toolkit for scientific computation; RPL: Routing protocol for low power and lossy networks; RX: Reception; SBD: Sender-based dedicated; TDMA: Time-division multiple access; TSCH: Time slotted channel hopping; TX: Transmission

Availability of data and materials

The software implementing the presented model is available at https://github.com/koalo/AnalyticalMultiHop under the terms of the GNU General Public License v3.0. It is written in C++ and based on the Portable, Extensible Toolkit for Scientific Computation (PETSc) available for Linux and Windows.

Authors' contributions

FK developed and evaluated the presented model and wrote the draft as main author. VT provided many useful proposals and significantly contributed to the revision of the whole paper. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

References

1. T O'donovan, T Brown, F Büsching, A Cardoso, J Cecílio, JD Ó, P Furtado, P Gil, A Jugel, W-B Pöttner, et al., The GINSENG System for wireless monitoring and control: design and deployment experiences. ACM Trans. Sen. Netw. **10**(1), 4:1–4:40 (2013)
2. A Pfahl, M Randt, F Meier, M Zaschke, CPW Geurts, M Buselmeier, in *Proceedings of the 20th International Conference on Concentrated Solar Power and Chemical Energy Technologies (SolarPACES)*. A holistic approach for low cost heliostat fields (Elsevier, Beijing, 2014)
3. F Meier, V Turau, in *Proceedings of the 11th International Conference on the Design of Reliable Communication Networks (DRCN)*. An Analytical Model for Fast and Verifiable Assessment of Large Scale Wireless Mesh Networks (IEEE, Kansas City, 2015)
4. IEEE, IEEE 802.15.4™-2015 - IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (WPANs) (2016)
5. M Köstler, F Kauer, T Lübkert, V Turau, in *Proceedings of the 15. GI/ITG KuVS Fachgespräch Sensornetze*, ed. by J Scholz, A von Bodisco. Towards an Open Source Implementation of the IEEE 802.154 DSME Link Layer (Hochschule Augsburg, Augsburg, 2016)
6. S Duquennoy, B Al Nahas, O Landsiedel, T Watteyne, in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*. Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH (ACM, Seoul, 2015)
7. F Kauer, Analytical Model for IEEE 802.154 Mesh Networks - Implementation. GitHub Repository (2017). https://github.com/koalo/AnalyticalMultiHop. Accessed 18 Jul 2017
8. P Pathak, R Dutta, A survey of network design problems and joint design approaches in wireless mesh networks. IEEE Commun. Surv. Tutorials. **13**(3), 396–428 (2011)
9. J Zheng, MJ Lee, in *Sensor Network Operations*, ed. by S Phoha, T LaPorta, and C Griffin. A Comprehensive Performance Study of IEEE 802.15.4. (Wiley, Hoboken, 2005), pp. 218–236
10. P Dutta, J Hui, J Jeong, S Kim, C Sharp, J Taneja, G Tolle, K Whitehouse, D Culler, in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks (IPSN)*. Trio: Enabling Sustainable and Scalable Outdoor Wireless Sensor Network Deployments (IEEE, Nashville, 2006)
11. J Mišić, S Shafi, VB Mišić, Performance of a Beacon Enabled IEEE 802.15.4 Cluster with Downlink and Uplink Traffic. IEEE Trans. Parallel Distributed Syst. **17**(4), 361–376 (2006)
12. P Di Marco, P Park, C Fischione, KH Johansson, Analytical Modeling of Multi-hop IEEE 802.15.4 Networks. IEEE Trans. Vehicular Technol. **61**(7), 3191–3208 (2012)
13. G Alderisi, G Patti, O Mirabella, LL Bello, in *Proceedings of the 13th International Conference on Industrial Informatics (INDIN)*. Simulative Assessments of the IEEE 802.15.4e DSME and TSCH in Realistic Process Automation Scenarios (IEEE, Cambridge, 2015)
14. DD Guglielmo, A Seghetti, G Anastasi, M Conti, in *Proceedings of the 19th IEEE Symposium on Computers and Communications (ISCC)*. A Performance Analysis of the Network Formation Process in IEEE 802.15.4e TSCH Wireless Sensor/Actuator Networks (IEEE, Madeira, 2014)
15. P Sahoo, S Pattanaik, S-L Wu, A Novel IEEE 802.15.4e DSME MAC for Wireless Sensor Networks. Sensors. **17**(1), 168 (2017)

16. W-C Jeong, J Lee, in *Proceedings of the First IEEE Workshop on Enabling Technologies for Smartphone and Internet of Things (ETSIoT)*. Performance Evaluation of IEEE 802.15.4e DSME MAC Protocol for Wireless Sensor Networks (IEEE, Seoul, 2012)

17. J Gronkvist, J Nilsson, D Yuan, in *Proceedings of the 59th IEEE Vehicular Technology Conference (VTC)*. Throughput of Optimal Spatial Reuse TDMA for Wireless Ad-hoc Networks (IEEE, Milan, 2004)

18. P Bjorklund, P Varbrand, Di Yuan, in *Proceedings of the 22th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. Resource Optimization of Spatial TDMA in Ad Hoc Radio Networks: A Column Generation Approach (IEEE, San Francisco, 2003)

19. C-H Ng, S Boon-Hee, *Queueing Modelling Fundamentals: With Applications in Communication Networks*, 2nd edn. (Wiley, Hoboken, 2008)

20. D-W Seo, Explicit Formulae for Characteristics of Finite-Capacity M/D/1 Queues. ETRI J. **36**(4), 609–616 (2014)

21. J MacGregor Smith, Properties and Performance Modelling of Finite Buffer M/G/1/K Networks. Comput. Oper. Res. **38**(4), 740–754 (2011)

22. K Khan, H Peyravi, in *Proceedings of the 6th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS)*. Delay and Queue Size Analysis of TDMA with General Traffic (IEEE, Montreal, 1998)

23. A Dunkels, B Grönvall, T Voigt, in *Proceedings of the 29th International Conference on Local Computer Networks*. Contiki - A Lightweight and Flexible Operating System for Tiny Networked Sensors (IEEE, Tampla, 2004)

24. D Dujovne, L Grieco, M Palattella, N Accettura, 6TiSCH 6top Scheduling Function Zero (SF0), Internet Engineering Task Force (2017)

25. S Anamalamudi, M Zhang, A Sangi, C Perkins, SVR Anand, Scheduling Function One (SF1) for hop-by-hop Scheduling in 6tisch Networks, Internet Engineering Task Force (IETF) (2017)

26. IEC, IEC 62591:2010 - Industrial communication networks - Wireless communication network and communication profiles - WirelessHART™ (2010)

27. W-B Pöttner, H Seidel, J Brown, U Roedig, L Wolf, Constructing Schedules for Time-Critical Data Delivery in Wireless Sensor Networks. ACM Trans. Sen. Netw. 10(3), 44:1–44, 31 (2014)

28. MR Palattella, N Accettura, LA Grieco, G Boggia, M Dohler, T Engel, On Optimal Scheduling in Duty-Cycled Industrial IoT Applications Using IEEE802.15.4e TSCH. IEEE Sensors J. **13**(10), 3655–3666 (2013)

29. N Accettura, E Vogli, MR Palattella, LA Grieco, G Boggia, M Dohler, Decentralized Traffic Aware Scheduling in 6TiSCH Networks: Design and Experimental Evaluation. IEEE Internet Things J. **2**(6), 455–470 (2015)

30. R-H Hwang, C-C Wang, W-B Wang, A Distributed Scheduling Algorithm for IEEE 802.15.4e Wireless Sensor Networks. Comput. Stand. Interfaces. **52**(C), 63–70 (2017)

31. R Soua, P Minet, E Livolant, Wave: a Distributed Scheduling Algorithm for Convergecast in IEEE 802.15.4e TSCH Networks. Trans. Emerging Telecommun. Technol. **27**(4), 557–575 (2016)

32. R Soua, P Minet, E Livolant, Wave: a Distributed Scheduling Algorithm for Convergecast in IEEE 802.15.4e TSCH Networks (Extended Version), Research Report RR-8661, Inria (2015)

33. E Municio, S Latré, in *Proceedings of the Workshop on Mobility in the Evolving Internet Architecture (MobiArch)*. Decentralized Broadcast-based Scheduling for Dense Multi-hop TSCH Networks (ACM, New York, 2016)

34. M Ojo, S Giordano, G Portaluri, D Adami, M Pagano, in *Proceedings of the IEEE International Conference on Communications Workshops (ICC Workshops)*. An Energy Efficient Centralized Scheduling Scheme in TSCH Networks (IEEE, Paris, 2017)

35. F Kauer, M Köstler, T Lübkert, V Turau, in *Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWIM)*. Formal Analysis and Verification of the IEEE 802.154 DSME Slot Allocation (ACM, Malta, 2016)

36. G Siegemund, V Turau, C Weyer, in *Proceedings of the International Conference on Ad-hoc, Mobile and Wireless Networks (ADHOC-NOW)*. A Dynamic Topology Control Algorithm for Wireless Sensor Networks (Springer, Athens, 2015)

37. T Kuruganti, W Dykas, W Manges, T Flowers, M Hadley, P Ewing, T King, *Wireless System Considerations When Implementing NERC Critical Infrastructure Protection Standards, Oak Ridge National Laboratory, TN, USA, Flowers Control Center Solutions, Todd Mission, TX, USA*. (Pacific Northwest National Laboratory, Richland, 2009)

38. Team SimPy, SimPy - Discrete Event Simulation for Python (2017). https://simpy.readthedocs.io. Accessed 14 Jul 2017

39. F Osterlind, A Dunkels, J Eriksson, N Finne, T Voigt, in *Proceedings of the 31st International Conference on Local Computer Networks*. Cross-Level Sensor Network Simulation with COOJA (IEEE, Tampla, 2006)

40. R Alexander, A Brandt, J Vasseur, J Hui, K Pister, P Thubert, P Levis, R Struik, R Kelsey, T Winter, *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. (Internet Engineering Task Force, Fremont, 2012)

41. F Meier, V Turau, Analytical Model for IEEE 802.15.4 Multi-Hop Networks with Improved Handling of Acknowledgements and Retransmissions (2015). arXiv:1501.07594 [cs.NI], http://arxiv.org/abs/1501.07594

42. V Turau, C Weyer, in *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*. TDMA-schemes for tree-routing in data intensive wireless sensor networks (IEEE, Pisa, 2007)

43. I Juc, O Alphand, R Guizzetti, M Favre, A Duda, in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*. Energy Consumption and Performance of IEEE 802.15.4e TSCH and DSME (IEEE, Doha, 2016)

44. P Di Marco, C Fischione, F Santucci, K Johansson, in *Proceedings of the IEEE International Conference on Communications (ICC)*, vol. Effects of Rayleigh-Lognormal Fading on IEEE 802.15.4 Networks (IEEE, Budapest, 2013)

45. S Balay, WD Gropp, LC McInnes, BF Smith, in *Modern Software Tools in Scientific Computing*, ed. by E Arge, AM Bruaset, and HP Langtangen. Efficient Management of Parallelism in Object Oriented Numerical Software Libraries, (Birkhäuser, Boston, 1997), pp. 163–202

46. S Balay, S Abhyankar, MF Adams, J Brown, P Brune, K Buschelman, L Dalcin, V Eijkhout, WD Gropp, D Kaushik, *et al*, PETSc Users Manual, Technical Report ANL-95/11 - Revision 3.7 (2016)

47. IM Chakravarti, On a Characterization of Irreducibility of a Non-Negative Matrix. Linear Algebra Appl. **10**(2), 103–109 (1975)

48. WJ Stewart, *Introduction to the numerical solution of markov chains*. (Princeton University Press, Princeton, 1994)