On Modeling Low-Power Wireless Protocols Based on Synchronous Packet Transmissions

Marco Zimmerling* Federico Ferrari* Luca Mottola† Lothar Thiele*
*Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland
†Politecnico di Milano, Italy and Swedish Institute of Computer Science (SICS)
{zimmerling,ferrari,thiele}@tik.ee.ethz.ch luca.mottola@polimi.it

***Abstract*—Mathematical models play a pivotal role in understanding and designing advanced low-power wireless systems. However, the distributed and uncoordinated operation of traditional multi-hop low-power wireless protocols greatly complicates their accurate modeling. This is mainly because these protocols build and maintain substantial *network state* to cope with the dynamics of low-power wireless links. Recent protocols depart from this design by leveraging *synchronous transmissions (ST)*, whereby multiple nodes simultaneously transmit towards the same receiver, as opposed to pairwise *link-based transmissions (LT)*. ST improve the one-hop packet reliability to an extent that efficient multi-hop protocols with little network state are feasible.**

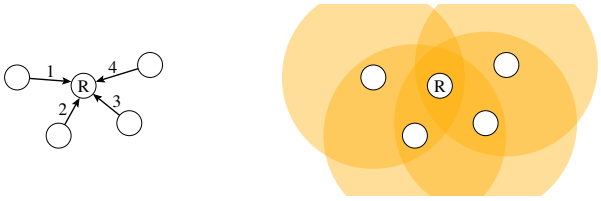
This paper studies whether ST also enable simple yet accurate modeling of these protocols. Our contribution to this end is twofold. First, we show, through experiments on a 139-node testbed, that characterizing packet receptions and losses as a sequence of independent and identically distributed (i.i.d.) Bernoulli trials—a common assumption in protocol modeling but often illegitimate for LT—is largely valid for ST. We then show how this finding simplifies the modeling of a recent ST-based protocol, by deriving (*i*) sufficient conditions for probabilistic guarantees on the end-to-end packet reliability, and (*ii*) a Markovian model to estimate the long-term energy consumption. Validation using testbed experiments confirms that our simple models are also highly accurate; for example, the model error in energy against real measurements is 0.25%, a figure never reported before in the related literature.

I. Introduction

Low-power wireless networks facilitate advanced applications that use wirelessly interconnected sensors and actuators to monitor and act on the physical world, such as environmental control, assisted living, and intelligent transportation [1]. Effectively employing low-power wireless in these applications demands thoroughly understanding the behavior of the protocols that power the network operation. For example, the ability to estimate the energy consumption is crucial to self-sustaining systems based on energy harvesting [2], and guarantees on the packet delivery are key to dependable wireless automation [3].

Unfortunately, the current literature falls short in modeling multi-hop low-power wireless protocols. Two aspects concur:

- Low-power wireless transmissions are subject to a number of unpredictable environmental factors, including wireless interference, presence of obstacles and persons, as well as temperature and humidity variations [4]. As a result, low-power wireless links suffer from *unpredictable packet loss* that varies in time and space [5]. This, in combination with failure-prone devices, for example, due to battery depletion or damage, makes the *network topology highly dynamic*.
- To tame this unpredictability, existing communication protocols gather substantial *network state*, such as link quality estimates [6] and packet queue occupancies [7]. Protocols use this information, for example, to build multi-hop routing paths [6] and to adapt packet transmission rates [7]. However, the network state needs to be *updated at runtime* against the topology dynamics. For scalability reasons, the network state is also *distributed* across the nodes, which operate *concurrently* with *little or no coordination*.



(a) Link-based transmissions (LT). (b) Synchronous transmission (ST).

Fig. 1. Using synchronous transmissions, multiple nodes transmit simultaneously to the same receiver, as opposed to pairwise link-based transmissions.

These reasons render multi-hop low-power wireless protocols intricate and difficult to model [8]. As a result, existing models often stop at the link layer, achieving model errors in the range of 2–7% [9]. Only a few attempts exist to model also higher-layer functionality [8], [10]–[12]. However, their validation is limited to numerical simulations, lacking precisely those real-world dynamics that complicate the modeling.

A new breed of communication protocols is emerging that leverages *synchronous transmissions (ST)* [13]–[20], illustrated in Fig. 1. Unlike single transmissions over sender-receiver links in (a), with ST *multiple* nodes transmit *simultaneously* towards the same receiver in (b). Because of two phenomena of low-power wireless communications, constructive interference [14] and capture effects [13], ST vastly improve the one-hop packet reliability compared with *link-based transmissions (LT)* [14].

As we further discuss in Sec. II, the salient features of ST enable multi-hop protocols that require very little network state and outperform LT-based protocols. The open question is whether ST also simplify accurately modeling these protocols. To answer this question, we put forward two key contributions:

1) We investigate in Sec. III to what extent the *Bernoulli assumption* applies to ST. The assumption stipulates that subsequent packet receptions and losses at a receiver adhere to a sequence of independent and identically distributed (i.i.d.) Bernoulli trials. Models of communication protocols often make this assumption to simplify the specification [9], [21], but prior work suggests that this may be invalid for LT [5], [22]. By contrast, nothing is known for ST, as far as we are

aware. By studying a specific flavor of ST, *Glossy* network floods [15], through experiments on a 139-node low-power wireless testbed, we show that the Bernoulli assumption is largely valid for ST, and way more than for LT.

2) We build upon these findings to demonstrate that modeling an ST-based protocol is in fact simpler and yields significantly higher accuracy than models of LT-based protocols. We do so based on *Low-Power Wireless Bus (LWB)* [18], a representative protocol, described in Sec. IV, of a growing number of solutions [16], [17], [19], [20] that build upon Glossy. Specifically, we present in Sec. V sufficient conditions for providing probabilistic guarantees on LWB's end-to-end packet reliability, and in Sec. VI a discrete-time Markov chain (DTMC) model to estimate LWB's expected long-term energy consumption. Results from our validation based on real-world experiments in Sec. VII indicate that the end-to-end reliability guarantees are correctly matched, and that the estimates of the energy model are within 0.25% of the real measurements. This error margin is unmatched in the low-power wireless literature we are aware of.

II. Background and Related Work

Our work builds upon recent advancements in low-power wireless communications. In this section, we provide the necessary background on multi-hop protocols exploiting different flavors of ST, contrast these with the existing literature on LT, and review related modeling efforts. We conclude with an outlook on how this paper fills the gaps in the current literature.

A. Synchronous Transmissions

Little work exists to deeply understand the behavior of ST in low-power wireless. For example, Son et al. [23] conduct an experimental study of the *capture effect*, a physical layer phenomenon that allows a receiver to correctly decode a packet despite interference from other transmitters. In low-power wireless, this is typically due to *power capture*, which occurs when the received signal from a node is 3 dB stronger than the sum of the signals from all other nodes [23]. Several protocols exploit the capture effect, for example, to implement fast network flooding [13] and efficient all-to-all communication [17].

Precisely overlapping transmissions of *identical* packets enable another phenomenon in low-power wireless: *constructive interference* of IEEE 802.15.4 symbols. This allows a receiver to correctly decode the packet also in the absence of capture effects, significantly boosting the transmission reliability. Using resource-constrained devices, however, the required timing accuracy of ST is difficult to achieve. One way to address this challenge is by using hardware-generated acknowledgments, a mechanism that has been employed to better resolve contention in media access control (MAC) protocols [14], [24].

Glossy, instead, uses a careful software design to make ST of the same packet precisely overlap, thus exploiting both constructive interference and capture effects for efficient network-wide flooding and time synchronization with microsecond accuracy [15]. Several protocols extend and improve Glossy, for example, in dense networks [16], when distributing large data objects [19], and for point-to-point communication [20]. LWB, which we use to examine the impact of ST on modeling multi-hop protocols, efficiently supports multiple traffic patterns by globally scheduling Glossy floods in an online fashion [18].

B. Link-Based Transmissions

By contrast, a large body of work exists on understanding the behavior of LT [25]. Srinivasan et al. [5], for example, conduct an empirical study of IEEE 802.15.4 transmissions, to provide guidelines for fine-grained design decisions such as the scheduling of link-layer packet retransmissions. The β factor [26] measures the link burstiness over time, which may be used by a protocol to determine how long to pause after a transmission failure to prevent unnecessary retransmissions. Cerpa et al. [22] examine both short- and long-term temporal aspects to improve simulation models and for enhancing point-to-point routing. Dually, the κ factor [27] measures the degree of correlation of packet receptions across different receivers—hence exploring LT's spatial diversity—possibly used to design better opportunistic routing and network coding schemes.

Despite the knowledge of LT, obtaining full-fledged models of LT-based multi-hop protocols is very difficult [8]. Several attempts stop at the MAC layer, where distributed interactions span only one hop and hence reasoning is still manageable. For example, pTunes provides runtime tuning of MAC parameters based on application-level performance goals, leveraging MAC protocol models [9]. Similarly, Polastre et al. [21] present a model of node lifetime for B-MAC, and Buettner et al. [28] model reliability and energy in X-MAC. Gribaudo et al. [11] use interacting Markovian agents to model a generic sender-initiated low-power MAC protocol [9]; they also acknowledge that the opportunistic operation of this class of protocols greatly complicates the modeling using standard techniques.

The dynamics of the network topology render the modeling of higher-layer functionality—where interactions typically extend across multiple hops—very complex. As a result, accurate models to estimate a system's performance are largely missing. Some exceptions model the Collection Tree Protocol (CTP) [6] to improve its performance in industrial scenarios [12], analyze swarm intelligence algorithms for sensor networks based on the Markovian agent model [10], apply diffusion approximation techniques to estimate the end-to-end packet travel times assuming opportunistic packet forwarding rules [29], or model generic multi-hop functionality through population continuous-time Markov chains [8]. Nevertheless, the validation of these models is limited to numerical simulations, which lack precisely those real-world dynamics of low-power wireless links that make accurate protocol modeling so complex and difficult.

C. Outlook

Motivated by the lack of a deeper understanding of ST, in the remainder of this paper we provide a thorough account on the behavior of ST and its impact on the modeling of emerging ST-based multi-hop protocols. To this end, we start by analyzing in Sec. III to what extent a key, yet sometimes illegitimate assumption in modeling low-power wireless protocols applies to ST. We base this study upon Glossy's specific incarnation of ST [15], because it serves as the communication primitive for a growing class of multi-hop protocols [16], [18]–[20]. We then apply the corresponding findings while closely examining LWB [18], one specific such protocol we illustrate in Sec. IV that exceeds the performance, reliability, and versatility of prior LT-based protocols. In doing so, we analyze end-to-end packet reliability in Sec. V and energy consumption in Sec. VI—two key performance indicators in low-power wireless [6].

III. Bernoulli Assumption in Low-Power Wireless

Because wireless networks are very complex, researchers make simplifying assumptions about their behavior when reasoning about a protocol. One such assumption is the *Bernoulli assumption* [5]. Let the reception of packets transmitted in a sequence be a random event with success or failure as the only possible outcomes: the Bernoulli assumption stipulates that the intended packet receiver observes a sequence of *independent* and *identically distributed* (i.i.d.) Bernoulli trials. In practical terms, success means a packet is received (with probability p), and failure means a packet is lost (with probability $1-p$).

However, several studies have shown that the Bernoulli assumption is not always valid in low-power wireless networks, because links have temporally correlated receptions and losses when they occur close in time (*i.e.*, on the order of a few tens of milliseconds) [5], [22]. In this section, we show empirically that the Bernoulli assumption is (*i*) in fact highly valid for ST in Glossy, and (*ii*) more appropriate in Glossy than in LT.

We first describe how we collect large sets of packet reception traces on a real-world sensor network testbed. Next, we discuss our analysis of these traces for weak stationarity, which is a necessary condition for further statistical analysis. We then construct a statistical test for packet reception independence based on the sample autocorrelation metric, and use this test to assess the validity of the Bernoulli assumption in our traces.

A. Experimental Methodology

We perform 80 hours of packet reception measurements on Indriya [30], a large testbed of 139 TelosB nodes deployed across three floors in a university building. Indriya provides a mixture of dense and sparse node clusters, as well as realistic interference from the presence of people and co-located Wi-Fi.

We conduct two types of experiments that match the building blocks of communication in ST- and LT-based protocols:

1) *ST-Type*: we select 70 nodes equally distributed across the three floors on Indriya and let them, one at a time, initiate 50,000 Glossy network floods. According to Glossy's operation, the remaining 138 nodes blindly relay the flooding packet, eventually delivering it to all nodes in the network.
2) *LT-Type*: all 139 nodes, one at a time, broadcast 50,000 packets, while the remaining nodes passively listen. As LT are bound by the transmission range of the sending node, only its one-hop neighbors can receive the packet.

In both types of experiments, the packets are 20 bytes long and carry a unique sequence number. The sender transmits at a fixed inter-packet interval (IPI) of 20 ms, corresponding to the typical minimum interval between consecutive Glossy floods in ST-based protocols [16], [18]–[20]. All other nodes record received and lost packets based on the sequence number. We use IEEE 802.15.4 channel 26 to reduce the influence of Wi-Fi interference, whose extent we cannot control and is difficult to assess afterwards [4]. We repeat both types of experiments for two transmit powers: 0 dBm is the maximum transmit power of a TelosB node, and -15 dBm is the lowest transmit power at which the network on Indriya remains connected. The resulting network diameters are 5 and 11 hops, respectively.

We represent every collected trace as a discrete-time binary time series $\{x_i\}_{i=1}^{n}$, where x_i is 1 if the i-th packet was received and 0 if it was lost. This time series representation forms the basis for our statistical analysis.

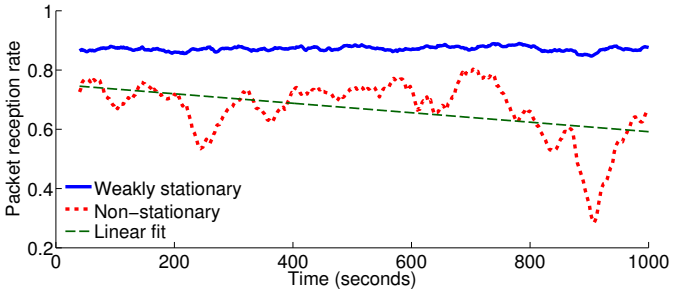


Fig. 2. Example of a weakly stationary and a non-stationarity trace. Packet reception rate is a moving average with a window size of 2,000 packets (40 s).

B. Weak Stationarity

A necessary condition for well-founded statistical analyses of time series is *weak stationarity* [31]. A weakly stationary time series has constant mean, constant variance, and the autocovariance between two values depends only on the time interval between those values. We investigate whether our traces conform to these criteria based on the *packet reception rate* (PRR), computed as a moving average of the fraction of received packets over a window of 2,000 packets (40 s).

Visual inspection of our traces reveals obvious violations of these criteria. For example, Fig. 2 shows the PRR for a stationary and a non-stationary trace. The latter has several abrupt changes and a significant trend in the mean, as evident from the linear fit. To avoid biases in our analysis, we need to identify and exclude such non-stationary traces.

TABLE I. Trace statistics

Type	Transmit power	Total	Non-stationary	Weakly stationary
ST-Type	0 dBm	9660	47	9613
ST-Type	-15 dBm	9660	256	9404
LT-Type	0 dBm	4189	1418	2771
LT-Type	-15 dBm	1777	588	1189

While there is a number of formal tests for stationarity, they often fail in practice due to their inability to detect general non-stationarity [32]. Thus, like [33], we apply two empirical tests to identify non-stationary traces. To test for trends in the mean, we compute a linear fit using ordinary least squares and declare a trace as non-stationary if the PRR changes by 0.015 or more over the entire trace of 50,000 packets (1,000 s). Then, we test for non-constant variance by checking whether the PRR decreases or rises by more than 0.05 within a window of 2,000 packets (40 s), which we interpret as an indication of non-stationarity. Table I summarizes the sets of traces before and after applying the two empirical tests for non-stationarity.

C. Validating the Bernoulli Assumption

To confirm or refute the Bernoulli assumption for a given trace, we use the *sample autocorrelation*, which measures the linear dependence between values of a weakly stationary time series as a function of the interval (lag) between those values. As we explain below, the Bernoulli assumption is valid if the values in the time series are independent already at lag 1.

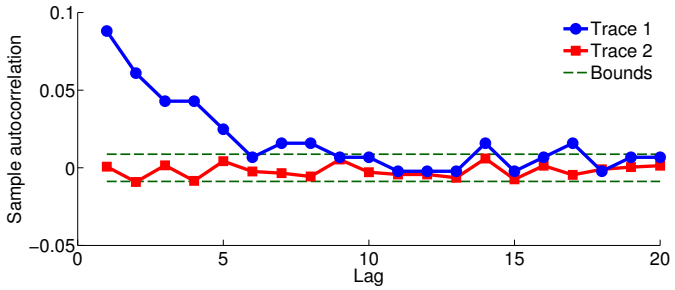


Fig. 3. Sample autocorrelation for two of our collected packet reception traces up to lag 20. The Bernoulli assumption holds for Trace 2, because its sample autocorrelation falls within the confidence bounds already at lag 1.

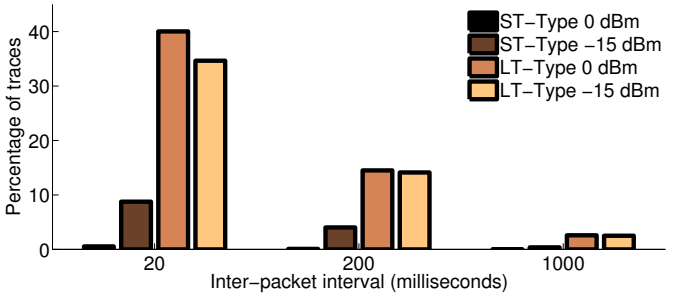


Fig. 4. Percentage of weakly stationary traces for which the Bernoulli assumption does not hold, for different IPIs and transmit powers.

For a discrete-time binary time series $\{x_i\}_{i=1}^{n}$ of length n, the sample autocorrelation $\hat{\rho}$ at lag $\tau = 1, 2, \ldots, n-1$ is

$$\hat{\rho}(\tau) = \begin{cases} \hat{\gamma}(\tau)/\hat{\gamma}(0) & \text{if } \hat{\gamma}(0) \neq 0 \\ 0 & \text{if } \hat{\gamma}(0) = 0 \end{cases} \quad (1)$$

where $\hat{\gamma}(\tau)$ is the estimated autocovariance given by

$$\hat{\gamma}(\tau) = \frac{1}{n} \sum_{i=1}^{n-\tau} (x_{i+\tau} - \overline{x})(x_i - \overline{x})$$

and $\overline{x} = 1/n \sum_{i=1}^{n} x_i$ is the sample mean.

The sample autocorrelation in (1) ranges between -1 and 1. Negative values indicate anti-correlation in packet reception: as more packets are received (lost), the next packet reception is more likely to fail (succeed). Positive values indicate positive correlation: packet receptions (losses) tend to be followed by more packet receptions (losses).

Values close to zero indicate independence among packet receptions at a given lag, assuming the x_i are i.i.d. Bernoulli random variables. Let $\{x_i\}_{i=1}^{n}$ a realization of an i.i.d. sequence $\{X_i\}_{i=1}^{\infty}$ of random variables with finite variance. It can be shown that, for a large number of samples n, about 95% of the sample autocorrelation values should lie within the confidence bounds $\pm 1.96/\sqrt{n}$ [31]. Based on this, we define the *correlation lag* as the smallest lag at which the sample autocorrelation lies within $\pm 1.96/\sqrt{n}$. Like [33], we consider the Bernoulli assumption valid if the correlation lag is 1. Formally: *Given a time series* $\{x_i\}_{i=1}^{n}$*, the Bernoulli assumption holds at the 0.05 significance level if* $|\hat{\rho}(1)| \leq 1.96/\sqrt{n}$.

For example, Fig. 3 plots $\hat{\rho}$ for two of our traces of length $n = 50,000$. The dashed lines at $\pm 1.96/\sqrt{50,000} \approx \pm 0.0088$ represent the confidence bounds. The chart shows that Trace 1 is dependent up to lag 5, but starting from lag 6 the autocorrelation values become insignificant except for a few stray points. By contrast, Trace 2 has an insignificant autocorrelation already at lag 1: there is no dependence between consecutive packets nor between any other packets in the trace. Thus, the Bernoulli assumption holds for Trace 2 but not for Trace 1.

D. Results

Based on the above reasoning, we analyze the validity of the Bernoulli assumption for our weakly stationary traces (see Table I). Fig. 4 shows the percentage of traces with correlation lag greater than 1 (for which the Bernoulli assumption does not hold) when examining different IPIs in our traces. We see that the Bernoulli assumption is significantly more legitimate for synchronous transmissions (ST-Type) than for the link-based transmissions (LT-Type). For example, at the highest transmit power, the Bernoulli assumption holds for more than 99% of the ST-Type traces irrespective of the IPI, whereas it holds only for 60% of the LT-Type traces at the smallest IPI of 20 ms.

We also observe that at the lower transmit power there are more ST-Type traces for which the Bernoulli assumption does not hold. This is mostly because nodes have fewer neighbors—at -15 dBm about 24% of the nodes have at most four well-connected neighbors—which makes their reception behavior approach the one of LT. This is also confirmed by the significant negative Pearson correlation of -0.31 between the number of well-connected neighbors and the percentage of traces for which the Bernoulli assumption does not hold. Finally, Fig. 4 shows that the autocorrelation decreases as the IPI increases, and becomes negligible at IPI $= 1$ s also for LT-Type traces. This observation is in line with prior studies on low-power wireless links [5], thus validating our methodology.

In summary, our results show that the Bernoulli assumption holds to a large extent for ST—because packet receptions in Glossy are largely independent, a *single* parameter is sufficient to precisely characterize the probability of receiving a packet. By contrast, packet receptions in LT are often not independent, which necessitates more complex models, such as high-order Markov chains [33], to accurately capture their behavior.

Next, we describe LWB, a Glossy-based protocol that we use throughout Secs. V and VI to demonstrate how the validity of the Bernoulli assumption for ST enables simple, yet highly accurate models of multi-hop low-power wireless protocols.

IV. Low-Power Wireless Bus

The basic idea behind LWB is to abstract a network's multi-hop nature by employing *only* ST for communication [18]. To this end, LWB maps *all* communication demands onto Glossy network floods [15]. Glossy always and blindly propagates every message from one node to all other nodes in the network, effectively creating the perception of a single-hop network for higher-layer protocols and applications. The resulting protocol operation of LWB is similar to a shared bus, where all nodes are potential receives of all messages; delivery to the intended recipients happens by filtering messages at the receivers.

LWB exploits Glossy's accurate time synchronization for a *time-triggered* scheme that arbitrates access to the (wireless) bus. Nodes communicate according to a global *communication schedule*. A dedicated *host* node computes the schedule online

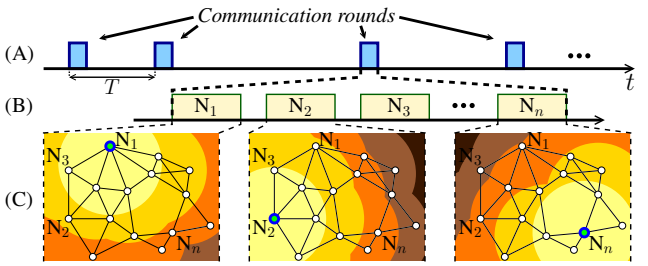


Fig. 5. LWB's time-triggered operation. Communication rounds occur with a possibly varying round period T (A); each round consists of a varying number of communication slots (B); every slot corresponds to a Glossy flood (C).

based on the current traffic demands and distributes it to the nodes, determining when a node is allowed to initiate a flood.

As shown in Fig. 5 (A), communication occurs in *rounds* that repeat with a possibly varying *round period* T. All nodes keep their radios off between two rounds to save energy. Every round consists of a possibly varying number of *communication slots*, as shown in Fig. 5 (B). In every slot, at most one node puts a message on the bus (*i.e.*, initiates a flood), while the remaining nodes read the message from the bus (*i.e.*, receive and relay the flooding packet), as illustrated in Fig. 5 (C).

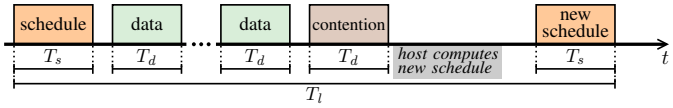


Fig. 6. Communication slots within a single LWB round.

Fig. 6 shows the different communication slots within one round of length T_l. Each round starts with a slot of length T_s in which the host distributes the communication schedule. The nodes use the schedule to time-synchronize with the host and to be informed of (*i*) the round period T and (*ii*) the mapping of source nodes to the following data slots of length T_d. A non-allocated *contention slot* of length T_d follows; nodes may contend in this slot to inform the host of their traffic demands. The host uses these to compute the schedule for the next round, which it transmits in a final slot of length T_s.

The host computes the communication schedule by determining a suitable round period T and allocating data slots to the current *streams*. A stream represents a traffic demand, characterized by a *starting time* and an *inter-packet interval (IPI)*, as LWB targets the periodic traffic pattern typical of many low-power wireless applications [34]. A node can source multiple streams and individually add or remove streams at runtime.

V. End-to-End Packet Reliability in LWB

End-to-end reliability refers to a protocol's ability to deliver packets from source to destination, perhaps over multiple hops. It is a key performance metric in low-power wireless [6], indicating the level of service provided to users. Many applications do require *probabilistic guarantees* on this metric, for example, to allow for post-processing of structural health data [35].

End-to-end reliability, however, is subject to unpredictable packet loss [5]. LT-based protocols, therefore, rely on per-hop retransmissions to achieve a certain end-to-end reliability, yet the necessary number of retransmissions depends on the ever-changing loss rates of single links. Further, LT-based protocols constantly adapt the routes in response to such changes, which renders reasoning about end-to-end guarantees very complex.

By contrast, ST-based protocols, such as LWB, often have no routes to adapt. This facilitates reasoning about end-to-end guarantees, even across multiple hops. To show this, we extend LWB with a packet retransmission scheme and derive sufficient conditions to meet given reliability guarantees. The key insight is that the proven validity of the Bernoulli assumption for ST greatly simplifies the specification of these conditions.

Packet retransmissions in LWB. We consider a typical data collection setting where the LWB host is also the sink [34]. We augment LWB with packet retransmissions by modifying the scheduling algorithm used at the host to compute the schedule for the next round. Originally, LWB allocates exactly one data slot for each data packet, regardless of the actual reception at the host [18]. In our modification, the host first checks whether in the current round it received every data packet assigned a slot. For each lost packet, it reallocates a slot in the next round, in which the source node retransmits the lost packet, provided that fewer than k_{max} slots have already been allocated for it. Then, the host allocates data slots for new packets as before.

Consider now an application that requires a minimum end-to-end reliability $\overline{p_d} > 0$ on data packets. We derive sufficient conditions on the minimum k_{max} and overall available bandwidth to provide such guarantee in a probabilistic sense.

Sufficient conditions: retransmissions. The validity of the Bernoulli assumption for Glossy-based ST allows us to consider consecutive retransmissions as independent. Thus, based on our retransmission scheme, the probability that the host receives a packet from stream x within $k_x \geq 1$ (re)transmissions is $1-(1-p_{d,x})^{k_x}$, where $p_{d,x}$ is the probability that the host receives a packet from x in one slot (*i.e.*, during one Glossy flood). To provide the desired guarantee $\overline{p_d} > 0$, we require

$$1-(1-p_{d,x})^{k_x} \geq \overline{p_d} \tag{2}$$

where $0 < \overline{p_d} < 1$ and $0 < p_{d,x} < 1$. Then, (2) holds if

$$k_x \geq \frac{\log(1-\overline{p_d})}{\log(1-p_{d,x})} \tag{3}$$

Thus, the host must allocate k_x slots to each packet of stream x to provide an end-to-end packet reliability of *at least* $\overline{p_d}$.

Because LWB nees to set an integer upper bound k_{max} on the number of data slots allocated to each packet, the reliability guarantee can only be provided if for all existing streams x

$$\lceil k_x \rceil \leq k_{max} \tag{4}$$

Example. Assume one stream with $p_{d,x} = 0.9$, and the host allocates at most $k_{max} = 2$ slots for each packet. In this case, a reliability guarantee of $\overline{p_d} = 0.99$ can be provided, because $1-(1-0.9)^2 = 0.99$. To guarantee $\overline{p_d} = 0.9999$, however, we need to increase k_{max} to $\lceil \log(1-0.9999)/\log(1-0.9) \rceil = 4$.

Sufficient conditions: bandwidth. The bandwidth available in LWB is a function of how often communication rounds unfold: the shorter the round period T, the more data slots are available, yielding increased overall bandwidth. Due to platform-specific constraints on timings and size of the schedule packet, however, at most d_{max} data slots can be allocated in a round.

The original scheduling policy minimizes energy while providing enough bandwidth to all traffic demands whenever possible. Specifically, given N streams, the host first computes

$$T_{opt} = \frac{d_{max}}{\sum_{x=1}^{N}(1/\text{IPI}_x)} \tag{5}$$

where $1/\text{IPI}_x$ is the number of data slots allocated to stream x per time unit, because without retransmissions LWB allocates exactly one slot for each packet. Then, the host obtains the new round period using $T = \lceil \max(T_{min}, \min(T_{opt}, T_{max})) \rceil$. The lower bound T_{min} ensures that T is longer than the duration of a round T_l, and the upper bound T_{max} ensures that the nodes stay time-synchronized with the host. If $T_{opt} < T_{min}$, the network is *saturated*: the maximum provided bandwidth is insufficient to support the current traffic demands. If saturation occurs, the host sets $T = T_{min}$ and informs the nodes.

On the other hand, if a packet must be transmitted $k_x \geq 1$ times to provide a guarantee on the end-to-end packet reliability, every stream x requires k_x/IPI_x data slots per time unit; and all N streams together require $\sum_{x=1}^{N}(k_x/\text{IPI}_x)$. Therefore, to account for packet (re)transmissions, we modify (5) as

$$T_{opt} = \frac{d_{max}}{\sum_{x=1}^{N}(k_x/\text{IPI}_x)} \tag{6}$$

As described above, T_{opt} cannot be smaller than the minimum round period T_{min}. Therefore, the total bandwidth is sufficient for k_x packet transmissions only if

$$\sum\nolimits_{x=1}^{N} \frac{k_x}{\text{IPI}_x} \leq \frac{d_{max}}{T_{min}} \tag{7}$$

Only if both conditions (4) and (7) are satisfied, it is guaranteed that packets are delivered with at least probability $\overline{p_d}$.

Example. Consider streams x_1 and x_2 that generate packets with $\text{IPI}_1 = 8\,\text{s}$ and $\text{IPI}_2 = 12\,\text{s}$, and deliver packets to the host with probabilities $p_{d,1} = 0.99$ and $p_{d,2} = 0.9$. The host allocates up to $k_{max} = 16$ slots per packet and up to $d_{max} = 5$ slots per round. The minimum round period is $T_{min} = 2\,\text{s}$. Can LWB guarantee an end-to-end packet reliability of $\overline{p_d} = 0.9999$ for both streams? The condition in (4) is satisfied for both streams, because the required numbers of slots $k_1 = 2$ and $k_2 = 4$ are smaller than k_{max}. The condition in (7) is satisfied as well, because the optimal round period $T_{opt} = \frac{5}{2/8+4/12} \approx 8.57\,\text{s}$ is longer than the minimum round period T_{min}. Thus, $\overline{p_d} = 0.9999$ can be guaranteed for both streams.

VI. Energy Consumption in LWB

Energy is a primary concern in low-power wireless systems, as it generally represents the key cost metric. Modeling a protocol's energy consumption is fundamental, for example, to dimension a system's power sources before deployment, or to estimate the remaining system lifetime during operation [34].

The major factor contributing to a node's energy consumption in low-power wireless is the time spent with the radio on, because the wireless transceiver typically draws several orders of magnitude more power than other components. Therefore, we use the radio on-time as a proxy for energy in the rest of the paper. The actual energy is simply obtained by multiplying the radio on-time with the transceivers' power draw when on.

Deriving a model that precisely estimates the radio on-time of a LT-based low-power wireless protocol is, however, difficult. Nodes generally experience different radio on-times depending on their position in the routing topology, and sudden route changes trigger actions that need to be coordinated across different nodes [25]. ST simplify a protocol's operation by sparing the need for routes, thus making modeling simpler.

We now demonstrate the above for LWB. This is because a single event—the reception of schedule packets—drives most of a node's actions, as illustrated in Sec. VI-A. Also, as shown in Sec. VI-B, we can derive precise radio on-times for each of the protocol's operational states. The validity of the Bernoulli assumption for ST allows us to consider consecutive schedule receptions as independent. This facilitates computing the long-term frequency of visits to these states, as described in Sec. VI-C, ultimately yielding the expected radio on-time.

A. LWB Operational States

The radio on-time of a LWB node depends only on the reception of schedules from the host, which allows a node to time-synchronize and to learn the mapping of nodes to slots for the current round. Based on this, a node knows when communication occurs and activates the radio accordingly. A node that misses a schedule in a round refrains from communicating during that round, since communication outside of the allocated slots (*e.g.*, due to inaccurate time information) may cause packet loss due to collisions with other transmissions.

Clock drift prevents a node from having perfect time information, even when it constantly receives schedule packets [36]. The synchronization error often increases when missing several schedules in a row, as the effects of clock drift accumulate over time. To compensate for these, a LWB node uses predefined *guard times* in order to turn the radio on shortly before a round is bound to begin, and increases them in discrete steps as it misses more schedules in a row. After more than $\overline{m}$ consecutive missed schedules, a node permanently keeps the radio on until it receives a schedule and time-synchronizes again.

This behavior is reflected in the finite state machine (FSM) in Fig. 7, which models the behavior of a LWB node depending on received (r) or missed ($\neg$ r) schedule packets. Table II lists the meaning of each state and the corresponding worst-case radio on-time. Key to the model is whether the schedule packet is received (or missed) at the end of the previous round or at the beginning of the current one. To differentiate the two cases, a node reaches a state labeled X_b following schedules sent at the beginning of a round, and a state labeled X_e following schedules sent at the end of a round. We now derive the radio on-times reported in Table II for every state in the FSM.

B. Radio On-Time of LWB States

A node starts in state B_e with the radio turned on until it receives a schedule at the beginning of a round ($\text{B}_\text{e} \rightarrow \text{R}_\text{b}$) and synchronizes with the host. As shown in Fig. 5, communication rounds occur every T. Therefore, a node remains in B_e for at most $T - T_l$, where T_l is the duration of a round illustrated in Fig. 6. If a node misses such schedule ($\text{B}_\text{e} \rightarrow \text{B}_\text{b}$), it keeps the radio on for T_l before it tries again to receive a schedule at the beginning of a round ($\text{B}_\text{b} \rightarrow \text{B}_\text{e}$).

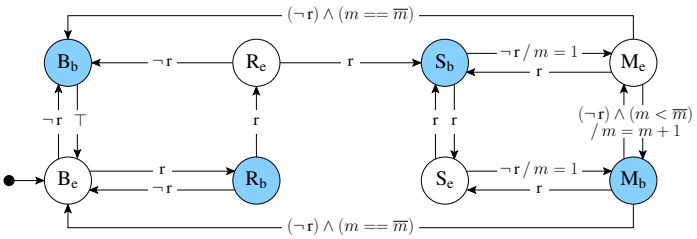


Fig. 7. FSM modeling the behavior of a LWB node depending on received (r) and missed ($\neg$ r) schedules. States X_b are reached after schedules sent at the beginning of a round, and states X_e are reached after schedules sent at the end of a round. The number of consecutive missed schedules m is updated on every transition to M_b or M_e. When m reaches a predefined threshold $\overline{m}$, a node returns to one of the two bootstrapping states B_e or B_b.

As shown in Table II, in all non-bootstrapping states the radio on-time includes the length of a schedule slot T_s and two additional terms: the guard time to compensate for synchronization errors and the radio on-time due to communication.

Guard times. A platform-specific guard time function $T_g(m)$ specifies the guard time before a schedule slot, based on the number of consecutive missed schedules m, with $0 \leq m \leq \overline{m}$. $T_g(m)$ is non-decreasing in m, since the synchronization error typically increases with more missed schedules, as discussed before. For example, if schedules are always received, a node alternates between states S_b and S_e using the smallest guard time $T_g(0)$, but switches to a longer guard time $T_g(1)$ if it misses the schedule at the beginning of the current round ($S_e \rightarrow M_b$) or at the end of the previous round ($S_b \rightarrow M_e$).

An exception to this processing occurs when a node has insufficient information to compute the drift of the local clock compared to the clock of the host. This is the case in states R_b and R_e, when a node has received only one schedule sent at the beginning of a round since the last bootstrapping state (see Fig. 7). As a result, it cannot estimate the drift of the local clock, because at least two time references are required [36]. Therefore, a node prudently uses the largest possible guard time $T_g(\overline{m})$ in state R_b or R_e, as shown in Table II.

Radio on-time due to communication. The radio on-time for data and contention slots is $T_c = (d_r + d_k)T_d$, where d_r is the expected number of data slots per round, d_k is the average number of contention slots per round, and T_d is the length of data and contention slots (see Fig. 6). As shown in Table II, T_c is accounted for only when a node participates in a round, after receiving the schedule at the beginning of the current round (state R_b or S_b) or at the end of the previous round (state M_b with $m = 1$; the Kronecker delta δ_{m1} in Table II equals 1 if $m = 1$ and 0 otherwise). We now derive d_k and d_r.

The host schedules one contention slot every $T_k > T$ to save energy under stable traffic conditions [18]. The average number of contention slots per round is thus $d_k = T/T_k$. The average number of data slots per round d_r depends on whether the network is saturated or not. If saturated, the host allocates all available d_{max} slots in every round, so $d_r = d_{max}$. Without saturation, given that each data packet can be (re)transmitted up to k_{max} times, on average $d_x \leq k_{max}$ data slots are allocated to each data packet of stream x. The expected number of data slots allocated per time unit to stream x is d_x/IPI_x. The general expression for d_r is thus

$$d_r = \min(d_{max}, T \sum_{x=1}^{N} d_x/\text{IPI}_x) \tag{8}$$

TABLE II. MEANING AND RADIO ON-TIMES OF STATES IN FIG. 7

State	Description	Worst-case radio on-time
B_b	**B**ootstrapping: not synced, radio always on	T_l
B_e	**B**ootstrapping: not synced, radio always on	$T - T_l$
R_b	**R**eceived schedule, drift not estimated	$T_g(\overline{m}) + T_s + T_c$
R_e	**R**eceived schedule, drift not estimated	$T_g(\overline{m}) + T_s$
S_b	**S**ynced: received schedule, drift estimated	$T_g(0) + T_s + T_c$
S_e	**S**ynced: received schedule, drift estimated	$T_g(0) + T_s$
M_b	**M**issed schedule at beginning of current round	$T_g(m) + T_s + \delta_{m1} T_c$
M_e	**M**issed schedule at end of previous round	$T_g(m) + T_s$

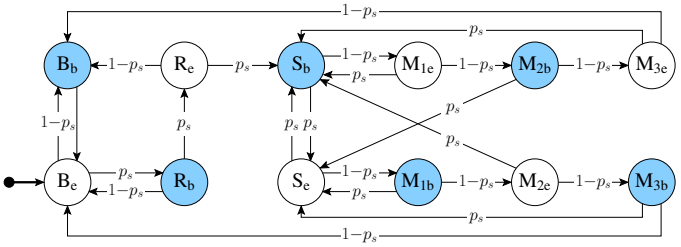


Fig. 8. DTMC corresponding to FSM in Fig. 7 if a nodes receives schedules with probability p_s and bootstraps after more than $\overline{m} = 3$ missed schedules.

As described in Sec. V, the host allocates data slots for each packet of stream x across multiple rounds until either it receives the packet or k_{max} slots are allocated. Thus, the expected number of data slots d_x allocated to a packet of stream x depends on the probability $p_{d,x}$ that the host receives from stream x and on the maximum number of transmissions k_{max}. For $0 < p_{d,x} \leq 1$, it can be shown that the host allocates

$$d_x = \frac{1 - (1 - p_{d,x})^{k_{max}}}{p_{d,x}} \tag{9}$$

slots on average to each packet of stream x. Note that for $p_{d,x} = 1$ the host receives the packet always at the first attempt and $d_x = 1$, whereas d_x approaches k_{max} as $p_{d,x}$ goes to 0.

Exploiting the Bernoulli assumption validated in Sec. III, we estimate next how frequently a LWB node is expected to visit each operational state in the long run. This information, together with the radio on-times we just derived for every state, allows us to estimate the expected radio on-time per round.

C. Expected Radio On-Time of a LWB Round

According to Sec. III, packet reception with Glossy-based ST can be modeled with high confidence as a Bernoulli trial. We thus characterize the reception of schedules at a node through a single parameter p_s, the success probability of Glossy-based ST from the host. As a result, the FSM in Fig. 7 translates into a discrete-time Markov chain (DTMC) where an event r ($\neg$r), corresponding to a successful (failed) reception of the schedule, occurs with probability p_s ($1-p_s$).

Our LWB implementation retains the original setting for $\overline{m}$, in which a node returns to bootstrapping after missing more than $\overline{m} = 3$ consecutive schedules. Fig. 8 shows the corresponding DTMC. States $\{M_{1b}, M_{2b}, M_{3b}\}$ are equivalent to state M_b in the original FSM for $m = \{1, 2, 3\}$; the same applies to states $\{M_{1e}, M_{2e}, M_{3e}\}$ and state M_e. Note that the DTMC in Fig. 8 is periodic with period 2: the host sends schedules at the beginning and at the end of a round, thus a node always visits state X_e after state X_b and vice versa.

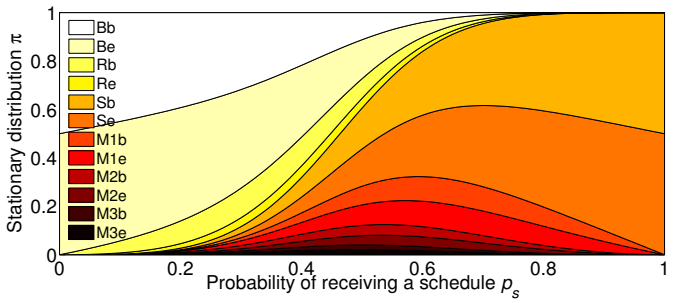


Fig. 9. Stationary distribution $\boldsymbol{\pi}$ of the DTMC in Fig. 8 against the probability of receiving a schedule p_s.

Knowing the radio on-time for each LWB state as per Table II, we can compute the expected radio-on time per round

$$T_{on} = 2(\mathbf{t_{on}} \cdot \boldsymbol{\pi}) \tag{10}$$

where $\mathbf{t_{on}} = (t_{on\mathrm{B_b}}, t_{on\mathrm{B_e}}, \ldots, t_{on\mathrm{M_{3e}}})$ is a vector containing the radio on-times of all DTMC states, $\cdot$ is the dot product, and $\boldsymbol{\pi}$ is the DTMC's stationary distribution. The factor 2 is because, during a round, a node always visits two states of the DTMC as the host transmits two schedules per round.

We can obtain $\boldsymbol{\pi}$ by determining the normalized left eigenvector with eigenvalue 1 of the DTMC's transition matrix. For $\overline{m} = 3$, we have $\boldsymbol{\pi} = (\pi_{\mathrm{B_b}}, \pi_{\mathrm{B_e}}, \ldots, \pi_{\mathrm{M_{3e}}})$, where π_s denotes the long-run frequency of visits to state $s \in \{\mathrm{B_b}, \mathrm{B_e}, \ldots, \mathrm{M_{3e}}\}$. Fig. 9 plots the stationary distribution $\boldsymbol{\pi}$ against actual values for the probability p_s. For example, we can see that when p_s approaches 1, a node visits more often states $\mathrm{S_b}$ and $\mathrm{S_e}$, as it consistently receives schedule packets, whereas it visits more often states $\mathrm{B_b}$ and $\mathrm{B_e}$ when p_s is close to 0. Typical values of p_s, however, range above 0.99 with Glossy [15], making it very unlikely that a node ever returns to a bootstrapping state.

The DTMC in Fig. 8 and the expression in (10) confirm our hypothesis that ST simplify the modeling of multi-hop protocols. This is due (*i*) the validity of the Bernoulli assumption for Glossy-based ST, and (*ii*) the absence of routes in LWB. In the following, we demonstrate that the resulting energy model is not only simple but also highly accurate.

VII. Validation

To verify accuracy and practical applicability, we compare the output of our models with real measurements. Prior to deployment, analyzing the sufficient conditions for a given end-to-end reliability against foreseeable network conditions can, for example, drive the node placement to increase connectivity; moreover, exercising the energy model for different network traffic settings can help designers dimension the power sources. At run-time, monitoring the expected network performance allows system operators to proactively perform maintenance tasks (*e.g.*, replacing nodes or batteries), and to adjust system parameters in response to changes in the network conditions. Our validation mimics these applications of our LWB models.

A. Settings and Metrics

We use the FlockLab testbed, which consists of 30 TelosB nodes deployed both inside and outside a university building [37]. We use the highest transmit power of 0 dBm, yielding a network diameter of 4 hops. To reduce sources of packet loss we cannot control, we use IEEE 802.15.4 channel 26 to minimize interference from co-located Wi-Fi networks. Instead, we artificially induce packet loss during ad-hoc experiments. In all experiments, data packets carry a payload of 15 bytes.

We extend the original LWB implementation with packet retransmissions, as described in Sec. V, and measure: (*i*) the *end-to-end reliability*, the fraction of generated data packets successfully delivered at the host; and (*ii*) the *radio on-time per round*. We measure the former based on packet sequence numbers received at the host and the latter using established software-based methods [38]. Unless otherwise stated, we set the maximum number of transmissions per packet k_{max} to 50.

Most of our model's inputs are implementation constants: the guard times $T_g(\{0,1,2,3\}) = \{1,3,5,20\}$ ms, the lengths of schedule and data slots T_s = 15 ms and T_d = 10 ms, the length of a round $T_l = 1$ s, and the maximum number of data slots per round $d_{max} = 45$. However, a few inputs are precisely known only at run-time:

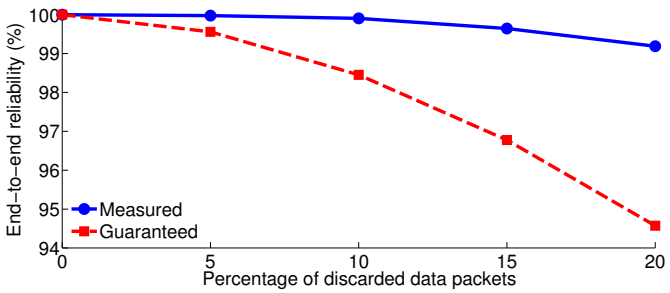
- The probability of receiving a schedule p_s. A node n estimates $p_{s,n}$ locally based on past schedule receptions, and reports it to the host by piggybacking it on data packets.
- The round period T and the expected number of data slots per round d_r. Both are determined by the scheduling policy, and depend on the streams' IPIs and the probability $p_{d,x}$ that the host receives a packet from stream x. The host estimates $p_{d,x}$ much like a node estimates $p_{s,n}$.
- The radio on-times during schedule and data slots $T_{s,n}$ and $T_{d,n}$, which are generally shorter than the lengths of schedule and data slots T_s and T_d, because nodes may turn off the radio before the end of a slot depending on their position in the network [18]. Each node n estimates $T_{s,n}$ and $T_{d,n}$ locally, and reports them to the host as it does for the probability p_s of receiving a schedule.

When using our models offline, one can consider conservative values for $p_{s,n}$, $p_{d,x}$, $T_{s,n}$, and $T_{d,n}$ based on coarse-grained deployment information or data from exploratory experiments [25]. From these and the expected streams' IPIs known from the application design, one can determine T and d_r based on the scheduling policy. During system operation, one can refine the values using up-to-date run-time estimates. Specifically, in our experiments, nodes maintain two counters to estimate $p_{s,n}$: the number of received r_s and the number of expected e_s schedule packets. Nodes embed r_s/e_s into data packets and halve both counters whenever e_s reaches a threshold, which behaves similarly to an exponentially weighted moving average (EWMA). The nodes estimate $T_{s,n}$ and $T_{d,n}$ in a similar way, and so does the host to estimate $p_{d,x}$.

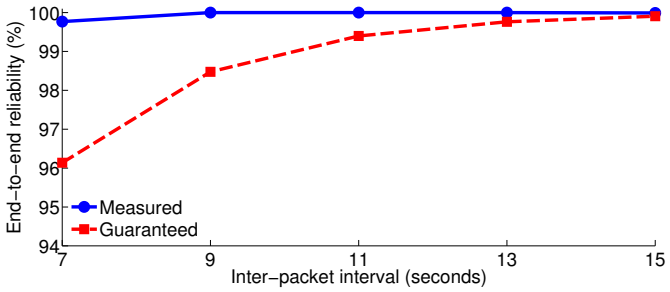
We run dedicated experiments on FlockLab to assess the accuracy of our intentionally-simple parameter estimations. To test different network conditions, three nodes at the edge of the testbed randomly discard up to 50% of schedule packets. Such high loss rates are extremely unlikely, as Glossy typically delivers more than 99% of packets [15]. Nevertheless, we find that our parameter estimates are accurate to within less than 1% across all settings, including the case of 50% missed schedules.

B. End-to-End Packet Reliability

We study how significant network unreliability and changes in traffic load may affect the guarantee on end-to-end packet



(a) Varying percentage of artificially discarded data packets at the host.



(b) Varying inter-packet interval (IPI) of generated data packets at the nodes.

Fig. 10. Measured end-to-end reliability and maximum end-to-end reliability that can be guaranteed according to the analysis of Sec. V. The gap between the two curves indicates how worse the system may possibly perform.

reliability. To test the former, we let 29 nodes generate packets with IPI $= 7$ s, while the host discards between 0% and 20% of the received data packets to emulate network unreliability. The round period is $T = 6$ s. To analyze changes in traffic load, all nodes generate packets with an increasing IPI in different runs: from 7 s to 15 s in steps of 2 s, while the host discards 5% of the data packets. The round period is set to $T = 10$ s. Both settings mirror conditions found in real deployments [34]. The experiments take 1.5 hours, and the maximum number of transmissions per packet is $k_{max} = 3$.

Fig. 10 shows for both experiments the measured end-to-end reliability and the maximum end-to-end reliability that can be guaranteed according to our analysis in Sec. V. As for the measured values, the slight drop in Fig. 10(a) is because the maximum number of transmissions $k_{max} = 3$ is insufficient to deliver all packets to the host, whereas the slight drop in Fig. 10(b) at the smallest IPI is due to insufficient bandwidth.

While the results confirm that our specific LWB executions never provide an end-to-end packet reliability lower than the guaranteed values, the figure also shows that the latter drop more severely than the measured ones. This is expected, as the analysis for guarantees on end-to-end packet reliability entails over-provisioning the number of data slots allocated to a packet, although often only a small fraction thereof is actually needed to receive it. In the experiments where we emulate network unreliability, for example, we compute from (7) that the host can allocate at most $k_x = 1.81$ slots for a packet of each stream x. Based on (2), this value guarantees an end-to-end reliability of $\overline{p_d} = 94.57\%$ when 20% of data packets are discarded, as shown in Fig. 10(a). However, according to (9) packets are actually received after $d_x = 1.24$ slots on average.

The conclusion is that the gap between the real executions and the guaranteed values provides information *in advance* about how worse the system may possibly perform. This allows system operators to take appropriate countermeasures before the problems actually manifest in the measurements, to effectively satisfy the application requirements at all times.

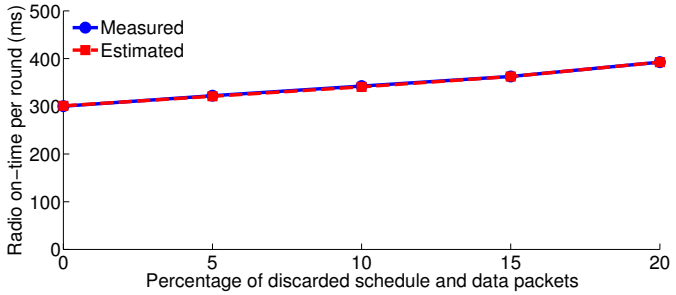


Fig. 11. Estimated and measured radio on-time per round when artificially discarding schedule and data packets. The average model error is 0.25%.

C. Energy Consumption

We evaluate the accuracy of our energy model in 5-hour experiments: 29 nodes generate packets with IPI $= 6$ s, and the round period is $T = 6$ s. In our model, the probabilities of receiving schedule and data packets $p_{s,n}$ and $p_{d,x}$ are the most critical inputs. To test the model output against different probability values, we let three nodes at the edge of the testbed randomly discard between 0% and 20% of schedules, while the host also discards the same percentage of data packets. Exercising the energy model against different IPIs and round periods T simply scales the model output proportionally.

Fig. 11 plots the estimated and measured radio on-time averaged over the three nodes. Overall, the results show that our energy model is highly accurate, with an average relative error of 0.25%. In comparison, recent work on modeling LT-based multi-hop protocols reports relative errors in energy consumption between 2 and 7% [9]—one order of magnitude larger than ours. Considering also that our work spans a *complete* multi-hop protocol rather than individual components, as discussed in Sec. II, this confirms our initial hypothesis that ST enable highly accurate protocol modeling.

We maintain that this is mainly due to the accuracy of the parameter estimation, as discussed in Sec. VII-A, and to the validity of the DTMC model. To verify the latter, we additionally run a 3-hour experiment with three nodes at the edge of the testbed discarding 50% of schedule packets. Using FlockLab's tracing facility [37], we precisely measure the fractions of time these nodes spend in each of the twelve DTMC states shown in Fig. 8. Fig. 12 shows these measurements next to what the DTMC model predicts for $p_s = 0.5$; for better visibility, we merge the corresponding states at the beginning and at the

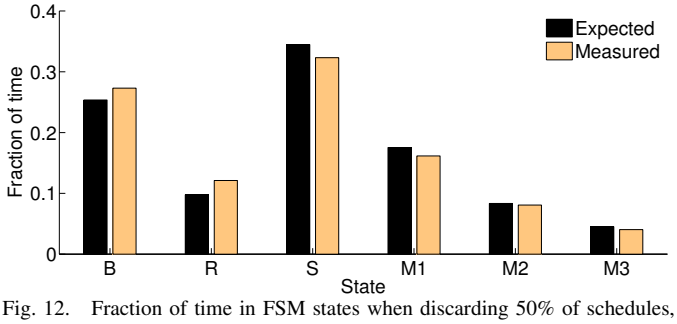


Fig. 12. Fraction of time in FSM states when discarding 50% of schedules, measured on three nodes and predicted by the DTMC model. For illustration the corresponding states at the beginning and at the end of a round are merged.

end of a round into single states, leaving six instead of twelve states in the plot. We see that expectations and measurements indeed match very well, and would do so even better for longer experiments as the long-term behavior of the system emerges.

VIII. Conclusions

We study whether ST enable simple yet accurate modeling of multi-hop low-power wireless protocols. Our experimental results show that the Bernoulli assumption is highly valid for ST in Glossy and more legitimate for ST than LT. We exploit these findings in the modeling of LWB's end-to-end reliability and long-term energy consumption. Our validation using real-world experiments confirms that accurate models of ST-based protocols are feasible, demonstrating a model error in energy of 0.25%. We believe our contributions represent a key stepping stone in the development and analysis of ST-based protocols.

Acknowledgments. We thank Olaf Landsiedel, Olga Saukh, and the anonymous reviewers for their helpful comments. This work was supported by Nano-Tera, NCCR-MICS under grant #5005-67322, the Swedish Foundation for Strategic Research, and programme IDEAS-ERC, project EU-227977 SMScom.

References

[1] E. A. Lee, "Cyber physical systems: Design challenges," in *Proc. IEEE Int. Symp. on Object Oriented Real-Time Distributed Computing (ISORC)*, 2008.

[2] C. Moser, L. Thiele, D. Brunelli, and L. Benini, "Adaptive power management for environmentally powered systems," *IEEE Trans. Comput.*, vol. 59, no. 4, 2010.

[3] K. P. Birman and T. A. Joseph, "Reliable communication in the presence of failures," *ACM Trans. Comput. Syst.*, vol. 5, no. 1, 1987.

[4] F. Hermans, O. Rensfelt, T. Voigt, E. Ngai, L.-Å. Nordén, and P. Gunningberg, "SoNIC: Classifying interference in 802.15.4 sensor networks," in *Proc. ACM/IEEE Int. Conf. on Information Processing in Sensor Networks (IPSN)*, 2013.

[5] K. Srinivasan, P. Dutta, A. Tavakoli, and P. Levis, "An empirical study of low-power wireless," *ACM Trans. Sens. Netw.*, vol. 6, no. 2, 2010.

[6] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection tree protocol," in *Proc. ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, 2009.

[7] S. Rangwala, R. Gummadi, R. Govindan, and K. Psounis, "Interference-aware fair rate control in wireless sensor networks," in *Proc. ACM SIGCOMM*, 2006.

[8] M. C. Guenther and J. T. Bradley, "PCTMC models of wireless sensor network protocols," in *Proc. UK Performance Engineering Workshop (UKPEW)*, 2012.

[9] M. Zimmerling, F. Ferrari, L. Mottola, T. Voigt, and L. Thiele, "pTunes: Runtime parameter adaptation for low-power MAC protocols," in *Proc. ACM/IEEE Int. Conf. on Information Processing in Sensor Networks (IPSN)*, 2012.

[10] D. Bruneo, M. Scarpa, A. Bobbio, D. Cerotti, and M. Gribaudo, "Markovian agent modeling swarm intelligence algorithms in wireless sensor networks," *Performance Evaluation*, vol. 69, no. 3–4, 2012.

[11] M. Gribaudo, D. Cerotti, and A. Bobbio, "Analysis of on-off policies in sensor networks using interacting Markovian agents," in *Proc. IEEE Int. Conf. on Pervasive Computing and Communications (PerCom)*, 2008.

[12] W. Yi-Zhi, Q. Dong-Ping, and H. Han-guang, "Pareto optimal collection tree protocol for industrial monitoring WSNs," in *Proc. IEEE GLOBECOM Workshops*, 2011.

[13] J. Lu and K. Whitehouse, "Flash flooding: Exploiting the capture effect for rapid flooding in wireless sensor networks," in *Proc. IEEE INFOCOM*, 2009.

[14] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, and A. Terzis, "Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless," in *Proc. ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, 2010.

[15] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with Glossy," in *Proc. ACM/IEEE Int. Conf. on Information Processing in Sensor Networks (IPSN)*, 2011.

[16] Y. Wang, Y. He, X. Mao, Y. Liu, Z. Huang, and X. Y. Li, "Exploiting constructive interference for scalable flooding in wireless networks," in *Proc. IEEE INFOCOM*, 2012.

[17] O. Landsiedel, F. Ferrari, and M. Zimmerling, "Capture effect based communication primitives: Closing the loop in wireless cyber-physical systems," in *Proc. ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, 2012.

[18] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele, "Low-power wireless bus," in *Proc. ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, 2012.

[19] M. Doddavenkatappa, M. C. Chan, and B. Leong, "Splash: Fast data dissemination with constructive interference in wireless sensor networks," in *Proc. USENIX Symp. on Networked Systems Design and Implementation (NSDI)*, 2013.

[20] D. Carlson, M. Chang, Y. Chen, A. Terzis, and O. Gnawali, "Forwarder selection in multi-transmitter networks," in *Proc. IEEE Int. Conf. on Distributed Computing in Sensor Systems (DCOSS)*, 2013.

[21] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, 2004.

[22] A. Cerpa, J. L. Wong, M. Potkonjak, and D. Estrin, "Temporal properties of low power wireless links: Modeling and implications on multi-hop routing," in *Proc. ACM Int. Symp. on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2005.

[23] D. Son, B. Krishnamachari, and J. Heidemann, "Experimental study of concurrent transmission in wireless sensor networks," in *Proc. ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, 2006.

[24] D. Carlson and A. Terzis, "Flip-MAC: A density-adaptive contention-reduction protocol for efficient any-to-one communication," in *Proc. IEEE Int. Conf. on Distributed Computing in Sensor Systems (DCOSS)*, 2011.

[25] N. Baccour, A. Koubâa, L. Mottola, M. A. Zúñiga, H. Youssef, C. A. Boano, and M. Alves, "Radio link quality estimation in wireless sensor networks: a survey," *ACM Trans. Sens. Netw.*, vol. 8, no. 4, 2012.

[26] K. Srinivasan, A. Kazandjieva, S. Agarwal, and P. Levis, "The β-factor: measuring wireless link burstiness," in *Proc. ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, 2008.

[27] K. Srinivasan, M. Jain, J. I. Choi, T. Azim, E. S. Kim, P. Levis, and B. Krishnamachari, "The κ factor: Inferring protocol performance using inter-link reception correlation," in *Proc. ACM Int. Conf. on Mobile Computing and Networking (MobiCom)*, 2010.

[28] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proc. ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, 2006.

[29] E. Gelenbe, "A diffusion model for packet travel time in a random multihop medium," *ACM Trans. on Sens. Netw.*, vol. 3, no. 2, 2007.

[30] M. Doddavenkatappa, M. Chan, and A. Ananda, "Indriya: A low-cost, 3D wireless sensor network testbed," in *Proc. ICST Int. Conf. on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom)*, 2011.

[31] P. J. Brockwell and R. A. Davis, *Time Series: Theory and Methods*. Springer, 1991.

[32] M. Dettling, "Applied time series analysis," 2013. [Online]. Available: http://bit.ly/12v93uh

[33] M. Yajnik, S. Moon, J. Kurose, and D. Towsley, "Measurement and modeling of the temporal dependence in packet loss," in *Proc. IEEE INFOCOM*, 1999.

[34] E. Gaura, L. Girod, J. Brusey, M. Allen, and G. Challen, Eds., *Wireless Sensor Networks: Deployments and Design Frameworks*. Springer, 2010.

[35] K. Chintalapudi, T. Fu, J. Paek, N. Kothari, S. Rangwala, J. Caffrey, R. Govindan, E. Johnson, and S. Masri, "Monitoring civil structures with a wireless sensor network," *IEEE Internet Comput.*, vol. 10, no. 2, 2006.

[36] C. Lenzen, P. Sommer, and R. Wattenhofer, "Optimal clock synchronization in networks," in *Proc. ACM Conf. on Embedded Networked Sensor Systems (SenSys)*, 2009.

[37] R. Lim, F. Ferrari, M. Zimmerling, C. Walser, P. Sommer, and J. Beutel, "FlockLab: A testbed for distributed, synchronized tracing and profiling of wireless embedded systems," in *Proc. ACM/IEEE Int. Conf. on Information Processing in Sensor Networks (IPSN)*, 2013.

[38] A. Dunkels, F. Österlind, N. Tsiftes, and Z. He, "Software-based on-line energy estimation for sensor nodes," in *Proc. Workshop on Embedded Networked Sensors (EmNets)*, 2007.