

OSF: An Open-Source Framework for Synchronous Flooding over Multiple Physical Layers

Michael Baddeley*, Yevgen Gyl†, Markus Schuß‡, Xiaoyuan Ma‡, and Carlo Alberto Boano§

*Technology Innovation Institute, AE; †Unikie, FI; ‡SKF Group, CN; §Graz University of Technology, AT

michael.baddeley@tii.ae, yevgen.gyl@unikie.com,
ma.xiaoyuan.mail@gmail.com, {markus.schuss; cboano}@tugraz.at

Abstract

Flooding protocols based on concurrent transmissions are regarded as the most reliable way to collect or disseminate data across a multi-hop low-power wireless mesh network. Recent works have shown that such protocols are effective for narrowband communication not only over IEEE 802.15.4, but also over the BLE 5 physical layers (PHYs). However, to date, existing literature has only built synchronous flooding solutions on top of a *single* PHY, and there has been no attempt to leverage *different* PHYs *at runtime* to increase performance. This paper fills this gap and presents OSF, an open-source framework that enables the design of multi-PHY synchronous flooding solutions thanks to a novel radio driver and middleware architecture capable of dynamically switching the underlying physical layer. This allows exploitation of the specific benefits of each PHY (e.g., higher data-rate, increased robustness) on-demand during each flood, increasing performance. We tailor OSF to the off-the-shelf nRF52840 platform, and showcase its benefits by comparing single-PHY and multi-PHY synchronous flooding solutions on a real-world testbed.

Keywords

BLE, Concurrent transmissions, Energy efficiency, IEEE 802.15.4, nRF52840, Reliability, Synchronous flooding.

1 Introduction

Interest in synchronous flooding (SF) solutions for low-power wireless networks has gained considerable interest since the observation that collisions in the presence of concurrent transmissions (CT) are not necessarily harmful [1]. The research community has especially showcased the benefits of SF for narrowband communication, with a large number of data collection and dissemination schemes developed on top of IEEE 802.15.4 devices: these have been empirically shown to outperform conventional multi-hop routing approaches across a number of key dependability metrics [2, 3].

With the increasing number of off-the-shelf hardware platforms offering support for multiple narrowband PHYs¹, researchers have shown that SF is also effective over the four physical layers supported by BLE 5 [4], but emphasized that the PHY selection has a strong impact on the network performance, especially in the presence of RF interference [5]. For example, the BLE 5 PHY operating at 2 Mbps (BLE 5 2M) PHY provides higher throughput, but has reduced range and struggles in dense networks due to the lower effectiveness of the capture effect (one needs a considerable power difference to capture the channel with different data). In contrast, the BLE 5 PHY operating at 125 kbps (BLE 5 125K) sacrifices throughput for a longer communication range, but struggles in the presence of RF interference (the longer air-time of packets results in a higher probability of being hit).

Harnessing the strengths of a PHY at runtime. As a result, the PHY to be used needs to be carefully chosen and tailored to the specific application scenario at hand. This is true under the assumption that the physical layer is *statically* selected at compile time and cannot be changed at runtime – which is currently the state-of-affairs for all known SF solutions. In fact, to date, there has been no attempt to dynamically switch PHY at runtime to leverage on their strengths and increase the reliability and efficiency of communications. We argue that one can enable significant performance gains by changing the PHY configuration in individual flooding rounds on demand.

Our contributions. We hence present Open Synchronous Flooding (OSF), a multi-PHY SF framework for the Nordic Semiconductor nRF52840 platform. In contrast to other public SF middlewares and architectures such as A² [6], Baloo [8], and BlueFlood [4], which utilize polling-based radio drivers, OSF implements a novel interrupt-based driver. This allows the complex and time-critical elements of CT-based operations to be handled with relative ease, and provides a robust platform for SF protocol implementation. More importantly, the middleware architecture of OSF enables unprecedented configuration of the flooding rounds: it enables not only fine-tuning of many SF variables (such as the number of transmissions, the maximum number of hops, and the flooding primitive), but also allows the underlying PHY to be switched *at runtime*. OSF in fact allows a dynamic selection of the phys-

¹Examples are the Texas Instruments CC2652R / CC1352 and the Nordic Semiconductor nRF52 / nRF53 families of low-power wireless devices, which support both IEEE 802.15.4 and the four BLE 5 physical layer modes.

Table 1: Comparison of OSF with other SF middlewares. OSF supports the development of multi-PHY protocols – allowing exploitation of multiple PHYs *in the same protocol*, as well as dynamic selection and (re)configuration of the PHY *at runtime*.

Framework	Public	Platform(s)	PHYs	Multi-PHY	Driver	Primitives	Protocols
A ² [6]	✓	TelosB	IEEE		Polling	Glossy	2PC, 3PC
Atomic [7, 5]		TelosB, nRF52840	IEEE, BLE		ISR	RoF	STA
Baloo [8]	✓	TelosB, CC430	IEEE		Polling	Glossy	Chaos, Crystal, LWB, Sleeping Beauty
Baloo-nRF52840 [9]	✓	nRF52840	IEEE, BLE		Polling	Glossy	
BlueFlood [4]	✓	nRF52840	IEEE, BLE		Polling	Glossy	
OSF	✓	nRF52840	IEEE, BLE	✓	ISR	Glossy, RoF	STA, Crystal

ical layers supported by the nRF52840 (e.g., IEEE 802.15.4 or any of the four BLE 5 PHYs), and therefore supports the implementation of multi-PHY SF protocols able to leverage previous insights on the PHY impact on CT performance [5]. We make OSF publicly available² as the first SF middleware to support multi-PHY experimentation. We run OSF across data dissemination and collection scenarios on the D-Cube testbed [3]: our results not only show significant performance improvement under RF interference in comparison to recent studies, but we are the first to benchmark SF for large (118 and 248 byte) data packets. Furthermore, we address recent findings on the weaknesses of BLE 5 uncoded PHYs [5, 10], where a large number of transmitters can result in a high percentage of collisions, and introduce mechanisms capable of achieving up to 100% reliability. Finally, we demonstrate how a multi-PHY approach can achieve reliability of the coded PHYs and the latency and energy benefits of the higher data-rate PHYs, delivering near-perfect reliability with energy gains of up to 40%. We further implement a reactive multi-PHY protocol capable of dynamically adjusting PHY utilization across a time-variant interference scenario, providing $\approx 20\%$ energy gains without sacrificing reliability.

Paper outline. We provide background information on how PHY aspects affect SF performance in Sect. 2, along with an overview of relevant works in this area. Sect. 3 details architectural aspects of OSF, and highlights a number of novel mechanisms designed to improve SF performance. In Sect. 4, we provide an experimental characterization of OSF’s performance across various scenarios, and demonstrate the potential behind multi-PHY solutions w.r.t. the reliability and energy efficiency of communications. Finally, we conclude the paper in Sect. 5 by providing a discussion on how this work may be taken forward in future research.

2 Background and Related Work

Enduring popularity of SF. Synchronous flooding has been popular in the context of low-power wireless systems since the development of Glossy [11]. By exploiting constructive interference and the capture effect, this technique can indeed provide highly-reliable flooding for one-to-many communication, and can be also used to achieve fast all-to-all data sharing as shown by Chaos [12]. Works such as Codecast [13] and Mixer [14] provide a more general approach to support many-to-many communication by introducing feedback-driven network coding. Especially when combined with channel hopping mechanisms, SF can outperform conventional routing-based solutions in terms of reliability, end-to-end latency, and

energy consumption, even in the presence of harsh RF interference [15, 16], as highlighted in the context of the EWSN dependability competition series [2, 3].

Middleware architectures. A number of works regard a CT-based flooding round (e.g., Glossy, Chaos, etc.) as primitives atop which SF protocols are built (and are comprehensively covered in a recent survey [1]). These provide an abstraction which separates the tight-timing-required by CT from in-network information processing. In this way, with a constant number of CPU cycles between packet receiving and transmitting, any in-network processing operations (e.g., Paxos [17]) can be applied. More recently, SF has been integrated into routing-based multi-hop protocols: T-RPL [18] introduces SF to asynchronous CSMA/CA IEEE 802.15.4 networks, improving the reliability of the downward traffic in the IETF RPL [19], while 6TiSCH++ [20] combines SF and the timeslot-based protocol [21], providing an encapsulation of a CT flood within an IEEE 802.15.4 TSCH slot frame.

However, CT operations rely on low-level control of timers and radio events. As such, SF protocols are notoriously difficult to implement. To address this challenge, Baloo [8] introduces a middleware layer providing a well-defined interface for network protocol design; specifically focusing on enabling runtime control from the network layer protocol logic over the execution of the underlying SF primitives. OSF goes considerably further – not only supporting SF primitive switching, but allowing full runtime configuration of round variables (e.g., number of transmissions, transmission power) and even the underlying radio PHY. We compare OSF against other SF middlewares and architectures in Tab. 1.

Impact of the physical layer. A majority of SF research has targeted IEEE 802.15.4 devices. Recent studies, however, have verified feasibility over other technologies such as ultra-wideband [22], LoRa [23, 24, 25], and BLE [5, 26, 10]. Al Nahas et al. [26] experimentally demonstrated feasibility over BLE 5 PHYs and proposed BlueFlood, which employs a Glossy-based CT primitive over both IEEE 802.15.4 and BLE. Baddeley et al. [5] perform large scale investigation of SF over IEEE 802.15.4 and the four BLE 5 PHYs in the presence of radio interference, while also examining how the CT-induced beating effect (sinusoidal patterns of constructive and destructive interference across the packet) can have significant impact on performance depending on the chosen PHY, and providing insights on how this may be countered.

3 OSF: Architecture and Implementation

To the best of our knowledge, OSF is the first SF middleware supporting multi-PHY protocols.

²<https://github.com/open-sf/osf.git> (will be populated after notification)

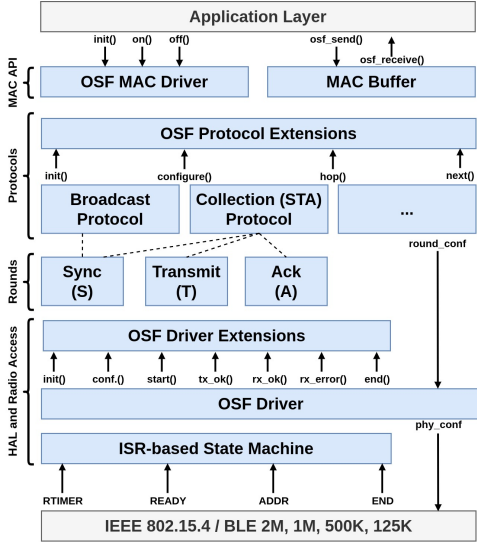


Figure 1: Overview of the OSF stack.

OSF stack. OSF employs a middleware approach inspired by recent community efforts, where simple broadcast protocols such as Glossy [11] (which operates in a reception-triggered manner, i.e., *Rx-Tx-Rx-Tx*) and RoF [15] (which operates in a time-triggered manner, i.e., *Rx-Tx-Tx-Tx*) are considered primitives upon which more complex protocols can be built. Fig. 1 provides an overview of the OSF stack. Like other SF middlewares such as Baloo [8] and Atomic [7], OSF introduces *rounds* as building blocks of network protocol logic. However, OSF takes this approach significantly further – not only allowing *runtime configuration* of flooding round parameters (e.g., number of transmissions, transmission power, and maximum number of hops), but also the underlying PHY configuration as handled by the radio HAL. Finally, as a platform for SF experimentation, OSF introduces a *driver extension* API allowing easy implementation of low-level SF enhancements (such as the RNTX mechanism introduced in Sect. 4) without prior knowledge of the underlying radio access.

Interrupt-based driver and slot management. Unlike many recent SF driver implementations, OSF uses event-triggered timer and radio interrupts generated by the modern Nordic Semiconductor nRF52840 low-power wireless chipset to exercise fine-grained control over a CT-based flood. Fig. 2 highlights how, after a $40\mu\text{s}$ radio rampup (RRU), the ADDRESS, READY, and END events are used to manage the ISR-based state machine – where actions such as channel hopping (HOP - approx. halfway between ADDRESS and END) and edge-case errors such as an END MISS can be handled through timer interrupts triggered after these events. Specifically, OSF’s interrupt-based approach means that the time-critical aspects of SF protocol development are automatically handled by the interrupt service routine (ISR), and can be ‘ignored’ while performing other tasks, while a polling-based approach introduces blocking. This has significant implications when considering USB communication to a host device: rather than scheduling a dedicated host communication time period before or after the flooding, OSF checks for late timer events and will automatically reschedule itself to account for delays, allowing OSF to run unobtrusively alongside other processes.

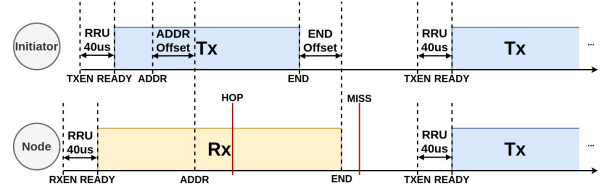


Figure 2: Interrupt-based slot management in OSF.

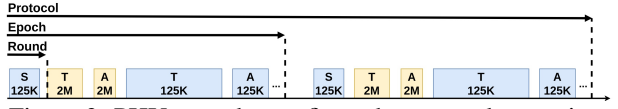


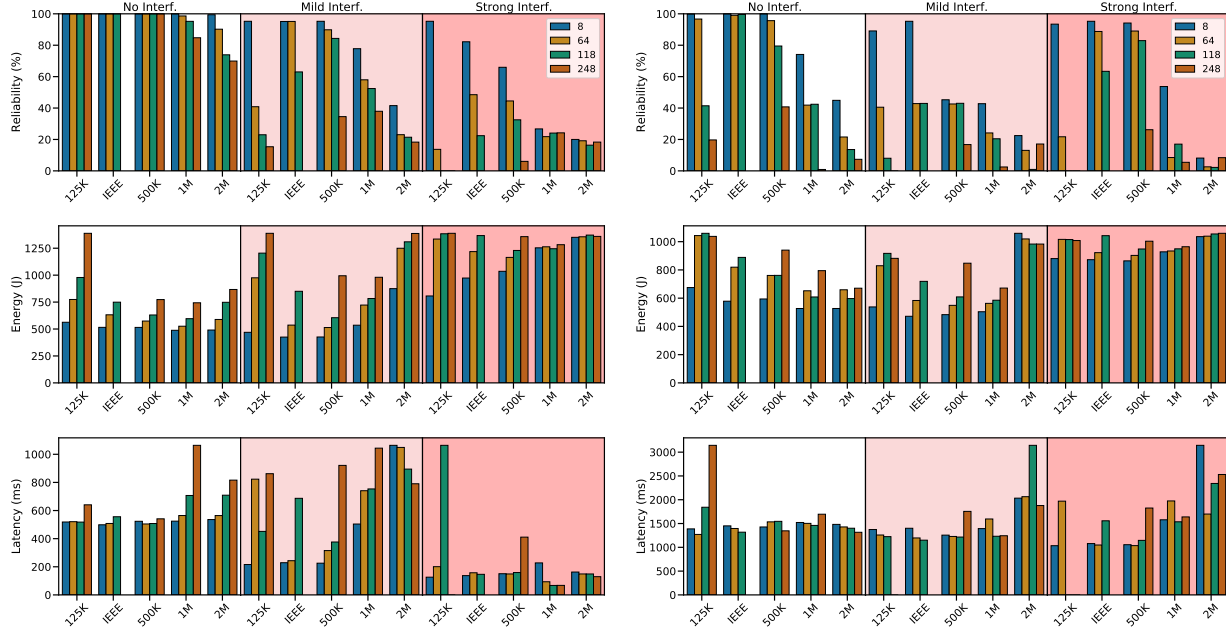
Figure 3: PHYs may be configured per-round at runtime.

Making use of OSF. OSF introduces two features for SF experimentation. Firstly, a novel *driver extension framework* (as shown in Fig. 1) provides hook-ins to key functions within the OSF driver, as well as OSF protocols. For example, the RNTX mechanism introduced in Sect. 4 is implemented as a *driver extension* and introduces a random number of transmissions within each round without touching the base driver, while the backoff mechanism (also introduced in Sect. 4) is implemented as a *protocol extension* without modification to the underlying collection protocol. Secondly, OSF enables *unprecedented runtime customization* of the SF round configuration and underlying radio PHY. Fig. 3 demonstrates how OSF is capable of switching PHY configuration on a per-round basis to build multi-PHY protocols. In this simple example, the *synchronization* (S) round is sent on the PHY with the furthest range, while *transmission and acknowledge* (TA) pairs are allocated either the BLE 5 2M or BLE 5 125K PHY. This ensures synchronization of far-away nodes through use of a long-range BLE 5 125K PHY during the S round, while equally providing opportunity for those nodes to transmit back to the sink during the BLE 5 125K TA pair – all this while nearer nodes may utilize the higher data-rate BLE 5 2M PHY, thus spending less time on-air and saving energy.

4 Experimental Evaluation

We evaluate OSF in comparison to recent SF benchmarking of a polling-based driver [5] based on BlueFlood [4], and provide the first evaluation of SF protocols for larger packet sizes (118 and 248 bytes). We subsequently show that simple randomization and backoff mechanisms are capable of avoiding the large number of collisions occurring when using BLE 5 uncoded PHYs in dense scenarios [5, 10]. Finally, we provide initial data on multi-PHY SF protocols, where we vary utilization between long-range (BLE 5 125K) and high data-rate (BLE 5 2M) PHYs and demonstrate a naïve multi-PHY protocol capable of adapting to time-variant interference.

Experimental setup. We perform all our experiments on the D-Cube testbed embedding 48 nRF52840 nodes. We set the transmission power for all nodes to 0dBm, the number of transmissions after the first reception (*NTX*) to 6, and the maximum number of slots in any flooding round (*NSLOTS*) to 12. RoF [15] with timeslot-level channel hopping is used as the underlying SF primitive for both dissemination and collection scenarios, where the epoch periodicity is set to 200 ms and 1 s, respectively. For data collection, we implement a modified Crystal [27] protocol (operating over RoF



(a) Data dissemination (b) Data collection
 Figure 4: OSF benchmarking over the RoF primitive for 8, 64, 118, and 248 byte data packets.

rather than Glossy), where we set the number of transmission-acknowledge (TA) pairs after the initial synchronization (S) round (*NTA*) to 12, and the number of empty TA pairs before exit to 4. Unless otherwise stated, all experiments used D-Cube’s 30s aperiodic data generation. These configuration settings are in line with recent benchmarking studies [5].

Data dissemination and collection using OSF. We benchmark the performance of OSF data dissemination and collection by comparing all IEEE 802.15.4 and BLE 5 PHYs across *no*, *mild*, and *strong* interference levels (jamming levels 0, 1, and 3 in D-Cube [28], respectively). Dissemination was evaluated over a *one-to-all* broadcast layout (layout 4 on D-Cube) with a single source transmitting to 47 nodes, while collection was evaluated over an *all-to-one* layout with all nodes transmitting to a single destination (layout 3 on D-Cube).

Fig. 4 shows that OSF exhibits a dependable performance. Specifically, in Fig. 4a dissemination reliability is roughly in-line with similar experiments performed on a polling-based driver [5, Fig. 10], while both latency and energy are reduced by up to 50%. Furthermore, in Fig. 4b OSF demonstrates high reliability in collection scenarios for smaller packets across the coded PHYs (particularly under *strong* interference) while retaining energy-efficient operations.

Additionally, this is the first study to evaluate the impact of larger packet sizes. In addition to 8 and 64 byte data, we evaluate 118 (the IEEE 802.15.4 MTU minus the CRC + OSF header) and 248 bytes data (the BLE 5 MTU minus the OSF header). Interestingly, we find that while previous studies suggest that IEEE 802.15.4 and BLE 5 500K had similar performance under interference [5], BLE 5 500K is better able to handle the 118B packets, likely due to the reduced on-air time of the PHY. However, when considering 248B (too large for IEEE 802.15.4) we see considerable performance degradation across all PHYs when under any sort of interference.

Random NTX and random backoff. While higher data-rate PHYs consume less energy, due to the beating effect and desynchronization BLE 5 uncoded PHYs are more susceptible to packet errors as the number of simultaneous transmitting nodes increases – meaning that the reliability of SF protocols adopting these PHYs drops significantly even when under no external interference [5]. OSF introduces two mechanisms to reduce the number of simultaneous transmitters on uncoded PHYs, thereby improving reception rate: (i) a random number of transmissions after the first reception (*RNTX*) that can be applied to any SF primitive; (ii) a fixed-threshold random backoff on the ‘T’ round (*BACKOFF*) specifically designed for SF-based data collection protocols.

RNTX. Before each round each node calculates a random number on the interval $[NTX_0, 2NTX_0]$, with NTX_0 representing the given NTX. This number is then applied to the round configuration, ensuring that transmission slots towards the end of the round will experience fewer numbers of transmissions as nodes will have random NTX. This approach can therefore provide considerable gains in dense network areas.

Random backoff. While protocols such as Crystal [27] have previously relied on significant capture effect to allow neighboring nodes to concurrently transmit different data during the same flooding round, recent studies have shown that BLE 5 uncoded PHYs perform poorly in this manner [5, 10]. We implement a simple random backoff mechanism to reduce the number of initiators within different-data transmission (T) rounds. Source nodes initiate a flood with a fixed probability of 80%; otherwise, they act as a forwarder for this round.

Results. To evaluate these mechanisms, we set the PHY to BLE 5 2M, and run experiments on D-Cube’s dense layout (19 source nodes), where all source nodes are roughly one hop away from the sink. To ensure a high degree of competition between the nodes, 5 s periodic data generation is used, while

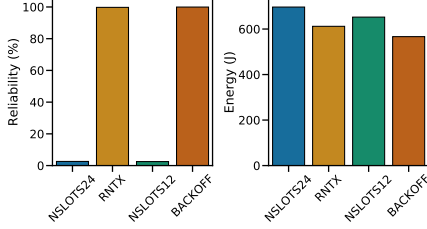


Figure 5: RNTX and backoff address collisions in BLE 5 2M.

the data length is set to 64 bytes. To ensure sufficient slots towards the end of the packet, the maximum number of slots for the RNTX mechanism is set to 24 (i.e., each node will transmit between 6 and 24 times), while the maximum number of slots remains at 12 for the random backoff mechanism. Fig. 5 shows that regardless of whether the maximum slots is set to 12 (NSLOTS12) or 24 (NSLOTS24), BLE 5 2M has extremely poor reliability. However, when using *either* RNTX or the random backoff, 100% reliability can be achieved, while the backoff mechanism consumes less energy than RNTX.

Static multi-PHY benchmarking. We evaluate OSF’s multi-PHY capabilities of over 3 collection scenarios: an *all-to-one* layout with 47 sources transmitting to a single node near the topology center; a *sparse* layout with 12 sources (mostly located at the edge of the network with some in the middle) and a destination (located in one corner); finally, a *dense* layout with 19 sources and the destination all in the same room. Fig. 6 shows benchmarking of static multi-PHY patterns with increasing BLE 5 2M utilization of the collection protocol TA pair schedule – from 0% (i.e., all BLE 5 125K) to 100% (i.e., all BLE 5 2M) at 25% increments. Specifically, after an initial S round at BLE 5 125K, static BLE 5 125K/2M patterns are repeated across the epoch’s 12 TA pairs. For a single-PHY approach, while the BLE 5 125K is capable of achieving near 100% reliability, the high data-rate BLE 5 2M PHY is unable to successfully deliver all messages (please note that this experiment does *not* make use of the RNTX or random backoff mechanisms previously introduced). However, across all layout scenarios, a multi-PHY compromise maintains a high reliability while introducing energy gains of $\approx 40\%$ with MPHY75 (i.e., where one of every four TA pairs is configured with the BLE 5 125K PHY, while the rest are configured with the BLE 5 2M PHY).

Dynamic multi-PHY protocol. Benchmarking of static PHY patterns in Fig. 6 fails to account for diverse networks, or changing RF conditions. That is, such an approach needs to be configured at compile-time and is unsuited to real-world scenarios. We therefore implement a naïve dynamic multi-PHY protocol, based on the receive ratio at the sink, as an initial demonstrator of OSF’s multi-PHY runtime capabilities and their potential. Specifically, we devise a time-varying interference scenario as per Fig. 7, where levels move from *no*, *mild*, *no*, *strong*, and finally returning to *no interference*. In such a scenario, uncoded PHYs perform well during the periods of *no interference*, yet struggle during the interference periods. On the other hand, BLE 5 500K can perform well even under *strong interference* (unlike BLE 125K [5]). We therefore utilize a BLE 5 500K/2M split rather than the 125K/2M split previously employed in the static benchmark-

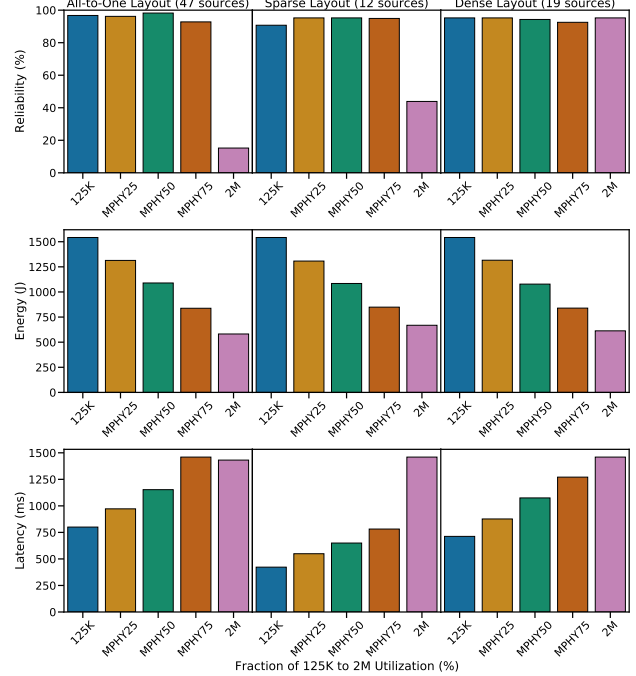


Figure 6: Static multi-PHY patterns (no interference).

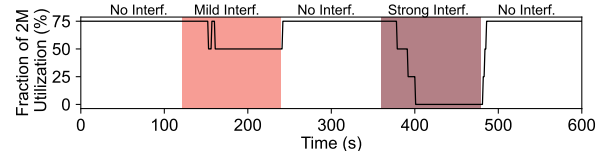


Figure 7: Timeline of 2M utilization in Fig. 8 sparse layout.

ing (under no interference). In every epoch, the destination node calculates the percentage of *late* nodes (i.e., sources that have not been received within the 30s aperiodic time window). Based on this percentage, an appropriate multi-PHY pattern is selected according to Tab. 2 and disseminated to the network. Fig. 8 compares this naïve multi-PHY approach against BLE 5 500K and BLE 5 2M single-PHY across the three D-Cube layouts previously outlined in Fig. 6. Due to interference, BLE 5 2M is incapable of achieving high reliability on any of the layouts, while BLE 5 500K is able to escape – providing close to 100% reliability in the sparse and dense layouts. As expected, the multi-PHY approach is able to provide similar levels of reliability to BLE 5 500K, with significantly reduced energy ($\approx 20\%$). Furthermore, Fig. 7 demonstrates how the multi-PHY protocol adapts during the experiments, dynamically switching to greater BLE 5 500K utilization during periods of interference, while increasing BLE 5 2M utilization during periods without interference.

5 Discussion and Outlook

This paper has introduced OSF, a novel SF architecture and middleware supporting *runtime configuration* of the underlying PHY. While some recent works have leveraged the multi-PHY support of modern low-power wireless chipsets such as the nRF52840 to support routing-based IEEE 802.15.4 solutions with SF data dissemination [18, 20], it has been shown that the underlying physical layer phenomena (namely capture

Table 2: Multi-PHY pattern selection.

# of late nodes	[0%, 25%]	(25%, 50%]	(50%, 75%]	(75%, 100%]
2M utilization	MPHY75	MPHY50	MPHY25	MPHY0

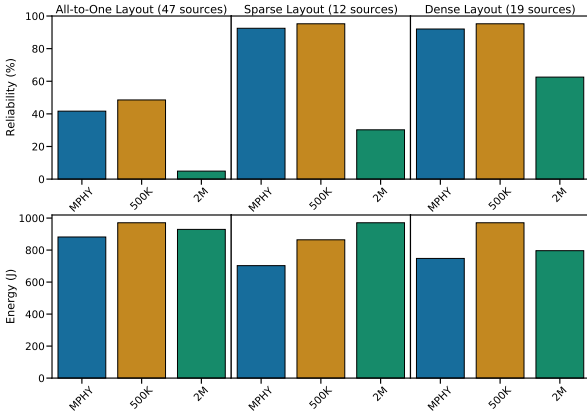


Figure 8: Performance of a naive multi-PHY protocol over time-variant interference compared to single-PHY settings.

effect, beating effect, and desynchronization) play a significant role in the overall performance of SF protocols [20]. These findings suggest that specific PHYs may enhance, or hinder, SF performance within certain scenarios – providing an argument for the research community to research and develop new SF protocols with multi-PHY capabilities.

To this end, we have made `OSF` publicly available as a platform for further research. In support of this, we have benchmarked `OSF` across the D-Cube testbed for single-PHY dissemination and collection scenarios, where `OSF` demonstrates notable performance improvements under interference when compared to polling-based SF implementations, and experiments on larger packet sizes suggest issues when using the full BLE 5 MTU. Furthermore, the `OSF` middleware not only features network-level SF protocol construction (as with other SF middlewares [6, 7, 8]), but provides unprecedented runtime configuration options, as well as a low-level driver extension framework to allow development of enhancements to the `OSF` driver while separating the complexities of radio access and the interrupt-based state machine. Based on this, we have implemented and evaluated two simple mechanisms that are capable of overcoming the critical weaknesses of uncoded BLE 5 PHYs in dense environments – achieving 100% reliability as opposed to near-total collisions.

Fundamentally, this paper has validated `OSF`’s multi-PHY approach through evaluation on the D-Cube testbed, where we show that static PHY patterns (of varying utilization between BLE 5 125K and 2M) achieve high reliability whilst reducing energy and latency. Furthermore, we have demonstrated the effectiveness of even a naive approach to dynamic runtime PHY switching, and achieve considerable energy gains over that of a single-PHY protocol when taking into account time-variant interference scenarios. While the majority of SF research has, to date, only considered controlled environments, these early results suggest that an intelligent multi-PHY approach can significantly improve SF protocol performance for unpredictable real-world scenarios.

6 References

- [1] M. Zimmerling *et al.*, “Synchronous Transmissions in Low-Power Wireless: A Survey of Communication Protocols and Network Services,” *ACM Computing Surveys (CSUR)*, 2020.
- [2] C. A. Boano *et al.*, “EWSN Dependability Competition: Experiences and Lessons Learned,” *IEEE Internet of Things Newsletter*, 2017.
- [3] M. Schuß *et al.*, “A Competition to Push the Dependability of Low-Power Wireless Protocols to the Edge,” in *Proc. of the 14th EWSN Conf.*, 2017.
- [4] B. A. Nahas *et al.*, “Concurrent Transmissions for Multi-Hop Bluetooth 5,” in *Proc. of the 16th EWSN Conf.*, 2019.
- [5] M. Baddeley *et al.*, “The Impact of the Physical Layer on the Performance of Concurrent Transmissions,” in *Proc. of the 28th IEEE ICNP Conf.*, 2020.
- [6] B. A. Nahas *et al.*, “Network-wide Consensus Utilizing the Capture Effect in Low-Power Wireless Networks,” in *Proc. of the 15th ACM SenSys Conf.*, 2017.
- [7] M. Baddeley *et al.*, “Atomic-SDN: Is Synchronous Flooding the Solution to Software-Defined Networking in IoT?” *IEEE Access*, 2019.
- [8] R. Jacob *et al.*, “Synchronous Transmissions Made Easy: Design Your Network Stack With Baloo,” in *Proc. of the 16th EWSN Conf.*, 2019.
- [9] L. Tuchtenhagen, “More Easy Synchronous Transmissions: Expanding Baloo to the nRF52840,” 2022.
- [10] R. Jacob *et al.*, “Synchronous Transmissions on Bluetooth 5 and IEEE 802.15.4: A Replication Study,” in *Proc. of the 3rd CPS-IoTBench Workshop*, 2020.
- [11] F. Ferrari *et al.*, “Efficient Network Flooding and Time Synchronization with Glossy,” in *Proc. of the 10th IPSN Conf.*, 2011.
- [12] O. Landsiedel *et al.*, “Chaos: Versatile and Efficient all-to-all Data Sharing and In-Network Processing at Scale,” in *Proc. of the 11th ACM SenSys Conf.*, 2013.
- [13] M. Mohammad and M. C. Chan, “Codecast: Supporting Data Driven In-Network Processing for Low-Power Wireless Sensor Networks,” in *Proc. of the 17th IPSN Conf.*, 2018.
- [14] C. Herrmann *et al.*, “Mixer: Efficient Many-to-All Broadcast in Dynamic Wireless Mesh Networks,” in *Proc. of the 16th SenSys Conf.*, 2018.
- [15] R. Lim *et al.*, “Competition: Robust Flooding using Back-to-Back Synchronous Transmissions with Channel-Hopping,” in *Proc. of the 14th EWSN Conf.*, 2017.
- [16] X. Ma *et al.*, “Harmony: Saving Concurrent Transmissions from Harsh RF Interference,” in *Proc. of the INFOCOM Conf.*, 2020.
- [17] V. Poirot *et al.*, “Paxos made wireless: Consensus in the air,” in *Proc. of the 16th EWSN Conf.*, 2019.
- [18] T. Istomin *et al.*, “Route or Flood? Reliable and Efficient Support for Downward Traffic in RPL,” *ACM TOSN*, 16(1), 2019.
- [19] A. Brandt *et al.*, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” RFC 6550, 2012.
- [20] M. Baddeley *et al.*, “6TiSCH++ with Bluetooth 5 and Concurrent Transmissions,” in *Proc. of the 18th EWSN Conf.*, 2021.
- [21] P. Thubert, “An Architecture for IPv6 over the TSCH mode of IEEE 802.15.4,” IETF, Tech. Rep. draft-ietf-6tisch-architecture-24, 2019.
- [22] D. Lobba *et al.*, “Concurrent Transmissions for Multi-hop Communication on Ultra-Wideband Radios,” in *Proc. of the 17th EWSN Conf.*, 2020.
- [23] C.-H. Liao *et al.*, “Multi-Hop LoRa Networks Enabled by Concurrent Transmission,” *IEEE Access*, 2017.
- [24] M. Bor *et al.*, “LoRa for the Internet of Things,” in *Proc. of the 13th EWSN Conf.*, 2016.
- [25] P. Tian *et al.*, “ChirpBox: An Infrastructure-Less LoRa Testbed,” in *Proc. of the 18th EWSN Conf.*, 2021.
- [26] B. A. Nahas *et al.*, “BlueFlood: Concurrent Transmissions for Multi-hop Bluetooth 5 – Modeling and Evaluation,” *ACM TIOT*, 2(4), 2021.
- [27] T. Istomin *et al.*, “Data Prediction + Synchronous Transmissions = Ultra-Low Power Wireless Sensor Networks,” in *Proc. of the 14th ACM SenSys Conf.*, 2016.
- [28] M. Schuß *et al.*, “JamLab-NG: Benchmarking Low-Power Wireless Protocols under Controllable and Repeatable Wi-Fi Interference,” in *Proc. of the 16th EWSN Conf.*, 2019.