Opportunistic Routing and Synchronous Transmissions Meet TSCH

Citation for the original published paper (version of record):
Harms, L., Landsiedel, O. (2021). Opportunistic Routing and Synchronous Transmissions Meet TSCH. Proceedings - Conference on Local Computer Networks, LCN, 2021-October: 107-114.
http://dx.doi.org/10.1109/LCN52139.2021.9524952

Opportunistic Routing and Synchronous Transmissions Meet TSCH

Laura Harms[1,2], Olaf Landsiedel[1,2]
[1]*Kiel University, Germany*
[2]*Chalmers University of Technology, Sweden*
{lah, ol}@informatik.uni-kiel.de

***Abstract*—Low-power wireless networking commonly uses either Time-Slotted Channel Hopping (TSCH), synchronous transmissions, or opportunistic routing. All three of these different, orthogonal approaches strive for efficient and reliable communication but follow different trajectories. With this paper, we combine these concepts into one protocol: AUTOBAHN.**

AUTOBAHN merges TSCH scheduling with opportunistically routed, synchronous transmissions. This opens the possibility to create long-term stable schedules overcoming local interference. We prove the stability of schedules over several days in our experimental evaluation. Moreover, AUTOBAHN outperforms the autonomous scheduler Orchestra under interference in terms of reliability by 13.9 percentage points and in terms of latency by a factor of 9 under a minor duty cycle increase of 2.1 percentage points.

***Index Terms*—TSCH, Opportunistic Routing, Synchronous Transmission, Central Scheduling, Wireless Sensor-Actuator Networks, (Industrial) Internet of Things**

I. INTRODUCTION

Within the past 20 years, research on low-power wireless networking resulted in a multitude of different protocols. They fall into three prominent fields: Time-Slotted Channel Hopping (TSCH), opportunistic routing, and synchronous transmissions. So far, all three of these fields have little to no overlap, while all strive for a common goal of stable, reliable communication in low-power wireless networks.

In the first field of protocols, the IEEE 802.15.4 Time-Slotted Channel Hopping (TSCH) [22] MAC layer protocol forms the basis for many routed communication protocols. This protocol is standardized and dominates the industry. One category of TSCH protocols uses centralized schedulers, separating the network communication from the routing and scheduling. In recent works [6], [21], centralized schedulers show high reliability and stability. Another category are autonomous schedulers with Orchestra [15] as a prominent example.

TSCH protocols offer stability regarding narrow-band interference. However, long-term stable schedules that are prone to wide-band interference are an open challenge. Wide-band interference likely leads to link failures or even node failures heavily affecting routed communication.

The other two fields can overcome these challenges. Opportunistic routing [3], [16], [31] utilizes anycasts instead of unicasts to add forwarding flexibility by addressing a packet to multiple potential forwarders. It increases the possibility of successful reception in the presence of wireless link dynamics. Protocols building upon synchronous transmissions [17], [18], [30] allow multiple nodes to transmit packets concurrently, commonly by network-wide flooding.

Synchronous transmissions achieve high reliability even in the presence of wide-band interference. However, they have an impact on all nodes in a network. If, for example, in a 1000 node network, two nodes two-hops apart want to communicate, the whole network is involved. In a routed network, only a fraction of these nodes needs to communicate.

In this paper, we ask the following question: Can we combine the benefits of opportunistic routing, synchronous transmissions and centralized TSCH scheduling? For this, we introduce AUTOBAHN: a hybrid routing scheme that combines the best of these worlds: centrally scheduled flows and one-to-one routing of packets as in traditional networking combined with the reliability and robustness of opportunistic routing and synchronous transmissions.

The basic concept of AUTOBAHN is as follows: Its central scheduler schedules a flow along a wider path and allows neighboring nodes to transmit concurrently the same data at the same timeslot and frequency. Thus, a node forwards a packet opportunistically to multiple neighboring nodes, which in turn, in the next slot, concurrently forward opportunistically to their neighbors. In our evaluation, we show that by combining these three approaches, AUTOBAHN efficiently provides reliable, low-latency packet delivery even when links fail, and its schedules are stable for days even in the presence of dynamic interference.

Overall, this paper makes the following contributions:

- We are the first to combine the concepts of opportunistic routing, synchronous transmissions, Time-Slotted Channel Hopping (TSCH) into a single protocol to achieve long-term stable routed communication.
- We design AUTOBAHN, a robust scheduling and routing policy that withstands link and node failures in the presence of interference.
- We implement AUTOBAHN for Contiki-NG [1] and evaluate it in environments susceptible to interference. We show the long-term stability of schedules using AUTOBAHN over 12 days and under various interference levels for 25 hours. These experiments achieve reliability under interference of 96.8% and latency of 4.2 slots outperforming both the central scheduler MASTER [21] and the autonomous TSCH scheduler Orchestra [15].

The remainder of this paper is organized as follows. Section II gives the necessary background information and reviews related work on TSCH as well as the concepts combined in AUTOBAHN. In Section III, we introduce the design of AUTOBAHN. In Section IV we evaluate AUTOBAHN's performance experimentally, followed by the conclusion in Section V.

II. BACKGROUND & RELATED WORK

In this section, we introduce the necessary background on TSCH, opportunistic routing, and concurrent transmissions and discuss the relevant related work.

A. Time-Slotted Channel Hopping (TSCH)

The MAC protocol Time-Slotted Channel Hopping (TSCH) [22] is a combined TDMA and FDMA MAC protocol. It uses 10 ms long time slots with up to 16 frequency channels at each time slot. All active channels follow a pseudo-random hopping sequence that is cycled through, using a different channel at each timeslot to counteract narrow-band interference.

TSCH groups communication slots in continuously repeated slot-frames. All slot-frames together form the TSCH schedule. A TSCH schedule is generated by a centralized, autonomous, or distributed scheduler.

Centralized Scheduling: Central schedulers use global knowledge about the network topology (esp. wireless link quality) to build a schedule and disseminate the schedule into the network. Many early ones, such as TASA [35] and others [20], [36], assume interference-free wireless channels without lossy links and, therefore, do not include retransmissions in their schedules. Later work focuses on increasing reliability in the presence of fading channels while ensuring end-to-end latency requirements of each flow. They achieve this by adding retransmissions, i.e., slot-based retransmissions, as used by AMUS [26], to the schedule. As interference can rarely be linked to a specific location beforehand, some recent works by Brummet et al. [6], and MASTER [21] introduce a new approach to retransmissions in TSCH scheduling: they introduce flow-based retransmissions achieving lower latency and a higher degree of adaptability to local interference level.

Autonomous/Distributed Scheduling: Next to these centralized TSCH protocols, a significant amount of work concentrates on autonomous scheduling, a concept introduced by Orchestra [15] and extended by others [28], [37]. Distributed scheduling on the other hand builds on 6TiSCH with its default scheduling function MSF [9], as well as, LLSF [11] and LDSF [29], focusing on improving latency in distributed TSCH.

Multipath TSCH: For multi-path communication in TSCH, several algorithms [33], [34] were studied for distributed and centralized scheduling scenarios. To some extent, these works propose similar ideas as AUTOBAHN, yet they clearly stay within the specifications of TSCH and do not apply opportunistic routing or synchronous transmissions. Moreover, their evaluation results are solely based on simulation.

B. Opportunistic Routing

Opportunistic routing is a routing approach to improve network throughput, communication reliability and efficiency in wireless multi-hop mesh networks. Instead of performing unicast communication as established TSCH schedulers do, opportunistic routing builds upon anycasts. By this, opportunistic routing sends each packet to a set of receivers. If any of them receives the packet, the transmission is successful. As multiple receivers might receive the packet, opportunistic routing has to overcome the challenge of selecting a unique forwarder. This forwarder selection has to wait until after the transmission [3], [8], [12].

While initial works do not use duty-cycled, low-power wireless networking, later works such as as ORW [31] and ORPL [16] bring opportunistic routing to these. Nonetheless, these protocols are not built for TSCH. Huynh et al. [25], Hermeto et al. [23], and Hosni et al. [24] study the use of opportunistic routing or anycasts in TSCH and propose changes to TSCH to allow non-colliding acknowledgments from multiple receivers. BOOST [27] introduces forwarder selection through sending delays with carrier sense in TSCH. In contrast to these approaches, AUTOBAHN does not use any preferred forwarder selection method. Instead, we overcome this challenge by using synchronous transmissions.

C. Synchronous Transmissions

Synchronous transmission protocols allow multiple nodes to transmit packets simultaneously. With precise timing, these packets do not collide destructively, allowing protocols to achieve high communication reliability [17], [18]. As a result, protocols employing synchronous transmissions do not maintain routes by selecting parent nodes, announcing routing metrics, discovering neighbors, and maintaining routing tables as traditional routing protocols.

For receiving such a packet, the senders must not significantly differ in timing. One common option of receiving synchronous transmissions is the so-called Capture Effect [32]. According to the capture effect in IEEE 802.15.4, a stronger signal must not arrive later than $160\mu s$ after the first signal [30]. When sending the same data, non-destructive interference is achievable if the time offset between multiple senders is within a bound of $0.5\mu s$ [17].

Synchronous transmissions are well studied. Glossy [17] laid the foundation for synchronous transmissions in wireless sensor networks. Since Glossy's introduction, many protocols including Chaos [30] and LWB [18] followed. They all are protocols that use network-wide flooding without a concept of routing. Protocols like WSNShape/Sparkle [38], CXFS [7] and LaneFlood [5] divert from network-wide flooding and use flooding with some notion of routing along a path of forwarders.

All of these protocols operate without a routing layer, whereas AUTOBAHN follows the principle of combining synchronous transmissions and TSCH as envisioned by Chang et al. [10]. Gomes et al. [19] study an initial approach of flooding-based routing in TSCH. This approach relies fully on broadcasts (no acknowledgements) and uses shorter TSCH slots. Baddeley et al. [2] present a hybrid between TSCH and synchronous transmissions by replacing some TSCH slots with synchronously transmitted BLE packets for exchanging control information.

While some protocols explore the field of combining TSCH and synchronous transmissions, AUTOBAHN explores it further by combining synchronous transmissions with TSCH, including both synchronous transmissions as well as synchronous acknowledgments in combination with opportunistic routing.

III. DESIGN

We continue with the design of AUTOBAHN. We begin with a simple example to present the basic idea of AUTOBAHN. Then, we introduce (1) general node selection requirements, (2) the forwarder selection through node ranks, and (3) the active nodes in each slot. After discussing these main points of the design, we present the system design, including the Contiki/TSCH extensions to allow anycast communication, and AUTOBAHN's routing layer adaptations.

A. AUTOBAHN*: General Idea*

As an example, we assume the network of five nodes in Fig. 1 where Node A communicates with node E. Further, let us assume that the link between nodes A and B fails due to interference.

To illustrate the benefits of AUTOBAHN, we first discuss how established centralized scheduling approaches suffer from link failures. Established approaches commonly employ a single routing path. Their schedule will fail if one of the links fails, such as the link between nodes A and B in this example, see Fig. 1a. Retransmissions, as scheduled in the example, usually happen on a different channel and thereby protect the protocol against narrow-band interference. Wide-band interference, however, can break links and result in packet loss. Eventually, the scheduler has to deploy an updated schedule. If this is done frequently, this adds significant overhead to the communication scheme.

The general idea behind AUTOBAHN is to add redundancy to the routing path, see Fig. 1b. In the example of AUTOBAHN, node A sends a packet that will be received by nodes B and C. These two forward the packet synchronously to nodes D and E, which receive one of the two transmissions due to the capture effect. Lastly, node D sends the packet to node E as well. In case of interference, node B is not reachable. That means that only node C receives the packet from node A. Node C then forwards the message to node E. Redundant routing paths in AUTOBAHN add non-neglectable overhead to the duty cycle of the network. In our evaluation, we, however, show that this overhead is justifiable in the interference-free case, and in the case of interference, it is essential for reliable communication.

B. Routing Set

Centralized schedulers have global knowledge over the network topology through long-term link quality metrics. They commonly route traffic along a single path, using single forwarders. In contrast to that, AUTOBAHN addresses packets to multiple forwarders (anycast). To achieve this, we employ a routing set with redundancy instead of a single path.

We define a routing set to consist of all nodes we use for end-to-end communication. A routing set $\{rs\}$ is a set of nodes $n_1, ..., n_k \in \{rs\}$ responsible for routing a data packet from the source node n_1 to the sink node n_k. This routing set contains the nodes forming the shortest path from source to sink as well as additional nodes used for opportunistic, anycast routing in AUTOBAHN, which adds path redundancy.

To build a routing set, we start with the shortest path from source to destination employing the ETX-metric [14] and Dijkstra's shortest-path algorithm [13]. Next, we add routing redundancies by including neighboring nodes along the path. For this, we introduce three schemes: (i) neighbor-based, (ii) hop-based, and (iii) cost-based selection of routing sets. Especially in dense networks, the number of these neighboring nodes for each of the three schemes is likely to be high and leads to the inclusion of massive parts of the network. Therefore, we specify a node overhead factor (scaling the number of additional nodes) and a cost overhead factor (scaling the max. allowed ETX cost of nodes of an end-to-end path).

Neighbor-based participant selection starts with determining all nodes neighboring at least one node of the shortest path. From these nodes, we continue with three different subsets: (a) all selected nodes that do not exceed the cost overhead, (b) a subset of (a) forming a second shortest path, and (c) a subset of (a) forming a shortest path from each node of the original path to the destination node. After selecting the respective nodes, we check whether the node overhead is too large. If so, we refine our selection only to include the allowed number of nodes with the lowest cost.

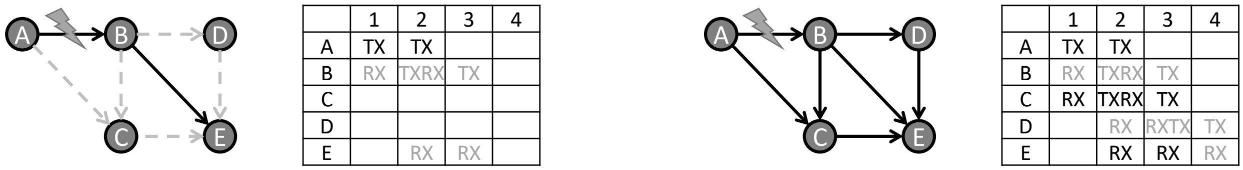
The hop-based selection possibility includes additional nodes with a similar combined distance to the shortest path's source and destination while not exceeding the cost overhead. Equation 1 shows the general idea of this selection strategy, that the combined hop distances from source to forwarder (d_{sf}) and forwarder to destination (d_{df}) must not exceed the direct distance d_{sd} plus a slack value s. The slack value has to be a natural number. If the node overhead is too large, nodes with the lowest hop count are preferred.

$$d_{sf} + d_{df} \leq d_{sd} + s \tag{1}$$

The cost-based selection possibility follows the same equation. However, instead of hop-based distances, we use ETX-based distances and an ETX-based slack (s), which can be a positive real number. However, the maximum slack value for this strategy equals the maximum cost overhead allowed according to the cost overhead factor. If, after node selection, the node overhead is too large, we rank the nodes regarding their cost and take those with the lowest cost. In addition, we exclude all other nodes that have the same cost as one of the already excluded nodes.

C. Anycast forwarding in AUTOBAHN

In anycast routing, we address a packet to a set of neighboring nodes, i.e., the ones making sufficient progress towards the destination. Thus, for each transmission in AUTOBAHN, this set of possible recipients listens for the packet. Practically,



	1	2	3	4
A	TX	TX		
B	RX	TXRX	TX	
C				
D				
E		RX	RX	

(a) Established central scheduling approaches employ a single routing path. Their schedule will fail if one of the links fails, such as the link between nodes A and B in this example.

	1	2	3	4
A	TX	TX		
B	RX	TXRX	TX	
C	RX	TXRX	TX	
D		RX	RXTX	TX
E		RX	RX	RX

(b) AUTOBAHN utilizes multi-path routing and thereby provides redundant options in case routes fail. In this example, packets can travel via node C to destination E.

Fig. 1: AUTOBAHN compared to established centralized TSCH scheduling approaches. In this example, we assume a topology of five nodes, with node A as source and node E as destination. We show both the scheduled paths and the TSCH schedule, using RX, RXTX, and TX slots as typical for flow-based retransmission schemes (with a retransmission window of two). Grayed-out slots present slots where reception and transmission are not possible due to previously failed interfered receptions.

we introduce node ranks: Each node has a rank according to its distance to the destination of a flow from source to destination. The sender of a packet has rank 0, and the rank increases towards the receiver, with the receiver having the highest rank.

If a node receives a packet, it compares its rank to the sender's rank, which we include in the packet header. If the own rank is higher, it acknowledges the packet and forwards it. Otherwise, it stays silent and acknowledges for itself that the packet has passed. The sender of a packet performs a similar action. If it receives an acknowledgment from a node with a higher rank, it concludes that the opportunistic anycast succeeded and stops forwarding this packet. This way, we ensure that only nodes closer to the packet's destination acknowledge the reception of the packet and forward the packet; thus, we avoid loops and packets stuck mid-flow.

In traditional opportunistic routing, packet duplicates are often a challenge [3], [16], [31]: There is always a risk that multiple forwarders receive a packet, and each individually forwards the packet, adding additional load on the network. In AUTOBAHN, all packets – including duplicates – are forwarded synchronously, and their spatial diversity is the basis for the reliability of our design in the presence of interference. Thus, duplicates are (i) inherently part of the design and (ii) do not add the overhead as in traditional designs.

D. Active slots in AUTOBAHN

AUTOBAHN uses flow-based retransmission schemes such as Sliding Windows introduced by MASTER [21], with multiple nodes possibly active in one slot. AUTOBAHN extends this by activating all nodes in a slot that are reachable by any previously active node.

In the first slot of a flow, the sender of a packet and all receivers in range are active. For each of the following slots, we include all additional nodes reachable by any previously active node. From this, we derive the first active slot of a node, i.e., the first point in time a packet in a flow can reach a node along one of the different paths employed by AUTOBAHN. We determine a node's last active slot based on the node's hop distance to the flow's receiver and the flow's number of total transmissions.

Due to the opportunistic nature of AUTOBAHN, a high network duty-cycle is expectable. Nevertheless, to still keep the energy consumption as low as possible, each node stays only active until we no longer need it for forwarding the packet. As we explain above, a node determines whether it is still needed through the received rank of other participants.

The schedules in Fig. 1 illustrate the difference between the active slots of a flow-based central scheduler (Fig. 1a) without opportunistic routing and AUTOBAHN (Fig. 1b).

E. System Integration

For the design of AUTOBAHN, we devise a TSCH implementation with support for opportunistic anycasts and a good enough time synchronization for synchronous transmissions. In our evaluation, we show that the TSCH implementation of Contiki-NG is sufficient for synchronous transmissions. However, as it does not support anycasts, we have to realize these ourselves. AUTOBAHN itself can be implemented on top of any centralized scheduler. We choose MASTER, a centralized scheduler implemented for Contiki-NG as our basis. We implement AUTOBAHN to replace MASTER's central routing and retransmission logic while keeping its scheduling module. Below, we discuss the integration into MASTER's Contiki routing layer and the extension of Contiki-NG/TSCH to allow opportunistic anycasts.

F. Integration in MASTER's routing layer

We extend MASTER's routing layer to have access to a node's rank and relay a packet back to the correct flow address instead of a neighbor address.

The routing layer is also responsible for the routing-specific header. In addition to the existing 7-byte routing header, AUTOBAHN requires one additional byte. The existing 7 bytes are a flow identifier (1 byte), a sequence number (2 bytes), the time-to-live (TTL) (2 bytes), and the earliest transmission slot (2 bytes) of a packet. AUTOBAHN adds the node's rank to the packet to allow the receiver to make its forwarding decision according to our description above.

1) Contiki-NG/TSCH extensions: The TSCH implementation of Contiki-NG does not support anycast communication. To add support for anycasts, we extend it with (1) the capability of accepting packets from any neighbor of a flow, as well as (2) using this flow as a sender and receiver simultaneously. Moreover, we (3) define a flow-specific sequence number to accept acknowledgments successfully.

With our modification, TSCH accepts packets from a flow address if the receiving node is a member of the respective flow. As we no longer need the sender's and receiver's addresses, we replace them with the flow address.

Besides accepting packets from any flow participant, TSCH needs the capability to accept acknowledgments from any possible forwarder of the flow. Therefore, we include the receiver's rank in the acknowledgment. If a node receives an acknowledgment acknowledging a different synchronous sender, it still needs to be accepted. Therefore, our routing layer replaces the TSCH sequence number with a flow and packet-specific end-to-end sequence number.

IV. Evaluation

In this section, we evaluate Autobahn's performance and compare it to the state-of-the-art. We start by showing the feasibility of synchronous transmissions in the context of TSCH. After that, we evaluate Autobahn's different routing set selection choices and compare those to Master in scenarios with and without interference. Afterward, we compare Autobahn's best routing selection algorithm against Orchestra, the default autonomous scheduler in Contiki-NG. We conclude our evaluation with long-term stability analysis of schedules in Autobahn.

A. Evaluation Setup

1) Testbed and Platform: We run our experiments on a 20-node testbed at our local university. This testbed (Fig. 2) covers the top floor of a university building with offices and student lab rooms and thus shares the wireless spectrum with WiFi and Bluetooth communication outside of our control.

2) Metrics, Comparison, and Duration: We evaluate Autobahn in terms of end-to-end reliability, end-to-end latency, and network energy consumption (network duty cycle). We measure these metrics for different routing and retransmission approaches for Master and Autobahn under different interference levels. Moreover, we compare Autobahn with Orchestra according to these metrics. We include six flows we give in Fig. 2. The duration of each experiment in sections IV-E, and IV-D is 75 minutes, with each flow sending 100 packets per minute. In section IV-F we run 75-minute experiments with 60 packets per minute and flow. For the long-term evaluations from Section IV-H onward, we specify the duration as part of the specific experiment.

3) Implementation: We implement Autobahn for Contiki-NG [1] and target the Zolertia Firefly Platform. This platform features a CC2538 Cortex-M3 CPU (32-bit, 32 MHz) with 32 KB of RAM, 512 KB flash storage, and an IEEE 802.15.4 compatible radio.

4) Channels and Interference: We perform most of our experiments under interference. To ensure comparable levels of interference for all tested protocols, we generate these ourselves in a repeatable manner using JamLab [4]. If not stated otherwise, we use an interference level of 10% channel occupancy. We use five interference sources depicted in Fig. 2. Two of the interference sources are in a central position surrounded by several nodes, while the other three are each in close vicinity to a forwarding node in the network. As our testbed only provides the capability of generating interference on one channel at a time, we use only a single channel (channel 26) for all experiments. As we target networks susceptible to wide-band interference, evaluating on only one channel is not a problem. Wide-band interference, such as WiFi, would cover multiple IEEE 802.15.4 channels, eliminating channel hopping advantages. Therefore, it is more realistic to use one channel with interference than multiple channels with interference on only one of them. Moreover, using only one channel lets us compare the worst-case performance of the discussed protocols.

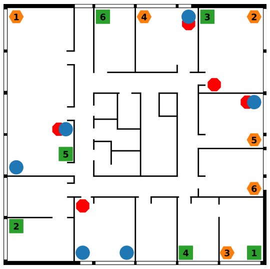


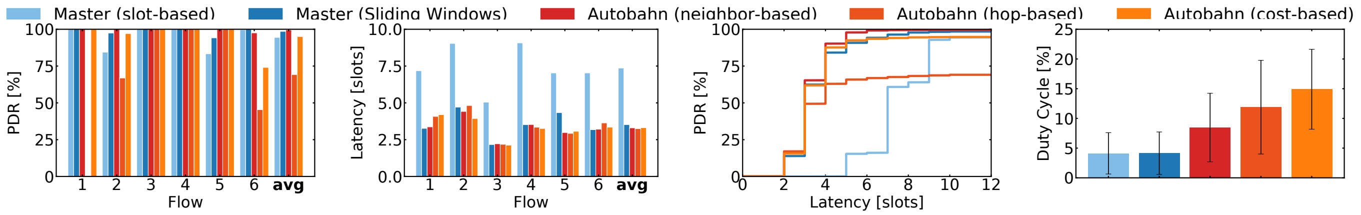
Fig. 2: Local testbed of **500m^2**. Source nodes: orange hexagons; Sink nodes: green squares; Relay-only nodes: blue circles; Red octagons: interferer; Numbers: corresponding flow

5) Application Payload and Overhead: We send packets with a 64-byte randomly generated payload for all experiments, a medium packet size for TSCH. Additionally to this data payload, we include 7-byte and 8-byte routing headers for Master and Autobahn, respectively. Orchestra uses IPv6 headers instead and requires additional network layer control traffic.

6) Routing Sets: We include three Autobahn routing sets marked as neighbor-based, hop-based, and cost-based. The neighbor-based one is option (c) of the neighbor-based routing sets in Section III-B, the one with an alternative path from each node through all neighbors. For the hop-based routing set, we use a slack value of 2. For the cost-based routing set, we use the maximum possible slack value, equaling the maximum cost overhead. This slack value is potentially different for each flow. This value ensures that we include all nodes, with an end-to-end ETX value not exceeding the cost overhead factor. We use overhead factors of 2 and 2.5 as node-overhead factor and cost overhead factor, respectively.

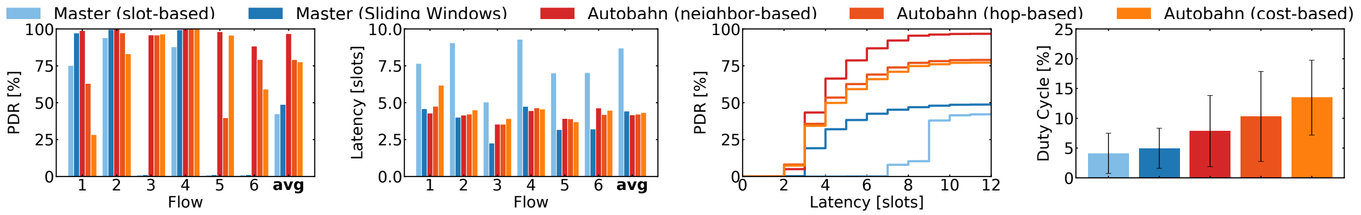
B. Baselines

We compare Autobahn's routing-set algorithms to three other TSCH scheduling policies. Two of these are Master's slot-based retransmission strategy and Master's flow-based transmission strategy called Sliding Windows. Master's slot-based retransmission strategy follows the traditional concept of replicating slots of single hops, done in several recent publications, including AMUS [26]. We use Master, as it provides us an implementation for Contiki-NG. As the last baseline we use Orchestra, to set Autobahn into relation to a well-known protocol. Orchestra [15] is an autonomous scheduler for TSCH included in Contiki-NG [1]. It autonomously maps



(a) Reliability comparison. (b) Latency comparison. (c) CDF: latency and reliability. (d) Duty cycle comparison.

Fig. 3: AUTOBAHN and MASTER without interference. AUTOBAHN's neighbor-based strategy outperforms MASTER while increasing the duty cycle by 4.3 percentage points.



(a) Reliability comparison. (b) Latency comparison. (c) CDF: latency and reliability. (d) Duty cycle comparison.

Fig. 4: AUTOBAHN and MASTER under interference. AUTOBAHN has a much better performance than MASTER with the best performance using the neighbor-based routing set.

links to resources, e.g., determines a node's send or receive slot based on a hash function.

C. Possibility of Synchronous Transmissions in TSCH

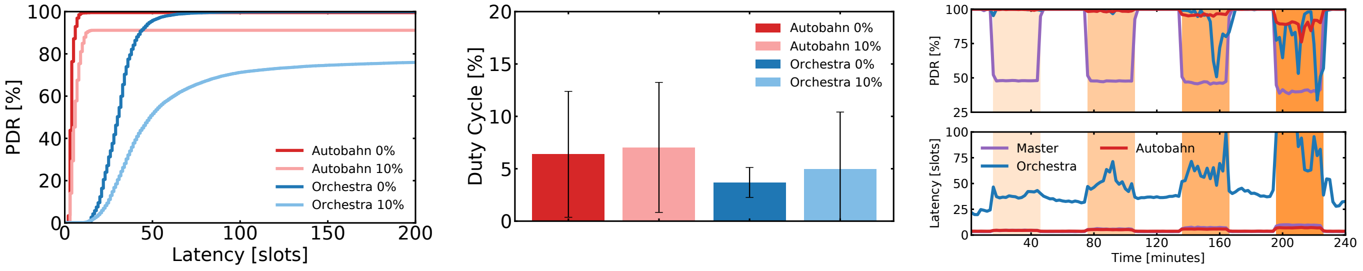
Before starting our main evaluation, we investigate the quality of synchronization in TSCH for synchronous transmissions. With a desk setup of 4 nodes, we can identify the feasibility of synchronous transmissions. Our data shows an average offset between two synchronously transmitting nodes of $16.4\mu s$ with a standard deviation of $16.8\mu s$ and a maximum offset of $65.7\mu s$. This offset clearly shows that the degree of synchronization in Contiki's TSCH implementation is by far not good enough for constructive interference (offset bound of $0.5\mu s$ [17]). However, the offset stays below the maximum offset for capture effect of $160\mu s$ [30]. TSCH generally does not require synchronization as strict as Glossy and therefore does not include additional physical layer time synchronization measures. Nonetheless, our results show that synchronous transmissions are possible due to the capture effect.

D. Performance without Interference

We begin our evaluation by comparing the performance of AUTOBAHN's different routing sets with the performance of MASTER's retransmission strategies. For this evaluation, we do not generate any interference. When comparing the reliability of the different routing sets (see Fig. 3a), we see a generally better performance of the neighbor-based routing set in comparison with the hop-based or cost-based ones. Especially the difference between the neighbor-based and hop-based routing sets is visible for flows 1 and 6. The hop-based routing set has too many simultaneously active nodes at similar distances to a forwarder or receiver. With this number of active nodes, no signal is strong enough for reception through the capture effect. For flow 1, we even see a destruction of the signal. The cost-based strategy has better reliability but generally does not achieve the high reliability of the neighbor-based routing set. The baseline strategies are not exposed to in-flow interference and achieve almost the reliability of neighbor-based AUTOBAHN, with a slight advantage for Sliding Windows over the slot-based retransmission strategy. However, if the network's link qualities are not perfect for a flow (flows 2 and 5), MASTER is more strongly affected. Latency-wise (see Fig. 3b, 3c), AUTOBAHN has a small advantage over MASTER, while both AUTOBAHN and Sliding Windows are better than the slot-based retransmission strategy. The reliability improvement comes at a cost of a higher network duty cycle (see Fig. 3d). With more active nodes, the duty cycle increases significantly. Nevertheless, the best performing routing set of AUTOBAHN leads to the least increase in duty cycle by, on average 4.3 percentage points.

E. Performance under Interference

Next, we compare the same strategies as before under induced interference. From Fig. 4a it is visible that the neighbor-based routing set once again performs best. The other routing sets still offer average reliability of around 80%. However, for some flows, these routing sets achieve very low reliability (e.g., flow 1). This low reliability indicates that too many nodes are active simultaneously plus the additional interference heavily impacting a successful capture effect. Comparing AUTOBAHN to MASTER shows that all routing sets of AUTOBAHN have higher average reliability than MASTER's strategies. This is due to MASTER being heavily impacted by interference in certain flows (flows 3, 5, and 6). In these flows, the shortest path passes closely to an interference source. Thus, we see that additional nodes offered by AUTOBAHN are necessary to route traffic around interference sources opportunistically. The latency differences (see Fig. 4b) follow the same trend as without interference, just slightly higher. As the latency comparison only includes received packets, we also show the



(a) Latency and reliability comparison between Orchestra and Autobahn with and without interference.

(b) Network duty cycle of Orchestra and Autobahn with and without interference.

(c) Performance under and recovery from interference over time. (Interference levels from left to right: **5%**, **10%**, **15%**, and **25%**)

Fig. 5: Comparison of Autobahn and Orchestra with and without interference. Fig. 5c includes the recovery performance of Master's Sliding Windows strategy as an additional baseline.

combination of latency and reliability in Fig. 4c. Autobahn requires a latency of 4 slots to reach a 50% reliability, while Master cannot reach this network-wide reliability at all. The higher overall reliability comes at the cost of a higher network duty cycle (see Fig. 4d). The cost is similarly high as for the interference-free case. However, especially in the presence of interference, an increase of network duty cycle of 2.89 percentage points for the best routing set should be acceptable if reliability has high priority.

F. Autobahn *vs. Orchestra*

Next, we compare Autobahn's best-performing routing set (neighbor-based) with another baseline, the autonomous scheduler Orchestra. As Orchestra is a best-effort protocol and is therefore not limited to a deadline, i.e., a certain number of slots to successfully transmit a packet, we, therefore, relax these limitations for Autobahn as well. However, as we send a new packet every second, Autobahn's schedule should not exceed this value. We achieve this by using a higher scaling factor of three instead of two to determine the number of transmission slots. That means that we use for each flow 50% more transmissions than in the previous experiments. The results of this comparison (see Fig. 5a-5b) show that Orchestra achieves slightly higher reliability than Autobahn without the presence of interference (99.96% vs. 99.51%). However, in the presence of interference, Autobahn clearly outperforms Orchestra with a 13.89 percentage points higher packet delivery rate (PDR). Latency-wise, Autobahn clearly outperforms Orchestra. In case of interference, by a factor 9. However, we need to attribute some of that to the fact that all flows communicate actively simultaneously in Orchestra, while in Autobahn's schedule, the flows communicate one after another. Energy-wise, we can see a similar trend as with the comparison of Autobahn and Master (Fig. 3d and Fig. 4d). Autobahn uses more active nodes and therefore has a higher duty cycle. In contrast both Master and Orchestra follow a single path, with Master's duty cycle being the lowest, while Orchestra occupies the middle ground.

G. *Recovery from interference*

Next, we compare how well Master, Autobahn, and Orchestra perform under interference over time and how good they are at recovering from interference. Fig. 5c shows that all three algorithms are influenced by interference but are successful at recovering independently of the interference level. Master and Autobahn have similar latency responses, while Orchestra has a less uniform curve as the rerouting in case of interference takes some time. Reliability-wise, Orchestra keeps high reliability for quite a long time but has to drop packets towards the end of an interference block. Master is generally hit the hardest by interference, while Autobahn clearly performs best under interference.

H. *Long-term stability of* Autobahn

After comparing Autobahn to various baselines under different interference levels, we evaluate Autobahn's long-term stability. As before, we use Autobahn's neighbor-based routing set. For this section, we use a schedule generated with a neighbor discovery on day one and run this schedule almost daily over 12 days. Moreover, after 12 days, we run a final 25 hour experiment with three-hour blocks of 5%, 10%, 15%, and 25% interference, respectively. While the performance varies within the days, reliability always stays above 96% (see Fig. 6a). Latency fluctuates slightly, yet it remains around the level of the first days, whereas the duty cycle shows an upwards trend in the beginning (see Fig. 6b-6c). The performance analysis over 25 hours (see Fig. 6d) shows that even an almost two-week-old Autobahn schedule is still able to perform under interference and quickly recover from it.

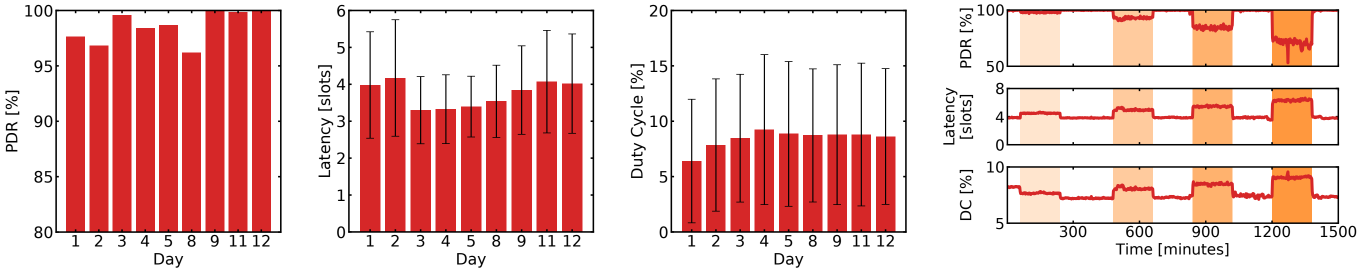
V. Conclusion

Centrally scheduled networks are sensitive to wireless link dynamics, esp. wide-band interference. Autobahn addresses this by adding spatial redundancies via combining TSCH with synchronous transmissions and opportunistic routing.

We show that Autobahn offers reliability of 95% and more under interference while mildly increasing the duty cycle by 4.3 percentage points. Moreover, experiments over 12 days show the long-term stability of Autobahn's schedules with 98.6% reliability.

References

[1] "Contiki-NG: The OS for Next Generation IoT Devices," 2020. [Online]. Available: http://www.contiki-ng.org/



(a) Reliability over the course of 12 days.

(b) Latency over the course of 12 days.

(c) Energy consumption (duty cycle) over the course of 12 days.

(d) Performance over 25 hours with 3 hour long interference blocks (interference levels from left to right: **5%**, **10%**, **15%**, **25%**).

Fig. 6: Long-term stability evaluation of AUTOBAHN. Figures 6a - 6c present (almost) daily runs of AUTOBAHN with the same schedule over the course of 12 days. Fig. 6d shows the performance of the same 12-day old schedule under different interference levels and its recovery from interference.

[2] M. Baddeley, A. Aijaz, U. Raza, A. Stanoev, Y. Jin, M. Schuß, C. A. Boano, and G. Oikonomou, "6tisch++ with Bluetooth 5 and Concurrent Transmissions," in *EWSN*, 2021.

[3] S. Biswas and R. Morris, "ExOR: opportunistic multi-hop routing for wireless networks," in *ACM SIGCOMM*, 2005.

[4] C. A. Boano, T. Voigt, C. Noda, K. Römer, and M. A. Zúñiga, "JamLab: Augmenting Sensornet Testbeds with Realistic and Controlled Interference Generation," in *IEEE IPSN*, 2011.

[5] M. Brachmann, O. Landsiedel, and S. Santini, "Concurrent Transmissions for Communication Protocols in the Internet of Things," in *IEEE LCN*, 2016.

[6] R. Brummet, D. Gunatilaka, D. Vyas, O. Chipara, and C. Lu, "A Flexible Retransmission Policy for Industrial Wireless Sensor Actuator Networks," in *IEEE ICII*, 2018.

[7] D. Carlson, M. Chang, A. Terzis, Y. Chen, and O. Gnawali, "Forwarder Selection in Multi-transmitter Networks," in *IEEE ICDCS*, 2013.

[8] S. Chachulski, M. Jennings, S. Katti, and D. Katabi, "Trading structure for randomness in wireless opportunistic routing," *ACM SIGCOMM Computer Communication Review*, vol. 37, 2007.

[9] T. Chang, M. Vučinić, X. V. Guillén, D. Dujovne, and T. Watteyne, "6TiSCH minimal scheduling function: Performance evaluation," *Internet Technology Letters*, vol. 3, 2020.

[10] T. Chang, T. Watteyne, X. Vilajosana, and P. H. Gomes, "Constructive Interference in 802.15.4: A Tutorial," *IEEE Communications Surveys Tutorials*, vol. 21, 2019.

[11] T. Chang, T. Watteyne, Q. Wang, and X. Vilajosana, "LLSF: Low Latency Scheduling Function for 6TiSCH Networks," in *IEEE DCOSS*, 2016.

[12] R. R. Choudhury and N. H. Vaidya, "MAC-layer anycasting in ad hoc networks," *ACM SIGCOMM Computer Communication Review*, vol. 34, 2004.

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, "Section 24.3: Dijkstra's algorithm," in *Introduction to Algorithms*, 2nd ed. MIT Press and McGraw-Hill, 2001.

[14] D. S. J. De Couto, "High-Throughput Routing for Multi-Hop Wireless Networks," PhD thesis, Massachusetts Institute of Technology, 2004, https://pdos.lcs.mit.edu/papers/grid:decouto-phd/thesis.pdf.

[15] S. Duquennoy, B. Al Nahas, O. Landsiedel, and T. Watteyne, "Orchestra: Robust Mesh Networks Through Autonomously Scheduled TSCH," in *ACM SenSys*, 2015.

[16] S. Duquennoy, O. Landsiedel, and T. Voigt, "Let the tree Bloom: scalable opportunistic routing with ORPL," in *ACM SenSys*, 2013.

[17] F. Ferrari, M. Zimmerling, L. Thiele, and O. Saukh, "Efficient network flooding and time synchronization with Glossy," in *ACM/IEEE IPSN*, 2011.

[18] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele, "Low-power wireless bus," in *ACM SenSys*, 2012.

[19] P. H. Gomes, T. Watteyne, P. Gosh, and B. Krishnamachari, "Competition: Reliability through Timeslotted Channel Hopping and Flooding-Based Routing," in *EWSN*, 2016.

[20] D. Gunatilaka and C. Lu, "Conservative Channel Reuse in Real-Time Industrial Wireless Sensor-Actuator Networks," in *IEEE ICDCS*, 2018.

[21] L. Harms and O. Landsiedel, "MASTER: Long-Term Stable Routing and Scheduling in Low-Power Wireless Networks," in *IEEE DCOSS*, 2020.

[22] R. F. Heile, R. Alfvin, P. W. Kinney, J. P. K. Gilb, and C. Chaplin, "IEEE Standard for Local and metropolitan area networks–Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs) Amendment 1: MAC sublayer," IEEE, Tech. Rep., 2012.

[23] R. T. Hermeto, A. Gallais, and F. Theoleyre, "Is Link-Layer Anycast Scheduling Relevant for IEEE 802.15.4-TSCH Networks?" in *IEEE LCN*, 2019.

[24] I. Hosni and F. Théoleyre, "Adaptive k-cast Scheduling for High-Reliability and Low-Latency in IEEE802.15.4-TSCH," in *Ad-hoc, Mobile, and Wireless Networks*, 2018, vol. 11104.

[25] T. Huynh, F. Theoleyre, and W. Hwang, "On the interest of opportunistic anycast scheduling for wireless low power lossy networks," *Comput. Commun.*, vol. 104, 2017.

[26] Y. Jin, P. Kulkarni, J. Wilcox, and M. Sooriyabandara, "A centralized scheduling algorithm for IEEE 802.15.4e TSCH based industrial low power wireless networks," in *IEEE WCNC*, 2016.

[27] Y. Jin, U. Raza, and M. Sooriyabandara, "BOOST: Bringing Opportunistic ROuting and Effortless-Scheduling to TSCH MAC," in *IEEE GLOBECOM*, 2018.

[28] S. Kim, H.-S. Kim, and C. Kim, "ALICE: autonomous link-based cell scheduling for TSCH," in *IEEE/ACM IPSN*, 2019.

[29] V. Kotsiou, G. Z. Papadopoulos, P. Chatzimisios, and F. Theoleyre, "LDSF: Low-Latency Distributed Scheduling Function for Industrial Internet of Things," *IEEE Internet of Things Journal*, vol. 7, 2020.

[30] O. Landsiedel, F. Ferrari, and M. Zimmerling, "Chaos: versatile and efficient all-to-all data sharing and in-network processing at scale," in *ACM SenSys*, 2013.

[31] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson, "Low Power, Low Delay: Opportunistic Routing meets Duty Cycling," in *ACM/IEEE IPSN*, 2012.

[32] K. Leentvaar and J. Flint, "The Capture Effect in FM Receivers," *IEEE Transactions on Communications*, vol. 24, 1976.

[33] P. Minet, I. Khoufi, and A. Laouiti, "Increasing reliability of a TSCH network for the industry 4.0," in *IEEE NCA*, 2017.

[34] E. Mozaffari Ahrar, M. Nassiri, and F. Theoleyre, "Multipath aware scheduling for high reliability and fault tolerance in low power industrial networks," *Journal of Network and Computer Applications*, vol. 142, 2019.

[35] M. Palattella, N. Accettura, M. Dohler, L. Grieco, and G. Boggia, "Traffic-Aware Time-Critical Scheduling in Heavily Duty-Cycled IEEE 802.15.4e for an Industrial IoT," in *IEEE PIMRC*, 2012.

[36] A. Saifullah, Y. Xu, C. Lu, and Y. Chen, "Real-Time Scheduling for WirelessHART Networks," in *IEEE RTSS*, 2010.

[37] J. Shi, M. Sha, and Z. Yang, "DiGS: Distributed Graph Routing and Scheduling for Industrial Wireless Sensor-Actuator Networks," in *IEEE ICDCS*, 2018.

[38] D. Yuan, M. Riecker, and M. Hollick, "Making 'Glossy' Networks Sparkle: Exploiting Concurrent Transmissions for Energy Efficient, Reliable, Ultra-Low Latency Communication in Wireless Control Networks," in *Wireless Sensor Networks*, 2014, vol. 8354.