

# (POSTER) OVERTAKE: Opportunistic Routing and Concurrent Transmissions for TSCH

Laura Harms<sup>1,2</sup>, Olaf Landsiedel<sup>1,2</sup>

<sup>1</sup>Kiel University, Germany

<sup>2</sup>Chalmers University of Technology, Sweden

{lah, ol}@informatik.uni-kiel.de

**Abstract**—In this paper, we present OVERTAKE, an opportunistic routing protocol for Time-Slotted Channel Hopping (TSCH). OVERTAKE combines (1) opportunistic routing, (2) concurrent transmissions and (3) TSCH. We show that this novel combination enables low-latency, central scheduling withstanding node failures. Our initial results show its ability to withstand node failures of up to 40% of nodes of a flow while keeping minimal latency.

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

## I. INTRODUCTION

**Context:** Applications in the context of the Industrial Internet of Things (IIoT) require high communication reliability and low latency. To achieve this goal, commonly, a central scheduler for TSCH [1] computes optimal routes (single-path) and schedules end-to-end communication flows incorporating possible retransmissions.

**Challenge:** These single-path strategies (e.g. Sliding Windows (MASTER) [2]) achieve the required reliability under usual amounts of interference. However, if a node of a scheduled flow fails, existing protocols fail. Moreover, central schedules are generally computed for worst-case situations, meeting worst-case latencies. In many situations, a communication flow could reach its destination with a fewer number of hops. Using less hops would reduce the average latency and the consumed energy.

Protocols overcoming the challenge of node failures do not forward a packet to a single node but apply an anycast approach instead. Known approaches are opportunistic routing (cf. Landsiedel et al. [3]) as well as flooding-based protocols (e.g., LWB [4]). Opportunistic routing allows flexibility in the forwarding process, as no fixed link has to communicate. A problem of opportunistic routing is choosing a node forwarding the packet. Flooding-based protocols, on the other hand, cover the whole network with a communication. They use highly time-synchronized concurrent transmissions eliminating the selection of a forwarder.

**Approach:** In this paper, we introduce OVERTAKE, an opportunistic routing strategy for Wireless Sensor-Actuator Networks using concurrent transmissions. We specifically target networks utilizing the IEEE 802.15.4 TSCH (Time-Slotted Channel Hopping) MAC protocol [1]. OVERTAKE combines the three approaches of opportunistic routing, concurrent trans-

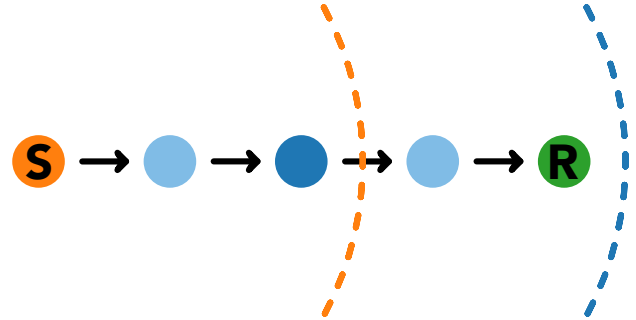


Fig. 1: Nodes included in each slot. The dashed lines represent the transmit ranges of the sender and the second relay node, respectively. If every node can reach the next two nodes, the minimal number of communication slots is 2 resulting in the first 3 nodes active in the first slot and the remaining nodes active from the second slot.

missions, and TSCH. Through this combination, we can create single-path-based schedules withstanding node failures.

We build OVERTAKE on top of the centralized scheduler MASTER [2] and its Sliding Windows transmission strategy. The combination of opportunistic routing mentioned above with flow-based retransmissions allows OVERTAKE not only to withstand node failures but offers a significant robustness improvement over MASTER’s current performance.

Our initial evaluation shows that with OVERTAKE, MASTER can schedule communication even more stable than in its default configuration. Moreover, OVERTAKE reduces the end-to-end latency of communication in environments susceptible to interference. Overall, we make three contributions:

- We present OVERTAKE, a single-path-based concurrent, opportunistic-routing strategy, extending Sliding Windows, achieving lower latency and resilience to node failures.
- We propose acknowledgment supported opportunistic transmissions for TSCH to enable reliable routing of concurrent communication.
- We implement OVERTAKE as a module of MASTER. We present initial results showing OVERTAKE’s latency improvement on MASTER’s schedules. Moreover, we show the resilience of OVERTAKE to node failures.

## II. DESIGN

In this section, we present OVERTAKE’s design. Moreover, we discuss our design modifications of MASTER enabling concurrent transmissions, as well as the TSCH modifications concerning opportunistic routing.

### A. OVERTAKE

OVERTAKE brings concurrent, opportunistic routing to TSCH. OVERTAKE extends MASTER’s transmission strategy of Sliding Windows with concurrent transmissions. We achieve opportunistic routing by sending a packet not to a single specified neighbor, but to all active participants of a flow instead. A flow is the set of nodes a packet traverses from its sender to its receiver. All nodes of a flow that successfully receive a packet, transmit it in the next slot concurrently.

We introduce our opportunistic routing approach in more detail through the following example. In Fig. 1, we show an example flow of 5 nodes with the most cost-effective path taking four hops. However, the communication range of each node covers the next two nodes, as represented by the dashed lines. With opportunistic routing and a schedule like the one given in Fig. 2, a successful transmission might be possible within two slots. The given schedule includes three retransmissions. If a packet overtakes a node, so to say, it arrives at a node earlier than taking each hop of the flow, the remaining section of the flow dynamically gains a retransmission per overtake. These additional retransmissions increase the end-to-end reception likelihood and add flexibility.

As we are combining opportunistic routing with concurrent transmissions, we do not have to select a specific forwarder if several nodes received a packet, nor do we have to use a collision avoidance mechanism.

### B. System Design

We design the OVERTAKE routing strategy for MASTER and TSCH. As both of them do not natively support opportunistic routing or concurrent transmissions, we discuss, in this section, the integration of OVERTAKE into MASTER and TSCH:

1) *MASTER extensions*: We extend MASTER’s routing header by a rank uniquely identifying a node’s position in a flow. This rank allows a receiver to determine whether the received packet is further down the flow or not. Moreover, MASTER sends packets using OVERTAKE directly to a flow address instead of a specific node address. These extensions enable the TSCH anycast described below.

2) *TSCH extensions*: TSCH generally supports broadcast as well as unicast communication. However, to perform opportunistic routing, it is necessary to have support for anycast followed by acknowledgments.

We realize anycast communication in TSCH by transmitting to a flow instead of a neighbor. As multiple nodes belong to a flow, each active node of the flow receiving the packet accepts and potentially acknowledges its reception. If multiple nodes receive a packet, only nodes that are closer to the flow’s receiver, then the current sender should acknowledge the reception to ensure a successful packet forwarding. We

	1	2	3	4	5	6	7
S	TX	TX	TX	TX			
1	RX	RXTX	RXTX	RXTX	TX		
2	RX	RXTX	RXTX	RXTX	RXTX	TX	
3		RX	RXTX	RXTX	RXTX	RXTX	TX
R		RX	RX	RX	RX	RX	RX

Fig. 2: Example Schedule for Overtake with 3 retransmissions according to the topology presented in Fig. 1. A node stays in receive mode (RX) until it receives a packet and transmits (TX) it starting the next slot.

base the decision on the node’s unique rank determined by the central scheduler MASTER.

The multiple nodes accepting the packet, and the modified acknowledgment behavior enable successful anycast communication in TSCH.

## III. EVALUATION

In this section, we evaluate the performance of OVERTAKE in comparison with the Sliding Windows transmission strategy. Moreover, we show OVERTAKE’s possibility to withstand node failures.

### A. Evaluation Setup

1) *Metrics and Comparison*: We evaluate the two transmission approaches, OVERTAKE and Sliding Windows, concerning end-to-end latency and end-to-end reliability. For the proof of concept of withstanding node failures, we present the end-to-end reliability for various numbers of node failures.

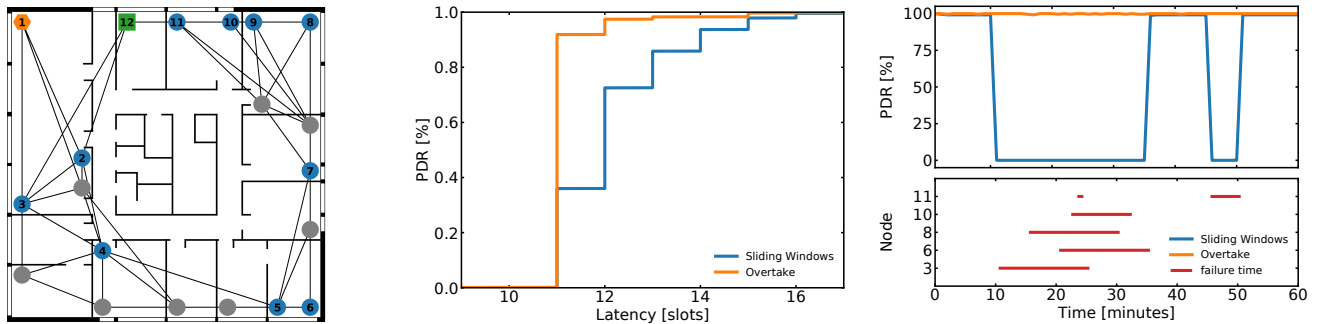
2) *Implementation*: We implement OVERTAKE for Contiki-NG [5]. We target the Zolertia Zoul Firefly platform featuring a 32 MHz 32-bit CC2538 Cortex-M3 CPU, 32 KB of RAM, 512 KB of flash, including an IEEE 802.15.4 compatible radio.

3) *Testbed*: We evaluate on a 20 node, 500 m<sup>2</sup> testbed deployed in student lab rooms and offices, see Fig. 3a. The testbed is exposed to interference, as it shares the wireless spectrum with Bluetooth and WiFi communication outside of our control.

4) *Channels and Application Payload*: Due to the significant amount of interference in the testbed’s environment, we use the default four channel hopping sequence of TSCH (channels 15, 20, 25, and 26). As we use only one flow, the channel offset for all communication is the same. In addition to the frame headers, and an 8-byte routing header, we include a 64 byte randomly generated application payload, a medium packet size supported by TSCH.

### B. OVERTAKE vs. Sliding Windows

We begin our evaluation by comparing the performance of OVERTAKE with the performance of MASTER’s default transmission strategy Sliding Windows. Throughout this evaluation, we use the following scenario. We run experiments with a single flow covering the entire floor in a counterclockwise circle. The flow consists of 11 hops from node 1 to 12 (see Fig. 3a),



(a) A testbed of 20 nodes at Kiel University. Source node: orange hexagon; Sink node: green square; Relay nodes: blue circles; Non participating nodes: grey circles; Numbers: Position in flow; Lines: connectivity > 70%.

(b) Combined latency-reliability CDF of Sliding Windows and OVERTAKE. OVERTAKE outperforms Sliding Windows latency-wise and achieves a 0.25 percentage points higher reliability of 99.80%.

(c) Reliability over time of Sliding Windows and OVERTAKE under node failures. OVERTAKE is capable of withstanding all node failures while Sliding Windows is not able to withstand any node failure.

Fig. 3: Evaluation showing OVERTAKE's superiority over the Sliding Windows transmission strategy.

using a transmit power of -7 dBm. To allow easy comparison between OVERTAKE and Sliding Windows, we use the same schedule for both strategies, created and optimized for Sliding Windows with six possible retransmissions. Contrary to the schedule presented in Fig. 2, the schedule used here, starts with two active nodes and increases the number of active nodes from hop to hop, until it reaches up to 7 active nodes in slot 7. It allows up to 7 simultaneously active nodes and a minimum latency of 11 slots. With this schedule, only a small latency improvement is possible. However, allowing more than 7 active nodes at a time would lead to unrealistic latencies and hop counts due to the circular shape of the scheduled flow. All experiments have a run-time of 1 hour and send a new packet every 250 ms.

Fig. 3b shows the performance of Sliding Windows and OVERTAKE for the presented scenario. While OVERTAKE offers only a small reliability improvement over Sliding Windows of 0.25 percentage points (99.80% vs. 99.55%), OVERTAKE's average latency of 11.1 slots is almost one slot smaller than the average latency of Sliding Windows. Moreover, a significant number of packets were received after the minimal possible latency of 11 slots. As the schedule was not optimized for OVERTAKE, an optimal schedule should result in even lower latency.

### C. Overtake under node failures

After comparing the general performance of OVERTAKE and Sliding Windows, we evaluate their respective performance in case of node failures. The general scenario is the same as described above. However, up to 5 nodes cease operation at different times, leading to up to 5 non-responding nodes at a given time. Fig. 3c shows the results of this experiment. While Sliding Windows is highly affected by any node failure, OVERTAKE is, in the given scenario, not at all influenced by the occurring node failures. It shows that there are at all times enough longer communication distances available, overtaking one or two nodes, which can be used by OVERTAKE while Sliding Windows has to stick to its single-path schedule. If

no or not enough node overtakes were possible, OVERTAKE would be affected by node failures as well. Nevertheless, it is clearly visible that OVERTAKE offers much higher stability compared to Sliding Windows, as long as each node can reach more nodes than its immediate neighbors.

## IV. CONCLUSION & FUTURE WORK

This paper introduces OVERTAKE an opportunistic routing and concurrent communication strategy for (centrally scheduled) TSCH networks. Instead of relying on traditional unicast or broadcast communication, OVERTAKE introduces concurrent opportunistic communication in TSCH. OVERTAKE is capable of reaching high reliability and low latency with early results showing a significant latency decrease compared with Sliding Windows. Moreover, we show OVERTAKE's feasibility of withstanding node failures.

We implement OVERTAKE for MASTER, a centralized scheduler for TSCH based on Contiki-NG. We demonstrate the practicality of concurrent transmissions in the context of TSCH networks and its ability to increase the stability of schedules created by MASTER.

As future work, we intend to extend MASTER by a multi-path communication strategy building upon OVERTAKE called AUTOBAHN. This strategy should not only be able to withstand node failures but even withstand higher levels of interference present in harsh wireless environments.

## REFERENCES

- [1] "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. [Online]. Available: <http://ieeexplore.ieee.org/document/6185525/>
- [2] L. Harms and O. Landsiedel, "Master: Long-Term Stable Routing and Scheduling in Low-Power Wireless Networks," in *IEEE DCSS*, 2020.
- [3] O. Landsiedel, E. Ghadimi, S. Duquennoy, and M. Johansson, "Low Power, Low Delay: Opportunistic Routing meets Duty Cycling," in *IEEE/ACM IPSN*, 2012, pp. 185–196.
- [4] F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele, "Low-power wireless bus," in *ACM SenSys*, 2012, p. 1–14.
- [5] "Contiki-NG: The OS for Next Generation IoT Devices." [Online]. Available: <http://www.contiki-ng.org/>