Performance gap between the optimal and TTN downlink schedulers

Carlos Fernandez Hernandez (INSA Lyon, Inria, CITI)
Oana Iova (INSA Lyon, Inria, CITI),
Fabrice Valois (INSA Lyon, Inria, CITI),
Christelle Caillouet (Université Côte d'Azur, I3S, Inria)
Alexandre Guitton(Université Clermont-Auvergne, CNRS)

LPWAN Days, Rennes 3th July,2025

SUMMARY

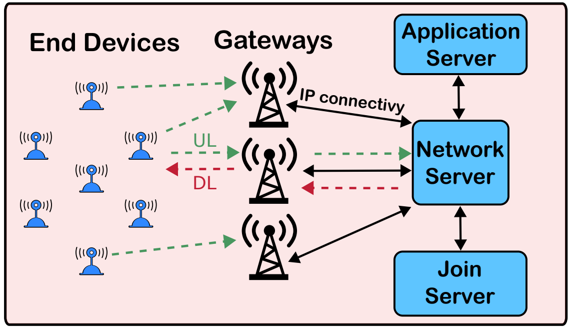
01

01 | Technology context and motivation

1.1 LoRa and LoRaWAN

- **LoRa [1]** defines the modulation and the codification of the data
  - Spreading factor as main parameter
  - Higher SF high receiver sensibility lower data rate (increase time on air)

- **LoRaWAN [2]** defines the protocol and the network architecture
  - Support bidirectional traffic, uplink from the end devices to the network server and **downlink** from network server to end devices
  - LoRaWAN use ALOHA for UL while DL is restricted



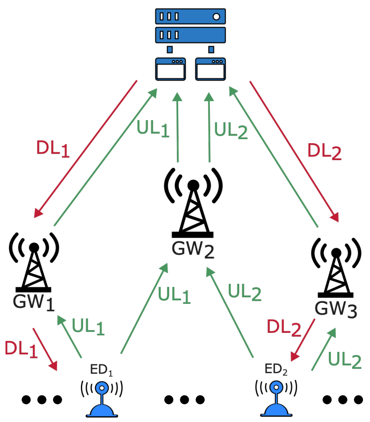
LoRaWAN network architecture

	Channel Frequency	Max Tx Power	Duty Cycle	SF	BW [KHz]
UL	868.1, 868.3, 868.5	14 dBm (27 mW)	1 %	7-12	125
DL(RX1)					
DL(RX2)	869.525	27 dBm (500 mW)	10 %	12	

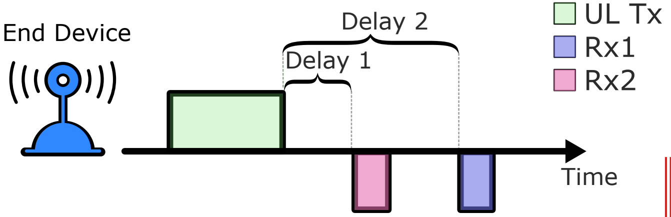
LoRaWAN regional parameters and restrictions for European band EU863-870

1.2 Scheduling problematic

- Downlink and uplink an asymmetrical relation
- Use of downlink traffic is underrated:
  - Gateway is identified as a bottleneck [3][4][5]
  - Use of downlink is discouraged
- Downlink traffic needed for:
  - LoRaWAN (OTAA, ADR, Sync)
  - Reliability (ACK)
  - Firmware Update Over the Air
  - Control of actuators
- Network server is responsible for **downlink scheduling:**
  - **Timing of class A:** Avoid conflicts
  - **Gateway selection:** Optimal use of resources
  - **Duty cycle**: Respect regulations
- **"How does downlink scheduling work in the network server, what is the gap compared to the optimal solution, and how can we approach to this optimal?**



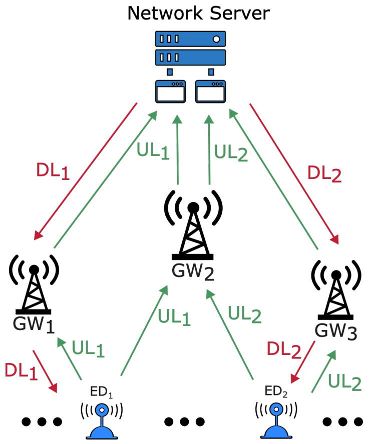
Transmission and reception of UL and DL traffic



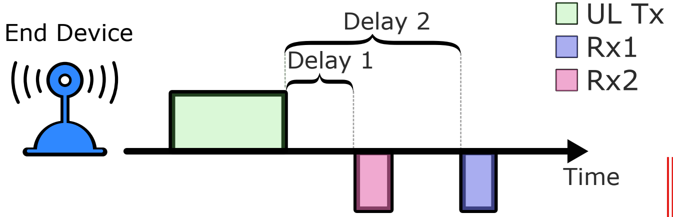
Class A behaviour

1.2 Scheduling problematic

- Downlink and uplink an asymmetrical relation
- Use of downlink traffic is underrated:
  - Gateway is identified as a bottleneck [3][4][5]
  - Use of downlink is discouraged
- Downlink traffic needed for:
  - **LoRaWAN (OTAA, ADR, Sync)**
  - **Reliability (ACK)**
  - **Firmware Update Over the Air**
  - **Control of actuators**
- Network server is responsible for **downlink scheduling:**
  - **Timing of class A:** Avoid conflicts
  - **Gateway selection:** Optimal use of resources
  - **Duty cycle**: Respect regulations
- **"How does downlink scheduling work in the network server, what is the gap compared to the optimal solution, and how can we approach to this optimal?**



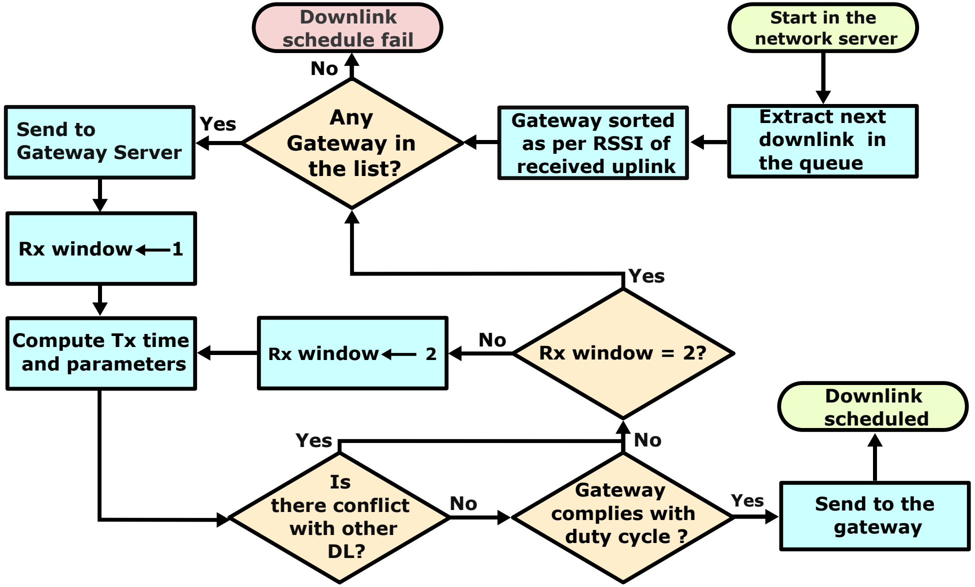
Transmission and reception of UL and DL traffic



Class A behaviour

1.3 The Things Stack downlink scheduling

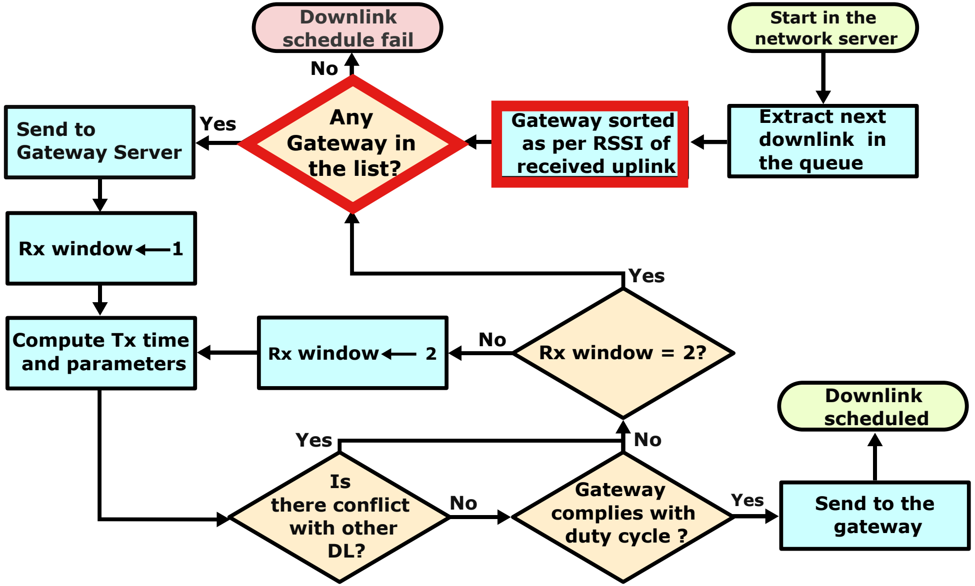
- Performance evaluation between Chirpstack (CS) and The things stack (TTS) [6]
- We choose TTS showed more complete scheduling algorithm
  - Implementation of duty cycle and scheduling conflict detection



The Things Stack scheduling algorithm flow

1.3 The Things Stack downlink scheduling

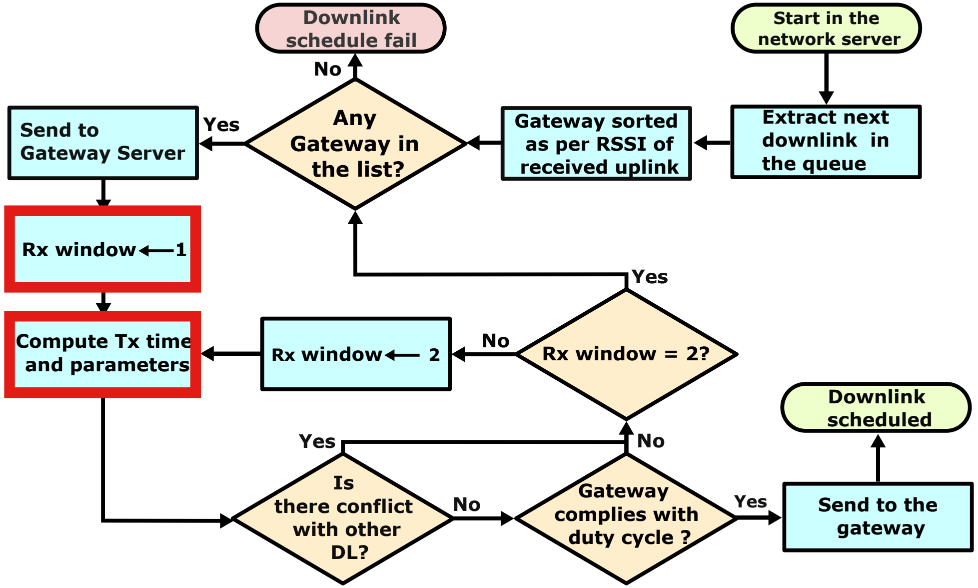
- Performance evaluation between Chirpstack (CS) and The things stack (TTS) [6]
- We choose TTS showed more complete scheduling algorithm
  - Implementation of duty cycle and scheduling conflict detection



The Things Stack scheduling algorithm flow

1.3 The Things Stack downlink scheduling

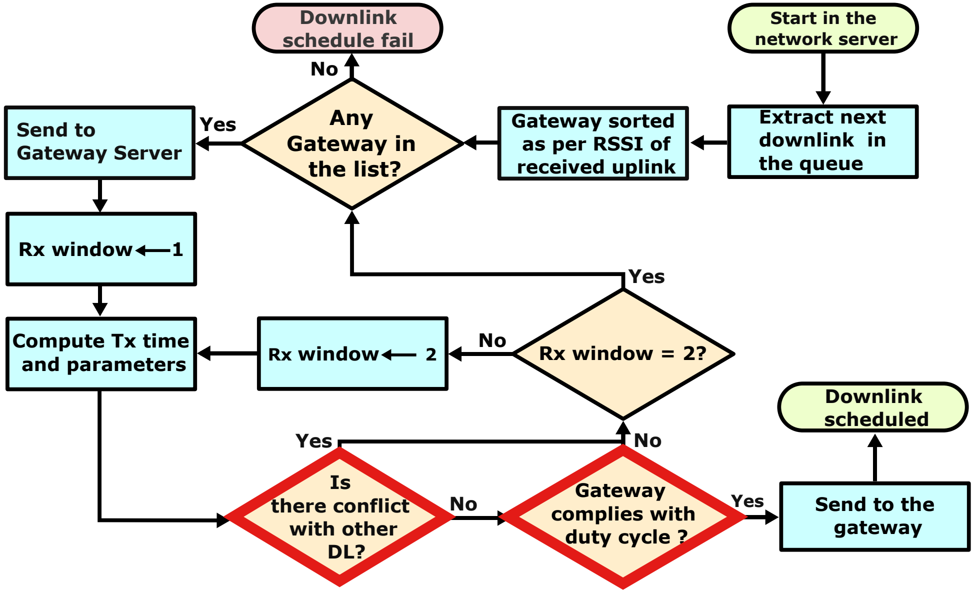
- Performance evaluation between Chirpstack (CS) and The things stack (TTS) [6]
- We choose TTS showed more complete scheduling algorithm
  - Implementation of duty cycle and scheduling conflict detection



The Things Stack scheduling algorithm flow

1.3 The Things Stack downlink scheduling

- Performance evaluation between Chirpstack (CS) and The things stack (TTS) [6]
- We choose TTS showed more complete scheduling algorithm
  - Implementation of duty cycle and scheduling conflict detection



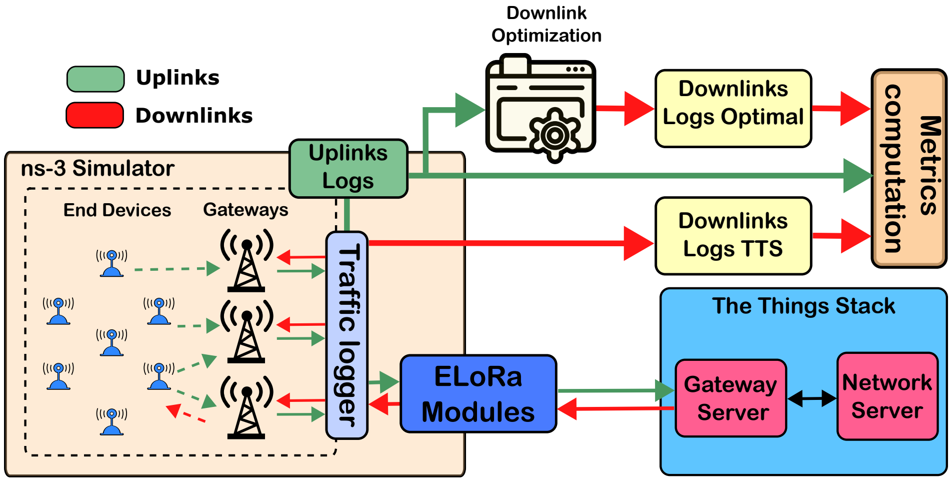
The Things Stack scheduling algorithm flow

02

02 | Performance gap between optimal and TTS

2.1 Experimental methodology for scheduler evaluation

- Quantification of gap between optimal downlink scheduler and TTS
  - Leverage optimal downlink scheduler already available [6]
- ***Fair comparison as same uplink traffic is used for generating downlinks***
- Metrics are defined for comparison and evaluation:
  - Acknowledgement percentage, Receive window usage and uplink frame delivery rate



Framework for comparison optimal Vs TTS

2.2 Optimal scheduler model

- Objective function is to maximise the amount of DL scheduled:

$$\max \sum_{t_j^i \in UL_c} \sum_{g_k \in Gr(t_j^i)} ((1+\alpha)y1(g_k, t_j^i) + y2(g_k, t_j^i))$$

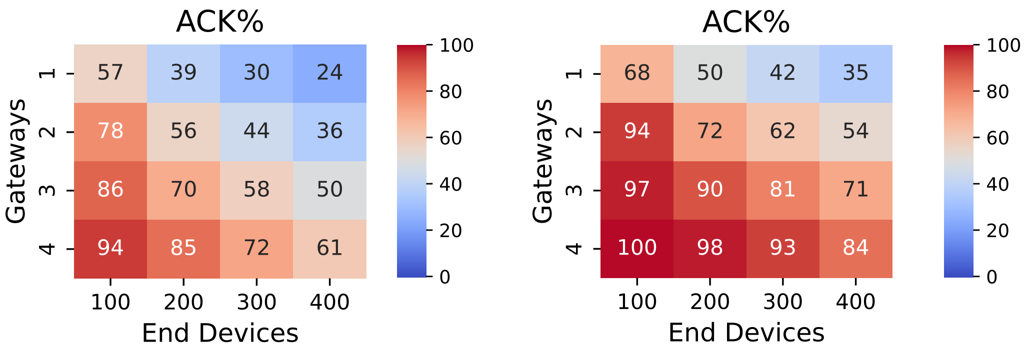
- Variables of interest
  - UL_c: Set of confirmed uplinks
  - t_j^i: Uplink frame j for end device i
  - $y1(g_k, t_j^i)$: Binary variable that is 1 if t_j^i is acknowledge in Rx1 0 if not
  - $y2(g_k, t_j^i)$: Binary variable that is if t_j^i is acknowledge in Rx2 0 if not
- Defines constraints of the Linear Program modelling:
  - **Half duplex property**: t_j^i will be lost if the gateway is already transmitting.
  - **Duty cycle restrictions**: Gateway is blocked after the transmission of a DL
  - **One downlink at the time**: There cannot be two overlapping DL in the same GW

2.3 Emulation setup

- Consider an Area of 1000 m x 1000 m where devices are distributed randomly
- End devices transmit at an average interval of 360 s poisson distribution
- SF configured based on the average Rx power of 20 transmission (ADR)
- Two types of end devices:
  - **High reliability end devices:** All uplink frames need to be acknowledged
  - **End devices with adaptive data rate (ADR):** Periodic confirm packets are transmitted (only 5% of all transmitted packets)
- Scenarios with N end devices (100, 200, 300, 400) and G gateways (1,2,3,4)
- 50 % of end devices are high reliability end devices
- Log distance path loss model

2.4 Results optimal vs TTS: Acknowledgement percentage

- Optimal achieve up to 23 % higher ACK % for the scenario with 400 end devices and 4 gateways and a minimum of 6 % for 100 end device and 4 gateways
  - On multi gateways scenarios Optimal performance is amplified
- Optimal scheduler selects Rx1 or Rx2 according to who maximises the number of DL

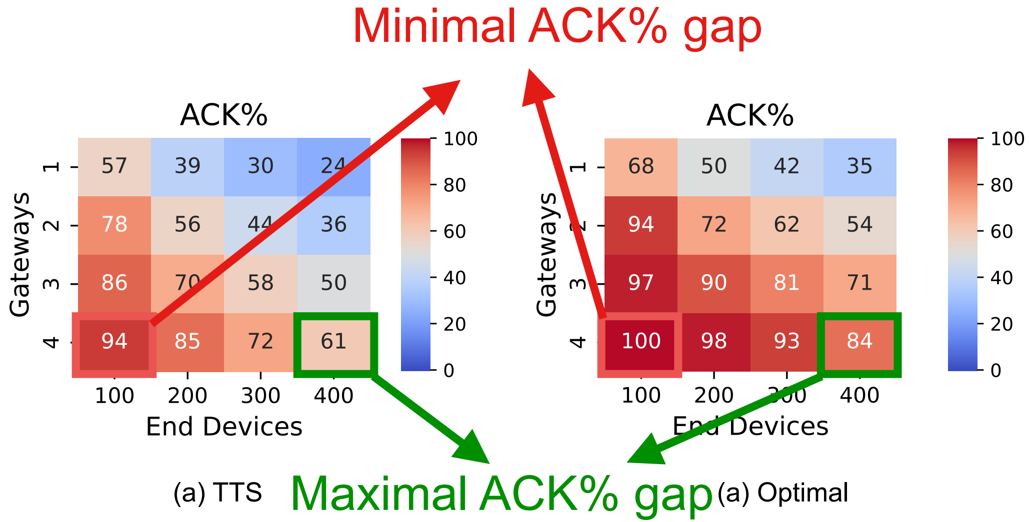


(a) TTS (b) Optimal

Percentage of acknowledgement for optimal and TTS

2.4 Results optimal vs TTS: Acknowledgement percentage

- **Optimal achieve up to 23 % higher ACK % for the scenario with 400 end devices and 4 gateways and a minimum of 6 % for 100 end device and 4 gateways.**
    - On multi gateways scenarios Optimal performance is amplified
- Optimal scheduler selects Rx1 or Rx2 according to who maximises the number of DL

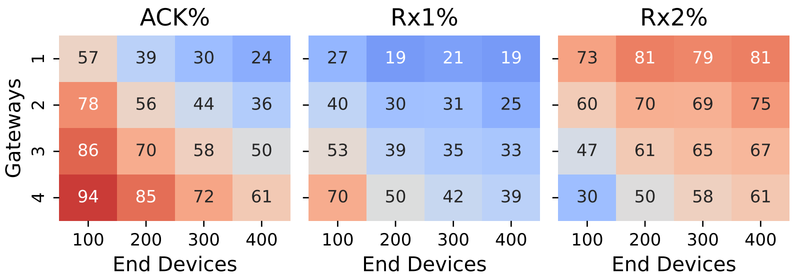


(a) TTS (a) Optimal

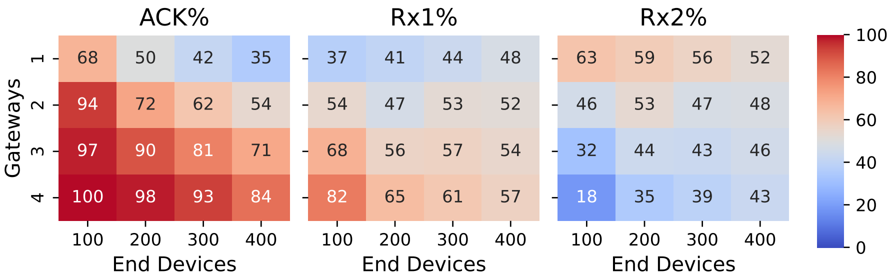
Percentage of acknowledgement for optimal and TTS

2.5 Results optimal vs TTS: Receive window usage

- Optimal scheduler selects Rx1 or Rx2 according to who maximises the number of DL
- TTS always prioritise Rx1 over Rx2
- Optimal has higher use of Rx1 over Rx2 while achieving higher ACK%
  - Is able to locate more packets in Rx1 while having less duty cycle



(a) TTS



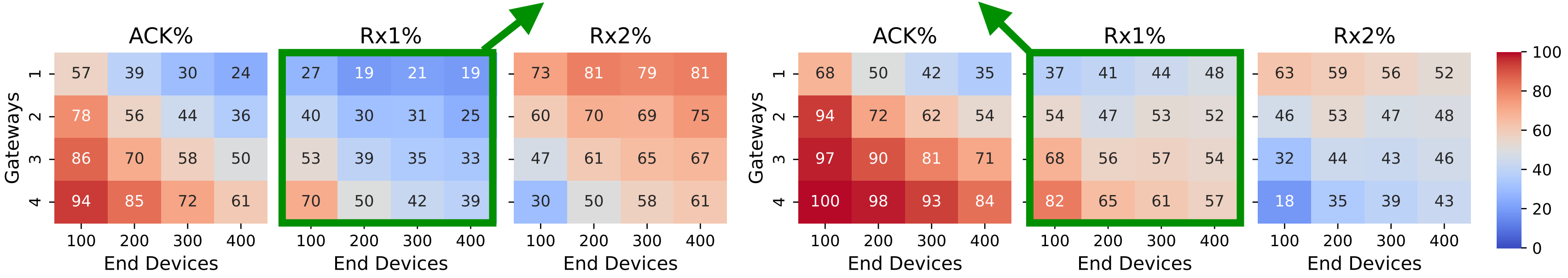
(a) Optimal

Percentage of acknowledgement for optimal and TTS

2.5 Results optimal vs TTS: Receive window usage

- Optimal scheduler selects Rx1 or Rx2 according to who maximises the number of DL
- TTS always prioritise Rx1 over Rx2
- **Optimal has higher use of Rx1 over Rx2 while achieving higher ACK%**
  - **Is able to locate more packets in Rx1 while having less duty cycle**

Maximal 29 % Rx1 usage gap

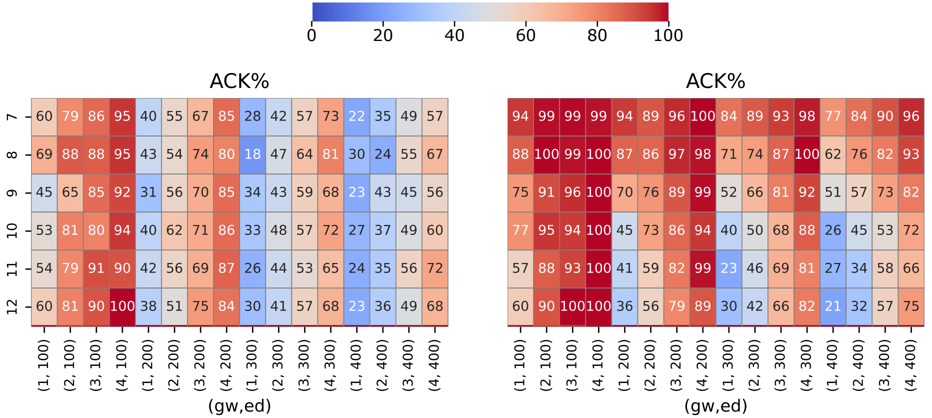


(a) TTS (a) Optimal

Percentage of acknowledgement for optimal and TTS

2.6 Results optimal vs TTS by spreading factor: Acknowledgement percentage

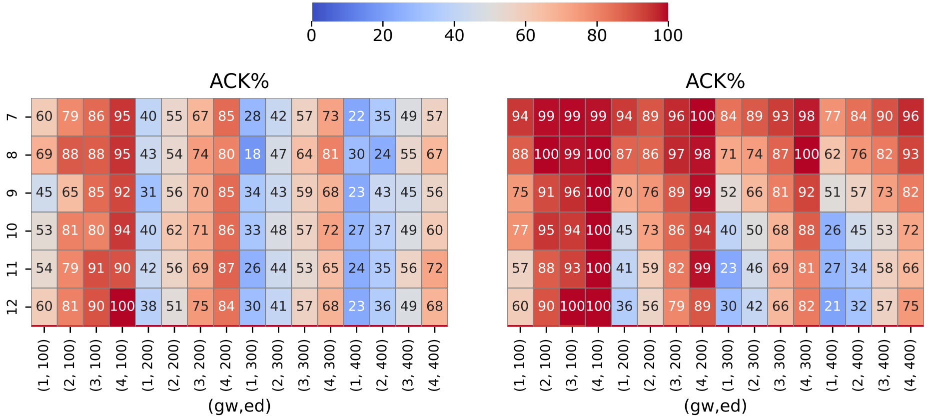
- Optimal performs notably better for low spreading factor
- TTS always prioritise Rx1 over Rx2
  - Unbalance use of receive windows and duty cycle
- Optimal tends to schedule low spreading factor to Rx1 and high spreading factor to Rx2:
  - High SF > 9
  - Low SF <=9



(a) TTS　　(a) Optimal

2.6 Results optimal vs TTS by spreading factor: Acknowledgement percentage

- Optimal performs notably better for low spreading factor
- TTS always prioritise Rx1 over Rx2
  - Unbalance use of receive windows and duty cycle
- Optimal tends to schedule low spreading factor to Rx1 and high spreading factor to Rx2:
  - High SF > 9
  - Low SF <=9

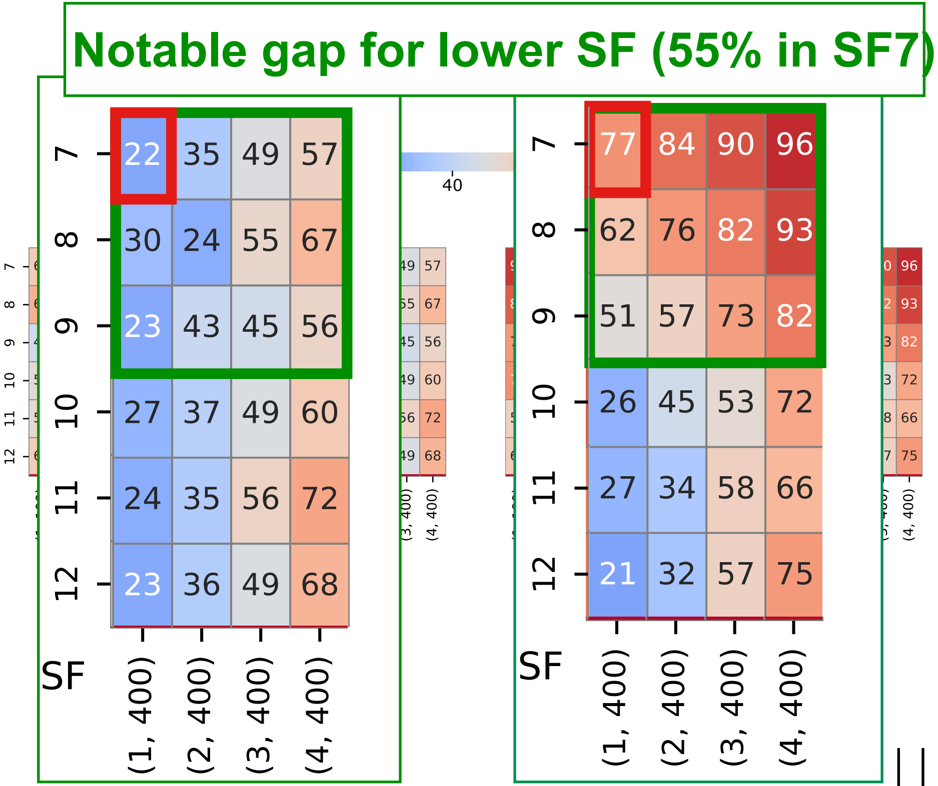


(a) TTS

(a) Optimal

2.6 Results optimal vs TTS by spreading factor: Acknowledgement percentage

- **Optimal performs notably better for low spreading factor**
- TTS always prioritise Rx1 over Rx2
  - Unbalance use of receive windows and duty cycle
- Optimal tends to schedule low spreading factor to Rx1 and high spreading factor to Rx2:
  - High SF > 9
  - Low SF <=9

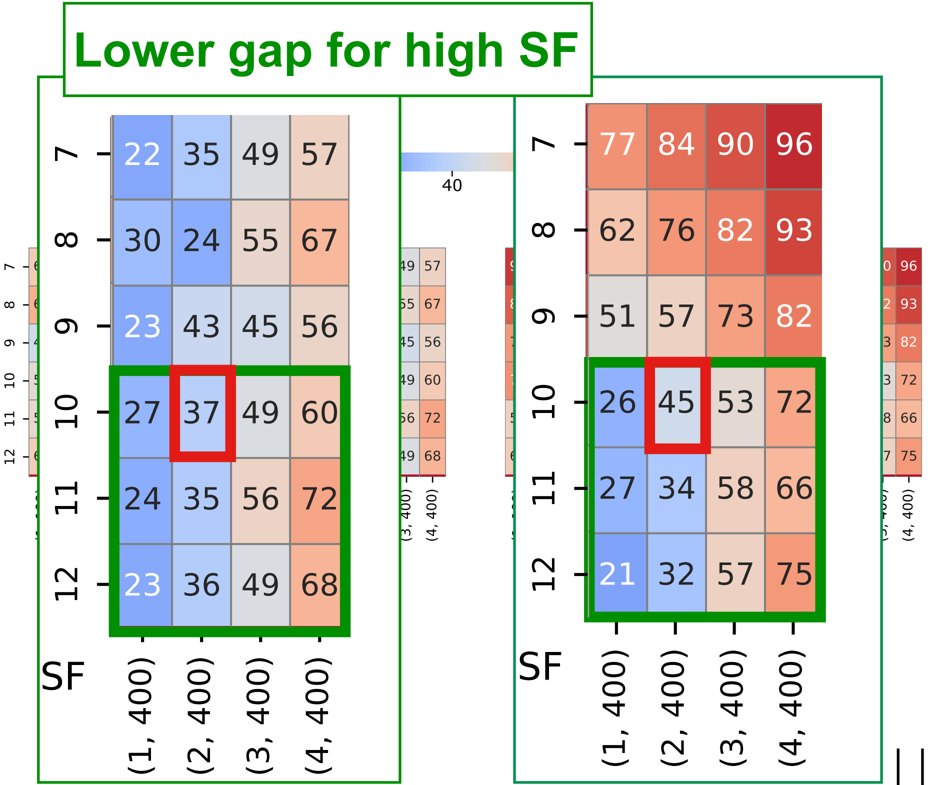


(a) TTS

(a) Optimal

2.6 Results optimal vs TTS by spreading factor: Acknowledgement percentage

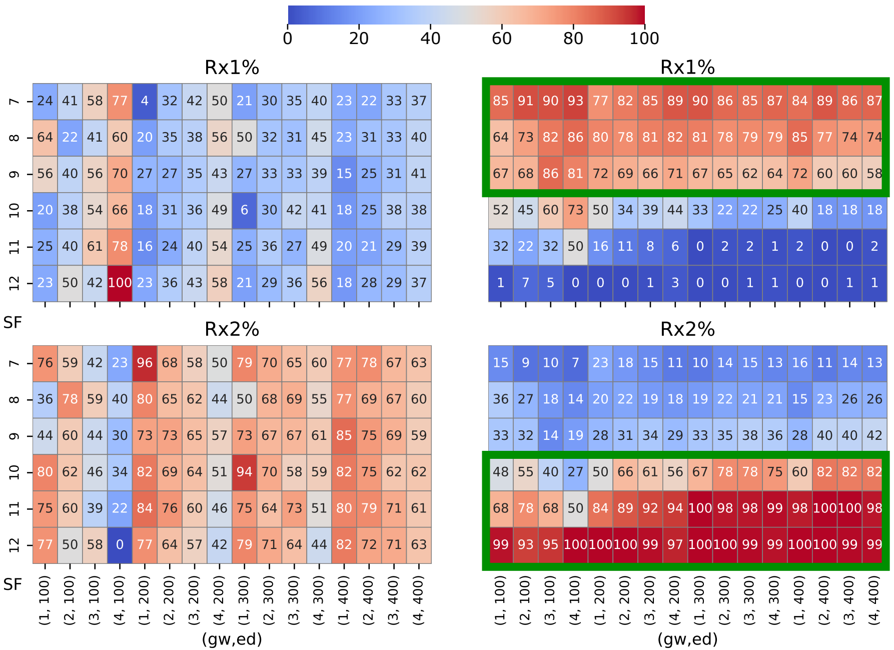
- **Optimal performs notably better for low spreading factor**
- TTS always prioritise Rx1 over Rx2
  - Unbalance use of receive windows and duty cycle
- Optimal tends to schedule low spreading factor to Rx1 and high spreading factor to Rx2:
  - High SF > 9
  - Low SF <=9



(a) TTS (a) Optimal

2.6 Results optimal vs TTS by spreading factor: Receive Window Usage

- **Optimal performs notably better for low spreading factor**
- TTS always prioritise Rx1 over Rx2
  - Unbalance use of receive windows and duty cycle
- Optimal tends to schedule low spreading factor to Rx1 and high spreading factor to Rx2:
  - High SF > 9
  - Low SF <=9

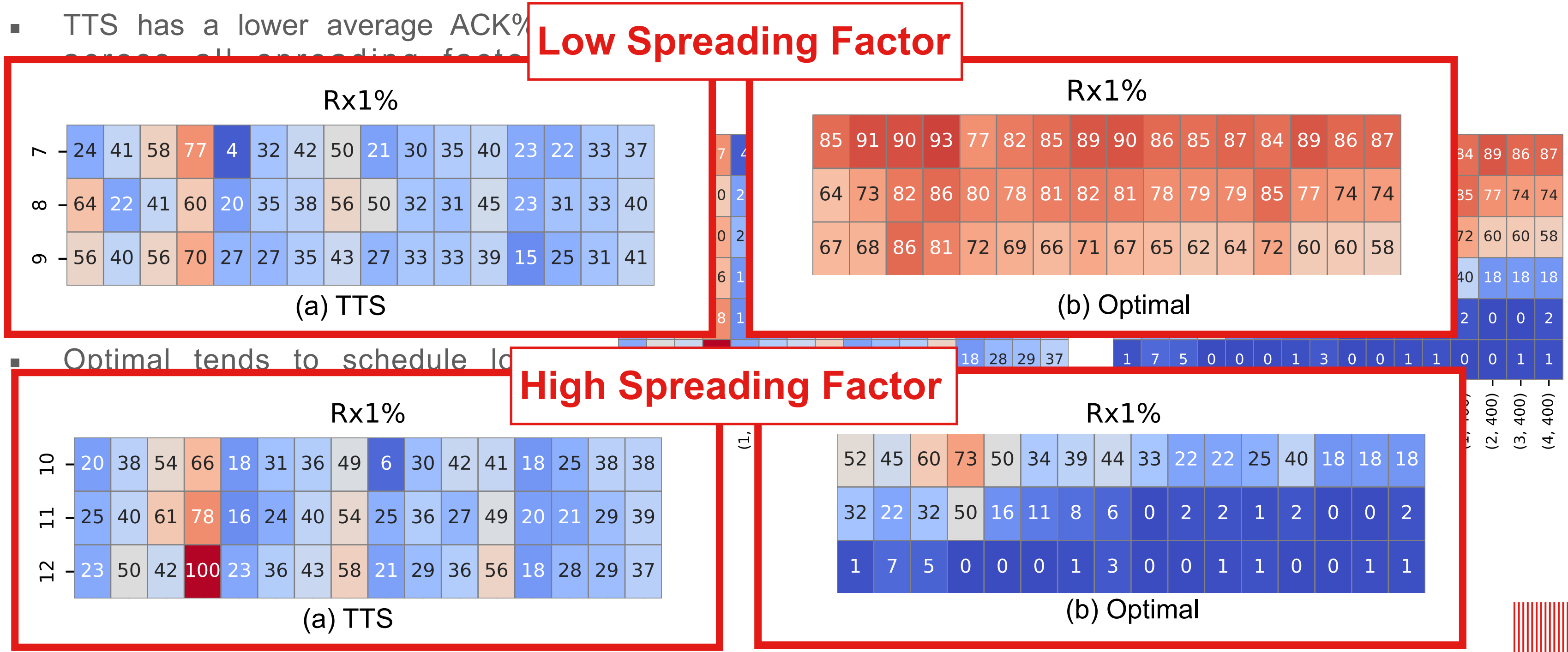


(a) TTS (a) Optimal

2.6 Results optimal vs TTS by spreading factor: Receive Window Usage

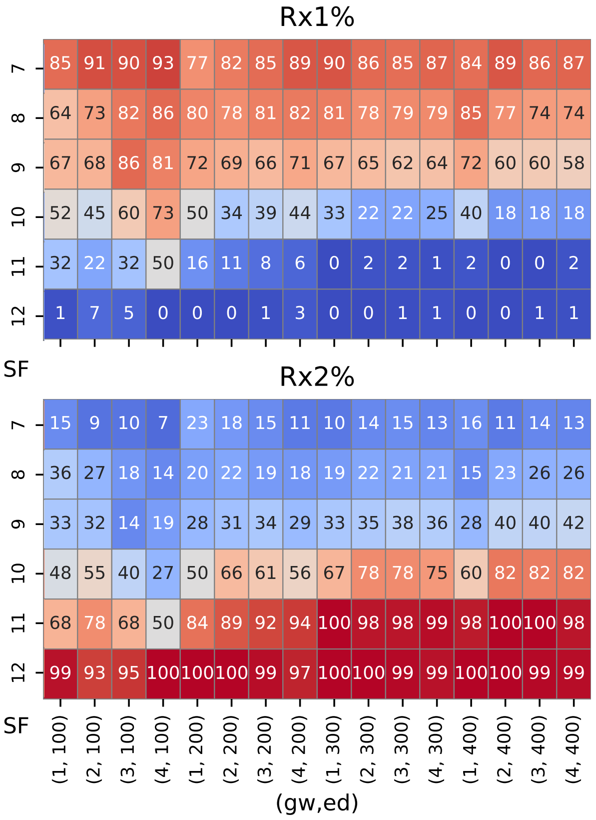
- TTS has a lower average ACK% across all spreading facto

- Optimal tends to schedule lo



2.6 Results optimal vs TTS by spreading factor: Receive Window Usage

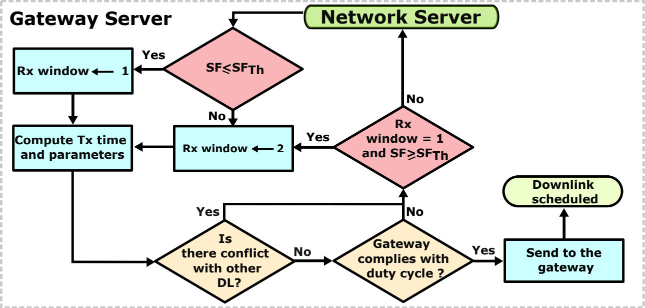
- TTS has a lower average ACK% across all spreading factor compared to the optimal
- Optimal performs notably better for low spreading factor
- TTS always prioritise Rx1 over Rx2
  - Unbalance use of receive windows and duty cycle
- **Optimal tends to schedule low spreading factor to Rx1 and high spreading factor to Rx2:**
  - **High SF > 9**
  - **Low SF <=9**



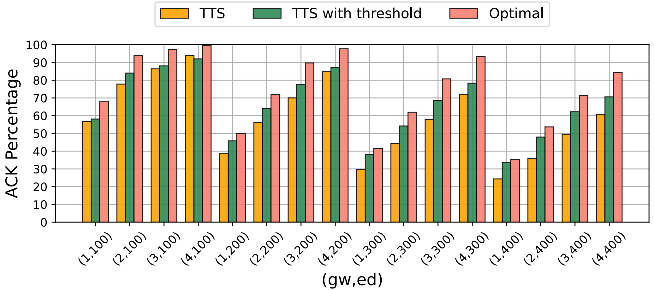
Optimal

2.7 Deterministic scheduler with Spreading Factor threshold

- We proposed modifications to the TTS scheduling algorithm to include a spreading factor threshold:
  - IF SF <= 9 downlink scheduled to Rx1
  - IF SF >=9 downlink scheduled to Rx2
  - IF SF=9 downlink scheduled to Rx1 first and then Rx2
- This will increase the amount of downlinks scheduled in Rx1
  - High spreading factor downlink in Rx1 blocked the gateway for high amount of time (due to high ToA)
- Lightweight implementation achieves up to 12% increase in ACK%



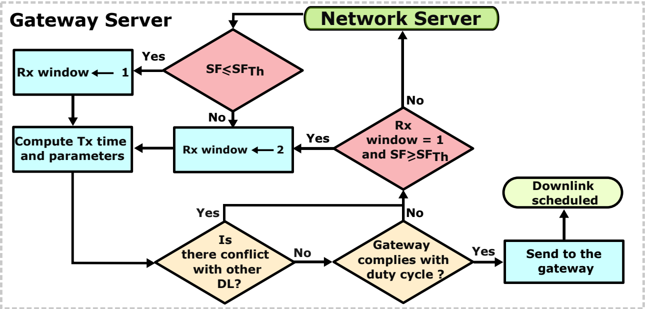
Comparison of TTS (without threshold), TTS (with threshold), and optimal scheduler.



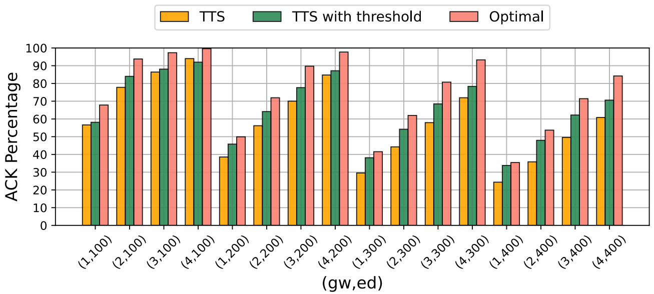
Comparison of TTS (without threshold), TTS (with threshold), and optimal scheduler.

2.7 Deterministic scheduler with Spreading Factor threshold

- **I proposed modifications to the TTS scheduling algorithm to include a spreading factor threshold:**
  - **IF SF < 9 downlink scheduled to Rx1**
  - **IF SF >9 downlink scheduled to Rx2**
  - **IF SF=9 downlink scheduled to Rx1 first and then Rx2**
- This will increase the amount of downlinks scheduled in Rx1
  - High spreading factor downlink in Rx1 blocked the gateway for high amount of time (due to high ToA)
- Lightweight implementation achieves up to 12% increase in ACK%



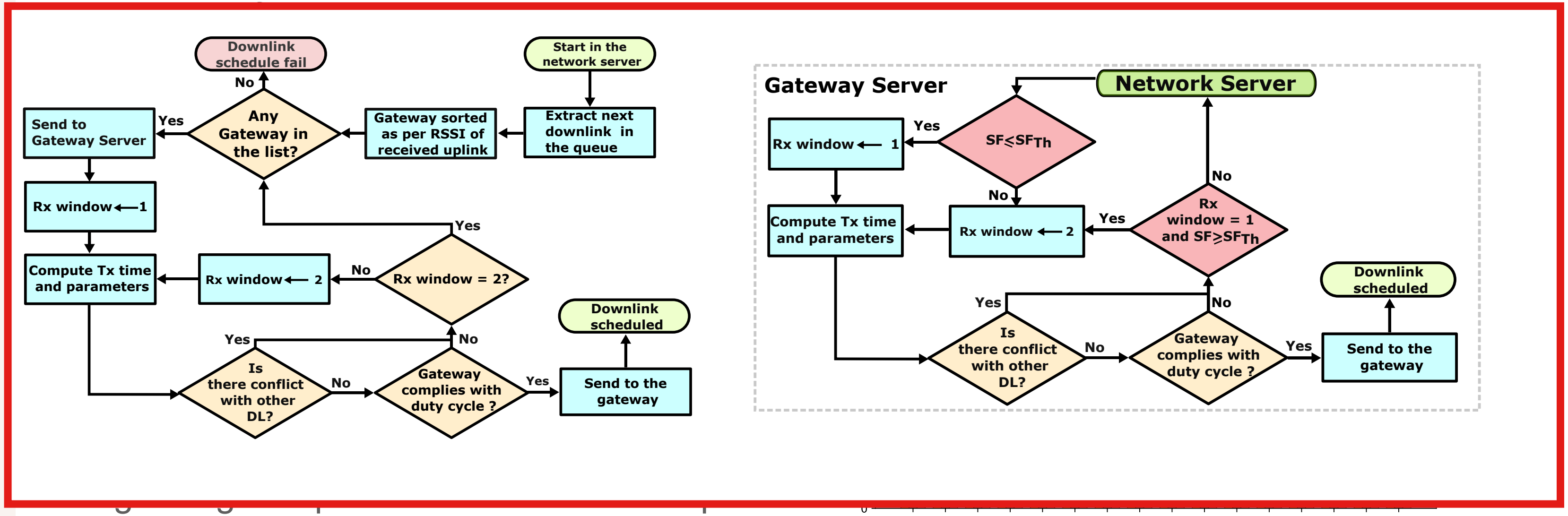
Comparison of TTS (without threshold), TTS (with threshold), and optimal scheduler.



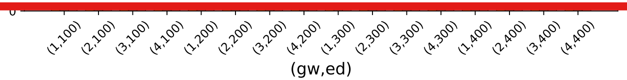
Comparison of TTS (without threshold), TTS (with threshold), and optimal scheduler.

2.7 Deterministic scheduler with Spreading Factor threshold

- I proposed modifications to the TTS scheduling algorithm to include a



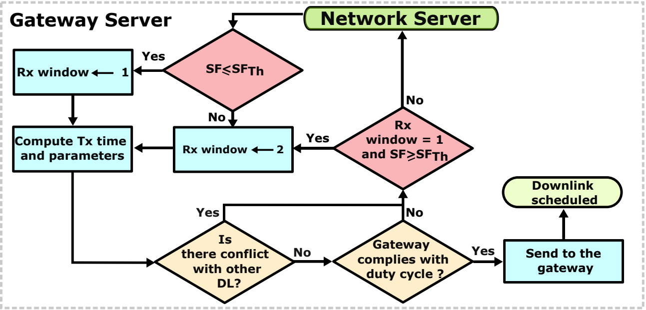
to 12% increase in ACK%



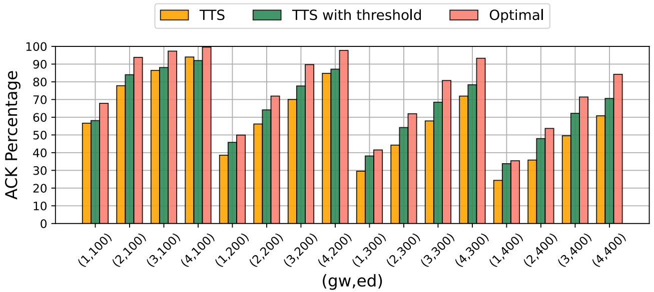
Comparison of TTS (without threshold), TTS (with threshold), and optimal scheduler.

2.7 Deterministic scheduler with Spreading Factor threshold

- I proposed modifications to the TTS scheduling algorithm to include a spreading factor threshold:
  - IF SF <= 9 downlink scheduled to Rx1
  - IF SF >=9 downlink scheduled to Rx2
- This will increase the amount of downlinks scheduled in Rx1
  - High spreading factor downlink in Rx1 blocked the gateway for high amount of time (due to high ToA)
- **Lightweight implementation achieves up to 12% increase in ACK%**



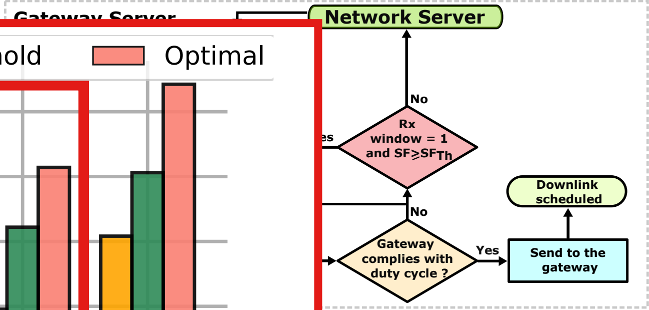
Comparison of TTS (without threshold), TTS (with threshold), and optimal scheduler.



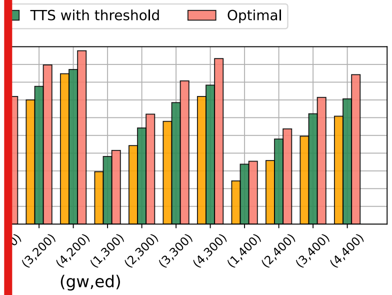
Comparison of TTS (without threshold), TTS (with threshold), and optimal scheduler.

2.7 Deterministic scheduler with Spreading Factor threshold

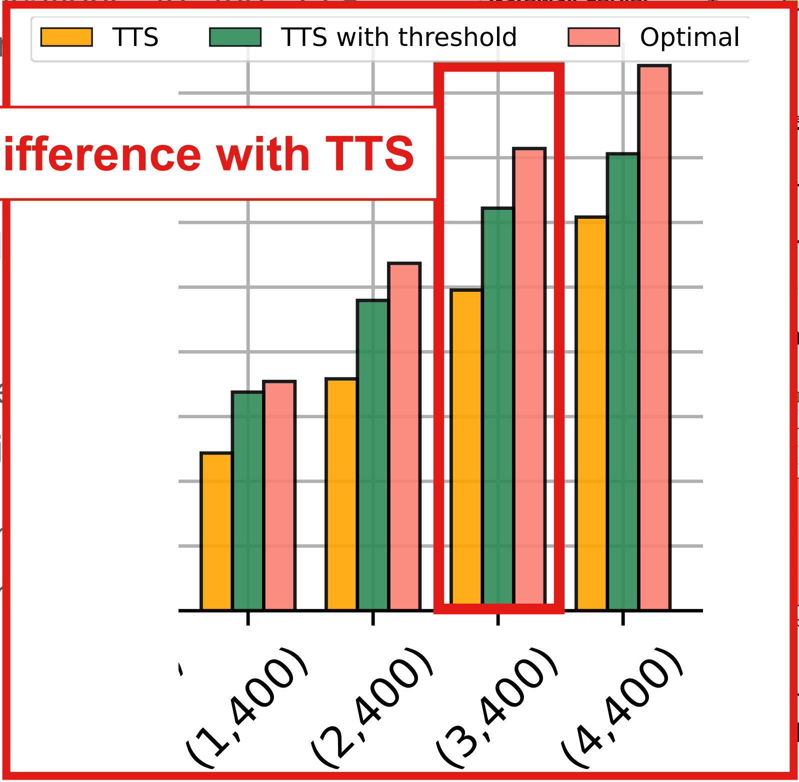
- I proposed modifications to the TTS scheduling algor… spreading factor th…
  - IF S… Rx1…
  - IF SF >=9 d… Rx2
- This will increa… downlinks schedule…
  - High spreadi… Rx1 blocked … amount of tim…
- Lightweight implem… to 12% increase in…



…(without threshold), TTS …nd optimal scheduler.

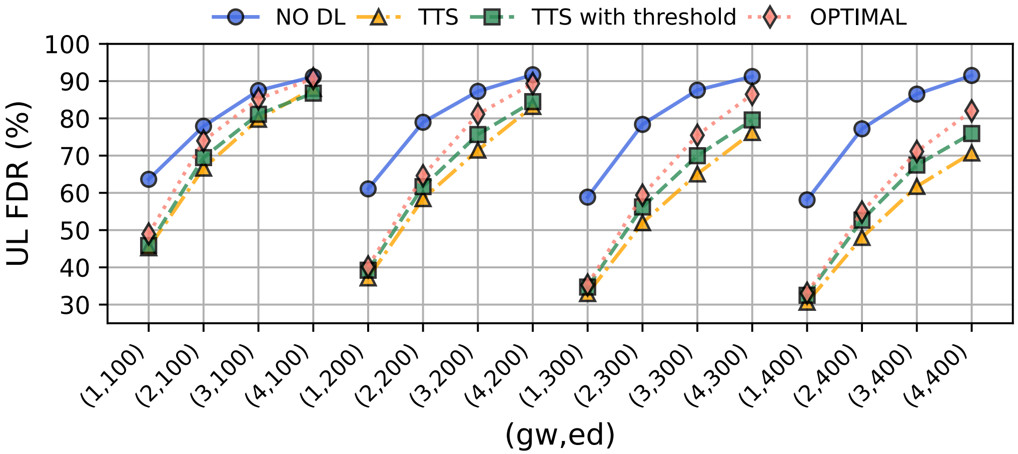


…TTS (without threshold), TTS …ld), and optimal scheduler.



2.8 Impact on Uplink Frame Delivery Rate

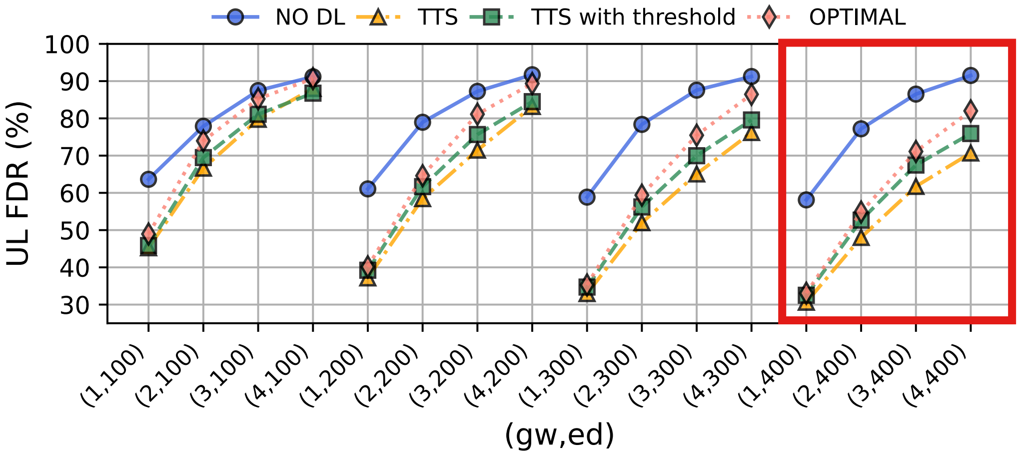
- Uplink Frame Delivery rate always impacted when using downlink traffic
  - Optimal lowers this impact in comparison to TTS up to 12 %
- Proposed modification to scheduler achieves up to 6% better UL FDR than TTS
- Considerable improvement in comparison to the modification made



Uplink Frame Delivery Rate (FDR) for TTS, TTS with threshold, and optimal scheduler. The case without downlink is included as a baseline

2.8 Impact on Uplink Frame Delivery Rate

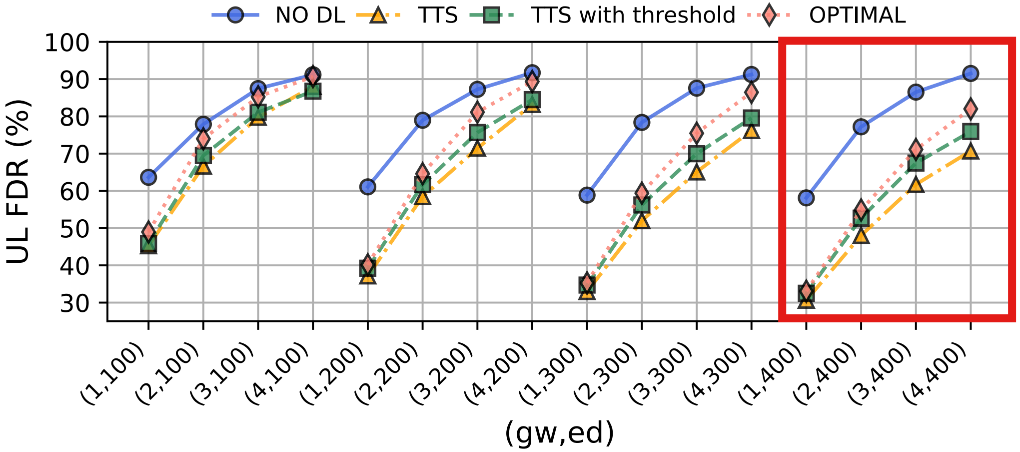
- Uplink Frame Delivery rate always impacted when using downlink traffic
  - **Optimal lowers this impact in comparison to TTS up to 12 %**
- Proposed modification to scheduler achieves up to 6% better UL FDR than TTS
- Considerable improvement in comparison to the modification made



Uplink Frame Delivery Rate (FDR) for TTS, TTS with threshold, and optimal scheduler. The case without downlink is included as a baseline

2.8 Impact on Uplink Frame Delivery Rate

- Uplink Frame Delivery rate always impacted when using downlink traffic
  - Optimal lowers this impact in comparison to TTS up to 12 %
- **Proposed modification to scheduler achieves up to 6% better UL FDR than TTS**
- Considerable improvement in comparison to the modification made



Uplink Frame Delivery Rate (FDR) for TTS, TTS with threshold, and optimal scheduler. The case without downlink is included as a baseline

03

03| Conclusion and perspectives

3.1 Conclusion and perspectives

- Conclusion
  - Comparison with optimal showed that there is a considerable gap for improvement in TTS up to 23 %
    - A first proposal was a deterministic scheduling according to a SF threshold that showed huge improvement compared to cost of implementation
- Future work
  - **It is possible to propose a new scheduling algorithm that solves the flaws of the existing algorithms**
    - Efficient duty cycle use
    - Downlink traffic balance across gateways

References

[1] Semtech, “LoRa technology,” https://www.semtech.com/lora, accessed on: 2023-03-23. LoRa is a registered trademark or service mark of Semtech Corporation or its affiliates

[2] LoRaWAN L2 1.0.4 Specification, TS001-1.0.4, LoRa Alliance, 2020.

[3] A.-I. Pop, U. Raza, P. Kulkarni, and M. Sooriyabandara, “Does bidirectional traffic do more harm than good in lorawan based lpwa networks?” in IEEE GLOBECOM 2017, 2017, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/8254509

[4] F. Van den Abeele, J. Haxhibeqiri, I. Moerman, and J. Hoebeke, “Scalability analysis of large-scale lorawan networks in ns-3,” IEEE Internet of Things Journal, vol. 4, no. 6, pp. 2186–2198, 2017. [Online]. Available: https://ieeexplore.ieee.org/document/8090518

[5] V. Di Vincenzo, M. Heusse, and B. Tourancheau, “Improving downlink scalability in lorawan,” in ICC 2019 - 2019 IEEE International Conference on Communications (ICC), 2019, pp. 1–7. [Online]. Available: https://ieeexplore.ieee.org/document/8761157

[6] C. F. Hernandez, O. Iova, and F. Valois. Downlink scheduling in LoRaWAN: ChirpStack vs The Things Stack. In IEEE LATINCOM, 2024.

[7]. Aimi & al., "ELoRa: End-to-end Emulation of Massive IoT LoRaWAN Infrastructures", 2023 IEEE/IFIP NOMS, 2023

Thank you for your attention

Contact Email: carlos.fernandez-hernandez@insa-lyon.fr