Studia i Materiały Informatyki Stosowanej, Tom1, Nr 1, 2009
str. 21-26

A NEW APPROACH TO MULTICAST TRANSMISSION SCHEDULING IN IOT NETWORKS

Maciej Piechowiak

Uniwersytet Kazimierza Wielkiego
Wydział Informatyki
ul. Kopernika 1, 85-072 Bydgoszcz
e-mail: maciej.piechowiak@ukw.edu.pl

Abstract: *The rapid expansion of the Internet of Things (IoT) has led to the deployment of massive networks connecting thousands of low-power devices that require long-range communication and minimal energy consumption. Among the various technologies enabling such large-scale connectivity, LoRaWAN has emerged as a leading Low Power Wide Area Network (LPWAN) protocol due to its cost efficiency and extensive coverage. However, while LoRaWAN offers scalable uplink communication, the downlink channel remains constrained by limited gateway capacity and duty-cycle restrictions, posing challenges for efficient group communication. To improve downlink performance, this paper proposes a time-slot-based multicast scheduling mechanism that coordinates multicast transmissions across gateways. The proposed approach leverages the spatial distribution of end devices and gateway coverage to identify non-interfering regions, allowing simultaneous multicast transmissions within shared time slots. By implementing this coordination at the network server level, the mechanism achieves reduced collision probability, improved spectral efficiency, and faster data dissemination, particularly beneficial for Firmware Update Over-The-Air (FUOTA) and other multicast-based operations in dense LoRaWAN deployments.*

Keywords: *LoRa, LoRaWAN, IoT, multicast, FUOTA*

NOWE PODEJŚCIE DO PLANOWANIA TRANSMISJI MULTICAST W SIECIACH IOT

Streszczenie: *Gwałtowny rozwój Internetu Rzeczy (IoT) doprowadził do wdrożenia ogromnych sieci łączących tysiące urządzeń o niskim poborze mocy, które wymagają komunikacji dalekiego zasięgu i minimalnego zużycia energii. Spośród różnych technologii umożliwiających taką łączność na dużą skalę, LoRaWAN wyłonił się jako wiodący protokół sieci rozległej o małej mocy (LPWAN) ze względu na swoją opłacalność i szeroki zasięg. Jednakże, podczas gdy LoRaWAN oferuje skalowalną komunikację łącza w górę, kanał łącza w dół pozostaje ograniczony przez ograniczoną przepustowość bramy i ograniczenia cyklu pracy, co stwarza wyzwania dla efektywnej komunikacji grupowej. Aby poprawić wydajność łącza w dół, w niniejszym artykule zaproponowano mechanizm planowania transmisji multicast oparty na szczelinach czasowych, który koordynuje transmisje multicastowe w bramach. Proponowane podejście wykorzystuje rozmieszczenie przestrzenne urządzeń końcowych i zasięg bram w celu identyfikacji niezakłóconych regionów, umożliwiając jednoczesne transmisje multicastowe w ramach współdzielonych szczelin czasowych. Dzięki wdrożeniu tej koordynacji na poziomie serwera sieciowego mechanizm ten osiąga mniejsze prawdopodobieństwo kolizji, lepszą wydajność widmową i szybszą dystrybucję danych, co jest szczególnie przydatne w przypadku aktualizacji oprogramowania sprzętowego przez łącze bezprzewodowe (FUOTA) i innych operacji opartych na transmisji wielokierunkowej w gęstych sieciach LoRaWAN.*

Słowa kluczowe: *LoRa, LoRaWAN, IoT, multicast, FUOTA*

1. INTRODUCTION

In recent years, Low Power Wide Area Networks (LPWANs) have emerged as a key enabler of large-scale Internet of Things (IoT) deployments, connecting millions of low-power devices across cities, industries, and rural areas [5]. Among various LPWAN technologies, LoRaWAN, developed and standardized by the LoRa Alliance, has gained remarkable popularity due to its long-range communication capability, low energy consumption, and cost-effective infrastructure requirements [11,12]. It is now widely adopted in applications ranging from smart metering and environmental monitoring to asset tracking and predictive maintenance.

However, while LoRaWAN provides a scalable and energy-efficient solution for uplink communication (from end devices to network servers), the downlink channel remains a critical bottleneck [1,2]. Downlink transmissions are limited by regulatory duty cycles, gateway capacity, and potential interference in shared ISM frequency bands. As the number of connected devices grows, efficient downlink management becomes increasingly important, especially when broadcasting common messages to large groups of end nodes [3].

To address this challenge, the LoRa Alliance introduced multicast functionality and *Firmware Update Over-The-Air* (FUOTA) specifications, allowing a network server to send a single message simultaneously to multiple devices [10,11]. This mechanism greatly improves efficiency for group commands or firmware distribution. Yet, in dense LoRaWAN deployments, simultaneous downlink multicast transmissions may still cause inter-gateway collisions, reducing the overall reliability of message delivery.

This paper proposes a time-slot-based multicast scheduling mechanism for LoRaWAN, aimed at coordinating downlink multicast transmissions across gateways. The key idea is that nodes belonging to the same multicast group may be served by different gateways operating on distinct frequency channels. By intelligently assigning multicast transmissions to time slots, and reusing slots among non-interfering gateways, the network server can minimize collisions and maximize downlink throughput. The proposed concept aligns with existing *LoRa Alliance* specifications [12] and could be integrated into network-level optimization strategies for large-scale IoT systems.

2. OVERVIEW OF LORA AND LORAWAN ARCHITECTURE

LoRa (*Long Range*) is a proprietary physical layer modulation scheme developed by Semtech Corporation [13]. It is based on chirp spread spectrum (CSS) modulation, which provides exceptional link budget performance and robustness against interference. The use of chirp signals allows LoRa to achieve long communication ranges (often exceeding 10 km in rural environments) while maintaining low power consumption, making it ideal for battery-powered IoT devices.

LoRa operates in unlicensed Industrial, Scientific, and Medical (ISM) bands – typically at 868 MHz in Europe, 915 MHz in North America, and 433 MHz in parts of Asia [14]. The data rate is adjustable and determined by the spreading factor (SF), ranging from SF7 to SF12. A higher SF increases range and link robustness at the cost of lower data throughput and longer airtime. This adaptive mechanism is managed through *Adaptive Data Rate* (ADR), which optimizes performance and energy efficiency based on link conditions.

2.1. LoRaWAN Network Architecture

Building on the LoRa physical layer, LoRaWAN defines the *Media Access Control* (MAC) and network layer protocols that enable scalable, secure, and bi-directional communication between devices and applications. The standard is maintained by the *LoRa Alliance*, an open, non-profit association of companies promoting global LoRaWAN interoperability.

A typical LoRaWAN network consists of four main elements:

- end devices (nodes): low-power IoT sensors or actuators communicating via LoRa modulation.
- gateways: radio concentrators that receive LoRa packets from multiple end devices and forward them via IP to the network server; gateways are typically transparent relays and do not process MAC logic.
- network server: the central intelligence of the network that handles device registration, deduplication of packets received by multiple gateways, adaptive data rate control, and scheduling of downlink messages.
- application server: the destination for application-layer data (e.g., sensor readings or control commands).

This architecture is often referred to as a star-of-stars topology, where end devices communicate directly with one or more gateways, which in turn connect to a centralized network server.

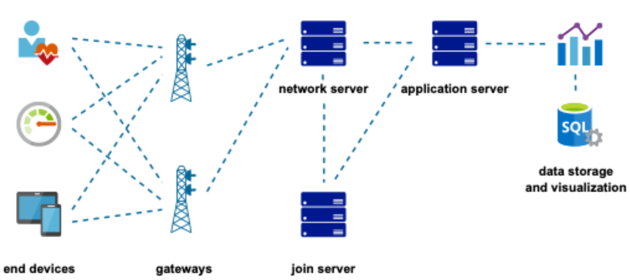

Figure 1. LoRaWAN network architecture (source: author's own work).

2.2. Device Classes and Downlink Behaviour

LoRaWAN supports three classes of end devices, defined to balance latency and energy consumption:

- *Class A*: the default mode for all devices; each uplink transmission is followed by two short receive windows, during which the device listens for possible downlink messages.
- *Class B*: devices open scheduled receive windows synchronized via periodic beacons transmitted by the network; this allows the network server to plan downlink transmissions more predictably.
- *Class C*: devices keep their receiver open continuously, allowing immediate downlink delivery but at the cost of higher power consumption.

The multicast functionality, which enables a single downlink message to be received by multiple end devices simultaneously, is typically implemented using *Class B* or *Class C* devices, since they can be synchronized or continuously listening.

2.3. Multicast and FUOTA in LoRaWAN Specifications

To enable efficient group communication, the LoRa Alliance defined two complementary specifications:

- *LoRaWAN Remote Multicast Setup Specification v1.0.0* [9,10] – this document describes how a network server can remotely create and manage multicast groups. Each group is associated with a multicast address, session keys, and specific parameters such as data rate, frequency, and device class. The mechanism allows multiple devices to share a common downlink session, enabling simultaneous reception of identical messages.
- *LoRaWAN Firmware Update Over-The-Air(FUOTA) Specification v1.0.0* [16] – FUOTA builds upon multicast delivery by allowing large data objects (such as firmware images) to be fragmented, transmitted via multicast, and reassembled by devices. Additional companion specifications, such as the *Fragmented Data Block Transport* (FDT) specification, define the format and reliability mechanisms for segmented transmission.

Together, these specifications provide a framework for efficient and scalable downlink data distribution. However, they primarily assume independent multicast scheduling per gateway, without inter-gateway coordination. In dense networks with overlapping gateway coverage, this can result in multicast transmission collisions and inefficient use of available spectrum.

The proposed time-slot-based multicast scheduling mechanism introduced in this paper builds upon these specifications by introducing temporal coordination between gateways. By analyzing which nodes in a multicast group are served by distinct gateways, the network server can determine safe slot reuse opportunities – effectively parallelizing downlink delivery across non-interfering regions.

3. LORA ALLIANCE MULTICAST AND FUOTA MECHANISMS

The *LoRa Alliance* has developed several complementary specifications extending the LoRaWAN protocol to support efficient group communication and over-the-air firmware updates. These include the *LoRaWAN Remote Multicast Setup Specification v1.0.0* [9], the *Fragmented Data Block Transport Specification v1.0.0* [8] and the *Firmware Management Protocol Specification v1.0.0* [16]. Together, these documents define a comprehensive framework that allows network operators to deliver large data objects, such as firmware images, to many devices simultaneously while ensuring security, reliability, and compliance with regional duty-cycle constraints.

The *Remote Multicast Setup* (RMS) specification defines the procedure for creating and managing multicast groups within a LoRaWAN network. Using dedicated MAC-layer commands, a network server can remotely configure end devices to join one or more multicast groups. Each group is associated with a unique multicast address, a network session key, and an application session key, which together secure the communication and ensure data confidentiality and integrity. The server also specifies parameters such as the frequency channel, data rate, and device class to be used for multicast reception. Once configured, all devices assigned to a multicast group can receive a single downlink message sent to the corresponding multicast address. This mechanism enables simultaneous delivery of identical content to many nodes, significantly reducing network load compared to unicast transmission. However, it assumes that multicast scheduling is managed individually by each gateway, without inter-gateway coordination.

While multicast allows the distribution of small payloads, large objects such as firmware images exceed the maximum

LoRaWAN frame size. To address this limitation, the *Fragmented Data Block Transport* (FDT) specification introduces a protocol for segmenting large data objects into smaller fragments that can be transmitted within standard LoRaWAN frames. Each fragment carries an index number and a portion of the payload, allowing devices to reconstruct the complete object after receiving all fragments. To improve reliability in lossy radio environments, FDT supports redundancy through forward error correction, enabling devices to recover missing fragments without requiring retransmissions. The fragmentation session is initialized and managed through MAC-layer commands that define parameters such as fragment size, total fragment count, and redundancy ratio.

Building on RMS and FDT, the *Firmware Update Over-The-Air* (FUOTA) specification [16] defines a complete process for distributing and applying firmware updates to groups of devices. FUOTA sessions generally consist of three sequential phases: setup, fragmentation, and application. During the setup phase, the network server configures multicast groups on all target devices using RMS procedures. The fragmentation phase involves the transmission of the firmware image in the form of data fragments sent over the established multicast channel according to FDT rules. Finally, during the application phase, the *Firmware Management Protocol* (FMP) governs the validation and activation of the new firmware image once all fragments have been successfully received and reassembled. This process allows many devices to be updated simultaneously while minimizing downlink airtime and gateway usage.

Despite the efficiency of this mechanism, current *LoRa Alliance* specifications do not define how multicast transmissions should be temporally coordinated across multiple gateways. In dense LoRaWAN deployments, where the coverage areas of several gateways overlap, independent scheduling can lead to simultaneous downlink transmissions on the same frequency and spreading factor (SF). This, in turn, increases the probability of inter-gateway collisions and reduces overall downlink reliability. The specifications assume that multicast traffic is scheduled locally by each gateway, which is sufficient for sparse networks but suboptimal in large-scale, high-density scenarios.

To address this gap, the concept proposed in this paper introduces an additional layer of coordination between gateways, implemented at the network server. The idea is that the server can analyze the spatial distribution of devices within a multicast group and determine which gateways are responsible for serving them. If the group's devices are distributed among gateways that do not interfere with each other, the same multicast transmission slot can be reused across those gateways. By assigning multicast transmissions to specific time slots and ensuring synchronization among gateways, the network server can effectively parallelize multicast delivery while minimizing collisions (Fig. 2). This time-slot-based multicast scheduling mechanism preserves compatibility with existing LoRaWAN standards while introducing an intelligent coordination layer that increases throughput, reduces delivery time, and improves the reliability of multicast-based firmware distribution.

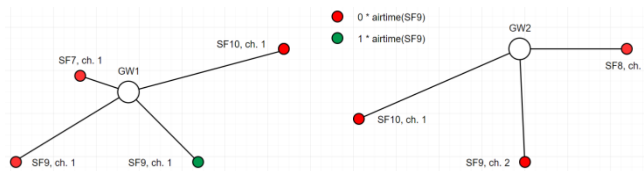

Figure 2. Example scenario illustrating time-slot allocation for multicast transmissions where nodes are served by different gateways (source: author's own work).

4. ALGORITHM OVERVIEW: TIME-SLOT-BASED MULTICAST SCHEDULING

The proposed algorithm operates within the network server and is designed to coordinate multicast downlink transmissions across gateways by assigning them to non-overlapping time slots. It considers differences in spreading factors (SFs), packet airtime, and device transmission intervals to achieve collision-free and efficient downlink delivery. The algorithm proceeds as follows:

1. Input Data Collection:
   The network server collects all relevant information about the multicast session, including the list of participating end devices, their assigned gateways, radio configuration (available frequency channels, spreading factors, and data rates), as well as regional constraints such as maximum duty cycles and transmission power limits. The algorithm assumes that end devices transmit uplinks at fixed intervals (e.g., every 15 minutes), allowing the network server to predict available downlink opportunities.
2. Gateway–Device Association Mapping:
   For each multicast group, the server identifies which gateways can communicate with the group's devices. This mapping is derived from uplink reception data and link quality metrics such as RSSI and SNR. A device may be served by multiple gateways, but one is typically selected as the primary downlink gateway for multicast scheduling.

3. Spreading Factor Assignment:
   Devices in the network may operate using different spreading factors (SF7–SF12), depending on their distance and link quality. The server groups devices by SF to ensure that multicast packets are transmitted using the correct physical layer configuration for each subgroup. This step guarantees that all devices in a multicast subgroup can decode the transmitted frames.
4. Packet Fragmentation:
   The multicast payload (e.g., firmware image or configuration data) is divided into fixed-length packets of identical size. Fragmentation ensures consistent airtime calculations and simplifies scheduling across gateways. Each fragment is transmitted sequentially according to the assigned slot plan.
5. Airtime and Slot Duration Calculation:
   For each spreading factor used in the multicast session, the network server computes the corresponding airtime based on LoRa modulation parameters (SF, bandwidth and coding rate). The slot duration for a given SF is then defined as the maximum expected airtime for a multicast fragment plus a small guard interval to account for synchronization tolerance. Different SF groups may therefore have different slot lengths.
6. Interference Analysis:
   The server constructs an interference model to determine which gateways may cause mutual interference. Two gateways are considered interfering if they serve devices in overlapping areas or operate on the same frequency and spreading factor with overlapping coverage. This information is used to build an interference graph representing potential conflicts.
7. Slot Allocation Across Gateways:
   Based on the interference graph, the network server assigns each gateway to a specific time slot. Gateways that do not interfere with one another are placed in the same slot, allowing parallel multicast transmissions. Slot allocation can be performed using heuristic methods such as greedy coloring or degree-based scheduling to minimize the total number of slots.
8. Multicast Transmission Planning:
   For each slot and spreading factor group, the network server defines the transmission parameters: frequency, data rate, payload size, and redundancy level (if used for FUOTA). It ensures that duty-cycle constraints are respected and that the transmission timing aligns with the periodic uplink schedule of end devices.
9. Time Synchronization and Schedule Distribution:
   The slot schedule is synchronized across gateways using a common time reference, such as *Class B* beacons or GPS-based timing. The network server distributes the schedule to all gateways, specifying the start times and parameters for each slot and spreading factor group.
10. Coordinated Multicast Execution:
    During operation, gateways transmit multicast packets according to the assigned slots. Gateways in the same slot send identical fragments simultaneously to their respective device groups. Since devices receive downlink transmissions only at predictable intervals (e.g., after each 15-minute uplink), the schedule ensures that the multicast packets coincide with these reception opportunities.
11. Monitoring and Feedback Collection:
    After each transmission round, the network server collects reception statistics, such as packet delivery ratios or missing fragment reports, from devices or gateways. This information is used to evaluate transmission success and detect potential interference.
12. Adaptive Rescheduling (Optional):
    If performance metrics fall below target thresholds, the server can dynamically adjust the schedule—reassigning slots, modifying spreading factors, or extending guard intervals. This adaptation helps maintain reliability under changing network conditions.
13. Session Completion:
    The process continues until all fragments have been successfully delivered to all devices in the multicast group. Once the session completes, the network server deactivates the multicast group and releases the allocated time slots.

5. CONCLUSIONS

In summary, the proposed algorithm introduces a gateway-aware, time-slot-based scheduling mechanism that adapts to different spreading factors and periodic device transmission intervals. By computing airtime-dependent slot durations and synchronizing multicast transmissions across gateways, it enables collision-free downlink communication and more efficient use of the limited LoRaWAN downlink capacity. The concept remains fully compliant with existing multicast and FUOTA specifications, since all coordination is managed at the network-server level without modifying end-device operation.

This work represents an initial stage and conceptual proposal, intended to outline the feasibility and potential of coordinated multicast scheduling in LoRaWAN. Further research is required to develop analytical models, perform large-scale simulations, and validate the proposed mechanism experimentally under realistic network conditions. Future work will focus on quantifying performance improvements, refining the scheduling heuristics, and exploring possible integration with intelligent network management frameworks such as SDN or adaptive FUOTA orchestration.

Bibliography

1. Adelantado F., Vilajosana X., Tuset-Pastor P., Martínez B., Melia-Segui J., and T. Watteyne,"Understanding the Limits of LoRaWAN,"IEEE Communications Magazine, vol. 55, no. 9, pp. 34–40, Sep. 2017.
2. Pop A.I., Raza U., Kulkarni P., Sooriyabandara M. "Does Bidirectional Traffic Do More Harm Than Good in LoRaWAN Based LPWA Networks?,"in Proc. IEEE Global Communications Conference (GLOBECOM), Singapore, Dec. 2017,
3. Zorbas D., Mantas K., Verstraeten V., Ostyn F. "Improving LoRaWAN Downlink Performance in the EU868 Sub-Band" Computer Communications, vol. 198, pp. 188–196, Oct. 2022.
4. Jouhari M., Khalil A.A., Al-Fuqaha A., Guizani M. "A Survey on Scalable LoRaWAN for Massive IoT,"IEEE Internet of Things Journal, vol. 9, no. 24, pp. 24354–24374, Dec. 2022.
5. Almuhaya M.A.M., Jubair M.F.A., Rahman F.A.A.M.A. "A Survey on LoRaWAN Technology: Recent Trends and Future Challenges,"Electronics, vol. 11, no. 1, p. 164, Jan. 2022.
6. LoRaWAN Application Layer Clock Synchronization Specification TS003-2.0.0. Available online.
7. LoRaWAN Application Layer Clock Synchronization Specification v1.0.0. Available online.
8. LoRaWAN Fragmented Data Block Transport Specification TS004-2.0.0. Available online.
9. LoRaWAN Fragmented Data Block Transport Specification v1.0.0. Available online.
10. LoRaWAN Remote Multicast Setup Specification TS005-2.0.0. Available online.
11. LoRaWAN Remote Multicast Setup Specification v1.0.0. Available online.
12. LoRaWAN Specification v1.0.3. Available online.
13. Semtech Corporation,LoRa Modulation Basics (Application Note AN1200.22),Camarillo, CA, USA, Rev. 2, Mar. 2015.Available online.
14. Centenaro M,, Vangelista L., Zanella A., Zorzi M., "Long-Range Communications in Unlicensed Bands: The Rising Stars in the IoT and Smart City Scenarios,"IEEE Wireless Communications, vol. 23, no. 5, pp. 60–67, Oct. 2016.
15. Mekki K., Bajic E., Chaxel F., Meyer F. "A Comparative Study of LPWAN Technologies for Large-Scale IoT Deployment,"ICT Express, vol. 5, no. 1, pp. 1–7, Mar. 2019.
16. LoRaWAN Firmware Management Protocol Specification v1.0.0, LoRa Alliance, Fremont, CA, USA, Oct. 2018.