POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

EXECUTIVE SUMMARY OF THE THESIS

Experimental Comparison of Matter over Thread and Zigbee

LAUREA MAGISTRALE IN
AUTOMATION AND CONTROL ENGINEERING - INGEGNERIA DELL'AUTOMAZIONE

Author: MASSIMO NOBILE
Advisor: PROF. ALESSANDRO ENRICO CESARE REDONDI
Co-advisor: ING. FABIO PALMESE
Academic year: 2024-2025

Abstract

This work presents an empirical performance evaluation comparing two dominant mesh networking technologies for smart home IoT systems: the established Zigbee protocol and the modern IP-based Matter over Thread stack.

This study utilizes a hardware testbed to quantitatively assess network overhead, latency, throughput, and mesh resilience under various topologies and load conditions. Furthermore, it investigates the feasibility of machine learning-based device classification for Matter over Thread traffic.

Zigbee's superior agility, as shown by lower single hop latency and near instantaneous route recovery, is offset by poor scalability due to its reactive routing protocol, which causes network instability in deep multi-hop topologies. Matter over Thread offers greater stability and scalability, with lower network overhead in larger configurations and robust multi-hop throughput, but slower network self-healing.

The results indicate that neither protocol is universally superior performance-wise, the choice represents a compromise between Zigbee's reactivity in smaller systems and Thread's stability for large-scale networks.

1. Introduction

The Internet of Things (IoT) has become a cornerstone of modern automation, with the smart home emerging as a primary application domain. These environments constitute complex distributed systems, often comprising dozens of heterogeneous devices that must communicate reliably and efficiently. From an automation engineering perspective, the performance of the underlying wireless communication protocol is paramount, directly impacting system responsiveness, fault tolerance, and scalability.

Two protocols stand at the forefront of this domain: Zigbee, a mature and widely deployed mesh networking standard, and Matter over Thread, a modern, IP-based stack designed to deliver universal interoperability. While technical specifications outline their architectural differences (primarily Zigbee's reactive, non-IP nature versus Thread's proactive, IP-based design) there is a lack of comprehensive, empirical data that directly compares their performance in realistic scenarios. This gap is particularly acute for the new Matter standard, for which public performance datasets are not yet available.

This thesis aims to fill that gap by presenting the results of a rigorous experimental analysis that addresses the following research questions:

1. How do the protocol overhead and traffic patterns of Zigbee and Matter over Thread scale with an increasing number of devices and network hops?
2. What are the comparative latencies and data throughput capacities of each protocol, particularly in multi-hop scenarios and with varying data payloads?
3. How do the differing routing architectures of Zigbee and Thread affect the network's self-healing time after a critical node failure?
4. Is it feasible to accurately classify IoT devices on a Matter over Thread network using machine learning techniques based on their traffic fingerprints?

To answer these questions, this work contributes a novel, labeled dataset of Matter over Thread traffic and presents a data-driven understanding of the fundamental trade-offs between these two competing ecosystems.

2. State of the art

The existing literature on Matter is predominantly composed of theoretical analyses and security assessments. Empirical studies in realistic deployment scenarios are scarce. The most relevant experimental work [3] evaluated Matter over Thread and WiFi, analyzing encapsulation overhead and application layer latency but lacking a comparative analysis with competing protocols such as Zigbee.

For the Thread protocol, numerous publications provide performance analysis, particularly concerning Round-Trip Time under various loads and network depths, such as in [5]. In [4] Thread network overhead and resilience were tested using the OpenThread Network Simulator, and packet count and reachability were measured.

Regarding protocol comparisons, several research studies exist; however, none have implemented Matter at the application layer. A notable white paper [7] compares Thread, Zigbee, and Bluetooth on latency and throughput, providing a crucial benchmark for the results obtained herein.

For network resilience, a documented value for Zigbee network self-healing time is available [1], which is critical for validating the results of this thesis' fault tolerance tests.

3. Experimental Methodology and Results

To ensure a fair and reproducible comparison, all experiments were conducted on a unified hardware testbed under controlled conditions. The testbed consisted of a central controller, network end-nodes, and a passive packet sniffer.

A Raspberry Pi 4 (4GB) running Home Assistant OS served as the smart home controller. For Matter over Thread, it was configured as a Thread Border Router using a Sonoff ZBDongle-E flashed with OpenThread RCP firmware. For Zigbee, it acted as the Zigbee Coordinator using a Texas Instruments CC2531 dongle.

Six ESP32-C6 development boards were used as the network nodes. The use of identical hardware for nodes in both Zigbee and Matter tests eliminated hardware-based performance variables. Nodes were flashed with custom firmware based on the Espressif ESP-IDF SDK examples, to emulate standard smart devices and to provide command-line interfaces for network diagnostics.

A dedicated laptop with a TI CC2531 dongle in promiscuous mode was used to passively capture all IEEE 802.15.4 frames. Wireshark, augmented with a Matter protocol dissector, was used for analysis.

3.1. Overhead and Scalability

Protocol overhead refers to all network traffic not part of the primary application payload, including beacons, route discovery messages, and keep-alives. This network verbosity was measured by systematically varying network topology, node count, and activity level.

Two distinct topologies were created: a fully connected mesh representing an ideal-case scenario (e.g., a single room), and a chain topology simulating a more challenging multi-hop layout (e.g., a long hallway). The chain was enforced using MAC address filtering at the firmware level to ensure a deterministic path. Tests were run under two conditions: an Idle State to measure baseline maintenance traffic, and a Controlled Traffic State, where an "On/Off Toggle" command was sent every five seconds.

For each unique scenario a 15-minute packet capture was recorded, with the total and application-layer packet counts serving as the primary metrics for overhead.

Results: The analysis revealed a clear divergence in scalability and stability between the two protocols.

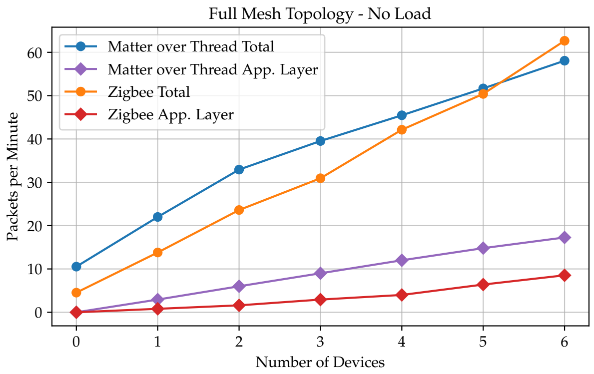

Figure 1: Overhead Test
Full Mesh Topology, Idle State

In a Fully Connected Mesh Topology, in idle state (Figure 1), Zigbee was initially more efficient, generating less baseline traffic for small networks (1-5 nodes). However, its overhead grew at a steeper rate, surpassing Matter over Thread at 6 nodes. This suggests Zigbee's underlying network maintenance mechanisms scale less efficiently in a dense mesh compared to Thread's more controlled MLE-based advertisements.

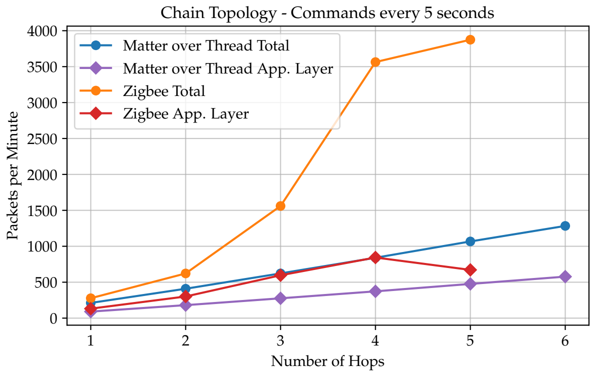

Figure 2: Overhead Test
Chain Topology, Controlled Load

A Chain Topology multi-hop scenario exposed the critical weakness of Zigbee's design. As shown in Figure 2, under a controlled traffic load, Matter over Thread's total traffic grows in a stable, linear fashion. In stark contrast, Zigbee's traffic escalates exponentially. Beyond three hops, its reactive AODV routing protocol, which initiates a network-wide broadcast for every new route, began to generate a cascade of Route Request (RREQ) packets that saturated the low-bandwidth network. This led to severe instability, and during the 6-hop test, the Zigbee coordinator repeatedly crashed, preventing the collection of a stable data point. Matter over Thread, with its proactive, quality-aware routing, maintained stability and predictability across all tested network depths, proving significantly more scalable.

3.2. Latency and Throughput

The raw network speed was assessed using standard benchmark tools, including `ping` and `iperf`. The tests were executed using the ESP32-C6 boards connected in a chain topology, and the firmwares (based on the ESP-IDF SDK) that were flashed did not include any application layer logic in the generated traffic, other than what is required to execute the benchmarking tool. The aforementioned firmwares incorporated a command-line interface, utilized for the configuration and operation of the tools from a PC connected to the boards via a serial port over a USB connection. Latency was gauged by calculating the mean Round-Trip Time (RTT) of 20 `ping` commands, with payload sizes ranging from 10 to 150 bytes and hop counts from 1 to 5. The maximum data throughput was measured using the `iperf` tool. The measurement was conducted by having a client on the first node send data to a server on the last node for 10 seconds. In the context of Thread, both UDP and TCP transport protocols were tested.

Results: Zigbee offered lower single-hop latency for small packets (10 bytes on 1 Hop in 24ms vs. 35ms for Thread), a benefit of its lightweight stack. However, as demonstrated in Figure 3, its performance exhibited a pronounced decline with an increase in hops or the presence of larger payloads necessitating fragmentation (>79 bytes for Zigbee). Its performance was found to be inferior to that of Thread at 3 hops, spiking dramatically at 5 hops, again due to network congestion from its routing protocol, resulting in a significant loss of packets. Thread demonstrated a marked improvement in the stability of the RTT, which increased predictably, even at a higher number of hops and with a larger payload. This enhancement was attributed to the utilization of the 6LoWPAN adaptation layer for fragmentation. It was ob-

served that fragmentation was required after 95 bytes at 1 hop and after 89 bytes for 2+ hops.
Throughput tests (Figure 4) further underscore this trend. Zigbee achieved the highest single-hop throughput (75 kbps), but its performance degraded very rapidly with each additional hop. Thread demonstrated significantly higher throughput across multiple hops, making it far better suited for bulk data transfers like Over-The-Air firmware updates, particularly when employing the TCP protocol. Despite exhibiting slightly suboptimal performance in comparison to UDP, TCP demonstrated consistent performance without necessitating tuning of the payload length for each hop.

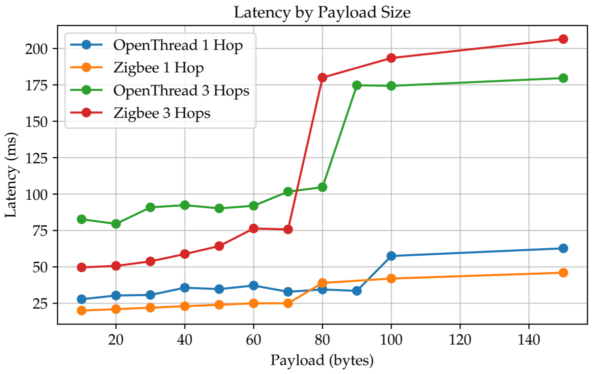

Figure 3: Latency (RTT) test results.

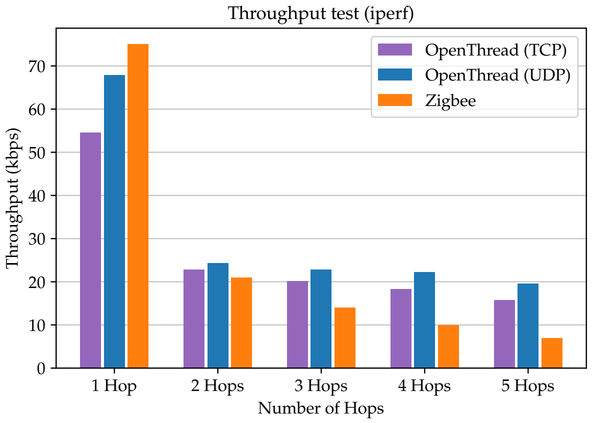

Figure 4: Throughput test results.

3.3. Mesh Route Recovery

This experiment was designed to empirically measure the network's self-healing capability by quantifying the route recovery time: the duration of the communication outage after a critical intermediate router fails. A four-node diamond topology was constructed, specifically designed to provide exactly two redundant 2-hop paths between the source and the destination node.
A continuous stream of `ping` traffic was generated, and the packet sniffer was used to observe the initial packets and identify which of the two available paths the protocol had chosen as the active route. The intermediate router on the active path was then abruptly powered off. The Route Recovery Time was calculated as the delta between the timestamp of the last successful packet on the failed route and the first successful packet on the new route. This experiment was repeated ten times for each protocol to ensure statistical validity.
Results: A stark contrast was observed. Zigbee demonstrated exceptional agility, recovering from a node failure in just 0.36 seconds. This aligns with existing literature on the rapid discovery mechanism of AODV routing. Thread was significantly slower, taking an average of 24 seconds to reconverge. This result, validated by repeating the test in a simulated environment with the OpenThread Network Simulator (OTNS), is a direct consequence of its proactive design. The network must wait for a node to miss several periodic MLE Advertisement broadcasts before declaring it "down" and reconverging, a process that prioritizes stability over immediate reactivity.

Protocol	Average	Std Dev
OpenThread	23.97	4.45
OpenThread (Simulator)	24.45	3.71
Zigbee	0.36	0.25
Zigbee (from [1])	0.4	n/a

Table 1: Route recovery time test results.

4. Traffic Classification

While existing research has explored device classification for established protocols such as Wi-Fi [6] and Zigbee [2], for Matter over Thread it still presents a new and unexplored domain. A key contribution of this work is the creation of a novel dataset for Matter over Thread traffic, addressing the absence of public datasets in this area. A heterogeneous network of 12 devices was deployed, including six off the shelf devices (smart bulbs, smart plugs and sensors) and six ESP32-C6 boards, flashed with three different firmwares that emulated different types of devices. To generate realistic traffic, a Home Assistant automation script sent commands to the devices, randomly adjusting the time between the commands and the destination node. The network traffic was captured for two 12-hour periods. It is crucial to note that the physical layout of the network underwent modifications between the two captures, resulting in the creation of two distinct network topologies (Topology A and B). This strategic alteration ensured that the dataset was not overfitted to a single layout. The raw packet captures were processed using a 5-second non-overlapping sliding window. For each window, a feature vector was computed for each source device, containing statistical features of packet lengths and inter-arrival times (e.g., mean, standard deviation, kurtosis, skew). A Random Forest classifier was chosen for its robustness and trained for two distinct tasks: identifying the broad device category (e.g., Light, Sensor) and identifying the individual device instance. To rigorously test generalization, the model was evaluated under three scenarios: a baseline single topology test, a test on the combined dataset, and a challenging cross-topology test (training on topology A, testing on topology B, or vice versa).

Results: The primary evaluation metric was the weighted F1-score, the results are summarized in Table 2. Classifying devices by their functional category is highly effective, the F1-score was exceptionally high in static scenarios, and critically, remained robust even in the cross-topology tests. This indicates that the fundamental traffic patterns defining a "light bulb" or "sensor" create a distinct statistical fingerprint that is largely independent of the device's position in the network mesh. The conclusions drawn from the F1-scores are visually confirmed by the analysis of confusion matrices (Figure 5 and 6). In these scenarios, the heatmaps show a strong diagonal, with most predictions correctly aligning with the true labels.

Experimental scenario	Category ID.	Device ID.
Combined dataset $(A + B)$	0.9802	0.8552
Cross-topology $(A \to B)$	0.8174	0.3643
Cross-topology $(B \to A)$	0.8313	0.3862
Single topology (A)	0.9798	0.8669
Single topology (B)	0.9887	0.9172

Table 2: Classification Weighted F1-scores.

In stark contrast, identifying individual device instances proved to be highly sensitive to network topology. While the model performed well in static scenarios, its performance plummeted in the cross-topology test, indicating a complete failure to generalize. The subtle features the model learned to distinguish two identical devices in one layout (e.g., timing variations from specific routing paths) were not transferable once the network structure changed.

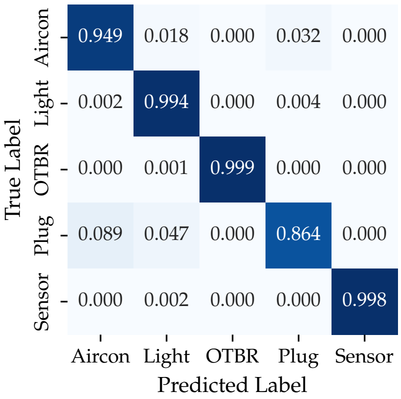

Figure 5: Confusion matrix heatmap Category Classification, Merged Topology

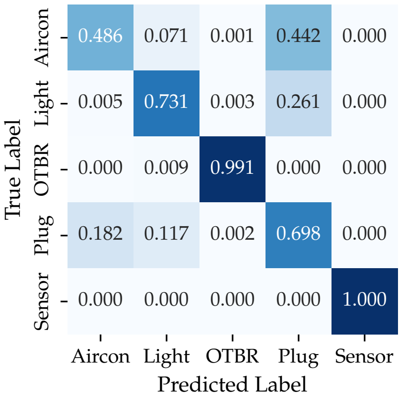

Figure 6: Confusion matrix heatmap Category Classification, Cross Topology

5. Conclusion

This thesis has provided a detailed, empirical analysis comparing the performance of Zigbee and Matter over Thread. The findings demonstrate that there is no single "best" protocol, but rather a clear engineering trade-off between agility and stability.

Zigbee is the more agile protocol, offering lower initial latency and near-instantaneous recovery from network failures. This makes it well-suited for smaller, time-critical applications where immediate response to network changes is valued. However, its scalability is poor, and its reactive routing mechanism generates overwhelming control traffic that leads to network instability in larger or deeper mesh topologies.

Matter over Thread is the more stable and scalable protocol. Its IP-native architecture, robust multi-hop throughput, and efficient overhead scaling make it the superior foundation for large, heterogeneous, and future-proof smart homes. This stability, however, comes at the cost of significantly slower network self-healing, a critical consideration for applications requiring high availability.

This data-driven understanding of the trade-off between reactivity and stability is critical for engineers and developers. The choice of protocol should be a deliberate decision based on the specific requirements of the target application, balancing the need for speed and agility against the demand for scalability and robustness in the ever-expanding landscape of the Internet of Things.

Future work should extend this analysis in several key directions. First, experiments with the Matter protocol should be scaled to much larger networks to further investigate the scalability for potential implementations in environments more complex than smart homes, such as smart buildings or smart cities.

Second, the tests should be repeated in environments with controlled and varied levels of 2.4 GHz RF interference to assess real-world robustness.

Third, a long-term comparative analysis of the energy consumption of battery-powered end devices in both ecosystems would provide a critical metric for a complete evaluation.

Finally, the development of more sophisticated machine learning models for Matter over Thread traffic, trained with data from numerous network topologies, to determine if an accurate and generalizing model for device identification can be obtained, even across changing network conditions.

References

[1] Odgerel Ayurzana. Building ZigBee Mesh Network using AODV Routing Protocol. *International Journal of Advanced Smart Convergence*, 13:249–259, 2024.

[2] Antonio Boiano, Alessandro Enrico Cesare Redondi, and Matteo Cesana. IoTScent: Enhancing Forensic Capabilities in Internet of Things Gateways. In *2023 IEEE 9th World Forum on Internet of Things (WF-IoT)*, pages 1–6. IEEE, 10 2023.

[3] Saeid Madadi-Barough, Pau Ruiz-Blanco, Jiadeng Lin, Rafael Vidal, and Carles Gomez. Matter: IoT Interoperability for Smart Homes. *IEEE Communications Magazine*, 63(4):106–112, 4 2025.

[4] Martins Mihaeljans and Andris Skrastins. Openthread Network Density Evaluation: Quantitative Analysis. In *2023 Symposium on Internet of Things, SIoT 2023*. Institute of Electrical and Electronics Engineers Inc., 2023.

[5] NXP Semiconductors. Thread Large Network. Technical report, 2017.

[6] Fabio Palmese, Alessandro E. C. Redondi, and Matteo Cesana. Feature-Sniffer: Enabling IoT Forensics in OpenWrt based Wi-Fi Access Points. In *2022 IEEE 8th World Forum on Internet of Things (WF-IoT)*, pages 1–6. IEEE, 10 2022.

[7] Silicon Labs. AN1142: Mesh Network Performance Comparison. Technical report, 11 2019.