

Thread Protocol

Subjects: Telecommunications

Contributor: Sohaib Bin Altaf Khattak, Moustafa M. Nasralla, Haleem Farman, Nikumani Choudhury

The increasing demand for Internet of Things (IoT) applications has resulted in vast amounts of data, requiring the utilization of big data analytics. The integration of big data analytics in IoT-based smart cities can greatly benefit from the development of wireless communication protocols, among which the Thread protocol has emerged as a promising option. Thread is IEEE 802.15.4 based and has advanced capabilities like mesh networking, IPv6 support, and multiple gateways providing no single point of failure.

Keywords: ad hoc networks ; IEEE 802.15.4 ; IPv6 ; internet of things ; mesh network ; smart cities ; thread network ; wireless sensor networks

1. Introduction

Advancements in computing technology and networking have given rise to the Internet of Things (IoT), which aims to enable objects to gather and share data over the Internet using smart devices ^{[1][2]}. The IoT was envisioned as a large-scale network of Internet-connected objects that are uniquely addressable and reachable using standard networking protocols ^[3]. It has played a pivotal role in the development of smart cities, where interconnected devices and systems work together to enhance the efficiency, sustainability, and quality of urban life ^[4]. Some prominent applications include smart energy management systems ^[5], building controls ^[6], and smart transportation ^{[7][8]}, which have gained significant attention from both academia and industry. Many networking protocol stacks have been designed to meet the specific needs of diverse smart city applications and IoT-based systems ^[9]. Zigbee and Bluetooth are widely adopted protocols in smart city applications. Zigbee is used in various monitoring systems and smart infrastructure ^[10], while Bluetooth is employed for occupancy detection, asset tracking, and seamless connectivity in smart buildings and transportation systems ^[11].

The majority of IoT applications in the real world make use of single-hop wireless connectivity to the gateway based on a star topology. Alternatively, the mesh topology has also been considered that enables each node to communicate with every other node and gateway, enhancing reliability ^[12]. While wireless standards such as Bluetooth and ZigBee are continuously evolving to support mesh networks, to extend the support towards mesh networks, a new protocol named Thread has been developed using open and proven standards, such as IEEE 802.15.4 and 6LoWPAN ^[13].

The Thread is a communication protocol that is maintained by Thread Group ^[14]. It has been designed to meet the requirements of applications that demand low power consumption, low data throughput, and a short communication range. Thread protocol is designed to meet the specific needs of diverse smart city applications, providing secure and scalable connectivity to a large number of devices in an energy-efficient and reliable manner. The Thread-based IoT mesh network can be connected to the global Internet using a gateway called a border router that provides connectivity through Ethernet or Wireless Local Area Network (WLAN). Thread provides end-to-end encryption, no single-point failure, and consumes low power. Thread is a relatively new protocol compared to ZigBee and Bluetooth, so its performance has not been extensively studied, especially in real-world applications with varying scenarios.

2. Protocol Stack

The Thread protocol stack is a combination of various standards like IEEE 802.15.4 and 6LoWPAN, etc. ^{[15][16]}. It has six different layers, as can be seen in **Figure 1**. Each of the layers is described below:

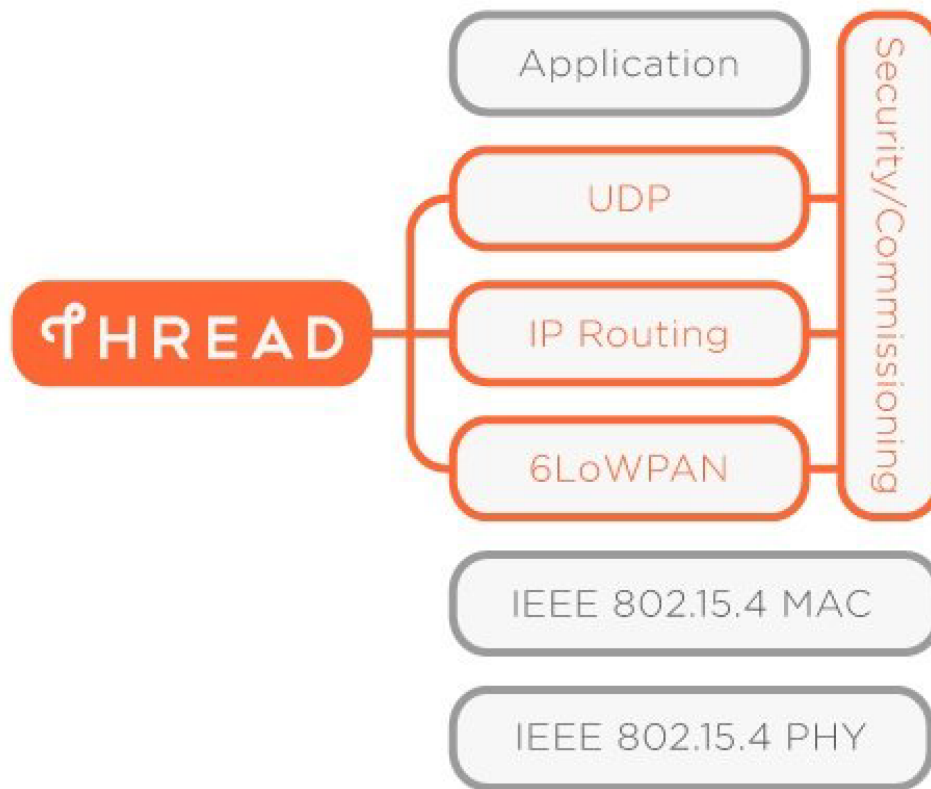


Figure 1. Thread protocol stack ^[14].

- **IEEE 802.15.4 Physical Layer**

The physical layer is used for the transmission and reception of packets over a physical medium. The physical and MAC layers of the Thread are defined according to the IEEE 802.15.4 standard. In the physical layer, it operates in the 2.4 GHz free ISM band and is divided into 16 channels numbered from 11 to 26. It offers a data rate of 250 kbps ^[12].

- **IEEE 802.15.4 MAC Layer**

This layer is responsible for message handling and management, congestion control, and error correction. The MAC layer includes CSMA-CA (Carrier Sense Multiple Access-Collision Avoidance), allowing Thread devices to utilize the bandwidth efficiently. It has features like beacon management, channel access, Guaranteed Time Slot (GTS) management, etc. ^[17].

- **6LoWPAN**

It stands for IPv6 for low-power wireless personal area network (6LoWPAN). It facilitates the Thread network by providing adaptation between the IP layer and the 802.15.4 MAC layer. Packet fragmentation happens here as the packets are assembled into IPv6 format while passing from the MAC layer to the IP layer and vice versa. The layer adapts the bigger frames into smaller ones, making it more suitable for energy-constrained and low-bandwidth devices. 6LoWPAN supports both mesh and star topologies ^[9].

- **IP Routing**

As Thread supports the IPv6 architecture, DHCPv6 is used for router address assignment. The IPv6 frames are 40 bytes long and contain an address field of 128 bits, allowing more addresses than the IPv4. In IP routing, the mesh local addresses of routers are maintained in a routing table, ensuring constant connectivity and the availability of paths. Thread uses the distance vector routing protocol, which aims to maximize the amount of routing information in a single message. The routing cost is estimated using the Mesh Link Establishment (MLE) mechanism ^{[14][16]}.

- **User Datagram Protocol**

Thread uses User Datagram Protocol (UDP) for messaging between devices. It also supports TCP-based services for application-layer communication. UDP enables messaging with minimum connection overhead, providing a faster and higher throughput of messages. Applications requiring a guarantee may use the TCP protocol, assuring reliability by detecting network congestion and packet acknowledgement and re-transmission ^{[13][14]}.

- **Application Layer**

The application layer provides an interface to the network. Thread is application-layer agnostic and provides the flexibility to choose from a variety of application layers to enable device connectivity like Message Queuing Telemetry Transport (MQTT), Constrained Application Protocol (CoAP), etc. [18][19]. Thread, however, provides multicast and UDP messaging as essential application layer services to allow communication over IP.

3. Device Types and Topology

Thread network supports a connectivity model where the end devices are only connected to their parent nodes while all the routers are connected together, forming a mesh network. The devices in Thread play different roles based on their functionalities [14][16][19], as shown in **Figure 2**. In IEEE 802.15.4 networks, the commonly used terminologies of coordinator, router, and end device are prevalent. However, the Thread network architecture introduces specific roles and functionalities that bring added clarity and facilitate customized network design for IoT applications.

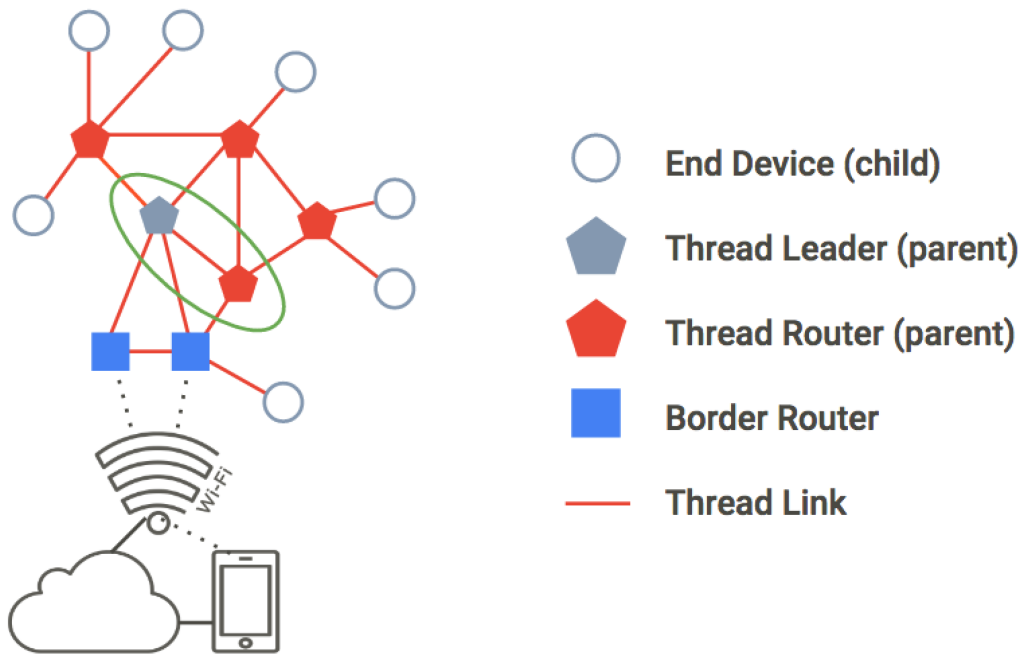


Figure 2. Basic topology and device types of Thread network [14].

- **End Device**

These devices have reduced functionalities and are energy-constrained, similar to their counterparts in other wireless protocols like ZigBee and Bluetooth. However, Thread provides specific features to optimize energy consumption, such as allowing end devices to enter sleep mode to conserve energy. Unlike some other protocols, Thread end devices are always connected to their parent node, which has routing capabilities and remains continuously active. Thread also introduces the concept of Router Eligible End Devices (REEDs) that can be promoted to routers for improved performance. The dynamic up-gradation and down-gradation of devices is managed both explicitly by the network manager and automatically by the network based on prevailing conditions.

- **Router**

Thread routers share similarities with routers in other protocols, such as ZigBee and Bluetooth, in terms of their routing capabilities. These devices require power throughout and are also called Full Thread Devices (FTD). All routers in the network are connected together, forming a mesh, and route the data packets from source to destination nodes. Additionally, Thread routers also provide joining and commissioning services to the end nodes. These nodes are also capable of becoming border routers or leader routers, and if a leader goes down due to some reason, these nodes can be automatically elected as leaders by the network.

- **Border Router**

This router connects the Thread network with the outside world and enables data flow in and out of the network. It can also be termed the gateway of the Thread network to the global internet or cloud. It can support both WiFi and Ethernet

for connection to an external network. Unlike other networks having a single gateway, in Thread multiple border routers can also exist in a network, thus avoiding a single point of failure.

- **Leader**

Any router or border router that manages the Thread network is called a leader. It has all the functionalities of the router but also has additional responsibilities like assigning IP, upgrading or downgrading the router status of REEDs, and distributing the network configuration information to the routers in the network. Unlike other wireless protocols, such as Bluetooth and ZigBee, which may have central devices or coordinators, if the Thread leader goes down, a router is elected as a leader by other thread routers to perform network management tasks.

References

1. Khattak, S.B.A.; Jia, M.; Marey, M.; Nasralla, M.M.; Guo, Q.; Gu, X. A Novel Single Anchor Localization Method for Wireless Sensors in 5G Satellite-Terrestrial Network. *Alex. Eng. J.* 2022, 61, 5595–5606.
2. Ilyas, A.; Mahfooz, S.; Mehmood, Z.; Ali, G.; ElAffendi, M. Two-way approach for improved real-time transmission in fog-iot-based health monitoring system for critical patients. *Comput. Syst. Sci. Eng.* 2023, 46, 3815–3829.
3. Elfouly, F.H.; Ramadan, R.A.; Khedr, A.Y.; Yadav, K.; Azar, A.T.; Abdelhamed, M.A. Efficient Node Deployment of Large-Scale Heterogeneous Wireless Sensor Networks. *Appl. Sci.* 2021, 11, 10924.
4. Jan, H.; Yar, H.; Iqbal, J.; Farman, H.; Khan, Z.; Koubaa, A. Raspberry Pi Assisted Safety System for Elderly People: An Application of Smart Home. In *Proceedings of the 2020 First International Conference of Smart Systems and Emerging Technologies (SMARTTECH)*, Riyadh, Saudi Arabia, 3–5 November 2020; pp. 155–160.
5. Tekler, Z.D.; Low, R.; Yuen, C.; Blessing, L. Plug-Mate: An IoT-based occupancy-driven plug load management system in smart buildings. *Build. Environ.* 2022, 223, 109472.
6. Zhuang, D.; Gan, V.J.; Tekler, Z.D.; Chong, A.; Tian, S.; Shi, X. Data-driven predictive control for smart HVAC system in IoT-integrated buildings with time-series forecasting and reinforcement learning. *Appl. Energy* 2023, 338, 120936.
7. Low, R.; Tekler, Z.D.; Cheah, L. Predicting commercial vehicle parking duration using generative adversarial multiple imputation networks. *Transp. Res. Rec.* 2020, 2674, 820–831.
8. Jan, B.; Farman, H.; Khan, M.; Talha, M.; Din, I.U. Designing a Smart Transportation System: An Internet of Things and Big Data Approach. *IEEE Wirel. Commun.* 2019, 26, 73–79.
9. Al-Fuqaha, A.; Guizani, M.; Mohammadi, M.; Aledhari, M.; Ayyash, M. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Commun. Surv. Tutorials* 2015, 17, 2347–2376.
10. Allahham, A.A.; Rahman, M.A. A smart monitoring system for campus using Zigbee wireless sensor networks. *Int. J. Softw. Eng. Comput. Syst.* 2018, 4, 1–4.
11. Tekler, Z.D.; Low, R.; Gunay, B.; Andersen, R.K.; Blessing, L. A scalable Bluetooth Low Energy approach to identify occupancy patterns and profiles in office spaces. *Build. Environ.* 2020, 171, 106681.
12. Orfanos, V.A.; Kaminaris, S.D.; Papageorgas, P.; Piromalis, D.; Kandris, D. A Comprehensive Review of IoT Networking Technologies for Smart Home Automation Applications. *J. Sens. Actuator Netw.* 2023, 12, 30.
13. Unwala, I.; Taqvi, Z.; Lu, J. Thread: An IoT Protocol. In *Proceedings of the 2018 IEEE Green Technologies Conference (GreenTech)*, Austin, TX, USA, 4–6 April 2018; pp. 161–167.
14. Thread Network Fundamentals, White Paper. September 2022. Available online: <https://www.threadgroup.org/support> (accessed on 27 April 2023).
15. Rzepecki, W.; Ryba, P. IoTSP: Thread Mesh vs. Other Widely used Wireless Protocols—Comparison and use Cases Study. In *Proceedings of the 2019 7th International Conference on Future Internet of Things and Cloud (FiCloud)*, Istanbul, Turkey, 26–28 August 2019; pp. 291–295.
16. Herrera, T.; Núñez, F. Design and Prototyping of a Thread Border Router Based on a Non Network-Co-Processor Architecture. *IEEE Access* 2020, 8, 60613–60625.
17. Kurunathan, H.; Severino, R.; Koubaa, A.; Tovar, E. IEEE 802.15.4e in a Nutshell: Survey and Performance Evaluation. *IEEE Commun. Surv. Tutorials* 2018, 20, 1989–2010.
18. Tupas Castro, C.M.; Sharma, A.; Kumar, D.S.; Abidi, K.; Kim, N. The implementation of Thread Network for a Smart Factory. In *Proceedings of the 2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, Singapore, 4–7 December 2022; pp. 253–260.

19. Chitanvis, R.; Ravi, N.; Zantye, T.; El-Sharkawy, M. Collision avoidance and Drone surveillance using Thread protocol in V2V and V2I communications. In Proceedings of the 2019 IEEE National Aerospace and Electronics Conference (NAECON), Dayton, OH, USA, 15–19 July 2019; pp. 406–411.
-

Retrieved from <https://encyclopedia.pub/entry/history/show/105792>