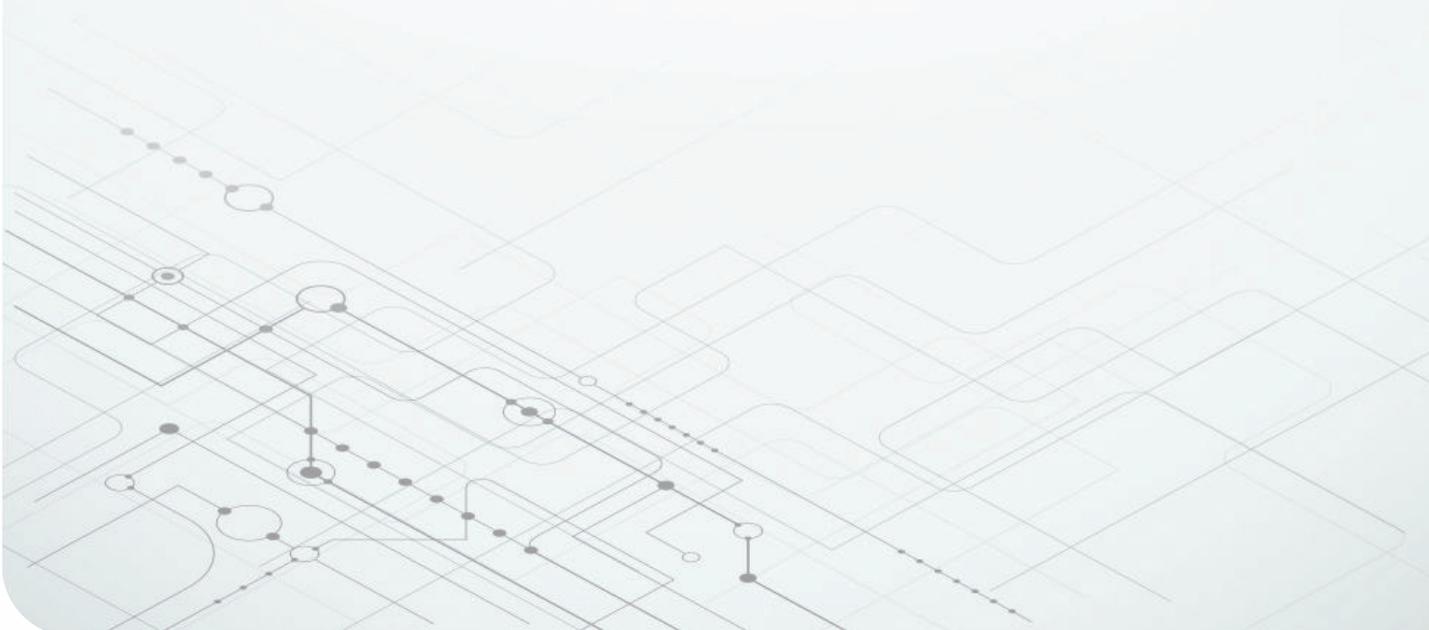


Exploring Thread and Zigbee for home and building automation



In our connected world, smart homes and buildings are an important part of our daily lives. Wireless connectivity technologies, like Thread and Zigbee, make home and building automation possible.

At a glance

Advances in wireless communication protocols and automation frameworks have made connected, integrated homes and commercial buildings a reality. This paper will highlight the advantages of wireless home and commercial building automation mesh technologies Zigbee and Thread.

1 **Design considerations and wireless protocols for connected homes and buildings**

Home and commercial building automation products and ecosystems have very different requirements to meet customer needs.

2 **Exploring the differences and similarities of Zigbee® and Thread**

Zigbee and Thread technologies provide a built-in mesh networking security and application infrastructure for embedded, low-power and low-cost devices.

3 **Project CHIP**

The Project Connected Home over IP (Project CHIP) is a working group within the Zigbee Alliance tasked to develop a royalty-free connectivity standard for home and building automation.

Introduction

In today's connected world, wireless communication protocols and automation frameworks have made smart homes and commercial buildings commonplace. This paper will go over the relative advantages of the available wireless home and building automation mesh technologies.

Design considerations

Home and commercial building automation products and ecosystems have very different requirements to meet customer needs. For example, a homeowner may accept occasional instability of a product, a commercial building operator could require a support contract with on-site support. Where a building operator may be willing to build a custom aggregator service, this is out of reach for even a dedicated hobbyist homeowner.

The similarities between home and commercial building automation customers should not be overlooked. Both need to operate within the regulatory restrictions of their geographic regions. Traffic profiles and network topologies will be of concern to enable all nodes are reachable. And protocol-level interoperability is essential. These customers will be interested in existing technologies such as Zigbee®, Thread®, Wi-Fi® and *Bluetooth*® Low Energy.

Wireless protocols

Wireless networking technologies offer a number of advantages over wired networking technologies when an existing infrastructure does not exist.

Table 1 compares the different wireless technologies for home and building automation.

	Zigbee	Thread	Bluetooth Low Energy	Wi-Fi
Band	2.4-GHz industrial-scientific-medical (ISM)	2.4-GHz ISM	2.4-GHz ISM	2.4-GHz/5-GHz ISM
Throughput	250 kbps	250 kbps	As high as 2 Mbps	As high as hundreds of Mbps
One-hop range	As high as a few hundreds of meters; routers can extend range through multihop	As high as a few hundreds of meters; routers can extend range through multihop	Can go as high as a few hundreds of meters, with long-range mode in Bluetooth 5	Tens of meters; extendable using multiple access points
Battery type and life	Years on a coin cell	Years on a coin cell	Years on a coin cell	AAA/AA for years
Topology	Mesh	Mesh	Point-to-point, mesh	Star; some mesh being standardized
Traffic profile	Local device-to-device, many to one and one to many	Device-to-device and device-to-cloud	Best for device-to-smartphones	Device-to-cloud
Protocol layering	Network and application	Network with Internet Protocol (IP) standards on top	Network and application	Network with IP standards on top
Certification program and interoperability	End product certification	Stack certification	Stack certification	Data-link layer and some upper-layer stack certification
Security	Networkwide encryption and authentication through installation codes	Password-based authentication with Datagram Transport Layer Security (DTLS)	Asymmetric encryption for key generation and exchange, connection pairwise keys	Password and certificate-based authentication, supports all IP-based security standards
IP connectivity and support	Requires gateway to perform IP address translation	Native IPv6 networking	Requires gateway to perform IP address translation	Native

Table 1. Comparison of Wireless Technologies for Home and Building Automation

Zigbee

The Zigbee protocol is a wireless personal area mesh networking protocol based on Institute for Electrical and Electronics Engineers (IEEE) 802.15.4 and maintained by the Zigbee Alliance, which ensures continuity between versions of the Zigbee Pro specification and certifies devices.

A Zigbee network begins with an initial coordinator device that acts as the arbiter for decisions of the network. Routers or end devices can be brought in after the formation of the network to handle forwarding and routing of packets within the mesh. End devices attach to the initial coordinator or to routers, and do not route for the Zigbee network. It is possible to configure the

end devices to sleep for long periods of time to preserve battery power.

The Texas Instruments (TI) **SimpleLink™ CC13x2 and CC26x2 software development kit** (SDK) has a certified implementation of the Zigbee Protocol called Z-Stack™ software.

Thread

The Thread protocol is a wireless personal area mesh networking protocol based on IEEE 802.15.4, IPv6 over Low-Power Wireless Personal Area Networks (6LoWPAN) and several Internet Engineering Task Force standards. Thread is maintained by the Thread Group, which ensures continuity between versions of the Thread protocol specification and certifies device interoperability.

A Thread network begins when a router-eligible device turns on but does not find an existing network. This device may then become the leader of a new Thread network, to which other devices may connect through a standard commissioning process as end devices. If an end device is router-eligible, it may be promoted to router within the Thread network as topology changes require.

Any router within the Thread network must be capable of being the leader. Routers within the Thread network participate in the forwarding and routing of network messages and are elected from the pool of router eligible end devices already in the network. End devices connect to routers and do not route for the Thread network. These end devices may operate as sleepy devices to save battery power.

The TI SimpleLink CC13x2 and CC26x2 SDK has a certified implementation of the Thread Protocol based on OpenThread. OpenThread is an open source implementation of the Thread protocol created and maintained by Google.

Similarities Between Zigbee and Thread

Zigbee and Thread technologies are standard-based protocols that primarily operate in the worldwide 2.4-GHz ISM band. These technologies provide a built-in mesh networking security and application infrastructure for embedded, low-power and low-cost devices.

The Zigbee Alliance and Thread Group both have a process for member companies to enact changes to the specification.

Both protocols leverage a common underlying data-link communication layer designed and maintained by the IEEE.

Figure 1 shows Zigbee and Thread protocol layering.

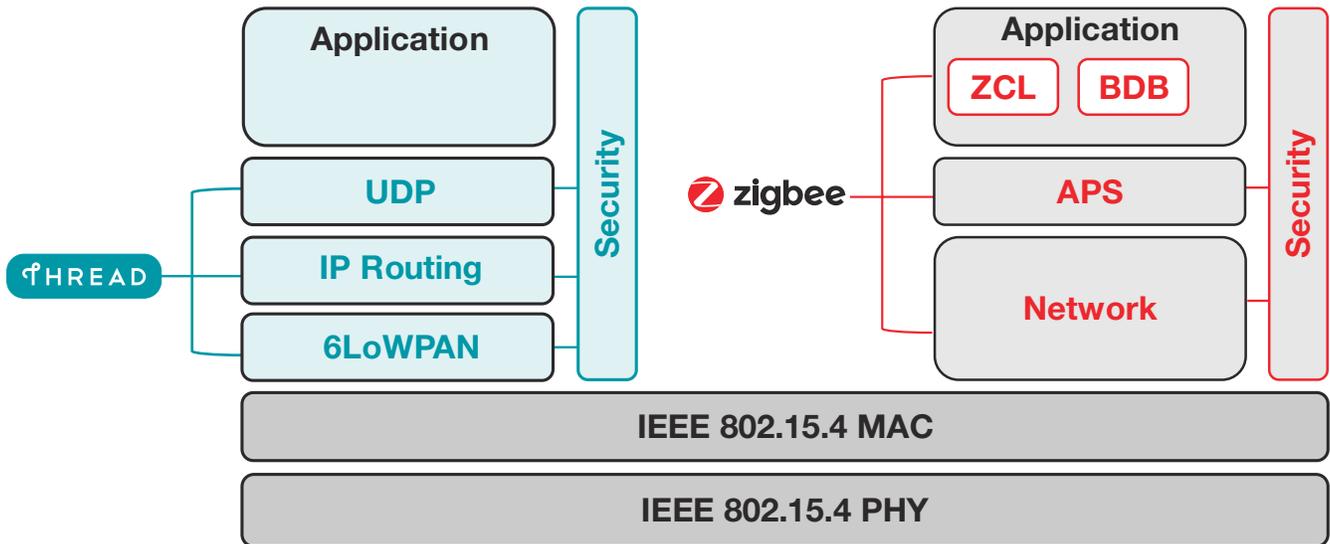


Figure 1. Zigbee and Thread Protocol Layering

The IEEE 802.15.4 standard specifies the Media Access Control (MAC) and physical (PHY) layers of the Open Systems Interconnection communication model. Both

Zigbee and Thread implement a personal area network that guarantees a reliable hop-to-hop link for the transfer of upper-layer data frames at very low-power operation.

Since higher, less timing-sensitive protocol layers are implemented in software, it's possible to implement 802.15.4-based standards like Zigbee and Thread as different software variants that run on the same silicon (as is the case for the TI SimpleLink multistandard CC2652R wireless microcontroller [MCU]). With a single unique hardware design, the corresponding firmware can be loaded at the factory or upgraded in the field, providing a simplified and futureproof solution.

Both Zigbee and Thread implement an asynchronous mode of operation within the IEEE 802.15.4 standard. This transmitter-initiated profile enables the efficient exchange of small packets in a low-power wireless network. Devices that do not generate data often can wake up and reliably send packets with extremely short latency.

Regardless of the data's destination in the network (one or multiple hops away), the battery-powered devices wake up from sleep, send the data to their one-hop relay node, and then quickly go back into a standby state. Between instances when the device is active and sending or receiving data, the radio can be off and operating in the realm of microamperes. For instance, the CC2652R device can sleep while retaining full random-access memory contents and consume only 0.9 μ A.

Both Zigbee and Thread use a distance vector algorithm to build routing tables between routers. Zigbee uses ad-hoc on-demand distance vector routing, and Thread uses a modified Routing Information Protocol. Having the routers of each network generate and store the routing information rather than the end devices minimizes network maintenance traffic to the end nodes, conserving radio time.

This efficiency is significantly advantageous for devices that typically generate data triggered by sporadic alarm events (such as door and window sensors) or user actions (such as switches/key fobs, alarm panels or shade systems). The battery-powered devices can sleep most of the time, only waking up for the occasional application-initiated data or periodic data poll messages.

The periodic data polls are needed for unsolicited down-link messages and to maintain connection with the end device's parent router.

With peak current levels around the single-digit microamperes, Zigbee and Thread enable devices in the home and building automation space to operate for years off of coin-cell batteries. Header compression and reuse make communication in both Zigbee and Thread efficient by creating smaller over-the-air packets. Thread leverages 6LoWPAN compression, fragmentation and link-layer forwarding. Zigbee was designed from the ground up, with binary-data optimization in the networking protocol for the underlying 802.15.4 frames.

The headers and networking management operations necessary to maintain and establish routes are short and reliably enable a 20-byte application frame (for a lighting control command or an alarm event) in a single 802.15.4 packet instance of 50 to 80 bytes, with a turnaround time of a few tens of milliseconds per hop. In most systems, with four to five hops as the mesh branch's biggest length, this speed still provides less than 100 ms of latency for actuating device-to-device communication.

Low-power operation and network scalability are both important requirements in residential systems with tens of nodes interoperating, such as lights, environmental sensors and thermostats. But these factors are an even bigger priority in commercial and industrial building automation systems, where the number of devices may reach hundreds or even thousands of nodes.

Both the Zigbee and Thread protocols implement an efficient routing algorithm to minimize over-the-air traffic and broadcasts. The receiver in these nodes is always on (they are usually mains-powered, like a light bulb/ fixture or a thermostat) and store next hops to the final destination by building a small and lean routing table. The networks don't relay packets by flooding the network through broadcasts, which ultimately can hinder scalability.

The routing nodes exchange only small intermittent broadcast messages, minimizing overall housekeeping traffic to maintain the mesh. Routing nodes in the network also have the important role of buffering the data for the downlink communication of their sleeping “children,” which can be configured to extract packets efficiently depending on the downlink requirements (which in many cases are latency-insensitive).

Both Zigbee and Thread technologies have been demonstrated successfully in large commercial deployments that reach hundreds of nodes within the same network. TI has deployed *Breaking the 400-Node ZigBee Network Barrier TI's ZigBee SoC & Z-Stack Software*, and the technology can enable even larger networks depending on node density, amount of traffic generated and application profile.

Differences Between Zigbee and Thread

While Zigbee and Thread both look similar, there are slight differences in how each protocol establishes and maintains the network.

Zigbee supports a centralized and distributed (touchlink) coordination scheme. In the centralized approach, Zigbee uses a statically allocated coordinator within the network to manage operations. In contrast, a leader device, elected from one of the network’s routers, handles networkwide decisions in the Thread network.

Devices statically configured as routers handle the forwarding and routing of messages within a Zigbee network. Thread elects devices that route within its network from an existing pool of router-eligible devices.

Zigbee offers many ways to add new devices onto the network. Thread adds new devices with a standardized commissioning protocol, which requires human intervention to complete.

Zigbee enables defined and administrated networks, which tends to entice corporate entities and homeowner hobbyists. Thread enables a self-forming and self-

healing mesh network geared toward an autonomously administered ecosystem of devices.

The differences become more pronounced as you move up the networking stack. Zigbee defines an application framework for device-to-device communication. Application interactions between devices are defined and certified with the Zigbee cluster library. Thread offers its application layer protocols as an option to be reused by the business logic of the end product, but does not mandate their usage. This could be the User Datagram Protocol for efficient transmission of messages and the Constrained Application Protocol for reliable interactions. However, a Thread application may use Transmission Control Protocol, Hypertext Transfer Protocol, Message Queuing Telemetry Transport or any other protocol to transport messages.

Zigbee offers both reliable network operation and application interactions. Thread only ensures reliable network operation, but offers the ability to define the application protocol best suited for the device requirements.

Interoperability does not end at the boundaries of the mesh. Thread offers native IP routing and a natural device-to-cloud connection. Zigbee requires a specialized hub and translation to interoperate with cloud services.

More than 100 million products have included the Zigbee Pro networking layer, alongside the Zigbee cluster library, with several revisions of the standards for both the core mesh networking functionality and application layers. The Thread protocol is a relatively newer offering and does not have the same market penetration. Since both technologies use the same radio MAC/PHY protocol, it is easy to pivot between both with the SimpleLink wireless MCU.

Table 2 lists the differences between the two core mesh networking standards and the implications for technology adopters.

Functionality	Zigbee	Thread
Authentication at joining	Centralized through the trust center with optional out-of-band device-based installation code, or distributed with proximity pairing	Smartphone-based, with device-specific quick response (QR) code scanning
Security	Advanced Encryption Standard (AES)-128 network-level, with a key transported from joiner to joining device Optional application-level key	AES-128 MAC level derived from an elliptic curve cryptography-based password juggling scheme and DTLS session establishment
Device bootstrapping and commissioning	Button-press easy mode or proximity-based (touchlink)	Smartphone-based, with device-specific QR code scanning
Network and mesh management	Centralized coordinator; may be distributed in touchlink network	Dynamic leadership
Self-healing	Native router and mesh self-healing	Routers and leader self-election and self-healing
Cloud integration	Zigbee gateway with a purpose built translation	Thread border router with native IPv6
Power performance for application packets	Optimum	Very good
Latency performance for application packets	Best	Very good
IP native integration	No	Yes
Standard longevity	First revision in 2005	First revision in 2015
Industry participation	Approximately 400 member companies	Approximately 270 member companies

Table 2. Zigbee and Thread Comparison

Project CHIP

A number of home and building automation ecosystem vendors are coming together within the Zigbee Alliance to produce a new application framework. The Project Connected Home over IP (Project CHIP) is a working group within the Zigbee Alliance tasked to develop a royalty-free connectivity standard for home and building automation. This standard will have an emphasis on security and be based on market-leading IP standards-based automation technologies. This working group plans to deliver an open-source reference connectivity framework across Wi-Fi and Thread.

Project CHIP will standardize device rendezvous and provisioning through a Bluetooth Low Energy and Wi-Fi based protocol. The standard Zigbee Alliance CHIP phone application will bring devices onto the user's Wi-Fi or Thread network. Once connected to the local IP network, the device will use standard service discovery to find other CHIP nodes locally. Application level interactions are standardized and certified to ensure

interoperability between product vendors. And, if full internet connectivity is available, the device will be able to interact with the CHIP certified cloud aggregator services.

A CHIP certified product can rely on the existence of all these features within the ecosystem. This has the benefit of shortening development time for end products. While also leaving open the possibility of extending functionality with manufacturer specific features.

Conclusion

Both the Zigbee and Thread protocols take advantage of asynchronous non-beacon mode IEEE 802.15.4. Both protocols have roughly the same traffic and power profile between mesh nodes. When viewed as a holistic protocol, however, some important differences appear. Zigbee's cluster library has been refined over the years to properly define application operation in the most efficient way possible. Thread relies on IP-based protocols, which have been defined to be flexible and readable. This flexibility allows for extensibility, but at the cost of over-the-air time.

The flexibility of Thread is a result of its basis in IP. The emphasis on network-level interoperability enables the addressing of Thread devices by other IPv6-enabled devices through border routers. This means that a computer on the local network can talk to a Thread device in the same way that a cloud server can talk to a Thread device. This level of adjacent network connectivity is not immediately available in Zigbee networks.

Finally, one of the most important questions to ask about a protocol is what the ecosystem around the device looks like. Zigbee offers a very well-defined local ecosystem of automation products, with years of application interoperability. Thread enables communication frameworks that have been industry-tested for decades on IP-bearing networks.

Designers can use the [CC26x2r wireless MCU LaunchPad™ development kit](#) to evaluate both of these connectivity technologies.

References

1. [SimpleLink™ CC13x2 and CC26x2 software development kit](#)
2. Texas Instruments: [Breaking the 400-Node ZigBee Network Barrier TI's ZigBee SoC & Z-Stack Software](#)
3. [CC26x2r wireless MCU LaunchPad™ development kit](#)

Important Notice: The products and services of Texas Instruments Incorporated and its subsidiaries described herein are sold subject to TI's standard terms and conditions of sale. Customers are advised to obtain the most current and complete information about TI products and services before placing orders. TI assumes no liability for applications assistance, customer's applications or product designs, software performance, or infringement of patents. The publication of information regarding any other company's products or services does not constitute TI's approval, warranty or endorsement thereof.

Z-Stack™ is a trademark of Texas Instruments.
Zigbee® are registered trademarks of ZigBee Alliance.
Wi-Fi® is a registered trademark of Wi-Fi Alliance.
Bluetooth® is a registered trademark of Bluetooth SIG, Inc and used by Motorola, Inc. under license.
All trademarks are the property of their respective owners.

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2022, Texas Instruments Incorporated