Thread 1.4 for Smart Buildings White Paper

September 2026

Thread 1.4 for Smart Buildings
White Paper

September 2026

Revision History		
Revision	**Date**	**Comments**
1.0	September 2019	First Public Release
2.0	September 2022	Fully Revised Version
3.0	September 2026	Updated for Thread 1.3/1.4

Authors: Jorg Kennis, Klaus Waechter, Sujata Neidig, Lance Looper

Table of Contents:

1. Overview of Thread

Thread is an open standard for wireless communication, providing a native IP (Internet Protocol) solution for reliable, low-power, secure, device-to-device, application-agnostic communication. Built for the IoT, Thread is the premier IPv6-based solution running on existing and broadly supported IEEE 802.15.4 radio technology.

Thread offers a number of benefits compared to other solutions. Its IPv6 base enables flexible set-up, monitoring, data analysis, provisioning, and a direct connection without compromising encryption and security, from local networks all the way up to the cloud.

Thread also offers flexibility in choosing which application protocols to run, even allowing multiple application standards to run concurrently. This enables flexible integration with other technologies and maintains an open path towards future developments.

Since Thread is based on IPv6, the application layers can also run on existing IP infrastructure, including Wi-Fi and Ethernet, allowing the most suitable networking technologies to be used together. For example, a network could consist of lighting endpoints connected with Thread and lighting endpoints connected via PoE, all operating in one fully integrated environment.

Today, the majority of building networks consist of technology silos, which also require separate security and maintenance efforts. Thread connectivity serves to provide a common IP networking solution for easy integration with enterprise networks. This enables more flexibility in functionality planning, better system features, easier maintenance, and lower cost. Thread scales to thousands of robustly connected wireless IP devices and handles both local communication between devices and cloud connectivity over the internet. In terms of self-configuring and re-configuring, it brings a familiar way of setting up and managing the network for system administrators, without the need to worry about address and topology assignment. End-to-end routing and addressability allow IPv6 packets to securely move from one endpoint to the other, whether on the same Thread mesh network or across the world. Its 6LoWPAN foundation is based on low- power, widely deployed, proven, and low-cost IEEE 802.15.4 radio technology that supports sleepy nodes and reduces network overhead.

Thread is a seamless part of the enterprise network and allows the usage of various application-level protocols to integrate with existing infrastructure while maintaining the security and flexibility to commission and maintain end-devices. It also allows for the creation of various individual and group profiles and domains, even spanning multiple individual Thread mesh networks.

Thread's proven, secure, and widely deployed specification is currently in use for residential and smart building IoT applications. With its powerful enhanced features for specific commercial use cases, Thread 1.2 scaled to reliably connect thousands of wireless products in commercial-grade buildings.

Building on Thread 1.1's robust foundation, Thread 1.2 introduced a set of enhancements specifically aimed at large-scale enterprise implementations. These include enterprise-grade security mechanisms, support for larger and denser networks, improved low-power operation for battery devices, and mechanisms to connect multiple Thread meshes on the same LAN. Subsequent versions, Thread 1.3 and Thread 1.4, further extend these capabilities with standardised Border Router behaviour, richer diagnostics, and new tools that simplify commissioning and day-to-day operations in commercial buildings.

Key benefits of Thread for smart building deployments include:

- Enterprise-level security that builds on proven internet standards, with support for end-to-end encryption and secure commissioning.
- Scalable deployments that can span many Thread meshes while still appearing as a single, routable IP network to IT and BMS systems.
- Flexible segmentation options that allow designers to divide large estates into multiple Thread networks for resilience, policy enforcement, or organisational reasons.
- Automatic roaming for devices between nearby Thread networks, enabling continuous coverage across large buildings or campuses.
- The ability to assign and manage a secure identity for every device on the network, supporting inventory, access control, and lifecycle management.
- Direct access to rich operational data and advanced analytics via the building's existing IP infrastructure, without protocol translation.
- Streamlined handover between installers, commissioners, and facility managers, supported by standardised tooling and Border Router behaviour in Thread 1.3.
- Commissioning and expansion at scale, using Thread 1.4 features such as Thread Credentials Sharing, enhanced Internet connectivity, Thread-over-Infrastructure, and secure commissioning over authenticated TLS, which reduce truck rolls and on-site time.

Since the publication of Thread 1.2, the specification has continued to evolve. Thread 1.3 standardised critical capabilities in Thread Border Routers, such as bidirectional IP connectivity, service discovery, and improved TCP support to ensure multi-vendor interoperability and to prepare networks for application protocols like Matter, KNX IoT, and DALI+.

Thread 1.4 then added features such as Thread Credentials Sharing, improved Internet connectivity for both IPv4 and IPv6 environments, enhanced network diagnostics, Thread-over-Infrastructure, and secure commissioning over authenticated TLS. Together, these evolutions make Thread even better suited for professional smart-building deployments that require predictable operation at scale.

3. The Need for a Secure, Mesh, IPv6 Networking Solution

Historically, the network of constrained embedded devices that we now call the "Internet of Things" typically used specialised, and often proprietary, communication protocols. By contrast, the Internet and the World Wide Web are built on a layered stack of open standards, with each layer independent and not tied to a specific application. This is why the Internet and its applications are so flexible and pervasive.

In recent years, technology has emerged that enables manufacturers to use internet standards with these constrained embedded devices, allowing devices and applications to be developed independently and run anywhere: in the cloud, on mobile devices, on in-home devices, and on in-business devices. At the same time, developments in low-power network technology and compression and packet-handling standards, such as the IEEE 802.15.4 standard and 6LoWPAN, enabled great leaps forward in low power consumption and network efficiency.

Since we're building for the Internet of Things, Thread Group incorporated the Internet and its open standards to create an Internet Protocol (IP) version 6 (IPv6)-based mesh networking protocol with 6LoWPAN as its foundation, running on standard IEEE 802.15.4 radios.

3.1. The Need for IPv6

IPv6 is the Internet's next-generation network protocol, designed to supplement and eventually replace the earlier version of the Internet Protocol, IP version 4. IPv6 provides a number of critical advantages:

- **Extreme address scalability**: In order to communicate over the Internet, computers and other devices must have unique sender and receiver addresses. These numeric addresses are known as Internet Protocol (IP). addresses. IPv6 allows for literally trillions of IP addresses, providing virtually unlimited address scalability.
- **End-to-end routing and addressability**: Two IPv6 endpoints, whether on the same Thread mesh, across networks, or across the world, can communicate end-to-end with straightforward and well-understood
- **Internet routing**, moving packets from one endpoint to the other.
- **Easy manageability**: Unlike IPv4 networks, IPv6 networks are inherently auto-configuring, removing the need for users and system

administrators to worry about address assignment, DHCP servers, etc.
- **Well-known and familiar**: IP networks are well-known to engineers, developers, technicians, and system administrators, leveraging a rich base of tools, knowledge, and experience.
- **Flexible**: IPv6 supports a wide variety of upper-layer protocols, including transports, sessions, security, and applications, and IPv6 is supported over a wide range of wired (e.g., Ethernet) and wireless (e.g., Thread, Wi-Fi) networking protocols.
- **Compatible**: IPv6 and IPv4 can coexist using well-understood and widely deployed techniques. IPv6 networks such as Thread can be deployed within IPv4 infrastructures.

3.2. The Need for Low Power

The Thread specification is designed from the ground up to enable extremely low power consumption and efficient device communication, without sacrificing a positive end-user experience. Two technologies in particular, the IEEE 802.15.4 standard and 6LoWPAN, form the backbone of Thread's low power solution.

As a point of comparison, in Ethernet links, a packet with the size of the IPv6 MTU (1280 bytes) is typically sent as one frame over the link. In the case of 802.15.4, 6LoWPAN acts as an adaptation layer between the IPv6 networking layer and the 802.15.4 link layer. It solves the issue of transmitting an IPv6 MTU by fragmenting the IPv6 packet at the sender and reassembling it at the receiver. 6LoWPAN also provides a compression mechanism that reduces the IPv6 header sizes sent over the air and thus reduces transmission overhead. The fewer bits sent over the air, the less energy is consumed by the device. Thread makes full use of these mechanisms to efficiently transmit packets over the 802.15.4 network.

Another important feature of 6LoWPAN is the ability to provide link-layer packet forwarding. It provides a very efficient and low-overhead mechanism for forwarding multi-hop packets in a mesh network. Thread uses IP layer routing with link-layer packet forwarding. It makes use of the 6LoWPAN link-layer forwarding capabilities to forward a packet, which avoids having to send it up to the network layer during packet transit. Thread uses the MAC layer to provide addressing based on short addresses (16-bit length) to further reduce the information sent over the air. This saves processing cycles and improves power efficiency, while still using an IP-based routing protocol.

Thread's low-power advantages:

- Extensive support for sleepy nodes, which allows for years of device operation, even on a single AA battery
- Based on the power-efficient IEEE 802.15.4 MAC/PHY
- Short message size conserves bandwidth and power

- Streamlined routing protocol reduces network overhead and latency
- Designed to run on readily available, low-power wireless system-on chips

3.3. Security at Every Layer of the Stack

Thread provides end-to-end security because of its IP-based foundation. There is no need for translation or decryption and re-encryption of data packets. A Thread Border Router simply transfers the data packets to and from the Thread network to the rest of the IP network infrastructure.

The Thread network is protected with a network-wide key, which is used at the MAC (Media Access Control) layer to protect the IEEE 802.15.4 MAC data frames. This is an elementary form of security used to prevent casual eavesdropping and targeted disruption of the Thread network from outsiders without knowledge of the network-wide key. As it is a network-wide key, compromise of any Thread Device could potentially reveal the key; therefore, it is not typically used as the only form of protection within the Thread network. In addition, Thread uses the DTLS protocol for internet-grade end to-end security when commissioning, and may use CASE, PASE, TLS or other encryption methods, based on the used application layer.

From the point of view of joining new devices into a network, the possession of the network key is used to discriminate between an authenticated and authorised Thread Device and the joining device (in its initial state). The network-wide key, along with other network parameters, is delivered securely to a joining device. This way, the network key is never exposed in the clear on a wireless link.

Recent versions of the Thread specification extend this security model with standardised mechanisms designed for professional deployments. Thread 1.4 introduces Thread Credentials Sharing, which uses a temporary DTLS-protected channel and one-time administrative passcodes to securely share network credentials between Border Routers, tools, and ecosystems without exposing long-lived secrets. It also adds Thread Commissioning over Authenticated TLS (TCAT), an optional, certificate-based commissioning method that enables fast and secure onboarding of large numbers of devices in commercial environments. In addition, enhanced diagnostic and Internet-connectivity features in Thread 1.4 are explicitly designed to operate over authenticated, encrypted links, allowing installers and operators to gain deep visibility into the network and integrate with cloud services while preserving strong end-to-end security.

3.4. Benefits for Building Automation and Lighting Control Installations

In addition to the general benefits of IPv6 and low power consumption, Thread's wireless mesh IPv6 solution offers further specific advantages to a variety of applications:

- Professional building owners will be able to install both small networks (such as a quick-fix retrofit RF solution) and very large building networks (thousands of nodes) using the same networking technology, simplifying and streamlining network installation and management processes and overhead.
- Wireless IPv6 technology is forward-compatible, more sustainable, and offers greater economies of scale than alternative technologies, so networking solutions become more cost-efficient and add functionality more easily.
- Network managers, application managers, and system integrators can independently execute tasks on the system, easing network management bottlenecks.
- A native IP system can collect and process vast amounts of data, from a broad range of intelligent connected devices, in near real-time. System data can then be accessed directly or via the cloud. Unique value can be extracted through advanced analytics.
- The flexibility of the application stack enables more sophisticated occupant individual control possibilities, increasing the levels of convenience, productivity, and operational efficiency.
- The network is easier to reconfigure than alternative technologies, implying decreased refurbishment costs and lower churn cost in buildings.
- The technology further enables remote and predictive maintenance, provides more opportunities for notifications on maintenance tasks, and simplifies system-wide fixes, software updates, and network reconfigurations.
- During building refurbishment, wireless technology requires less physical alteration of the site, reducing installation costs and offering increased flexibility and installation speed.

4. Topology of Thread in a Smart Building Network

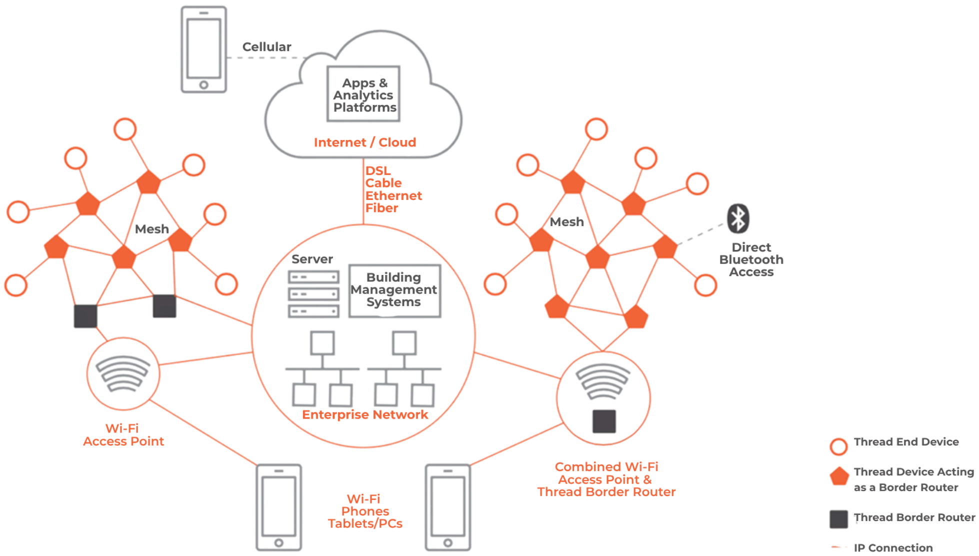

Figure # 1: Topology of Thread in a Smart Building Network

The above topology diagram shows an example of how Thread can be implemented in an enterprise network environment. The diagram illustrates Thread's key advantage over other networking technologies: it is fully based on IPv6. Thread's IPv6-based foundation brings many advantages to the Smart Building.

Not only can it leverage widely proven cybersecurity mechanisms, but it can also integrate seamlessly into the existing building network infrastructure. Contrary to legacy automation systems, which are usually siloed in separate networks for each application, multiple applications, even those based on different standards and protocols, can share the same network at the same time, and use the most appropriate physical network type for each individual device. Thread's low-power mesh network is ideal for power-constrained device types covering wide areas in a building.

In this section we will explain the various parts of a Thread network, describe their purposes, and explain how they operate in the context of a wider building network. It shows how easy it is to integrate with existing products and how it allows for the maintenance and expansion of the network by network managers, independent from application and system integrators.

4.1. Thread Mesh Extender Device

Thread is a mesh network technology, which means a device on the network can not only receive data, but also pass data along to other devices. In this way, these devices act as Thread Mesh Extender devices. This results in a very stable network with a large reach, without the need for additional repeaters that rebroadcast wireless signals to devices farther away.

4.2. Thread Battery Operated Device

Thread Battery Operated Devices are devices on the network that generally only operate when requested, like light switches. They don't reroute data and can be sleepy devices to save energy, only to be activated and immediately become part of the Thread network upon use.

4.3. Thread Border Router

A Thread Border Router provides the connection between a Thread mesh network and the wider IP infrastructure of a building or campus. It terminates the 802.15.4 / Thread mesh on one side and connects to Ethernet or Wi-Fi on the other, routing IPv6 traffic between Thread devices and the rest of the network. This allows building-management systems, cloud services and other IP-based applications to communicate with low-power devices using standard IP protocols, without needing proprietary gateways or protocol translation.

In many professional deployments, Thread Border Router functionality is integrated into existing network infrastructure. Manufacturers of wireless access points and building gateways increasingly offer products that combine Wi-Fi and Thread radios in a single device, often together with Thread Border Router capabilities. This reduces the number of boxes that need to be installed and simplifies design, deployment, and maintenance: the same infrastructure that provides Wi-Fi coverage for users can also provide Thread coverage for low-power building devices. Thread Mesh Extender Devices and battery-operated Thread end devices still require at least one Thread Border Router to communicate with systems outside the Thread mesh itself, such as building management servers or cloud services; without a Border Router, they can only form an isolated Thread network.

Recent versions of the Thread specification formalise a common feature set for Thread Border Routers to ensure consistent behaviour and strong interoperability across vendors. This includes standardised bidirectional IPv6 connectivity between Thread and the enterprise network, support for service discovery, and improved handling of TCP traffic within the mesh. Enhanced Internet connectivity features make it easier for Thread devices to reach cloud services in both IPv6- and IPv4-dominated environments, for example, by using DHCPv6 Prefix Delegation, NAT64, and DNS64 where appropriate. As a result, modern Thread Border Routers can be treated as first-class citizens in enterprise network designs, analogous to Wi-Fi access points for low-power devices, and can be deployed, monitored, and managed using familiar IT and OT operational practices.

4.4. Building Automation Standards

Indicated in figure number 2 below, various Building Automation Standards can be used on a Thread network. As with other IP-based applications and devices (like printers, servers or copiers), multiple application types (like lighting, access control or climate systems) can share that same network, and these applications can be based on different protocols, like DALI+, KNX IoT or Matter.

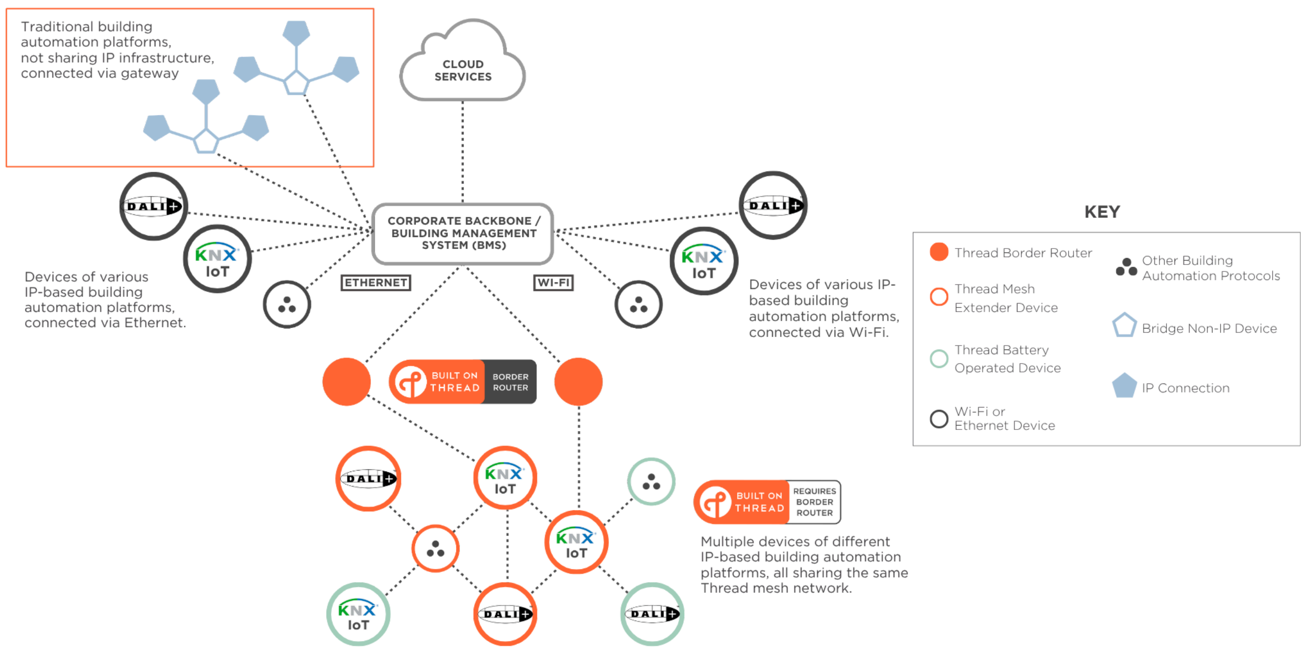

Figure # 2: Topology of Thread with Building Automation Standards

Since the first publication of this white paper, these standards have continued to evolve their IP-based offerings. KNX IoT and DALI+ now provide mature specifications for professional building automation over IP, and the Matter standard has added richer support for energy-management devices such as large appliances, EV charging equipment, solar, storage and HVAC systems. This evolution reinforces the value of a common, secure, low-power IP mesh like Thread that can host multiple application protocols on the same physical network.

4.5. Building Management System

A Building Management System (BMS) may be part of the enterprise network. Such a BMS can directly address each individual Thread device, as Thread's IPv6 connectivity is fully transparent to the existing enterprise network. Likewise, authorised PCs, laptops and mobile devices like smartphones and tablets are directly connected to the enterprise network and can also immediately access Thread devices using the existing IPv6 networking protocol.

4.6. Cloud Connection

IP makes the life of system maintainers and IT staff much easier. All devices, including those on the Thread wireless mesh, can be remotely configured and maintained using familiar tools and technologies. It offers easy remote control, data collection and monitoring from anywhere in the world, while maintaining full end-to-end encryption.

4.7. Bridge to Traditional Non-IP Systems

Many buildings have legacy building automation systems in place that will only over time make the full transition to IP to unlock the many benefits as described above. They are either based on other physical network standards or non-IP-based application protocols. These systems can be integrated with the rest of the network by means of a gateway that allows access to the respective network technology and application protocol.

5. Thread Supports Multiple Application Layers

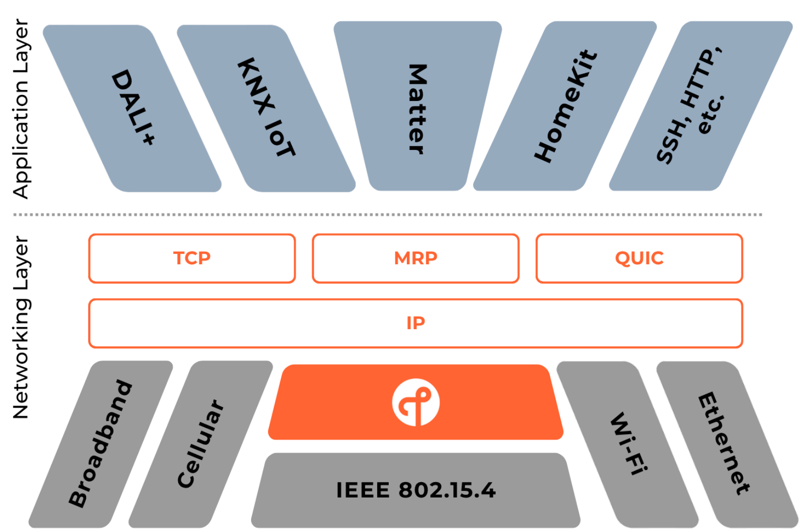

Figure # 3: Thread over IP with Multiple Applications Layers

Thread’s application support is comparable to the way Wi-Fi works with multiple applications. By simply joining one Wi-Fi network, due to the common Internet Protocol, we can access every service, regardless of who is offering it or from wherever it is being provided.

Today's wide offering of IoT solutions often requires that we set up and maintain a network for every protocol we want to use. This not only requires surveillance and maintenance to keep them up and running, but networks might also interfere with each other. Furthermore, since they are separate networks, devices do not "talk" to each other, and there is no way to control them effectively from a single location, monitor and analyse their data, or provision them over various locations and users, in an end-to-end secure manner.

This is where Thread comes in. Like Wi-Fi, it can run every modern IoT application that runs over IP, using a common physical network.

Having one single network for all your IoT devices provides major benefits. These devices do not interfere with each other, but instead can work together to extend the reach of the network by forming a mesh. Since the entire network is IPv6-based and does not need "conversion" of application data, encryption of data packets remains intact throughout the entire chain, all the way from the individual device up to the cloud. Every device is uniquely addressable, making it easier for system administrators to commission them individually or as a group, and to flexibly change those arrangements. All devices can be managed and monitored from a single location, opening the possibility for powerful data analysis.

What's more, Thread is designed so that in networks consisting of devices that use different application protocols, all these benefits remain in effect. Thread can run multiple application layers concurrently on the same network. This is simply not possible with other IoT solutions on the market today. If you use multiple devices based on different systems, all devices require their own network, which often creates maintenance issues. And if in the future you want to replace or add an application layer, you need to replace the underlying wireless network. As with Wi-Fi, Thread allows you to simply start using the new application on the existing network.

All popular IoT-standards for building automation control recognise the importance of moving towards a universal, secure and interoperable IPv6-based network. And most of them have chosen Thread to be their future proof solution. This includes DALI+, KNX IoT, Matter, and BACnet. And since Thread is fully agnostic to the application layer, it is even possible to design and develop a custom application layer to suit specific needs (e.g. for unique factories and machinery, or medical applications).

Application standards will continue to focus on compelling new use cases, satisfying new IoT application requirements and interoperability aspects between different applications on the same network.

Instead of developing yet another IoT protocol, Thread combines current and proven Internet-based technologies into a standard that's optimised for security and mesh networking in low-cost, low-power devices. Thread is transparent to the applications that run on the network, which makes it the only future-proof choice for modern IP-based networks for IoT devices.

In addition to traditional lighting, HVAC and access-control use cases, energy-management applications are increasingly moving to IP-based application protocols that can run over Thread. Standards such as Matter 1.3 and 1.4, for example, now define device types and clusters for EV charging equipment, solar generation, home batteries and other distributed energy resources, all of which can use Thread as their low-power mesh networking foundation.

6. IPv6 Thread Networks in an IPv4-Based Enterprise Network

When defining requirements for building automation, network designers want to make sure that their solution operates well within the existing network. At the same time, they need assurance that the solution is both future proof and scalable. This is also true for developers defining smart products for the home.

Thread is based on existing and widely deployed standards, like IPv6, and has been specifically designed for technologies that provide long-term deployment. Due to its extremely large addressing space, IPv6 allows every single device to have its own unique address which has not been possible before.

IPv6 is the successor to IPv4, the "legacy" Internet Protocol. IPv4 is known for its xxx.xxx.xxx.xxx addressing structure, where xxx can be a number between 1 and 255. With IPv4, approximately 4 billion unique addresses are possible, which is not nearly enough to cover every device on the planet.

To overcome the limitations of IPv4, local networks (including enterprise networks) use a technology called NAT (Network Address Translation) to act as an agent between the local network and the public Internet. NAT functionality is included in a single device such as a router, allowing one IPv4 address to represent a larger group of computers to the outside network.

However, this form of address translation has several limitations which do not solve all the issues that arise due to IPv4's limited addressing-range. This is why IPv6 was developed.

IPv6 allows for 340,282,366,920,938,000,000,000,000,000,000,000,000 IP addresses. In addition to the extended addressing space, IPv6 adds several other benefits while fixing some shortcomings of IPv4. It has built-in Quality of-Service allowing delay-sensitive packets to get priority in the data transfer, and includes automatic address configuration by default–eliminating complicated manual address assignment. And its improved header structure results in less processing overhead.

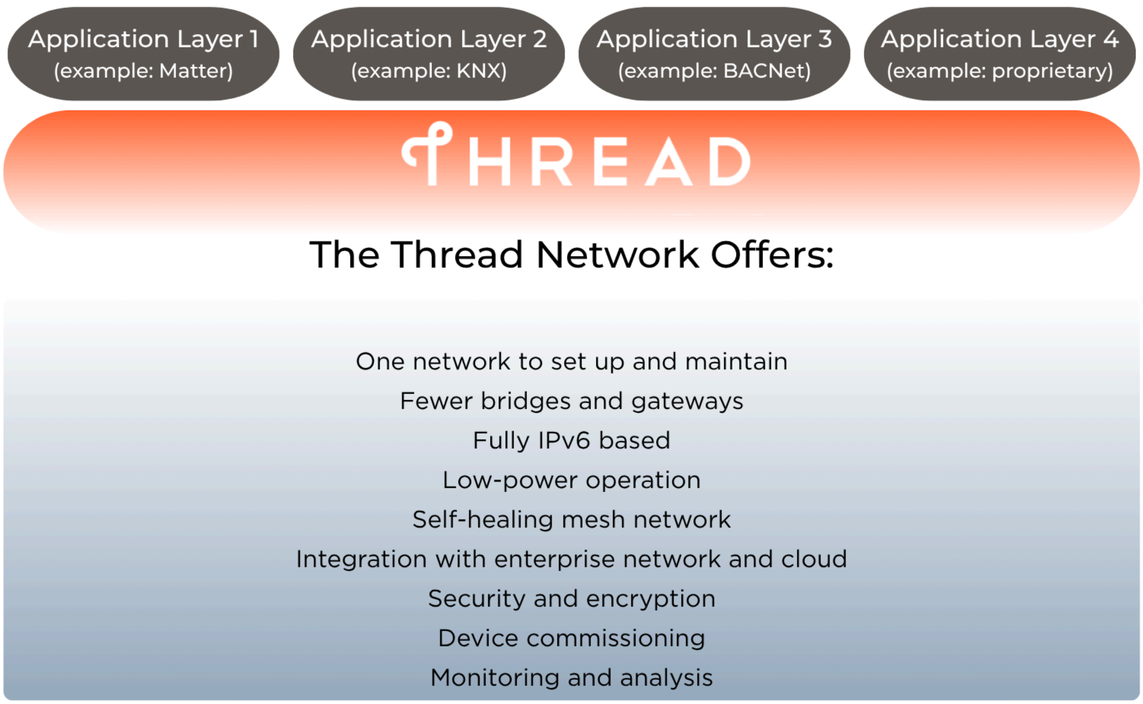

Figure # 4: Thread Network Offerings

More and more Internet Service Providers and enterprise networks support IPv6. For these networks, Thread can be integrated simply without the need for address translation. For networks that still run on the legacy IPv4 protocol, some adaptations are needed. Luckily, much of this is handled by the Thread Border Router which is the wireless access point that connects the Thread network to the enterprise network and thus also to the cloud.

Thread defines an IPv6 transport over an IEEE 802.15.4 data-link layer. A 6LoWPAN adaptation layer in between implements IPv6 header compression to maximize efficiency. In the context of the OSI layers, only the network layer is affected where IPv4 networks are used, as all of the other layers are fully transparent to the existing network.

A Thread Border Router is a function of a powered Thread device that connects the Thread wireless network to the rest of the enterprise network. Unlike "gateways" found in many wireless solutions, it is fully transparent to the transport and application protocols that reside above the network layer.

As a result, applications can communicate securely from end-to-end without any application-layer translation. Furthermore, since Border Routers do not maintain any application-layer state, there can be multiple Border Routers in a network, eliminating a "single point of failure" in the event one of them malfunctions.

The Border Router enables every Thread device to directly connect to global cloud services, when enterprise networks run IPv6 and IPv4, or IPv4 only.

So how can a Thread network connect to an IPv4 network? A Thread Border Router typically handles all the requirements. A Border Router is not required to form a local Thread mesh-network. It is required to connect a Thread network to the outside world via the local home or enterprise network.

Typically, Border Routers implement NAT64 communication between IPv6-based Thread devices and IPv4 networks. NAT64 translates IPv6 packets to IPv4 and vice versa. The Border Router functions as an IPv4 host on the wide area network, capable of obtaining an IPv4 interface and router address. It will acquire an address using DHCP from an IPv4 address pool. The Border Router may also implement Port Control Protocol (PCP) to control how incoming IPv4 packets are translated and forwarded and support static mappings.

Most of the IPv4 to IPv6 (and vice versa) translations can be handled by the Thread Border Router, with minimal changes needed to the existing network. This means that Thread can be immediately implemented in current working situations, while ensuring that the Thread network is ready for the future: when the time comes for a partial or full transition to IPv6.

Thread 1.4 refines this model with standardised mechanisms for both IPv4 and IPv6 Internet connectivity. For IPv6-enabled environments, Thread Border Routers can obtain globally routable prefixes via DHCPv6 Prefix Delegation and distribute them to Thread devices, while in predominantly IPv4 environments they can use NAT64 and DNS64 to bridge traffic to IPv4-only services. These capabilities make it easier for facility and IT teams to expose selected devices or services to cloud-based management platforms without custom gateways or protocol conversions.

7. Key Points and Conclusion

Thread provides a **secure, low-power, IP-based mesh network** for connected devices in smart buildings. By reusing proven Internet standards and integrating cleanly with existing IP infrastructure, it allows building systems to evolve from siloed, proprietary networks towards converged, flexible architectures.

Key advantages of Thread for smart buildings include:

- Native IPv6 networking with low-power mesh characteristics
- Robust, self-healing topology with no single point of failure
- Layered, standards-based security and secure commissioning
- Clean integration with building and enterprise IP networks via Thread Border Routers
- Support for multiple application standards such as KNX IoT, DALI+ and Matter on the same physical network
- Flexibility to segment networks by zone, tenant, or function, while still maintaining a coherent IP architecture
- Tooling and diagnostics suited to professional installation, commissioning, and operations

In the ever-expanding universe of IoT applications, the smart-building sector continues to be one of the most promising areas for Thread adoption. Thread 1.2 laid the groundwork for robust, scalable commercial networks; Thread 1.3.0 and Thread 1.4 add standardised Border Router behaviour, commissioning at scale, Thread-over-Infrastructure, and enhanced Internet connectivity that align tightly with professional installation and operations. Combined with IP-based application standards such as KNX IoT, DALI+, and Matter, Thread enables converged networks that meet modern expectations around interoperability, cybersecurity, energy management, and lifecycle flexibility.

For building owners, system integrators, and product manufacturers, Thread offers a long-term foundation on which new applications and services can be added over time, without redesigning the underlying wireless infrastructure.