Thread 1.4 Features White Paper

September 2024

This Thread Technical white paper is provided for reference purposes only. The full technical specification is available publicly. To gain access, please follow this link: https://www.threadgroup.org/ThreadSpec.

If there are questions or comments on these technical papers, please send them to help@threadgroup.org.

This document and the information contained herein is provided on an "AS IS" basis and THE THREAD GROUP DISCLAIMS ALL WARRANTIES EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO (A) ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OF THIRD PARTIES (INCLUDING WITHOUT LIMITATION ANY INTELLECTUAL PROPERTY RIGHTS INCLUDING PATENT, COPYRIGHT OR TRADEMARK RIGHTS) OR (B) ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, TITLE OR NONINFRINGEMENT.

IN NO EVENT WILL THE THREAD GROUP BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OF DATA, INTERRUPTION OF BUSINESS, OR FOR ANY OTHER DIRECT, INDIRECT, SPECIAL OR EXEMPLARY, INCIDENTAL, PUNITIVE OR CONSEQUENTIAL DAMAGES OF ANY KIND, IN CONTRACT OR IN TORT, IN CONNECTION WITH THIS DOCUMENT OR THE INFORMATION CONTAINED HEREIN, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

Thread 1.4 Features White Paper

September 2024

Revision History

Revision	Date	Comments
1.0	September 4, 2024	First public release

Contents

1. Authors: Saurabh Kumar, Esko Dijk

2. Introduction

The Internet of Things (IoT) continues to evolve rapidly, with connected devices proliferating across homes, businesses, and industries. As this landscape grows more complex, ensuring seamless interoperability, robust reliability, and flexible scalability becomes paramount. Recognizing these challenges, the Thread Group, an industry alliance driving the adoption of the Thread wireless IPv6 networking protocol, has unveiled the Thread 1.4 Specification. This latest update to the Thread specification builds upon the proven foundation of the Thread low-power mesh networking technology and adds six significant new features and enhancements to the existing Thread 1.3 Specification.

Informed by real-world experiences and feedback from Thread Group members, these enhancements are designed to simplify the way developers, product makers, and businesses approach the design and deployment of Thread-enabled devices. By addressing critical areas such as device commissioning ("onboarding"), security credentials management, public internet connectivity, network diagnostics, and infrastructure-network integration, Thread 1.4 aims to unlock new opportunities for IoT innovation while delivering a more seamless and secure connected experience to users.

The key features and enhancements introduced in Thread 1.4 include:

- Thread Credentials Sharing
- Improved internet connectivity
    - Works for IPv4 internet access (using NAT64 and DNS64)
    - Works for IPv6 internet access (using DHCPv6 Prefix Delegation)
- Enhanced network diagnostics
- Thread-over-Infrastructure
- Secure Thread Commissioning over Authenticated TLS (TCAT)
- Numerous enhancements to mesh robustness and scalability

With these enhancements, product managers and developers of smart home and commercial building devices leveraging Thread at the network layer will have unprecedented opportunities to innovate, reduce overhead and enable richer end-user experiences that foster brand loyalty. Professional installers can benefit from improved profitability and productivity, with streamlined installation processes, enhanced troubleshooting capabilities, and reduced on-site time and support requests. Ultimately, end-users of Thread 1.4-enabled devices will enjoy a more robust, scalable, and secure smart home/building experience, unlocking the full potential of the connected world.

In the following sections, we will delve deeper into each of these Thread 1.4 enhancements, exploring their technical details, benefits, and potential impact on the IoT ecosystem.

3. Thread 1.4 Features

3.1 Thread Credentials Sharing

Adding a new device to an existing Thread network requires a user to retrieve the Thread network credentials, typically from a secure key storage on a mobile device. This could be a barrier for users/installers who don't have these credentials stored on the particular mobile device that is used to commission the new Thread device. Thread 1.4 introduces Thread Credentials Sharing to address this challenge.

3.1.1 The Problem: No Access to Network Credentials

The core problem was that the Thread network credentials, needed for commissioning a new Thread device, were in certain cases not accessible to the (mobile) application assisting the commissioning process. In a typical commissioning session to onboard a new Thread device, the mobile application retrieves these credentials from a secure key store – where these were previously stored – with the user's permission. This is useful because a user doesn't have to remember any credentials such as the Commissioning Credential. Note that although the Commissioning Credential is similar to a Wi-Fi password, this credential is often not made available to the user to keep the Thread onboarding experience simple – with no additional password to create or remember – and secure. Furthermore, the Network Key is not user-generated at all, to guarantee a high level of security, and it cannot be easily remembered.

However, in some cases, the mobile application may not have access to these credentials – for example, because the application resides on a different mobile device, or a different user has set up the devices in the Thread network so far.

Without these credentials, adding a new Thread device to an existing network is impossible due to Thread's strong security. And setting up a new Thread Border Router is possible but would typically cause the forming of an entirely new Thread network, in a home where one already exists. This fragmentation of Thread devices over multiple networks could weaken the overall mesh network resilience and connectivity.

3.1.2 The Solution: Streamlined Credentials Sharing

Thread Credentials Sharing provides a secure and user-friendly way to grant temporary administrative access to an existing Thread network. This access allows for the extraction of the necessary network credentials, which can then be used to commission new Thread devices seamlessly. The process leverages a short-lived, single-use key called the Ephemeral Pre-Shared Key for the Commissioner (ePSKc), derived from a user-friendly Thread Administration One-Time Passcode, which is remarkably similar to the one-time numeric codes that users are already familiar with for login purposes.

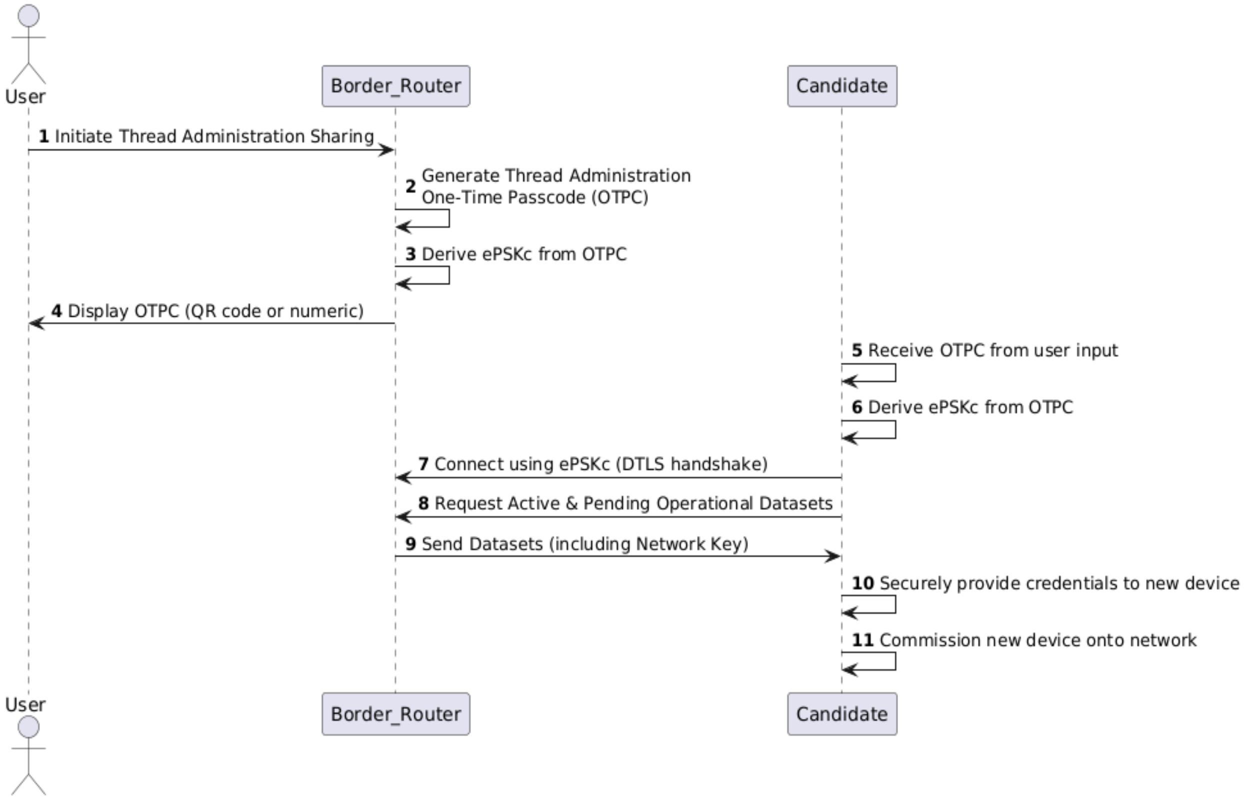
This approach makes it as easy as possible for the user, while still meeting the strict security requirements of Thread. This feature can be compared to the "forgot my password" option that many websites offer today. The procedure described below only needs to be executed in exceptional cases where the Thread network credentials are unavailable to the user performing the onboarding of a new Thread device.

3.1.3 How It Works: Step-by-step Overview

The Thread Credentials Sharing procedure includes the steps detailed below, which are also detailed in the sequence diagram that follows.

1. **Automatic Detection of Need for Sharing:** A user wishes to add a new Thread device, but the assisting application finds no credentials. The application does discover (via DNS-SD) that a Thread 1.4 Border Router with Thread Credentials Sharing support is available on the local home network. The application can optionally detect which credentials sharing process needs to be used and guide the user toward initiating it.

2. **User Initiates Sharing:** Based on the application's suggestion, the user starts the Thread Credentials Sharing process on their existing network's Thread 1.4 Border Router. This involves a vendor-specific authentication step to ensure that only authorized users can gain administrative access. Typically, a UI on the Border Router or a mobile application managing the Border Router is used to accomplish this.
3. **Passcode Generation:** The Border Router randomly generates a Thread Administration One-Time Passcode (OTPC).
4. **ePSKc Derivation:** The Border Router derives the ePSKc from the OTPC.
5. **Passcode Presentation:** The passcode is presented to the user as a QR code and/or a nine-digit numeric code. This passcode is the unique secret that temporarily provides administrative access to the Border Router's Thread network.
6. **Passcode Entering:** If a QR code is displayed, the user can scan the QR code using the application assisting the user in the onboarding process. Otherwise, the user types the passcode into the application.
7. **ePSKc Derivation:** From the passcode, the ePSKc security credential is derived by the assisting application.
8. **Candidate Connects:** The application assisting the user (called the "Candidate" because it is still a candidate for getting access to the credentials) uses the ePSKc to establish a secure DTLS connection to the Border Router. This connection acts as a temporary gateway to the Thread network. The Border Router verifies that the correct ePSKc is known by the application and grants access.
9. **Credentials Request:** The Candidate (application), now with temporary administrative privileges, can extract the necessary credentials from the existing Thread network, as well as other configuration parameters of the network stored in the Active and Pending Operational Datasets (e.g., channel, PAN ID, etcetera).
10. **Credentials Response:** The Candidate receives the requested credentials and other data.
11. **Using the Credentials:** The extracted credentials can now be securely stored on key storage on the mobile device running the application – this is important to make future onboarding of new Thread devices easier than using the present (exceptional) procedure. Also, the credentials can be provisioned directly into a new Thread device, allowing it to join the existing Thread network without the need for further complex user input.


User
Border_Router
Candidate
1 Initiate Thread Administration Sharing
2 Generate Thread Administration One-Time Passcode (OTPC)
3 Derive ePSKc from OTPC
4 Display OTPC (QR code or numeric)
5 Receive OTPC from user input
6 Derive ePSKc from OTPC
7 Connect using ePSKc (DTLS handshake)
8 Request Active & Pending Operational Datasets
9 Send Datasets (including Network Key)
10 Securely provide credentials to new device
11 Commission new device onto network
User
Border_Router
Candidate

3.1.4 Thread Credentials Sharing: Support in Thread 1.4 Border Routers

This section focuses on the technical details of functions that a Thread 1.4 certified Border Router must be able to support, to play its role (when needed) in the Thread Credentials Sharing procedure. In technical terms, this support is called "ePSKc mode". The functions include the following:

1. **Always-on ePSKc Support Signaling:** The Border Router continuously signals its capability for Thread Credentials Sharing through a flag in its DNS-SD (mDNS) advertisement records.
2. **User-Triggered ePSKc Activation:** The Border Router enters ePSKc mode only when a user initiates the specific procedure, which must include user authentication.
3. **Passcode Generation and Display:** Once in ePSKc mode, the Border Router generates a unique Thread Administration One-Time Passcode and derives the corresponding ePSKc. It then displays this passcode as three groups of three digits, along with a clear message about the

implications of sharing it. A QR code version is optionally shown. This display may occur on the device itself, or on a remote display such as a mobile app.

4. **Secure One-time Connection:** The Border Router, while in ePSKc mode, operates normally and also waits for a potential Candidate to connect using a DTLS connection secured by the ePSKc shared secret. Upon successful authentication, it responds to requests for the Thread Active and Pending Operational Datasets, effectively handing over the Thread network credentials to the Candidate.
5. **Timed Deactivation:** After the Candidate disconnects or a timeout occurs, the Border Router deactivates ePSKc mode, deleting the temporary passcode and ePSKc. This ensures that the elevated access is short-lived and single-use: a security measure to prevent brute-force connection attacks.

3.1.5 Use Cases: Where Thread Credentials Sharing Delivers Value

1. Onboarding of New Thread Devices:

- **For Users:** The most prominent use case is to streamline the addition of new devices to an existing Thread network. Users with limited technical knowledge can easily grant temporary access, allowing a commissioning app or tool to obtain the necessary network credentials (Network Key, Commissioning Credential, PSKc). This eliminates the need for manual entry or knowledge of long, complex credentials. Also, it increases the success rate for new device onboarding.
- **For Installers:** Professionals setting up Thread networks in homes or businesses benefit from a more efficient process. They can quickly and securely add devices without requiring manual entry of complex credentials, reducing setup time and potential errors.

2. Enhanced Mesh Network Resilience and Connectivity:

- **Avoiding Network Fragmentation:** By facilitating the joining of new devices to existing Thread networks, Thread Credentials Sharing helps prevent the creation of multiple, isolated Thread networks within a single location. This promotes a more cohesive and robust mesh network, improving overall coverage and reliability.

- **Better Device Interactions:** With a unified network, devices can communicate more effectively, enabling features like smart home automation routines that span multiple device types and brands.

3. Greater Flexibility and Interoperability:

- **Multi-Vendor Ecosystems:** Users can more easily integrate devices from different manufacturers or different ecosystems into their Thread network. This is because the commissioning app or tool, once granted access, can obtain the necessary credentials regardless of the new device's origin.
- **Future-Proofing:** As Thread evolves and new Thread devices with varying credentials requirements emerge, the standardized sharing mechanism ensures compatibility and adaptability.

4. Streamlined Troubleshooting and Network Management:

- **Access for Diagnostics:** In cases where network issues arise, Thread Credentials Sharing provides a secure way for authorized technicians or support personnel to access the network for troubleshooting. They can extract diagnostic information or temporarily assume control to identify and resolve problems.
- **Remote Management:** In scenarios where remote network management is desired, the sharing mechanism offers a controlled way to grant access to authorized entities without compromising overall network security.

3.2 Enhanced Internet Connectivity

Thread 1.4 introduces significant enhancements to how Thread devices can connect to the public internet. This expansion is achieved through robust support for both IPv6 and IPv4, ensuring compatibility with the existing internet while preparing for the future.

3.2.1 IPv6: A Direct Connection

In-home networks or infrastructure networks that fully support IPv6, Thread devices can now obtain globally routable IPv6 addresses. This is accomplished through DHCPv6 Prefix Delegation (PD), a mechanism where the Thread 1.4 Border Router (the gateway between the Thread network and the internet) requests a block of IPv6 addresses from the internet service provider. Addresses from this block are then automatically configured on all Thread devices, enabling them to communicate directly with other IPv6-enabled devices and services anywhere on the internet.

3.2.2 IPv4: Bridging the Gap

To ensure compatibility with the large number of home networks, infrastructure networks, and internet servers that still rely on IPv4, Thread 1.4 employs a combination of Network Address Translation IPv6-to-IPv4 (NAT64) and Domain Name System IPv6/IPv4 synthesis (DNS64). NAT64 translates IPv6 packets from Thread devices into IPv4 packets, allowing them to communicate with IPv4-only destinations or to transit IPv4-only infrastructure networks. DNS64 synthesizes AAAA records (IPv6 addresses) from A records (IPv4 addresses), enabling Thread devices to discover IPv4 internet services using standard DNS queries and access these via IPv6.

3.2.3 Enhanced Internet Connectivity: Under the Hood

Thread's ability to interact with the internet is a result of a well-coordinated effort between the Thread 1.4 Border Router and Thread devices. Let's delve deeper into how this is achieved, focusing on both inbound and outbound IP connectivity:

1. **Border Router Configuration:** The Thread 1.4 Border Router acts as the IPv6 router between the Thread network and external networks, including the user's home network and the larger internet. It is by default configured to support external communication, either by obtaining a block of global IPv6 addresses using Dynamic Host Configuration Protocol IPv6 (DHCPv6) Prefix Delegation for use by Thread devices or by automatically enabling NAT64/DNS64 for compatibility with IPv4 services and networks where needed.
2. **Network Data Propagation:** The Thread Border Router shares its external connectivity capabilities with the Thread Leader, which then disseminates this information throughout the Thread network in the form of Thread Network Data. This ensures all Thread devices are aware of the best available paths to reach the internet via Border Router(s).
3. **Packet Forwarding (Outbound):** When a Thread device initiates communication with an internet device or service, it sends IPv6 packets

to a selected Border Router. The Border Router inspects the destination IPv6 address. If it's a native IPv6 address, the packet is routed directly. If it's an IPv4 address embedded into an IPv6 address, NAT64 translates the packet first into IPv4 before forwarding it to the IPv4 destination.

4. **Packet Forwarding (Inbound):** When an internet device or service responds to a communication request by a Thread device, the 1.4 Border Router receives the incoming packets via the home router or site IP router. The router's firewall typically applies packet filtering to ensure that only valid, expected traffic can enter the site. For IPv6 packets, the Border Router performs additional filtering and necessary translations (e.g., to 6LoWPAN fragments) and then forwards the packets to the appropriate Thread device based on its IPv6 address. For IPv4 packets, NAT64 on the Border Router checks if the packets are expected, and if so, translates these into IPv6 format before forwarding into the Thread network.

In essence, the Thread 1.4 Border Router acts as a translator, forwarder, and traffic cop (filter), managing the flow of information between the Thread network and the internet. This intricate mechanism, while automatically set up and transparent to the end-user, is what allows Thread devices to communicate seamlessly with the broader internet, opening up a world of possibilities for connected applications and services.

Especially for inbound packet forwarding from internet sources, there are a number of security considerations to highlight:

- **NAT64**: When NAT64 is used, it provides an additional layer of security by hiding a Thread device's IPv6 address behind the Border Router's IPv4 address. NAT64 is stateful and only acts on valid, expected responses to a previously made outbound communication request. This limits direct unsolicited inbound connections from any external network over IPv4.
- **Global IPv6**: If a Thread device configures a globally routable IPv6 address (instead of using NAT64), it can become directly accessible over the internet like any other IPv6-enabled device. In practice, this access is still limited by the firewall rules and filtering applied by the ISP, the home/site IPv6 router, and the Thread Border Router. Also, the sheer size of IPv6 address space makes it infeasible for an attacker to "scan" for targets as can be done with IPv4. Still, it is important that firewalls are properly configured, to prevent unauthorized access. The IETF RFC 6092

and RFC 7084 documents provide more details about the expected security functions of a home router.

- **Inbound IPv6 Packets from the Local Link:** for IoT applications that require bi-directional local communication between a Thread device and a non-Thread IPv6 device in the same network, it is important that the Thread Border Router allows unsolicited inbound IPv6 packets. For this reason a Border Router is by default not configured to block such packets. The Border Router will by default block packets that are malformed or that would otherwise exceed the Thread radio's maximum transmission capacity.
- **Thread Device Security:** Regardless of the IPv6 addresses used or local firewall configurations, a Thread device itself should be designed with security in mind. This includes secure software design, procedures to keep firmware up-to-date and proper checking of any inbound communication requests.

3.2.4 Benefits for Consumers and Developers

Thread's enhanced internet connectivity brings several advantages:

- **Simplified Cloud Integration:** Thread devices can now seamlessly connect directly to cloud services, enabling remote control, monitoring, and over-the-air firmware updates.
- **Broader Application Scope:** Thread devices are now enabled to seamlessly interact with a wider array of internet services and protocols. This opens up possibilities for diverse applications beyond direct cloud integration, such as local network interactions, compatibility with legacy systems, and integration with emerging IoT technologies.
- **Future-Proofing:** By supporting IPv6 natively, Thread is aligned with the long-term evolution of the internet, ensuring that internet connectivity remains as IPv4 is gradually phased out.

3.2.5 Use Cases: Where Enhanced Internet Connectivity Delivers Value

The features described in Section 3.2, particularly those enabling IPv6 and IPv4 connectivity, unlock several compelling use cases for Thread networks:

- **Smart Home Integration:** Thread devices can now seamlessly interact with a wider range of smart home ecosystems and controllers, including those that use IPv6 or IPv4, and those that are located in the cloud. This allows for more flexible and comprehensive home automation scenarios.
- **Cloud Connectivity:** Direct internet connectivity enables Thread devices to easily connect to cloud services. This facilitates remote monitoring, control, and firmware updates, enhancing the user experience and enabling remote device management capabilities. Device vendors can now directly service and support their deployed fleet of devices, regardless of which application ecosystem each device is integrated with.
- **Legacy Network Compatibility:** By supporting IPv4, Thread networks can offer internet connectivity even in older home networks, or infrastructure networks. Also, connectivity to internet hosts and services that have not yet transitioned to IPv6 is enabled. This ensures that Thread can be deployed in environments with a mix of new and legacy infrastructure.
- **Enterprise and Industrial Applications:** Thread's enhanced IP connectivity makes it suitable for enterprise and industrial use cases where devices need to communicate with both local and cloud-based systems. This opens possibilities for asset tracking, monitoring, and control in various industries.
- **Service Discovery:** Thread devices can now leverage standard DNS mechanisms to discover and interact with services on the local network or the internet. This simplifies app development that relies on service discovery.

3.3 Enhanced Network Diagnostics

While the standard Thread diagnostic functions provide a valuable baseline for network analysis, more comprehensive tools are often needed to effectively pinpoint and resolve issues in Thread networks as they grow. Enhanced network diagnostics supported in Thread 1.4 are built upon the existing Thread diagnostics framework yet can provide a deeper understanding of network operations and significantly improve network troubleshooting capabilities.

The new diagnostics in Thread 1.4 include more detailed information on network topology, revealing the wireless links between devices and their link properties. Additionally, Enhanced Network Diagnostics offers insights into data packet flows, quantifying and categorizing the types of packets sent and received by each Thread device. Device-specific information, such as device vendor, model, software version, and Thread device type is also readily available. This comprehensive visibility into all network operations enhances network management and streamlines (automated) troubleshooting processes.

Here is a list of the Enhanced Network Diagnostics that have been added in Thread 1.4:

- EUI-64
- Thread Protocol Version and Thread Stack Version
- Vendor Name, Vendor Model, and Vendor Software Version
- Detailed Child information – for Thread Mesh Extenders only
- Child IPv6 Address List – for Thread Mesh Extenders only
- Mesh Extender (Router) Neighbors – for Thread Mesh Extenders only
- Mesh Link Establishment (MLE) Protocol Counters
- Vendor App URL
- A method ("Answer TLV") to split a particularly large diagnostic data set over multiple UDP messages

The entity collecting Thread diagnostic information may be directly connected itself to a Thread network, for example situated on a Thread 1.4 Border Router. The collected information can then be retrieved via an authenticated, secure connection from an app or back-end to this Border Router. The entity could also be situated outside the Thread Network, collecting diagnostic information in the role of a Thread Commissioner - via a secured IPv6 link to a Thread Border Router.

Benefits for Consumers, Product Companies and Professional Installers

Enhanced network diagnostics offer significant benefits for consumers, product companies, app developers, and professional installers.

Consumers: Diagnostics enable more informative tools that provide insights into their Thread network, similar to the well-received Thread topology view in the Eve app. This empowers users to better understand and manage their smart home ecosystem.

Product Companies: These can leverage enhanced diagnostics to create more powerful apps for configuring and managing their IoT products. For instance, apps could incorporate automated network troubleshooting and configuration guidance, reducing the need for customer service calls and improving user satisfaction.

Professional Installers: Enhanced diagnostics offer improved visibility into Thread networks, streamlining device testing and network troubleshooting. This includes the ability to track mesh network structure changes and monitor Mesh Extender states. Logging packet delivery and loss metrics can help identify areas needing additional Thread Mesh Extenders or pinpoint sources of RF interference. This data-driven approach enables faster issue identification and resolution, benefiting both developers during testing and operators/administrators managing large-scale network deployments. The collected information can be presented to end-users or to administrators, promoting transparency and informed decision-making.

3.4 Thread-over-Infrastructure

Thread 1.4 introduces a significant enhancement known as Thread over Infrastructure Links. This feature enables Thread 1.4 Border Routers to leverage existing IP-based network technologies like Wi-Fi and Ethernet in addition to their low-power IEEE 802.15.4 radio. By doing so, Thread mesh networks can achieve greater coverage, higher throughput, and improved reliability, all while maintaining compatibility with existing Thread 1.1, 1.2, and 1.3 devices that use only their IEEE 802.15.4 radio.

3.4.1 Key Benefits of Thread over Infrastructure Links

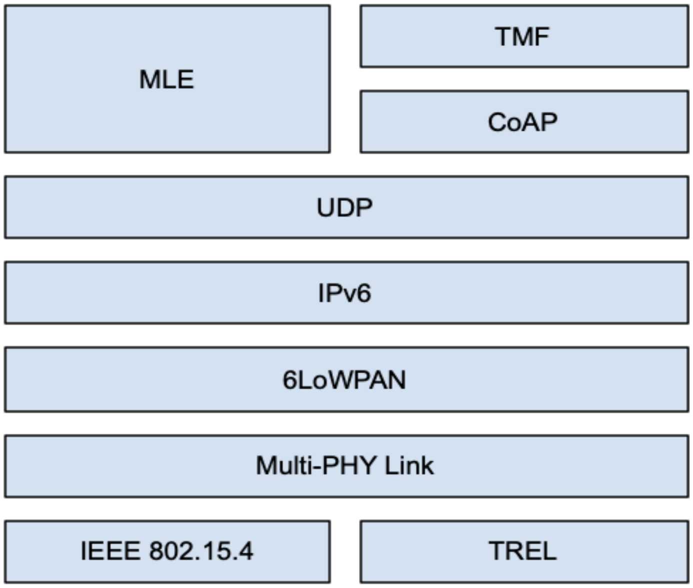
Reduced Network Partitioning: Thread networks can in some cases become partitioned, leading to communication breakdowns between devices in different partitions that were communicating via mesh links. Thread over Infrastructure Links helps mitigate this issue by creating a more interconnected network topology by re-using existing IPv6-capable links. This is particularly valuable in deployments with few Mesh Extender capable Thread devices and a lot of Thread end devices.

Optimized Data Routing: By utilizing the additional IP-based infrastructure links, Thread can route packets more efficiently, on average reducing the number of hops a packet needs to traverse in the mesh network. This leads to lower latency for data and also to less congestion on the IEEE 802.15.4 radio channel. Furthermore, it preserves energy on battery-powered Thread Mesh Extenders.

Enhanced User Experience: Coping with multiple Thread partitions can be complex and counter-intuitive for users. Thread over Infrastructure Links simplifies network diagnosis and management by minimizing the occurrence of partitioning, thus providing a more unified network experience.

3.4.2 How It Works

At the heart of Thread over Infrastructure Links are two primary components: the Multi-PHY link layer and the Thread Radio Encapsulation Link (TREL) protocol. These components work in concert to seamlessly tunnel Thread mesh data traffic over IP-based networks while preserving compatibility with existing Thread protocols. See the figure below for an overview of how these components integrate into the overall Thread protocol stack on a Thread 1.4 Border Router.



Multi-PHY Link Layer

The multi-PHY link layer serves as an intelligent intermediary, determining the optimal physical layer (PHY) for transmitting Thread messages. It considers factors like signal strength, data rate, and power efficiency to make informed routing decisions. To ensure broad reachability, broadcast messages essential for device discovery are transmitted over all available PHYs (e.g., both Wi-Fi and IEEE 802.15.4). PHYs other than IEEE 802.15.4 are accessed via the TREL protocol as explained below.

For unicast messages, the layer employs a preference system, dynamically assigning values to each PHY based on its communication history with neighboring devices. The PHY with the highest preference is generally chosen, with additional factors like power profile considered in the case of ties.

Thread Radio Encapsulation Link (TREL) Protocol

TREL provides the mechanism for encapsulating IEEE 802.15.4 MAC frames within UDP/IPv6 packets. This encapsulation allows Thread messages to traverse IP-based networks while remaining compatible with existing Thread devices, once the Thread messages are again de-encapsulated from the UDP/IPv6 packet. Key aspects of TREL include:

- **TREL Interface:** TREL operates over standard IPv6 network interfaces, such as Wi-Fi or Ethernet.
- **Message Formats:** TREL defines specific message formats for different packet types (unicast, broadcast, acknowledgment), incorporating fields for versioning, packet type, channel information, addresses, and the encapsulated IEEE 802.15.4 MAC frame.
- **Discovery and Message Transmission:** TREL devices use DNS-SD to discover and advertise their presence on the infrastructure network link. Upon discovery, they establish communication channels and exchange TREL UDP/IPv6 packets containing encapsulated Thread messages.
- **Maximum Transmission Unit (MTU) Determination:** TREL can dynamically adjust its MTU based on the destination and the presence of 6LoWPAN mesh headers, optimizing bandwidth utilization.
- **Acknowledgments and Retransmissions:** TREL employs a lightweight acknowledgment mechanism using TREL Ack packets. It leverages the reliability features of the underlying IPv6 network, avoiding the overhead of IEEE 802.15.4 acknowledgments.

- **Message Security:** TREL inherits the security features of IEEE 802.15.4, including MAC layer encryption and frame counter checks. It also derives a separate MAC key for TREL communication, ensuring end-to-end security.

3.4.3 Use Cases: Where Thread-over-Infrastructure Delivers Value

Thread over Infrastructure Links opens doors to a variety of scenarios where its benefits can be fully realized. Here are some compelling use cases:

- **Large Home and Building Automation:** In expansive homes or commercial buildings, relying solely on IEEE 802.15.4 might lead to coverage gaps and network partitioning. Thread over Infrastructure Links leverages existing Wi-Fi or Ethernet networks, accessible via Thread 1.4 Border Routers, to extend Thread's reach, ensuring seamless communication with Thread devices across the entire premises.
- **Outdoor IoT Deployments:** In outdoor environments with sparse Thread Mesh Extender density, mesh network partitioning can be a concern. Thread over Infrastructure Links can bridge these gaps by utilizing long-range Wi-Fi or other available IPv6-based connections, ensuring reliable communication for applications like smart agriculture or environmental monitoring.

These are just a few examples of how Thread over Infrastructure Links can be applied to enhance connectivity, reliability, and performance in diverse IoT deployments. As the Thread ecosystem continues to evolve, we can expect to see even more innovative use cases emerge, further solidifying Thread's position as a leading low-power and reliable wireless technology for the Internet of Things.

3.5 Secure Commissioning at Scale with TCAT

Thread Commissioning over Authenticated TLS (TCAT) is a new feature that enables fast and secure commissioning of large volumes of Thread devices. Strong security is provided by an authenticated, certificate-based TLS session between the new Thread device and a commissioning tool/device.

TCAT support is optional for Thread devices: it is expected that Thread products designed specifically for commercial (B2B) markets will support this feature, while consumer-only (B2C) Thread products would not necessarily support it.

3.5.1 TCAT Architecture

TCAT is designed to securely transmit data and exchange security materials such as Thread network credentials between devices. This is achieved through a TLS connection established between the TCAT Commissioner (the mobile device initiating the configuration and acting as a TLS client) and the TCAT Device (a Thread device to be commissioned and configured, that acts as a TLS server).

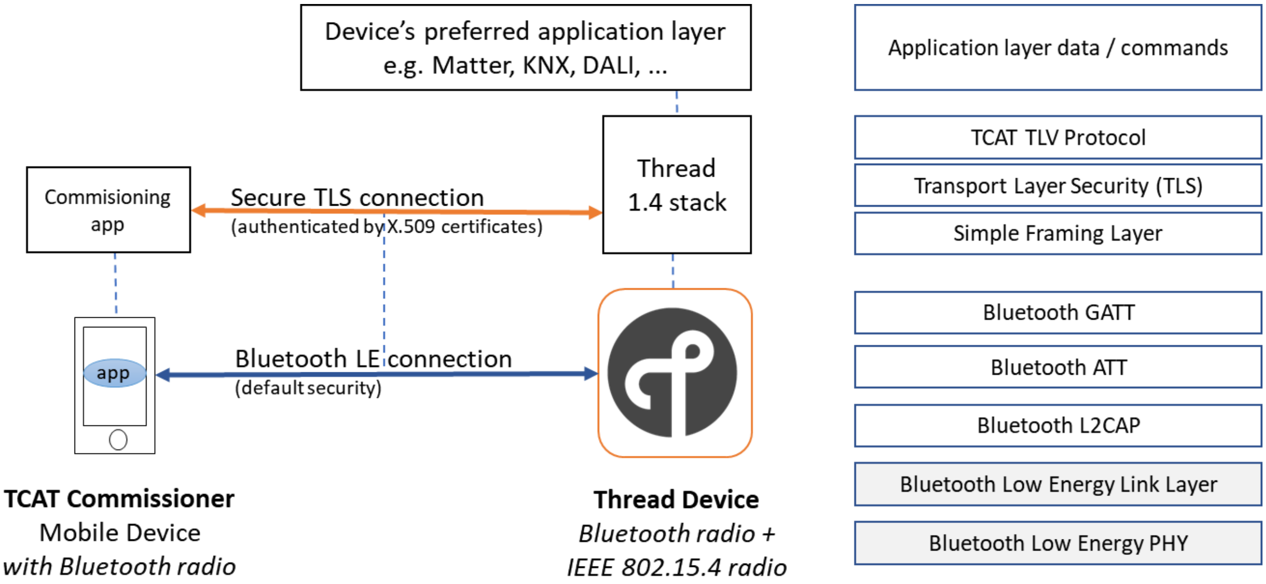
In Thread 1.4, the TCAT TLS session is established over a Bluetooth Low Energy link, so that any of today's mobile devices supporting Bluetooth LE can act as TCAT Commissioner. Yet, TCAT has been designed to support other physical layers (PHYs) in the future as well, such as IEEE 802.15.4, without requiring changes to the core protocol.

The architecture of TCAT over Bluetooth LE is illustrated in the figure below. The top layers contain the generic TCAT architecture, while the bottom layers contain the Bluetooth-specific instantiation.

The core of the architecture is the TLS connection, which carries TCAT TLVs (Type-Length-Value) as its payload. Each TLV functions as a container for specific data items, actions, status codes, or requests for data. The data that may be transferred includes network-related configuration, as well as application-layer data that is passed up by the Thread stack to the device's supported application layer(s).

In scenarios where a non-commissioned TCAT Device also supports standard Thread MeshCoP commissioning, it maintains activity on both its Bluetooth LE and Thread radios – either simultaneously or time-sliced. The Bluetooth LE radio is responsible for transmitting TCAT Bluetooth LE Advertisements, which can be received by a nearby TCAT Commissioner. Concurrently, the Thread radio is used to scan for nearby Thread networks that might accept the device as a new Joiner.

For non-commissioned TCAT Devices that exclusively rely on TCAT-over-Bluetooth LE for commissioning, the Bluetooth LE radio is initially active, while the IEEE 802.15.4 radio is not yet operational. There's also flexibility in the architecture to accommodate TCAT devices that implement additional commissioning methods or support TCAT over multiple physical/link layers.



3.5.2 TLS Connection and Mutual Authentication

When the TLS connection is established, both parties mutually authenticate based on their respective X.509 certificates. A TCAT device can support either TLS version 1.2 or TLS version 1.3, while a TCAT Commissioner must support both TLS versions 1.2 and 1.3.

The TLS client certificate is an X.509 v3 compliant certificate that identifies the TCAT Commissioner. This certificate is part of a chain that is signed by a root Certificate Authority (CA). The manufacturer of the TCAT Device controls this CA and the certificates issued by it. This manufacturer also determines which commercial business partners are entitled to receive TCAT Commissioner certificates, as explained in Section 3.5.4. Details of such decisions are not in the scope of Thread Group but can be defined between the involved parties or defined by an ecosystem standard.

The TLS server certificate is an X.509 v3 compliant certificate that identifies the TCAT Device. This certificate is part of a chain signed by a root Certificate Authority (CA), which is typically the same root CA that is used by the TCAT client certificate. The TLS server certificate may be an Initial Device Identifier

(IDevID) certificate as defined by the IEEE 802.1AR standard, but other certificate types are equally possible. This certificate must be provisioned immutably into the TCAT device at manufacturing time, along with the associated private key.

3.5.3 TCAT Procedures and Options

TCAT defines a series of interactions between a TCAT Commissioner and a TCAT Device, facilitated by the exchange of TCAT TLVs over the secure TLS session. These procedures cater to various use cases, ranging from commissioning new Thread devices to retrieving lost Thread network credentials and diagnosing Thread network issues for Thread devices that were already commissioned. TCAT procedures that a user operating a TCAT Commissioner is able to use include the following:

1. **Commissioner Authorization:** The TCAT Device meticulously validates the TCAT Commissioner's authorization to execute specific command classes. The basic authorization is encoded within the TCAT Commissioner's certificate and is essential for maintaining security and preventing unauthorized access. There are various operations for the TCAT Commissioner to increase its level of authorization, by providing proof-of-possession of particular secrets (e.g. an install code, PSKc, or PSKd) to the TCAT Device.
2. **New Device Commissioning:** This procedure is for commissioning a new Thread device or re-commissioning an existing one with fresh credentials and/or application-layer configuration. It includes setting a Thread Active Operational Dataset, which contains essential information for the device's operation within the Thread network.
3. **Start/Stop Thread Interface on Thread Device:** This procedure activates the Thread interface on the TCAT Device, enabling it to participate actively in the Thread network while the TCAT Commissioner stays connected to verify operation. Deactivation is also supported.
4. **Decommissioning:** This procedure focuses on decommissioning a Thread device. The decommissioning process removes the TCAT device from its Thread network and deletes its credentials from internal storage. After decommissioning, a TCAT device can be commissioned again with a new configuration.
5. **Application Layer Communication and Configuration:** This procedure allows for interaction with a vendor-specific or ecosystem-specific

application layer running on the TCAT Device. It includes selecting the desired application layer protocol (which can be any protocol registered by name at IANA) and sending and receiving application-specific data.

6. **Thread Credentials Extraction:** If the operator of the TCAT Commissioner has obtained a certificate with the "extraction" capability from the TCAT Device's vendor, this procedure enables the extraction of the Thread credentials from a commissioned TCAT Device. This authorization is only granted by the vendor in exceptional circumstances to a well-known business partner of the vendor – it may be needed in particular cases, for example, a commercial building installation where the Thread credentials of an existing Thread network have been lost due to human error or change in business situation.
7. **Collecting Thread Diagnostic Information:** This procedure facilitates the retrieval of diagnostic information from a TCAT Device. This information is valuable for troubleshooting network problems and identifying potential issues with the device's operation.

These TCAT procedures, along with the associated security mechanisms, provide a robust and versatile framework for the secure and efficient management of large volumes of Thread devices in the commercial buildings space.

3.5.4 Use Cases: Where TCAT Secure Commissioning Delivers Value

The main use case for TCAT is commissioning factory-new Thread devices into a Thread Network. This is particularly useful in scenarios like a commercial building where many new Thread devices are being installed, and there is no operational IT infrastructure yet in place.

In this scenario, a commissioner can use a handheld mobile tool/device to scan for nearby TCAT devices that need to be configured. The tool identifies the non-commissioned devices and configures them with the necessary Thread network credentials, network configuration, and application-level information required for their operation as planned or decided on the spot.

The handheld tool used by the commissioner contains information that authenticates the commissioner (using the TCAT Commissioner certificate and private key) and authorizes them to perform the necessary commissioning operations on the TCAT devices. This authorization information is obtained either directly through business contact with the device manufacturers or indirectly through business contact with the resellers of these devices.

The benefit of TCAT in this use case is that it eliminates the need for direct physical contact with each of the devices being commissioned, which may already be installed in inaccessible locations like inside walls or ceilings. This streamlines the commissioning process and makes it more efficient, especially in large-scale deployments.

A similar scenario where TCAT commissioning is valuable is in a building with isolated Thread networks that are operating without a backbone IP connection to an IT infrastructure. The commissioner can then use TCAT to assign new Thread devices to their respective networks efficiently and securely. If the backbone IP connections are enabled later on, the Thread devices can then automatically start making use of this connection without the need for the commissioner to revisit the devices physically.

3.5.5 Benefits for Professional Installers and Product Companies

Benefits for Professional Installers

TCAT commissioning offers significant advantages for professional installers working in commercial building environments, where large-scale Thread deployments and interactions with diverse building systems are common. These benefits, which directly address common challenges faced during the installation and configuration of Thread IoT devices, include:

- **Effortless and Wireless Onboarding:** TCAT enables installers to easily onboard and configure Thread IoT devices wirelessly, even when they are installed in locations that are difficult to access physically. This eliminates the need for time-consuming and potentially disruptive tasks, such as accessing devices installed within ceilings or embedded in walls.
- **Streamlined Commissioning:** With TCAT, authorized installers can commission devices seamlessly without the need to scan install codes on individual products. This simplifies the process and reduces the chance of errors, leading to faster and more efficient installations.
- **Inherent Security:** TCAT is designed with security in mind. The use of industry-standard TLS authentication and encryption ensures that the configuration process remains secure, even when performed wirelessly.

This protects sensitive network credentials and data from unauthorized access.

- **Leveraging Existing Devices:** TCAT leverages the built-in Bluetooth LE radio present in today's off-the-shelf mobile devices, such as smartphones and tablets. This means installers can use familiar tools for commissioning, without the need for specialized equipment.
- **Direct Diagnostics:** In cases where a device is not reachable through the Thread network, TCAT allows for the retrieval of diagnostic information through a direct link to a commissioner or maintenance application on a mobile device. This facilitates troubleshooting and issue resolution, even in challenging network conditions.

Benefits for Product Companies

TCAT also provides valuable benefits for product companies, enabling them to enhance their commissioning and service applications:

- **Versatile Applications:** TCAT can be integrated into various aspects of device management, including network onboarding, device and network diagnostics, application commissioning, and device service and maintenance. This versatility allows companies to streamline their operations and improve overall efficiency.
- **Controlled Partner Access:** Device manufacturing companies can carefully manage which partner companies are authorized to perform commissioning operations. TCAT provides granular control over the scope of these operations, allowing companies to define specific permissions for different partners, and even differentiate on a per-project basis or based on situation-specific needs.
- **Targeted Permissions:** Companies can grant permission for TCAT commissioning on a granular level, limiting access to specific product lines or even specific installation sites. This ensures that commissioning activities align with business requirements and security policies.
- **Secure Service Functions:** Trusted service professionals can be given access to protected service functions through TCAT, allowing for secure remote diagnostics, maintenance, and troubleshooting by these professionals.
- **Lifecycle Management:** TCAT supports de-commissioning and owner change processes, ensuring that devices can be securely removed from

networks or transferred to new owners while maintaining confidentiality and integrity.

In summary, TCAT commissioning offers a range of benefits that improve efficiency, security, and flexibility for both professional installers and product companies in the Thread IoT ecosystem.

3.6 Enhancements to Mesh Robustness and Scalability

Thread 1.4 includes several maintenance updates designed to improve mesh network robustness, performance, and overall efficiency. Some of the key enhancements include:

Border Router as Mesh Extender: Border Routers are now allowed to always operate as Thread Mesh Extender, independent of the usual automated activation rules for Mesh Extenders. This enhances routing efficiency and reduces latency for external IP communication, contributing to both robustness and performance.

Expanded Address Cache for Border Routers: By mandating a larger address cache for Border Routers, Thread 1.4 reduces the need for multicast mesh address queries. This optimization streamlines routing operations and improves overall network performance.

Network Data Management: Thread 1.4 introduces mechanisms to reduce the size of Thread Network Data, the mesh configuration data that governs routing. Size reduction prevents routing issues caused by the network data reaching its size limit when the number of Border Routers increases. These changes ensure symmetrical routing and improve network performance and robustness, especially in high-traffic scenarios in networks with multiple Border Routers.

MAC Layer Prioritization: Prioritizing Thread's 802.15.4 Coordinated Sampled Listening (CSL) transmissions over indirect transmissions in the MAC layer helps reduce overall latency and ensures time-sensitive (CSL) communications are handled efficiently.

These and other updates collectively enhance Thread's ability to operate reliably in diverse network and RF conditions, scale effectively with increasing network size, and maintain high performance levels even under heavy loads.

4. Summary

Thread 1.4 represents a significant evolution in the Thread wireless mesh network technology, introducing a suite of enhancements designed to address the growing complexity and demands of the IoT landscape. These enhancements span critical areas such as device commissioning, security, internet connectivity, network diagnostics, and infrastructure network integration.

By streamlining the onboarding of new Thread devices, improving network resilience, and expanding compatibility with existing and future technologies, Thread 1.4 empowers developers, product manufacturers, and professional installers to create more innovative, reliable, and user-friendly IoT solutions. For consumers, this translates to a more seamless and secure connected experience, unlocking the full potential of the smart home and beyond.

Thread 1.4 is not just an incremental update; it is a transformative leap that solidifies Thread's position as a leading technology for the Internet of Things. With its comprehensive enhancements and forward-looking approach, Thread 1.4 is poised to accelerate the adoption of Thread across a wide range of applications, from smart homes and commercial buildings to industrial automation and beyond.

The Thread Group invites all stakeholders to explore the possibilities offered by Thread 1.4 and join us in shaping the future of connected devices.