Application Report

SWRA654A – December 2019 – Revised April 2020

SimpleLink WLAN Access Point

Elin Wollert

ABSTRACT

This application report presents the benefits and the possibilities in extending an existing WLAN Access Point to use an additional IoT protocol such as Zigbee®, Thread, or *Bluetooth*® Low Energy. It will present the different technologies and what they can be used for. It will also present the proposed hardware solutions and example applications.

Contents

List of Figures

Trademarks

1 Introduction

Why settle for a Wi-Fi™-only WLAN Access Point when you can create a smart access point? Adding Bluetooth Low Energy, Zigbee, or Thread to the access point will enable the use of localization, provisioning, and IoT for smart building. The TI coexistence feature will allow multiple protocols to operate at the same frequency, coexist, and not transfer data simultaneously. When two RF protocols are used in the same MCU, the Dynamic Multi-protocol Manager (DMM) is used to allow the protocols to operate concurrently. This will innovate enterprise, retail, and medical environments.

2 WLAN Access Point Using SimpleLink Platform

It does not have to be complicated to add additional functionalities in an existing WLAN Access Point. Adding extra hardware from the SimpleLink family provides the opportunity to add Zigbee, Thread, or Bluetooth Low Energy to an existing WLAN Access Point. Zigbee and Thread add mesh network abilities to expand the network and reach distant nodes within the network. Bluetooth Low Energy will add multiple features, for example, the ability to interact with Smartphones or track beacons. The addition of Bluetooth allows using Angle of Arrival (AoA) to triangulate and find the direction of the incoming Bluetooth packets.

To be able to use the technologies concurrently, the Dynamic Multi-protocol Manager (DMM) and the TI Coexistence feature are used. The DMM handles the scheduling within the chip, making it possible to use multiple protocols in the same CPU. The TI Coexistence feature allows different technologies operating at the same frequency to coexist and not use the antenna simultaneously to decrease the risk of losing packets.

There are two proposed solutions to extend the functionalities of the WLAN Access Point using one or two CC26x2 devices. The CC2652 device, see *CC2652R Product Page*, has support for Zigbee, Thread, and Bluetooth Low Energy and is used in the one-chip solution. The two-chip solution uses the CC2652 device for Zigbee and Thread while either a CC2652 device or a CC2642 device, see *CC2642R Product Page*, is used for Bluetooth.

There is a large variety when it comes to developing applications for either of the solutions. For example, some applications might only require the use of Bluetooth to enable device tracking, using either connection based or connectionless communication. Other applications might require only using a Zigbee or Thread network to fetch data from remote sensors. Examples can be detecting smoke or classifying sound to detect if glass breaks, gunshots, and so forth. The different technologies can also be combined to create a more extensive solution.

The two-chip solution, illustrated in Figure 1, uses dedicated MCUs for each protocol. Both *CC2652R* and *CC2642R* can be used as the Bluetooth device, but only CC2652 can be used for Zigbee or Thread. This solution is recommended for applications that require controlling roles for two protocols, such as Bluetooth Low Energy Central and Zigbee Controller. Both the Central and Controller must listen to incoming packets at all times; they cannot share the use of one antenna. Instead, they must use dedicated devices and antennas in order to function properly.

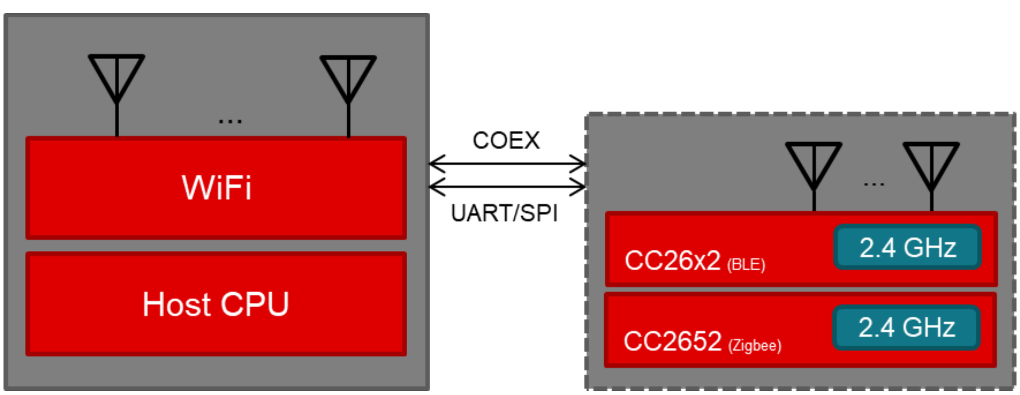

Figure 1. WLAN Solution Using a Two-Chip CC26x2 Solution

The one-chip solution, illustrated in Figure 2, uses a single CC2652R device that supports Zigbee, Thread, and Bluetooth Low Energy. This solution is recommended for applications that only use one protocol, or use roles for two protocols that can share the antenna resource. The DMM handles the prioritization if two protocols are used.

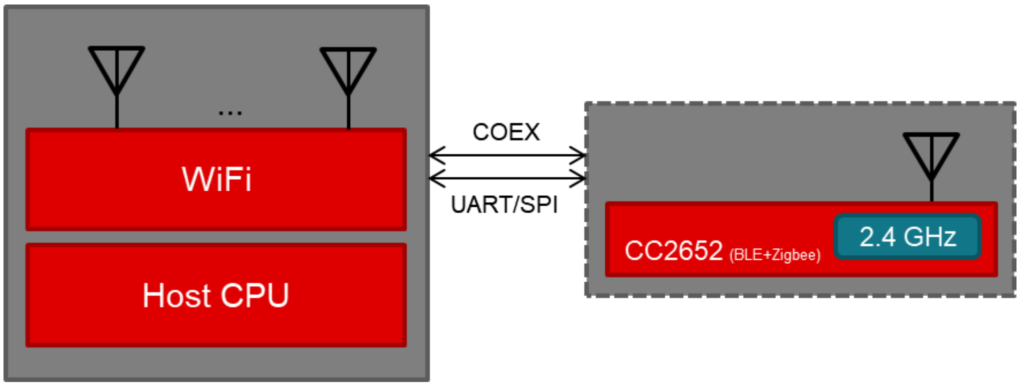

Figure 2. WLAN Solution Using a One-Chip CC2652 Solution

Both solutions have the advantage of using the same Software Development Kit (SDK) for Zigbee, Thread, Bluetooth Low Energy and the DMM. They also use the same tools such as Code Composer Studio™ (CCS). The firmware images of the devices can be updated by using a serial bootloader or over the air, Over the Air Download (OAD) (Bluetooth low energy, Zigbee, Thread) or Over The Air update (OTA) (Wi-Fi).

3 Technologies

To create the previously mentioned solution, some concepts have to be known. These are presented in this section.

3.1 Zigbee

Zigbee is a wireless communication system operating at 2.4 GHz, which targets battery-powered applications. There are three logical device types for Zigbee networks, Coordinator, Router, and End Device. The role of the Coordinator is to start a network and manage the keys; it also handles all devices entering and leaving the network. Besides that, it behaves like a Router device. The Router is active at all times to ensure that other devices can join the network. Routers also have child devices, End Devices, which they assist with the communication. The End Devices are usually asleep, except when communicating with its parent Router, because they have no responsibility for maintaining the network infrastructure. Figure 3 shows a typical Zigbee network with one Coordinator in black, multiple Routers in red, and multiple End Devices in white.

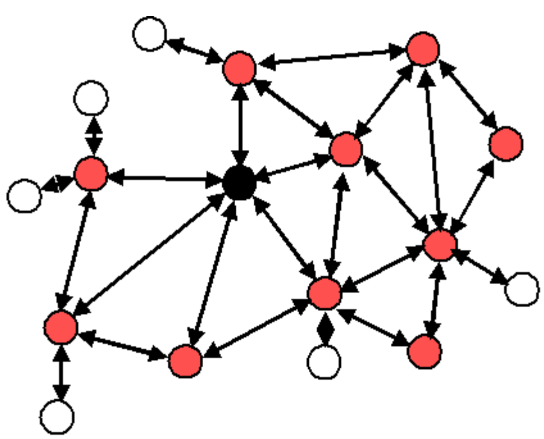

Figure 3. A Typical Zigbee Network

Zigbee networks use a mesh structure to transmit data over long distances. The data passes through multiple intermediary devices to reach the final destination in the remote device. The remote devices can be different kinds of sensors, for example, temperature, smoke detector, audio detection, and so forth. They can also be used for smart home applications such as ventilation.

The TI implementation of Zigbee is called the Z-stack and is a part of the CC13x2/CC26x2 SDK. For more information, see the *Z-Stack User's Guide*.

3.2 Thread

Thread is a wireless protocol that operates at 2.4 GHz. There are four different devices in a Thread network: Leader, Router, Border Router, and End Device. All devices start as End Devices but are promoted as the network requires. There is only one Leader in the network that handles the decisions in the network, for example, Router upgrading. The Leader is elected by being the first Router in the network. The Routers handle the communication between devices in the network by transmitting, receiving, and forwarding data around the network. Routers can also act as parents for End Devices. When the Router is doing network activities, such as responding to network queries, they act as proxies for their children. Border Routers are Routers that act as a bridge between the Thread Network an adjacent network, like Wi-Fi or Ethernet. The End Device only connects to one Router, which handles all communication. If the connection between the End Device and its parent Router is lost, the End Device will search for a new Router.

For another device to join the network, they have to know the channel and master-key. This information is shared when the attaching device is authenticated before securely brought onto the network, also known as the commissioning process. There are two different categories when it comes to commissioning: external and native commissioning. For off-mesh nodes, such as Border Routers, external commissioning is required. Native commissioning is used for the on-mesh nodes.

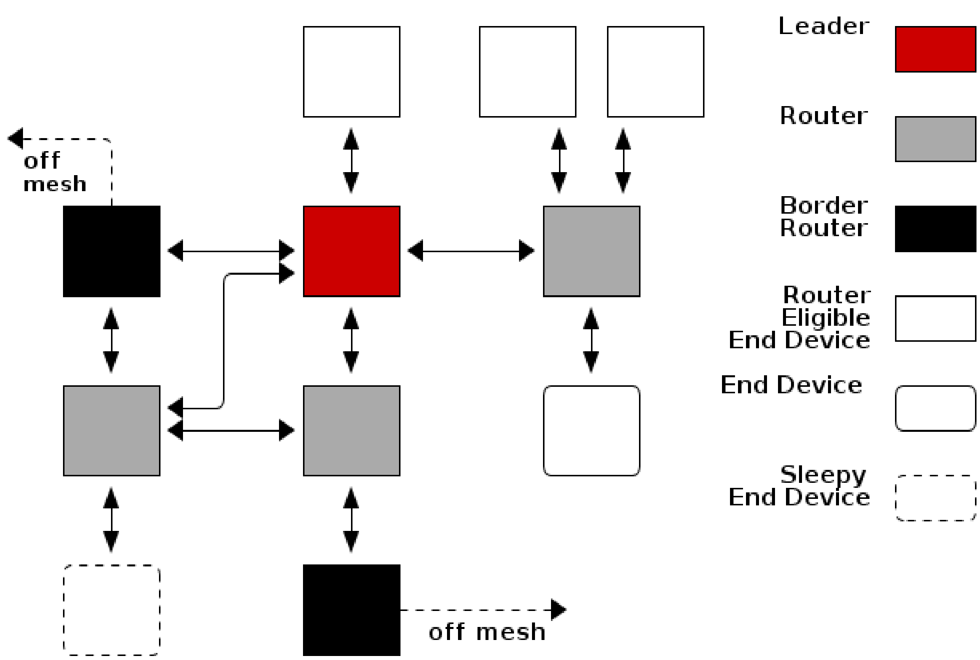

Figure 4. A Typical Thread Network

Thread devices can be either a Full Thread Device (FTD) or a Minimal Thread Device (MTD). An FTD can be any of the above-mentioned roles, while an MTD can only be an End Device. An FTD can be configured at runtime to be an MTD, but the reversed is not permitted since the MTD does not contain all necessary sections to act as an FTD.

Thread networks use a mesh structure to transmit data over long distances. The data passes through multiple intermediary devices to reach the final destination in the remote device. The remote devices can be different kinds of sensors, for example, temperature, smoke detector, classification of sound, and so forth. The data can also be transferred off mesh by using the Border Routers to interface other networks. This way, the data from the sensors within the Thread Network will be available to send to the cloud, for example.

The TI-OpenThread stack is an open-source implementation of the Thread specification, designed and developed for single-threaded environments. The TI-OpenThread stack handles a queue of pending tasks to run. For more information, see *TI-OpenThread User's Guide*.

3.3 Bluetooth

Bluetooth is a wireless communications system operating at 2.4 GHz. There are two wireless Bluetooth technologies within the same specification: Bluetooth Classic and Bluetooth Low Energy. Although both technologies partly share the same name, they are not compatible, which means that a Classic device cannot directly communicate with a Low Energy device. This document only covers Bluetooth Low Energy. For more information about Bluetooth Classic, see the *Bluetooth Core Specification*.

A device operates in one of the four roles defined by the Bluetooth Core Specification: Broadcaster, Observer, Peripheral, and Central. The Broadcaster and Observer are non-connectable, which means that they cannot connect with any device. The Broadcaster broadcasts data without knowing if another device is receiving the data. The Observer listens to the data transmitted from Broadcasters without creating a link between the devices. The Peripheral and Central are connectable, which means that they will not share data before establishing a connection first. The Central can connect to multiple Peripherals to receive data.

The Bluetooth Low Energy protocol stack consist of the host and the controller, see Figure 5. The lower layers are located in the controller, and the higher layers are in the host. The application sits on top of the host. For more information about the different layers and the Bluetooth Low Energy stack, see the *BLE5-Stack User's Guide*.

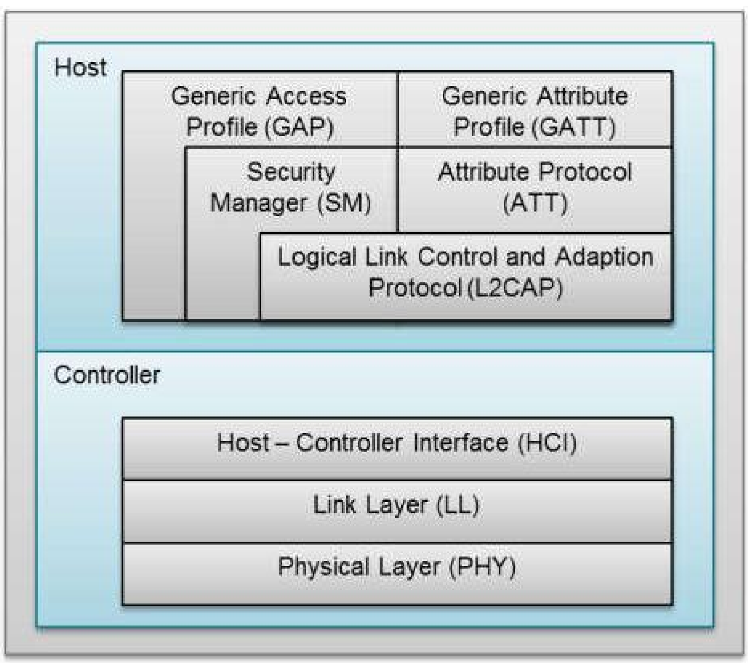

Figure 5. Bluetooth Low Energy Protocol Stack

Angle of arrival (AoA) is a direction-finding technique specified in the Bluetooth Core Specification that finds the angle from the receiver to the transmitter to create a basis for triangulation. The technique uses a well-defined antenna array to notice small changes in the path length between each antenna. The AoA packets must contain a section of a constant tone where there are no phase shifts that are caused by modulation. In a 2D scenario, the possible locations seen by one device is a straight line. As seen in Figure 6, a single intersection point can be determined using only two devices.

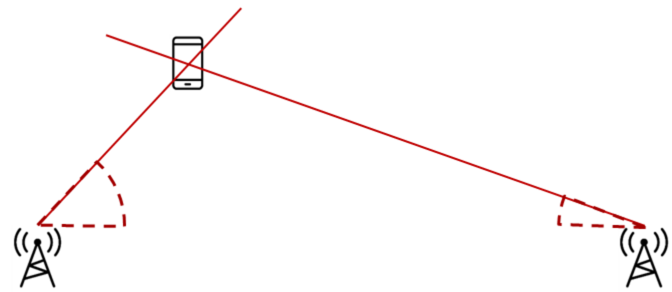

Figure 6. Triangulation Using Connected AoA

AoA is a part of the Real-Time Localization System (RTLS) Toolbox. Because of the inherent flexibility of the CC26x2 RF Core, RTLS features and security can be added without a significant impact on cost and energy consumption with no increase in peak power. Currently, AoA is only supported for connected devices: Peripheral and Central. Future implementations will make it possible to use non-connectable devices: Broadcaster and Observer. For more information about the RTLS Toolbox and AoA, see the *BLE5-Stack User's Guide*.

Because the Low Energy aspect of Bluetooth Low Energy is achieved by letting the device go to sleep when it is not transmitting or receiving packets, it is ideal to use for applications sending or receiving data infrequently, for example, sensors. Beacons can be used as sensors, for example, asset or personnel tracking, making sure that valuable assets are not lost or misplaced or that there is not overlap or deficiency of personnel. For more information, see *Bluetooth low energy Beacons*. Additional sensors can also be used to track ambient data such as temperature, detecting smoke, and so forth. The Sensor Controller, an ultra-low power 16-bit CPU core, should be used to achieve the lowest possible power consumption for sensor applications. It runs independently of the rest of the system, allowing the main MCU to sleep while the sensor controller is reading and processing sensor data. For more information about the sensor controller, see *Meet the SimpleLink Sensor Controller* and *CC26xx, CC13xx Sensor Controller Studio*.

The majority of Smartphones has support for Bluetooth Low Energy, making it suitable for applications that require user interaction or the ability to detect nearby devices. Example applications are detecting nearby Smartphones to send information to, identifying a specific device, and so forth.

3.4 Coexistence

When two or more devices in close proximity are using different radio technologies operating at the same frequency band, some mechanism must be used to make sure that the performance of the devices is maintained. This feature is called coexistence and can be used regardless of whether the devices are using a shared antenna or dedicated antennas. This section describes how coexistence works for different protocols.

3.4.1 Wi-Fi

Wi-Fi typically operates at 2.4 GHz and 5 GHz. Because Zigbee, Thread, and Bluetooth do not operate above 2.4 GHz, coexistence only affects Wi-Fi devices operating in the 2.4 GHz band. Coexistence schemes are often designed to give other protocols priority over Wi-Fi due to the fact that Wi-Fi is inherently more tolerant to time-domain disturbances. Delaying a transmission or missing an incoming reception over a Wi-Fi interface will usually not have a significant impact on the user experience.

For information about coexistence on SimpleLink Wi-Fi, see *CC3135, CC3235x SimpleLink Wi-Fi Internet-on-a chip Solution –BLE Coexistence*.

3.4.2 Bluetooth

The Bluetooth implementation of coexistence can either be three- or one-wire. The feature works for both separate or shared antennas. The three-wire coexistence consists of; REQUEST, PRIORITY, and GRANT signals, and the one-wire coexistence consist of either the REQUEST or GRANT signal. Figure 7 displays the three-wire coexistence between a SimpleLink CC13x2/CC26x2 device and the Wi-Fi chip.

Figure 7. Block Diagram of Three-Wire Coexistence

REQUEST is sent from the Bluetooth device to the Wi-Fi device to request permission to use the antenna. The Wi-Fi device responds with the GRANT signal to either grant or deny the request. PRIORITY is a time-shared output signal from the Bluetooth device indicating both priority and the type of RF activity. REQUEST and PRIORITY are active high, and GRANT is active low. Figure 8 shows an illustration of the typical behavior when the Bluetooth device wants to transmit or receive data. The illustration only shows an example of the signals, not physical measurements.

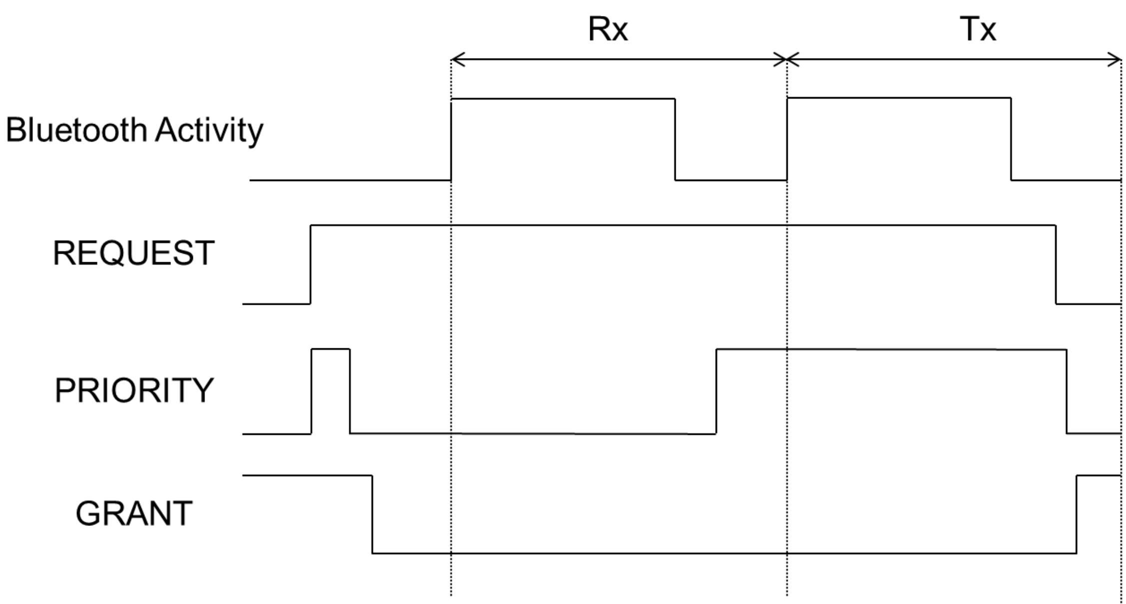

Figure 8. Timing Diagram of the Three-Wire Coexistence Signals

Figure 9 illustrates a block diagram of one-wire REQUEST coexistence. The one-wire REQUEST coexistence only uses the REQUEST signal to notify the Wi-Fi device that Bluetooth activity is pending. This means that the Bluetooth device does not wait for the Wi-Fi device to indicate whether the right to transmit has been granted to the Bluetooth device. This option may lead to losing data packets since both protocols may send or receive data at the same time. Bluetooth assumes it always has the highest priority because Wi-Fi is less sensitive to time-domain disturbances. The SimpleLink Wi-Fi devices that currently support coexistence are CC3135, see *CC3135 Product Page*, or any of the CC3235 devices.

Figure 9. Block Diagram of One-Wire REQUEST Coexistence

Figure 10 illustrates a block diagram of one-wire GRANT coexistence. The one-wire GRANT coexistence only uses the GRANT signal. This means that when the Wi-Fi device asserts GRANT, the Bluetooth device is allowed to transmit or receive packets. When the GRANT signal is de-asserted, the Bluetooth device has to stop sending or receiving data. Since the Bluetooth device cannot request the use of the antenna, there is a risk of missing essential packets or creating a large queue of outgoing packets from the Bluetooth device.

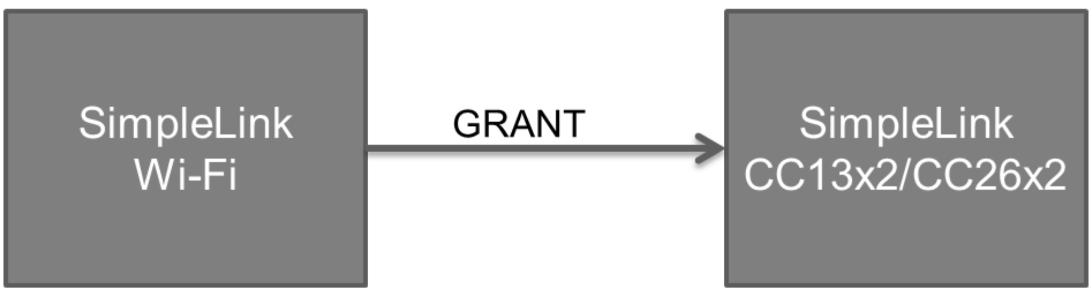

Figure 10. Block Diagram of One-Wire GRANT Coexistence

More information about Bluetooth coexistence is found in the coexistence chapter of the *User's Guide* and the *SimpleLink Academy coexistence* lab.

3.4.3 Zigbee and Thread

Currently, there is no coexistence implementation for Zigbee and Thread.

3.5 Dynamic Multi-protocol Manager (DMM)

The Dynamic Multi-protocol Manager (DMM) is a software module that allows multiple wireless protocols on the same device to operate concurrently on a single radio. It acts as an arbitrer between the stacks and the shared RF core resource. The DMM consist of two components: the Policy Table and the Scheduler. The DMM intercepts the commands from the stacks to create a queue based on the policies from the policy manager. The architectural overview of the DMM is shown in Figure 11.

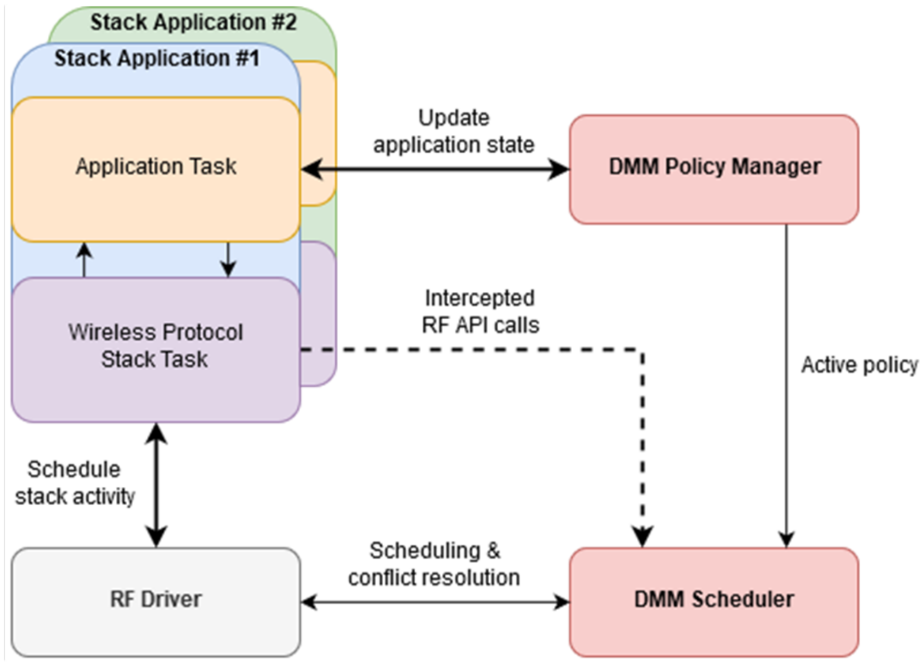

Figure 11. DMM Architectural Overview Block Diagram

The DMM Scheduler keeps track of the RF commands from each stack and the PHY switching events from the RF driver. The Scheduler does not redefine the RF APIs but provides the RF driver hooks for scheduling and conflict resolution. The DMM resolves any conflict based on the current active policy. The current state of the stack application and the scheduling parameters are known to the DMM policy manager to use for a specific combination of application states.

The DMM Policy table is a complete set of all the policies for the application. A DMM policy provides the application state, application weight, applied stack activities, and the pause policy. The policy table contains one or more policies where the last policy is the default one. The policy table is generated by SysConfig.

Learn more about the DMM in the *Dynamic Multi-protocol Manager (DMM) Tech Note* and the *Dynamic Multi-protocol Manager User's Guide*.

4 References

- Bluetooth SIG *Bluetooth Core Specification*
- *BLE5-Stack User's Guide*
- Texas Instruments: *Bluetooth low energy Beacons*
- *Z-Stack User'sGuide*
- *TI-OpenThread User's Guide*
- *CC2652R Product Page*
- *CC2642R Product Page*
- *CC3135 Product Page*
- Texas Instruments: *CC3135, CC3235x SimpleLink Wi-Fi Internet-on-a chip Solution –BLE Coexistence*
- Texas Instruments: *Dynamic Multi-protocol Manager (DMM)*
- *Dynamic Multi-protocol Manager User's Guide*
- Texas Instruments: *Meet the SimpleLink Sensor Controller*
- Texas Instruments: *CC26xx, CC13xx Sensor Controller Studio*

Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Original (December 2019) to A Revision	Page
• Updates were made in Section 3.4.2.	7
• Update was made in Section 3.4.3.	8
• Update was made in Section 4	10