



Amazon Sidewalk Power Optimization Developer Guide

Protocol Stack 1.0, Document Revision A
March 3, 2026

Use of these Amazon Sidewalk specifications (the “Specifications”) is subject to your compliance with the AWS Customer Agreement and the Service Terms (collectively, the “Agreement”), including all disclaimers and limitations as to such use contained therein.

All statements, information, and data contained herein is subject to change without further notice to improve reliability, function, or design. Certain parameters may vary in different applications and performance may vary over time. It is your responsibility to validate that Amazon Sidewalk is suitable for your particular device or application.

No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted by this document.

Amazon Sidewalk is not intended for use in, or in association with, the operation of any hazardous environments or critical systems that may lead to serious bodily injury or death or cause environmental or property damage, and you are solely responsible for all liability that may arise in connection with any such use.

This document is Non-Confidential.

©2024 Amazon Technologies, Inc. Amazon and all related marks are trademarks of Amazon.com, Inc. or its affiliates.

Contents

1	BLE Power Optimization	5
1.1	Introduction	5
1.2	BLE Configuration Options	5
1.2.1	SID_OPTION_BLE_USER_CONFIG	5
1.2.2	SID_OPTION_BLE_CONNECTION_POLICY	5
1.3	Advertisement Power Optimization	6
1.3.1	Advertisement Interval Configuration	6
1.4	Connection Power Management	6
1.4.1	Connection Parameter Configuration	6
1.4.2	Connection Policy Configuration	7
1.5	Inactivity Timeout Configuration	7
1.5.1	Inactivity Timeout Configuration	7
1.6	Configuration Strategy	7
1.6.1	Policy vs Custom Configuration	7
1.7	Platform-Specific Notes: Nordic NCS	8
1.7.1	Impact on Power Optimization	8
1.7.2	Recommended Approach for NCS	8
1.7.3	Legacy nRF5 SDK Compatibility	8
2	FSK Power Optimization	9
2.1	Introduction	9
2.2	Device Profile Optimization	9
2.2.1	Profile 1 vs Profile 2	9
2.3	Beacon Synchronization Optimization	10
2.3.1	Beacon Interval Configuration	10
2.4	RX Window Optimization	10
2.4.1	RX Duration Configuration	10
2.4.2	RX Window Separation	11
2.5	FSK Transmit Power Control (TPC)	11
2.5.1	FSK TPC Testing Results	12
3	LoRa Power Optimization	13
3.1	Introduction	13
3.2	Device Profile Optimization	13
3.2.1	Profile A vs Profile B vs Profile C	13
3.3	Low Latency Configuration	15
3.3.1	Low Latency Mode	15
3.4	LoRa Transmit Power Control (TPC)	15
3.4.1	LoRa TPC Testing Results	16
4	Change History	19

Glossary

21

Chapter 1

BLE Power Optimization

1.1 Introduction

Bluetooth Low Energy (BLE) in Amazon Sidewalk provides local connectivity and serves as a bridge to cloud services. This chapter focuses on power optimization through proper configuration of `SID_OPTION_BLE_USER_CONFIG` and `SID_OPTION_BLE_CONNECTION_POLICY` options.

1.2 BLE Configuration Options

1.2.1 `SID_OPTION_BLE_USER_CONFIG`

The `SID_OPTION_BLE_USER_CONFIG` option provides full customizable parameter configuration for BLE operations, allowing fine-tuned power optimization:

```
typedef struct sid_ble_user_config {
    sid_ble_cfg_adv_param_t adv_param;
    sid_ble_cfg_conn_param_t conn_param;
    bool is_set; // True: Set, False: Get
    enum sid_ble_user_config_type cfg_type; // Configuration type
    uint32_t inactivity_timeout;
} sid_ble_user_config_t;

enum sid_ble_user_config_type {
    SID_BLE_USER_CFG_ADV,
    SID_BLE_USER_CFG_CONN,
    SID_BLE_USER_CFG_ADV_AND_CONN,
    SID_BLE_USER_CFG_INACTIVITY_TIMEOUT,
};
```

1.2.2 `SID_OPTION_BLE_CONNECTION_POLICY`

The `SID_OPTION_BLE_CONNECTION_POLICY` option provides pre-configured settings (overridable with `SID_OPTION_BLE_USER_CONFIG`) and long-lived connection policies to reduce frequent advertising and connection establishment overhead:

```
typedef struct sid_ble_connection_policy {
    bool is_set;
    enum sid_ble_llc_optimization_policy llc_policy;
    enum sid_ble_conn_policy conn_policy;
} sid_ble_connection_policy_t;
```

```
enum sid_ble_llc_optimization_policy {
    SID_BLE_LLC_POWER_OPTIMIZED,
    SID_BLE_LLC_LATENCY_OPTIMIZED,
};

enum sid_ble_conn_policy {
    SID_BLE_CONN_POLICY_DEFAULT,
    SID_BLE_CONN_POLICY_LONG_LIVED_CONNECTION,
    SID_BLE_CONN_POLICY_OPTIMAL_ADVERTISING,
};
```

1.3 Advertisement Power Optimization

1.3.1 Advertisement Interval Configuration

Advertisement intervals should be configured based on application requirements and power constraints:

```
// Example advertisement configuration - tune for your application
sid_ble_user_config_t adv_config = {
    .is_set = true,
    .cfg_type = SID_BLE_USER_CFG_ADV,
    .adv_param = {
        .fast_interval = 800, // Adjust based on discovery requirements
        .fast_timeout = 3000, // Tune for your application
        .slow_interval = 3200, // Balance power vs discoverability
        .slow_timeout = 0 // Set timeout as needed
    }
};

// Apply configuration
sid_error_t err = sid_option(handle, SID_OPTION_BLE_USER_CONFIG,
    &adv_config, sizeof(adv_config));
```

Power Benefit: Custom advertisement intervals allow balancing discovery time vs power consumption based on specific application needs.

1.4 Connection Power Management

1.4.1 Connection Parameter Configuration

Connection parameters should be configured based on application data rate and power requirements:

```
// Example connection parameters - adjust for your use case
sid_ble_user_config_t conn_config = {
    .is_set = true,
    .cfg_type = SID_BLE_USER_CFG_CONN,
    .conn_param = {
        .min_conn_interval = 400, // Adjust based on data rate needs
        .max_conn_interval = 800, // Longer intervals = lower power
        .slave_latency = 9, // Higher latency = lower power
        .conn_sup_timeout = 6000 // Set appropriate for application
    }
};
```

```
sid_error_t err = sid_option(handle, SID_OPTION_BLE_USER_CONFIG,
                             &conn_config, sizeof(conn_config));
```

1.4.2 Connection Policy Configuration

Select pre-configured policies based on application usage patterns:

```
// Example connection policy - choose based on usage
sid_ble_connection_policy_t policy_config = {
    .is_set = true,
    .llc_policy = SID_BLE_LLC_POWER_OPTIMIZED, // vs SID_BLE_LLC_LATENCY_OPTIMIZED
    .conn_policy = SID_BLE_CONN_POLICY_LONG_LIVED_CONNECTION // Choose based on needs
};
```

```
sid_error_t err = sid_option(handle, SID_OPTION_BLE_CONNECTION_POLICY,
                             &policy_config, sizeof(policy_config));
```

Power Benefit: Long-lived connections avoid frequent advertising and connection establishment, which are power-intensive operations. The device maintains connection longer, reducing overall power consumption.

1.5 Inactivity Timeout Configuration

1.5.1 Inactivity Timeout Configuration

Configure automatic disconnection when BLE has no activity:

```
// Configure inactivity timeout
sid_ble_user_config_t timeout_config = {
    .is_set = true,
    .cfg_type = SID_BLE_USER_CFG_INACTIVITY_TIMEOUT,
    .inactivity_timeout = 30000 // Tune based on usage patterns
};
```

```
sid_error_t err = sid_option(handle, SID_OPTION_BLE_USER_CONFIG,
                             &timeout_config, sizeof(timeout_config));
```

Power Benefit: Automatically breaks BLE connections when inactive, preventing unnecessary power drain from maintaining idle connections.

1.6 Configuration Strategy

1.6.1 Policy vs Custom Configuration

Choose between pre-configured policies and custom parameters:

```
// Option 1: Use pre-configured policy (recommended for most cases)
void use_preconfigured_policy(sid_handle_t handle) {
    sid_ble_connection_policy_t policy = {
        .is_set = true,
        .llc_policy = SID_BLE_LLC_POWER_OPTIMIZED,
        .conn_policy = SID_BLE_CONN_POLICY_LONG_LIVED_CONNECTION
    };
    sid_option(handle, SID_OPTION_BLE_CONNECTION_POLICY, &policy, sizeof(policy));
}
```

```
// Option 2: Override with custom parameters for specific needs
void override_with_custom_config(sid_handle_t handle) {
    // Custom config overrides policy settings
    sid_ble_user_config_t config = {
        .is_set = true,
        .cfg_type = SID_BLE_USER_CFG_ADV_AND_CONN,
        .adv_param = { .fast_interval = 1600, .slow_interval = 8000 },
        .conn_param = { .min_conn_interval = 400, .slave_latency = 9 }
    };
    sid_option(handle, SID_OPTION_BLE_USER_CONFIG, &config, sizeof(config));
}
```

1.7 Platform-Specific Notes: Nordic NCS

Nordic Semiconductor has not yet support for runtime BLE configuration in their nRF Connect SDK (NCS) platform. This has direct implications for BLE power optimization on NCS-based devices.

1.7.1 Impact on Power Optimization

On NCS builds, the runtime advertisement and connection parameter tuning described in section 1.3 and section 1.4 is not available through `SID_OPTION_BLE_USER_CONFIG`. The NCS BLE stack does not support modifying these parameters at runtime.

1.7.2 Recommended Approach for NCS

For NCS-based devices, BLE power optimization must be achieved through compile-time configuration:

- Configure advertisement intervals via Kconfig options or devicetree overlays at build time rather than through `SID_OPTION_BLE_USER_CONFIG` at runtime.
- Configure connection parameters via Kconfig options or devicetree overlays at build time.
- `SID_OPTION_BLE_CONNECTION_POLICY` and `SID_BLE_USER_CFG_INACTIVITY_TIMEOUT` remain functional on NCS, as these are handled at the Sidewalk stack level.

1.7.3 Legacy nRF5 SDK Compatibility

Devices using the older nRF5 SDK (SoftDevice-based BLE stack) continue to support full runtime BLE power optimization through `SID_OPTION_BLE_USER_CONFIG`. Applications that must support both platforms should account for this difference in behavior and use compile-time configuration as the default strategy for NCS builds.

Chapter 2

FSK Power Optimization

2.1 Introduction

FSK (Frequency Shift Keying) Sub-GHz communication in Amazon Sidewalk provides long-range connectivity with optimized power consumption. This chapter details power optimization techniques specific to FSK links.

2.2 Device Profile Optimization

2.2.1 Profile 1 vs Profile 2

Choose the appropriate profile based on power requirements:

2.2.1.1 Profile 1 (Fixed Configuration)

```
// Profile 1 - Fixed configuration with infinite RX windows
struct sid_device_profile profile1 = {
    .unicast_params = {
        .device_profile_id = SID_LINK2_PROFILE_1,
        .rx_window_count = SID_RX_WINDOW_CNT_INFINITE,
        .beacon_interval_unit = SID_LINK2_BEACON_INTERVAL_UNIT_1,
        .wakeup_type = SID_TX_AND_RX_WAKEUP,
        .l2_rx_duration_sec = 0 // Infinite for Profile 1
    }
};

sid_error_t err = sid_option(handle, SID_OPTION_900MHZ_SET_DEVICE_PROFILE,
                             &profile1, sizeof(profile1));
```

Power Characteristics:

- Fixed configuration with infinite RX windows
- Always reachable for downlink messages
- Higher power consumption due to continuous listening
- Limited configurability

2.2.1.2 Profile 2 (Configurable for Power Optimization)

```
// Profile 2 - All configurable parameters for power optimization
struct sid_device_profile profile2 = {
    .unicast_params = {
        .device_profile_id = SID_LINK2_PROFILE_2,
        .rx_window_count = SID_RX_WINDOW_CNT_INFINITE, // Fixed for FSK
        .beacon_interval_unit = SID_LINK2_BEACON_INTERVAL_UNIT_2, // 1, 2, or 3
        .unicast_window_interval.sync_rx_interval_ms = 315, // 63-5040ms
        .wakeup_type = SID_TX_AND_RX_WAKEUP, // TX_ONLY, RX_ONLY, TX_AND_RX
        .l2_rx_duration_sec = 3600 // 0 to 65535 seconds
    }
};

sid_error_t err = sid_option(handle, SID_OPTION_900MHZ_SET_DEVICE_PROFILE,
                             &profile2, sizeof(profile2));
```

Profile 2 Configurable Parameters:

- beacon_interval_unit: 1 (10.08s), 2 (20.16s), 3 (30.24s)
- sync_rx_interval_ms: 63, 315, 630, 945, 2520, 3150, 5040ms
- wakeup_type: SID_TX_ONLY_WAKEUP, SID_RX_ONLY_WAKEUP, SID_TX_AND_RX_WAKEUP
- l2_rx_duration_sec: 0 (no downlink) to 65535 seconds

2.3 Beacon Synchronization Optimization

2.3.1 Beacon Interval Configuration

Configure beacon intervals for power efficiency:

- SID_LINK2_BEACON_INTERVAL_UNIT_1: 10.08s (shortest interval)
- SID_LINK2_BEACON_INTERVAL_UNIT_2: 20.16s (balanced)
- SID_LINK2_BEACON_INTERVAL_UNIT_3: 30.24s (least frequent)

2.4 RX Window Optimization

2.4.1 RX Duration Configuration

Profile 2 allows configurable L2 RX duration for power optimization:

```
// Profile 2 RX duration configuration for power optimization
void configure_profile2_rx_duration(enum application_type app_type) {
    struct sid_device_profile profile = {
        .unicast_params = {
            .device_profile_id = SID_LINK2_PROFILE_2,
            .rx_window_count = SID_RX_WINDOW_CNT_INFINITE, // Fixed for FSK
            .wakeup_type = SID_TX_AND_RX_WAKEUP
        }
    };

    switch(app_type) {
        case SENSOR_ONLY:
            profile.unicast_params.l2_rx_duration_sec = 0; // No downlink listening
            break;
    }
}
```

```
    case STATUS_REPORTER:
        profile.unicast_params.l2_rx_duration_sec = 300; // 5 minutes
        break;
    case INTERACTIVE_DEVICE:
        profile.unicast_params.l2_rx_duration_sec = 3600; // 1 hour
        break;
    case ALWAYS_REACHABLE:
        profile.unicast_params.l2_rx_duration_sec = 0; // Infinite (same as Profile 1)
        break;
}

sid_option(handle, SID_OPTION_900MHZ_SET_DEVICE_PROFILE,
           &profile, sizeof(profile));
}
```

2.4.2 RX Window Separation

Optimize RX window separation for Profile 2:

- **63 ms**: Highest responsiveness, highest power
- **315 ms**: Balanced responsiveness and power
- **630 ms**: Lower responsiveness, better power
- **5040 ms**: Lowest responsiveness, best power

2.5 FSK Transmit Power Control (TPC)

Amazon Sidewalk does include FSK TPC algorithm that dynamically manages transmit power between Gateway (GW) and End Point (EP) devices by utilizing RSSI and SNR measurements to optimize power consumption while maintaining reliable communication.

How It Works:

- EP sends uplink message using current power level
- Gateway measures RSSI/SNR and calculates optimal power adjustment
- Gateway provides power adjustment feedback via ACK messages
- EP maintains a Channel-Power Mapping Table for 69 channels and adjusts transmit power accordingly

Key Features:

- Dynamic power adjustment range: -20 dBm to +20 dBm
- Channel-specific power mapping for optimal performance
- Quick response to changing link quality conditions
- Configurable SNR thresholds based on data rates

Benefits:

- Optimizes power consumption while maintaining reliable communication
- Adapts to real-time RF conditions
- Channel-specific power optimization

Note: This feature requires enabling application layer acknowledgment (AWS IoT) for optimal operation.

2.5.1 FSK TPC Testing Results

The FSK TPC feature was evaluated under different environmental conditions to assess its power optimization performance. Testing scenarios included interference-free environments, weak interference (4 additional 0 dBm endpoints), strong interference (4 additional 20 dBm endpoints), and various attenuation levels.

Test Environment Configurations:

- **Interference-free:** Single Endpoint (EN) communicating with single Gateway (GW)
- **Weak interference:** Single EN with connected GW, plus 4 additional 0 dBm EN transmitting
- **Strong interference:** Single EN with connected GW, plus 4 additional 20 dBm EN transmitting
- **Attenuation:** Testing TPC performance with different attenuation levels between EN and GW

TPC Performance Results:

Table 2.1: FSK TPC Power Consumption Results

Environment	EN TPC	GW TPC	TPC On Power (mA)	TPC Off Power (mA)	Power Reduction
Interference-free	On	Off	26.868	73.090	63.24%
Interference-free	Off	Off	26.532	74.430	64.35%
Interference-free	On	On	22.886	76.242	69.98%
Interference-free	Off	On	21.726	75.978	71.40%
Weak Interference	On	Off	37.929	41.465	8.53%
Weak Interference	Off	Off	27.300	63.136	56.76%
Strong Interference	On	Off	34.922	60.646	42.42%
Strong Interference	Off	Off	26.777	46.355	42.23%

Key Findings:

- **Maximum power reduction:** Up to 71.40% when both EN and GW support TPC in interference-free environments
- **Interference resilience:** TPC maintains 42% power reduction even under strong interference conditions
- **Optimal configuration:** Best results achieved with EN TPC Off and GW TPC On in interference-free scenarios
- **Interference impact:** Weak interference significantly reduces TPC effectiveness (8.53% reduction with EN TPC On)
- **Consistent performance:** TPC provides reliable power savings across different interference scenarios

Implementation Recommendations:

- Consider EN TPC configuration based on expected interference levels
- Monitor interference conditions to optimize TPC settings
- Expect significant power savings (40-70%) in typical deployment scenarios

Chapter 3

LoRa Power Optimization

3.1 Introduction

LoRa (Long Range) Sub-GHz communication provides excellent range and power efficiency for Amazon Sidewalk devices. This chapter covers power optimization techniques specific to LoRa links, including CSS (Chirp Spread Spectrum) modulation characteristics.

3.2 Device Profile Optimization

3.2.1 Profile A vs Profile B vs Profile C

Choose the appropriate LoRa profile based on power and connectivity requirements:

3.2.1.1 Profile A (Customizable RX Windows)

```
// Configure LoRa Profile A with customizable RX window count
struct sid_device_profile profileA = {
    .unicast_params = {
        .device_profile_id = SID_LINK3_PROFILE_A,
        .rx_window_count = 0, // Custom: 0 for uplink-only, or any number needed
        .unicast_window_interval.async_rx_interval_ms = 5000, // 5 second intervals
        .wakeup_type = SID_TX_AND_RX_WAKEUP
    },
    .profile_misc_config = {
        .link3_misc_config = {
            .low_latency = SID_LINK3_LOW_LATENCY_DISABLE
        }
    }
};
```

Profile A Characteristics:

- **Fully customizable RX window count** (any number, including 0 for uplink-only)
- RX window count of 0 = no downlink capability (maximum power savings)
- Power consumption scales directly with RX window count
- No periodic network sync
- Suitable for applications with predictable or no downlink needs

3.2.1.2 Profile B (Infinite RX Windows)

```
// Configure LoRa Profile B with infinite RX windows (fixed)
struct sid_device_profile profileB = {
    .unicast_params = {
        .device_profile_id = SID_LINK3_PROFILE_B,
        .rx_window_count = SID_RX_WINDOW_CNT_INFINITE, // Fixed: must be infinite
        .unicast_window_interval.async_rx_interval_ms = 5000, // 5 second intervals
        .wakeup_type = SID_TX_AND_RX_WAKEUP
    },
    .profile_misc_config = {
        .link3_misc_config = {
            .low_latency = SID_LINK3_LOW_LATENCY_DISABLE
        }
    }
};
```

Profile B Characteristics:

- **Fixed infinite RX window count** (cannot be changed)
- Continuous downlink reachability with periodic RX windows
- Periodic network sync (configurable interval: 1-240 mins, default 5 mins)
- Higher power consumption due to continuous listening
- Suitable for applications requiring always-on connectivity

3.2.1.3 Profile C (Continuous RX Mode)

```
// Configure LoRa Profile C with continuous RX mode
struct sid_device_profile profileC = {
    .unicast_params = {
        .device_profile_id = SID_LINK3_PROFILE_C,
        // rx_window_count and async_rx_interval_ms are automatically set:
        // rx_window_count = SID_RX_WINDOW_CONTINUOUS (0xFF)
        // async_rx_interval_ms = SID_LINK3_RX_WINDOW_SEPARATION_CONTINUOUS (0)
        .wakeup_type = SID_TX_AND_RX_WAKEUP
    },
    .profile_misc_config = {
        .link3_misc_config = {
            .low_latency = SID_LINK3_LOW_LATENCY_DISABLE
        }
    }
};
```

Profile C Characteristics:

- **Continuous RX mode** - device stays in receive mode continuously
- rx_window_count = SID_RX_WINDOW_CONTINUOUS (0xFF)
- async_rx_interval_ms = SID_LINK3_RX_WINDOW_SEPARATION_CONTINUOUS (0)
- No periodic network sync (same as Profile A)
- **Lowest downlink latency** - immediate response to incoming messages
- **Highest power consumption** - not suitable for battery-powered devices

- Suitable for line-powered devices requiring minimal downlink latency

Power Consumption Comparison:

Table 3.1: LoRa Profile Power Consumption Comparison

Profile	Power Consumption	Downlink Latency	Use Case
Profile A (rx=0)	Lowest	No downlink	Uplink-only sensors
Profile A (rx>0)	Low-Medium	Medium	Battery devices with occasional DL
Profile B	Medium-High	Low	Always-reachable battery devices
Profile C	Highest	Lowest	Line-powered, latency-critical

3.3 Low Latency Configuration

3.3.1 Low Latency Mode

Configure low latency mode for time-critical applications:

```
// Enable low latency mode (higher power consumption)
struct sid_device_profile low_latency_profile = {
    .unicast_params = {
        .device_profile_id = SID_LINK3_PROFILE_B,
        .rx_window_count = SID_RX_WINDOW_CNT_INFINITE,
        .async_rx_interval_ms = 5000,
        .wakeup_type = SID_TX_AND_RX_WAKEUP
    },
    .profile_misc_config = {
        .link3_misc_config = {
            .low_latency = SID_LINK3_LOW_LATENCY_ENABLE // Enable for low latency
        }
    }
};
```

Power Impact:

- Increased transmission repetitions
- Faster battery drain
- Use only when latency is critical

3.4 LoRa Transmit Power Control (TPC)

LoRa Transmit Power Control (TPC) is a feature that optimizes power consumption while maintaining reliable communication by intelligently adjusting transmission power levels. The system uses a probe power level for testing lower settings and a normal power level (probe power + 4dB margin) for regular transmissions. It operates in two phases: Search Phase, which finds the lowest workable power using binary search, and Trial Phase, which periodically checks (every 12 hours) if power can be reduced further. TPC monitors message acknowledgments from existing user data traffic without generating extra control messages. If messages fail to get acknowledged, TPC automatically increases power to maintain connectivity.

Key Features:

- Conservative power adjustment using transport-ack feedback
- Binary search for initial power setting, then linear adjustment

- Normal power range: +4 dBm to +22 dBm with small step adjustments (typically 1 dB)
- Probe power range: 0 dBm to +17 dBm
- Channel-specific power mapping with safety margins
- Self-recovery mechanisms if connection degrades

Algorithm Approach:

- Uses acknowledgments from user data transmissions for power optimization
- Adaptive algorithm with binary search and trial phases
- Cooldown periods between adjustments
- Only reduces power when proven successful through transport-ack

Benefits:

- Optimizes battery life by reducing unnecessary transmission power
- Reduces channel congestion from multiple gateway reception
- Maintains reliable connectivity with conservative approach

Note: This feature requires enabling application layer acknowledgment (AWS IoT) for optimal operation.

3.4.1 LoRa TPC Testing Results

The LoRa TPC feature was evaluated to assess its power optimization performance across different hardware platforms and device profiles. Testing focused on LoRa Profile A and Profile B configurations with high acknowledgment rates to validate TPC effectiveness.

Test Environment Configurations:

- **Device profiles:** LoRa Profile A and Profile B
- **ACK success rate:** 95% application layer acknowledgment (AWS IoT)
- **Test environment:** Controlled shielding room with distance <1m between device and gateway, single device and gateway operation

TPC Performance Results:

Table 3.2: LoRa TPC Power Consumption Results

Environment	TPC On Power (mA)	TPC Off Power (mA)	Power Reduction	Avg RF Power TPC On (dBm)	Avg RF Power TPC Off (dBm)
ACK 95%, Profile A	71.932	128.680	44.1%	11	19
ACK 95%, Profile B	68.896	128.625	46.44%	11	19

Key Findings:

- **Maximum power reduction:** Up to 46.44% achieved with Profile B configuration
- **Profile performance:** Both Profile A and Profile B show effective TPC operation with consistent results
- **ACK rate dependency:** TPC effectiveness correlates with acknowledgment success rates, performing optimally at 95% ACK rate
- **Reliable power savings:** TPC provides consistent 44-46% power reduction across different profiles

Implementation Recommendations:

- Deploy TPC in environments with high ACK success rates for maximum benefit
- Monitor ACK rates to ensure TPC operates in optimal conditions
- Expect 44-46% power reduction in typical high-reliability scenarios

Chapter 4

Change History

Version	Summary of Changes
Protocol Stack 1.0, Document Revision A	First release of Power Optimization Developer Guide.
Protocol Stack 1.0, Document Revision A.1	Add LoRa Profile C documentation for continuous RX mode. Add power consumption comparison table for all LoRa profiles.

Glossary

BLE	Bluetooth low energy.
CSS	Chirp Spread Spectrum.
Downlink	Data sent from a Gateway to an Endpoint.
Endpoint	A device that accesses services of Amazon Sidewalk to transport messages to and from the application services through AWS IoT.
FFN	Frustration Free Networking.
FFS	Frustration Free Setup.
FSK	2-Gaussian Frequency Shift Keying modulation scheme.
Gateway	Connections between Endpoints and Amazon Sidewalk are established and maintained through these devices. Gateways are Amazon and Ring devices. Amazon Sidewalk messages are transported between Amazon Sidewalk and the AWS IoT service through these devices.
ISR	Interrupt Service Routine.
LoRa	Long Range proprietary protocol based on Chirp Spread Spectrum modulation scheme (CSS).
MTU	Maximum Transmission Unit.
PAL	Platform abstraction Layer.
SBDT	Sidewalk Bulk Data Transfer.
Sid API	Amazon Sidewalk Application Programming Interface.
Uplink	Data sent from an Endpoint to a Gateway.