**oneM2M White Paper:**

**Power Optimization for IoT**

When it comes to optimizing power consumption, connected IoT devices have a unique set of technical challenges that differ from smartphones. For this reason, along with the fact that connected IoT devices already outnumber smartphones 2 to 1 (a gap that is expected to widen to 4 to 1 by [2025](https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/#:~:text=The%20total%20installed%20base%20of,that%20are%20expected%20in%202021) [1]), optimizing power consumption for this category of device has been a top priority for oneM2M.

**Power Optimization for IoT**

IoT devices differ from smart phones in a few key ways. IoT devices typically handle comparatively small amounts of data and are instead optimized to prioritize battery life. Battery-powered IoT devices are required to remain operational for long durations (e.g. a decade or more) without recharging or replacing their batteries. They often have a much smaller battery or no battery at all and instead rely on harvesting ambient energy such as solar or vibrations. As such, to conserve their batteries, many of these devices need to remain disconnected and completely powered off for long durations until they are awoken to perform an operation, for instance a sensor that wakes-up monthly to report a reading and then returns to sleep. Other devices must minimize the amount of circuitry they leave powered on so they are still able to be remotely triggered to fully power-up if/when needed, for example an actuator that must be sent a remote command to close a critical valve in a timely manner. Unlike smartphones which can rely on users for daily recharging, and/or transitioning in and out of low power modes when needed, like for airplane mode, or cellular, Wi-Fi or Bluetooth radios that extend the life of the battery, IoT devices typically do not interact with users. Instead, these connected IoT devices must rely on more autonomous forms of management and control within the devices themselves or the networks to which they connect.

**Solutions**

There are several key technologies that can be used to help optimize the power consumption IoT systems. The following are a few examples.

*IoT Device Triggering*

Device triggering technology enables IoT devices to “sleep” for long periods of time when not in use, and intelligently “wake up” when needed. Specifically, IoT devices register to a service in the network and provide instructions for triggering the device. The instructions can include a device’s contact information as well as its sleep schedule. Once registered, the IoT device powers down to sleep to conserve power. Based on its sleep schedule, the device can periodically wakeup for a short duration of time to listen for triggers by optimally enabling only a small, select portion of its circuitry required to receive a trigger, but otherwise, the device consumes a relatively small amount of power. If the service in the network needs to communicate with the device (e.g., send it a command), it schedules a trigger to be sent to the device during a scheduled wake up time. If the device receives a trigger, it fully powers-up, connects to the network and communicates with, or received a command from, the service. Afterwards, the device powers back down and goes back to sleep to continue conserving its power. To provide even further power savings and the ability to trigger an IoT device in a more on-demand manner without having to wait for a scheduled wake-up time, ultra-low power wake-up receiver technology can also be used. This technology is embedded within an IoT device and can receive a low power wake-up signal. The ultra-low power wake-up receiver circuitry does not draw power from the battery of the IoT device. Instead, the receiver is passive in nature and is powered by the energy captured from the radio waves used to transmit the wake-up signal to the IoT device. This solution is analogous to other forms of passive receiver technologies like radio frequency identification (RFID) used in applications like electronic toll collection, however, this ultra-low power wake-up receiver technology operates in longer-range wide-area cellular networks.

*IoT Device Twins*

Device twin technology enables physical IoT devices deployed in the field to be represented by digital counterparts that reside within services hosted in the cloud or on network edge compute nodes. These digital counterparts, or twins, are then used to exchange information between the devices themselves and the applications and services that wish to interact with them. For example, a sensor can publish readings to its digital device twin. Use of device twin technology can provide a huge power consumption benefit for IoT devices because device twins allow IoT devices to disconnect, power down and sleep for longer durations of time without impacting the functionality of applications, services and other devices that require data from these IoT devices. Even while IoT devices are sleeping, their data can be accessed from their device twin.

Device twin technology also reduces the number of requests IoT devices must send and receive, since they only need to interact directly with their device twins instead of other applications, services, or devices. An IoT device is only required to wake-up long enough to send an update to its device twin, and then can immediately go back to sleep. Even an actuator-type device, which requires receiving commands, can benefit from device twins because actuator commands can be stored within the digital twin. The IoT device can then wake-up based on its own schedule, check its device twin for any new commands, perform these commands, update its device twin with the results, and then return to sleep.

*IoT Message Profiles*

Many IoT devices generate a stream of sensor readings on a periodic schedule or event basis. The number of bytes of data included within each message the device sends or receives can have a major impact on the battery life of the device. The larger the message sent or received over the network, the longer that device must remain powered to send the message. Over time, this can add up and dramatically reduce the lifetime of the device battery.

In many IoT use cases, there is a certain amount of information contained within these messages that is static in nature and does not change – for example, certain fields within upper layer protocol headers of the message or in the data payload of the message. This static information is critical to processing the message and provides critical context to the services and applications, like analytics, consuming and processing the message. However, this static information can result in significant overhead on the device and the network if included in every message the device sends. To alleviate this, IoT message profiles can be used. Within a profile, static information elements can be defined along with criteria defining which device(s) and/or message type(s) a profile is applicable to. IoT message profiles can be configured within the network services to which an IoT device sends it messages, like a cloud or edge data service. As the service receives messages, it compares the messages against the profiles and if a match is found, the messages are enriched with the static information defined in the profiles and further processed by the service or applications (e.g., analytics operations are performed). This allows devices to send only the bare minimum amount of information that has dynamically changed per message (e.g., a sensor reading) and streamline the size of the message. IoT message profiles can also be configured on IoT devices themselves and used in a similar manner by the devices to enrich the messages they receive with static information. Therefore, the use of IoT message profile technology helps minimize the size of messages flowing between IoT devices and services in the network, which in turn translates into less energy consumed by the IoT devices as well as reduced load and energy consumption by the network infrastructure used by the IoT devices.

*Offloading IoT Event Processing*

Offloading event detection and processing from user applications to services deployed in the cloud and on edge compute nodes in the network reduces the overall number of messages exchanged over the network. Rather than an application retrieving every sensor reading published by the device(s) it is interested in and checking whether sensor readings have crossed a certain threshold value of interest, the application instead offloads an event detection rule to a service in the network. Within the rule, the application specifies the devices it is interested in, the data of interest from these devices, conditions of interest for this data, and actions that are to be performed if/when these conditions are detected. For example, if pressure within a specified boiler device exceeds a specified threshold, then power to the boiler is turned off. Using the rule, the service can efficiently offload and perform this work on behalf the application.

Offloading eliminates the need to send individual copies of device data to each application to process, which greatly reduces the amount of messaging in the system. It also enables the pooling and reuse of compute and storage resources of the service by the IoT devices and applications in the system. For example, data offloaded by IoT devices to a service in the network can be used both for the monitoring of a single condition defined in a single rule from a single application, and also simultaneously used by the service to monitor many conditions defined in many rules from many applications. As result, offloading IoT event processing can play a major role in helping optimize energy consumption in the overall 5G system.

*To Do – Consider add additional solutions from oneM2M sustainability case study(s). Some potential candidates:*

* *Time synchronization management [<timeSyncBeacon>]*
* *IoT message delivery scheduling and aggregation [<schedule>, <delivery>]*
* *Interworking to different IoT technologies [IPEs]*
* *Performing optimized operations on groups of devices [<group>, <fanoutPoint>]*
* *Managing end-to-end sessions between applications [<e2eQosSession>]*
* *Offloading a software management campaign targeting a set of devices from an application to service layer [<softwareCampaign>, <mgmtObj>]*
* *Assist with managing congestion in underlying network transports [<nwMonitoringRequest>]*
* *Offloading semantic reasoning operations from applications to service layer [<reasoningRules>, <reasoningJobInstance>]*
* *Offloading and managing data storage and brokering on behalf of devices and applications (e.g., storage, discovery, authorization, stats collection, charging of data) [data management resources, <accessControlPolicy>, <statsConfig>, <eventConfig>, <statsCollect>]*
* *Offloading the execution of individual requests from applications to service layer [<mgmtCmd>, <action>, <dependency>, <transactionMgmt>, <transaction>]*
* *Offloading data analytics operations from devices and applications to service layer [<timeSeries>, <timeSeriesInstance>]*
* *Offloading semantic mashup operations from applications to service layer [<semanticMashupJobProfile>, <semanticMashupInstance>, <semanticMashupResult>]*
* *Sending requests to 3GPP devices using Non-IP Data Delivery (use of 3GPP control plane to deliver messages)*
* *Detect network congestion and throttle messages to help alleviate congestion*
* *Managing network connectivity used to communicate with devices [CMDH]*

**Conclusion**

As the mass scale out of connected IoT devices continues, the importance of using and responsibly deploying the technologies described above will become critical. This will help reduce the power consumption of IoT systems and make these deployments more sustainable. For this reason, oneM2M is committed and actively engaged in forward looking research to identify technologies that can further optimize the sustainability of IoT deployments.

[1] <https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/#:~:text=The%20total%20installed%20base%20of,that%20are%20expected%20in%202021>.