GATEWAY

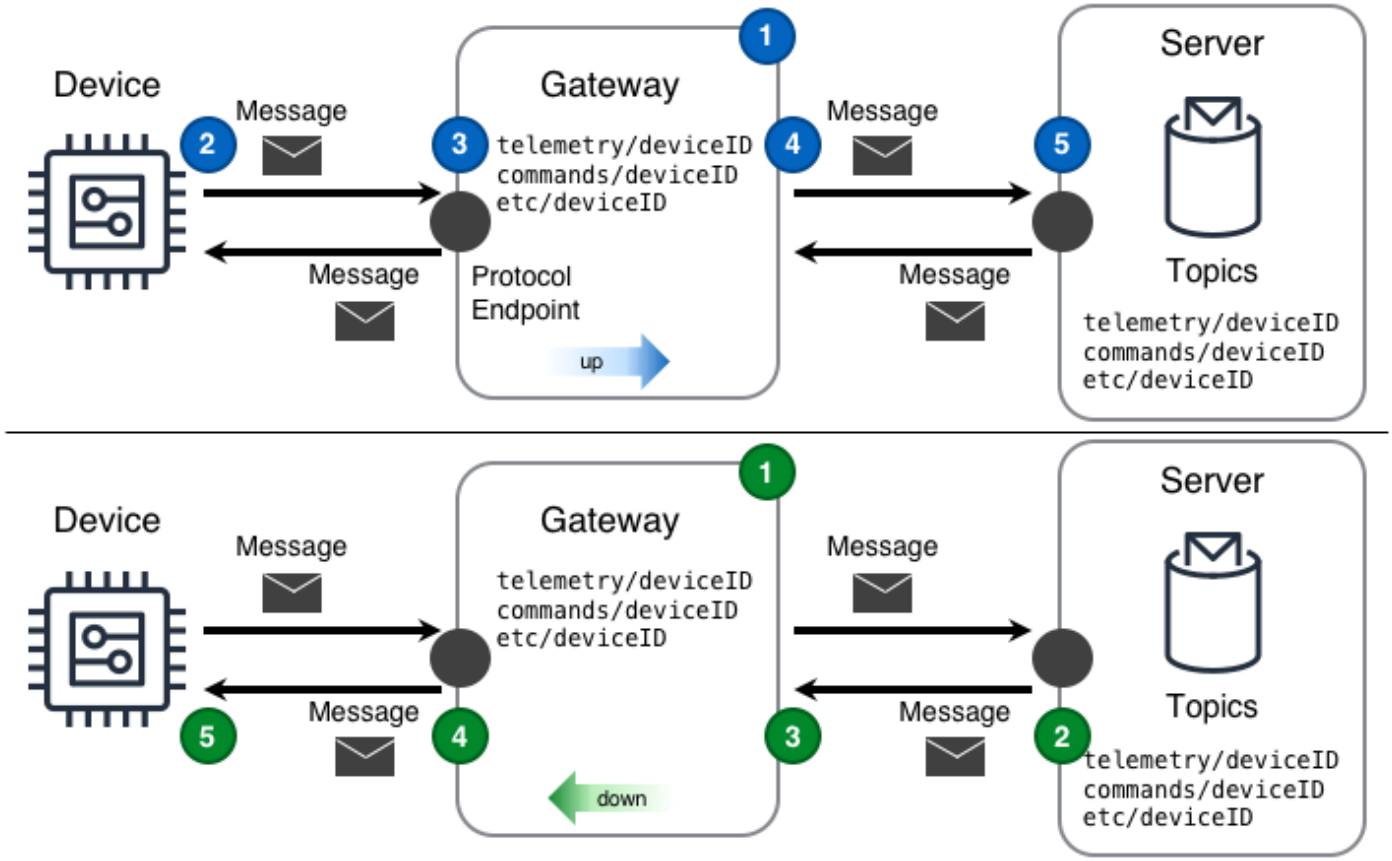
Challenge

Endpoints in an IoT solution are often not capable enough to connect directly to the internet nor are they operating on networks with direct access to the internet. Even with these constraints, obtaining data from and interacting with endpoints requires a mechanism of connectivity.

Solution

IoT solutions use the Gateway design to overcome the common constraints experienced by endpoints. In doing so a gateway enables reliable and secure communication with otherwise inaccessible endpoints. Additionally a gateway enables continued isolation between the local endpoints and cloud connectivity.

The Gateway design shown in the following diagram can deliver this functionality. In the diagram, the Server resides on a cloud. The Gateway resides on a network the Device can access.



Both designs in the above diagram expose a gateway endpoint using the same type of protocol endpoint as the server. In both the *up* and *down* gateway diagrams, the gateway is connected to the server.

Up gateway (aka “North”)

1. the “up” gateway design is configured to mirror “upward” messages; in the diagram, messages arriving from the device with the `telemetry/deviceID` topic will be mirrored up to the server using the same topic.
2. The device publishes a message containing the measurement via a transport protocol to the local protocol endpoint exposed by the gateway.
3. The gateway receives the message.
4. The gateway publishes the message to the server on the same topic as the received message.
    - if the gateway is unsuccessful sending the message to the server, the message is processed using an upward message approach Buffer Messages
5. The server receives the message.

Down gateway (aka “South”)

1. the “down” gateway is configured to listen to the server and mirror “downward” messages; in the diagram, messages arriving from the server with the `commands/deviceID` topic will be mirrored down to the device listening for messages with the same topic.
2. The server publishes a message to the gateway via the transport protocol’s endpoint.
3. The gateway receives the message.
4. The gateway publishes the message to the device listening on the gateway endpoint on the same topic as the received message
    - if the gateway is unsuccessful sending the message to the device, the message is processed using a downward message approach. Buffer Messages
5. The device receives the message.

Considerations

When implementing this design, consider the following questions:

Why should the Gateway explicitly mirror/route only certain topics in a certain direction?

Since message topics are the interface through which components in an IoT solution interact with one another, by configuring a Gateway design to take explicit steps to mirror certain topics in certain directions, devices in the solution will only have the ability to interact with those topics which are essential to perform a device’s intended function. This aligns well with the security best practice of following the principle of least privilege for device-to-cloud and cloud-to-device communications.

How should the Gateway process data when the network to the Device is unavailable?

The simple answer is, the Gateway needs a *downward message approach* used to save the messages on the gateway until they can be reported to the device.
Unfortunately the simple answer belies the reality, which is more complex. A key thing to determine is the right approach to take with the downward messages when the network is absent.

How should the Gateway process data when the network to the Server is unavailable?

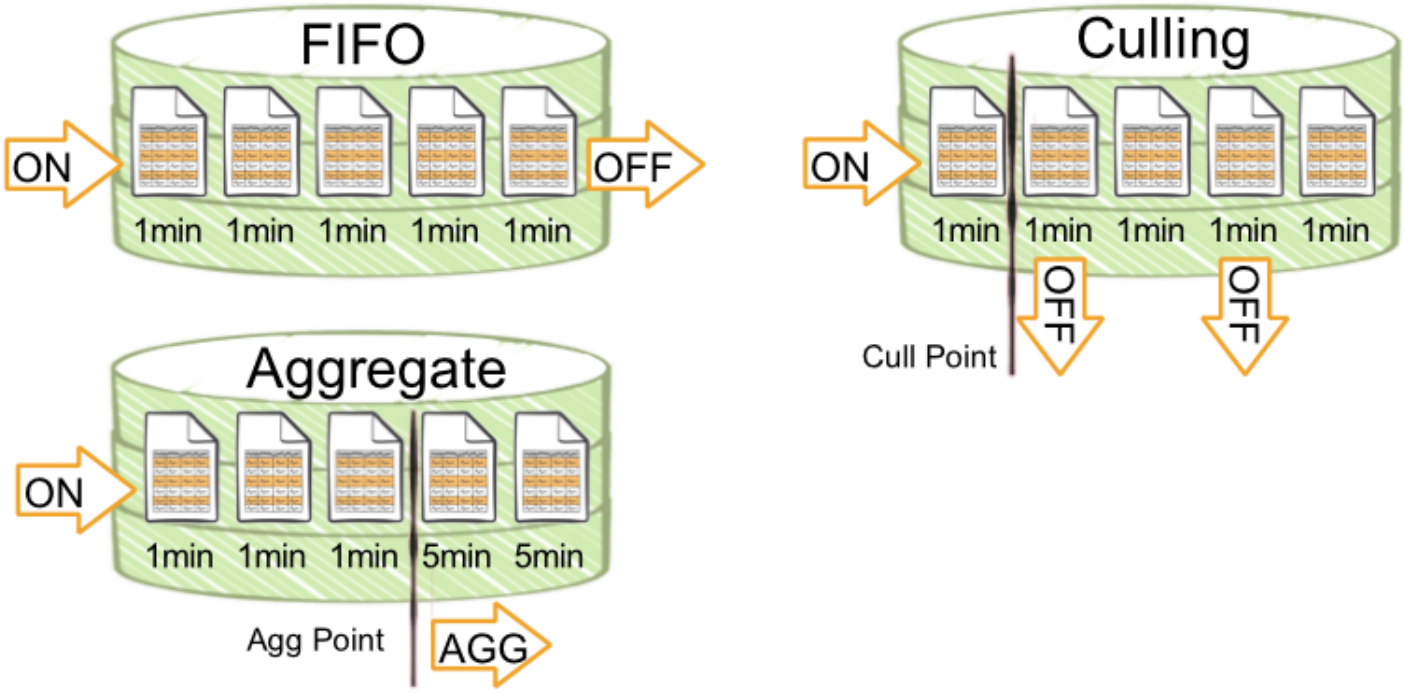
The simple answer is, the Gateway needs an *upward message approach* used to save the messages on the gateway until they can be reported to the server.

Unfortunately the simple answer belies the reality, which is more complex. A key thing to determine is the right approach to take with the upward messages when the network is absent.

What approach should be used when storing messages for later delivery?

Generally, local storage and processing of messages on the gateway will follow a First-In-First-Out (aka. **FIFO**) approach. That being said, the answer *might* be different depending upon the data actually contained in the message. In this case determining how the gateway's logging approach influences the actual reported data can help avoid future solution issues.

The general categories of approaches to consider are: **FIFO**, **Culling**, and **Aggregate** as shown in the following diagram.



FIFO – This approach is usually straightforward to implement and useful in a wide variety of situations. In the message processing diagram this approach's data arrives from the left and exits to the right when the allocated local storage is full. Examples of data include: operations measurements and general-purpose telemetry.

Culling – This approach is useful for retaining absolute point values at a loss of curve detail. In the message processing diagram this approach's data arrives from the left. Once local storage has been filled beyond a *culling point*, some sweeper logic then removes every other (or every N^{th}) sample. Examples of data include: kW, Amperage, Voltage, etc.

Aggregate – This approach is useful when the detailed shape of the curve is not as important as the minimum, maximum, average, and sum values over a period of time. In the message processing diagram this approach's data arrives from the left. Once local storage has filled beyond an *aggregate point* some sweeper logic performs aggregation on the stored values. Examples of data include: kWh, insolation, flow, CPU time, temperature, wind speed, etc.

Which local area network topology is used by devices connected to a gateway?

There are two topologies that devices most commonly take: a mesh network and a hub-and-spoke network (aka. star network).

Hub-and-spoke Network - A gateway in a hub-and-spoke network provides all device connectivity to and from the cloud, device-to-device communication, and additional local capabilities such as time series data storage, data analysis, and machine learning inference. Since the gateway in this topology provides device-to-device communication, *upward messages* can come *from* a device spoke to the gateway and then immediately *down* to another device spoke, **or** *upward messages* can come *from* a device spoke destined for the server's protocol endpoint. The message topic should be route-able by the gateway to either type of destination.

Mesh Network – A gateway in a mesh network provides cloud routing capabilities to some or all of the devices on the mesh. Since devices physically close to one another communicate directly, a gateway is usually not responsible for device-to-device communication; however, the gateway may provide additional local capabilities such as time series data storage, data analysis, and machine learning inference.

In both a hub-and-spoke or a mesh network topology, to enable explicit routing of all messages, each device and the gateway itself should be addressable by a unique message topic.

Example

```
<tbd written scenario>
```