MQTT-S, A MQTT-BASED PUB/SUB PROTOCOL FOR SENSOR NETWORKS In this section we will describe MQTT-S, a pub/sub protocol based on MQTT and especially designed for WSNs. MQTT-S is developed based on the following design points:

 1) As close as possible to MQTT: This allows a seamless connection of the SA devices to an MQTT broker, thus enabling a smooth integration of the WSNs with the existing communication infrastructure. This also enables a very simple and lossless implementation of the gateways. As a consequence, MQTT-S supports not only all MQTT features (e.g., those described in Section IIIB) but also almost all the message flows and contents defined by MQTT.

2) Optimized for tiny SA devices: The protocol is designed in such a way that it can be implemented for lowcost, battery-operated devices with limited processing and storage. Whenever complexities are required, they reside on the gateway/broker’s side; the client running on the SA devices is kept as simple as possible.

3) Consideration of wireless network constraints such as high link failure rates, low bandwidth, and short message payload: Wireless radio links in general have higher failure rates than wired links, owing to their susceptibility to fading and interference disturbances. They have also a lower transmission capacity. For example, WSNs based on the IEEE 802.15.4 standard provide an aggregate (shared) bandwidth of a theoretic maximum of 250 kbit/s in the 2.4 GHz band [10]. In practice the bandwidth is even lower because of free channel assessment and retransmissions. Procedures should be defined to reduce the risk of having SAs disconnected from the infrastructure owing to link failures or network congestion.

Moreover, to be resistant against transmission errors, wireless networks have a much shorter packet length than wired networks. In the case of IEEE 802.15.4, the physical layer provides a maximum packet length of 128 bytes. Half of these 128 bytes could be taken away by the overhead information required by other supporting layers and functions such as MAC, network, security, etc., see for example [22] for the case of ZigBee. That means, MQTT-S messages should be shorter than 64 bytes. This is very little if human-readable data formats (such as topic names) are to be supported. How MQTT-S copes with these issues is described below. 4) Network independent: MQTT-S is designed to run on any network that provides the two following services: a) Point-to-point data transfer service (unicast service): A datagram service that allows the transport of messages between any two points based on their network address. The two points involved may be multiple hops away from each other.