# New connectivity protocols for small IoT devices

**By Michael May,** Express Logic

*The most popular of the new connectivity protocols for small IoT devices are MQTT, CoAP, LWM2M and 6LoWPAN. This article gives some information on each of these, and a comparison of their differences and relative merits.*

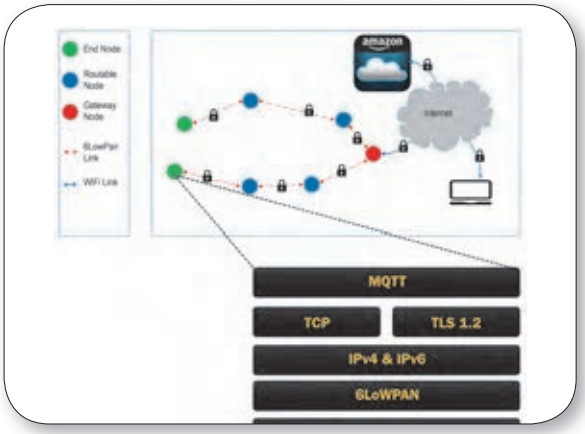


*Figure 1. MQTT is a publish/subscribe-messaging protocol designed for lightweight machine-to-machine (M2M) connections.*

The expansion of the Internet of Things (IoT) to incorporate small devices is projected to incorporate tens of billions of things by 2020. That's an enormous number of devices, and each one of them is a source or a consumer of tons of data. The worldwide deployment of smart meters is expected to reach 131 million by 2018, primarily in residential settings. Those meters measure and report on electricity used 24/7, providing daily, even hourly reporting. And that's just smart meters. Think of home thermostats, surveillance cameras, traffic signals, and all the other things that gather data and send it to the cloud for analysis and additional processing to better support consumer needs.

As exciting as these opportunities are, the Internet was not designed to accommodate this many nodes with a 32-bit field for device addresses. Fortunately, in 1994 the Internet Engineering Task Force (IETF) anticipated the need for more IP addresses and initiated the development of a suite of protocols and standards now known as Internet Protocol Version 6 (IPv6). IPv6 uses a 128-bit address size compared with the 32-bit system used in IPv4, and allows for as many as 3.4x1038 possible addresses.

The expansion of the IoT and the new small devices that are fuelling its expansion, however, require more than just addressability. Many of these devices have small memory, inexpensive CPUs, are battery-powered, yet require real-time responsiveness. Today, we have the hardware technology to create these devices, and small-footprint real-time operating systems (RTOSes) to manage their internal system functions, but common TCP/IP network software protocols are hard-pressed to support them. To address these limitations, a number of new connectivity protocols have been developed. These protocols are intended today for use on resource-constrained, low-end microcontroller-driven, battery-powered IoT devices. The goal is to enable these things

**MQTT/CoAP Comparison**

| MQTT | CoAP |
|---|---|
| Requires TCP, which is larger than UDP, making MQTT not as small as CoAP. | Runs on UDP and thus can be run on extremely resource-constrained environments. |
| Offers publish/subscribe semantics (on the same socket) which makes it easier to program on the IoT device side. IoT cloud service providers such as AWS IoT and Everything and others offer MQTT-based device connectivity. | Is a good mechanism for local network communication, particularly when there is an ecosystem of other CoAP devices. |
| Requires a message broker (server) for its functioning.<br><br>This makes it a good option for remote/cloud communication, since the cloud server acts as the message broker between the IoT device and other app/services.<br><br>This also makes it not a great option for local network communication between devices, because it requires the end-user to deploy an additional broker in the system. | |

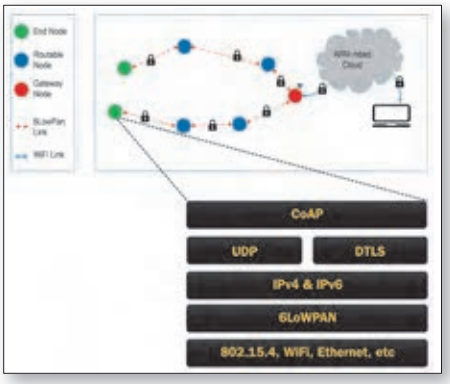


*Figure 2. Express Logic has implemented the Internet-of-Things protocols (CoAP, UDP, IPv6, and 6LoWPAN) in just 25KB of code. This configuration supports a device end-node, with limited IPv6 and 6LoWPAN functionality. The 25KB code size does not include the ThreadX RTOS (6KB) nor the optional DTLS (5KB), or any application code. It represents only the code necessary for the cloud communication.*

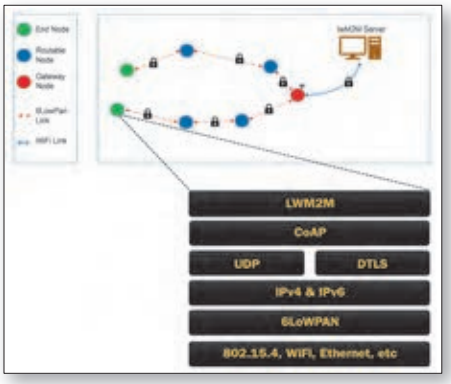


*Figure 3. With LWM2M, the end-node application's interface to Amazon AWS, for example, is through the CoAP protocol. Messages will go from LwM2M to CoAP, through UDP, TLS/DTLS, and IPv6. If Ethernet is the physical transport path for the device, the IPv6 packet goes out directly via the Ethernet driver. If 802.15.4 is the physical transport path of the device, the message is sent out over 6LoWPAN and the 802.15.4 driver.*

to communicate with each other, with the Internet, and with the cloud. In some cases, multiple protocols are available for similar functionality, and the choice of the best one is dependent on the application. The good news is that many protocols are available from many software and system vendors. The bad news is that not all of the implementations are industrial grade. That is: ready for mass market distribution, developed for reliable, long-term IoT performance, and implemented in a format that meets demanding design requirements. This is the IP stack challenge, which IoT developers must understand and anticipate in order to make the right protocol decisions for each product design. The most popular of the new connectivity protocols for small devices are MQTT, CoAP, LWM2M and 6LoWPAN. Here is some information on each of these, and a comparison of their differences and relative merits.

MQ Telemetry Transport (MQTT) is a publish/subscribe-messaging protocol designed for lightweight machine-to-machine (M2M) connections with remote locations where a small code footprint is required, or the network bandwidth is limited. It was originally developed by IBM and is now an open standard. MQTT employs a client/server model, where every sensor end node is a client and connects to a server, known as a broker, over TCP through routable nodes and/or a gateway. The broker might be a cloud service provided by a vehicle manufacturer, for example or a general-purpose supplier such as Amazon. The publisher-subscriber model allows MQTT clients to communicate one-to-one, one-to-many, and many-to-one. Even though MQTT is designed to be lightweight, it has two drawbacks for very constrained devices.

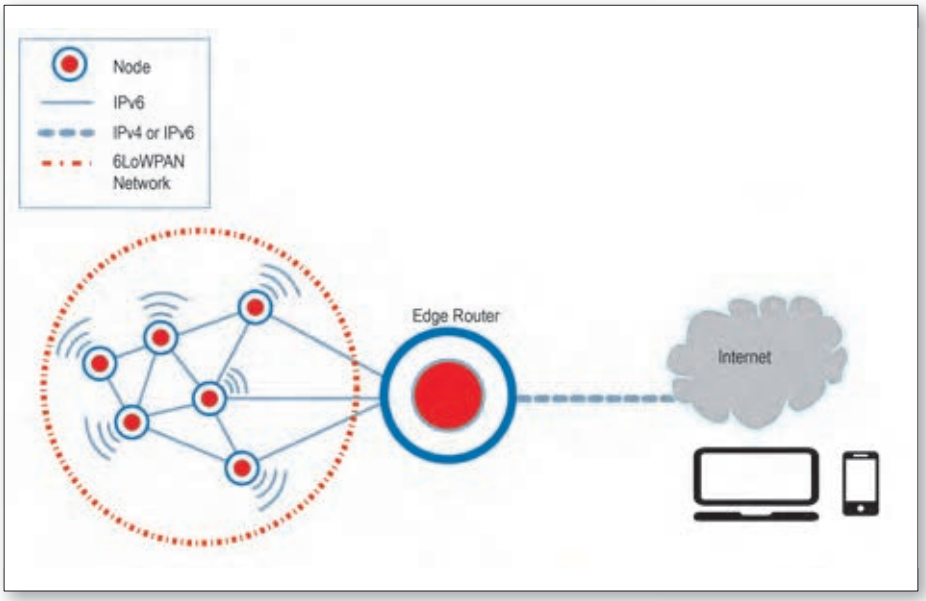


*Figure 4. 6LoWPAN is a mesh network protocol, allowing IPv6 datagrams to be transmitted by low-power short-range radio (such as IEEE 802.15.4).*

1) Every MQTT client must support TCP and will typically hold a connection open to the broker at all times. For some environments where packet loss is high or computing resources are scarce, this is a problem. 2) MQTT topic names are often long strings which make them impractical for 802.15.4 (low-power, low-speed wireless radio). Both of these shortcomings are addressed by the MQTT-SN protocol, which defines a UDP mapping of MQTT and adds broker support for indexing topic names.

Constrained Application Protocol (CoAP) enables constrained devices to communicate with the Internet using similar protocols. CoAP is designed for use between devices on the same constrained network, between devices and general nodes on the Internet, and between devices on different constrained networks, joined by an Internet. CoAP is designed to easily translate to HTTP for simplified integration with the web, while also meeting specialized requirements such as multicast support, very low overhead, and simplicity. Unlike MQTT, which requires TCP, CoAP uses the smaller and simpler UDP, which is extremely important for resource-constrained IoT devices.

Lightweight M2M (LwM2M) is an open industry protocol from the Open Mobile Alliance (OMA), built to provide a lightweight, low-cost means to remotely perform service enablement and application management for IoT embedded devices and connected appliances over wireless connections. It is a communication protocol for use between client software on an M2M device and server software on a M2M management and service platform. LWM2M was designed to overcome issues from technical fragmentation, find a suitable mechanism to cater to the needs of constrained M2M devices, and to generate benefits from decoupling system components via standardized interfaces. The LWM2M protocol has four main characteristics: Its architectural design is based on a representational state transfer application protocol interface, it defines a resource and data model, it has been designed with performance and the constraints of M2M devices in mind, and it reuses and builds on the constrained application protocol secure data transfer standard that has been standardized by the IETF as a variation of the internet HTTP protocol (appropriate for data transfer to and from low-cost connected IoT devices).

6LoWPAN is an acronym of IPv6 over Low Power Wireless Personal Area Networks. The 6LoWPAN concept originated from the idea that the Internet Protocol could and should be applied even to the smallest devices, and that low-power devices with limited processing capabilities should be able to participate in the IoT. 6LoWPAN is a mesh network protocol, allowing IPv6 datagrams to be transmitted by low-power, short-range radio such as IEEE 802.15.4. A low-power radio typically has a range of 30 feet, with low bandwidth (in the kilobits per second range), and small packet size (128 bytes). 6LoWPAN bridges IPv6 and the low-power radio network. 6LoWPAN provides: Open standards including TCP, UDP, HTTP, COAP, MQTT, and websockets, end-to-end IPv6 addressable nodes, no need for a gateway or proxy (a 6LoWPAN border router connects the 6LoWPAN network to the Internet), one-to-many and many-to-one routing, robustness and scalability, use across multiple communications platforms (i.e., Ethernet/Wi-Fi/802.15.4/Sub-1GHz ISM), as well as interoperability at the IP level.

Each of the protocols described relies on an underlying IP stack for IPv6 communication. The new protocols are designed at the application level (except for 6LoWPAN), without regard for the means of transport (i.e., Ethernet, WiFi, or cellular). As such, much of the heavy lifting is relegated to the IP stack, which as it turns out is significantly larger than the cloud protocols themselves. It's not surprising, then, that the underlying IP stack is much more complex, and much more critical to the exchange of information, even though it is more general in design, and not specifically tailored for the cloud. When choosing cloud protocols, it is equally critical - if not more so - to select the right IP stack on which these cloud protocols will rely for proper and efficient operation.

There are various requirements for an industrial grade IP stack. Of course, the IP stack must support the IPv6 protocol. Ideally, this support should be validated and certified by an independent authority. Beyond that, the cloud protocols must be tightly integrated with the IP stack to assure efficiency and correctness of operation under all demanding use cases. The IP stack must be industrial-grade and ready for production use, and must be small, safe, secure, advanced, fast, and easy-to-use.


*Figure 5. X-Ware IoT Platform is built on top of Express Logic high-performance ThreadX RTOS and NetX Duo dual IPv4/IPv6 TCP/IP stack. The industrial-grade platform adds new IoT protocol support including 6LoWPAN, MQTT, CoAP, and LWM2M for securely connecting the smallest of IoT devices to the cloud.*

It would be of no ultimate benefit for a cloud protocol to be small in size so it could fit within the memory constraints of a low-cost microcontroller if the underlying IP stack were too big itself. The IP stack must be small as well, so as not to interfere with the goal of the small cloud protocol. The IP stack should also satisfy popular safety standards for electronic device software, including IEC 61508 SIL 4, IEC 62304 Class C, ISO 26262 ASIL D, UL/IEC 60730, UL/IEC 60335, UL 1998, and EN 50128 SW-SIL 4. This assures its ability to be certified for use in safety-critical systems, as well as being beneficial for use in other systems. The IP stack should be closed - with external access defined by the application, not the stack itself. It should also support security protocols such as IPsec, TLS, SSL, and DTLS. The IP stack should offer advanced technology, such as the ability to communicate with IPv4 as well as IPv6, hardware checksum support where available, and support for optional application protocols beyond the cloud such as AutoIP, DHCP, DNS, and mDNS.

Performance and effi ciency of the IP stack is critical to its mission. It must be able to operate at near wire-speed - the theoretical maximum of the transport hardware - lest it introduce overhead that interferes with its mission. It must also be designed from the ground up for ease of use - with an intuitive API and clean, clear source code - to help developers get products to market faster than less-capable stacks. The IoT is exciting, both for consumers and for vendors of technology that enables the design and development of the kinds of products that consumers want. The new cloud protocols extend the IoT from device to cloud, and are best-suited for use with small-memory, limited-performance microcontrollers. To do so, the cloud protocols must be implemented in a small, efficient fashion, and importantly, must be designed for use on top of a capable, small, fast IP stack. Designers must make a careful evaluation of the underlying IP stack before committing to any cloud-only solution. ■