# COMPARING MESSAGING TECHNIQUES FOR THE IOT

Mike Anderson

Chief Scientist

The PTR Group, Inc.

http://ThePTRGroup.com

mike@theptrgroup.com

# Who is The PTR Group?

- The PTR Group was founded in 2000
- We are involved in multiple areas of work:
  - Robotics (NASA space arm)
  - Flight software (over 35 satellites on orbit)
  - Offensive and defensive cyber operations
    - I'll leave this to your imagination ☺
  - Embedded software ports to RTOS/Linux/bare metal
  - IoT systems architecture and deployment

# Speaker/Author Details


- **Website:**
  - **http://www.theptrgroup.com**
- **Email:**
  - **mailto:mike@theptrgroup.com**
- **Linked-in:**
  - **https://www.linkedin.com/in/mikeandersonptr**
- **Twitter:**
  - **@hungjar**

Almost 40 years in the embedded and real-time industry for both commercial and Government customers.

# What We'll Talk About…

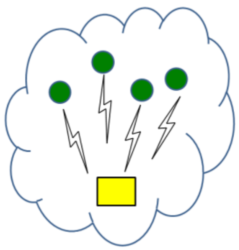
- Connectivity in the IoT
- Messaging models
- The major techniques
- Issues of efficiency
- Summary

# The World of the IoT

- Given the billions of devices that are forecast to be attached to the Internet, communications is a key concern
- Other related topics include the communications media, addressability, protocols, security, ease of use and much more
- We’ll touch on these briefly with respect to how they impact the messaging techniques

# IoT Connectivity Models

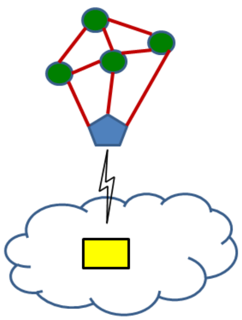
- There are two primary connectivity models used in the IoT – cloud and fog
- In the cloud model, all of the IoT devices are directly connected to the Internet for data transfer to cloud-based servers
  - Unfortunately, this leaves your sensors exposed to the bad guys
- The data analysis people want access to the raw data
  - Maybe there is some hidden nugget in the raw data

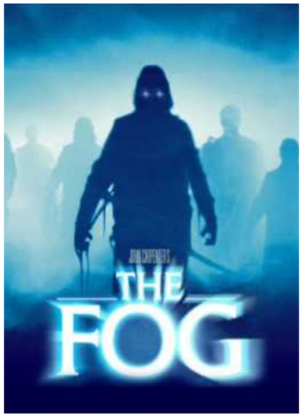


Source: fortune.com

# IoT Connectivity Models (2)

- In the fog model, the sensors are connected to a gateway/ border router and never expose themselves to the Internet directly
- You then can harden the security on the border router (typically Linux) to isolate and protect the sensors from direct attack
- However, all data then needs to be relayed from the router to/from the cloud
- Often, the router is doing data filtering and aggregation to limit the amount of traffic to the servers
  - Reduces probability of finding the nugget hidden in the raw data






Source: youtube.com

# Communications Media

- There are a lot of communications techniques that are vying for developer's attention
- These range from the traditional Wi-Fi and IEEE 802.15.4 to new radio standards and even new modes of LTE cellular
  - As you can tell, the emphasis is on wireless communications

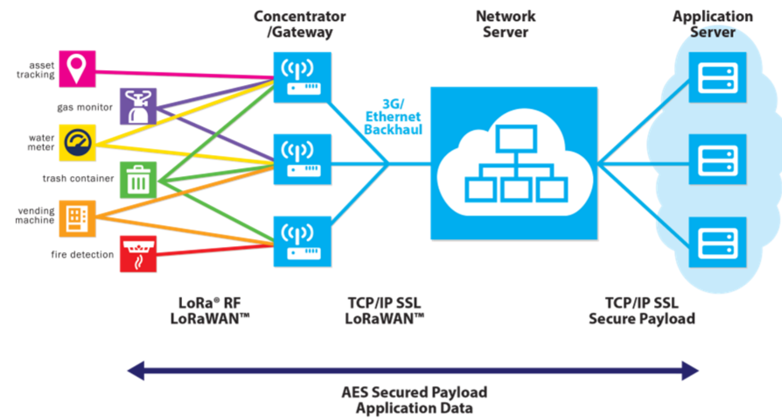
# Wireless Standards – Wi-Fi HaLow

- We're familiar with the traditional Wi-Fi IEEE 802.11abgn/ac flavors
  - Ranges from 11 Mbps to 1 Gbps
  - However, these are notoriously power hungry
- The new IEEE 802.11ah (a.k.a., Wi-Fi HaLow) provides support for sub-GHz, low-power Wi-Fi
  - Ranges up 1 km and thousands of nodes connected to the AP
- Special APs will relay between HaLow and normal Wi-Fi
- IP-based communications @ 20-40 Mbps

Source: wifi-alliance.org

# Wireless Standards – LoRaWAN

- New, sub-GHz star-of-stars topology with E2E AES-128 encrypted links
  - EU 868, EU 433, US 915, AS 430 bands
- Based on proprietary radio technology from Semtech, Inc.
- Symmetric link speeds
  - But, data rates are < 100kbps
    - Typically, 38.4Kbps
- Range is ~2km in urban and 22km in rural applications
- Not IP based
  - Depends on concentrators to relay with IP-based networks




Source: semtech.com

# Wireless Standards -- SigFox

- SigFox is a proprietary cellular-like communications service in the sub-GHz band
- Targets really low-throughput devices like remote sensors
  - Up to 140 messages/day
  - Payload is 12 bytes
  - Throughput is 100 bits/second
- Range is ~10km in urban or ~50km in rural applications
- Very low power consumption
- Requires a gateway to get to IP-based devices

Source: twitter.com

# Wireless Standards – IEEE 802.15.4

- IEEE 802.15.4 is available in multiple radio frequencies including 2.4 GHz and sub-GHz bands
- 802.15.4 really only defines to L2
  - Suppliers like ZigBee, Z-Wave and Thread Alliance supply L3-L7
- ZigBee IP and Thread's 6LoWPAN are IPv6 based
- Other 802.15.4 suppliers use proprietary protocols
  - They look like UARTs to the code

# Wireless Standards – LTE Evolution

- The cellular carriers want in on the action of the IoT
  - However, their emphasis has been on very high data rates that aren't typically needed in IoT applications
- LTE has 3 new flavors targeting LPWAN applications
  - LTE Cat.1 (<10Mbps(DL) and < 5Mbps (UL))
  - LTE Cat.M1 (< 1Mbps (DL/UL))
  - LTE Cat.NB1 (< 170Kbps (DL) and < 250 Kbps (UL))
- These work just like normal cellular except that the data rates are limited to help preserve battery life
  - Supports IPv4/IPv6
- These will typically be billed on data usage


Source: nimbelink.com

# Wireless Standard -- Bluetooth™

- Bluetooth has been a long time standard for use in PAN connectivity in the 2.4 GHz band
  - Limited range ( <30m) can be a problem
- Comes in Boothtooth Classic and Bluetooth Smart (BLE) varieties
  - Classic targets bi-directional communications (< 1Mbps) in the serial profile and requires pairing
  - BLE is more focused on uplink traffic and does not require pairing
- Either could run IP via PPP, but Classic is better targeted at IP because of its connection-orientation

Source: logodatabases.com

# IP or Not IP?

- Most of us in this room are familiar and comfortable with IP-based communications
  - TCP/UDP for communications and TLS/DTLS or IPsec/VPNs for security
- However, many of the wireless standards do not support IP
  - We need to consider alternative messaging protocols if we are to use these other wireless connectivity types
- Fortunately, there are messaging approaches that can lend themselves to both IP and non-IP communications channels

Source cafepress.com

# Messaging Patterns

- In the IoT, the communications patterns tend to fall into one of just a few models
- Publish/Subscribe (pub/sub)
  - Sensors publish their data to a centralized server and the server distributes that data to those who subscribe to the data
    - MQTT is an example of this pattern
- Client/Server
  - This pattern is more of a traditional send the data to the server and hope that the server knows what to do with it
    - RESTful and CoAP are examples of this
- Peer-to-peer (P2P)
  - This is direct messaging between the source and sink of the data
    - XMPP can use this model

# Messaging Protocols -- MQTT

- Message Queue Telemetry Transport was originally developed by IBM in 1999
  - It is now an ISO standard (ISO/IEC PRF 20922) as well as an OASIS standard
- Designed for lightweight messaging that rides on top of IP protocols
- Uses a pub/sub messaging model that requires a broker/server for message distribution
- No particular format required for the payloads although the messaging methods are well defined

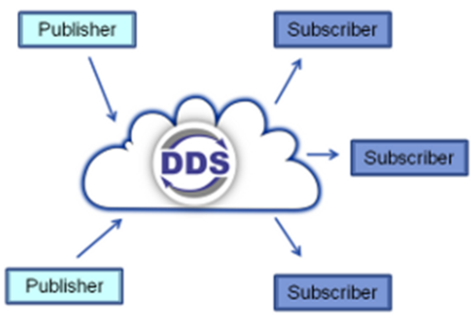
Source: mqtt.org

# Messaging Protocols -- MQTT (2)

- Methods include:
  - Connect, Disconnect, Subscribe, UnSubscribe and Publish
- Used by IBM Bluemix and Amazon IoT platforms among others
- Most IoT frameworks have support for MQTT
- Can run easily on small uCs
- Several open-source implementations of the message brokers including Eclipse's Mosquito, OpenStack and MyQtt

# Messaging Protocols -- DDS

- The Data Distribution Service in an Object Management Group M2M standard
  - Aims at real-time, dependable message exchange




Source: twinoakscomputing.com

- Originally designed in the 1990s as a distributed simulation standard, it is now used in many Government-related projects owing to its reliability
- This uses pub/sub, but does not use a message broker
  - It uses IP multicast

# Messaging Protocols – DDS (2)

- DDS has two levels of interfaces:
  - The lower data-centric publish-subscribe (DCPS) ensures delivery
    - Has broadcast, send w/ acknowledge and other modes
  - The optional higher-level data local reconstruction layer (DLRL) is an application layer integration
- DDS for Lightweight CORBA Component Model (CCM) is focused on business model integration
- Support for UML profile and platform-specific modeling
  - Support for Java, C/C++, Python, Lua, Ada, Pharo, Ruby and more APIs as well as access to CCM QoS profiles
- The open-source OpenDDS implementation is available

# Messaging Protocols -- XMPP

- Extensible Messaging and Presence Protocol is the protocol used by Jabber and Facebook messaging
  - Described in numerous RFCs
- Messages are in XML and can be sent using TCP or HTTP transports
- XMPP can be used in client-server, pub/sub or P2P models
- There are multiple open source implementations

XMPP

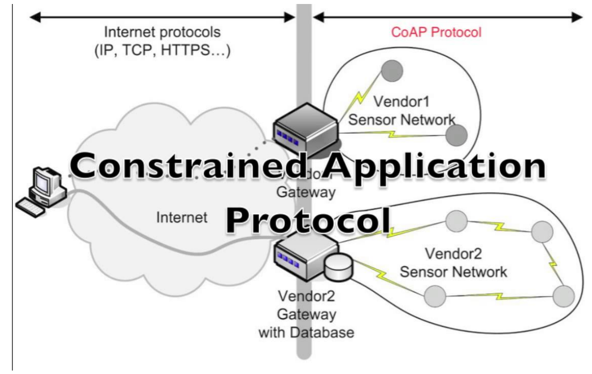
Source: wikipedia.org

# Messaging Protocols -- REST

- Representational State Transfer is a protocol that uses HTTP verbs (GET/POST/PUT/DELETE, etc.) for message transfer and storage
  - Also known as RESTful Web Services
- Primarily targets the client/server model
  - Allows access and manipulation of web resources using a URI and implementations in XML, HTTP, JSON and others
- Any implementation that uses HTTP for data transfer and storage can be said to use REST
  - As such, there are multiple open-source implementations

# Messaging Protocols -- CoAP

- Constrained Application Protocol is an application layer intended for use in constrained resource devices
  - Essentially, it is a binary version of REST that can be translated into HTTP semantics
- Supports multicast and has very low overhead using a UDP-based transport mechanism
  - Security provided via DTLS and is compatible with 6LoWPAN
- Has support for resource discovery
- Simple subscription for a resource with resulting push notifications
  - Can also be used in client/server or P2P modes




Source: youtube.com

# Messaging Protocols -- Proprietary

- There is no shortage of proprietary protocols in use in IoT frameworks
  - Often derived from pre-existing serial formats that predate IP
- ZigBee, Z-Wave, Wireless HART and others all have proprietary implementations
  - You must be a member of the respective alliance to gain access
- No open-source implementation of ZigBee, Z-Wave or Wireless HART is currently available ☹

ZigBee®

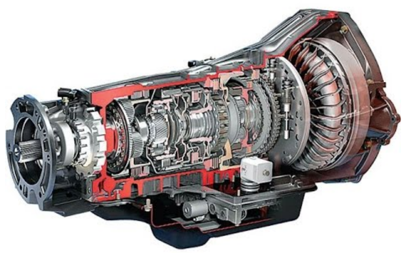
Source: zigbee.org

# Lack of IP Limits Options

- The major proprietary protocols do not use any IP-related transport
  - This means that the local network segment must interface with a gateway to convert the data to IP using one of the established protocols like REST or MQTT
- This limits your options on the messaging protocols and complicates debugging because you can’t use tools like WireShark for monitoring

# Transmission Issues

- The cellular carriers prefer that you use REST and XMPP for messaging
  - They really seem to like you using XML, JSON or HTTP oriented messaging
- This makes perfect sense when you consider that they make money from every single byte you transfer across their system
  - Verbose protocols like XML and JSON send a *lot* of data in a single transaction = more money for the carrier
- If you prefer to think in HTTP verbs, then consider using protocols like CoAP
  - HTTP verbs in binary



Source: youtube.com

# Cyber Security Issues

- Regardless of your application, you cannot ignore cyber security these days
- Lots of bad actors out there to cause trouble
  - Like the DDoS from IoT devices against DNS servers last October
- At a minimum, encrypt the links
  - Using the radio for link encryption or via TLS/DTLS for E2E encryption
- Use code signing and certificates to verify source of updates and identities of devices
  - Provisioning 1000s of devices will be an issue
- The fog model is easier to secure than the cloud model
  - You limit the attack surface

# Which Messaging API to Use?

- It depends on your device and application
- If you're looking for the broadest support, then use MQTT
  - Most of the major IoT frameworks support it
  - Some pub/sub approaches can be confusing because of the requirement for a broker
- If you want an web-like model, then use CoAP on the device and REST for transfers from the border routers to the cloud
  - Remember to use secure links across the cloud infrastructure
- There are a lot of wireless options, most support IP
  - So, most of the message middleware will work fine

Source: aaronroth.net

# Summary

- The IoT/IIoT has no shortage of offerings in the way of options
- Standards such as MQTT, DDS, REST, XMPP provide some hope for inter-operability
  - Wireless standards such as BLE, Wi-Fi and IEEE 802.15.4 help deal with physical connectivity
- Use of proprietary protocols or wireless solutions will work, but probably with vendor lock-in
- Consider attack surfaces, open-source availability and transmission costs in your messaging decision making process