IOT and System Platform
From Concepts to Code

Andy Robinson
Avid Solutions

avid solutions
Trusted Partners. Engaged Experts.
Introductions
Introductions

• Andy Robinson
• Information Solutions Consultant with Avid Solutions in Raleigh
• 2 years as Principal at Phase 2 Automation in Taiwan
• 10 years with Avid Solutions
• 5 years with Monsanto/Solutia in Decatur, AL and Houston, TX
• Co-Founder of aaOpenSource
• The Archestranaut
Overview
Overview/One-Eyed People

I am NOT an expert

In the land of the blind

...the one eyed man is king
What would/could you do differently if the cost to send data to/from your system approached $0?
Overview /Parts and Pieces

• Many Moving Pieces
Overview/Data Transmission

• Two Core Pieces
  • Transport
    • TCP – connection oriented, packets are confirmed and retransmitted if failed
    • UDP - connectionless, fire and forget
  • Protocol
    • HTTP
    • XMPP
    • CoAP
    • AMQP
    • MQTT
Overview/Data Transmission/HTTP

• Same core protocol as web page delivery

• Uses a RESTful pattern (GET/POST)

• Client-Server with Request/Response
Benefits

- Easy path through firewalls (HTTP/80, HTTPS/443)
- Out of the box transport security with SSL/TLS
- Very well known so easy to get started
Overview/Data Transmission/HTTP

• Drawbacks
  • Huge overhead from headers 700-800 bytes\(^1\) (should be better with HTTP/2)
  • RESTful patterns are inherently slow and require polling
  • Websockets

(1) http://dev.chromium.org/spdy/spdy-whitepaper
Overview/Data Transmission/XMPP

- eXtensible Messaging and Presence Protocol
- XML over TCP
- Originally started as a chat protocol to support contact lists and presence
- Ask Eliot Landrum anything

https://en.wikipedia.org/wiki/XMPP
Overview/Data Transmission/XMPP

- Benefits
  - Open standard
  - Flexible
  - Very mature (originated in 1999)

https://en.wikipedia.org/wiki/XMPP
Overview/Data Transmission/XMPP

• Drawbacks
  • No QOS inherent at the protocol layer
  • Higher network overhead due to XML being text based

https://en.wikipedia.org/wiki/XMPP
Overview/Data Transmission/CoAP

• Constrained Access Protocol

• UDP but similar to HTTP with RESTful functions (GET/PUT/POST/DELETE)

• Asynchronous communications model

• Client-Server

Overview/Data Transmission/CoAP

• Benefits
  • Ultra small header (4 bytes) leads to ultra small packet sizes
  • DTLS for security
  • Combined with LWM2M for robust device management
  • Discovery methods built into protocol
  • Great for ultra low power, ultra low bandwidth
  • Datagram packet model can work over non IP transports like SMS
  • IETF Standard

Overview/Data Transmission/CoAP

• **Drawbacks**
  
  • Core protocol is client-server with server initiated communications → requires inbound packets to client device → security concerns

  • As single client to single server there is no built-in concept of data broadcast
Overview/Data Transmission/AMQP

• Advanced Message Queue Protocol

• Originally developed by big financial orgs for middleware messaging

• Message oriented binary protocol typically over TCP

• OASIS Standard

Overview/Data Transmission/AMQP

• Benefits
  • Built for resiliency and scaling
  • Load balancing patterns
  • Messages can contain payload and metadata
  • Powerful routing patterns
  • Security based on SSL/TLS
Overview/Data Transmission/AMQP

• Drawbacks
  • Not lightweight – built with servers and fat networks in mind (60 byte minimum packet size)
  
  • Can be complex to implement client – thanks Paolo for SBLite
Overview/Data Transmission/MQTT

• Message Queue Telemetry Transport

• Created by IBM to support oil pipeline telemetry data over slow, unreliable networks

• TCP Transport

• Publish-Subscribe with Broker Model

Overview/Data Transmission/MQTT

• Benefits
  • Very small header (7 bytes) leads to very small packet sizes
  • TLS for security
  • Great for ultra low power, ultra low bandwidth
  • No incoming connection to client → much more secure
  • 3 levels of QOS
  • OASIS Standard

Overview/Data Transmission/MQTT

• Benefits - Small size and low overhead

Stephan Nicholas did a fascinating apples to apples comparison of MQTT vs HTTPS on Android, 3G and WiFi which you can read here. The 3G results are quite interesting:

- 93x faster throughput
- 11.89x less battery to send
- 170.9x less battery to receive
- 1/2 as much power to keep connection open
- 8x less network overhead

https://mobilebit.wordpress.com/2013/05/03/rest-is-for-sleeping-mqtt-is-for-mobile/
Overview/Data Transmission/MQTT

• **Drawbacks**
  
  • TCP is connection oriented so more complex to put device to sleep (MQTT-S supports UDP)
  
  • No good model for metadata and discoverability
  
  • No standard method for device management

• And the best protocol is...the one that matches your requirements and capabilities

• One of many comparisons
  • www.slideshare.net/paolopat/mqtt-iot-protocols-comparison

• ..... but let’s talk about MQTT
MQTT – The Details
MQTT/Details/Transport

• Utilizes single, duplex, persistent TCP connection for transport
• Client connects to broker and broker sends data back down the connection
• Broker NEVER connects directly to client. Security Win!
• Client can use SSL/TLS to connect to Broker. Security Win!
• Many brokers support authentication and ACL
MQTT/Details/Communication Model

• Pub/Sub with Central Broker

http://www.codeproject.com/KB/IP/PubSubUsingWCF/pubsub.png

• Conceptually similar to Mxaccess in System Platform
MQTT/Details/Broker

- Broker is meeting point for publishers and subscribers
- Many different brokers available
- Run locally or in the cloud
MQTT/Details/Broker

- Local
  - mosquitto – The old man of the brokers, full featured, easy, single exe
  - GnatMQ – written in C#, open source
  - HiveMQ – extra features, commercial, extensible
  - Mosca – node.js, open source
  - Verne.MQ – written in Erlang, scalable and resilient, open source
MQTT/Details/Broker

• Cloud
  • CloudMQTT – uses mosquitto

  • Verne.MQ - extensible, highly available

• Public testers – test.mosquito.org, broker.mqtt-dashboard.com, ...
MQTT/Details/Topics

- Mailboxes organized into folders
- “/” separates the folders
- Wildcards
  - Single Level → “+”
    - home/+/temp
      - home/1/temp
      - home/2/temp
  - Multi-Level → “#”
    - /home/#
      - home/1/temp
      - home/2/temp
      - home/1/rh
      - home/2/rh
### MQTT/Details/Topics vs Queues

<table>
<thead>
<tr>
<th>Feature</th>
<th>Topics</th>
<th>Queues</th>
</tr>
</thead>
<tbody>
<tr>
<td>Big Idea</td>
<td>Real Time</td>
<td>Asynchronous FIFO/Stack</td>
</tr>
<tr>
<td>Dropped Data</td>
<td>Can Be OK (but QOS)</td>
<td>Never Ok</td>
</tr>
<tr>
<td>Pub/Sub</td>
<td>Yes</td>
<td>Partial (only one client)</td>
</tr>
<tr>
<td>Applications</td>
<td>Sensor Data, Alarms</td>
<td>Alarms, Events, Value History</td>
</tr>
</tbody>
</table>

#### Diagram:
- **Publish**
  - "Client"
  - Value
  - Alarms
  - Events
- **Broker**
  - Topics
- **Subscribe**
  - Topic Values
  - Alarms
  - Events
  - HMI
  - Historical Values, Alarms, Events
MQTT – The Code
MQTT/Code/Client Library

- You only need a client library to pub and sub .. unless you want to run a broker too

- Client libraries for almost every language

- Best C# library is M2MQTT from Paolo Patierno
  - Open Source
  - Full Featured
  - Nuget Package Available
Create Client
Set Options (ClientID, SSL/TLS?, QOS, Credentials)
Connect to Single Broker by Name or IP Address

......
Subscribe to Topic(s) → Receive Callback for Subscribed Topic

....
Publish UTF-8 encoded binary data to Topic(s)
... Declarations....
configHost = "localhost";
configclientID = Me(Tagname);

MQTTClient = new uPLibrary.Networking.M2Mqtt.MqttClient(configHost);
MQTTClient.Connect(configclientID);

if (MQTTClient.IsConnected) then
    Topic = "data/" + Me.Tagname + "/value";
endif;
MQTT/Code/System Platform/Subscribe

- Not possible due to requirement for callback registration

```csharp
_mqttClient.MqttMsgPublishReceived += _mqttClient_MqttMsgPublishReceived;
_mqttclient.subscribe(Topic)
...

void _mqttClient_MqttMsgPublishReceived(object sender, MqttMsgPublishEventArgs e)
{
    Process new message e
}
```

- But you can use the AOT – been there done that.. mostly
MQTT – The Demos
MQTT/Demo

• Demo 1 – Publishing simple data from System Platform
MQTT/Demo

• Demo 2 – Publishing complex data from System Platform
MQTT/Demo

• Demo 3 – Subscribing to Data “in” System Platform
MQTT – Resources
MQTT/Resources

- MQTT.org
- Paolo! - m2mqtt.wordpress.com/
- Eclipse paho- www.eclipse.org/paho/

- Brokers
  - HiveMQ
  - Verne.MQ
  - CloudMQTT

- AMQP on Microsoft Azure - github.com/ppatierno/azuresblite
Wrap-Up
Wrap-Up/Takeaways

• Just a seed

• No solutions, just ideas

• Take these concepts and scale up!

• We are all now composers, not luthiers
Wrap-Up/Contact

• Andy Robinson

aroobinson@avid solutionsinc.com
@archestranaut

@aaOpenSource
github.com/aaopensource