

# IoT-ALE: Connecting to the Internet MQTT

putting the I in IoT

John 'Warthog9' Hawley

SCaLE 17x - March 2019

Let us lay some ground works...

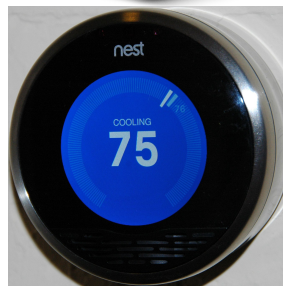
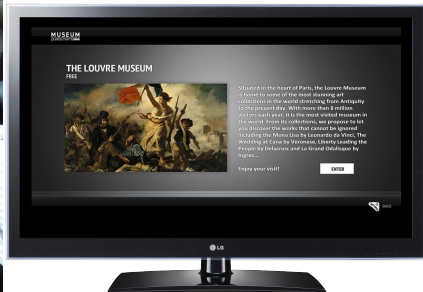
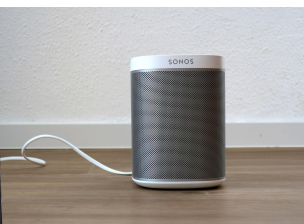
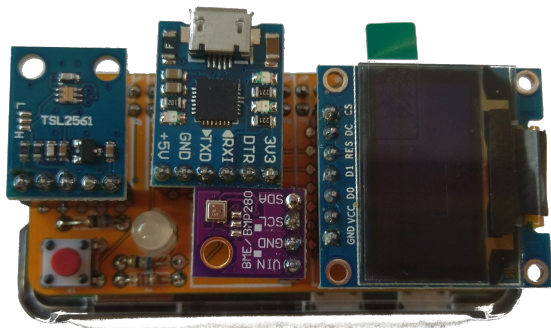
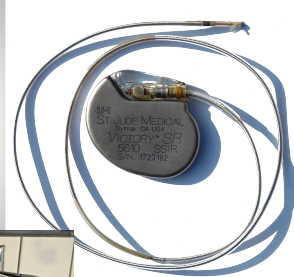
What most “home” networks look like:

# Firewall

Main  
Network

Wireless  
Guest

# More Groundwork: IoT devices



# Typical ways devices connect to the Internet

- Through a Gateway:
  - Bluetooth
  - Z-wave
  - 802.11.6
  - Zigbee
  - IR
  - Smoke Signals
  - Carrier Pigeons
- Directly:
  - Wifi
  - Ethernet
- Using:
  - IPv4
  - IPv6

Lets come back to this for a minute to talk about  
IPv4 vs. IPv6

# Firewall

Main  
Network

Wireless  
Guest

# Local Access vs. Remote Access

- IPv4 - Local

- Direct Access
- Straight Forward
- Mostly ubiquitous support

- IPv4 - Remote

- NAT traversal
- Punching holes in firewalls
- Port Forwarding
- UPNP
- Cloud reverse proxies

- IPv6 - Local

- Direct Access
- Straight Forward
- Getting more ubiquitous but not there

- IPv6 - Remote

- Direct Access
- Punching holes in firewalls
- UPNP
- Cloud based IP lookup (and/or reverse proxies)

# Some general words of caution...

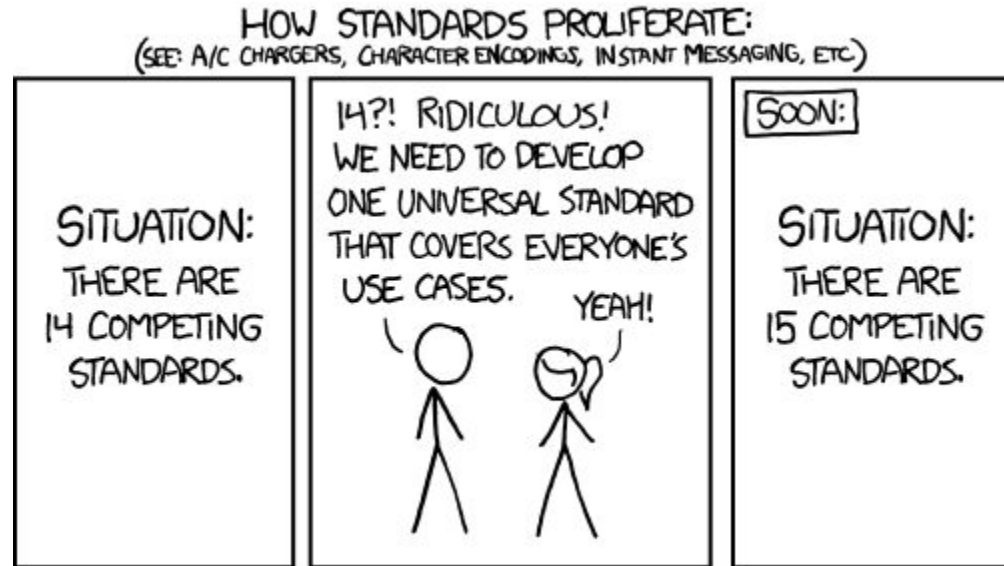
- Think about what you are using the Internet for
- Be mindful of where your services live
- Sometimes UX the user can use may make you less secure
- Always change the default passwords!
- Make it possible to do things without auto-discovery
- Don't always assume you are on the same network as the device
- Upgrade schemes need to be done

# Shifting gears & talk about how to talk to the devices

But the real advantage to IoT is the I - Internet!

Lots of good ways to do this...

- MQTT
- Liota
- AMQP
- STOMP
- RabbitMQ
- REST
- WAMP
- ZeroMQ
- Java Message Service (JMS)
- CoAP
- CLOUD!
- XMPP-IOT
- XMPP
- etc.....





# Now lets talk about something to try

- MQTT - Mosquitto, MQTT broker, good for local passing of data
- Think of it as a message bus on the network
- Clients Subscribe to Topics that can be hierarchical, and listen to the Topic
  - /myhome/groundfloor/livingroom/temperature for example
  - You can listen at any level of the hierarchy, anything below your level will be filtered to you
  - Wildcards, +, are allowed /myhome/+/temperature
- Devices Publish data to topics
  - The data is freeform, the receiving end is expected to interpret it

# Lets just try listening...

On your laptop/VM:

```
yum install mosquitto
```

```
apt-get install mosquitto-clients
```

then

```
mosquitto_sub \  
  -h 10.100.0.5 \  
  -t "pugnose/temp/core0" \  
  -u "ale" \  
  -P "Penguins"
```

**Expected output:**

+67.0°C

What's running on "pugnose":

```
while [[ 1 ]];do \  
  mosquitto_pub \  
    -h 10.100.0.5 \  
    -t "pugnose/temp/core0" \  
    -m "$( \  
      sensors | \  
      grep "Core 0" | \  
      tr " " "\n" | \  
      grep "°" | \  
      head -n 1 \  
    )" \  
    -u "ale" \  
    -P "Penguins"; \  
  sleep 10;\  
done
```

# Listening from the IoT device (subscribing)

## From the repl prompt:

```
>>> from umqtt.simple import MQTTClient
>>> import socket
>>> import time
>>> from ubinascii import hexlify
>>> CLIENT_ID = hexlify(machine.unique_id())
>>> def sub_cb(topic, msg):
...     print((topic, msg))
...
...
...
>>> c.set_callback(sub_cb)
>>> c = MQTTClient(CLIENT_ID,
... "10.100.0.5")
>>> c.connect()
>>> c.subscribe(b"topic/yourname")
```

```
>>> while True:
...     if True:
...         c.wait_msg()
...     else:
...         c.check_msg()
...         time.sleep(1)
...
...
>>> c.disconnect()
```

## From your VM / Laptop

```
mosquitto_pub \
    -h 10.100.0.5 \
    -t "topic/yourname" \
    -m "Hello YourName" \
    -u "ale" \
    -P "Penguins"
```

# Publishing from the IoT device

## From the repl prompt:

```
>>> from umqtt.simple import MQTTClient
>>> import socket
>>> from ubinascii import hexlify
>>> CLIENT_ID = hexlify(machine.unique_id())
>>> c = MQTTClient(CLIENT_ID,
... "10.100.0.5")
>>> c.connect()
>>> c.publish(b"topic/yourname",
... b"hello from mpy")
>>> c.disconnect()
```

## On your laptop/VM:

```
yum install mosquitto
```

```
apt-get install mosquitto-clients
```

then

```
mosquitto_sub \  
-h 10.100.0.5 \  
-t "topic/yourname" \  
-u "ale" \  
-P "Penguins"
```

# For the way advanced!

```
from umqtt.simple import MQTTClient
from machine import Pin
from ubinascii import hexlify
import machine
import micropython
led = Pin(0, Pin.OUT, value=1)
SERVER = "10.100.0.5"
CLIENT_ID = hexlify(machine.unique_id())
TOPIC = b"topic/yourname"
state = 0
def sub_cb(topic, msg):
    global state
    print((topic, msg))
    if msg == b"on":
        led.value(0)
        state = 1
    elif msg == b"off":
        led.value(1)
        state = 0
```

```
elif msg == b"toggle":
    led.value(state)
    state = 1 - state

def main(server=SERVER):
    c = MQTTClient(CLIENT_ID, server)
    c.set_callback(sub_cb)
    c.connect()
    c.subscribe(TOPIC)
    print("Connected %s, sub to %s topic"
          % (server, TOPIC))

    try:
        while 1:
            c.wait_msg()
    finally:
        c.disconnect()
```