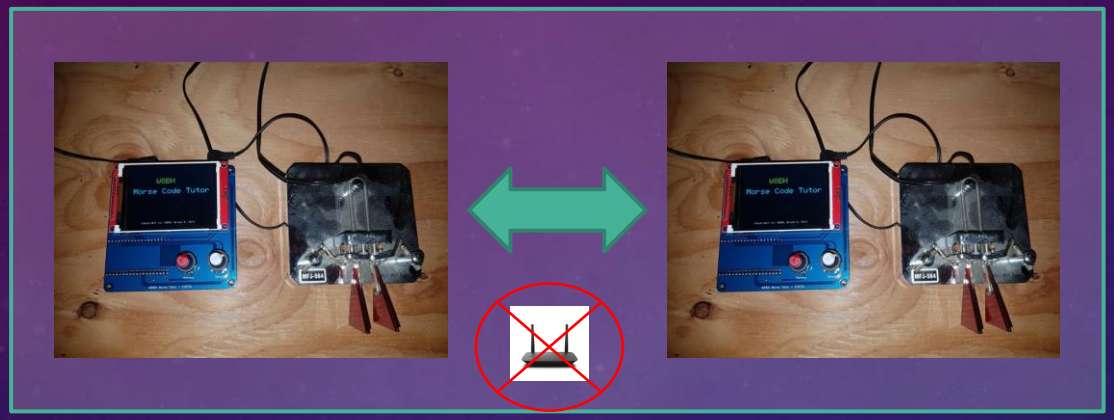
UPDATE TO W8BH'S MORSE CODE TUTOR

FIRMWARE UPDATED BY VE3OOI

AGENDA

- Background
  - Bruce Hall W9BH Morse Tutor Kits (ESP Version)
- Two Way Communication (Current)
  - ESP-NOW
- Proposed Two Way Communication
  - MQTT

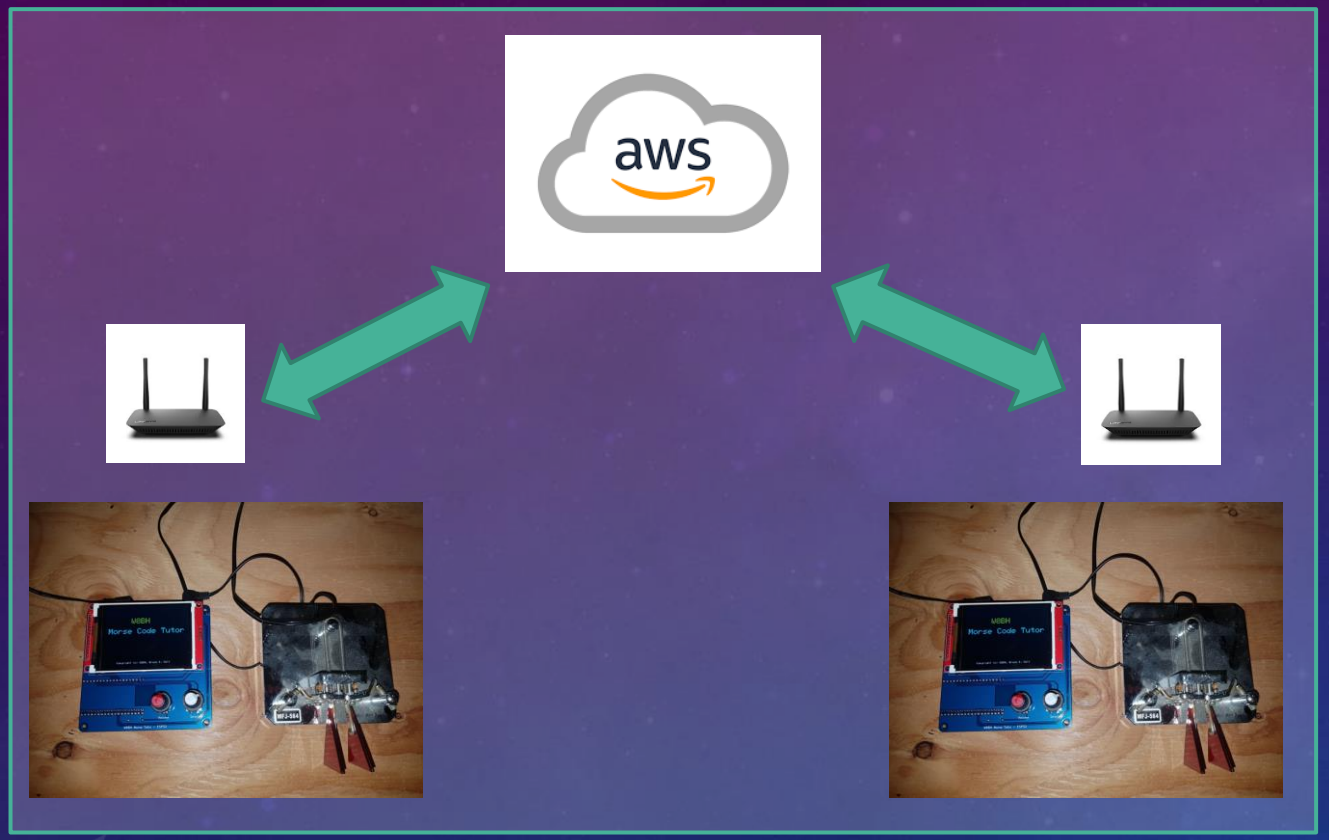
TWO WAY “ADHOC” COMMUNICATION (AS IS)



- Uses ESP_NOW library to perform communications between ESP32 Wi-Fi without an Access Point (i.e. no Wi-Fi Router). This is called ADHOC Network
- Limited to a few hundred feet range. i.e. W8BH kits must be close to each other
- Current default firmware is good for “classroom” setting. Not suitable for geographic separation (e.g. hundreds of Km away)
- See Bruce’s YouTube Video (https://www.youtube.com/watch?v=tp74gO6lAm0)

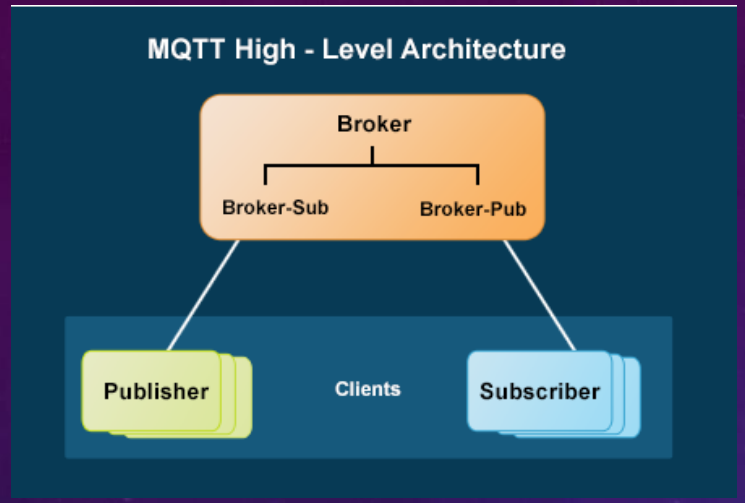
A **wireless ad hoc network**[1] (**WANET**) or **mobile ad hoc network** (**MANET**) is a decentralized type of wireless network. The network is ad hoc because it does not rely on a pre-existing infrastructure, such as routers in wired networks or access points in wireless networks. Instead, each node participates in routing by forwarding data for other nodes, so the determination of which nodes forward data is made dynamically on the basis of network connectivity and the routing algorithm in use.[2]

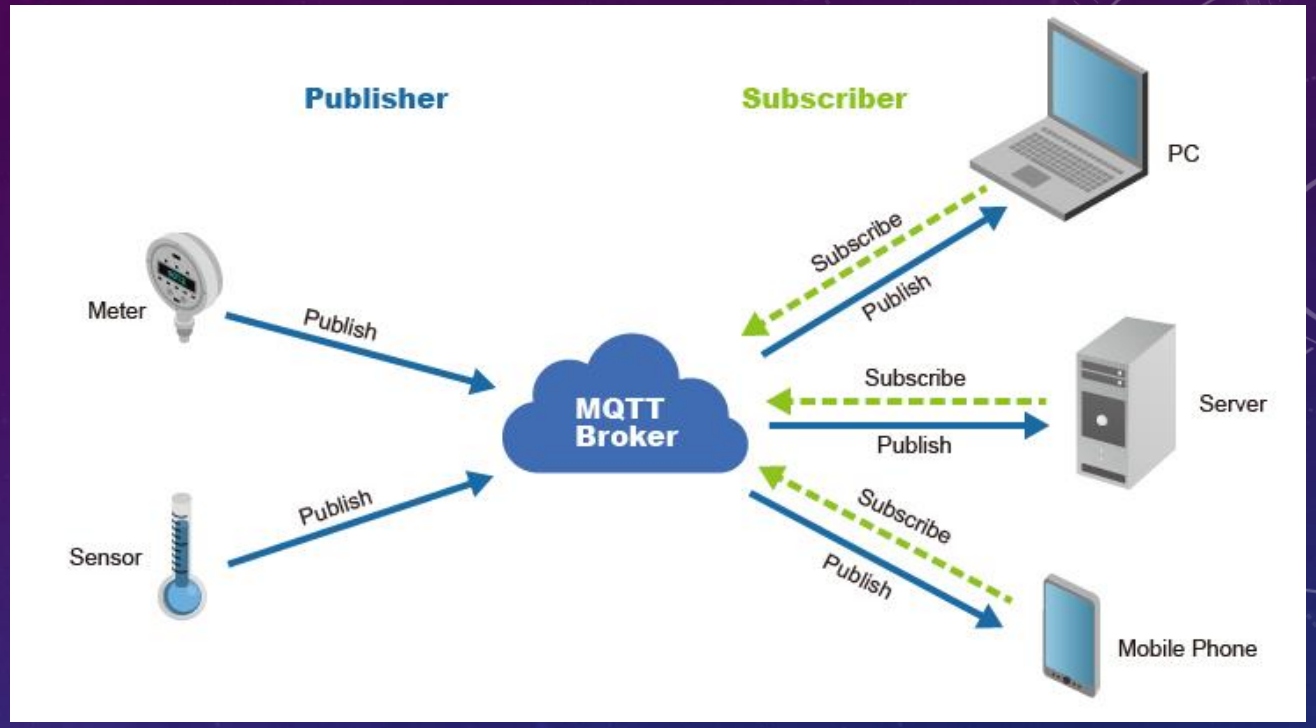
TWO WAY IP COMMUNICATION (VE3OOI FIRMWARE)



- ✓ Based on VE3OOI firmware
  - ➢ Original firmware written by Bruce Hall
  - ➢ Compiled using PlatformIO which is compatible with Arduino IDE. Must use latest IDE and libraries.
  - ➢ Fixed a few issues in the code
- ✓ Uses MQTT as the underlying messaging
- ✓ Leverage an internet server to facilitate communications between W8BH kits.
- ✓ Leverages Free AWS EC2 Linux platform configured for MQTT
- ✓ W8BH kits must be connected to an Access Point and to the Internet
  - ➢ Must use DHCP to feed gateway address and DNS server to ESP32 Wi-Fi
- ✓ Each participating W8BH kit MUST use a unique ID.
  - ➢ This is used to identify the station.
  - ➢ Uses a random 3 character string

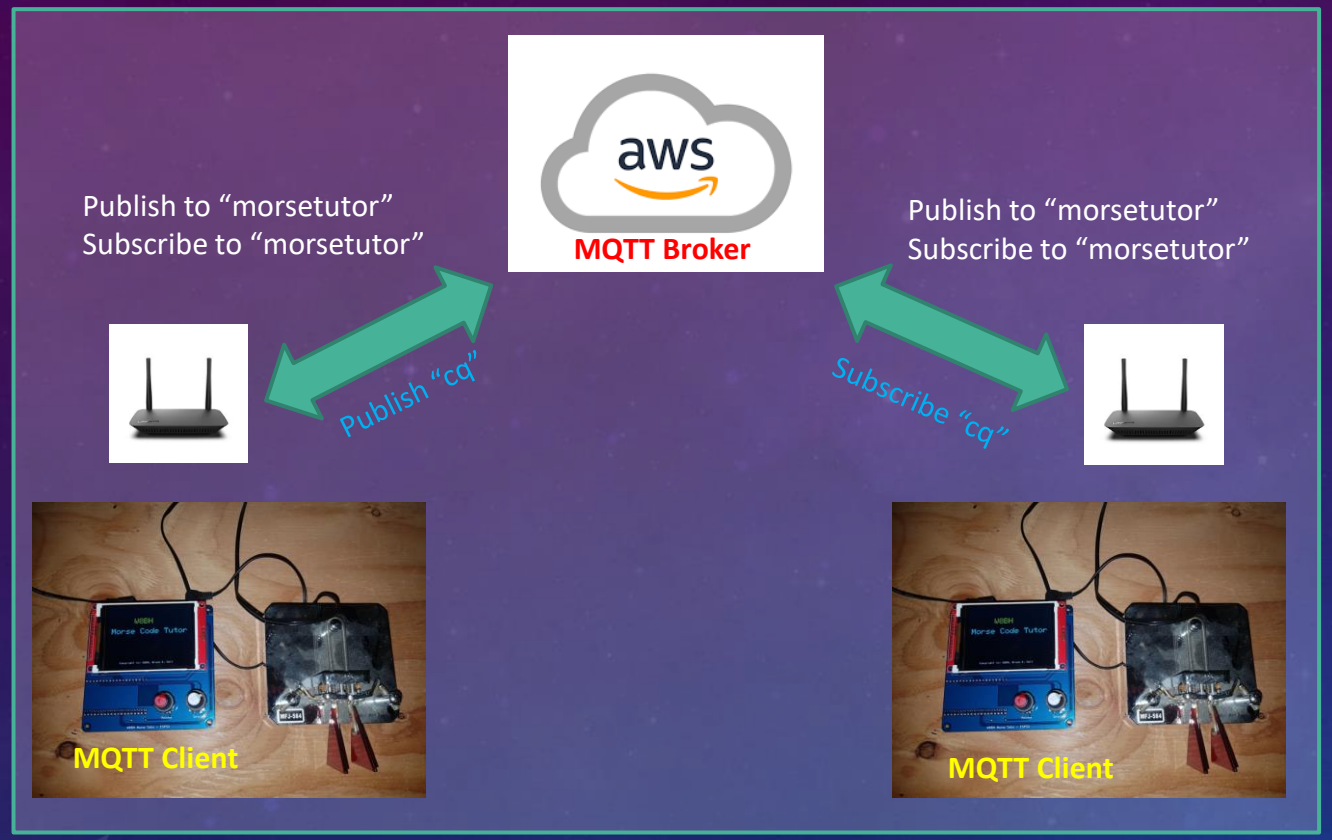
MQTT





- MQ Telemetry Transport
  - "MQ" came from the IBM 'MQSeries' product line and is "Message Queue“
- MQTT Broker and Client (Pub/Sub)
  - Publish to or Subscribe to a “Topic”
  - E.g. groups.io, twitter, discord
- MQTT uses text messages (anything). You parse it

MQTT & THE MORSE TUTOR



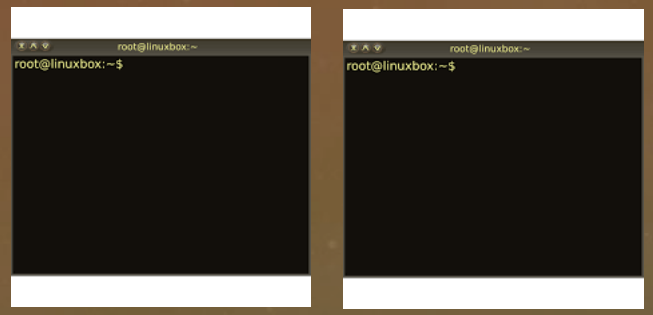
1. Client Connects to MQTT Broker
   - Uses DNS hardcoded name
2. Subscribes to a topic (I refer to it as a room)
3. Paddles/Key generates a character that is “published” to a topic
4. Every client that subscribes to the topic receives the message
   - Publisher ignores messages it sends

DEMO

Don’t have 2 W8BH Morse Tutors to Test

Subscribe to “morsetutor”

Publish to “morsetutor”



MQTT Client

MQTT Broker

Publish to “morsetutor”
Subscribe to “morsetutor”

Publish to “morsetutor”
Subscribe to “morsetutor”

MQTT Client

Outstanding (TODO):

1. Live test with another morse tutor
2. Add configuration parameters
3. Figure out distribution


Fin