Università
della
Svizzera
italiana

Facoltà
di
scienze
informatiche

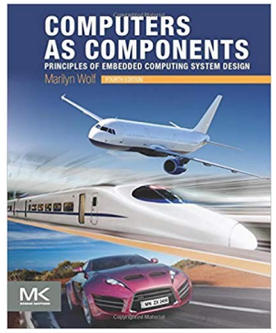
Edge Computing in the IoT

Networking

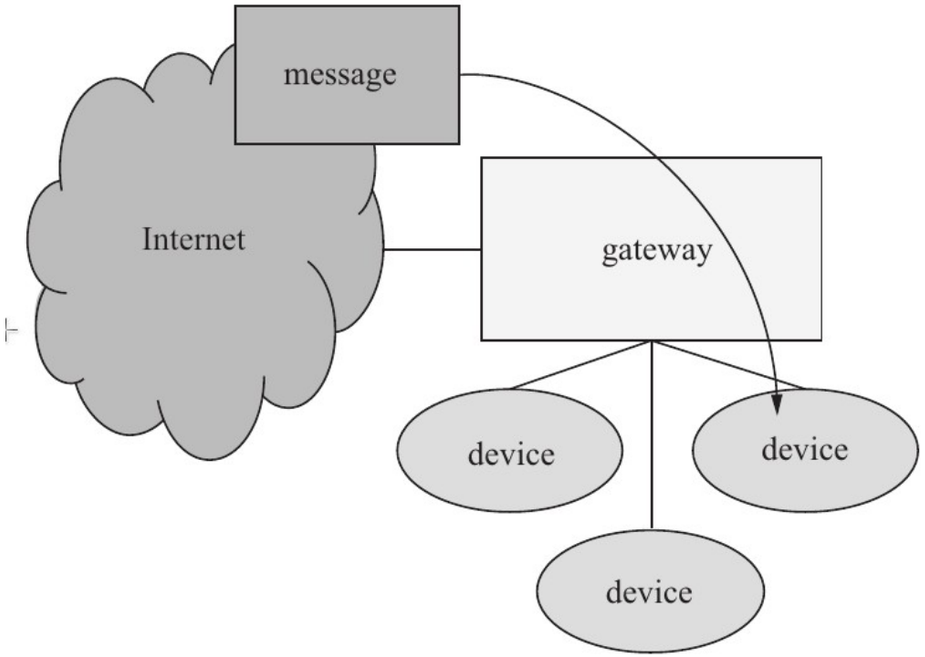
Alberto Ferrante (alberto.ferrante@usi.ch)

Networking for CPS and IoT devices

- In general, we consider the standard OSI model as a reference
- Heterogeneous solutions
  - Based on wired or wireless communication
  - (Mix of) different network protocols
  - Devices may or may not be directly connected to the Internet



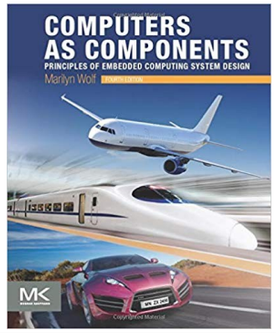
Network Architectures



- Many embedded devices cannot be directly connected to the Internet
- Devices communicate over non-IP networks called **edge networks**
  - A gateway connects edge networks to the Internet
- Ad hoc networks used as edge networks are usually created by self organization of a set of nodes
  - Nodes route messages without relying on additional networking equipment

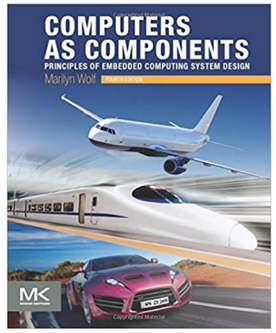
Edge Network Architectures

- We can evaluate a network on both its functional and nonfunctional characteristics:
  - Does it provide adequate security and privacy?
  - How much energy is required for communication?
    - Many IoT network devices are designed to operate from a button battery for an extended period: ultra-low energy
    - Key concern for some networks/devices
  - Cost for adding a device to the network?
- Some networks can support Quality of Service (QoS)

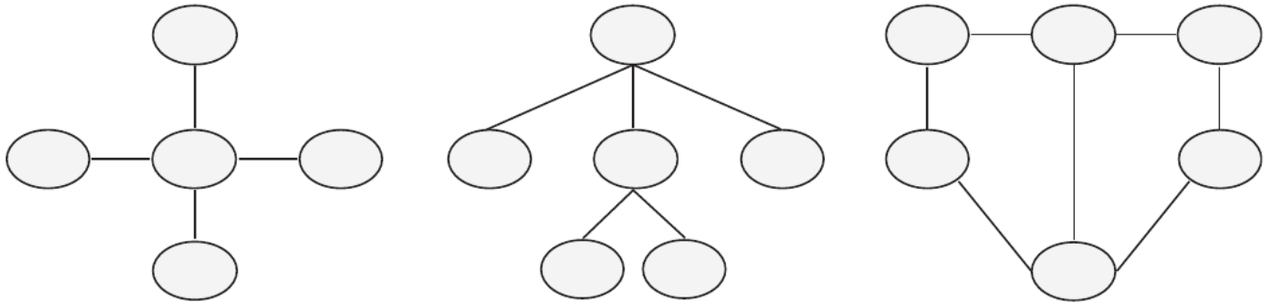


Edge Network – Synchronous/Asynchronous

- Many IoT networks support both synchronous and asynchronous communication
- Many wireless networks provide synchronous communication using beacons
  - A transmission from a node that marks the beginning of a communication interval
  - The time between beacons is usually divided into two segments, one for synchronous and the other for asynchronous communications

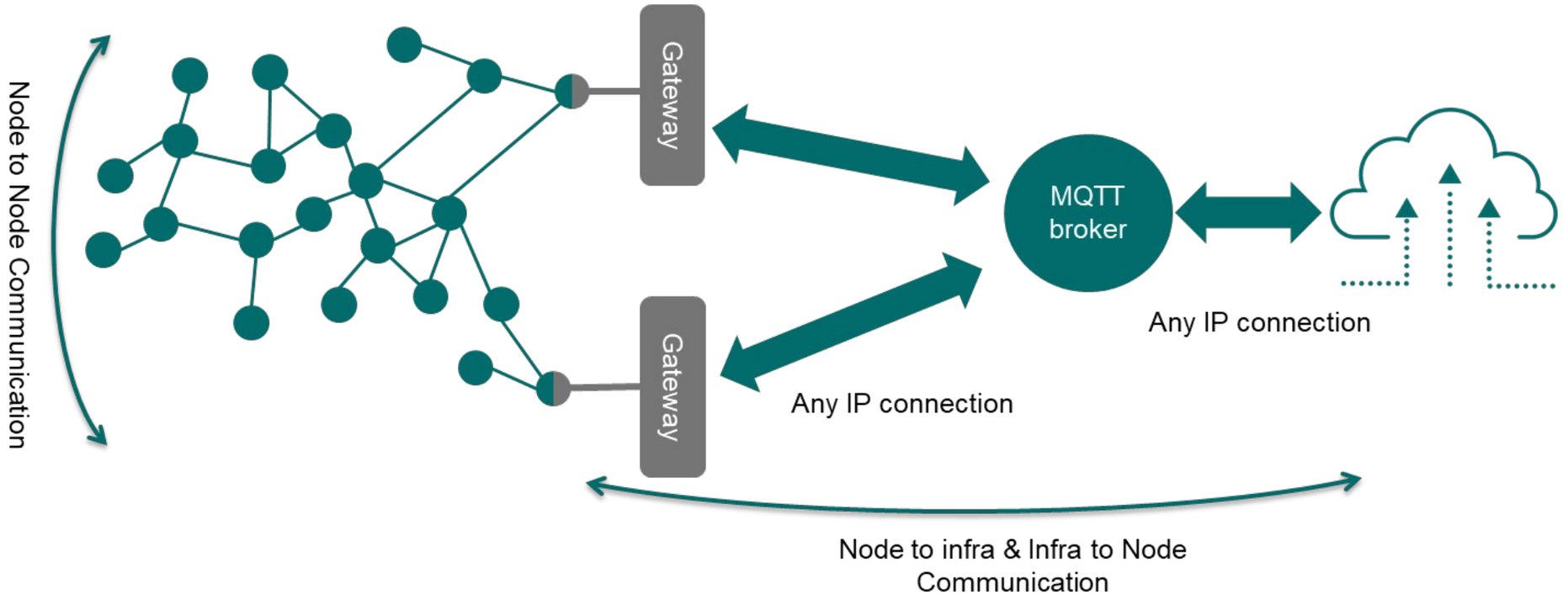


Edge Network Topologies



- The topology of a network describes the structure of communication within the network
  - The star network uses a central hub through which all other nodes communicate
  - A tree network provides a more complex structure but still only provides one path between a pair of nodes
  - A mesh network is a general structure where each node communicates with its neighbors and information is routed from one node to the other through the other nodes
- Routing discovery determines the routes that will be used by packets that travel from/to a node to/from others

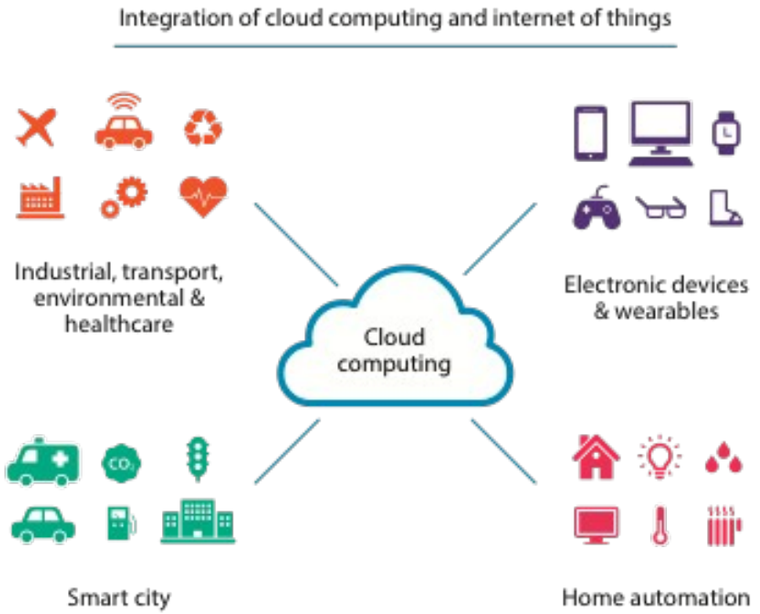
Network Topologies



https://staceyoniot.com/wirepas-is-a-mesh-network-built-for-scale/

Network Topology and Computation – Cloud Computing

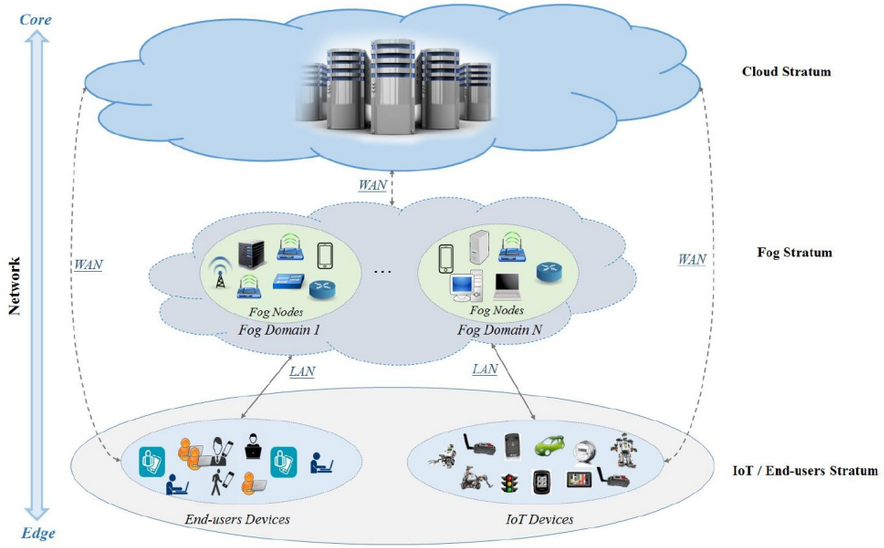
- Often, nodes are used only for collecting data or for actuation
  - Collected data are sent to remote servers (cloud) where computation happens



https://internetinitiative.ieee.org/newsletter/september-2018/integration-of-internet-of-things-iot-and-cloud-computing-privacy-concerns-and-possible-solutions

Network Topology and Computation – Fog Computing

- Fog refers to the network connections between edge devices and the cloud
- Fog computing extends traditional cloud computing to the edge of the network
  - The processing can take place at the edge of the network (**fog nodes**)
  - Other processing can happen in the cloud
- Low-latency, by allowing processing to take place at the network edge



https://arxiv.org/pdf/1710.11001.pdf

Network Topology and Computation – Edge Computing

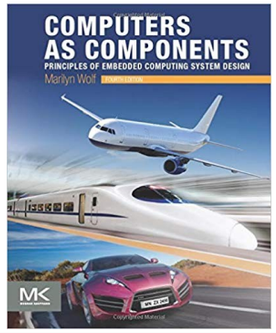
- Computation is moved into IoT nodes
- Fog includes edge computing, but fog would also incorporate the network needed to get processed data to its final destination

https://www.networkworld.com/article/3224893/what-is-edge-computing-and-how-it-s-changing-the-network.html

Network Protocols

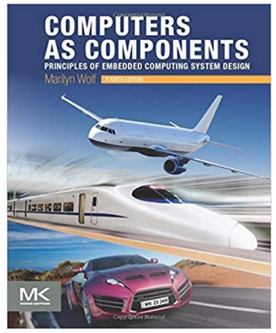
Bluetooth

- Bluetooth was introduced in 1999
- Originally for telephony applications such as wireless headsets for cell phones
- It is now used to connect a wide range of devices to host systems
- Designed to operate in a radio band known as the “instrumentation, scientific, and medical (ISM) band”
  - 2.4 GHz frequency range
  - No license required to operate in this band
    - Some restrictions on how it can be used, such as 1 MHz bandwidth channels and frequency-hopping spread spectrum
- Bluetooth networks are often called *piconets*, thanks to their small physical size



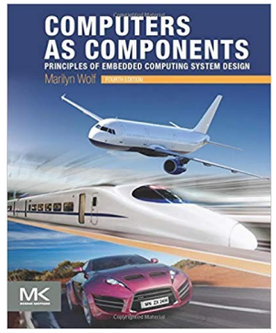
Bluetooth Stack

- The Bluetooth stack is divided into three groups:
  - Transport protocol
  - Middleware protocol
  - Application



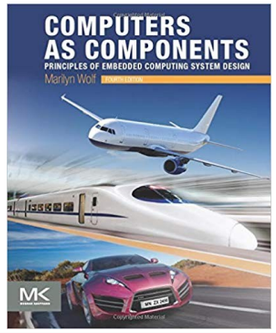
Bluetooth Transport Protocol

- The radio provides the physical data transport
- The baseband layer defines the Bluetooth air interface.
- The link manager performs device pairing, encryption, and negotiation of link properties
- The logical link control and adaptation protocol (L2CAP) layer
  - Provides a simplified abstraction of transport for higher levels
  - Breaks large packets into Bluetooth packets
  - Negotiates the quality of service required
  - Performs admission control



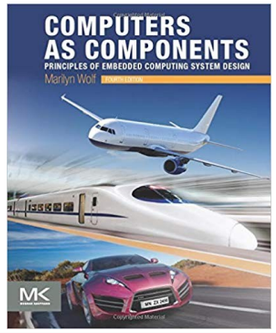
Bluetooth Middleware Group

- The RFCOMM layer provides a serial port style interface
- The service discovery protocol (SDP) provides a directory for network services
- The Internet Protocol and IP-oriented services such as TCP and UDP
- A variety of other protocols, such as IrDA for infrared and telephony control



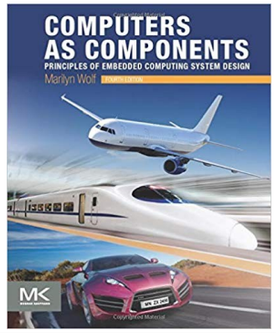
Bluetooth

- Every Bluetooth device is assigned a 48-bit Bluetooth Device Address
- Every Bluetooth device also has its own Bluetooth clock
  - Used to synchronize the radios on a piconet, as required for frequency-hopping spread spectrum communication
    - When a Bluetooth device becomes part of a piconet, it adjusts its operation to the clock of the master
- Transmissions on the network alternate between master and slave directions
- Two types of packets:
  - Synchronous connection-oriented (SCO) packets are used for quality-of-service-oriented traffic such as voice and audio
  - Asynchronous connectionless (ACL) packets are used for non-QoS traffic
- SCO traffic has higher priority than ACL traffic



Bluetooth Low Energy

- Bluetooth Low Energy is designed to support very low energy radio operation
  - A radio operated by a button-sized battery for an extended period is an example scenario of BLE usage
- BLE is part of the Bluetooth standard, but it differs in some fundamental ways from Classic Bluetooth
- BLE shares some features and components of Classic Bluetooth, such as the L2CAP layer

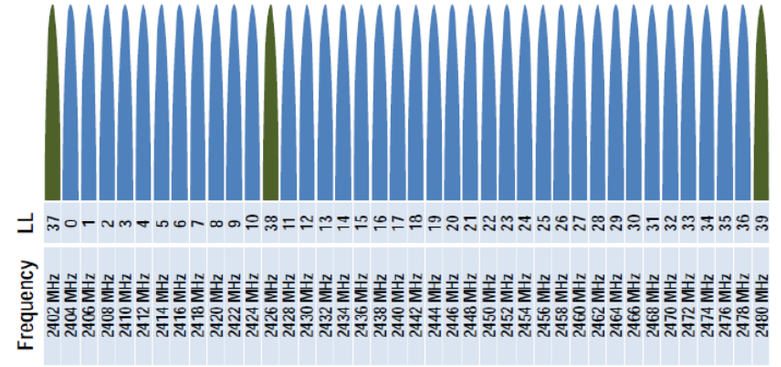


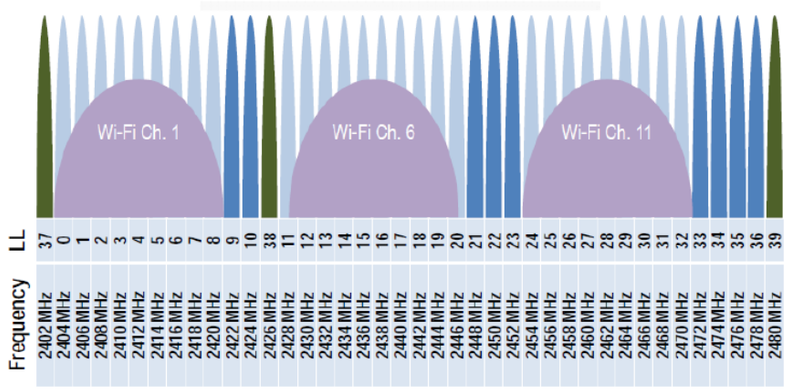
Bluetooth Low Energy

- Minimizing the amount of active time of the radio is critical to low energy operation
  - At the link level, packets are designed to be relatively small
  - BLE is designed to support communications that do not require long-lived connections
  - Advertising is one form of communication that is designed to support low energy
    - A device can transmit advertising packets; devices can also listen for advertising packets
    - Advertising can be used to discover devices or to broadcast information
    - Some short communications may be possible entirely through advertising

BLE Physical Layer

- 2.4GHz ISM Band
- Adaptive Frequency Hopping (AFH)
  - Reliable
  - Robust
  - Adapts to interference
- 40 channels
  - 3 advertisement channels
  - 37 data channels
- 1 Mbps bandwidth
  - Typical throughput <= 100kbps due to small packets
- TX power Limited by CE and FCC regulations
- Range
  - 0 – 500 meters
  - Typically 0-50 meters to a smart phone





https://www.silabs.com/documents/public/user-guides/ug103-14-fundamentals-ble.pdf

BLE Link Layer

- Provides the first level of control and data structure over the raw radio operations and bit stream transmission and reception
- The link layer defines the following:
  - Bluetooth state machine and state transitions
  - Data and advertisement packet formats
  - Link Layer operations
  - Connections, packet timings, retransmissions
  - Link layer level security

https://www.silabs.com/documents/public/user-guides/ug103-14-fundamentals-ble.pdf

BLE Link Layer Operations: Advertisement

- One of the most important operations in BLE
- Provides a way for devices to
  - Broadcast their presence
  - Allow connections to be established
  - Broadcast data
    - e.g., the list of supported services, or the device name and TX power level
- A BLE device can broadcast packets on one or multiple advertisement channels through advertising

https://www.silabs.com/documents/public/user-guides/ug103-14-fundamentals-ble.pdf

BLE Link Layer Operations: Scanning

- Scanning is the operation where a scanner is listening for incoming advertisement in order to
  - Discover
  - Discover and connect
  - Receive the data broadcast by the advertising devices
- Two types of scanning modes are supported
  - Passive scanning: the scanner simply listens for incoming advertisement packets
    - The scanner cycles through each advertisement channel in a round-robin fashion one channel at a time
  - Active scanning: the scanner listens for incoming advertisement packets and, upon receiving one, sends an additional scan request packet to the advertiser
    - To learn more about it
    - Typically the scan response contains information like the list of supported services and friendly name, but the application has full control of the scan response data payload

https://www.silabs.com/documents/public/user-guides/ug103-14-fundamentals-ble.pdf

BLE Link Layer Operations: Connections

- Allow application data to be transmitted in a reliable and robust manner
- Connections use
  - CRCs (Cyclic Redundancy Ceck)
  - Acknowledgements
  - Retransmissions of lost data
- Adaptive Frequency Hopping (AFH) to detect and adapt to the surrounding RF conditions and provide a reliable physical layer
- Support encryption and decryption of data to ensure confidentiality

- Timeline of a connection:
  - A scanner receives an advertisement packet by an advertiser that allows connections
  - The scanner becomes the initiator of the connection by sending a connect request
  - Once the connection is set
    - The scanner becomes the master
    - The advertiser become the slave

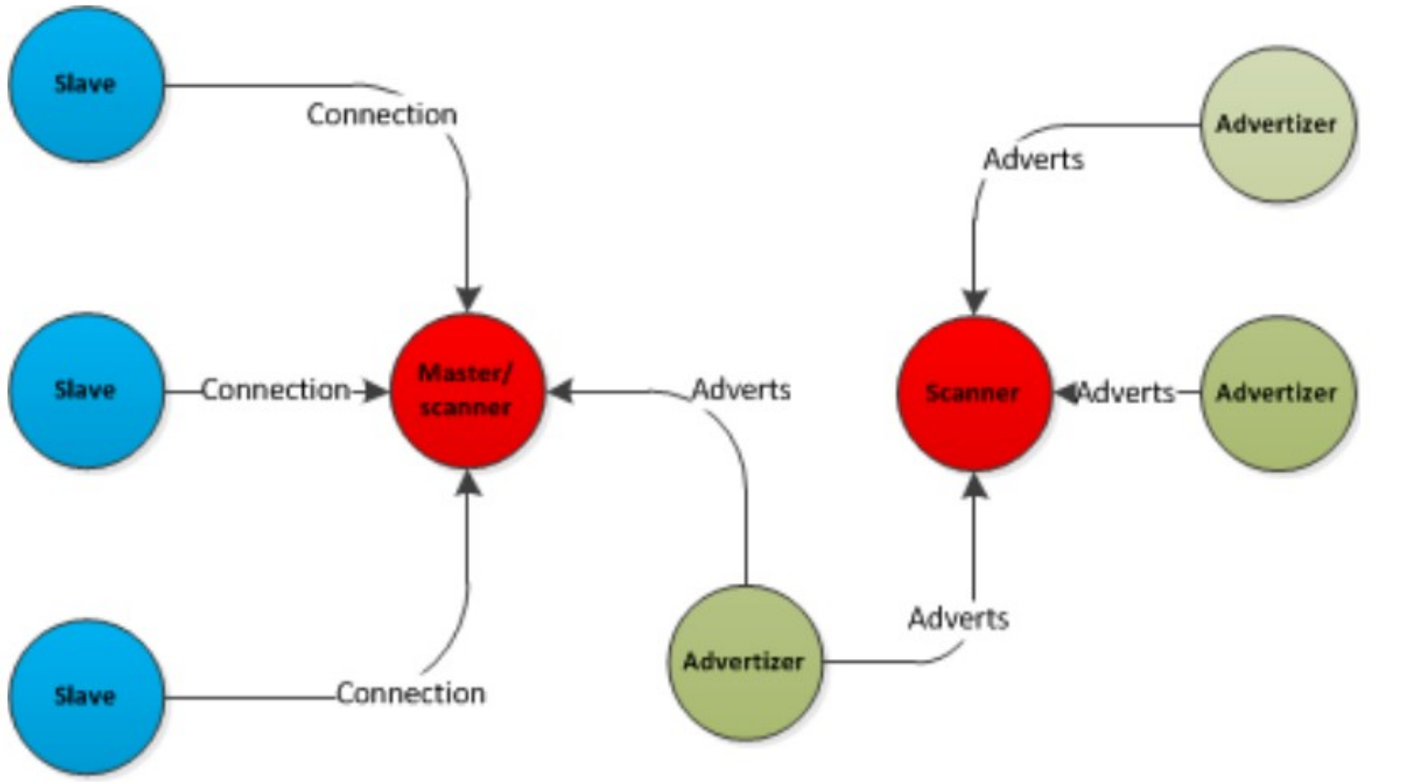
https://www.silabs.com/documents/public/user-guides/ug103-14-fundamentals-ble.pdf

BLE Network Topologies

- Device roles in Bluetooth low energy technology are:
  - **Advertiser**: A device that broadcasts advertisement packets, but is not able to receive them
    - It can allow or disallow connections
  - **Scanner**: A device that only listens for advertisements
    - It can connect to an advertiser
  - **Slave**: A device connected to a single master (BT 4.0) or multiple masters (BT 4.1 and newer)
  - **Master**: A device that is connected to one or more slaves
    - Theoretically a master can have an unlimited number of slave devices connected to it
    - In practice the master can connect 4-20 slaves at a time
  - **Hybrid**: It is possible for a device to advertise and scan at the same time or be connected to a master and advertise or scan simultaneously
    - Vendor-specific, and the exact features that are supported are vendor-specific too

https://www.silabs.com/documents/public/user-guides/ug103-14-fundamentals-ble.pdf

BLE Network Topologies



https://www.silabs.com/documents/public/user-guides/ug103-14-fundamentals-ble.pdf

BLE Security

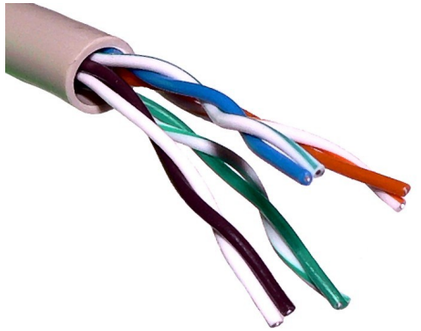
- BLE provides features to ensure trust, integrity, privacy and encryption of the data
- Three basic security services
  - Authentication and Authorization: Establishing trusted relationships between devices
  - Encryption and Data Protection: Protecting data integrity and confidentiality
  - Privacy and Confidentiality: Preventing device tracking
    - Advertising packets contain randomly generated MAC addresses disguising device identity, the real MAC address remains hidden

https://www.silabs.com/documents/public/user-guides/ug103-14-fundamentals-ble.pdf

CANBus

CANBus

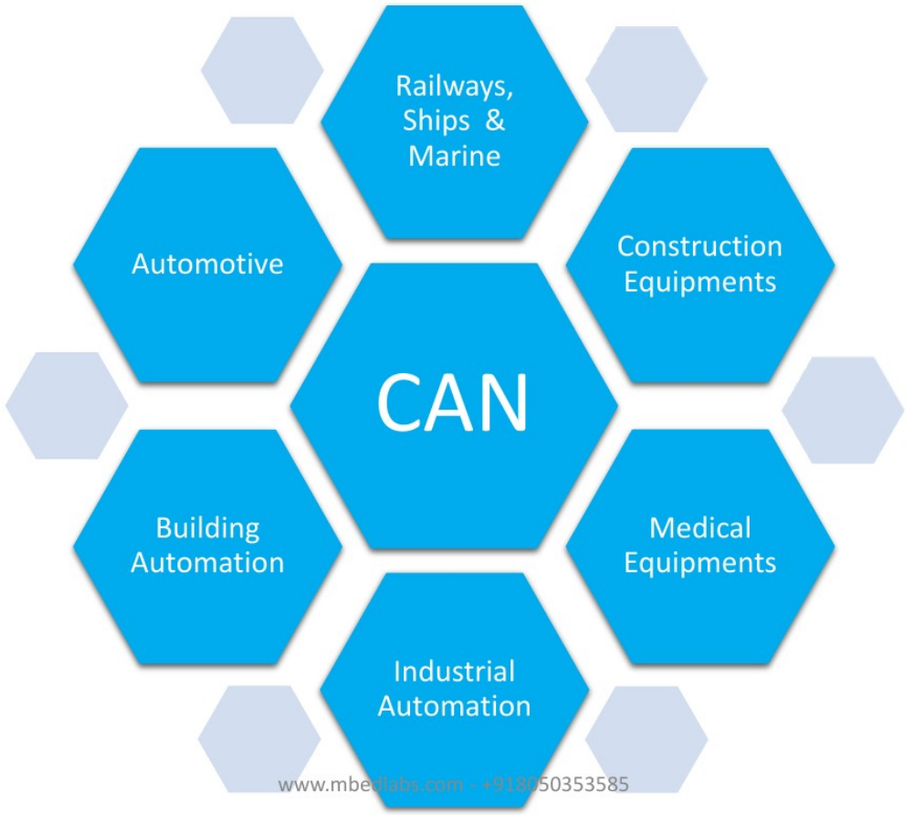
- Controller Area Newtork
- Developed in the early ‘80s for automotive
- Wired
  - 2 wires (usually a twisted pair)
- Data rate up to 1Mbps
- Bus architecture
- Serial
- Asynchronous
- Priorities + low latency: supports real-time



By Baran Ivo - Own work, Public Domain, https://commons.wikimedia.org/w/index.php?curid=2964670

mbedlabs Technosolutions, *Controller Area Network*

Applications



mbedlabs Technosolutions, *Controller Area Network*

Bus

- Linear bus structure
- Small latency
- Typically 3 to 40 nodes per bus
- Hot plug-in and plug-out

- Data rate depends on bus length
  - Class C
    - Data rate: 1 Mbit/sec
    - Bus length: up to 40 meters
  - Class B
    - Data rate: 125 kBit/sec
    - Bus length: up to 500 meters
  - Class A
    - Data rate: 50 kBit/sec
    - Bus length: up to 1000 meters

mbedlabs Technosolutions, *Controller Area Network*

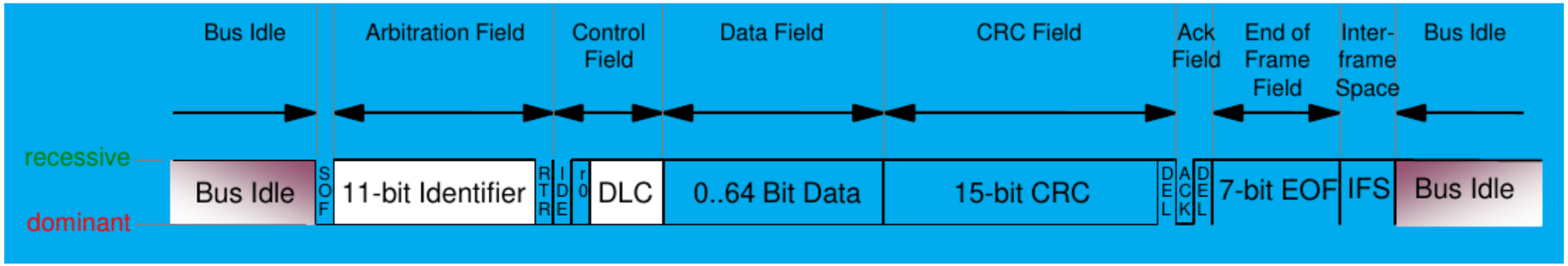
Transmission Principles

- Messages are broadcast
  - Recipients filter messages based on sender
- Bus access: CSMA/CA (Carrier Sense Multiple Access with Collision Avoidance)
  - Carrier Sense: Every node monitors the bus level, all the time
    - Monitoring of foreign and own CAN frames
  - Multiple Access: every node can start a transmission any time when the bus is free
  - Collision Avoidance: when several nodes start a transmission at the same time, all but one withdraw from sending

mbedlabs Technosolutions, *Controller Area Network*

Transmission Principles

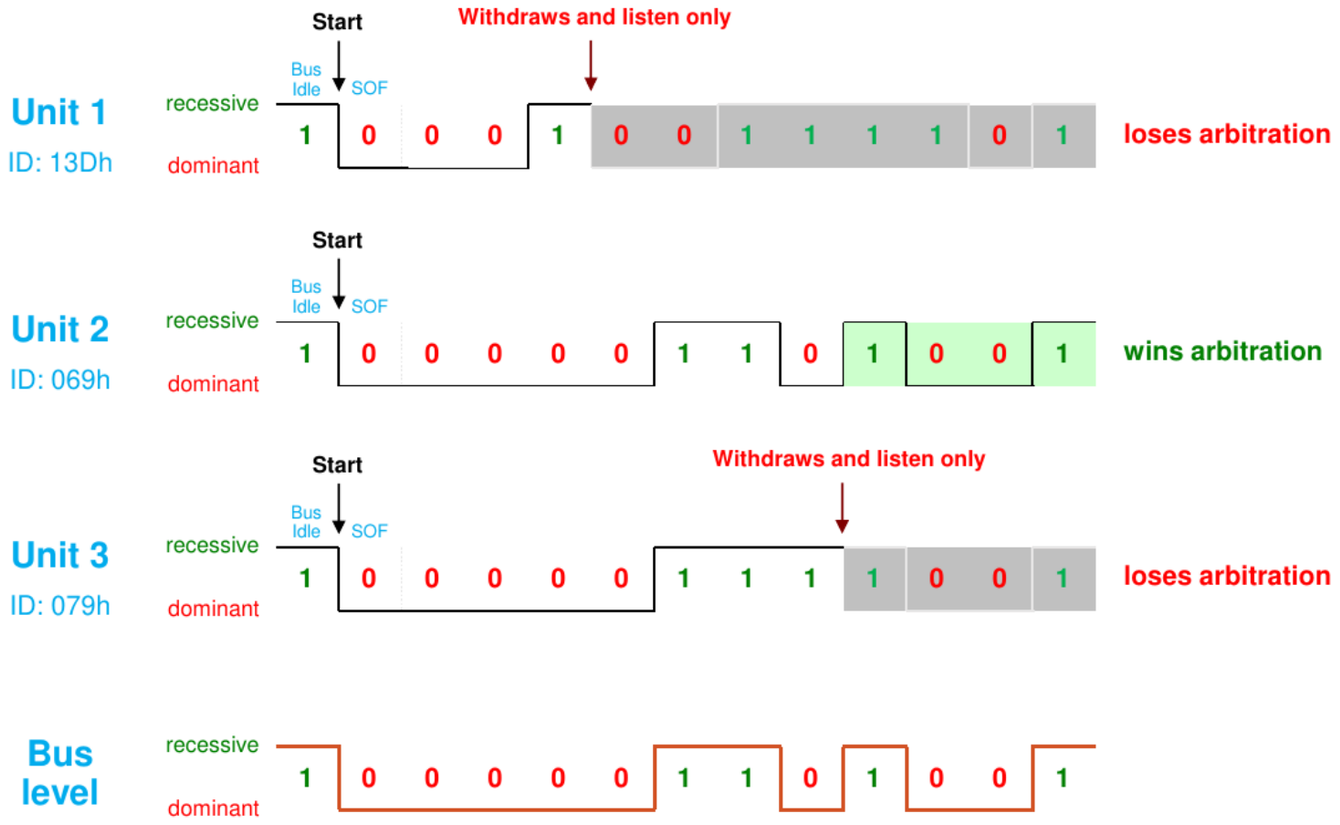
- Data frame, 11-bit device identifier



- The identifier is used for bus arbitration: low identifiers=higher priority
  - Devices sense the bus and, as soon as it is free, they can start transmitting
  - If multiple devices start transmitting at the same time, arbitration is performed by the devices themselves
    - Recessive (1) and dominant (0) bits on the bus

mbedlabs Technosolutions, *Controller Area Network*

Arbitration: Example



mbedlabs Technosolutions, *Controller Area Network*

Transmission Principles

- CRC error detection
  - Recognizes up to 5 single-bit errors per frame
  - Recognizes burst errors with lengths of up to 14 bits
    - i.e., sequences of errors
  - Recognizes all odd numbers of bit errors
- When an error is detected, an error frame is immediately transmitted
  - Sender and receivers reject the erroneous frame immediately
  - Sender retries transmission
- Error counters are used to automatically deactivate devices
- The probability for not discovering an error is 4.7×10^{-11}

mbedlabs Technosolutions, *Controller Area Network*

MQTT

MQTT – Message Queue Telemetry Transport

- MQTT is a lightweight messaging protocol
  - OASIS Standard: https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.html
  - Useful for low power sensors
- Based on the principle of publishing messages and subscribing to topics
- Brokers and clients
  - Multiple clients connect to a broker and subscribe to topics
    - Many clients may subscribe to the same topics
  - Clients also connect to the broker and publish messages to topics

http://mosquitto.org/man/mqtt-7.html

MQTT Topics

- Messages in MQTT are published on topics
- Topics are treated as a hierarchy, using a slash (/) as a separator
  - E.g., sensors/room/temperature

http://mosquitto.org/man/mqtt-7.html

MQTT Topics

- Clients can receive messages by creating subscriptions
  - A subscription may be to an explicit topic or it may include wildcards:
    - + can be used as a wildcard for a single level of hierarchy
      - E.g., sensors/+/temperature
    - # can be used as a wildcard for all remaining levels of hierarchy
      - E.g., sensors/#
  - Zero-length topic levels are valid, but can lead to some slightly non-obvious behaviour
    - For example, a topic of "a//topic" would correctly match against a subscription of "a/+/topic"

http://mosquitto.org/man/mqtt-7.html

MQTT Features

- Retained Messages:
  - The broker will keep the message even after sending it to all current subscribers
    - If a new subscription is made that matches the topic of the retained message, then the message will be sent to the new client
  - Useful as a "last known good" mechanism
- Clean session / Durable connections (clean session flag/clean start flag)
  - When the client disconnects, any subscriptions it has will remain and any subsequent QoS 1 or 2 messages will be stored until it connects again in the future
    - If clean session is false, then the connection is treated as durable
    - If clean session is true, then all subscriptions will be removed for the client when it disconnects

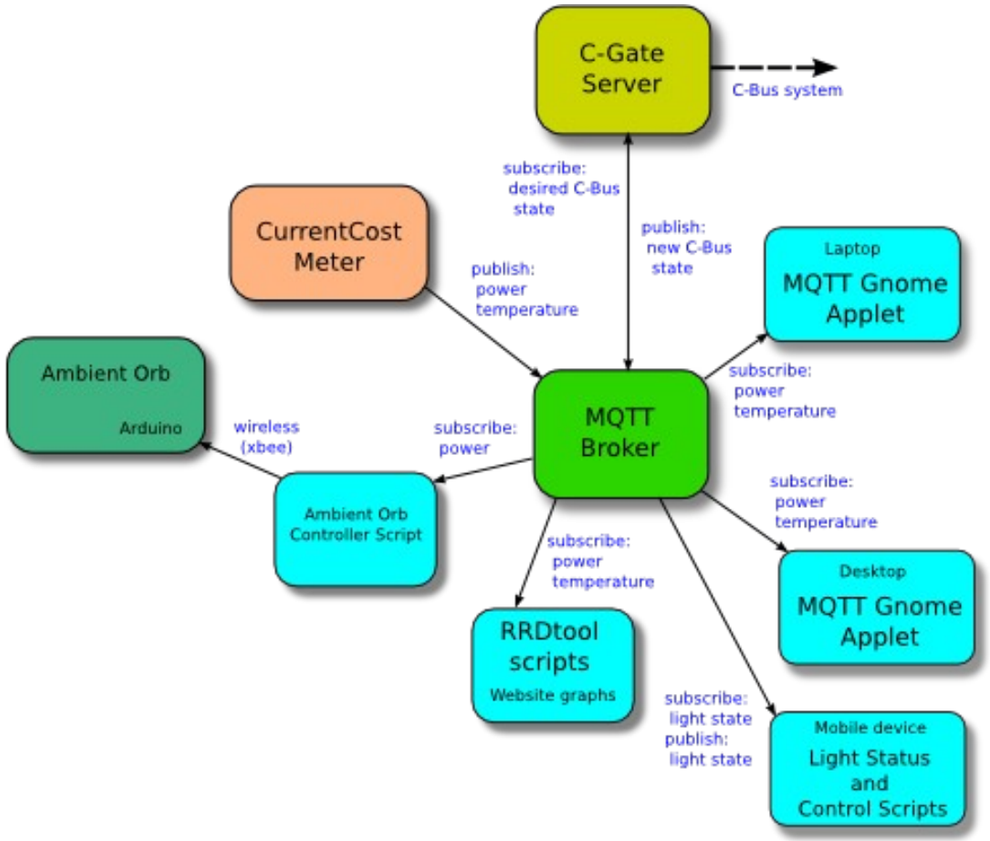
http://mosquitto.org/man/mqtt-7.html

MQTT Features

- When a client connects to a broker, it may inform the broker that it has “a **will**”:
  - A message that it wishes the broker to send when the client disconnects unexpectedly
    - It has a topic, QoS, and retain status

http://mosquitto.org/man/mqtt-7.html

MQTT Example



MQTT Quality of Service (QoS)

- Quality of service in MQTT does not impact communication bandwidth, but reliability of communication
- The QoS defines how hard the broker/client will try to ensure that a message is received
- Three levels
  - 0: The broker/client will deliver the message once, with no confirmation
  - 1: The broker/client will deliver the message at least once, with confirmation required
  - 2: The broker/client will deliver the message exactly once by using a four step handshake
  - Higher levels are more reliable, but involve higher latency and have higher bandwidth requirements

http://mosquitto.org/man/mqtt-7.html

MQTT Quality of Service

- The QoS of each specific communication depends on
  - The QoS level provided by the broker → max QoS level available
  - The maximum QoS level specified by the client → Desired QoS level, if available

- If a message is published at QoS 2
  - Client subscribed with QoS 0 → Message delivered to that client with QoS 0
  - Client subscribed with QoS 2 → Message delivered to that client with QoS 2
- If a message is published at QoS 0
  - Client subscribed with QoS 2 → Message delivered to that client with QoS 0

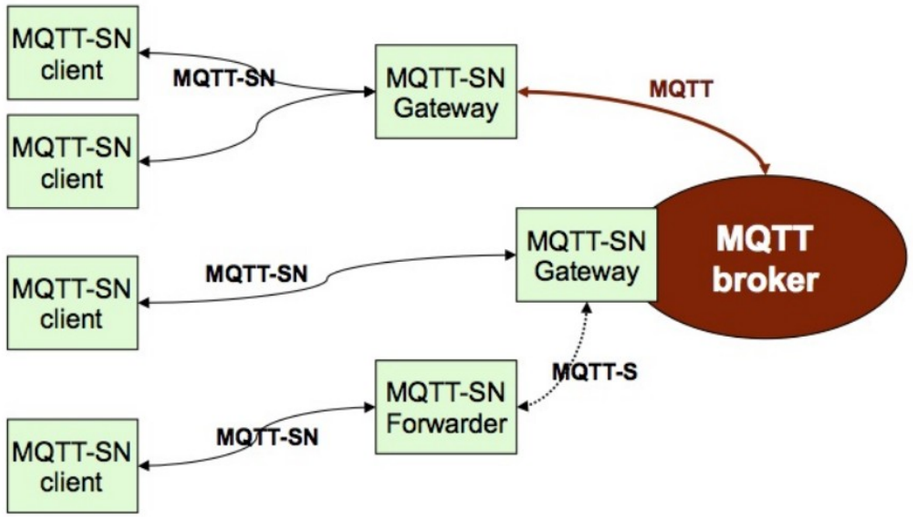
http://mosquitto.org/man/mqtt-7.html

MQTT-SN – MQTT for Sensor Networks

- MQTT-SN is similar to MQTT, but adapted to the peculiarities of a wireless communication environment such as
  - Low bandwidth
  - High link failures
  - Short message length
  - Low-cost, battery-operated devices with limited processing and storage resources
- The network architecture also includes gateways

http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf

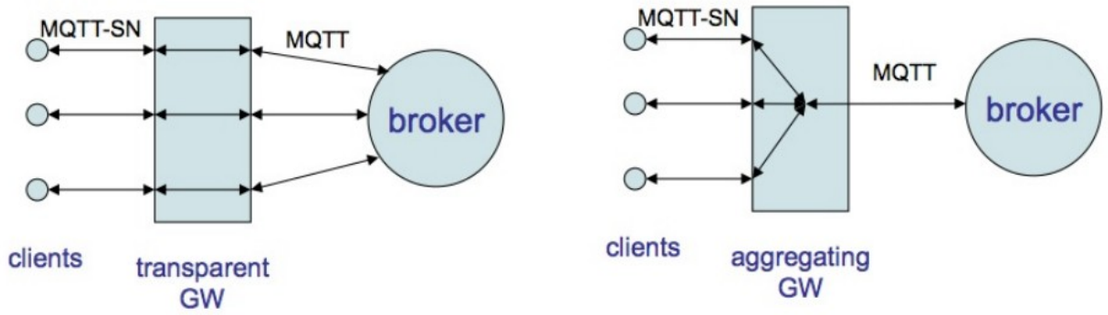
MQTT-SN Architecture



- There are three kinds of components
  - MQTT-SN clients: connect to an MQTT broker via an MQTT-SN GW
  - MQTT-SN gateways (GW): translates between MQTT and MQTT-SN
    - May be integrated with an MQTT broker
      - Stand-alone GW → the MQTT protocol is used between the broker and the GW
  - MQTT-SN forwarders: allows MQTT-SN clients not directly connected to a GW network to access a GW
    - Encapsulate the MQTT-SN frames from the wireless side to the GW
    - Decapsulate the frames from the gateway to the clients

http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf

MQTT-SN Gateways



- Although the implementation of the transparent GW is simpler when compared to the one of an aggregating GW, it requires the MQTT server to support a separate connection for each active client
  - Some MQTT server implementations might impose a limitation on the number of concurrent connections that they support

http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf

What's different in MQTT-SN?

- The CONNECT message is split into three messages
  - The two additional ones are optional and used to transfer the Will topic and the Will message to the server
- The topic name in the PUBLISH messages is replaced by a short, two-byte long "topic id"
  - A registration procedure is defined to allow clients to register their topic names with the server and obtain the corresponding topic id
- "Pre-defined" topic ids: two-byte long replacement of the topic name, their mapping to the topic names is known in advance by both the client's application and the gateway/server
    - Both sides can start using pre-defined topic ids without registration

http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf

What's different in MQTT-SN?

- A discovery procedure helps clients without a pre-configured server/gateway's address to discover the actual network address of an operating server/gateway
  - Multiple gateways may be present at the same time within a single wireless network and can co-operate in a load-sharing or stand-by mode
- The semantic of a "clean session" is extended to the Will feature
  - Not only client's subscriptions are persistent, but also Will topic and Will message
  - A client can also modify its Will topic and Will message during a session
- A new offline keep-alive procedure is defined for the support of sleeping clients
  - Battery-operated devices can go to a sleeping state during which all messages destined to them are buffered at the server/gateway and delivered when they wake up

http://mqtt.org/new/wp-content/uploads/2009/06/MQTT-SN_spec_v1.2.pdf

MQTT in mBed

- MQTT library available for Arduino: *ArduinoMqttClient* library
  - Reference: https://www.arduino.cc/reference/en/libraries/arduinomqtt
  - MQTT-SN libraries are also available for some Arduino devices
- MQTT libraries are provided in mBed
  - Part of the Eclipse Paho project, open-source client implementations of MQTT and MQTT-SN messaging protocols
  - The MQTT API is portable across network interface stack
  - Example available at: https://os.mbed.com/teams/mqtt/code/HelloMQTT/

In short...

- Network architectures for IoT
- Cloud / Fog / Edge computing
- BLE
- CANBus
- MQTT