3b) (i) XML

* Markup language defines a set of rules for encoding documents that can be read by both humans and machine Designed with focus on storing and transporting data.

* 1996 - Invented year
* Extended form - SGML
* Usage to transport data between the application and the database. To develop other markup languages
* It is dynamic
* Processing / rules - St ; rules must be followed or processor will terminate processing the file.

* Language type - Neither presentation or programming
* Tage - Custom tags can be created by the author
* White space - Preserves white space

HTML.

*Markup language for displaying web pages in web browsers. Designed to display data with focus on how the data looks.

* 1990 - invented year
* SGML
* Diplay web page.

* It is static.
* No strict rules. Browsers will still generate data to the best of it's ability

* Presentation.

* Predefined tags.

* Cannot preserve white space

Q) HTML v/s XML

Limitations - Cannot be used as a subtype of a sql_variant instance.
Does not support casting or converting to either text or ntext. Does not support following column and table constraints XML provides its own encoding. Collations apply to string types only. Cannot be compared or sorted. Cannot be used in Distributed Partioned views. Not well supported by browsers.

* Data does not know itself very well. Data cannot change in response to environment Data cannot be easily maintained. Cannot store or call variables. Lacks the capability to define new structures by defining relationship between the classes. Tags are not useful for exchanging the document between applications.

ii) SOAP

* Its a protocol.

* SOAP stands for Simple Object Access protocol
* SOAP can't use REST because its a protocol.

* SOAP uses services interfaces to expose the business logic
* JAX-WS is the java API for SOAP web services.

REST.

* It is an architectual style.
* REST stands for Representational State Transfer.
* REST can use SOAP web services : it is a concept and can use any protocol like HTTP, SOAP.
* REST uses URI to expose business logic.
* JAX-RS is the java API for RESTful web services.

| | |
|---|---|
| * SOAP defines standards to be strictly followed | * REST does not define too much standards like SOAP |
| * SOAP requires more bandwidth and resources than REST | * REST requires less bandwidth and resources than SOAP. |
| * SOAP defines its own security | *. RESTful web service inhuits security measures from the underlying transport. |
| *. SOAP permits XML data format only | * REST permits different data formats such as Plain text, HTML, XML, JSON etc. |
| *. SOAP is less preferred than REST. | * REST is more preferred than SOAP. |

2a) AMQP - Advanced message Queuing Protocol

→ Its goal was to provide a vendor - neutral protocol for managing the flow of message across enterprises business systems.

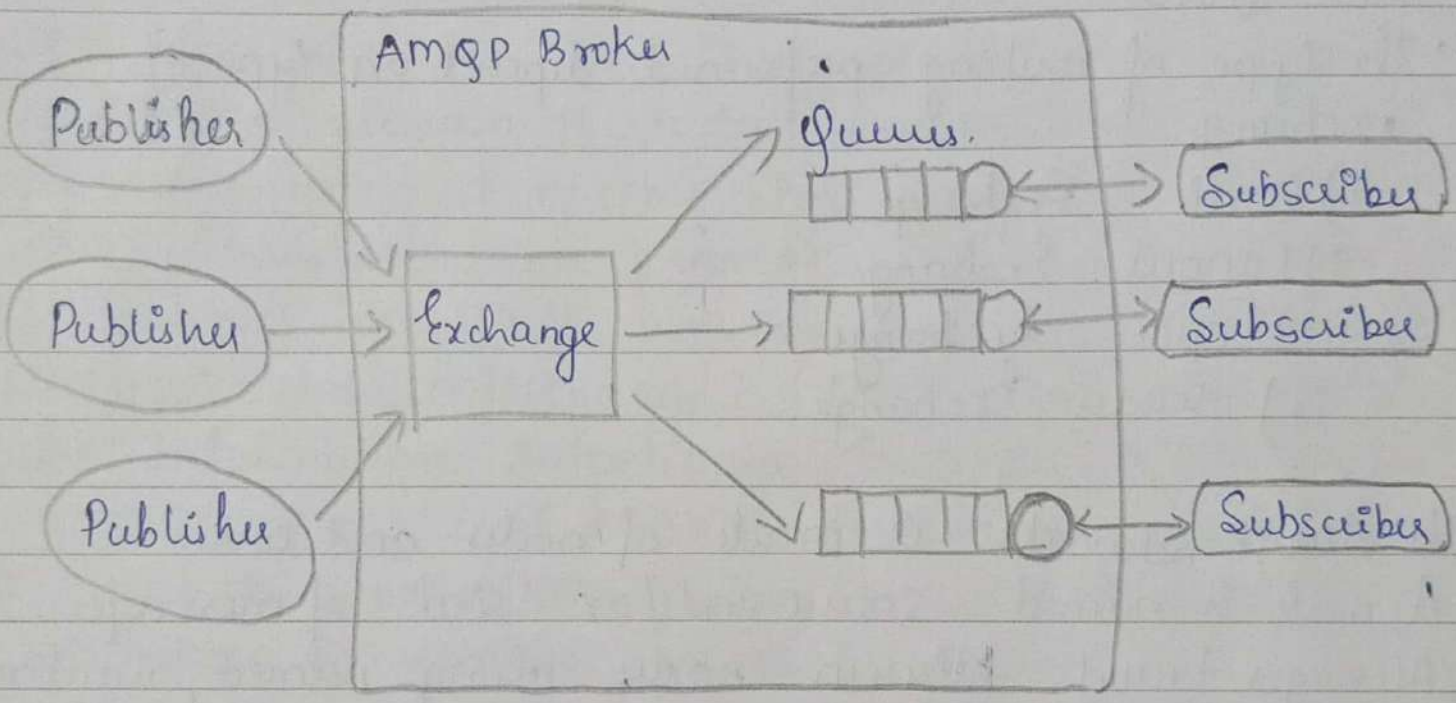
→ Secure, compact, multiplexed, reliable, symmetric, binary message transfer protocol to move messages between the applications.

→ The AMQP is an open standard application layer protocol for message - oriented middleware.

→ The defining features of AMQP are message orientation, queing, routing (including point to point and publish-

and subscribe), reliability and security.

Hybrid Model




AMQP Components:

1) Exchange / Broker: A part of broker (i.e server) which receives message from publishers (source of data) like sensors for example. and routes them to different queues.

2) Message Queue: A named entity in which messages are associated with and from where customers recieve them

3) Binding: Rules for distributing messages from exchanges to queues.

9

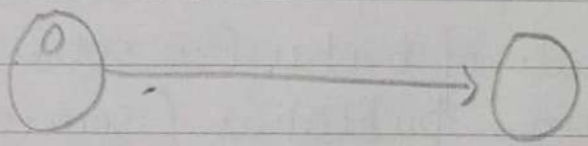
Working:

→ After recieving messages from publishers (i.e clients), the exchange process them and route them to one or more queues

→ The type of routing performed depend on type of exchange.

i) Direct Exchange
ii) Fanout Exchange
iii) Topic Exchange
iv) Headers Exchange.

The AMQP networks is made of nodes and links. A node is named source and/or sink of messages. Messages travel between nodes along named, unidirectional links.

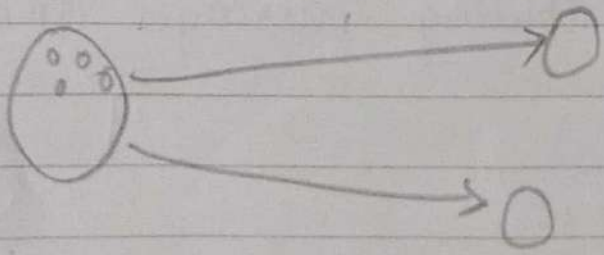


Source Node Target Node.

Types of links:

i) Destructive: the transfer along the link removes the message from the source.

ii) Non-destructive: the message remains at the source node, and "copied" to destination.



The MQTT uses publish/subscribe model. but in AMQP Client/Broker or Client/server architecture is used. The MQTT uses only Client/Broker architecture.
In MQTT broker sends message to clients subscribed to the topic that broker recieved from publisher. but in AMQP it is transfered to Queues.

The message size in MQTT is small and undefined whereas in AMQP it is undefined and negotiable.

Methods used in MQTT - Connect, Disconnect, Publish-Subscribe, Unsubscribe Close
AMQP - Consume, deliver, Publish, Get, Select, Ack, delete NACK, Recover, Reject, Open Close.

2b). The Raspberry Pi device looks like motherboard, with the mounted chips and ports exposed, but it has all its components that is needed to connect input, output and storage devices and start computing.

i) ARM CPU/GPU - This a Broadcom BCM2835 System on a Chip (SoC) that's made up of an ARM CPU and a Videocore 4 GPU. The CPU handles all the computations that make a computer work (taking input, making calculations, and producing output), and GPU handles graphic output.

ii) GPIO - These are exposed general-purpose input/output

connection points that will allow the real hardware hobbyists the opportunity to tinker.

RCA - An RCA jack allows connection of analog TV's and other similar output device.

Audio out - this is the standard 3.55 mm jack for connection of audio output devices such as headphones / speakers. There is no audio on.

LED's - light emitting diodes. as indicators.

USB - common connection port for peripheral devices of all types like mouse / keyboard.

HDM - this connector allows you to hook up a high-definition tv or other compatible device using HDMI cable.

Power - This a 5v micro USB power connector into which you can plug your compatible power supply.

Ethernet - this connector allows for wired network access and is available in model B.

SD card slot is full-sized. An SD card with an OS installed is required for booting the device.

i) Web-publishing: XML allows you to create interactive pages, allows the customer to customize those pages, and makes creating e-commerce apps more intuitive.. With XML, you store the data once and then renders that content for different viewers or devices based on style sheet processing using Extensible Style Language (XSL) / XLS Transformation processor.

ii) e-business apps: XML provides implementations make electronic data interchange (EDI) more accessible for information interchange, business-to-business transactions and business-to-consumer transaction.

iii) Metadata apps: XML makes it easier to express metadata in portable, reusable format.

iv) General apps: XML provides a standard method to access information, making it easier for apps and devices of all kinds to use, store, transmit and display data.

v). Pervasive computing: XML provides portable and structured information types for display on pervasive (wireless) computing devices such as personal digital assistants (PDA's), cellular phones, and others. Ex: WML (Wireless Markup language) and Voice XML are currently evolving standards for describing visual and speech - driven wireless interfaces

7) Web searching and auto mating web tasks : XML defines the type of information contained in a document, making it easier to return useful results when searching the Web.
Ex: using HTML to search for books authored by 'X' is likely to return instances of the term 'Y' outside the context of author. Using XML restricts the search to the correct context and are more efficient and produce more useful results.

7a)

RIOT is a small OS for networked, memory-constrained systems with a focus on low-power wireless IOT devices
→ It is open-source software, released under the LGPL
→ Due to this undoubable license and its large independent community RIOT is often refered to as the Linux of the Internet of Things

Features of RIOT are:
→ There are no new programming environments. C or C++ can be used directly with existing tools like gcc, gdb, etc
→ Less hardware dependent code
→ Supports 8-, 16- and 32-bit microcontroller platforms
→ Energy efficiency is maintained
→ Less interrupt latency, so real-time capability is ensured
→ Multithreading is enabled
→ Supports the entire network stack of IoT (802.15.4 Zigbee, 6LoWPAN, ICMP6, IPV6, RPL, CoAP, etc)
→ Both static and dynamic memory allocation
→ POSIX compliant (partial)
→ All output can be seen in the terminal if hardware is not available; however, there is a visualization tool called RIOT-TV that is provided
→ flexible memory management
→ RPL (storing mode, P2P mode)
→ high resolution, long-term timers.
→ a preemptive, tickless scheduler

```
import RPi.GPIO as GPIO
import time
GPIO.setmode (GPIO.BROAD)
TRIG = 16
ECHO = 18
i=0
GPIO.setup (TRIG, GPIO.OUT)
GPIO.setup (ECHO, GPIO.IN)
GPIO.output (TRIG, False)
print" Calibrating ---"
time.sleep (2)
print" Place the object ---"
try:
    while True:
        GPIO.output (TRIG, True)
        time.sleep (0.00001)
        GPIO.output (TRIG, False)
        while GPIO.input (ECHO) == 0:
            pulse_start = time.time()
        while GPIO.input (ECHO) == 1:
            pulse_end = time.time()
        pulse_duration = pulsecend - pulse_start
        distance = pulse_duration * 17150
        distana = round (distance + 1.15, 2)
        if distanca <= 20 and distance >= 5:
            print" distance :", distance, "cm"
            i=1
        if distance > 20 and i == 1:
            print "place object"
            i=0
time
```

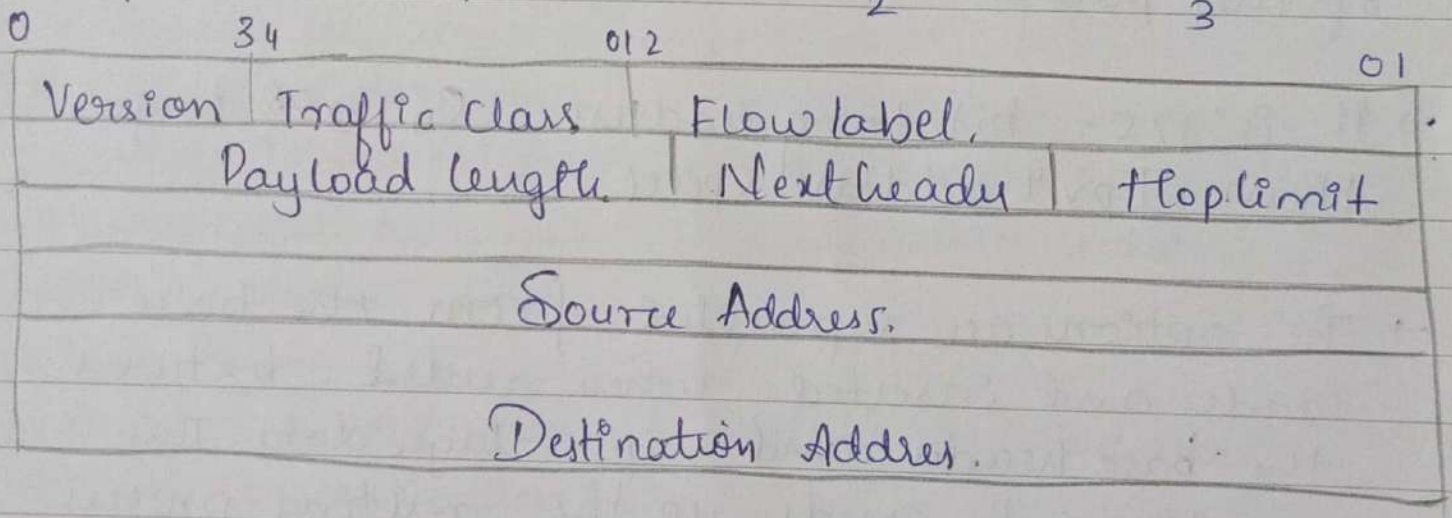
```
        time.sleep(2)
except KeyboardInterrupt:
        GPIO.cleanup()
```

2b. 2.

Ipv6 over Ipv4

→ Large address space. It is the address space is huge.
→ Better header format than Ipv4.
→ New options are available
→ Support for resource allocation.
→ Support for more security
→ Support for mobility.

o Ipv6 header.




→ Ipv6 header length is 40 and Ipv4 is 20bytes
→ IPv6 provides interoperability and mobility capabilities. which are already ~~ent~~ widely embedded in network devices.
→ Direct addressing is possible due to vast address space. so the need of address translation is effectively reduced.
→ Allowance for extension: Ipv6 is designed to allow the extension of the protocol if required by new technologies or application.

→ Support for resource allocation In Ipv6, the type of service field has been removed, but a mechanism. (called flow label) has been added to enable the source to request special handling of the packet. This mechanism can be used to support traffic such as realtime audio and video.

→ Support for more security: The encryption and decryption and authentication options in Ipv6 provide confidentiality and integrity of the packet.

→ It is 128-bit long address. So $2^{96}$ huge than Ipv4 address space.

→ The options are separated from the base header and inserted, when needed, between the base header and upper-layer data. This simplifies and speeds up the routing process because most of the options do not need to be checked by route

(a)

→ Low Power Wide Area (LPWA) networks represent a novel communication paradigm, which all complement traditional cellular and short range wireless technologies in addressing diverse ~~dep~~ requirements of IoT applications
→ It has battery life of 20 years
→ LPWA networks are unique because they make different tradeoffs than the traditional technologies such as Zigbee, Bluetooth, Z-wave, Wi-fi, LTE etc.
→ It operates at 20mWatt power.
→ It achieves a long range with low power consumption and low cost. It works on ALOHA and uses star topology.
→ It covers wide area 15-50 km of urban indoor

Techniques Used:

1. Long Range
a. Use of sub frequencies of 1GHz band (More robut & Reliable communication)
b Modulation Techniques (Slows down modulation to put more energy)

2. Ultra low Power Operation
a. Topology (from mesh to Star)
b. duty cycling (allows radios to turn off transceiver)
→ It is a key requirement to tap into huge business opportunity provided by battery powered IoT and machine to machine devices.

→ A battery lifetime of 10 years or more with or coin cell batte is desirable to bring maintainence cost down

→ It uses star topology for the ultra low power operation

→ for duty cycling only during transmission of data transmitter is turned on

3) It has light weight Medium Access Control (MAC) (Simple random-acess ALOHA) and off loading complexity from end devices

3. Low Cost

1) Reduction in hardware complexity compared to the cellular and short range wireless technology. LPWAN transceiver needs to process less complex waveform It enables them to reduce transceiver footprints, peak data rates and memory sizes. Thus minimizing the hardware complexity that in turn reduces cost A single LWPAN can connect 1000's of devices distributed across the geographical area.

2) Minimum infrastructure

3) Using license free or Own license band

4) Scalability