MQTT协议说明

eric.xiao

ericssonxiao@gmail.com

谁创造了MQTT协议?

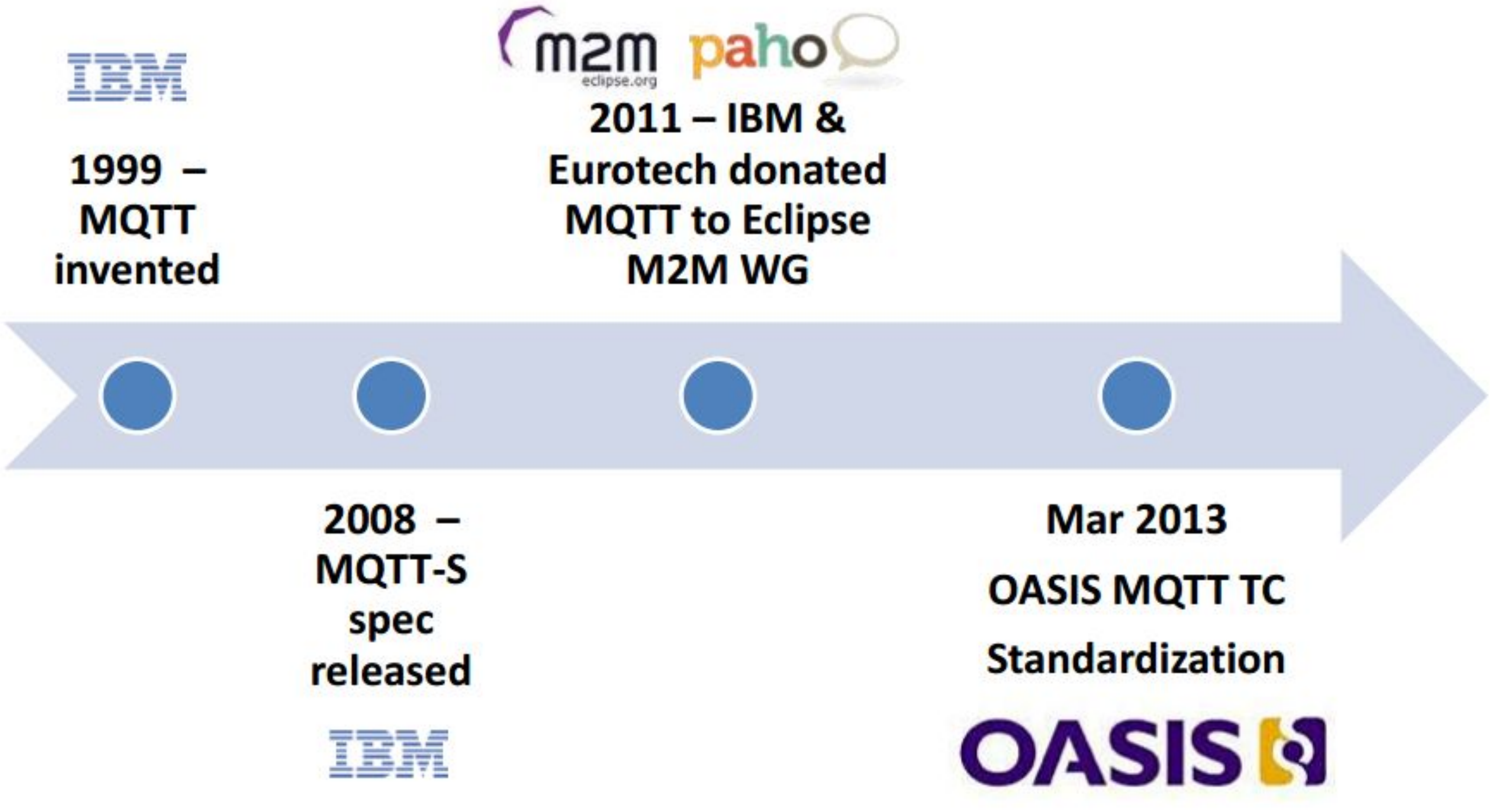
在1999年，来自IBM公司的Andy Stanford-Clark 和 来自Arcom (现在改名为Eurotech)公司的Arlen Nipper创造了MQTT协议。

Dr Andy Stanford-Clark

Arlen Nipper

MQTT协议的发展历程

MQTT Timeline



1999 – MQTT invented

2008 – MQTT-S spec released

2011 – IBM & Eurotech donated MQTT to Eclipse M2M WG

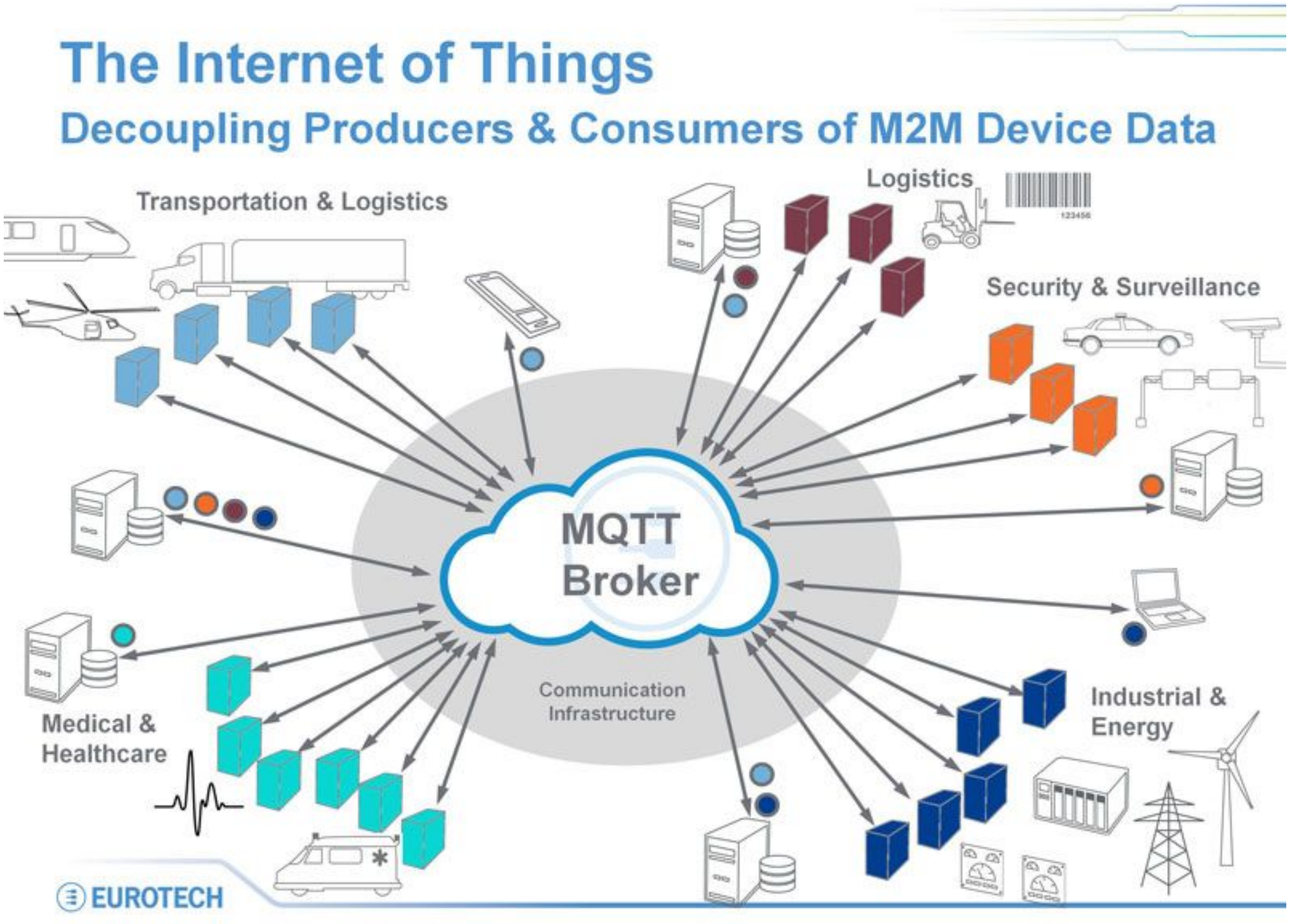
Mar 2013

OASIS MQTT TC Standardization

标准化组织

- 2013年3月, MQTT 协议提交到**OASIS**组织并不断演进过程中。
- MQTT协议规格书被公开发布，它使用免版税的许可证，**Eurotech** 公司(之前的名字是 **Arcom**) 在他们的产品中实现了这个协议。
- 在2011年11月**IBM** 和 **Eurotech** 公司对外宣布加入**Eclipse M2M Industry 工作组** 并捐赠 MQTT 代码给Eclipse Paho 项目组。

MQTT协议在物联网中的位置



什么是MQTT协议?

MQTT 是消息队列遥感传输协议的缩写（Message Queueing Telemetry Transport Protocol）.

- 基于轻量级代理的发布/订阅协议
- 开放
- 简单
- 容易实现

所以它能够应用在资源受限的环境中.

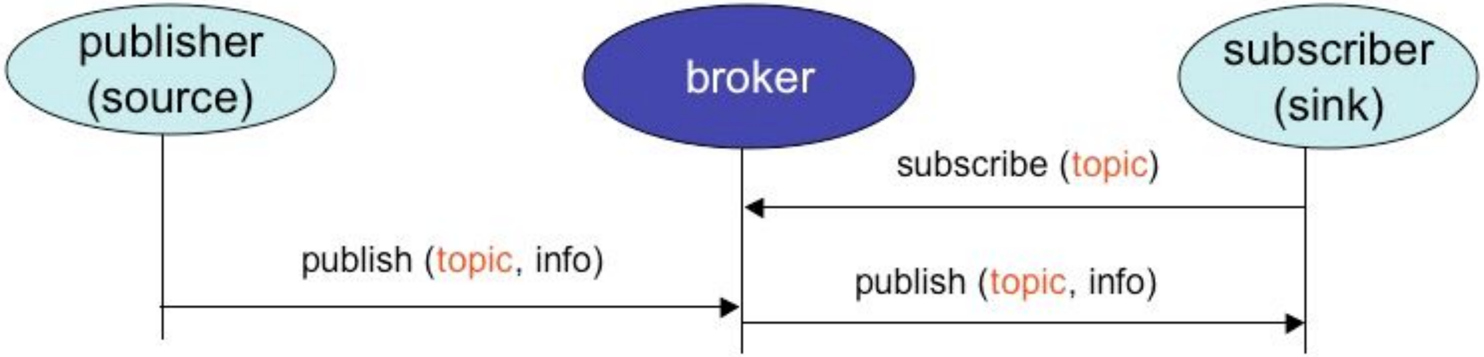
例如:

- 传感器
- 手机移动设备及平板电脑
- 物联网 (IoT)

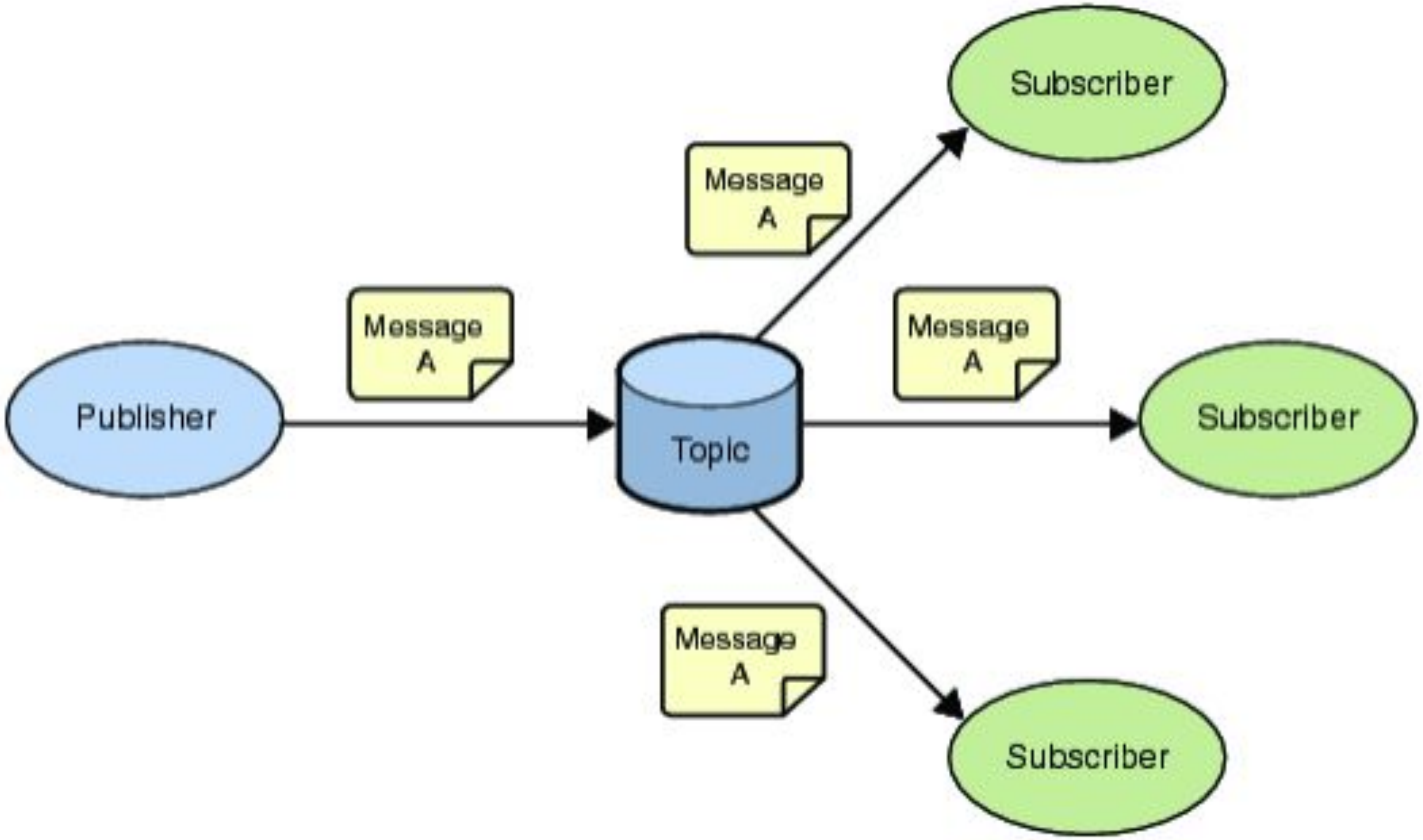
MQTT.ORG
MQTT
lightweight, easy-to-use, standards-based messaging
ideals for sensors, mobile & the Internet of Things
learn more at mqtt.org

发布/订阅 协议

发布/订阅 协议（简单的场景）

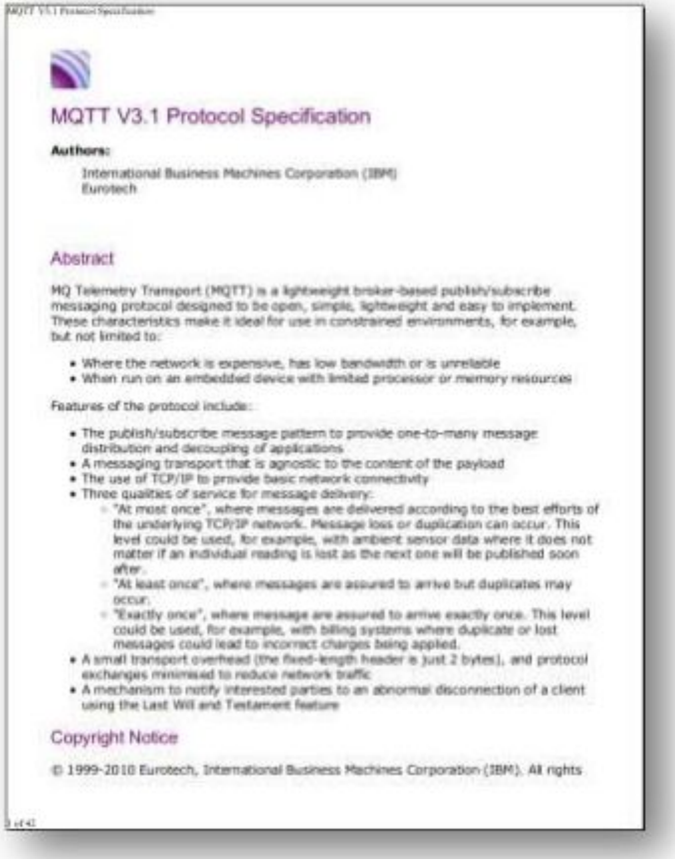


发布/订阅消息模式支持广播



MQTT 协议规格书

MQTT v3.1 spec



MQTT V3.1 Protocol Specification

MQTT V3.1 Protocol Specification

Authors:

International Business Machines Corporation (IBM)
Eurotech

Abstract

MQ Telemetry Transport (MQTT) is a lightweight broker-based publish/subscribe messaging protocol designed to be open, simple, lightweight and easy to implement. These characteristics make it ideal for use in constrained environments, for example, but not limited to:

- Where the network is expensive, has low bandwidth or is unreliable
- When run on an embedded device with limited processor or memory resources

Features of the protocol include:

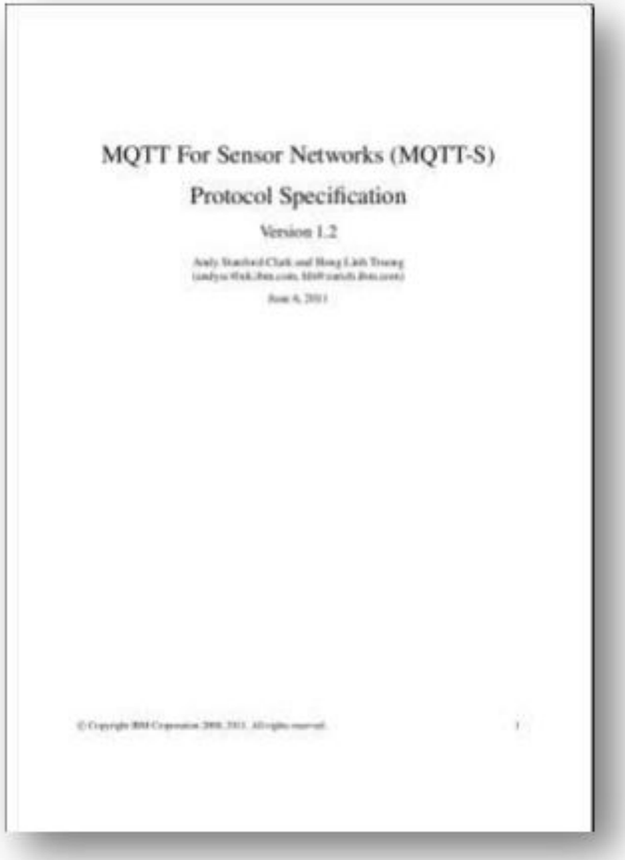
- The publish/subscribe message pattern to provide one-to-many message distribution and decoupling of applications
- A messaging transport that is agnostic to the content of the payload
- The use of TCP/IP to provide basic network connectivity
- Three qualities of service for message delivery:
  - "At most once", where messages are delivered according to the best efforts of the underlying TCP/IP network. Message loss or duplication can occur. This level could be used, for example, with ambient sensor data where it does not matter if an individual reading is lost as the next one will be published soon after.
  - "At least once", where messages are assured to arrive but duplicates may occur.
  - "Exactly once", where message are assured to arrive exactly once. This level could be used, for example, with billing systems where duplicate or lost messages could lead to incorrect charges being applied.
- A small transport overhead (the fixed-length header is just 2 bytes), and protocol exchanges minimised to reduce network traffic
- A mechanism to notify interested parties to an abnormal disconnection of a client using the Last Will and Testament feature

Copyright Notice

© 1999-2010 Eurotech, International Business Machines Corporation (IBM). All rights

3 of 41

MQTT-S v1.2 spec



MQTT For Sensor Networks (MQTT-S)

Protocol Specification

Version 1.2

Andy Stanford-Clark and Hong Linh Truong
(andysc@uk.ibm.com, hlt@zurich.ibm.com)

June 4, 2011

© Copyright IBM Corporation 2008, 2011. All rights reserved.

1

MQTT 协议规格书

- **MQTT v3.1** -- MQTT V3.1 协议规格书
  它是一个基于TCP/IP协议的，可提供发布/订阅消息模式的，十分轻量级的通信协议

- **MQTT v3.0** -- MQTT V3.0 协议规格书
  它是一个基于TCP/IP协议的，可提供发布/订阅消息模式的，十分轻量级的通信协议

- **MQTT-SN v1.2** -- MQTT for Sensor Networks V1.2 协议规格书
  它是一个基于非TCP/IP协议的（例如：ZigBee协议），专门为嵌入式设备提供消息通信模式的协议

设计原理和假设条件

- 简单，简单，简单!
- 发布/订阅消息传递
- 零管理，零配置
- 低带宽占用
- 预料并且满足网络的频繁中断带来的问题。

(低带宽, 高等待时间, 不可靠, 高成本的网络)

- 可提供持续的会话
- 客户端仅需做有限的处理工作，即可使用
- 在环境许可的情况下，可提供传统的QoS服务
- 无严格的数据内容的规定

MQTT协议的特性是什么?

- 使用发布/订阅消息通信模式来提供一对多的分布式消息和解耦合的应用
- 在消息传输过程中提供与消息内容无关的有效载荷
- 基于TCP/IP协议的基础网络连接
- 提供三种QoS的消息传递质量:

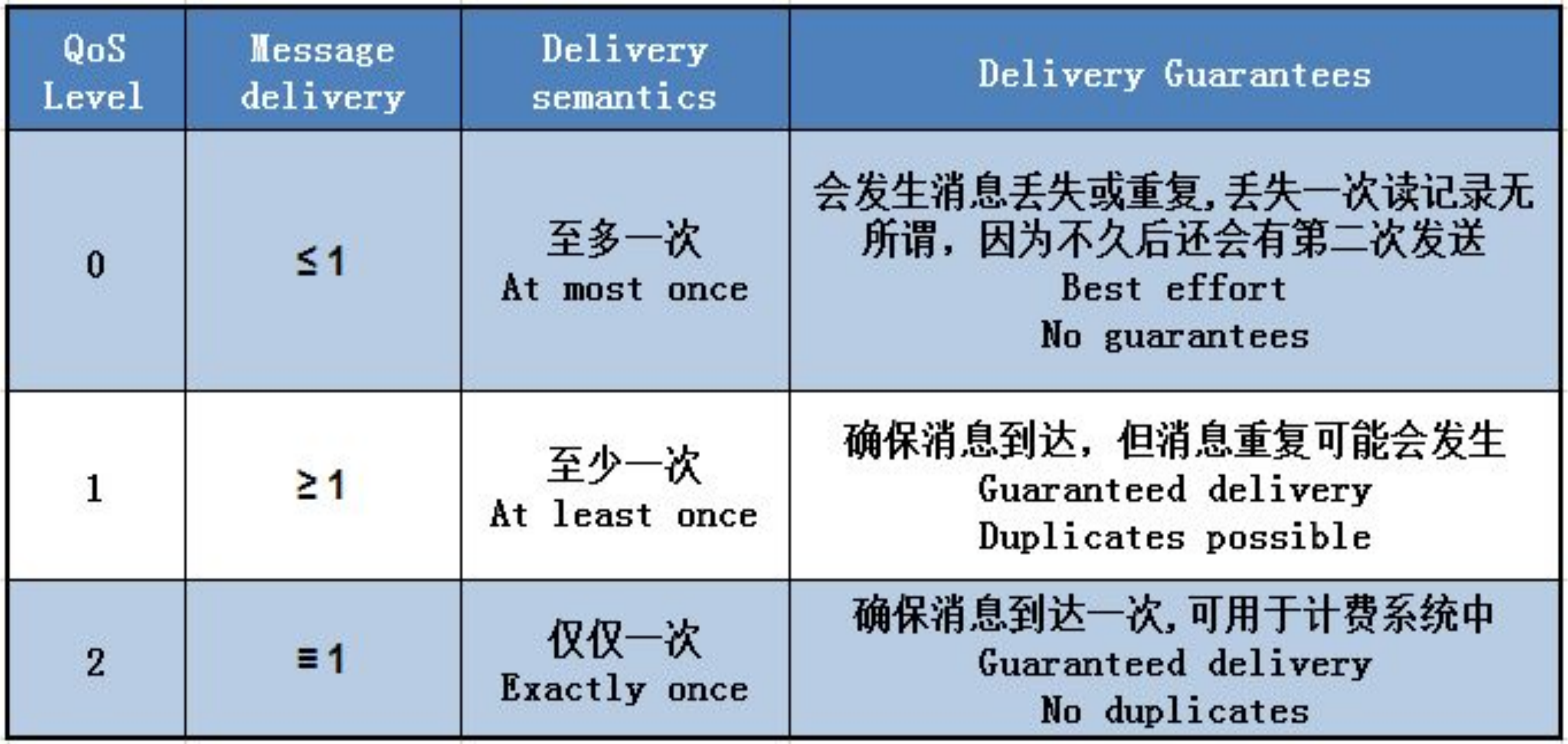
  a. 最多一次（ At most once ）

  b. 最少一次（ At least once ）

  c. 仅仅一次（ Exactly once ）
- 很小的传输费用开销 (固定头部的消息仅仅2个字节), 和很少的协议交换来减少网络传输通信量
- 使用约定的机制来提醒相关各方，客户端产生了一个异常的连接断开

三种消息传递的QoS质量

Qualities of Service



QoS Level	Message delivery	Delivery semantics	Delivery Guarantees
0	≤1	至多一次 At most once	会发生消息丢失或重复,丢失一次读记录无所谓，因为不久后还会有第二次发送 Best effort No guarantees
1	≥1	至少一次 At least once	确保消息到达，但消息重复可能会发生 Guaranteed delivery Duplicates possible
2	≡1	仅仅一次 Exactly once	确保消息到达一次,可用于计费系统中 Guaranteed delivery No duplicates

在什么场景下使用MQTT协议?

- 间歇性的连接
- 昂贵的带宽
- 需要与一个或多个手机终端进行通信的企业级应用
- 在不使用代码处理逻辑的情况下，使手机和平板电脑有可靠的数据传输
- 确保消息的有效分发
- 很低的等待响应时间

谁正在使用MQTT协议？

Certified for IBM | WebSphere. software

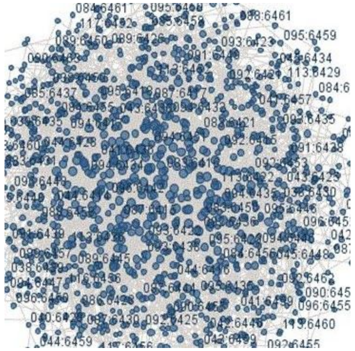
企业级应用：

1. WebSphere MQ By IBM

特性: 它能使用消息的形式传输任何数据, 使SOA商业系统架构更加灵活可靠

2. GaianDB

特性: 分布式联邦数据库使用生物启发自组织原理来减少管理



3. LAMA By IBM Extreme Blue Project

全名: Location Aware Messaging for Accessibility

特性: 在2006年作为IBM’s Extreme Blue 项目的一部分， LAMA 是一个可提供给人们感兴趣的东西和位置点的系统

4. SiSi By IBM Extreme Blue Project

全名: Say It, Sign It

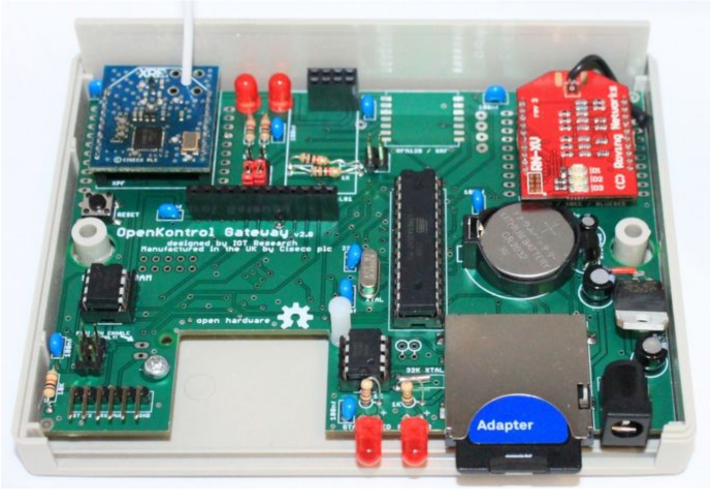
特性: 在2007年作为IBM’s Extreme Blue项目的一部分，SiSi 帮助聋哑人将语音转换为英国手语（BSL），通过MQTT和Java来实现

谁正在使用MQTT协议？

智慧家居:

1. Andy SC's Twittering / Automated House
2. Power Monitoring
3. Lighting Control
4. Gardening
5. Energy monitoring with an old-style analog ammeter
6. Android/TV/Burglar detection system
7. Ciseco OpenKontrol Gateway
8. WarmDirt
9. homA
10. St Jude Medical

ST. JUDE MEDICAL
MORE CONTROL. LESS RISK.

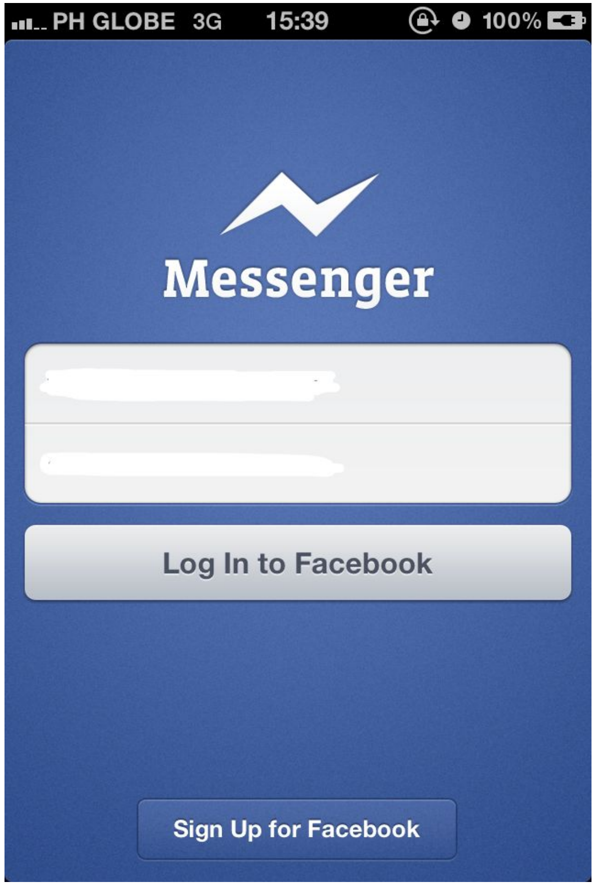


Ciseco OpenKontrol Gateway

谁正在使用MQTT协议？

移动端应用:

1. Facebook Messenger



- Facebook Messenger 使用MQTT协议，使得手机与手机之间的通信更加快捷，节省手机电源和带宽
- MQTT可以使用在iOS iPhone，iPad，Android 和 Windows 应用程序中。

MQTT-SN (MQTT For Sensor Networks)

- **MQTT-SN v1.2** -- MQTT for 传感器网络 V1.2 协议规格书
  它是一个基于非TCP/IP协议的（例如：ZigBee协议），专门为嵌入式设备提供消息通信模式的协议

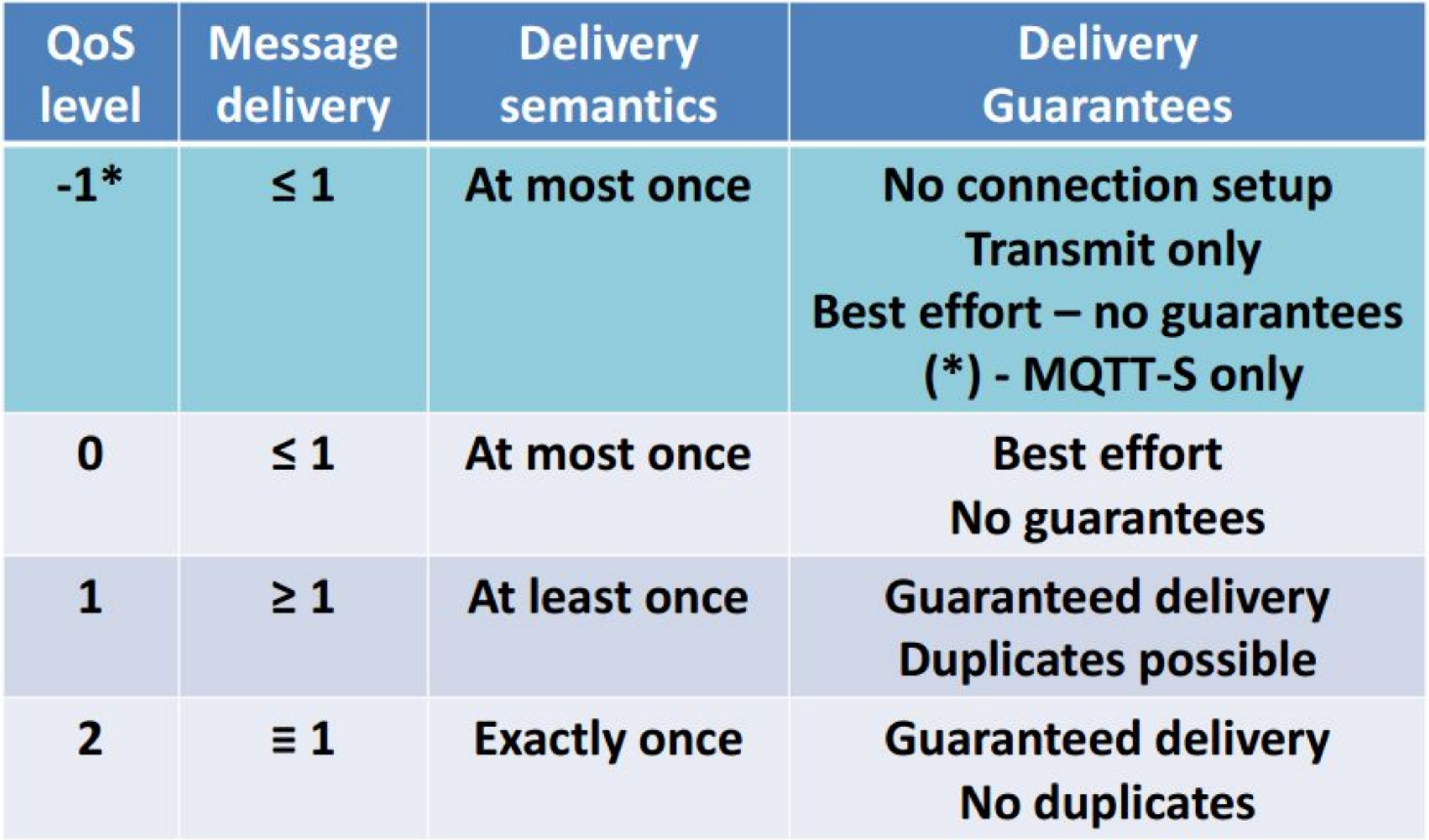
MQTT-S 综述

- MQTT-S协议设计和MQTT协议很相似
  - 例如：使用MQTT 语义
- 传感器网络中每个节点是客户端，这些客户端通过网关与基于IP网络的MQTT代理进行通讯
- 网关可以在MQTT-S 与MQTT之间传输数据
- 专为无线传感器网络架构及数据传输而设计的协议

MQTT vs MQTT-S

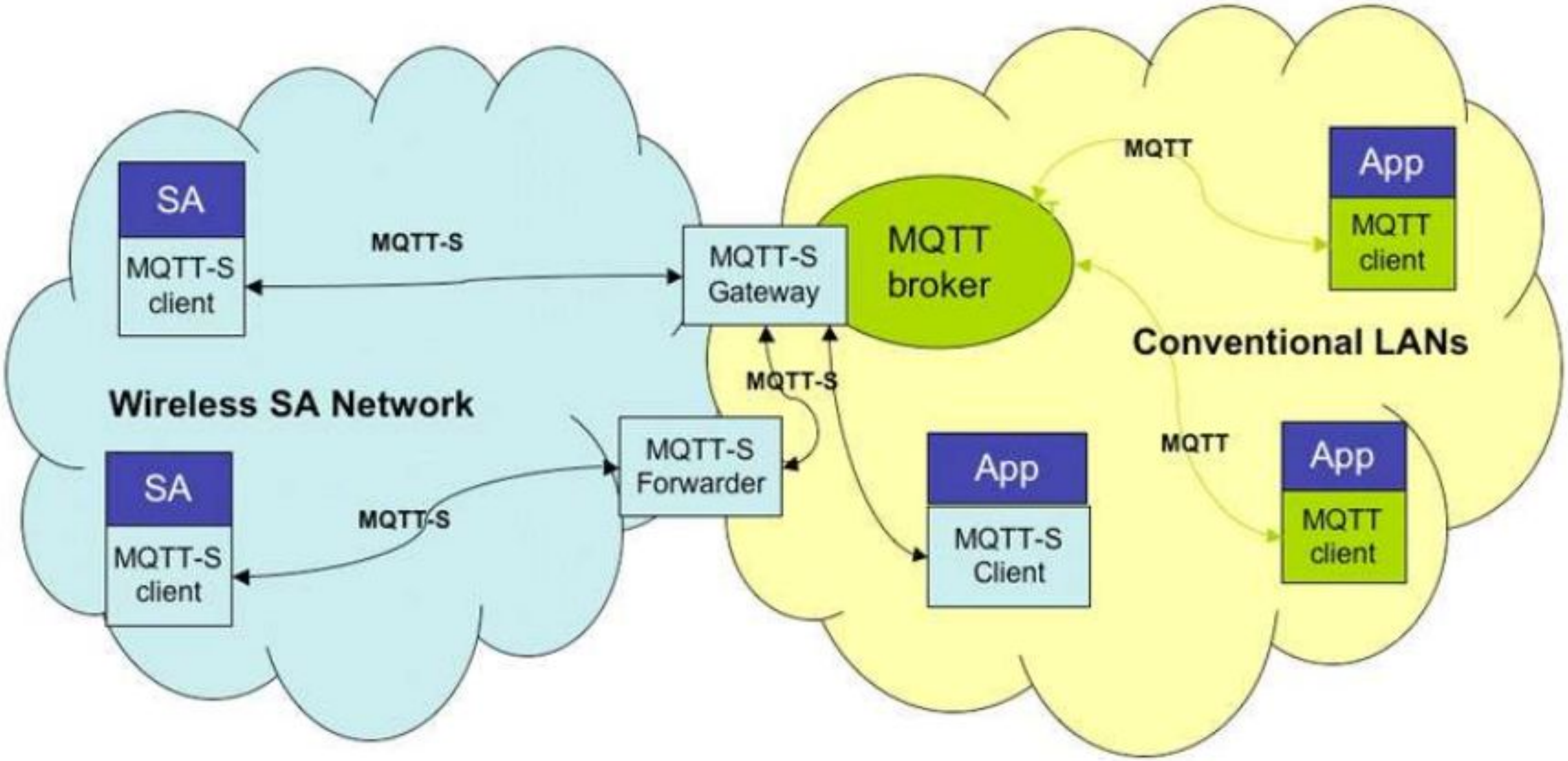
	MQTT	MQTT-S
Transport type	Reliable point to point streams	Unreliable datagrams
Communication	TCP/IP	Non-IP or UDP
Networking	Ethernet, WiFi, 3G	ZigBee, Bluetooth, RF
Min message size	2 bytes - PING	1 byte
Max message size	≤ 24MB	< 128 bytes (*)
Battery-operated		√
Sleeping clients		√
QoS: -1 “dumb client”		√
Gateway auto-discovery & fallbacks		√

"简单客户端" QoS = -1

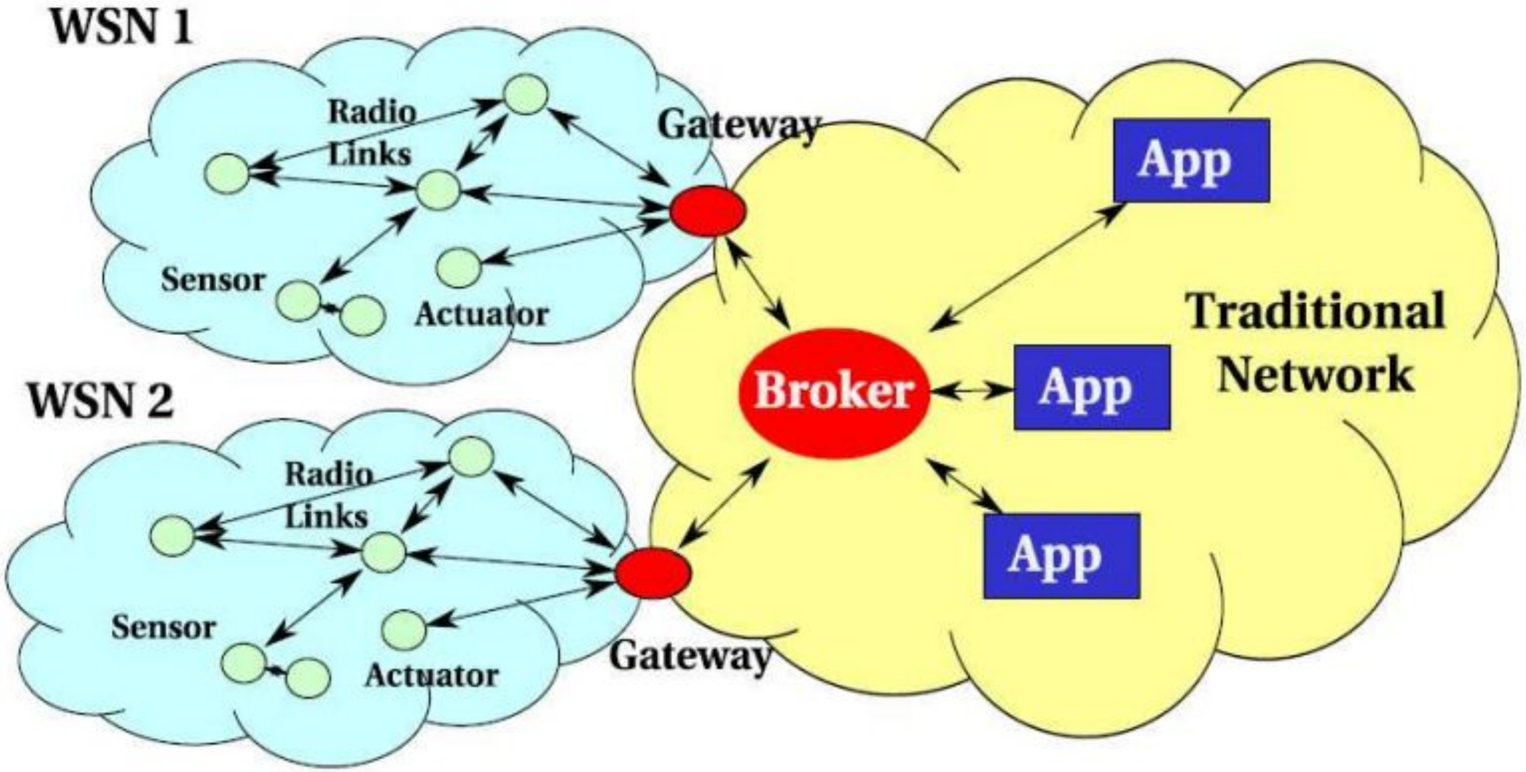


QoS level	Message delivery	Delivery semantics	Delivery Guarantees
-1*	≤ 1	At most once	No connection setup Transmit only Best effort – no guarantees (*) - MQTT-S only
0	≤ 1	At most once	Best effort No guarantees
1	≥ 1	At least once	Guaranteed delivery Duplicates possible
2	≡ 1	Exactly once	Guaranteed delivery No duplicates

MQTT-S 网关 <--> MQTT 代理



MQTT-S 网关 <--> MQTT 代理



支持MQTT协议的服务器

- IBM Websphere MQ Telemetry
- IBM MessageSight
- IBM Integration Bus
- Mosquitto
- Eclipse Paho
- Eurotech Everywhere Device Cloud
- Xively
- eMQTT
- m2m.io
- webMethods Nirvana Messaging
- RabbitMQ
- Apache ActiveMQ
- Apache Apollo
- Moquette
- HiveMQ
- Mosca
- Litmus Automation Loop

IBM BPM and WebSphere Message Broker V8

Christopher Frank – WebSphere Connectivity

Bill Hahn – BPM Solution Architect

HiveMQ

paho

Mosquitto

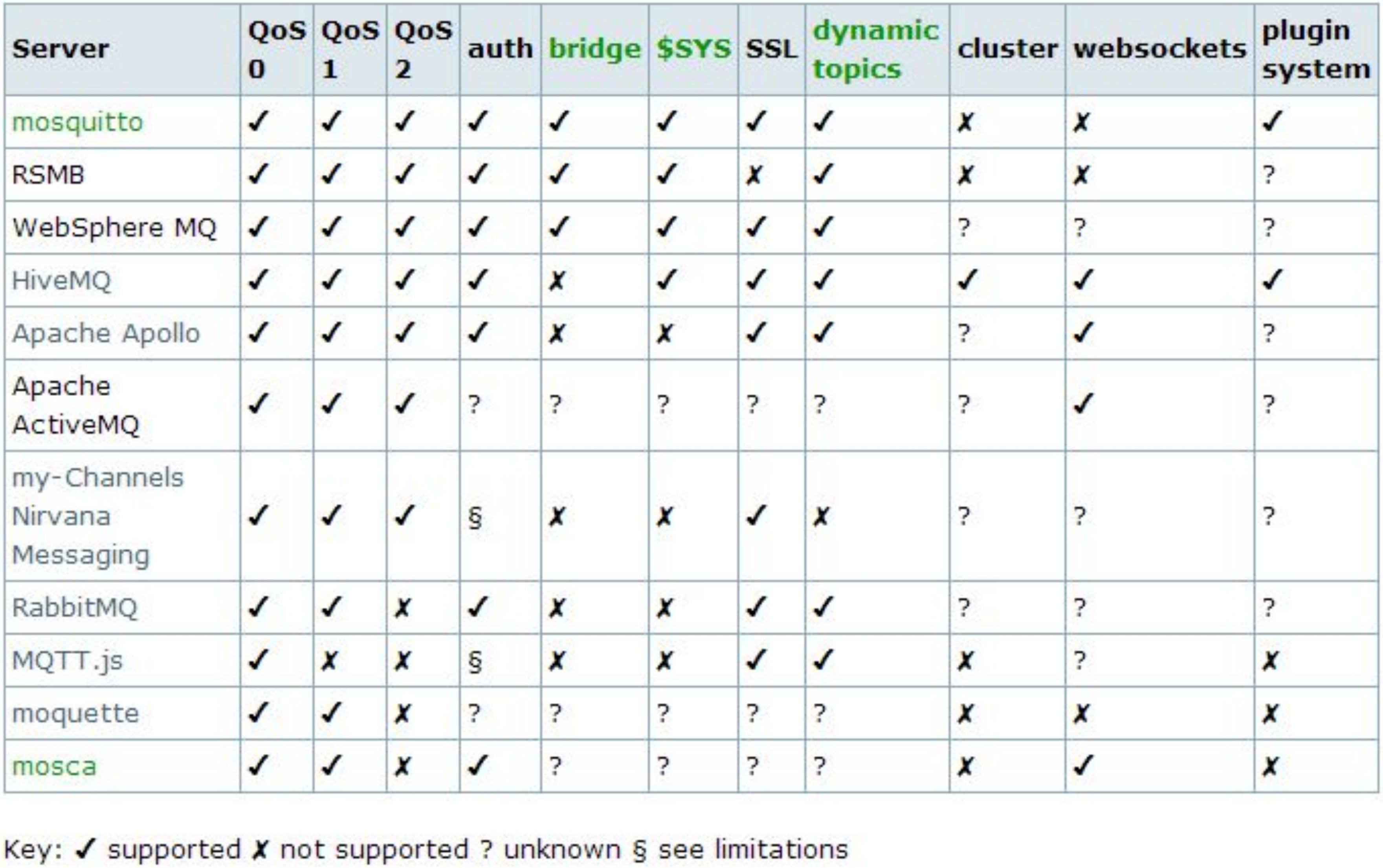
An Open Source MQTT v3.1 Broker

ActiveMQ™

RabbitMQ™

Messaging that just works

支持MQTT协议的服务器 性能对比图



Server	QoS 0	QoS 1	QoS 2	auth	bridge	$SYS	SSL	dynamic topics	cluster	websockets	plugin system
mosquitto	✓	✓	✓	✓	✓	✓	✓	✓	✗	✗	✓
RSMB	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	?
WebSphere MQ	✓	✓	✓	✓	✓	✓	✓	✓	?	?	?
HiveMQ	✓	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓
Apache Apollo	✓	✓	✓	✓	✗	✗	✓	✓	?	✓	?
Apache ActiveMQ	✓	✓	✓	?	?	?	?	?	?	✓	?
my-Channels Nirvana Messaging	✓	✓	✓	§	✗	✗	✓	✗	?	?	?
RabbitMQ	✓	✓	✗	✓	✗	✗	✓	✓	?	?	?
MQTT.js	✓	✗	✗	§	✗	✗	✓	✓	✗	?	✗
moquette	✓	✓	✗	?	?	?	?	?	✗	✗	✗
mosca	✓	✓	✗	✓	?	?	?	?	✗	✓	✗

Key: ✓ supported ✗ not supported ? unknown § see limitations

支持MQTT协议的客户端

Device-Specific :

- Arduino
- mbed
- mbed (simple port of the Arduino pubsubclient)
- Nanode
- Netduino
- M2MQTT (works with .Net Micro Framework)

Actionscript :

- as3MQTT

C :

- Eclipse Paho
- libmosquitto
- libemqtt - an embedded C client

C++ :

- libmosquittopp

Clojure :

- Machine Head

Dart :

- mqtt.dart

Delphi :

- TMQTTClient

Erlang :

- erlmqtt
- mqtt4erl
- my-mqtt4erl - updated fork of mqtt4erl

支持MQTT协议的客户端

Java :

- Eclipse Paho
- Xenqtt Includes a client library, mock broker for unit/integration testing, and applications to support enterprise needs like using a cluster of servers as a single client, an HTTP gateway, etc.
- MeQanTT
- Fusesource mqtt-client
- moquette
- "MA9B" zip of 1/2 dozen mobile clients source code. Includes Android-optimized Java source that works with Android notifications, based on Paho
- IA92 - deprecated IBM IA92 support pack, use Eclipse Paho GUI client instead. A useful MQTT Java swing GUI for publishing & subscribing. The Eclipse Paho GUI is identical but uses newer client code

MQTT Clients Librarys

Javscript / Node.js :

- Eclipse Paho HTML5 JavaScript for MQTT over WebSocket.
- mqtt.js
- node_mqtt_client
- IBM-provided PhoneGap / Apache Cordova MQTT plug-in for Android - JavaScript API is identical to Eclipse Paho HTML5 JavaScript
- mosquitto websocket client (deprecated, use Eclipse Paho)
- Ascoltatori - a node.js pub/sub library that allows access to Redis, AMQP, MQTT and ZeroMQ with the same API.

LotusScript :

- MQTT from LotusScript

Lua :

- Eclipse Paho Lua client
- mqtt_lua (deprecated use Paho)

MQTT Clients Librarys

.NET / dotNET :

- MqttDotNet
- nMQTT
- M2MQTT

Perl :

- net-mqtt-perl
- anyevent-mqtt-perl
- WebSphere-MQTT-Client

PHP :

- phpMQTT
- Mosquitto-PHP

Objective-C :

- mqttIO-objC
- libmosquitto - via wrappers
- MQTTKit
- "MA9B" zip of 1/2 dozen mobile clients source code including Objective-C

Python :

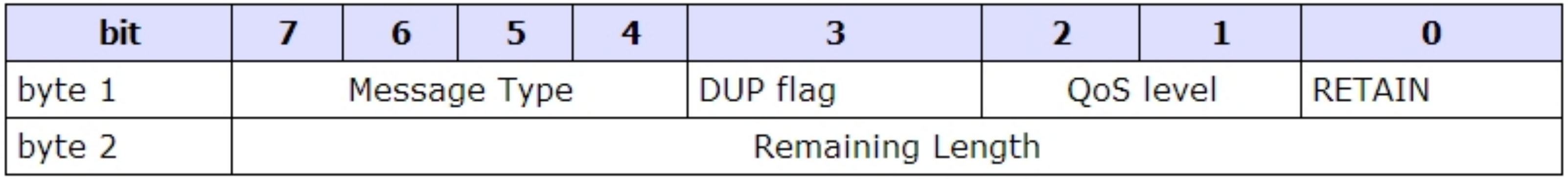
- Eclipse Paho Python client - originally the mosquitto Python client
- python-mosquitto (deprecated use Paho code)
- nyamuk
- MQTT for twisted python

Ruby :

- ruby-mqtt
- em-mqtt

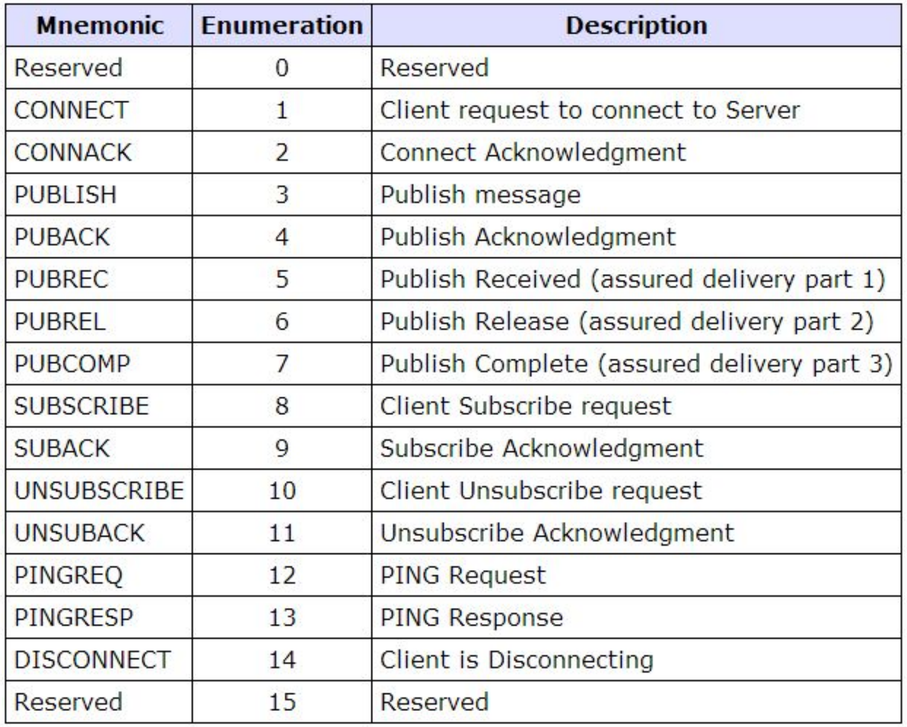
消息格式

Fixed Header



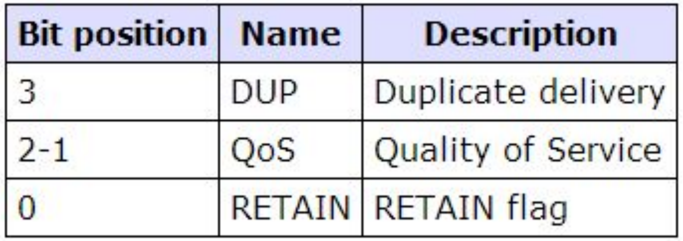
bit	7	6	5	4	3	2	1	0
byte 1	Message Type				DUP flag	QoS level		RETAIN
byte 2	Remaining Length							

Message Type



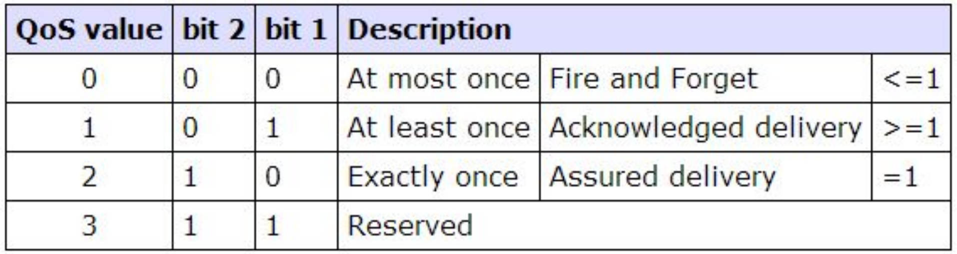
Mnemonic	Enumeration	Description
Reserved	0	Reserved
CONNECT	1	Client request to connect to Server
CONNACK	2	Connect Acknowledgment
PUBLISH	3	Publish message
PUBACK	4	Publish Acknowledgment
PUBREC	5	Publish Received (assured delivery part 1)
PUBREL	6	Publish Release (assured delivery part 2)
PUBCOMP	7	Publish Complete (assured delivery part 3)
SUBSCRIBE	8	Client Subscribe request
SUBACK	9	Subscribe Acknowledgment
UNSUBSCRIBE	10	Client Unsubscribe request
UNSUBACK	11	Unsubscribe Acknowledgment
PINGREQ	12	PING Request
PINGRESP	13	PING Response
DISCONNECT	14	Client is Disconnecting
Reserved	15	Reserved

DUP Flag



Bit position	Name	Description
3	DUP	Duplicate delivery
2-1	QoS	Quality of Service
0	RETAIN	RETAIN flag

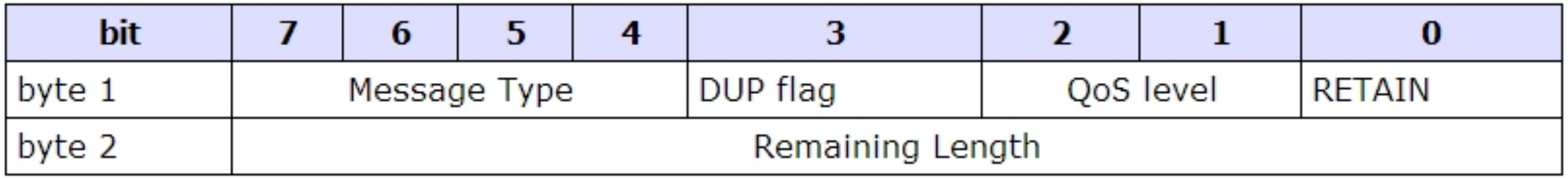
QoS



QoS value	bit 2	bit 1	Description		
0	0	0	At most once	Fire and Forget	<=1
1	0	1	At least once	Acknowledged delivery	>=1
2	1	0	Exactly once	Assured delivery	=1
3	1	1	Reserved		

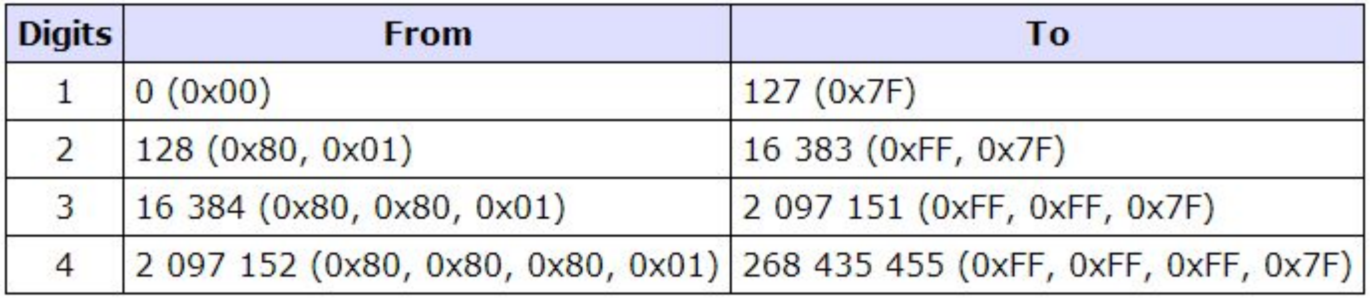
消息格式

Fixed Header



bit	7	6	5	4	3	2	1	0
byte 1	Message Type				DUP flag	QoS level		RETAIN
byte 2	Remaining Length							

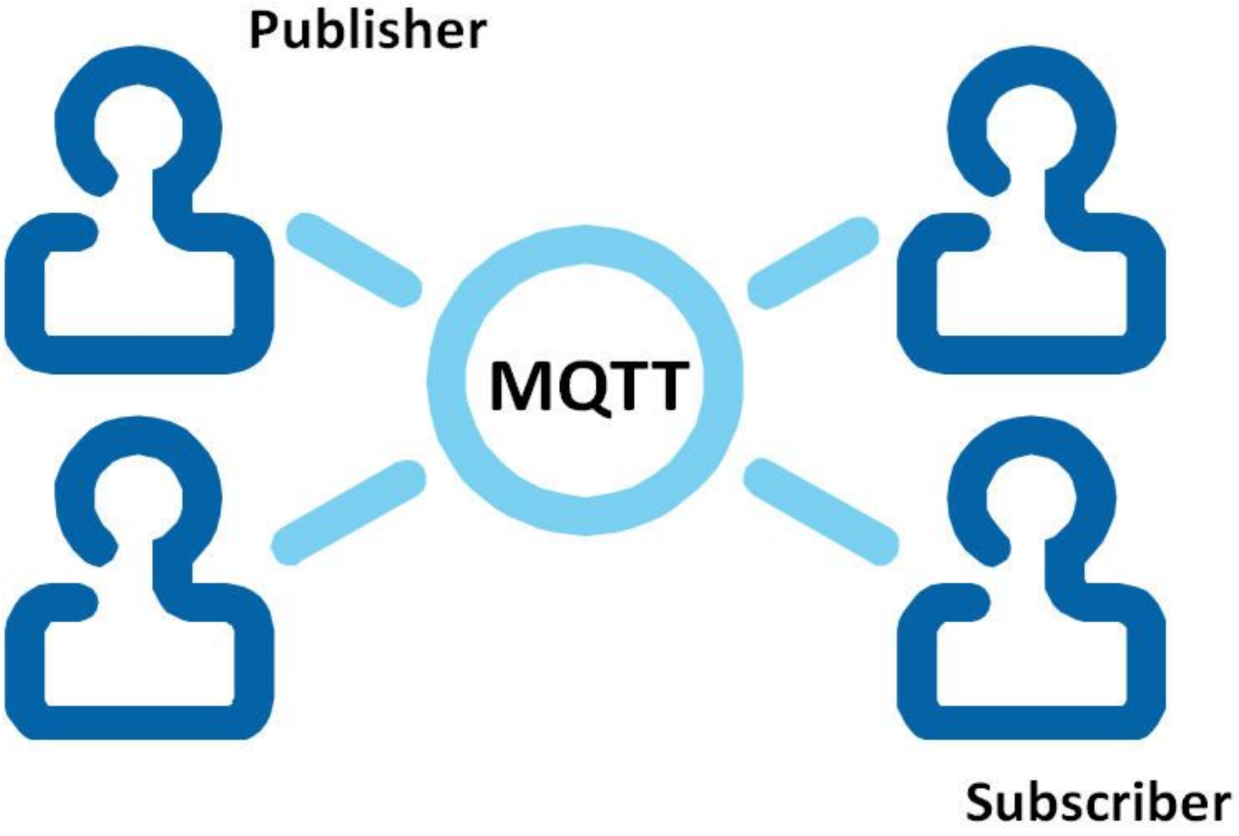
Remaining Length



Digits	From	To
1	0 (0x00)	127 (0x7F)
2	128 (0x80, 0x01)	16 383 (0xFF, 0x7F)
3	16 384 (0x80, 0x80, 0x01)	2 097 151 (0xFF, 0xFF, 0x7F)
4	2 097 152 (0x80, 0x80, 0x80, 0x01)	268 435 455 (0xFF, 0xFF, 0xFF, 0x7F)

消息命令列表

- CONNECT
- CONNACK
- PUBLISH
- PUBACK
- PUBREC
- PUBREL
- PUBCOMP
- SUBSCRIBE
- SUBACK
- UNSUBSCRIBE
- UNSUBACK
- PINGREC
- PINGRESP
- DISCONNECT



MQTT Message	4-bit code	Description
CONNECT	1	Client request to connect to Server
CONNACK	2	Connect Acknowledgment
PUBLISH	3	Publish message
PUBACK	4	Publish Acknowledgment
PUBREC	5	Publish Received (assured delivery part 1)
PUBREL	6	Publish Release (assured delivery part 2)
PUBCOMP	7	Publish Complete (assured delivery part 3)
SUBSCRIBE	8	Client Subscribe request
SUBACK	9	Subscribe Acknowledgment
UNSUBSCRIBE	10	Client Unsubscribe request
UNSUBACK	11	Unsubscribe Acknowledgment
PINGREC	12	PING Request
PINGRESP	13	PING Response
DISCONNECT	14	Client is Disconnecting

一个简单的样例（服务器与客户端）

服务器端

Mosquitto Broker v3.1

它是一个开源的MQTT 代理

- mosquitto -- the broker
- mosquitto.conf -- broker configuration
- mosquitto_passwd -- tool for managing mosquitto password files
- mosquitto_tls -- very rough cheat sheet for helping with SSL/TLS
- mosquitto_pub -- command line client for publishing
- mosquitto_sub -- command line client for subscribing

客户端

Paho MQTT Client 来自于Eclipse 物联网工作组

- C client
- C++ client
- Java client
- JavaScript client
- Lua client
- Python client

样例：介绍Mosquitto 服务器/消息代理

- 1883端口 -- the standard unencrypted MQTT port and can be used with any MQTT client.
- 8883和8884端口 -- using certificate based SSL/TLS encryption(TLS v1.2) and require client support to connect. In both cases should use the certificate authority file mosquitto.org.crt to verify the server connection.
- 8883端口 -- allows unrestricted connections.
- 8884端口 -- requires clients to provide their own certificate to authenticate their connection.
- 8885端口 -- it is the same as 8883 but using TLSv1 instead of TLSv1.2.

样例：用Mosquitto作为消息代理

如何安装Mosquitto在Debian Linux 系统中

首先我们需要导入respository package 签名的密钥：

```
wget http://repo.mosquitto.org/debian/mosquitto-repo.gpg.key
sudo apt-key add mosquitto-repo.gpg.key
```

使APT可以使用respository：

```
cd /etc/apt/sources.list.d/
sudo wget http://repo.mosquitto.org/debian/mosquitto-stable.list
```

更新APT 信息：

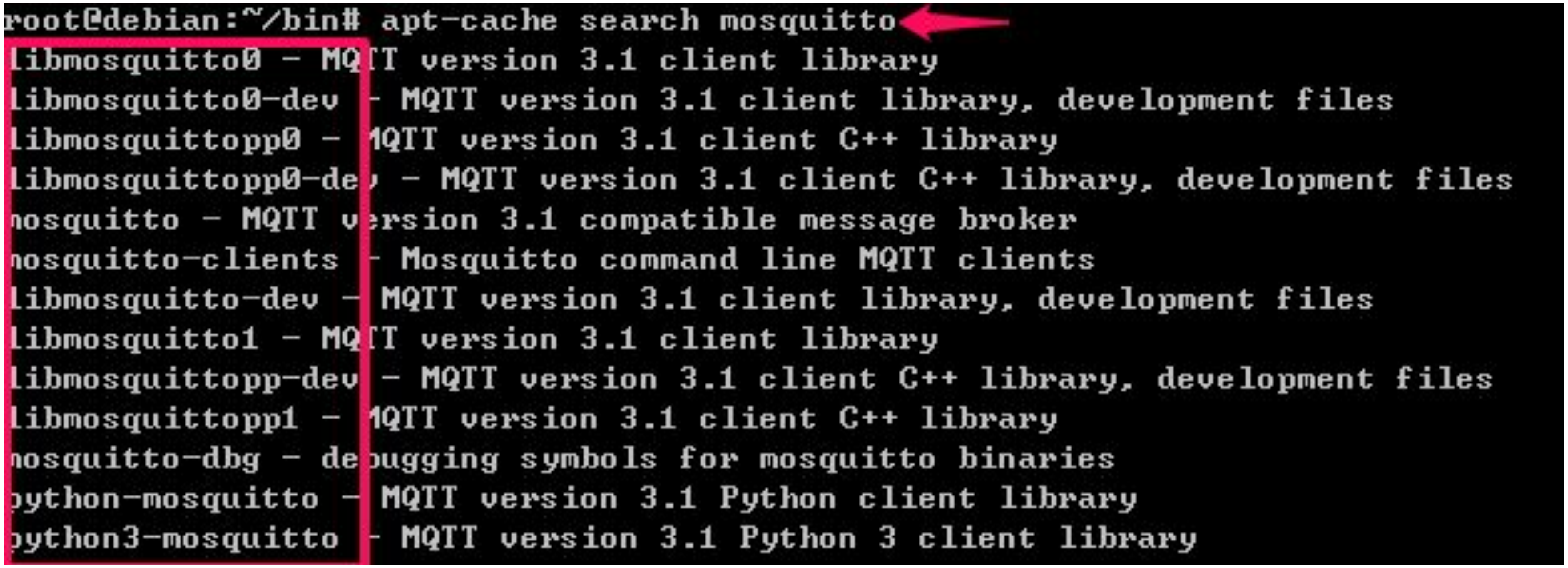
```
apt-get update
```

样例：用Mosquitto作为消息代理

查询什么样的mosquitto 包可以被使用：

```
apt-cache search mosquitto
```

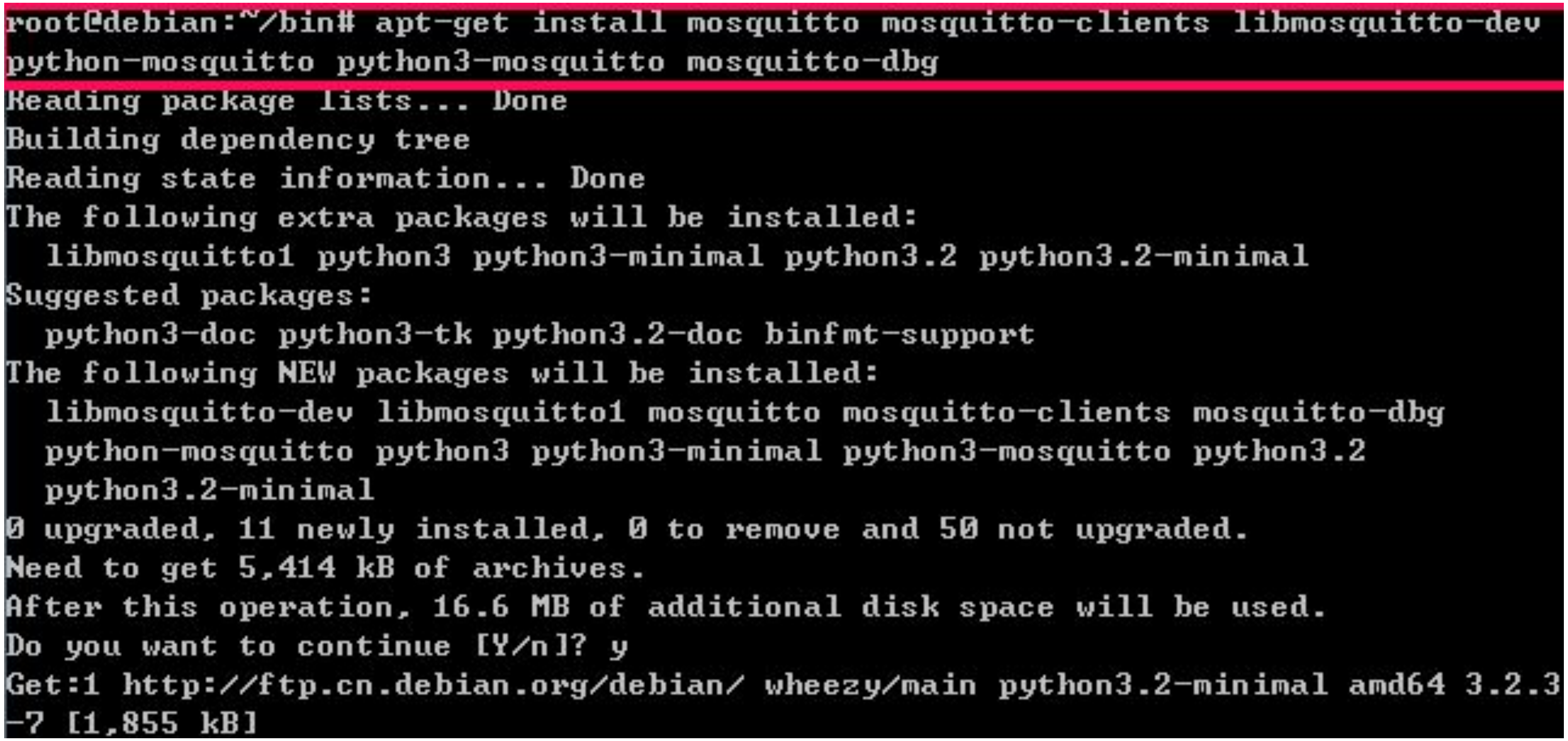
查询结果将会是：



```
root@debian:~/bin# apt-cache search mosquitto
libmosquitto0 - MQTT version 3.1 client library
libmosquitto0-dev - MQTT version 3.1 client library, development files
libmosquittopp0 - MQTT version 3.1 client C++ library
libmosquittopp0-dev - MQTT version 3.1 client C++ library, development files
mosquitto - MQTT version 3.1 compatible message broker
mosquitto-clients - Mosquitto command line MQTT clients
libmosquitto-dev - MQTT version 3.1 client library, development files
libmosquitto1 - MQTT version 3.1 client library
libmosquittopp-dev - MQTT version 3.1 client C++ library, development files
libmosquittopp1 - MQTT version 3.1 client C++ library
mosquitto-dbg - debugging symbols for mosquitto binaries
python-mosquitto - MQTT version 3.1 Python client library
python3-mosquitto - MQTT version 3.1 Python 3 client library
```

样例：用Mosquitto作为消息代理

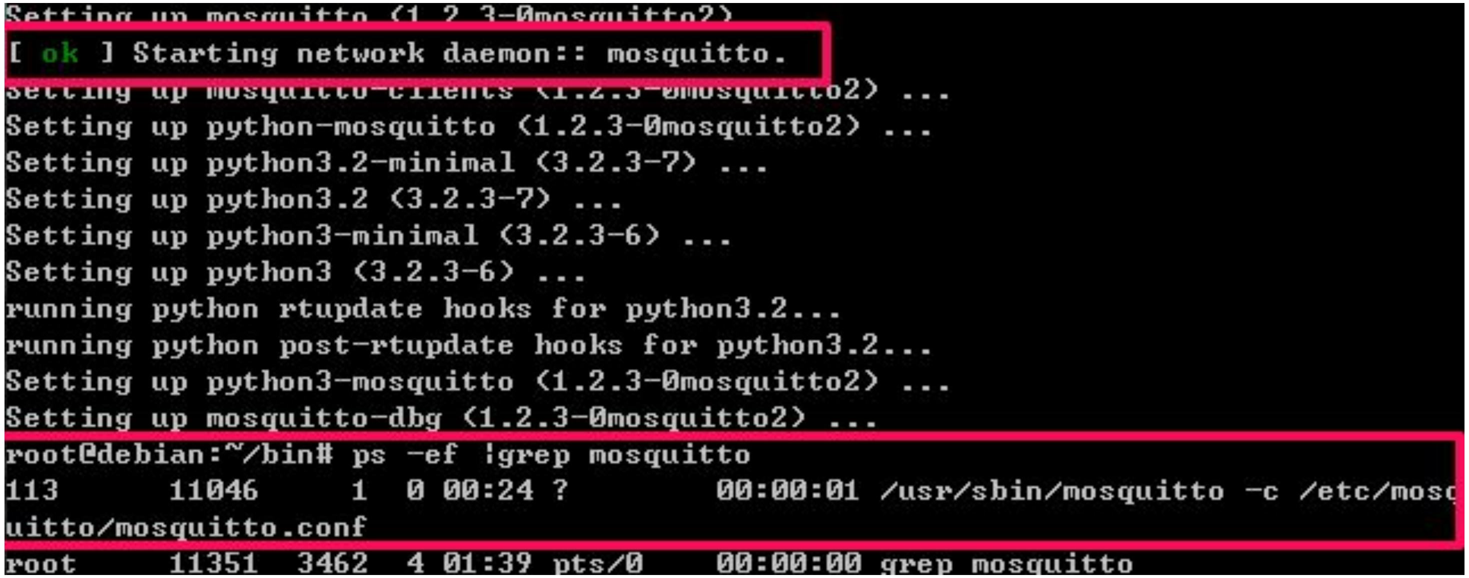
然后安装mosquitto



```
root@debian:~/bin# apt-get install mosquitto mosquitto-clients libmosquitto-dev
python-mosquitto python3-mosquitto mosquitto-dbg
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following extra packages will be installed:
  libmosquitto1 python3 python3-minimal python3.2 python3.2-minimal
Suggested packages:
  python3-doc python3-tk python3.2-doc binfmt-support
The following NEW packages will be installed:
  libmosquitto-dev libmosquitto1 mosquitto mosquitto-clients mosquitto-dbg
  python-mosquitto python3 python3-minimal python3-mosquitto python3.2
  python3.2-minimal
0 upgraded, 11 newly installed, 0 to remove and 50 not upgraded.
Need to get 5,414 kB of archives.
After this operation, 16.6 MB of additional disk space will be used.
Do you want to continue [Y/n]? y
Get:1 http://ftp.cn.debian.org/debian/ wheezy/main python3.2-minimal amd64 3.2.3
-7 [1,855 kB]
```

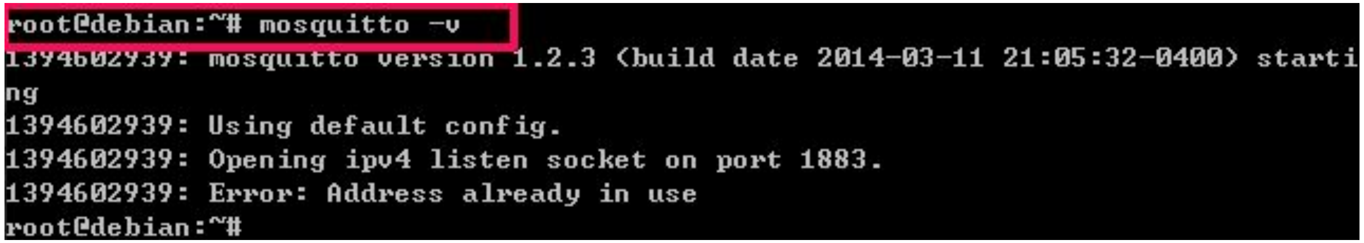
样例：用Mosquitto作为消息代理

安装mosquitto完成后，将会显示如下信息：



```
Setting up mosquitto (1.2.3-0mosquitto2)
[ ok ] Starting network daemon:: mosquitto.
Setting up mosquitto-clients (1.2.3-0mosquitto2) ...
Setting up python-mosquitto (1.2.3-0mosquitto2) ...
Setting up python3.2-minimal (3.2.3-7) ...
Setting up python3.2 (3.2.3-7) ...
Setting up python3-minimal (3.2.3-6) ...
Setting up python3 (3.2.3-6) ...
running python rtupdate hooks for python3.2...
running python post-rtupdate hooks for python3.2...
Setting up python3-mosquitto (1.2.3-0mosquitto2) ...
Setting up mosquitto-dbg (1.2.3-0mosquitto2) ...
root@debian:~/bin# ps -ef |grep mosquitto
113       11046     1  0 00:24 ?        00:00:01 /usr/sbin/mosquitto -c /etc/mosq
uitto/mosquitto.conf
root      11351  3462  4 01:39 pts/0    00:00:00 grep mosquitto
```

当运行 "mosquitto -v "命令，将会显示如下信息，这就表示安装成功了

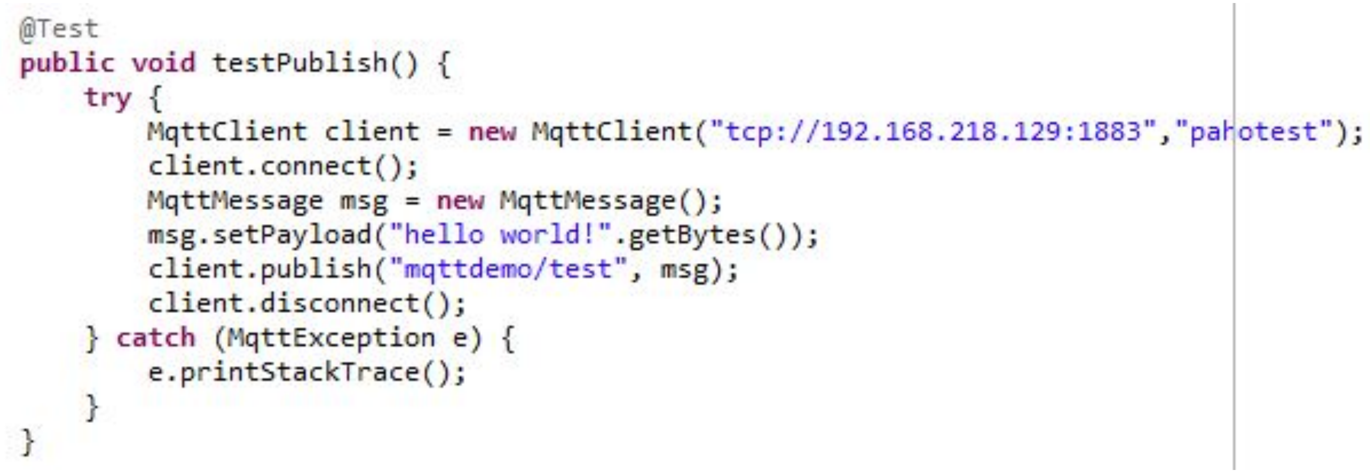


```
root@debian:~# mosquitto -v
1394602939: mosquitto version 1.2.3 (build date 2014-03-11 21:05:32-0400) starti
ng
1394602939: Using default config.
1394602939: Opening ipv4 listen socket on port 1883.
1394602939: Error: Address already in use
root@debian:~#
```

测试mosquitto的命令：mosquitto_sub -h test.mosquitto.org -t "#" -v

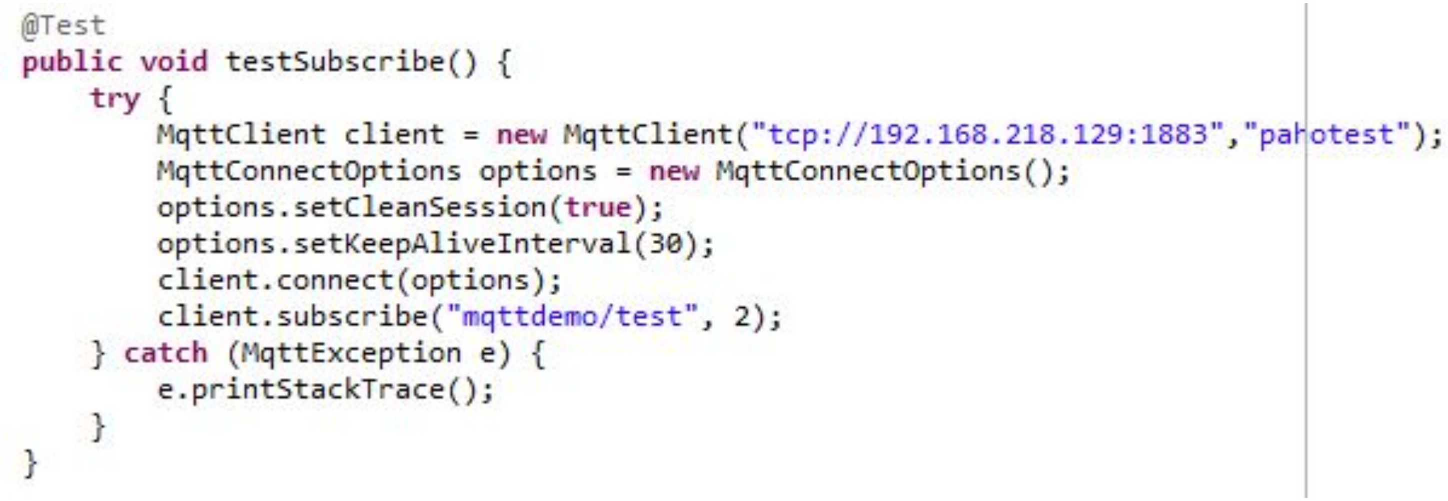
样例：客户端

发布消息



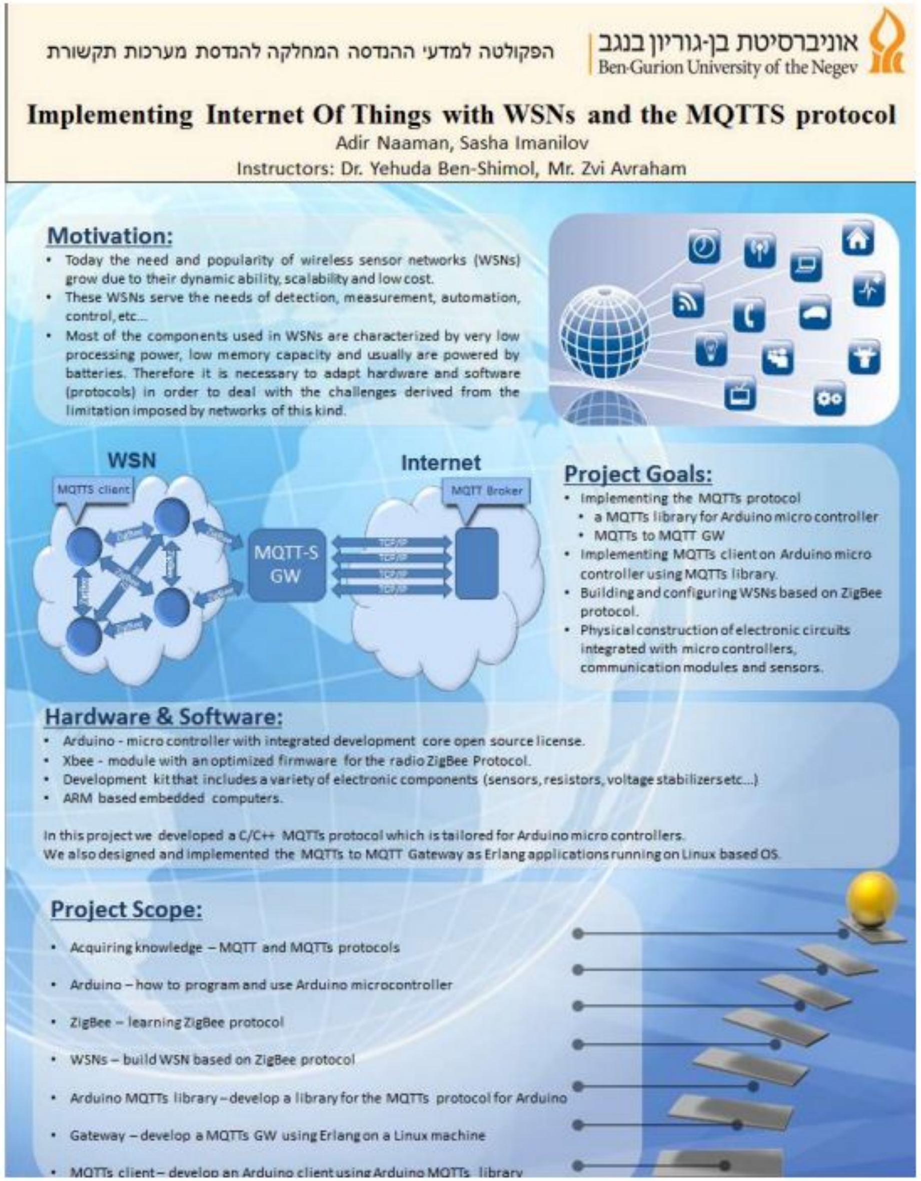
```java
@Test
public void testPublish() {
    try {
        MqttClient client = new MqttClient("tcp://192.168.218.129:1883","pahotest");
        client.connect();
        MqttMessage msg = new MqttMessage();
        msg.setPayload("hello world!".getBytes());
        client.publish("mqttdemo/test", msg);
        client.disconnect();
    } catch (MqttException e) {
        e.printStackTrace();
    }
}
```

订阅消息



```java
@Test
public void testSubscribe() {
    try {
        MqttClient client = new MqttClient("tcp://192.168.218.129:1883","pahotest");
        MqttConnectOptions options = new MqttConnectOptions();
        options.setCleanSession(true);
        options.setKeepAliveInterval(30);
        client.connect(options);
        client.subscribe("mqttdemo/test", 2);
    } catch (MqttException e) {
        e.printStackTrace();
    }
}
```

使用WSNs 和 MQTT-S 协议实现物联网(IoT)



הפקולטה למדעי ההנדסה המחלקה להנדסת מערכות תקשורת

אוניברסיטת בן-גוריון בנגב
Ben-Gurion University of the Negev

Implementing Internet Of Things with WSNs and the MQTTS protocol

Adir Naaman, Sasha Imanilov
Instructors: Dr. Yehuda Ben-Shimol, Mr. Zvi Avraham

Motivation:

- Today the need and popularity of wireless sensor networks (WSNs) grow due to their dynamic ability, scalability and low cost.
- These WSNs serve the needs of detection, measurement, automation, control, etc…
- Most of the components used in WSNs are characterized by very low processing power, low memory capacity and usually are powered by batteries. Therefore it is necessary to adapt hardware and software (protocols) in order to deal with the challenges derived from the limitation imposed by networks of this kind.

Project Goals:

- Implementing the MQTTs protocol
  - a MQTTs library for Arduino micro controller
  - MQTTs to MQTT GW
- Implementing MQTTs client on Arduino micro controller using MQTTs library.
- Building and configuring WSNs based on ZigBee protocol.
- Physical construction of electronic circuits integrated with micro controllers, communication modules and sensors.

Hardware & Software:

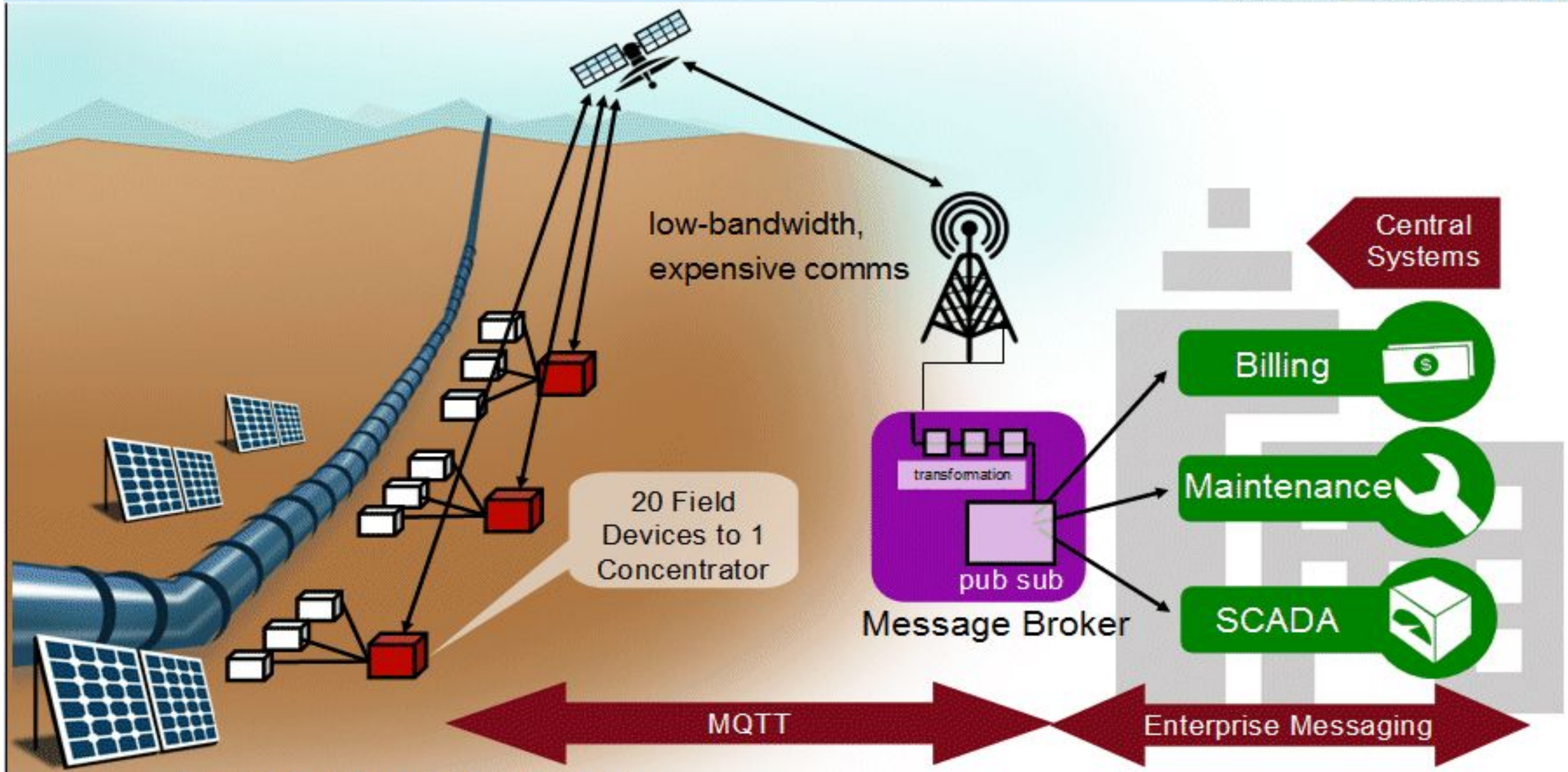
- Arduino - micro controller with integrated development core open source license.
- Xbee - module with an optimized firmware for the radio ZigBee Protocol.
- Development kit that includes a variety of electronic components (sensors, resistors, voltage stabilizers etc…)
- ARM based embedded computers.

In this project we developed a C/C++ MQTTs protocol which is tailored for Arduino micro controllers.
We also designed and implemented the MQTTs to MQTT Gateway as Erlang applications running on Linux based OS.

Project Scope:

- Acquiring knowledge – MQTT and MQTTs protocols
- Arduino – how to program and use Arduino microcontroller
- ZigBee – learning ZigBee protocol
- WSNs – build WSN based on ZigBee protocol
- Arduino MQTTs library – develop a library for the MQTTs protocol for Arduino
- Gateway – develop a MQTTs GW using Erlang on a Linux machine
- MQTTs client – develop an Arduino client using Arduino MQTTs library

案 例



Scalability for whole pipeline!

Network traffic much lower - events pushed to/from devices and report by exception

Network cost reduced

Lower CPU utilization

Broken out of the SCADA prison – data accessible to other applications

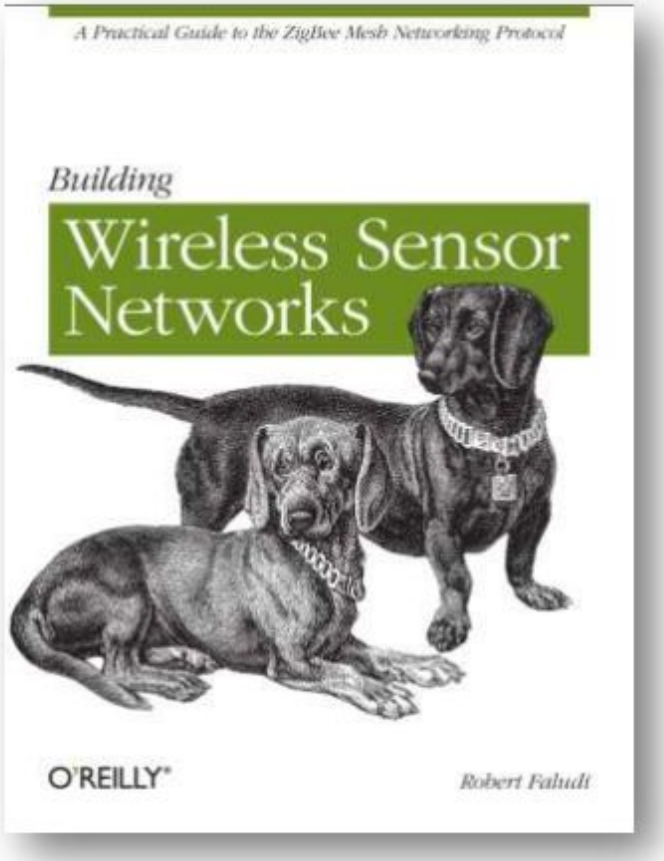
案例 : Nike + iPod

experience
NIKE+
+ iPod
iPod
Music
Videos
Photos
Podcasts
Extras
Radio
Nike + iPod
Settings
Shuffle Songs
MENU

如何学习基于ZigBee协议的开发?

BWSN: 书 + 开发工具包

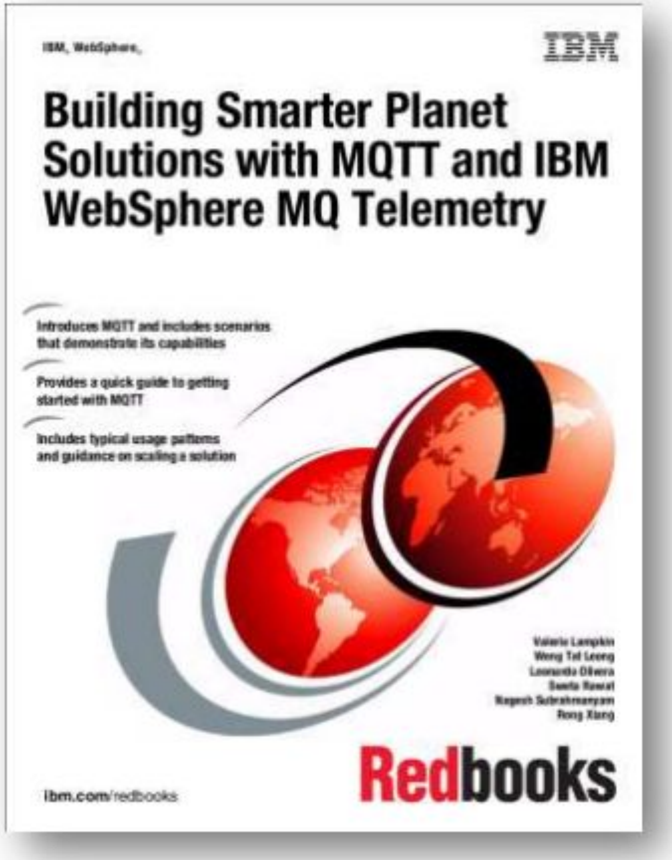
Book



Sparkfun kit ~ $115

MQTT 书籍：

IBM MQTT Redbook



IBM, WebSphere,

IBM

Building Smarter Planet Solutions with MQTT and IBM WebSphere MQ Telemetry

Introduces MQTT and includes scenarios that demonstrate its capabilities

Provides a quick guide to getting started with MQTT

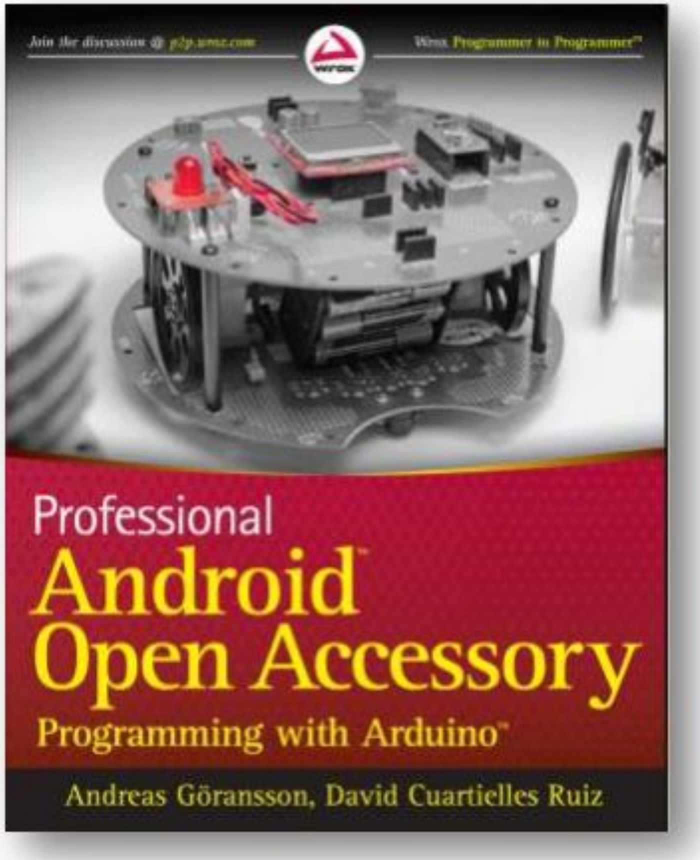
Includes typical usage patterns and guidance on scaling a solution

Valerie Lampkin
Weng Tat Leong
Leonardo Olivera
Sweta Rawat
Nagesh Subrahmanyam
Rong Xiang

ibm.com/redbooks

Redbooks

Chapter 3 – talks about MQTT



Join the discussion @ p2p.wrox.com

Wrox Programmer to Programmer™

Professional Android™ Open Accessory Programming with Arduino™

Andreas Göransson, David Cuartielles Ruiz

参考资源：

官方网站:

- MQTT official site

协议规格书:

- MQTT v3.1 Protocol Specification
- MQTT-S v1.2 Protocol Specification

论文:

- MQTT-S -- A Pub/Sub protocol for Wireless Sensor Networks

一个简单的样本项目:

- Controlling the house lighting via MQTT

参考资源：

物联网资源:

- special report the internet of things
- smarter sensors
- the value of privacy
- will the internet of things crush it
- whats coming next the internet of everything
- conferences marchoctober 2014
- help with building the next big thing
- setting the stage for the internet of things
- yenkuang chen improving lives
- ask the expert the internet of things
- tech news the internet of things
- books of interest march 2014

维基百科:

- MQ Telemetry Transport

参考资源：

Youtube网站的视频资源:

- Android Home Automation Demo | Voice + NFC
- Fully Automated Digital Home Systems
- Enterprise exploitation of the internet of things (IoT) with BlackBerry 10
- MQTT + BeagleBone Black + Augmented Reality = FUN!
- MQTT Starfighter, JazzHub, BlueMix and live Scaling Out
- Starfighter - IBM MessageSight and MQTT for multiplayer gaming
- M2Mqtt : MQTT client testing
- IBM Cluster Code Off - CICS monitoring application using IBM MessageSight, MQTT and Arduino
- IBM Cluster Code Off - The Big Blue Line mobile geo-location race application
- London Green Hackathon: Kindle Energy Dashboard
- MQTT FOR multi-users gaming

Twitter 资源:

- https://twitter.com/mqttorg

参考资源：

开源项目:

- http://mosquitto.org/
- http://mosquitto.org/download/
- http://www.eclipse.org/paho/
- http://git.eclipse.org/c/paho/org.eclipse.paho.mqtt.c.git/
- http://git.eclipse.org/c/paho/org.eclipse.paho.mqtt.cpp.git/
- http://git.eclipse.org/c/paho/org.eclipse.paho.mqtt.java.git/
- http://git.eclipse.org/c/paho/org.eclipse.paho.mqtt.javascript.git/
- https://github.com/fusesource/mqtt-client
- https://github.com/TomoakiYAMAGUCHI/MQTT-S
- http://build.eclipse.org/technology/paho/C/
- https://repo.eclipse.org/content/repositories/paho-snapshots/
- https://repo.eclipse.org/content/repositories/paho-releases/
- https://github.com/dpslwk/OpenKontrol-Gateway
- http://shop.ciseco.co.uk/openkontrol-gateway-starter/

Thanks! Questions?

Contact:

Eric Xiao

Email: ericssonxiao@gmail.com

Twitter: @ericssonxiao

LinkedIn:

http://www.linkedin.com/in/ericssonxiao