第二讲
MQTT（II）
Lecture 2
MQTT (II)

明玉瑞 Yurui Ming

yrming@gmail.com

声明
Disclaimer

- 本讲义在准备过程中由于时间所限，所用材料来源并未规范标示引用来源。所引材料仅用于教学所用，作者无意侵犯原著者之知识版权，所引材料之知识版权均归原著者所有；若原著者介意之，请联系作者更正及删除。

  The time limit during the preparation of these slides incurs the situation that not all the sources of the used materials (texts or images) are properly referenced or clearly manifested. However, all materials in these slides are solely for teaching and the author is with no intention to infringe the copyright bestowed on the original authors or manufacturers. All credits go to corresponding IP holders. Please address the author for any concern for remedy including deletion.

术语
Terminology

- 在MQTT术语中，一些词汇有在MQTT上下文特别的意义，总结如下：
  - 网络连接：MQTT中，网络连接必须以双向方式，有序地、完整地传输字节流
  - 应用消息：应用消息一般包含载荷数据、QoS字段、属性集合，以及主题名称
  - 会话：会话是客户端与服务器或代理之间的有状态交互。一些会话与网络连接有相同的生命期，但其他会话可能横跨客户端和服务器之间的多个连续的网络连接
- In the MQTT terminology, some are of special meanings in the MQTT context, just summarized as below:
  - Network Connection: it provides the means to send an ordered, lossless, stream of bytes in both directions.
  - Application Message: it contains payload data, a Quality of Service (QoS), a collection of Properties, and a Topic Name.
  - Session: A stateful interaction between a Client and a Server. Some Sessions last only as long as the Network Connection, others can span multiple consecutive Network Connections between a Client and a Server.

术语
Terminology

- 在MQTT术语中，一些词汇有在MQTT上下文特别的意义，总结如下：
  - 订阅：订阅包括主题过滤器和最大QoS。订阅与单个会话相关联，但一个会话可以包含多个订阅。会话中的每个订阅都有不同的主题过滤器。
  - 共享订阅：共享订阅包括主题过滤器和最大QoS。共享订阅可以与多个会话相关联，以允许更广泛的消息交换模式。与共享订阅匹配的应用程序消息仅发送到与这些会话之一关联的客户端。一个会话可以订阅多个共享订阅，并且可以包含共享订阅和非共享订阅。
- In the MQTT terminology, some are of special meanings in the MQTT context, just summarized as below:
  - Subscription: A Subscription comprises a Topic Filter and a maximum QoS. A Subscription is associated with a single Session. A Session can contain more than one Subscription. Each Subscription within a Session has a different Topic Filter.
  - Shared Subscription: A Shared Subscription comprises a Topic Filter and a maximum QoS. A Shared Subscription can be associated with more than one Session to allow a wider range of message exchange patterns. An Application Message that matches a Shared Subscription is only sent to the Client associated with one of these Sessions. A Session can subscribe to more than one Shared Subscription and can contain both Shared Subscriptions and Subscriptions which are not shared.

数据表示
Data Representation

- 整数：无论是双字节整数，还是四字节整数，当在网络上传输时，均为大字节序。

  Integer: Both double-byte and quad-byte integers are of big-endian when presented on the network.

- UTF-8编码字符串：MQTT控制包中的文本字段是UTF-8编码的字符串。这些字符串以两字节整数长度字段为前缀，该字段给出了UTF-8编码字符串本身的字节数。因此，UTF-8编码字符串的最大大小为 65,535 字节。

  UTF-8 Encoded String: Text fields within the MQTT Control Packets are encoded as UTF-8 strings. Each of these strings is prefixed with a Two Byte Integer length field that gives the number of bytes in a UTF-8 encoded string itself. Consequently, the maximum size of a UTF-8 Encoded String is 65,535 bytes.

Bit	7	6	5	4	3	2	1	0
byte 1	String length MSB							
byte 2	String length LSB							
byte 3 ….	UTF-8 encoded character data, if length > 0.							

数据表示
Data Representation

- 可变字节整数：可变字节整数使用编码方案进行编码，该方案使用单个字节的值最大为127。较大的值按如下方式处理：每个字节的最低七位对数据进行编码，最高位用于指示表示中是否有字节跟随。因此，每个字节编码128个值和一个“连续位”。可变字节整数字段中的最大字节数为4。编码值必须使用表示值所需的最小字节数。下表展示了可变字节整数的编码大小范围。

  Variable Byte Integer: The Variable Byte Integer is encoded using an encoding scheme which uses a single byte for values up to 127. Larger values are handled as follows. The least significant seven bits of each byte encode the data, and the most significant bit is used to indicate whether there are bytes following in the representation. Thus, each byte encodes 128 values and a "continuation bit". The maximum number of bytes in the Variable Byte Integer field is four. The encoded value MUST use the minimum number of bytes necessary to represent the value. The following table shows the size of variable byte integer.

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)

数据表示
Data Representation

- 二进制数据：二进制数据由表示数据字节数的两字节整数长度表示，后跟该数目的二进制数。因此，二进制数据的长度被限制在 0 到 65,535 字节的范围内。

  Binary Data: Binary Data is represented by a Two Byte Integer length which indicates the number of data bytes, followed by that number of bytes. Thus, the length of Binary Data is limited to the range of 0 to 65,535 Bytes.

- UTF-8字符串对：UTF-8字符串对由两个UTF-8编码字符串组成。此数据类型用于保存名称-值对。 第一个字符串用作名称，第二个字符串包含值。两个字符串都必须符合UTF-8编码字符串的要求。 如果接收者（客户端或服务器）接收到不满足这些要求的字符串对，则它是格式错误的数据包。

  UTF-8 String Pair: UTF-8 String Pair consists of two UTF-8 Encoded Strings. This data type is used to hold name-value pairs. The first string serves as the name, and the second string contains the value. Both strings MUST comply with the requirements for UTF-8 Encoded Strings. If a receiver (Client or Server) receives a string pair which does not meet these requirements it is a Malformed Packet. .

MQTT控制包格式
MQTT Control Packet format

- MQTT协议通过以定义的方式交换一系列MQTT控制数据包来运行。一个MQTT控制包最多由三个部分组成，始终按以下顺序排列，如下图所示：

  The MQTT protocol operates by exchanging a series of MQTT Control Packets in a defined way. An MQTT Control Packet consists of up to three parts, always in the following order as shown below:

Fixed Header, present in all MQTT Control Packets
Variable Header, present in some MQTT Control Packets
Payload, present in some MQTT Control Packets

MQTT控制包固定报头
MQTT Control Packet Fixed Header

- 每个MQTT控制包都包含一个固定报头，如下图所示：

  Each MQTT Control Packet contains a Fixed Header as shown below:

Bit	7	6	5	4	3	2	1	0
byte 1	MQTT Control Packet type				Flags specific to each MQTT Control Packet type			
byte 2…	Remaining Length							

- 其中MQTT控制数据包类型位于字节 1，位[7-4]；固定报头中字节1的剩余位[3-0]包含特定于每个MQTT控制数据包类型的标志

  MQTT Control Packet type is positioned at byte 1, with bits 7-4; The remaining bits [3-0] of byte 1 in the Fixed Header contain flags specific to each MQTT Control Packet type

MQTT控制包固定报头
MQTT Control Packet Fixed Header

- 部分MQTT控制数据包类型如下图所示：

Part of the MQTT Control Packet type is shown as below:

Name	Value	Direction of flow	Description
Reserved	0	Forbidden	Reserved
CONNECT	1	Client to Server	Connection request
CONNACK	2	Server to Client	Connect acknowledgment
PUBLISH	3	Client to Server or vice versa	Publish message
PUBACK	4	Client to Server or vice versa	Publish acknowledgment (QoS 1)
…	…	…	…
SUBSCRIBE	8	Client to Server	Subscribe request
SUBACK	9	Server to Client	Subscribe acknowledgment
UNSUBSCRIBE	10	Client to Server	Unsubscribe request
…	…	…	…
DISCONNECT	14	Client to Server or vice versa	Disconnect notification
AUTH	15	Client to Server or vice versa	Authentication exchange

MQTT控制包固定报头
MQTT Control Packet Fixed Header

- 部分针对MQTT控制数据包类型的标志如下图所示：

Part of the flags to MQTT Control Packet type is shown as below:

MQTT Control Packet	Fixed Header flags	Bit 3	Bit 2	Bit 1	Bit 0
Reserved	-	-	-	-	-
CONNECT	Reserved	0	0	0	0
CONNACK	Reserved	0	0	0	0
PUBLISH	Used in MQTT 5.0	DUP	QoS		RETAIN
PUBACK	Reserved	0	0	0	0
…	…	…			
SUBSCRIBE	Reserved	0	0	1	0
SUBACK	Reserved	0	0	0	0
UNSUBSCRIBE	Reserved	0	0	1	0
…	…	…			
DISCONNECT	Reserved	0	0	0	0
AUTH	Reserved	0	0	0	0

MQTT控制包固定报头
MQTT Control Packet Fixed Header

- MQTT控制包结构中的剩余长度字段是一个可变字节整数，表示当前控制包中剩余的字节数，包括Variable Header和Payload中的数据。剩余长度不包括用于编码剩余长度字段本身的字节。MQTT控制包大小是包中的总字节数，它等于固定头部的长度加上剩余长度表示的值。注意，由于剩余长度字段本身采用可变字节整数编码，因此，固定头部的长度实际上并不事先固定。

  The Remaining Length is a variable byte integer that represents the number of bytes remaining within the current control packet, including data in the Variable Header and the Payload. The Remaining Length does not include the bytes used to encode the Remaining Length itself. The packet size is the total number of bytes in an MQTT Control Packet, this is equal to the length of the Fixed Header plus the Remaining Length. Notably, the remaining length field is encoded by using variable bytes, so the Fixed Header is not exactly pre-fixed as in other networking protocols.

MQTT控制包可变报头
MQTT Control Packet Variable Header

- 某些类型的MQTT控制数据包包含可变报头部分，它位于Fixed Header和Payload之间，其内容因数据包类型而异。Variable Header的Packet Identifier字段在几种数据包类型中很常见。

  Some types of MQTT Control Packet contain a Variable Header component. It resides between the Fixed Header and the Payload. The content of the Variable Header varies depending on the packet type. The Packet Identifier field of Variable Header is common in several packet types.

- 许多MQTT控制数据包类型的变量头组件包括一个两字节整数数据包标识符字段。这些MQTT控制数据包包括PUBLISH（其中QoS>0）、PUBACK、SUBSCRIBE、SUBACK、UNSUBSCRIBE、UNSUBACK等等。

  The Variable Header component of many of the MQTT Control Packet types includes a Two Byte Integer Packet Identifier field. These MQTT Control Packets are PUBLISH (where QoS > 0), PUBACK, PUBREC, SUBSCRIBE, SUBACK, UNSUBSCRIBE, UNSUBACK, etc.

MQTT控制包可变报头
MQTT Control Packet Variable Header

- 关于数据包标识符有如下注意事项：
  - 如果PUBLISH数据包的QoS值设置为0，则PUBLISH数据包不得包含数据包标识符。
  - 每次客户端发送一个新的SUBSCRIBE、UNSUBSCRIBE或PUBLISH（其中QoS>0）MQTT控制数据包时，它必须为其分配一个当前未使用的非零数据包标识符。
  - 每次服务器发送一个新的PUBLISH（QoS > 0）MQTT控制数据包时，它必须为其分配一个当前未使用的非零数据包标识符。
- Users should pay attention to the following note tips for Packet Identifier:
  - A PUBLISH packet MUST NOT contain a Packet Identifier if its QoS value is set to 0.
  - Each time a Client sends a new SUBSCRIBE, UNSUBSCRIBE,or PUBLISH (where QoS > 0) MQTT Control Packet it MUST assign it a non-zero Packet Identifier that is currently unused.
  - Each time a Server sends a new PUBLISH (with QoS > 0) MQTT Control Packet it MUST assign it a non zero Packet Identifier that is currently unused.

MQTT控制包可变报头
MQTT Control Packet Variable Header

- 关于数据包标识符有如下注意事项：
  - 只有在发送方处理了相应的确认数据包后，数据包标识符方可重新使用： 在PUBLISH的QoS为1的情况下，对应的是PUBACK；在PUBLISH的QoS为2的情况下，对应的是PUBCOMP或原因代码为128或更大的PUBREC。对于SUBSCRIBE或UNSUBSCRIBE，对应的是SUBACK或UNSUBACK。
  - 与PUBLISH、SUBSCRIBE和UNSUBSCRIBE数据包一起使用的数据包标识符形成了一个单独的、统一的标识符集，分别用于会话中的客户端和服务器。一个数据包标识符在任何时候都不能被多个命令使用。
- Users should pay attention to the following note tips for Packet Identifier:
  - The Packet Identifier becomes available for reuse after the sender has processed the corresponding acknowledgement packet: In the case of a QoS 1 PUBLISH, this is the corresponding PUBACK; in the case of QoS 2 PUBLISH it is PUBCOMP or a PUBREC with a Reason Code of 128 or greater. For SUBSCRIBE or UNSUBSCRIBE it is the corresponding SUBACK or UNSUBACK.
  - Packet Identifiers used with PUBLISH, SUBSCRIBE and UNSUBSCRIBE packets form a single, unified set of identifiers separately for the Client and the Server in a Session. A Packet Identifier cannot be used by more than one command at any time.

MQTT控制包可变报头
MQTT Control Packet Variable Header

- 关于数据包标识符有如下注意事项：
  - PUBACK、PUBREC、PUBREL或PUBCOMP数据包必须包含与最初发送的PUBLISH数据包相同的数据包标识符。SUBACK和UNSUBACK必须包含分别在相应的SUBSCRIBE和UNSUBSCRIBE数据包中使用的数据包标识符。
  - 客户端和服务器相互独立地分配数据包标识符。因此，客户端-服务器对可以使用相同的数据包标识符参与并发消息交换。
- Users should pay attention to the following note tips for Packet Identifier:
  - A PUBACK, PUBREC, PUBREL, or PUBCOMP packet MUST contain the same Packet Identifier as the PUBLISH packet that was originally sent. A SUBACK and UNSUBACK MUST contain the Packet Identifier that was used in the corresponding SUBSCRIBE and UNSUBSCRIBE packet respectively.
  - The Client and Server assign Packet Identifiers independently of each other. As a result, Client-Server pairs can participate in concurrent message exchanges using the same Packet Identifiers.

MQTT控制包可变报头
MQTT Control Packet Variable Header

- 可变报头中的最后一个字段是属性集合，其由属性长度和属性组成。。

  The last field in the Variable Header is a set of Properties, which is composed of a Property Length followed by the Properties.

- 属性长度采用可变字节整数编码。属性长度不包括用于对其自身进行编码的字节，但包括属性的长度。如果没有属性，则必须通过包括零属性长度来表示。

  The Property Length is encoded as a Variable Byte Integer. The Property Length does not include the bytes used to encode itself, but includes the length of the Properties. If there are no properties, this MUST be indicated by including a Property Length of zero.

- 属性由定义其用途和数据类型的被编码为可变字节整数的标识符组成，后跟一个值。原因代码为 0x81用于处理含对其数据包类型无效的标识符或包含不是指定数据类型的值的控制数据包是格式错误的数据包。具有不同标识符的属性的顺序没有意义。。

  A Property consists of an Identifier which defines its usage and data type and encoded as a Variable Byte Integer, followed by a value. Reason Code 0x81 is used to indicate a Control Packet which contains an Identifier which is not valid for its packet type, or contains a value not of the specified data type. There is no significance in the order of Properties with different Identifiers..

MQTT控制包载荷和原因代码
MQTT Control Packet Payload and Reason Code

- 一些MQTT控制数据包包含一个有效载核作为数据包的最后部分。如在PUBLISH数据包中的应用程序消息。

  Some MQTT Control Packets contain a Payload as the final part of the packet. For instance, in the PUBLISH packet this is the Application Message.

- 最后一个是原因代码，其是一个单字节无符号值，指示操作的结果。小于0x80 的原因代码表示操作成功完成，成功的正常原因代码为0。原因代码值为0x80或更大表示失败。CONNACK、PUBACK、PUBREC、PUBREL、PUBCOMP、DISCONNECT和AUTH控制包有一个单一的原因代码作为可变报头的一部分。 SUBACK 和 UNSUBACK 数据包在有效载荷中包含一个或多个原因代码的列表。

  The last one is Reason Code, which is a one byte unsigned value that indicates the result of an operation. Reason Codes less than 0x80 indicate successful completion of an operation. The normal Reason Code for success is 0. Reason Code values of 0x80 or greater indicate failure. The CONNACK, PUBACK, PUBREC, PUBREL, PUBCOMP, DISCONNECT and AUTH Control Packets have a single Reason Code as part of the Variable Header. The SUBACK and UNSUBACK packets contain a list of one or more Reason Codes in the Payload.

MQTT服务质量
QoS in MQTT

- 服务质量（QoS）是英文Quality of Service的缩写，在网络领域中表示的是通信线路的品质保证。在MQTT里存在着3个等级的QoS，在“发布者与代理之间”以及“代理与订阅者之间”都分别定义了不同的QoS等级，以异步的方式运行。特别地，当“代理与订阅者之间”的QoS小于“发布者与代理之间”信息交换的QoS时，“代理与订阅者之间”的QoS会被降级到指定的QoS。

  QoS is an initialism for Quality of Service, which in the networking field indicates the quality assurance of communications. There are 3 levels of QoS in MQTT, and different levels of QoS are defined between the publisher and the broker, and between the broker and the subscriber. QoS between these entities run asynchronously. In particular, when the QoS between the agent and the subscriber is less than that between the publisher and the broker, the QoS between the broker and the subscriber is downgraded to the specified QoS.

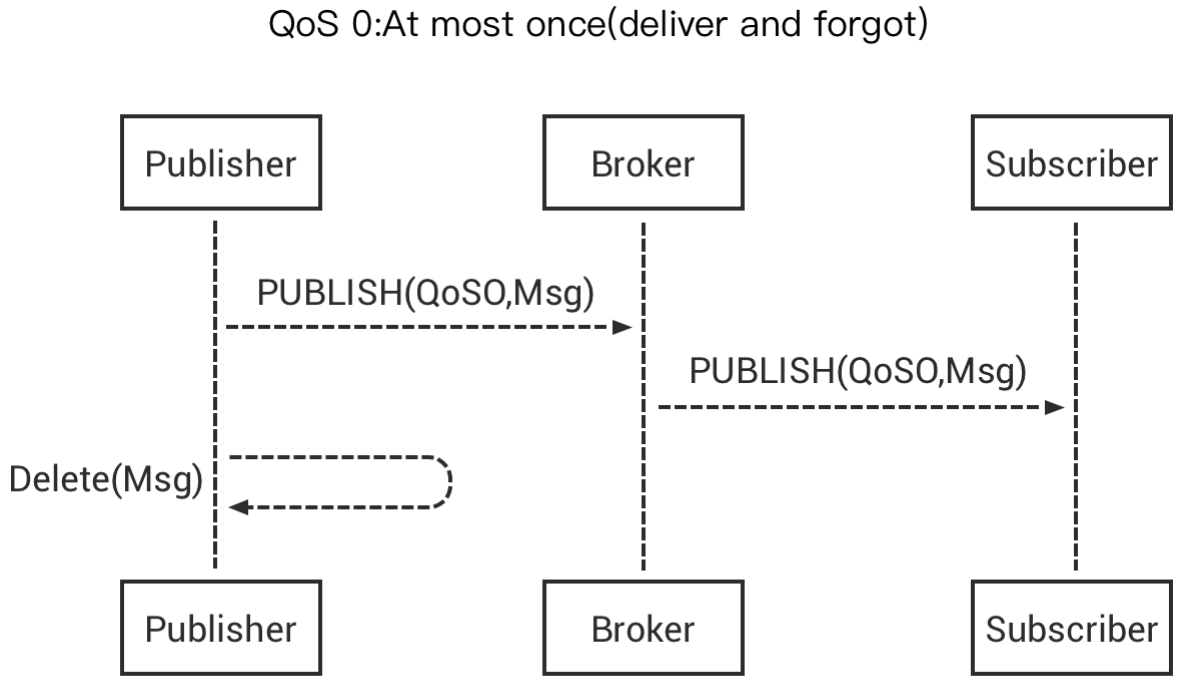
- QoS 0指的是最多发送一次消息，发送时遵循TCP/IP通信的尽力而为模式。

  QoS 0 refers to sending a message at most once, following the best effort pattern of TCP/IP communication.

MQTT服务质量
QoS in MQTT

- 对于QoS 0，消息有两种情况，即到达了代理处一次，或没有到达代理处。

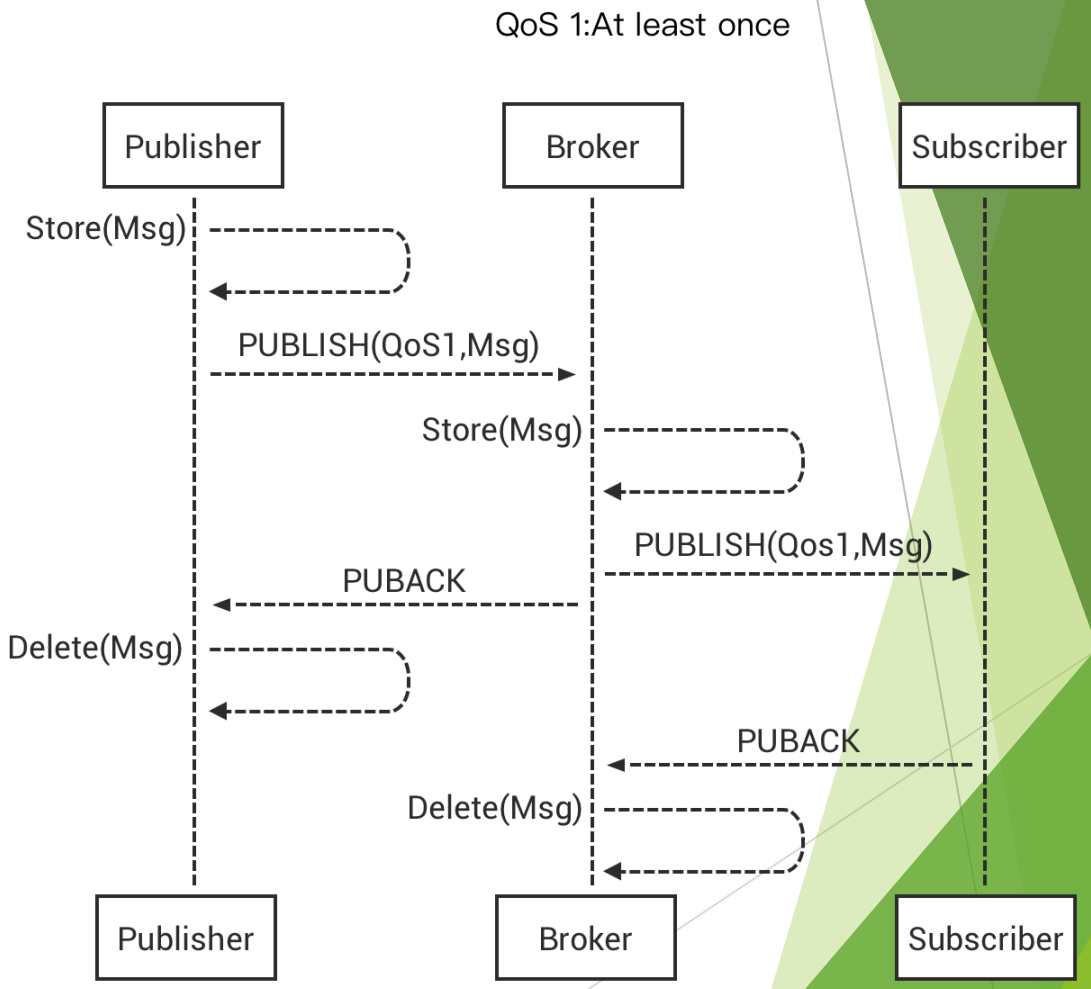
  For QoS 0, there are two consequences for the message, aka, the message reaches the broker once, or the message never reaches the broker.



MQTT服务质量
QoS in MQTT

- QoS 1指的是至多发送一次消息。消息接收者收到消息之后，会向发布者发送PUBACK消息响应。当消息发送者在发送后经一段时间仍没有收到PUBACK确认响应，则会重新发送该消息。

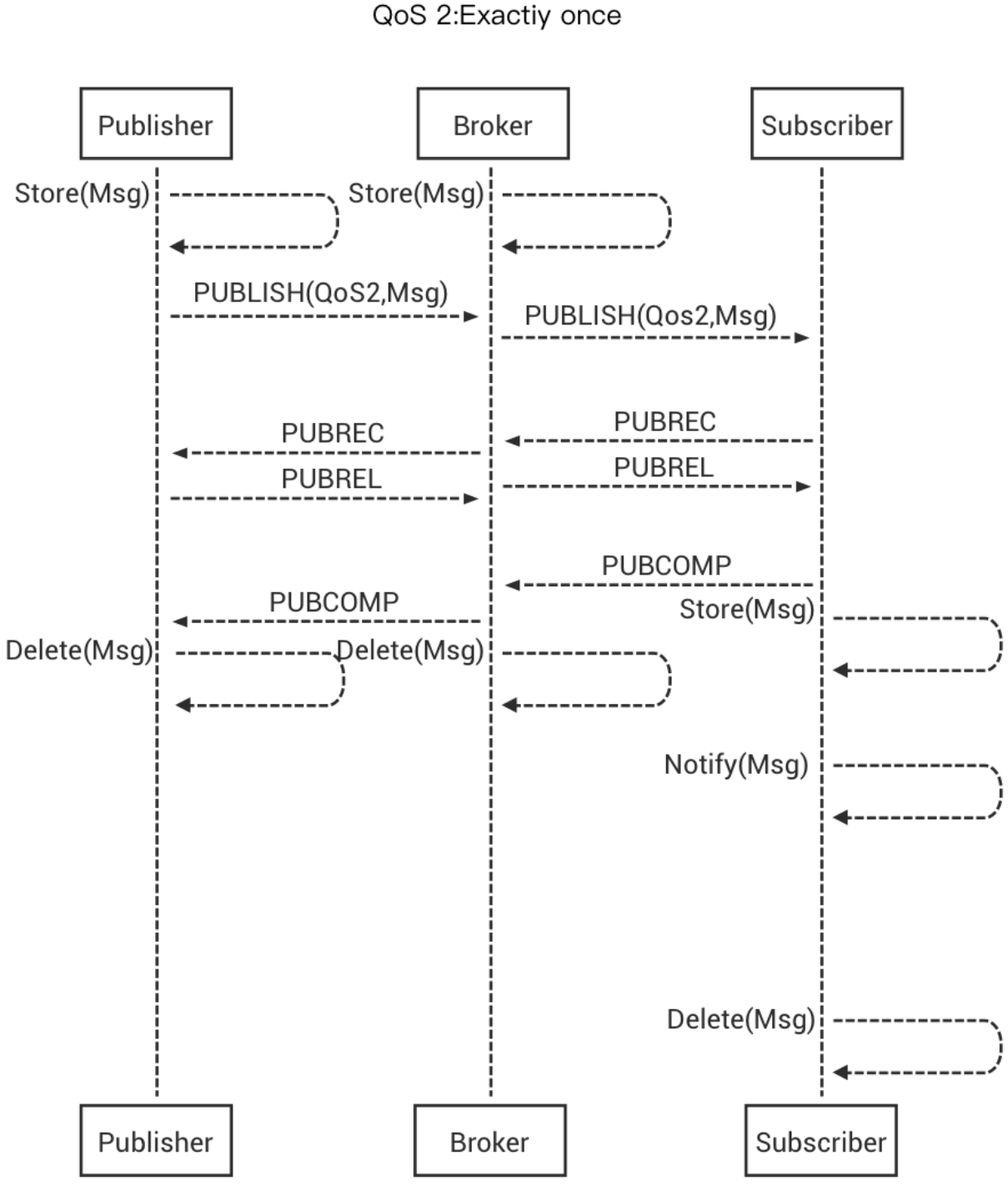
  QoS 1 means that the message is sent at most once. After the message receiver receives the message, it will send a PUBACK message response to the publisher. When the sender of the message does not receive a PUBACK confirmation response for a period of time after sending, the message will be resent.



MQTT服务质量
QoS in MQTT

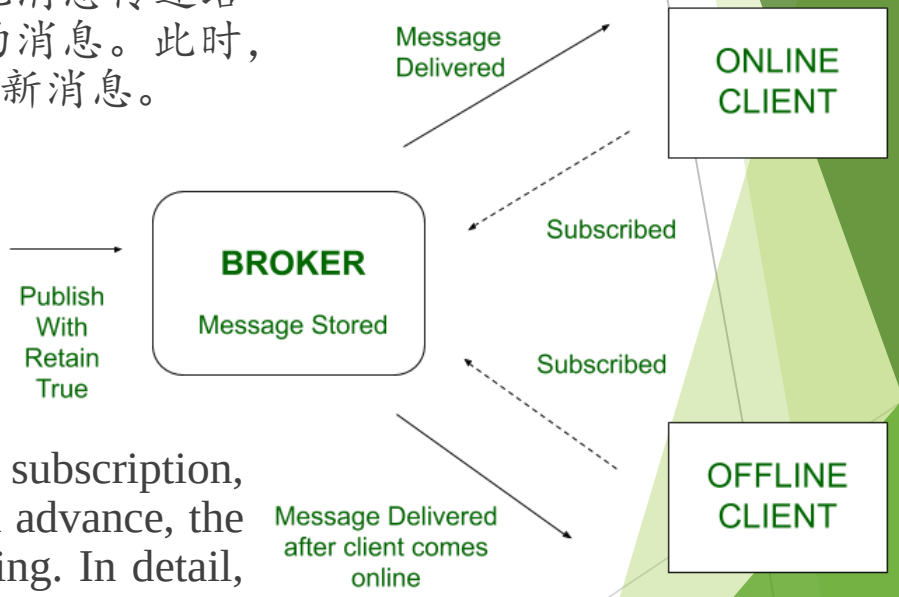
- QoS 2指的是唯一发送一次消息，其中消息包含消息ID。消息接收者收到消息之后，会将消息保存起来；然后向发布者发送PUBREC消息响应。消息发送者再向接收者发送PUBREL消息，然后接收者再向发送者发送PUBCOMP消息。

  QoS 2 refers to sending a message exactly once, where the message contains a message ID. After the message receiver receives the message, it stores the message and then sends a PUBREC message response to the publisher. Then the sender of the message sends a PUBREL message to the receiver, and the receiver sends a PUBCOMP message back to the sender.



MQTT Retain标识
MQTT Retain Flag

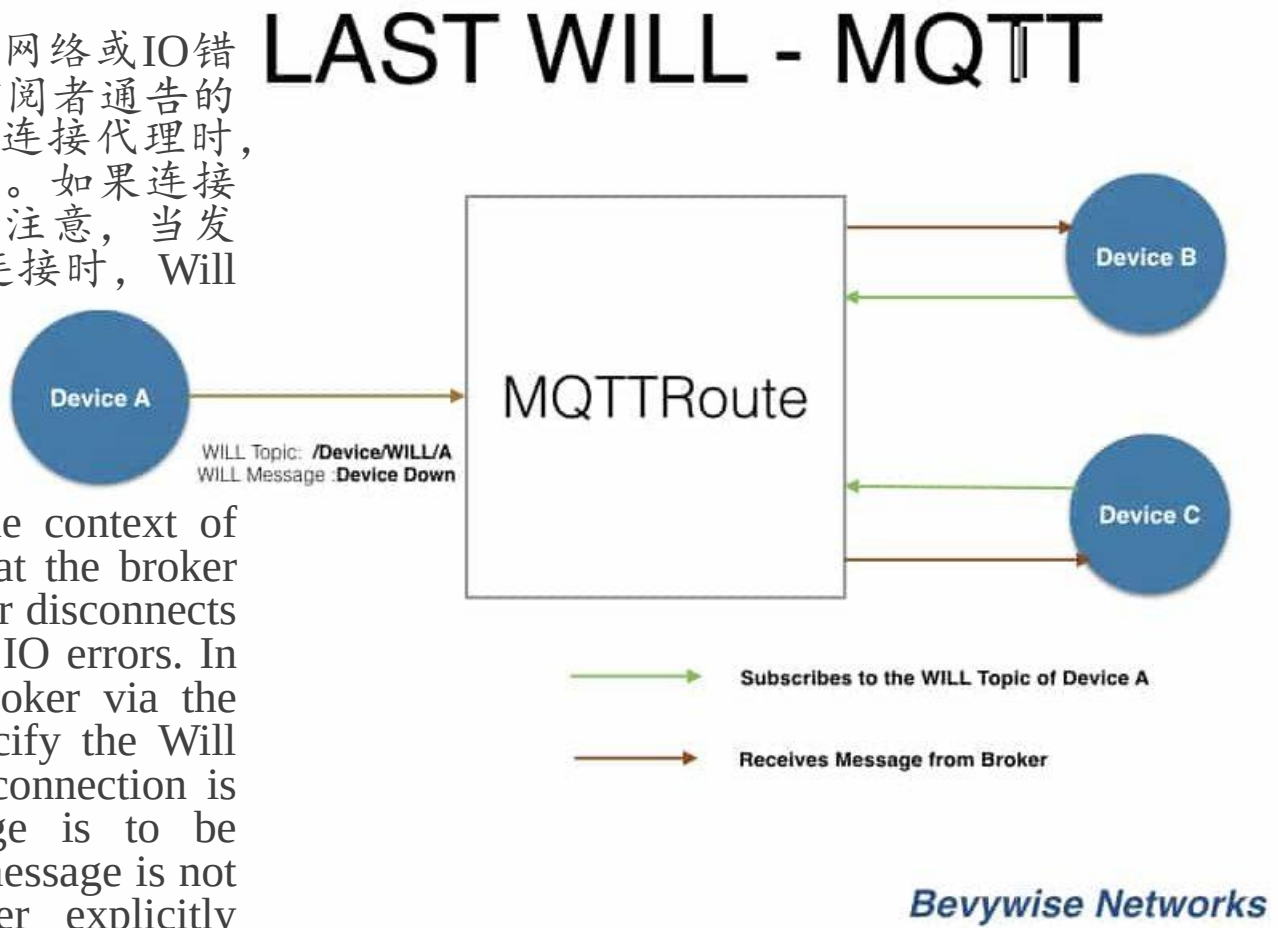
- 通常，订阅者只能接收在订阅之后发布的消息，但如果发布者事先发布了带有Retain标识的消息，则订阅者就能在订阅后马上收到该消息。具体地，当发布者发布了带有Retain标识的消息时，代理会把消息传递给订阅了该主题的订阅者，同时保存带有Retain标识的最新的消息。此时，若别的订阅者订阅了主题，就能马上收到带有Retain标识的新消息。



Usually, a subscriber can only receive messages published after subscription, but if the publisher publishes a message with the Retain flag in advance, the subscriber can receive the message immediately after subscribing. In detail, when the publisher publishes a message with the Retain flag, the broker will deliver the message to the subscribers who subscribe to the topic, and substitute it by the latest message with the Retain flag. Now, if other subscribers subscribe to the topic, they can immediately receive this new messages with the Retain flag.

MQTT Will标识
MQTT Will Flag

- Will有遗言或遗嘱的意思，用于定义当由于网络或IO错误，发布者非正常从代理断开时，代理向订阅者通告的消息。具体地，发布者通过CONNECT消息连接代理时，如有需要则会指定Will标识与要发布的消息。如果连接意外断开，则Will消息会被传递给订阅者。注意，当发布者使用DISCONNECT消息明确表示断开连接时，Will消息不会传递给订阅者。



Will means the last words or testament in the context of MQTT, and is used to specify the message that the broker uses to notify the subscribers when the publisher disconnects from the broker abnormally due to network or IO errors. In detail, when the publisher connects to the broker via the CONNECT message, if necessary, it will specify the Will flag and the message to be published. If the connection is disconnected unexpectedly, the Will message is to be delivered to the subscriber. Note that the Will message is not delivered to subscribers when the publisher explicitly disconnects using the DISCONNECT message.

MQTT Clean Session标识
MQTT Clean Session Flag

- Clean Session用于指定代理是否保留了订阅者的已订阅状态。用CONNECT消息连接时，订阅者可以选择把Clean Session置为0（保留）或1（不保留）。若指定Clean Session为0且代理已经和订阅者建立连接，则代理需要在订阅者断开连接后保留订阅的消息。若指定Clean Session为1并连接，代理会废弃以往保留的客户端信息，将其当成一次“干净”的连接来对待。此外，订阅者断开连接时，代理也会废弃所有的信息。

  Clean Session is used to specify whether should the broker retain the subscriber's subscribed state or not. When to connect to the broker with a CONNECT message, the subscriber can choose to set the Clean Session to 0 (reserved) or 1 (not reserved). If the specified Clean Session is 0 and the proxy has established a connection with the subscriber, the broker needs to retain the subscribed messages after the subscriber disconnects. If you specify Clean Session as 1 and connect, the proxy will discard the client message retained in the past and treat it as a "clean" start. Additionally, the broker discards all messages when the subscriber disconnects in this situation.