

# Internet das Coisas (IoT) Protocolos

**Prof. Dr. Peterson A. Belan**  
**belan@uni9.pro.br**

# Internet das Coisas (IoT)

## Protocolos

### **7.3 A World Wide Web**

A World Wide Web é uma estrutura arquitetônica que permite o acesso a documentos vinculados espalhados por milhões de máquinas na Internet. Em dez anos, ela deixou de ser um meio de distribuição de dados sobre física de alta energia para se tornar a aplicação que milhões de pessoas consideram ser "A Internet". Sua enorme popularidade se deve à sua interface gráfica colorida, de fácil utilização para principiantes. Além disso, ela oferece uma imensa variedade de informações sobre quase todos os assuntos imagináveis, desde aborígenes até zoologia.

A Web (também conhecida como **WWW**) teve início em 1989 no CERN, o centro europeu para pesquisa nuclear. O CERN tem vários aceleradores de partículas, nos quais grandes grupos de cientistas dos países europeus participantes desenvolvem pesquisas na área da física de partículas. Esses grupos são quase sempre compostos por membros de mais de meia dúzia de países diferentes. A maioria das experiências é altamente complexa e exige anos de planejamento para a construção dos equipamentos necessários. A Web nasceu da necessidade de fazer com que esses grupos de cientistas de diferentes nacionalidades pudessem colaborar uns com os outros através da troca de relatórios, plantas, desenhos, fotos e outros documentos.

# Internet das Coisas (IoT)

## Protocolos

A proposta inicial para uma teia de documentos vinculados veio de um físico do CERN, Tim Berners-Lee, em março de 1989. O primeiro protótipo (no modo texto) já era operacional um ano e meio depois. Em dezembro de 1991, foi realizada uma demonstração pública na conferência Hypertext '91, em San Antonio, no Texas.

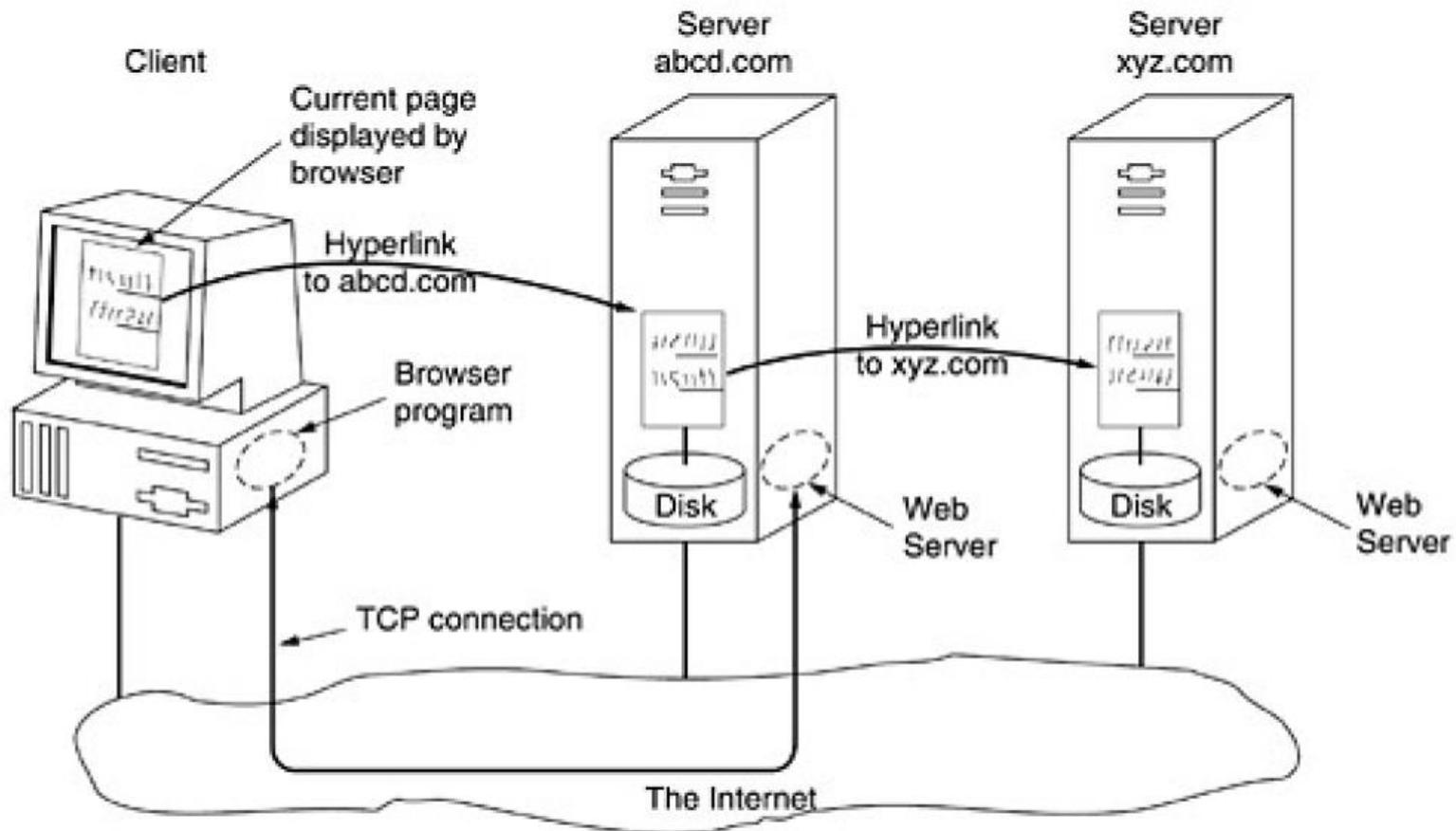
Essa demonstração e a publicidade resultante atraíram a atenção de outros pesquisadores, o que levou Marc Andreessen, da University of Illinois, a iniciar o desenvolvimento do primeiro navegador gráfico, o Mosaic, lançado em fevereiro de 1993. O Mosaic se tornou tão popular que, um ano mais tarde, Andreessen fundou sua própria empresa, a Netscape Communications Corp., cujo objetivo era desenvolver clientes, servidores e outros produtos de software para a Web. Quando a Netscape abriu seu capital ao público em 1995, os investidores, aparentemente pensando que ela seria a sucessora da Microsoft, pagaram 1,5 bilhão de dólares por suas ações. Esse recorde foi ainda mais surpreendente porque a empresa tinha apenas um produto, operava no vermelho e já havia anunciado que não previa a possibilidade de lucros para um futuro próximo. Nos três anos seguintes, o Netscape Navigator e o Internet Explorer da Microsoft se engajaram em uma "guerra de navegadores", cada um procurando incluir mais recursos (e, portanto, mais bugs) que o outro. Em 1998, o America Online comprou a Netscape Communications Corp. por 4,2 bilhões de dólares, encerrando assim a breve vida da Netscape como empresa independente.

# Internet das Coisas (IoT)

## Protocolos

Em 1994, o CERN e o MIT assinaram um acordo criando o World Wide Web Consortium (às vezes abreviado como W3C), uma organização voltada para o desenvolvimento da Web, a padronização de protocolos e para o incentivo à interoperabilidade entre os sites. Berners-Lee tornou-se o diretor do consórcio. Desde então, centenas de universidades e empresas juntaram-se ao consórcio. Embora existam agora mais livros sobre a Web do que se pode imaginar, o melhor lugar para obter informações atualizadas sobre a Web é (naturalmente) a própria Web. A home page do consórcio está no endereço [www.w3.org](http://www.w3.org). Os leitores interessados são levados por links a páginas que englobam todos os numerosos documentos e atividades de consórcio.

# Internet das Coisas (IoT) Protocolos



# Internet das Coisas (IoT)

## Protocolos

Atualmente existem diversos protocolos presentes na camada de aplicação, e em sua maioria, eles concentram-se em questões acerca da comunicação Máquina a Máquina (M2M). Al-Fuqaha et al. (2015) relata os cinco protocolos de comunicação (Data Distribution System (DDS), Extensible Messaging and Presence Protocol (XMPP), Advanced Message Queuing Protocol (AMQP), Constraint Application Protocol (CoAP) e Message Queue Telemetry Transport (MQTT)) mais indicados para a transmissão de mensagens entre dispositivos IoT. Outros autores como Yassein, Shatnawi et al. (2016), Mun, Dinh e Kwon (2016), Yokotani e Sasaki (2016), incluem em suas pesquisas WebSocket e HTTP, contudo esses protocolos não são específicos para comunicação de aplicações IoT.

# Internet das Coisas (IoT)

## Protocolos

Dos protocolos citados, esta dissertação seleciona três deles voltados essencialmente para ambientes de Internet das Coisas, que são: AMQP, CoAP e MQTT. Esses protocolos foram selecionados de acordo com alguns fatores, entre eles, o escopo do protocolo, a disponibilidade de código aberto e bibliotecas e o suporte da comunidade e documentação. Em virtude disso, o protocolo XMPP não foi utilizado devido ao fato de que este protocolo é voltado principalmente para chats, e o protocolo DDS dispõe de suporte à comunidade e disponibilidade de código limitada.

# Internet das Coisas (IoT)

## *Advanced Message Queuing Protocol*

Advanced Message Queuing Protocol (AMQP) é um protocolo de aplicação para troca de mensagens. De acordo com Frigieri, Mazzer e Parreira (2015), o AMQP é um protocolo de padrão aberto de mensagens de middleware baseado no paradigma de filas de mensagens orientadas a tópicos, em que produtos escritos para diferentes plataformas e em diferentes linguagens podem trocar mensagens. Conforme sua documentação, Standard (2012), o AMQP é um protocolo de internet aberto para mensagens de negócios. No padrão AMQP, os produtos de middleware escritos para diferentes plataformas e em diferentes idiomas podem enviar mensagens de um para outro (FERNANDES et al., 2013).

# Internet das Coisas (IoT)

## *Advanced Message Queuing Protocol*

Esse protocolo foi criado em 2003 por John O'Hara no JPMorgan Chase (NAIK, 2017), e atualmente sua documentação padronizada encontra-se em Organization for the Advancement of Structured Information Standards (OASIS). O principal objetivo do AMQP é entregar as mensagens sem perda e fornecer segurança e interoperabilidade (PONNUSAMY; RAJAGOPALAN, 2018). Ele define um protocolo de nível de fio binário que permite a troca confiável de mensagens de negócios entre duas partes, possui uma arquitetura em camadas e a especificação é organizada como um conjunto de partes que reflete essa arquitetura (STANDARD, 2012).

# Internet das Coisas (IoT)

## *Advanced Message Queuing Protocol*

O AMQP suporta a comunicação Publisher/Subscriber (Publicação/Assinatura), possui alguns níveis de serviço de entrega como até uma vez, ao menos uma vez e exatamente uma vez, semelhante ao protocolo MQTT. As mensagens trocadas entre os editores e assinantes são por meio do Transmission Control Protocol (TCP), e é fornecida uma conexão ponto-a-ponto confiável (PONNUSAMY; RAJAGOPALAN, 2018).

# Internet das Coisas (IoT)

## Advanced Message Queuing Protocol

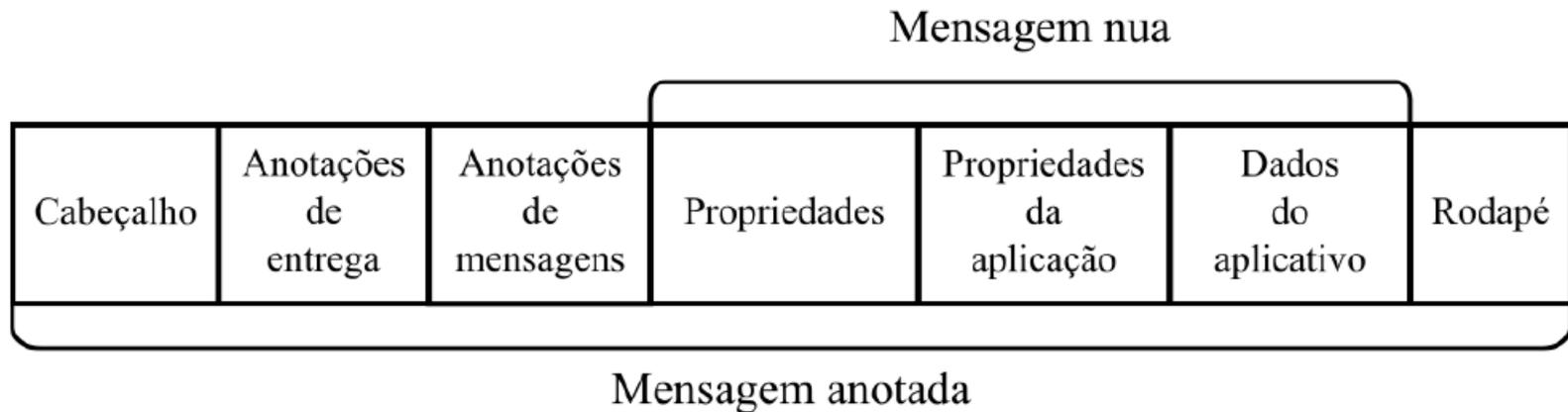


Figura 3 – Formato da mensagem AMQP

Fonte: Adaptado de Standard (2012)

# Internet das Coisas (IoT)

## Advanced Message Queuing Protocol

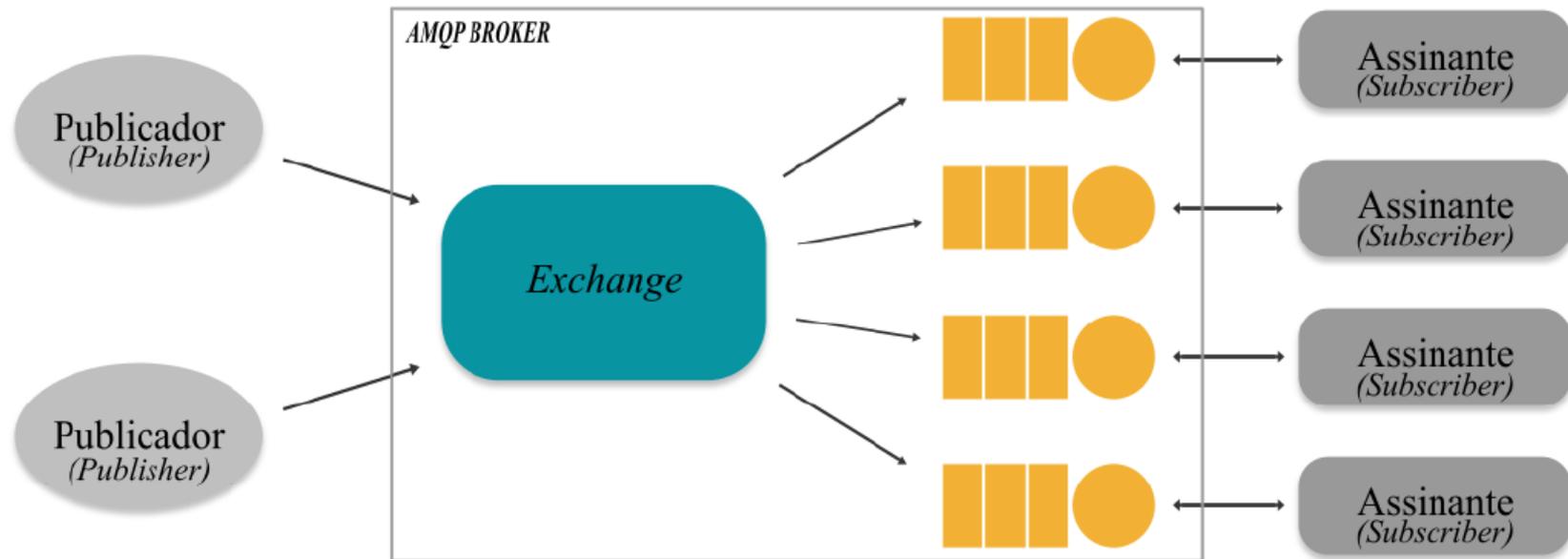


Figura 4 – Arquitetura AMQP

Fonte: Adaptado de Standard (2012)

# Internet das Coisas (IoT)

## *Advanced Message Queuing Protocol - QoS*

A Qualidade de Serviço no AMQP, ocorre durante a transmissão das mensagens por meio de uma tag de entrega, a qual também é utilizada para realizar o rastreamento do estado de entrega quando a mensagem está sendo transmitida. Segundo Standard (2012), existem três tipos de garantia de entrega:

- ***At-least-once*** (No máximo uma vez) – Ocorre quando o aplicativo de envio atribui a tag de entrega como entregue antes da transferência da mensagem se iniciar. Ou seja, o remetente indica antes de transmitir a mensagem que deletou as informações sobre ela e, portanto, não será possível realizar a retransmissão destes dados. Se o destinatário receber, ele não precisará responder com a atualização do estado de entrega.

# Internet das Coisas (IoT)

## *Advanced Message Queuing Protocol - QoS*

- ***At-most-once* (Pelo menos uma vez)** – Ocorre quando o aplicativo de envio não atribui a tag estado de entrega como delivered e o aplicativo de recebimento opte por liquidar imediatamente após o processamento da mensagem em vez de aguardar o remetente liquidar primeiro. Destarte, se a mensagem for perdida, o remetente irá perceber que a tag de entrega não foi modificada e concluirá que a entrega foi perdida, executando assim, a retransmissão da mensagem. O destinatário por sua vez, poderá receber essa mensagem de forma duplicada.
- ***Exactly-once* (Exatamente uma vez)** – Ocorre quando o aplicativo de envio é liquidado (settled=True) no mesmo momento em que o destinatário atinge um estado de terminal, e o aplicativo de recebimento é liquidado (settled=True) quando o remetente se estabelece.

# Internet das Coisas (IoT)

## Advanced Message Queuing Protocol - QoS

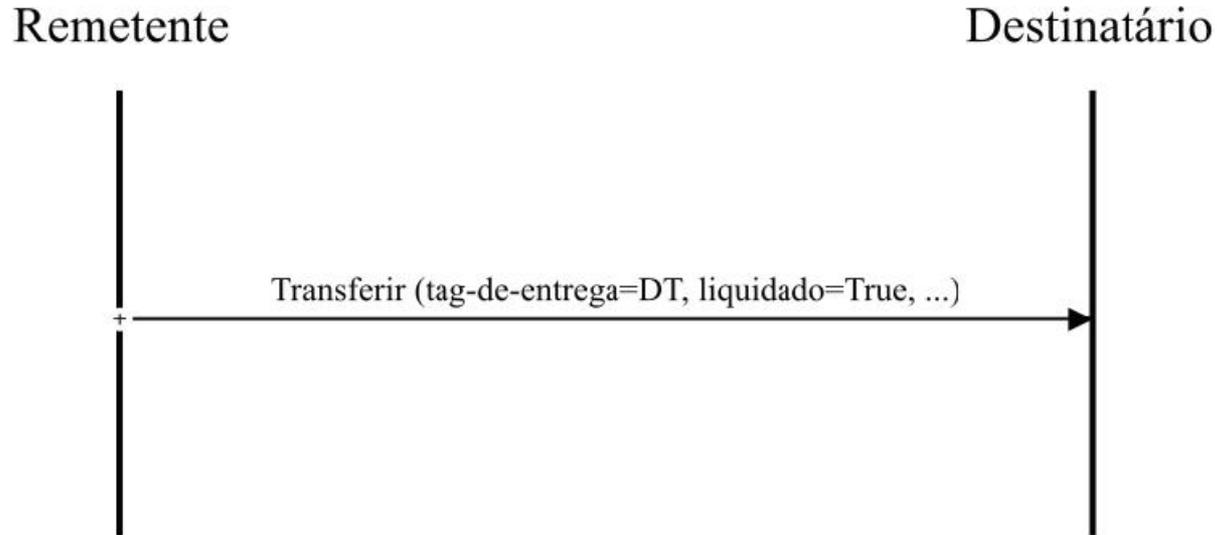


Figura 5 – Garantia de entrega no máximo uma vez AMQP

Fonte: Adaptado de Standard (2012)

# Internet das Coisas (IoT)

## Advanced Message Queuing Protocol - QoS

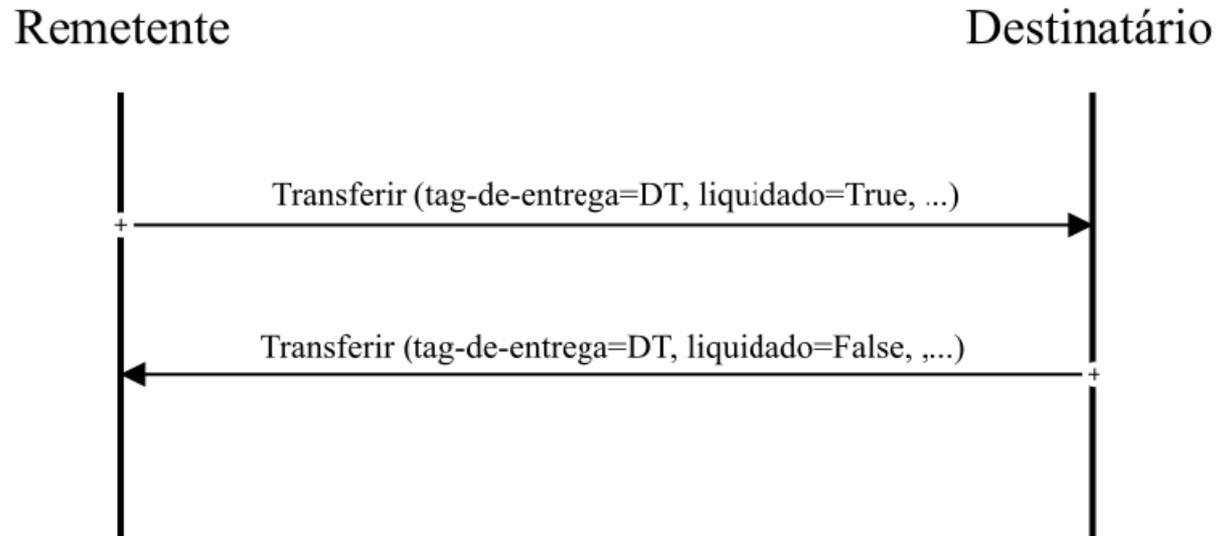


Figura 6 – Garantia de entrega pelo menos uma vez AMQP

Fonte: Adaptado de Standard (2012)

# Internet das Coisas (IoT)

## *Constrained Application Protocol*

Constraint Application Protocol (CoAP) é um protocolo que foi projetado para ser usado na comunicação de dispositivos com recursos limitados (THANGAVEL et al., 2014). Esse protocolo foi criado por um grupo da Internet Engineering Task Force (IETF) composto por Carsten Bormann, Andrew McGregor e Barry Leiba chamado Constrained RESTful Environments (CoRE) em 2010, e em 2014 sua documentação foi lançada na Request for Comments (RFC) 7252 (MARTINS; ZEM, 2016).

# Internet das Coisas (IoT)

## *Constrained Application Protocol*

Um dos objetivos do projeto CoAP era o de minimizar a sobrecarga de mensagens e limitar a fragmentação de pacotes (CARO et al., 2013). O CoAP foi desenvolvido principalmente para interoperar com o Hypertext Transfer Protocol (HTTP) e Web RESTful por meio de proxies simples (NAIK, 2017).

De acordo com Shelby, Hartke e Bormann (2014), um dos principais objetivos do CoAP é projetar um protocolo web genérico para os requisitos especiais de ambientes restritos, especialmente, considerando energia, automação predial e outras aplicações M2M. Também, segundo Shelby, Hartke e Bormann (2014), o objetivo do CoAP não é simplesmente compactar o HTTP, mas sim, para realizar um subconjunto REST comum com o HTTP mais otimizado para aplicativos M2M.

# Internet das Coisas (IoT)

## Constrained Application Protocol

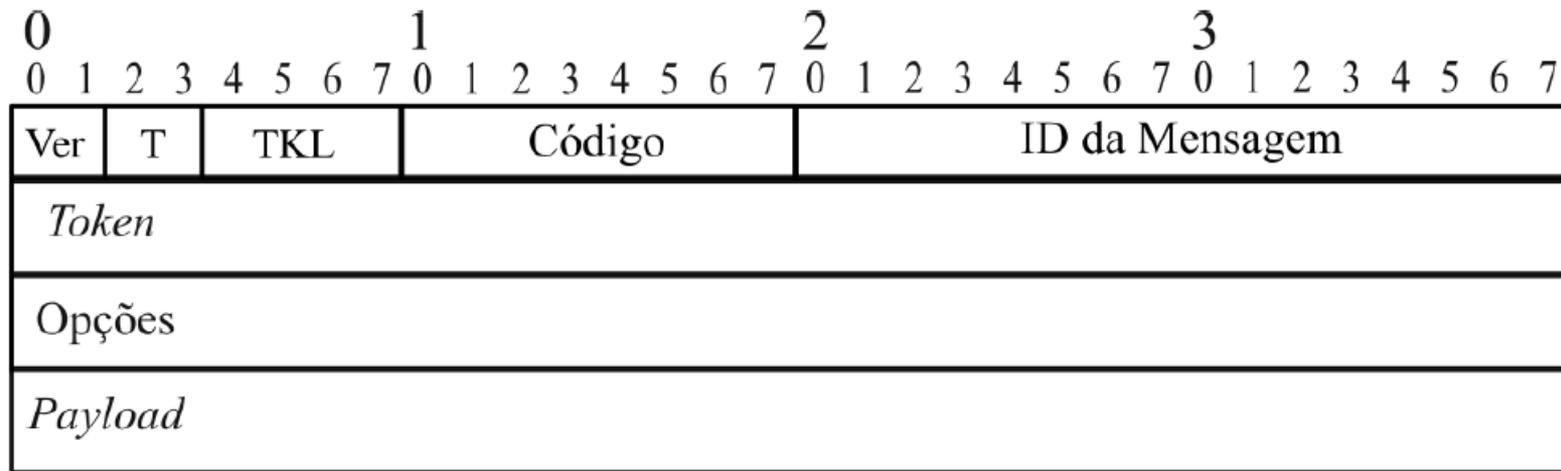


Figura 7 – Formato da mensagem CoAP

Fonte: Adaptado de Shelby, Hartke e Bormann (2014)

# Internet das Coisas (IoT)

## *Constrained Application Protocol - Métodos*

No CoAP, é proposto uma semântica semelhante ao HTTP, utilizando para si os métodos GET, POST, PUT e DELETE. Assim, esses métodos possuem propriedades semelhantes às propriedades dos métodos análogos ao HTTP, como por exemplo seguro, ou seja, somente para recuperação; e idempotente, isto é, possui o mesmo efeito ao ser chamado diversas vezes (JOSHI; KAUR, 2015; SHELBY; HARTKE; BORMANN, 2014). Contudo, ao contrário do HTTP que funciona por meio do TCP, o CoAP funciona por meio do User Datagram Protocol (UDP).

# Internet das Coisas (IoT)

## *Constrained Application Protocol - Métodos*

Esses métodos citados são modificados para atender aos requisitos de Internet das Coisas, como baixo consumo de energia e operação na presença de links com e sem ruído (AL-FUQAHA et al., 2015). A seguir, temos a definição dos métodos que esse protocolo utiliza para a comunicação entre cliente e servidor de acordo com Shelby, Hartke e Bormann (2014):

# Internet das Coisas (IoT)

## *Constrained Application Protocol - Métodos*

- **GET** – Recupera uma representação para as informações que correspondem ao recurso identificado pelo *Uniform Resource Identifier* (URI) da solicitação realizada pelo cliente. Esse método é seguro e idempotente.
- **POST** – Solicita que a representação incluída no pedido seja processada resultando geralmente em uma criação ou atualização de um recurso. Esse método não é seguro e não é idempotente.
- **PUT** – Solicita que um recurso identificado pela solicitação URI seja criado ou atualizado com uma representação em anexo. Esse método não é seguro, mas é idempotente.
- **DELETE** – Solicita que o recurso identificado pelo URI seja excluído. Esse método não é seguro, mas é idempotente.

# Internet das Coisas (IoT)

## Constrained Application Protocol - Métodos

Código de Resposta	Definição
2.xx	Sucesso
2.01	<i>Created</i>
2.02	<i>Deleted</i>
2.03	<i>Valid</i>
2.04	<i>Changed</i>
2.05	<i>Content</i>

# Internet das Coisas (IoT)

## Constrained Application Protocol - Métodos

Código de Resposta	Definição
4.xx	Erro no Cliente
4.00	<i>Bad Request</i>
4.01	<i>Unauthorized</i>
4.02	<i>Bad Option</i>
4.03	<i>Forbidden</i>
4.04	<i>Not found</i>
4.05	<i>Method not allowed</i>
4.06	<i>Not Acceptable</i>
4.12	<i>Precondition Failed</i>
4.13	<i>Request Entity Too Large</i>
4.15	<i>Unsupported Content-Format</i>

# Internet das Coisas (IoT)

## Constrained Application Protocol - Métodos

Código de Resposta	Definição
5.xx	Erro no Servidor
5.00	<i>Internal Server Error</i>
5.01	<i>Not Implemented</i>
5.02	<i>Bad Gateway</i>
5.03	<i>Service Unavailable</i>
5.04	<i>Gateway Timeout</i>
5.05	<i>Proxying Not Supported</i>

Fonte: Adaptado de Shelby, Hartke e Bormann (2014)

# Internet das Coisas (IoT)

## Constrained Application Protocol - Métodos

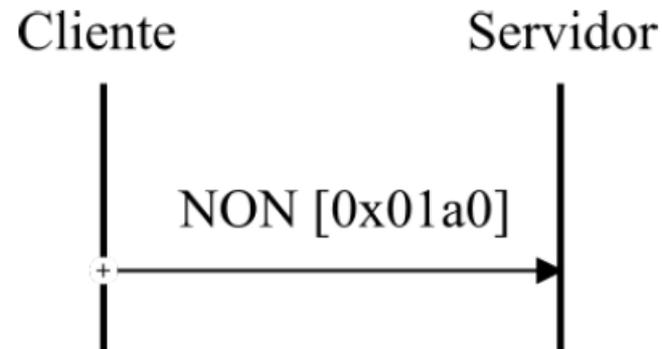


Figura 8 – Troca de Mensagens *Non-Conformable*

Fonte: Adaptado de Shelby, Hartke e Bormann (2014)

# Internet das Coisas (IoT)

## Constrained Application Protocol - Métodos

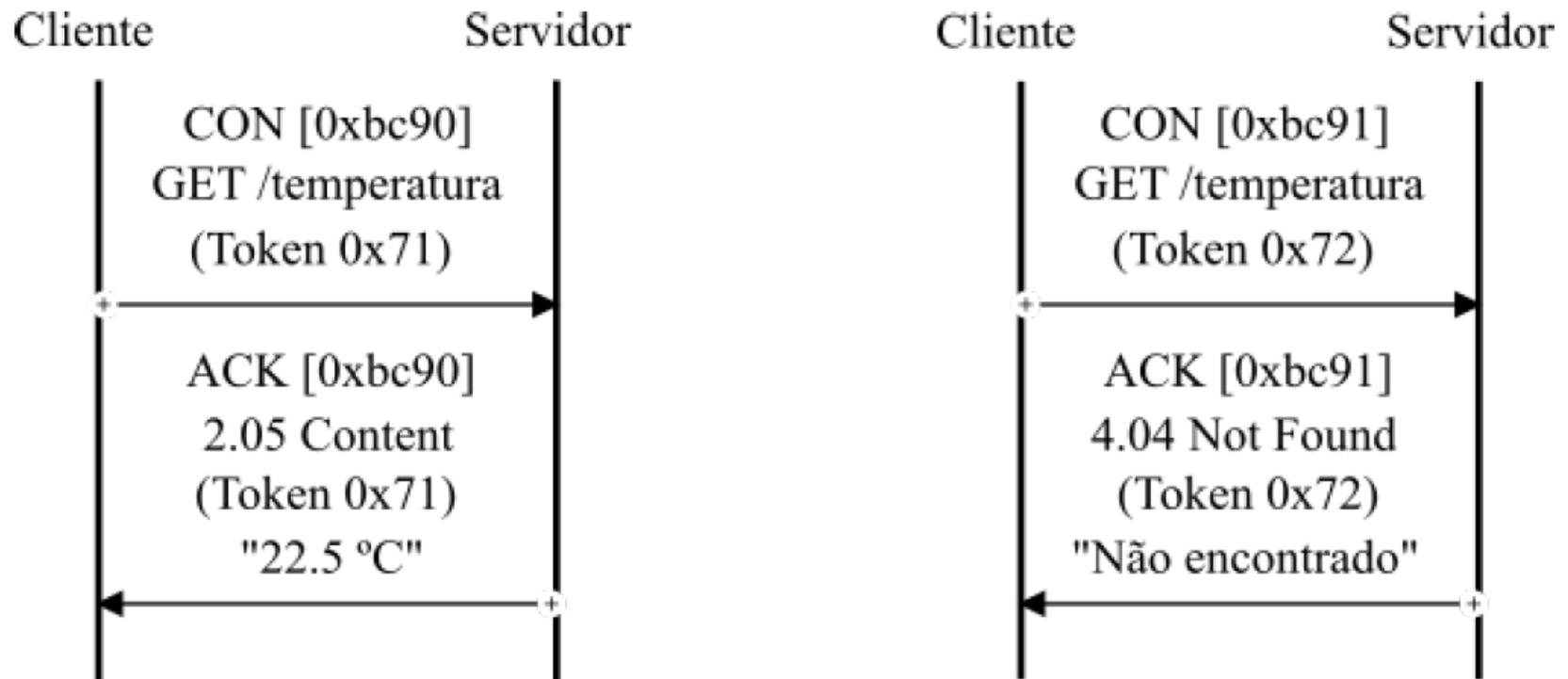


Figura 9 – Troca de mensagem *Conformable Piggybacked*

Fonte: Adaptado de Shelby, Hartke e Bormann (2014)

# Internet das Coisas (IoT)

## Constrained Application Protocol - Métodos

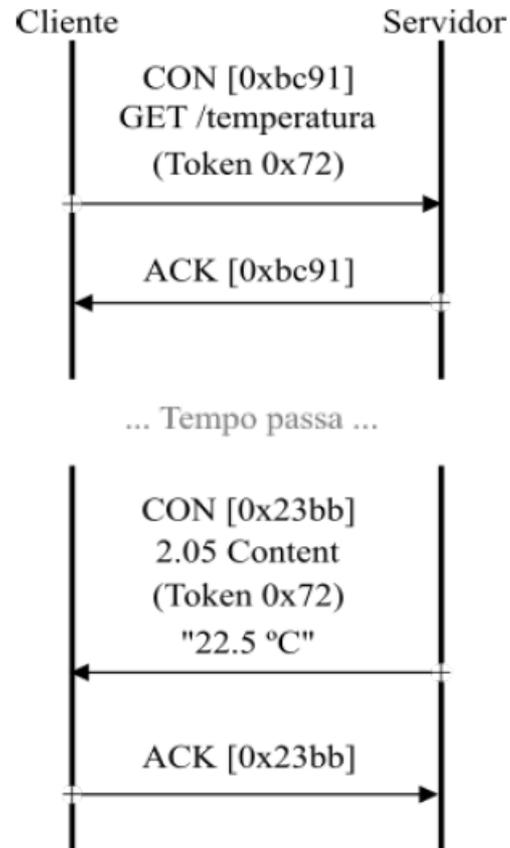


Figura 10 – Troca de mensagem *Confirmable Separate Response*

Fonte: Adaptado de Shelby, Hartke e Bormann (2014)

# Internet das Coisas (IoT)

## *Message Queuing Telemetry Transport*

Message Queue Telemetry Transport (MQTT) é um protocolo de aplicação que foi projetado para atuar em dispositivos que possuem poucos recursos computacionais. Ele foi criado, no final da década de 1990, pela International Business Machines (IBM), e em 2013, foi documentado como protocolo na Organization for the Advancement of Structured Information Standards (OASIS) (YASSEIN et al., 2017).

# Internet das Coisas (IoT)

## *Message Queuing Telemetry Transport*

Inicialmente, sua aplicação original era vincular sensores em pipelines de petróleo a satélites, e como seu nome sugere, ele é um protocolo de mensagem com suporte para a comunicação assíncrona entre as partes (YUAN, 2017). A operação de conexão usa um mecanismo de roteamento (um-para-um, um-para-muitos, muitos-para-muitos) e utiliza padrões de roteamento Máquina a Máquina (M2M), Máquina a Servidor (M2S) e Servidor a Servidor (S2S) (AL-FUQAHA et al., 2015; YASSEIN et al., 2017).

# Internet das Coisas (IoT)

## *Message Queuing Telemetry Transport*

O MQTT possui características intrínsecas, que o tornam uma opção valiosa em ambientes de Internet das Coisas que comumente dispõem de baixa largura de banda, são intermitentes, entre outros fatores (MANSO et al., 2018; BELLAVISTA; ZANNI, 2016). Este protocolo, assim como o protocolo AMQP, trabalha sobre o Transmission Control Protocol (TCP).

# Internet das Coisas (IoT)

## Message Queuing Telemetry Transport

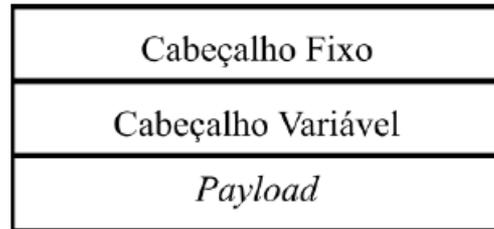


Figura 11 – Formato geral da mensagem MQTT

Fonte: Adaptado de Banks e Gupta (2015)

Bit	7	6	5	4	3	2	1	0
Byte 1	Tipo da Mensagem				DUP	QoS		Retain
Byte 2	Largura Restante							

Figura 12 – Cabeçalho fixo MQTT

Fonte: Adaptado de Banks e Gupta (2015)

# Internet das Coisas (IoT)

## Message Queuing Telemetry Transport

Nome	Valor	Descrição
<i>Reserved</i>	0	Reservado
<i>CONNECT</i>	1	Solicitação do cliente para se conectar ao servidor
<i>CONNACK</i>	2	ACK da conexão
<i>PUBLISH</i>	3	Mensagem Publicação
<i>PUBACK</i>	4	ACK da Publicação
<i>PUBREC</i>	5	Publicação recebida (garantia de entrega parte I)
<i>PUBREL</i>	6	Publicação liberada (garantia de entrega parte II)
<i>PUBCOMP</i>	7	Publicação completa (garantia de entrega parte III)
<i>SUBSCRIBE</i>	8	Solicitação de assinatura do cliente
<i>SUBACK</i>	9	ACK de assinatura
<i>UNSUBSCRIBE</i>	10	Solicitação de cancelamento de assinatura do cliente
<i>UNSUBACK</i>	11	ACK de cancelamento de assinatura
<i>PINGREQ</i>	12	Solicitação PING
<i>PINGRESP</i>	13	Resposta PING
<i>DISCONNECT</i>	14	Cliente desconectado
<i>Reserved</i>	15	Reservado

Fonte: Adaptado de Banks e Gupta (2015)

# Internet das Coisas (IoT)

## *Message Queuing Telemetry Transport*

A comunicação do protocolo MQTT, ocorre por meio da arquitetura Publicação/Assinatura, ilustrada na Figura 13. Nela é possível observar três componentes principais: Publisher (Publicador), o Broker (Intermediário) e o Subscriber (Assinante). Segundo Xu, Mahendran e Radhakrishnan (2016), cada componente é definido por:

# Internet das Coisas (IoT)

## *Message Queuing Telemetry Transport*

- **Publisher** – Os Publicadores, como o próprio nome indica, são os dispositivos que geram dados. Os dados são publicados seguindo o formato de tópicos.
- **Broker** – O Intermediário age como um nó central realizando a comunicação entre Publicadores e Assinantes.
- **Subscriber** – Os Assinantes recebem as mensagens publicadas, de acordo com o tópico em que elas se inscreveram.

# Internet das Coisas (IoT)

## *Message Queuing Telemetry Transport*

Para publicar os dados por meio do MQTT, um Publisher, que pode ser um sensor, por exemplo, inicialmente realiza uma solicitação para conexão com o Broker. Após realizada a conexão alguns elementos são estabelecidos: QoS, tópico e mensagem.

# Internet das Coisas (IoT)

## Message Queuing Telemetry Transport

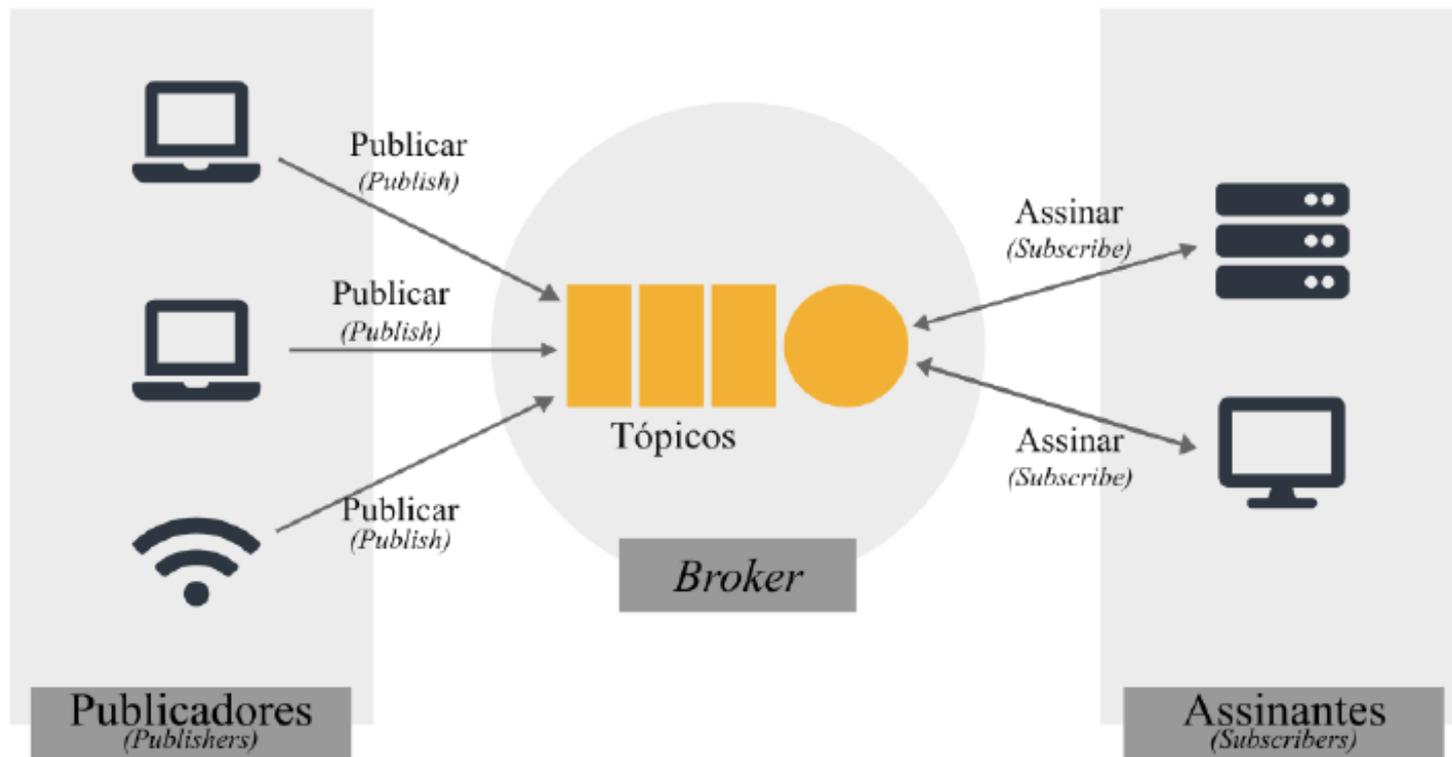


Figura 13 – Arquitetura MQTT

Fonte: Adaptado de Al-Fuqaha et al. (2015)

# Internet das Coisas (IoT)

## *Message Queuing Telemetry Transport*

O nível da Qualidade de Serviço (QoS) irá definir se as mensagens irão ser entregues ao menos uma vez, pelo menos uma vez ou exatamente uma vez, como relatado posteriormente nesta seção. O tópico é definido como uma cadeia que é usada pelo intermediário para decidir qual assinante receberá a mensagem, sendo utilizado pelo Broker para decidir qual assinante encaminhará a mensagem (TANTITHARANUKUL et al., 2016). Um exemplo que pode ser citado é a utilização de um sensor que publica dados por meio do MQTT fazendo uso do seguinte tópico:

*Tópico: sensor*

# Internet das Coisas (IoT)

## *Message Queuing Telemetry Transport*

Ademais, um tópico pode possuir um ou mais níveis hierárquicos, onde cada nível é separado por uma barra (/) (TANTITHARANUKUL et al., 2016). Utilizando o mesmo exemplo citado anteriormente, este sensor pode publicar mensagens empregando o seguinte tópico:

*Tópico: sensor/temperatura*

# Internet das Coisas (IoT)

## *Message Queuing Telemetry Transport*

Neste caso, somente os *Subscribers* que estão inscritos nesse tópico é que receberão as mensagens publicadas pelo sensor. A mensagem consiste basicamente na informação que será publicada pelo *Publisher*. O *Subscriber* inicialmente também solicita uma conexão com o *Broker*.

Após a conexão ser realizada, ele vai assinar um determinado tópico, para posteriormente receber informações sobre o tópico assinado. Por exemplo, caso deseje receber informações sobre o sensor de temperatura, sua inscrição será por meio da expressão:

*Tópico: sensor/temperatura*

# Internet das Coisas (IoT)

## *Message Queuing Telemetry Transport*

Seguindo este exemplo de sensores, também é possível receber informações das publicações de todos os sensores, por meio do seguinte tópico:

*Tópico: sensor/#*

# Internet das Coisas (IoT)

## *Message Queuing Telemetry Transport*

Além disso, o *Subscriber* pode solicitar o cancelamento da inscrição para remover uma solicitação de mensagens e caso não deseje mais receber informações poderá se desconectar do Broker (BANKS; GUPTA, 2015). Como pode ser observado na Figura 13, ao ser publicada uma mensagem, o Broker irá transmitir essa mensagem a todos os interessados que possuem assinatura no tópico, sendo possível lidar com milhares de dispositivos conectados ao mesmo tempo (YASSEIN et al., 2017).

# Internet das Coisas (IoT)

## *Message Queuing Telemetry Transport - QoS*

Como relatado anteriormente, o protocolo MQTT possui três níveis de Qualidade de Serviço. Estes são: Até uma vez, Ao menos uma vez e Exatamente uma vez. Eles serão detalhados a seguir.

- **QoS 0 - Até uma vez** – De acordo com Banks e Gupta (2015), nesse nível de QoS nenhuma mensagem de resposta é enviada pelo receptor e nenhuma tentativa é executada pelo remetente, ou seja, o remetente envia e não recebe nenhuma confirmação de que a mensagem enviada chegou ao seu destinatário. Isto significa que uma mensagem enviada pode ou não chegar aos inscrito do tópico. Segundo Manso et al. (2018), nesse nível, as mensagens são entregues de acordo com o melhor esforço da rede e a mensagem não é armazenada. A Figura 14 ilustra a troca de mensagens para QoS-0.

# Internet das Coisas (IoT)

## Message Queuing Telemetry Transport - QoS

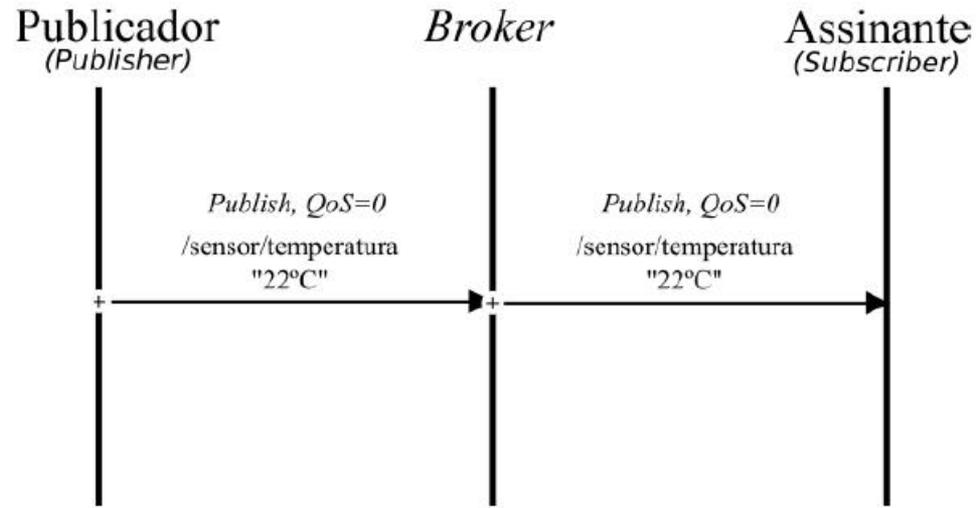


Figura 14 – QoS 0 - Até uma vez MQTT

Fonte: Adaptado de Banks e Gupta (2015)

# Internet das Coisas (IoT)

## *Message Queuing Telemetry Transport - QoS*

- **Qos 1 - Ao menos uma vez** – Neste nível, uma mensagem de resposta é enviada confirmando o recebimento da mensagem, garantindo assim, que a mensagem chegue ao destinatário pelo menos uma vez. O remetente deve tratar o pacote como unacknowledged até que tenha recebido o pacote PUBACK correspondente ao receptor (BANKS; GUPTA, 2015). Desse modo, pode ocorrer o recebimento de pacotes duplicados. De acordo com Bellavista e Zanni (2016), em QoS-1, as mensagens devem ser armazenadas localmente no remetente, até que tenham sido entregues ao seu receptor, de modo que sejam viáveis possíveis retransmissões. Após o recebimento da mensagem PUBACK, a mensagem é descartada, como visto na Figura 15.

# Internet das Coisas (IoT)

## Message Queuing Telemetry Transport - QoS

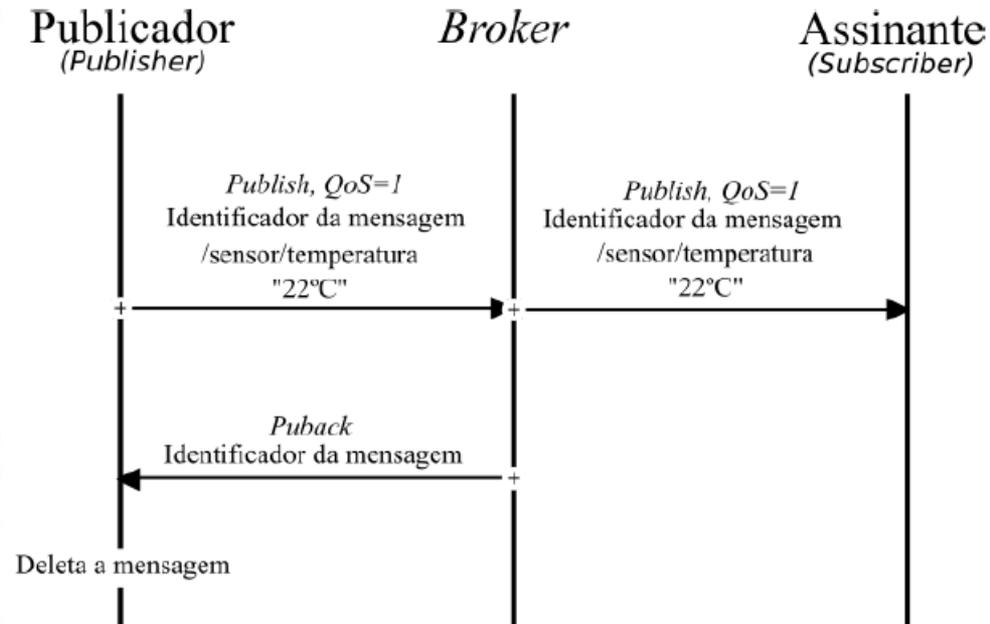


Figura 15 – QoS 1 - Ao mesmo uma vez MQTT

Fonte: Adaptado de Banks e Gupta (2015)

# Internet das Coisas (IoT)

## *Message Queuing Telemetry Transport - QoS*

- **QoS 2 - Exatamente uma vez** – Esta é considerada a mais alta Qualidade de Serviço, sendo recomendado para os casos em que a perda de pacotes e a duplicação de pacotes não sejam aceitáveis, ocorrendo assim, a entrega de exatamente uma única mensagem ao destinatário. O remetente deve tratar o pacote como unacknowledged até que tenha recebido o pacote PUBREC correspondente, e após o recebimento desse pacote, deve enviar um pacote PUBREL tratando esse pacote como unacknowledged até que receba um pacote PUBCOMP correspondente (BANKS; GUPTA, 2015). Após o recebimento desse último pacote a mensagem é deletada, como pode ser observado na Figura 16. Devido a esta série de troca de pacotes confirmando o recebimento, há uma sobrecarga associada a ela. Ademais, é possível ocorrer um atraso relativo na entrega das mensagens, contudo não há perda de pacotes (GRGIĆ; ŠPEH; HEđI, 2016).

# Internet das Coisas (IoT)

## Message Queuing Telemetry Transport - QoS

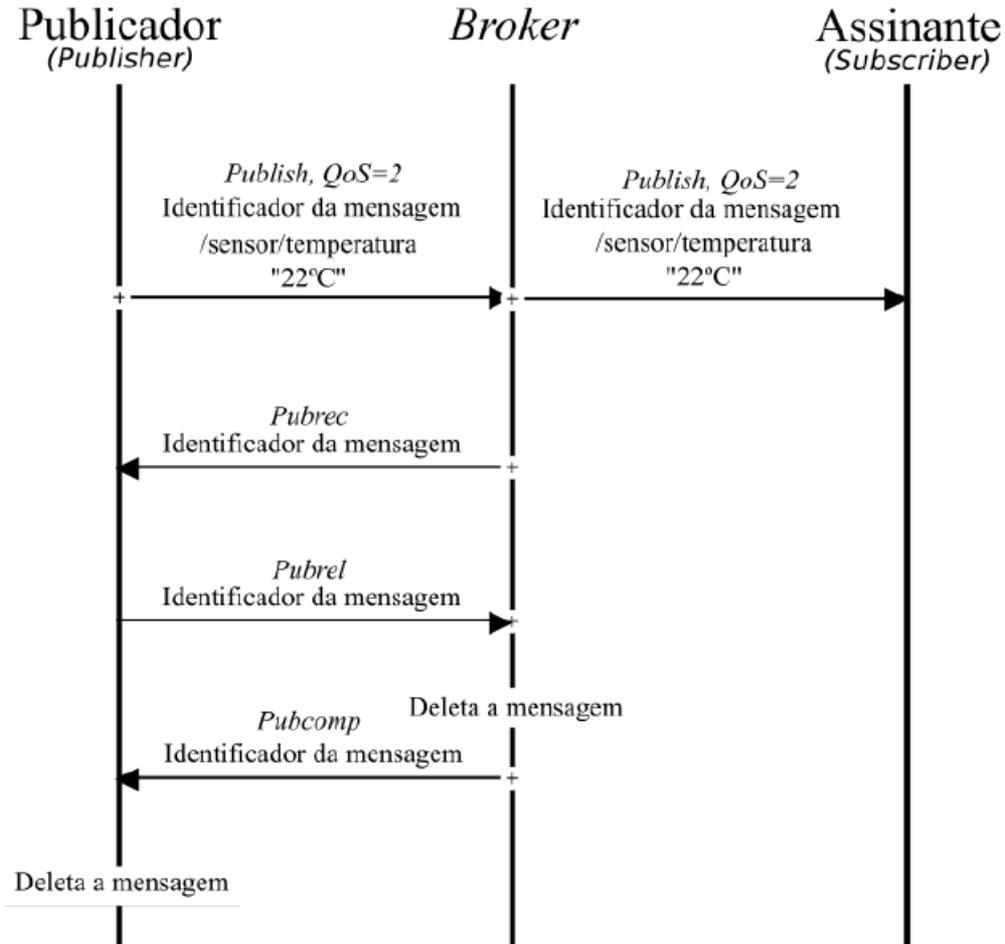


Figura 16 – QoS 2 - Exatamente uma vez MQTT

Fonte: Adaptado de Banks e Gupta (2015)