



MQTT



Prof. Me. Hélio Esperidião

MQTT(Message Queuing Telemetry Transport)



Protocolo de comunicação M2M - Machine to Machine próprio para Internet of Things (IoT) que funciona sobre o protocolo TCP/IP.



MQTT se baseia na comunicação entre cliente e servidor, em que o cliente pode fazer “postagens” ou “ouvir” publicações. O servidor administra os dados a serem recebidos e enviados.



É utilizado um Paradigma chamado Publish-Subscribe. O protocolo se popularizou pela simplicidade, baixo consumo de dados e pela possibilidade comunicação bilateral.

História



Nos anos 90 a IBM criou o protocolo MQTT.



Nasceu devido à necessidade de um protocolo leve que conseguisse comunicar várias máquinas(microcontroladores) entre si.



O MQTT é uma das principais tecnologias que podem tanto participar quanto impulsionar o desenvolvimento de dispositivos de Internet das Coisas (IoT).

Publish-Subscribe

O Padrão Publish-Subscribe permite que dispositivos se inscrevam em canais e escutem todas as publicações desses canais.

- Quem deseja comunicar outros dispositivos publica no canal (Publish)
- Quem fica esperando por novas publicações faz um Subscribe no canal

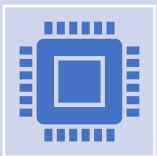
Dessa forma, o publisher e subscriber não precisam se conhecer diretamente e apenas precisam conhecer.

É preciso apenas conhecer o canal.

Broker



O Broker é o servidor de comunicação entre os dispositivos.



É ele que recebe os dados enviados pelos microcontroladores e é nele onde esses dados são tratados e passados adiante.

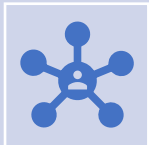
Broker



O Broker gerencia as inscrições em canais.



O Broker envia os dados para todos os inscritos.

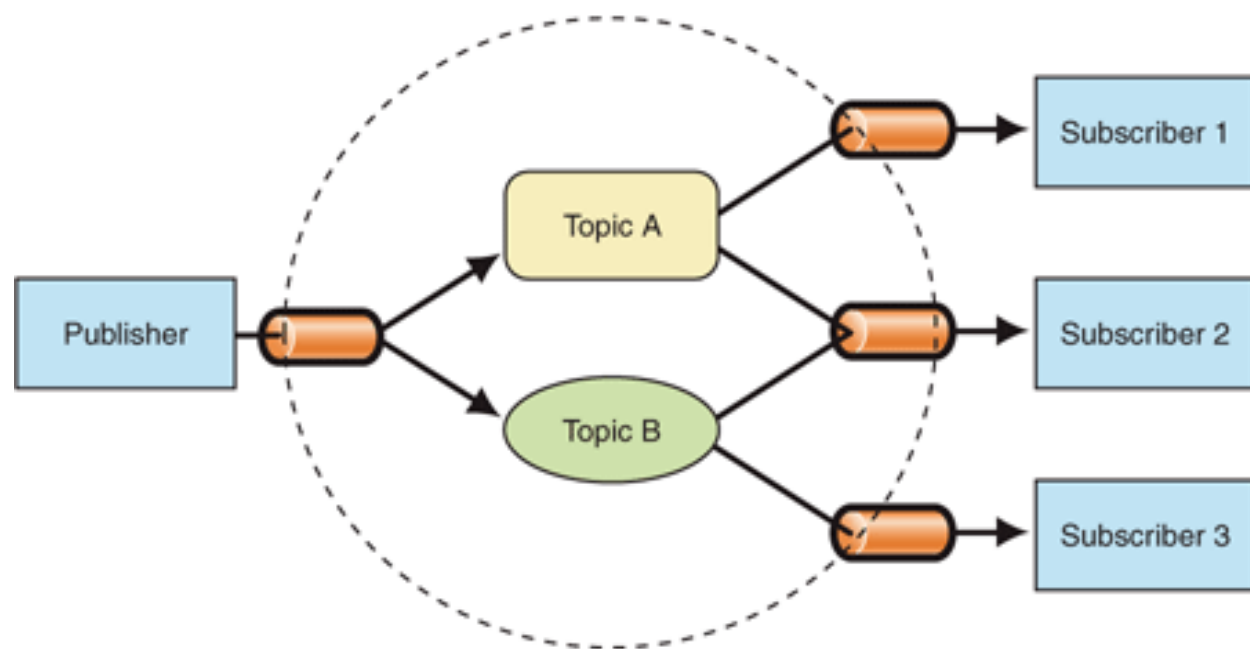


O broker gerência as conexões no servidor.

Publisher e Subscriber ao mesmo tempo

É comum em alguns casos os sistemas terem atuações tanto de Publisher quanto de Subscriber.

Ou seja, hora recebe dados de um canal hora escreve em um canal.



Communication Infrastructure

Cliente

O Cliente possui duas áreas de atuação sobre a informação:

- Postagem e Recebimento, "publish" e "subscriber".

Ele pode escolher em qual área atuar, sendo possível trabalhar na Postagem, no Recebimento ou nos dois ao mesmo tempo.

Mas independente da função sempre será necessária a presença de um Broker para realizar a intermediação dos dados entre todos os clientes.

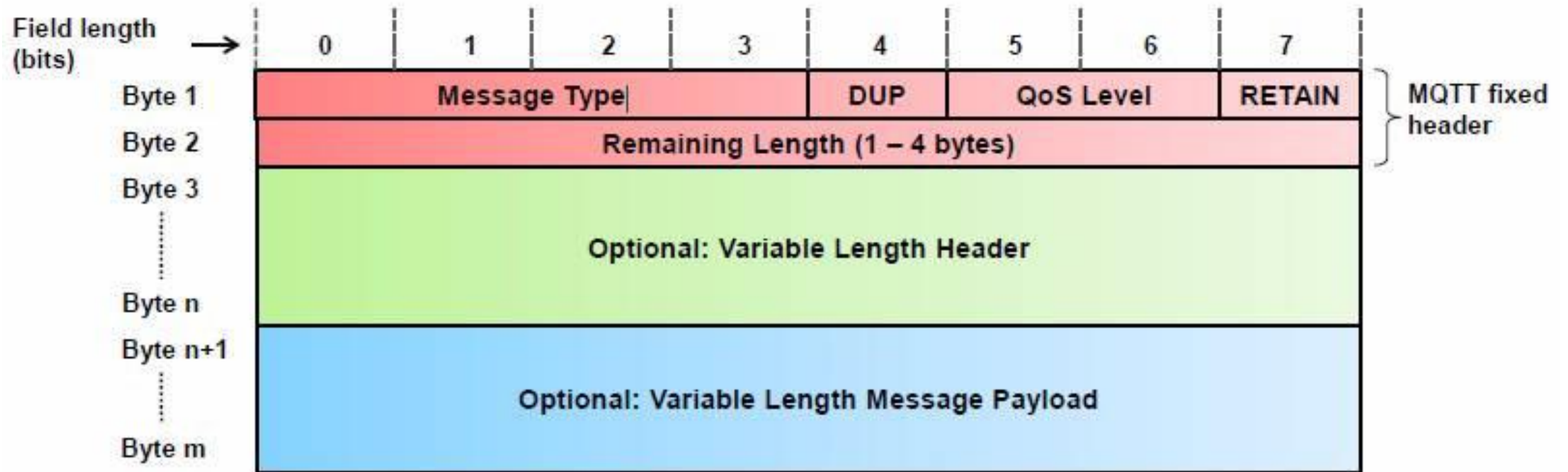
Transmissão de Dados

O protocolo MQTT utiliza outro protocolo chamado TCP para a transmissão de dados.

Broker, subscribers e publishers

- O Broker é um intermediador entre os clientes do protocolo.
 - É ele quem vai receber as informações de um cliente publisher e vai enviar essa informação aos clientes subscribers inscritos no tópico referente àquela informação.
- O Cliente Publisher é aquele que envia ao broker.
 - Um exemplo seria um sensor, que envia os dados captados (como temperatura), ao broker que, por sua vez, enviará esse dado aos clientes subscribers.
- Cliente Subscriber é o cliente que recebe uma informação do broker.
 - Ele deve se inscrever em tópicos definidos pelo broker, cada um sobre uma certa informação, que por sua vez foi captada por um cliente publisher.
- Vale dizer que o cliente pode ser, ao mesmo tempo, publisher e subscriber, ou seja, enviar e consumir dados ao/do broker.

Encapsulamento de comunicação.

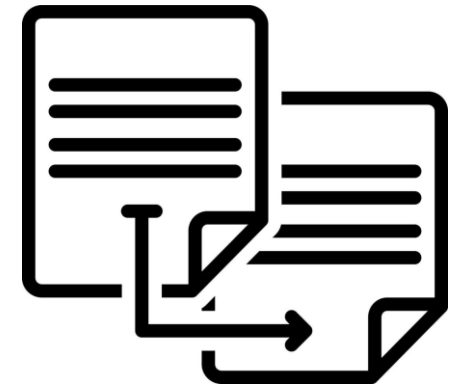


Header

- O Header do MQTT pode variar de 2 a 5 bytes.
- O primeiro byte obrigatório
 - Os 4 primeiros bits do primeiro byte referem-se ao tipo de mensagem,
 - O bit seguinte refere-se ao indicador de mensagem duplicada,
 - Dois bits para identificar o QoS(qualidade de serviço) do pacote e bit para indicar de se a mensagem deve ser retida ou não para quando alguém se conectar receber a última mensagem enviada.
- Os próximos 4 bytes irão definir o tamanho do resto do pacote, podendo ir de 0 a 268 435 455 bits.
- O restante são informações que podem variar e não existe um padrão.

Quality of Service (QoS)

- QoS é indicado por um número que varia de 0 a 2.
- Cada nível oferece um grau diferente de confiabilidade e garantia para a entrega da mensagem



MQTT DUP

- MQTT DUP indica que uma mensagem é uma duplicata e foi reenviada porque o destinatário pretendido (cliente ou corretor) não reconheceu a mensagem original.
- Isso é relevante apenas para mensagens com QoS maior que 0.
- Quando um cliente ou broker recebe uma mensagem com a bandeira DUP definida, ele deve ignorar a mensagem se já tiver recebido uma mensagem com o mesmo ID de mensagem.
- O cliente ou corretor deve processar a mensagem normalmente se não a tiver recebido anteriormente.

Retain

- É um recurso importante que determina se o corretor (broker) salva uma mensagem como o último valor conhecido válido para um tópico especificado.
- Quando a bandeira de retenção é definida como verdadeira, o broker salvará a mensagem mais recente que corresponda ao tópico especificado, independentemente de haver clientes inscritos.



Payload

What is a
Payload?



- A carga útil (payload) é o conteúdo real da mensagem e pode conter qualquer tipo de dados. O MQTT pode lidar com diferentes tipos de dados, incluindo imagens, texto em qualquer codificação, dados criptografados e dados binários.
- é importante observar que o tamanho da carga útil pode afetar o desempenho da rede e o uso de memória no cliente e no Broker.
- Portanto, é recomendável o mais pequenas possível, especialmente ao publicar mensagens com alta frequência.

Tipos de mensagens: Connect

- Tenta criar uma conexão com o Broker e espera até que a conexão seja estabelecida, começando a escutar as mensagens publicadas.
- Este tipo de mensagem é usada por um cliente para iniciar uma conexão com o broker.
- Ela inclui informações sobre a identificação do cliente, o nível de QoS desejado e outros parâmetros de conexão.



Tipos de mensagens: Disconnect

- Espera que até o cliente terminar alguma ação que esteja realizado e finaliza a conexão TCP/IP, parando assim de escutar as mensagens que serão publicadas.



Tipos de mensagens: SUBSCRIBE

- Este tipo de mensagem é enviada pelos clientes para se inscreverem em tópicos específicos.
- A mensagem SUBSCRIBE informa ao broker os tópicos de interesse do cliente e o nível de QoS desejado para cada tópico assinado.



Tipos de mensagens: UNSUBSCRIBE

- Similar à mensagem SUBSCRIBE, o tipo de mensagem UNSUBSCRIBE é enviada pelos clientes para informar ao broker os tópicos dos quais eles não desejam mais se inscrever.



Tipos de mensagens: PUBACK

- Este tipo de mensagem é usada para confirmar o envio bem-sucedido de uma mensagem PUBLISH do remetente para o destinatário. Ela confirma que a mensagem foi recebida e processada.



Tipos de mensagens: Publish

- O tipo de mensagem PUBLISH é usado tanto por clientes quanto por brokers para enviar dados de um publicador (remetente) para um ou mais assinantes (destinatários).
- Ela carrega os dados da carga útil, o tópico e o nível de QoS.



Como começar a usar o protocolo MQTT

Existem inúmeras implementações tanto para brokers como para clientes MQTT em diversas linguagens, tais como Python, JavaScript, C#, entre outros.

Uma dessas implementações mais famosas é o broker Mosquitto.

Mosquitto é um dos brokers mais utilizados atualmente devido à sua simplicidade e facilidade de implementação, esta tecnologia open source ganhou um grande espaço no mercado de IoT.

Mosquitto

- O "Mosquitto" pode ser facilmente baixado para vários sistemas operacionais.
- Os arquivos de download podem ser encontrados na página oficial do broker.
- Ao baixar o broker escolhido, este vai ser utilizado para rodar o servidor onde o sistema e o relacionamento entre os clientes e o broker será hospedado.



Outros tipos de Brokers

Broker	Server	Ports	Websocket
Mosquitto	iot.eclipse.org	1883 / 8883	n/a
HiveMQ	broker.hivemq.com	1883	8000
Mosquitto	test.mosquitto.org	1883 / 8883 / 8884	8080 / 8081
mosca	test.mosca.io	1883	80
HiveMQ	broker.mqttdashboard.com	1883	

QoS 0 - No máximo uma vez

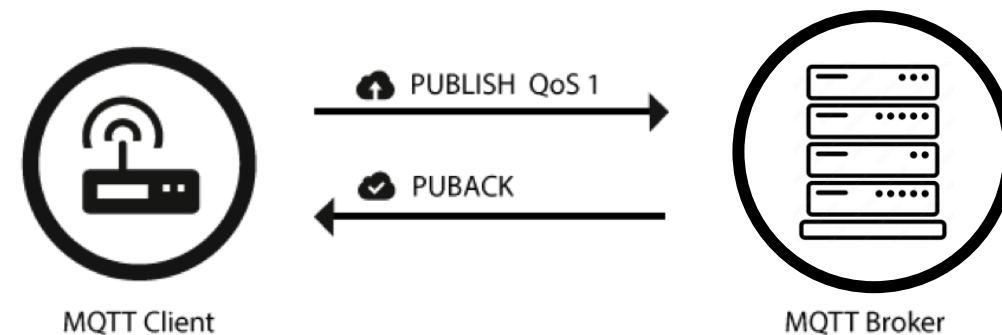


- Conhecido como *fire and forgot* (atirar e esquecer)
- a mensagem é enviada apenas uma vez e não haverá passos seguintes, dessa forma a mensagem não será armazenada, nem haverá um feedback para saber se ela chegou ao destinatário.
- Esse modo de transferência é o mais rápido, porém o menos seguro.

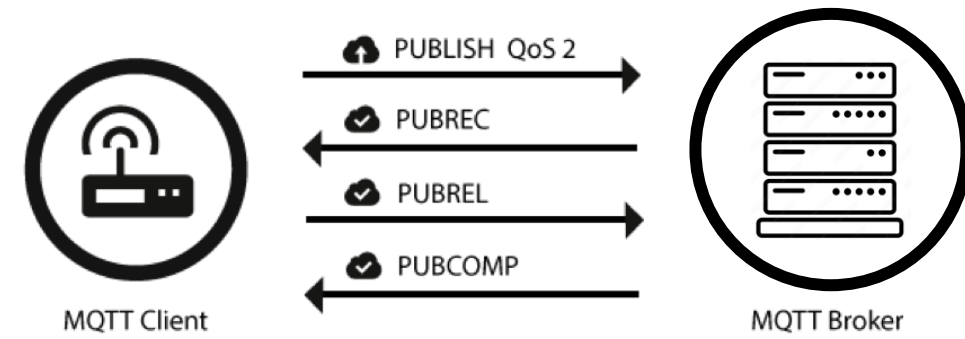


QoS 1 - Pelo menos uma vez

- Nesse modo de transferência, a mensagem é entregue pelo menos uma vez, havendo uma espera da recepção de feedback da entrega da mensagem, o chamado PUBACK.
- Não recebendo o PUBACK, a mensagem continuará sendo enviado até que haja o feedback. Nesse QoS pode acontecer da mensagem ser enviada diversas vezes e ser processada diversas vezes.



QoS 2 - Exatamente uma vez



- A mensagem é entregue exatamente uma vez, necessitando que a mensagem seja armazenada localmente no emissor e no receptor até que seja processada.
- É necessário o envio de 2 pares de request-response(chamado de four-part handshake),
 - 1 - o envio da mensagem(PUBLISH),
 - 2 - a resposta de recepção(PUBREC),
 - 3 - o aviso do recebimento do PUBREC(PUBREL)
 - 4 - e a confirmação de que o processo foi concluído e pode ser feita a exclusão(PUBCOMP).
- Após o recebimento do PUBREL, o receiver pode excluir a mensagem e ao sender receber o PUBCOMP ele poderá excluir a mensagem.

Vantagens do MQTT

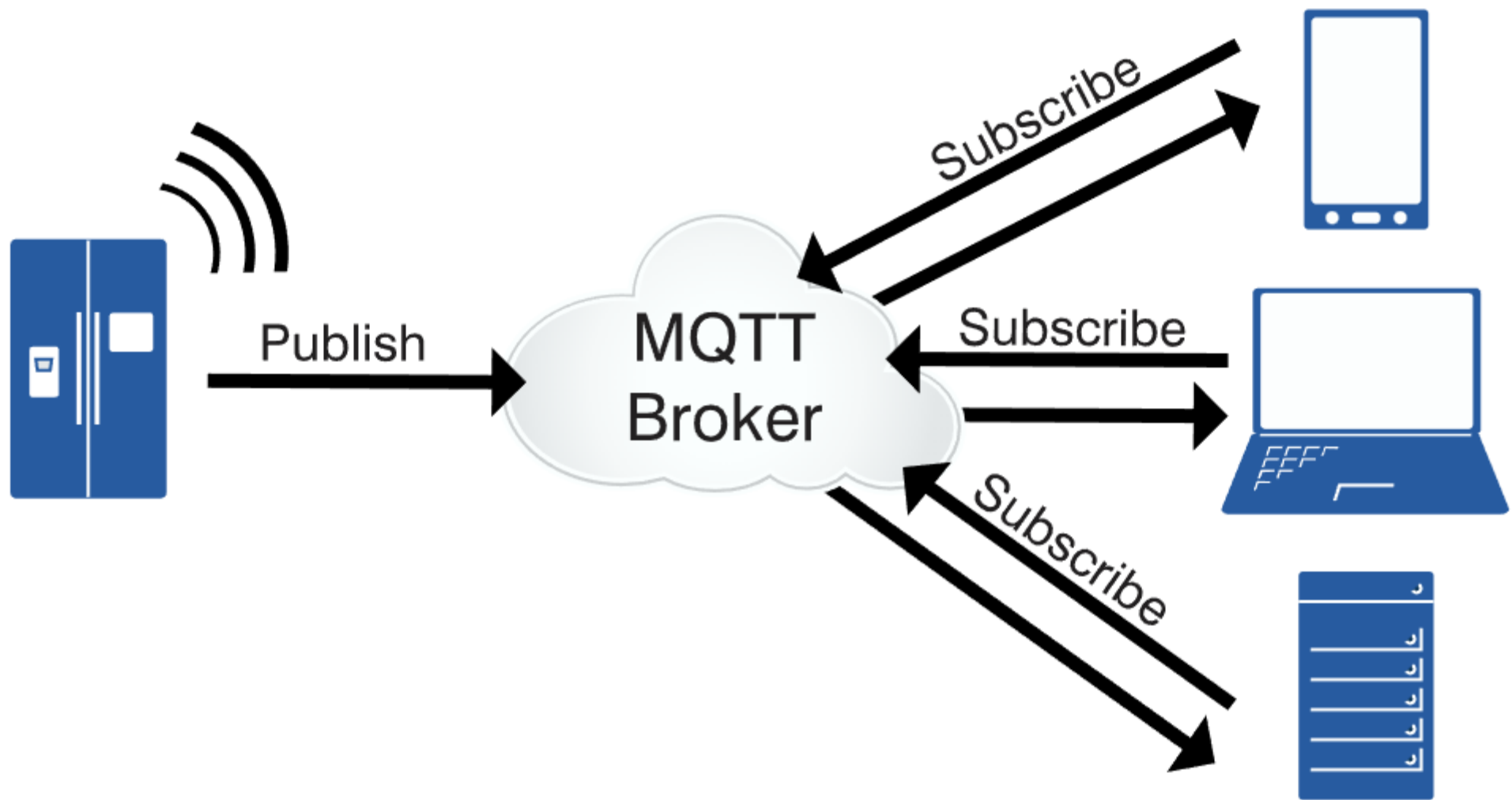
- Baixo consumo de memória, baixa necessidade de processamento para o envio de mensagem e baixo consumo de banda.
- Como o **publisher** não envia a informação direto para os subscribers, ele não precisa guardar a informação de todos os seus subscritores e nem precisa fazer várias envios de informação(uma para cada subscriber).
 - Apenas é necessário que ele realize um envio de informação para o broker com a informação que ele quer que seja enviada daquele tópico, dessa forma o processamento realizado e o consumo de memória do Publisher pode ser reduzido.
- O header de uma mensagem no protocolo MQTT é muito menor do que um Header no protocolo HTTP, o que economiza muito o consumo de banda.

Vantagens do MQTT sobre o HTTP

Baixo consumo de memória,

Baixa necessidade de
processamento para o envio de
mensagem

Baixo consumo de banda.

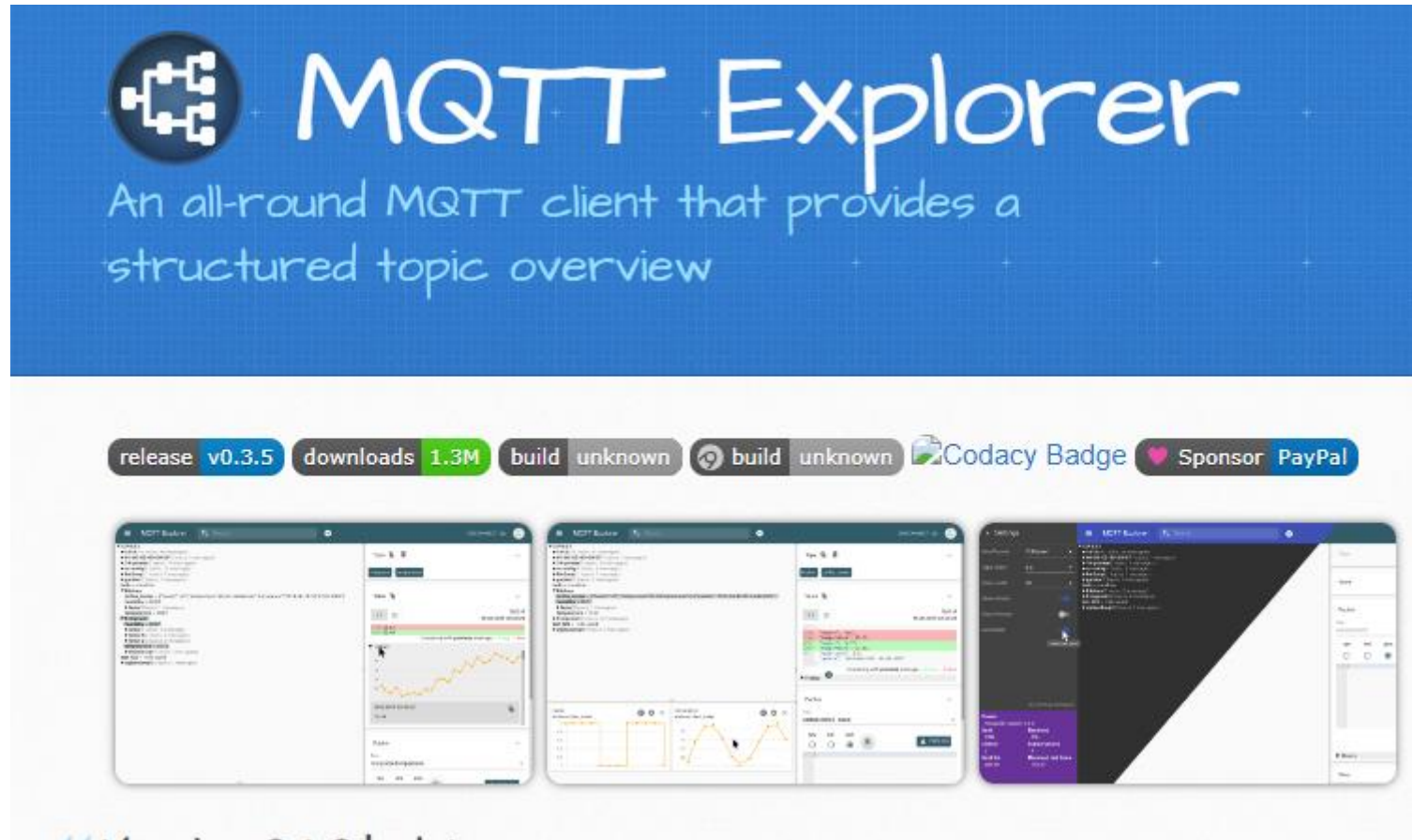


paradigma Publish- subscribe

- Nesse paradigma existem três atores,
 - Broker
 - Publisher
 - Subscriber.

Software mqtt explorer

- <http://mqtt-explorer.com/>



- WilliamTCC
- 123456

Application Edit View

MQTT Explorer

Search...

DISCONNECT

▼ mqtt.iotsistemas.com.br

- ▶ \$SYS (42 topics, 751 messages)
- ▼ aula
 - teste = ola mundo

Topic

aula / teste

Value

Publish

Topic

aula/teste

raw xml json

PUBLISH

ola mundo

QoS 0 retain

