

Avaliação dos Diferentes Protocolos de Aplicação numa implementação OM2M no ESP32

Artur Ribeiro
SDIS - MIEEC
FEUP
Porto, Portugal
up201404880@fe.up.pt

Ricardo Esteves
SDIS - MIEEC
FEUP
Porto, Portugal
up20140535@fe.up.pt

Resumo—Dado ao crescente aumento de soluções IoT, e de modo a existir uma plataforma horizontal, escalável e que garantes interoperabilidade, foram desenvolvidos *standards*, OneM2M.

O trabalho em questão teve como objectivo avaliar, num implementação os OM2M, o impacto que os diferentes protocolos de aplicação, têm no tempo de entrega de uma mensagem. Foram estudados os protocolos: HTTP, MQTT e COAP, na qual se concluiu que o COAP, era o protocolo que garante um tempo de entrega menor da mensagem, bem como, o que garante uma menor diferença de tempo entre mensagens recebidas, numa situação com múltiplos dispositivos receptores.

I. INTRODUÇÃO

Muitas plataformas que suportam o desenvolvimento de soluções para a Internet das coisas (IoT) tem sido desenvolvidas [1]. No entanto, cada uma destas plataformas, tem sido desenvolvidas de modo a satisfazer problemas específicos nos vários domínios (transportes, smart homes, smart cities, saúde, por exemplo).

Surgiu assim, a necessidade de desenvolver uma plataforma IoT *standard*, de modo a ser uma plataforma horizontal, isto é, que *podesse* ser a base das várias soluções IoT nos vários domínios. Foi criado um projeto global, *oneM2m*, com vista a criar as especificações que garantissem escalabilidade e interoperabilidade numa plataforma IoT.

Várias implementações, que tem por base as especificações *oneM2M* foram criadas, um exemplo é a plataforma OM2M, usada neste trabalho.

Esta plataforma *suporta a utilização de* vários protocolos de aplicação diferentes (HTTP, MQTT, COAP).

O principal objectivo do trabalho *realizada*, foi avaliar a o efeito, em termos do tempo que *um* mensagem demora a ser transmitida, usando os vários protocolos de aplicação, suportados pela implementação OM2M.

II. DESCRIÇÃO DA SOLUÇÃO

A solução centra-se na implementação do protocolo OneM2M no microcontrolador ESP32, utilizando e comparando três protocolos de aplicação distintos: HTTP, MQTT e CoAP. A métrica de avaliação é o tempo de atraso nas mensagens (round-trip delay) entre o nó publisher e o(s) nó(s) subscriber(s), bem como o desfasamento na chegada das mensagens aos subscribers. É utilizado um computador

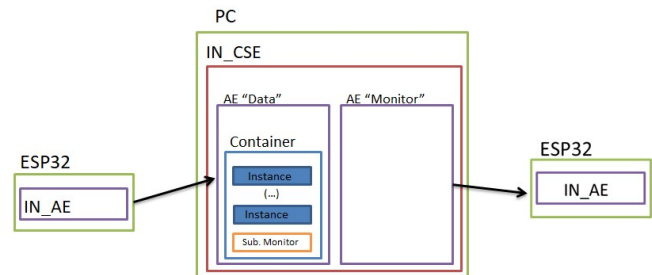


Figura 1. Implementação da arquitetura OM2M utilizando HTTP e CoAP

a correr um broker OpenM2M num sistema Ubuntu ligado através de uma rede Wi-Fi doméstica a vários ESP32, que podem funcionar como publisher ou subscriber.

A. HTTP

1) *Protocolo*: O protocolo de aplicação HTTP (HyperText Transfer Protocol) segue o paradigma Client-Server. Este protocolo é a base da comunicação de dados na World Wide Web (WWW). É construído sobre TCP/IP, o que lhe garante um canal de comunicação fiável.

2) *Implementação*: No diagrama 1 encontra-se representado a arquitetura que foi implementada para realizar este ensaio experimental. Onde, como já foi referido, foi colocado num computador a correr o broker OneM2M. Depois, e antes de poder fazer haver a troca de dados entre os dispositivos, foi necessário realizar as seguintes acções:

- Registo de cada um dos dispositivos no servidor (AE "data" e "Monitor");
- Criação da identidade "Container" por parte do Esp que pretende publicar os dados.
- Subscrição da identidade "Container" por parte do Esp que pretende receber os dados.

Após isto, o esp (associado à AE "data") pode publicar dados dentro da identidade "Container", que por sua vez serão entregues ao esp (associado à AE "Monitor"). Para realizar estas operações, foram usadas as bibliotecas fornecidas, na qual implementa estas acções, segundo as especificações OneM2M.

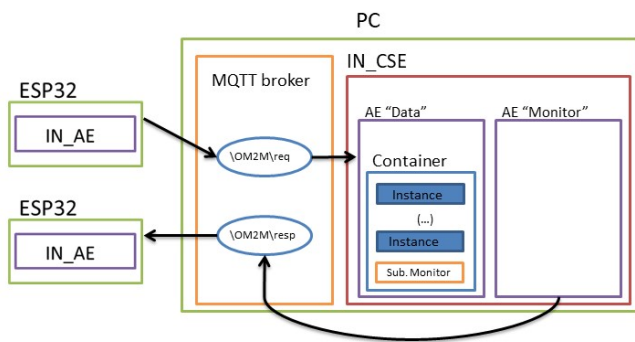


Figura 2. Implementação da arquitetura OM2M utilizando MQTT

B. MQTT

1) *Protocolo*: O protocolo de aplicação MQTT (Message Queueing Telemetry Transport) segue o paradigma Publisher-Subscriber. Foi desenhado para dispositivos **restringidos** e redes com baixa largura de banda, alta latência ou com perdas. Os objetivos do protocolo focam em assegurar fiabilidade e garantia de entrega de mensagens (a um certo nível). É desenvolvido sobre TCP. Como segue o paradigma Publisher-Subscriber, é necessário a utilização de um *broker* MQTT.

2) *Implementação*: Para este caso, e uma vez que o OM2M não implementa um *broker* MQTT foi necessário colocar a correr no mesmo computador um *broker* MQTT (Mosquitto), onde depois o *broker* OM2M realizava a subscrição de um determinado tópico no *broker* M2M, Diagrama 2. A sequência de ações **iniciais** realizadas pelos dispositivos, são as mesmas que as que foram descritas no protocolo HTTP, exceptuando-se, que neste caso o **esp** (associado à AE "Monitor") terá também de subscrever um determinado tópico do *broker* MQTT de **modo receber** os valores que são publicados pelo outro **esp**. As medições foram realizadas com um **Qos** de 0 (*at most once*) onde não existem garantias de entrega adicionais, para além das realizadas pelo canal TCP.

C. CoAP

1) *Protocolo*: O protocolo de aplicação CoAP (Constrained Application Protocol) [2] é um protocolo especializado para dispositivos restringidos, e está definido no RFC 7252 [3]. Foi desenvolvido para usos em redes com baixa largura de banda e/ou com perdas. Permite que dispositivos comuniquem nessas redes, entre dispositivos restringidos e nós gerais da Internet, ou entre dispositivos em redes diferentes restringidas ligadas através da Internet.

Como é desenvolvido sobre UDP, pode utilizar *multicast*, apesar de não usufruir de um canal fiável. Possui uma estrutura simples, e com baixo *overhead*, o que lhe garante alta eficiência. Isto é de extrema importância para nós IoT (Internet of Things) e M2M (Machine To Machine).

2) *Implementação*: Foi seguida a mesma arquitetura que o protocolo HTTP, diagrama 1. Foram usadas mensagens do tipo **Non-Conformable**, onde não existe qualquer tipo de garantia na entrega das **mensagens**. Em mensagens do tipo

Conformable o protocolo garante uma comunicação fiável sobre UDP.

D. Medições

1) *O que foi medido*: A avaliação dos diferentes protocolos de aplicação (HTTP, MQTT e CoAP) numa implementação OneM2M foi feita com as seguintes métricas:

- **Round-trip delay**: Atraso entre o envio de uma mensagem num nó *publisher* e a receção num nó *subscriber*;
- Diferença no tempo de chegada: Atraso na receção de uma mensagem em nós *subscribers* diferentes.

2) *Como foi medido*: O computador que corria o *broker* em Java e todos os ESP32 utilizados estavam ligados à mesma rede Wi-Fi doméstica e foram colocados em proximidade uns dos outros, com o *router/AP* **localizado um piso acima**. De modo a obter as medições dos tempos, todos os ESP32 foram ligados por USB ao mesmo computador. Exatamente antes de um nó *publisher* enviar uma mensagem, ou exatamente depois de um nó *subscriber* receber uma mensagem, o ESP32 efetua um *print* através de comunicação série. Utilizando o programa *extraPutty*, foi possível obter os *timestamps* de envio e receção de mensagens sob o mesmo relógio (do computador). Os *logs* foram analisados por um programa desenvolvido de modo a retirar apenas a informação relevante.

III. RESULTADOS

Os resultados obtidos são visíveis nos gráficos 3 4 5.

O gráfico 3 ilustra a distribuição do atraso de chegada de mensagens entre o nó *publisher* e um nó *subscriber*, mediadas por um *broker*. Da esquerda para a direita temos as implementações usando HTTP, MQTT e CoAP.

O gráfico 4 permite visualizar a distribuição do atraso de mensagens entre o nó *publisher* e dois nós *subscribers*, mediadas pelo mesmo *broker*. Cada protocolo de aplicação tem duas medições (uma para cada *subscriber*) e correspondem, da esquerda para a direita, a HTTP, MQTT e CoAP.

O gráfico 5 ilustra a diferença em valor absoluto dos tempos de chegada da mesma mensagem aos dois nós *subscribers*. Corresponde novamente, da esquerda para a direita, aos protocolos de aplicação HTTP, MQTT e CoAP.

IV. ANÁLISE CRÍTICA

O protocolo de aplicação CoAP apresenta tempos de entrega de mensagem com uma média de aproximadamente 25ms e desvio-padrão de 35ms (após filtragem de tempos excessivamente altos, causados por retransmissão a nível físico (**camada 1 OSI**) originadas por possíveis colisões na rede) para um *subscriber*. Os valores aumentam ligeiramente (cerca de 3ms) para o caso com dois *subscribers*.

No caso de HTTP, os tempos de entrega têm média de 61ms e desvio-padrão de 40ms (após a mesma correção). O aumento para o caso de dois *subscribers* também é de cerca de 5ms, com um desvio-padrão idêntico.

No caso de MQTT, os tempos de entrega são muito superiores aos outros dois protocolos. Com um *subscriber*, o tempo de entrega médio ronda os 300ms, com um desvio-padrão de

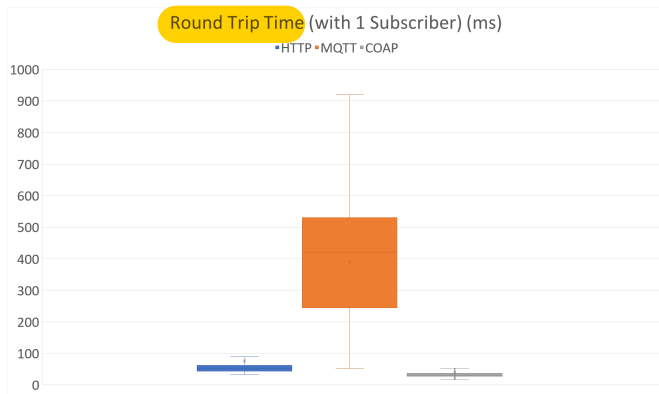


Figura 3. Distribuição dos tempos de chegada das mensagens com um subscriber

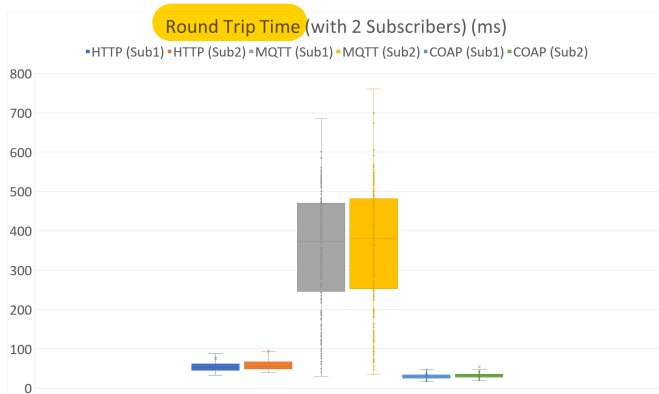


Figura 4. Distribuição dos tempos de chegada das mensagens com dois subscribers

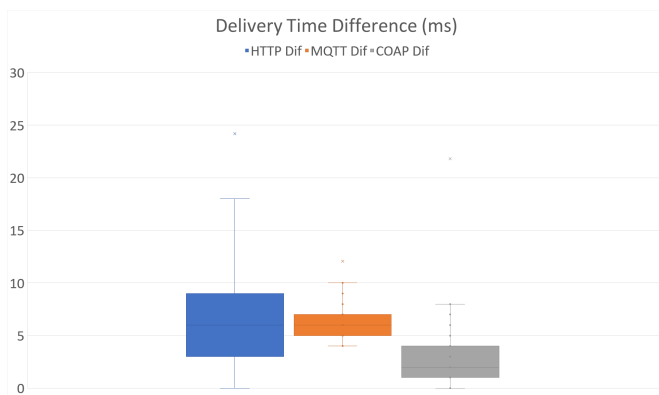


Figura 5. Distribuição da diferença de chegada das mensagens

300ms. Utilizando dois *subscribers*, o tempo médio sobe para cerca de 360ms, com um desvio-padrão de 180ms.

Analisando agora a diferença de tempos de entrega a dois *subscribers*, verificamos que o HTTP apresenta diferenças de 9ms (com desvio-padrão de 60ms); o MQTT tem uma média de 12ms (com um desvio-padrão de 40ms); o CoAP apresenta uma média de 3ms (com um desvio-padrão de 4ms).

Estes resultados estão de acordo com o expectável. O protocolo MQTT necessita de um broker adicional, o que aumenta o tempo de processamento intermédio (em *brokers*) necessário. O protocolo HTTP utiliza canais de comunicação TCP (tal como MQTT), ao passo que CoAP utiliza canais UDP. Este facto, aliado à simplicidade e baixo *overhead* do CoAP explicam a redução de atraso obtida em comparação com HTTP e MQTT.

Contudo, devido à utilização de canais UDP (não fiáveis, em que não existe garantia de entrega de mensagem), os protocolos HTTP e MQTT podem ser considerados numa utilização que necessita de garantia de entrega de mensagens. Se existir necessidade de tempos de entrega baixos mas com garantia de entrega de mensagens, deve ser escolhido o protocolo HTTP; se, por outro lado, for mais importante a coerência no tempo de entrega a vários *subscribers*, aliado a garantia de entrega de mensagens, deve ser escolhido o protocolo MQTT.

É de notar que, mesmo utilizando o protocolo CoAP com canais de comunicação UDP, não foi observada a perda de nenhuma mensagem nos testes efetuados, mesmo utilizando mensagem do tipo *Non-Conformable*.

Concluindo, segundo as métricas que foram analisadas, a implementação de OM2M utilizando o protocolo de aplicação CoAP apresenta o tempo de entrega de mensagens (*publisher-subscriber(s)*) mais baixo e mais constante. Esta implementação também apresenta maior coerência no tempo de entrega da mesma mensagem a vários *subscribers*.

V. PARTICIPAÇÃO DO GRUPO

A participação dos elementos do grupo para a realização do trabalho em questão foi:

Artur Ribeiro - 50%;

Ricardo Esteves- 50%;

REFERÊNCIAS

- [1] H. Park, H. Kim, H. Joo, and J. Song, "Recent advancements in the internet-of-things related standards: A onem2m perspective," *ICT Express*, vol. 2, no. 3, pp. 126 – 129, 2016, special Issue on ICT Convergence in the Internet of Things (IoT). [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S2405959516300911>
- [2] K. H. Z. Shelby, ARM and C. Bormann, "Coap - rfc 7252 constrained application protocol," 2014, last accessed 17 January 2019. [Online]. Available: <https://www.rfc-editor.org/rfc/rfc7252.txt>
- [3] "The constrained application protocol (coap)," 2016, last accessed 17 January 2019. [Online]. Available: <http://coap.technology/>