

Protocolos para IoT: Comparativo entre MQTT e CoAP

Ricardo Artur Staroski
ricardo.staroski@senior.com.br

1. Introdução

Esta pesquisa se propõe a comparar as diferenças de performance entre os protocolos Message Queue Telemetry Transport (MQTT) e Constrained Application Protocol (Coap). Como base de pesquisa utilizou-se o artigo de Thangavel [1], a documentação do projeto Paho [2], a documentação do projeto Californium [3] bem como o próprio código fonte das ferramentas Paho, Moquette e Californium [4, 5, 6].

2. Metodologia

Serão criadas duas aplicações cliente-servidor de funcionalidade análoga, uma utilizando o protocolo MQTT e outra o CoAP. Os clientes irão simular um dispositivo publicando dados de sensores a cada 300 milissegundos em um servidor. Os dados para envio serão extraídos de amostras de dados em arquivos CSV. Para cada conjunto de dados enviados, será mensurada a quantidade de bytes enviados, o tempo de envio e a memória em uso. Para o comparativo, serão analisadas apenas os primeiros mil dados enviados.

Para a criação das aplicações, são necessárias as bibliotecas do Paho, Moquette e Californium [4, 5, 6], portanto, faz-se necessário efetuar o download dos repositórios e compilar os códigos fontes dos mesmos. As etapas de download e compilação destas bibliotecas é um processo um tanto extenso e está documentado nos respectivos repositórios, portanto não será descrita neste artigo.

Com as bibliotecas compiladas, deve-se criar dois projetos Java, incluindo estas bibliotecas ao classpath da aplicação. Conforme Thangavel [1], o protocolo MQTT é baseado em TCP IP e o CoAP é baseado em REST, mesmo assim será implementado de comportamento análogo utilizando as duas arquiteturas [7, 8].

3. Resultados

Executando os códigos desenvolvidos [7, 8] com amostras de mil dados, obteve-se os valores apresentados no Quadro 1.

MQTT		
tamanho médio dos pacotes (bytes)	tempo médio de transferência (ms)	memória alocada (MB)
80	4	3,9
CoAP		
tamanho médio dos pacotes (bytes)	tempo médio de transferência (ms)	memória alocada (MB)
80	2	4,4

Quadro 1 – Medições após 1000 pacotes de dados enviados.

4. Conclusão

Observa-se que o protocolo CoAP tende a ser uma boa escolha de protocolo ao desenvolver em Java. Embora suas bibliotecas necessitem de aproximadamente 12% mais memória do que as bibliotecas do MQTT, sua vantagem é a velocidade de transmissão dos pacotes de dados, com o uso de CoAP obteve-se uma redução de tempo de 50% em relação ao MQTT.

Referências

- [1] D. Thangavel, “Performance Evaluation of MQTT and CoAP via a Common Middleware”, ISSNIP, Pag 1-6, 2014
- [2] Paho, “Eclipse Paho”, <https://eclipse.org/paho/>, 2016
- [3] Californium, “Eclipse Californium”, <http://www.eclipse.org/californium/>, 2016
- [4] Paho GitHub, “Paho MQTT Java”, <https://github.com/eclipse/paho.mqtt.java/>, 2016
- [5] Moquette GitHub, “Moquette”, <https://github.com/andresel/moquette/>, 2016
- [6] Californium GitHub, “Californium”, <https://github.com/eclipse/californium/>, 2016
- [7] Staroski MQTT, “prototipo-MQTT”, https://github.com/staroski/Desafio_Pesquisador_2016/tree/master/prototipos/prototipo-MQTT/test-MQTT/src/test/mqtt/, 2016
- [8] Staroski CoAP, “prototipo-CoAP”, https://github.com/staroski/Desafio_Pesquisador_2016/tree/master/prototipos/prototipo-CoAP/test-CoAP/src/test/coap/, 2016