UNIVERSIDADE DE ÉVORA

Trabalho prático de Redes de Computadores

Redes de Computadores

DISCENTES

Yaroslav Kolodiy nº 39859 Pedro Amaro nº42348

DOCENTES

Pedro Patinho
José Saias

Ano Letivo 2019/2020

1 Descrição do trabalho

O trabalho consiste num conjunto de aplicações que permita captar, processar e disponibilizar dados sobre a qualidade do ar em várias cidades. Este conjunto de aplicações permite interações dos administradores e dos clientes com o broker.

Os clientes podem listar locais onde existem sensores de determinado tipo, obter a última leitura de um local ou obter a última leitura de um local com base na data e hora. Os clientes ainda conseguem ativar o modo publish-subscribe com base num local de interesse, ficando assim aptos a receber leituras novas dos locais a que se subescrevem

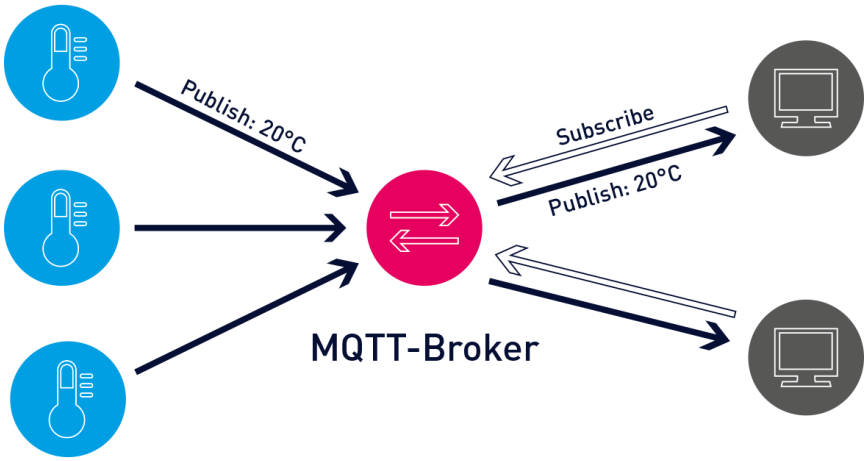
Os administradores podem obter a última leitura de um sensor com identificador X, ver os sensores registrados no broker obtendo assim as informações sobre o local, tipo de sensor e versão firmware.

O broker é uma interface de comunicação entre os clientes/administradores com os sensores. E os sensores emitem leituras sobre a qualidade do ar do poluente pretendido com intervalos desejados.

2 Decisões tomadas

Optamos por criar apenas um canal de comunicação (socket) entre cliente, administrador e sensor com o broker, todas estas ligações são feitas através do protocolo TCP. Deste modo consideramos o broker como um servidor e as restantes aplicações (sensor.py ; public_client.py ; admin_client.py) como clientes.

Segundo esta implementação obtemos o seguinte modelo:



Para troca de informação obtamos por fazer o envio de um ‘header’ fixo que contém o tamanho da mensagem, desta forma o buffer de leitura da mensagem é dinâmico.

Para interpretação das mensagens decidimos transmitir informação em forma de dicionário, implementando uma versão simplificada de JSON. Todas as aplicações suportam configuração por YAML ou por argumentos.

Durante a implementação do sensor obtamos por apenas gerar valores aleatório válidos para cada poluente.

O pucblic_client.py consiste em 3 processos, um para receber as mensagens ,outro para receber input do utilizador e o terceiro para fazer ‘listening’ para verificar a existência de conexão.

Optamos por implementar o pucblic_client.py desta forma para garantir que este recebe as mensagens do broker enquanto navegamos pelo menu do utilizador.

No admin_client.py optamos por apenas implementar dois processos, um para realizar a comunicação com o broker e outro, tal como no pucblic_client.py, para fazer ‘listening’ para verificar a existência de conexão.

3 Balanço

Durante o trabalho conseguimos implementar todas as funcionalidades pretendidas. Através deste trabalho conseguimos entender melhor a comunicação via sockets, e quais as melhores formas para a sua implementação.

Apesar do trabalho cumprir os requisitos, reconhecemos que de forma a ser possível aumentar o escalonamento de conexões e respostas seria uma mais valia criar um processo para cada pedido ao broker, deste modo evitávamos as filas de espera.