

Trabalho pratico de Redes de Computadores

Yaroslav Kolodiy nº 39859 Pedro Amaro nº42348

Ano Letivo 2019/2020

Redes de Computadores, Pedro Patinho, José Saias

# Descrição do trabalho

O trabalho consiste num conjunto de aplicações que permita captar, processar e disponibilizar dados sobre a qualidade do ar em varias cidade.

Este conjunto de aplicações permite interações dos administradores e dos clientes com o broker.

Os clientes podem listar locais onde existem sensors de determinado tipo, obter a ultima leitura de um local ou obter a ultima leitura de um local com base na data e hora. Os cliente 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.

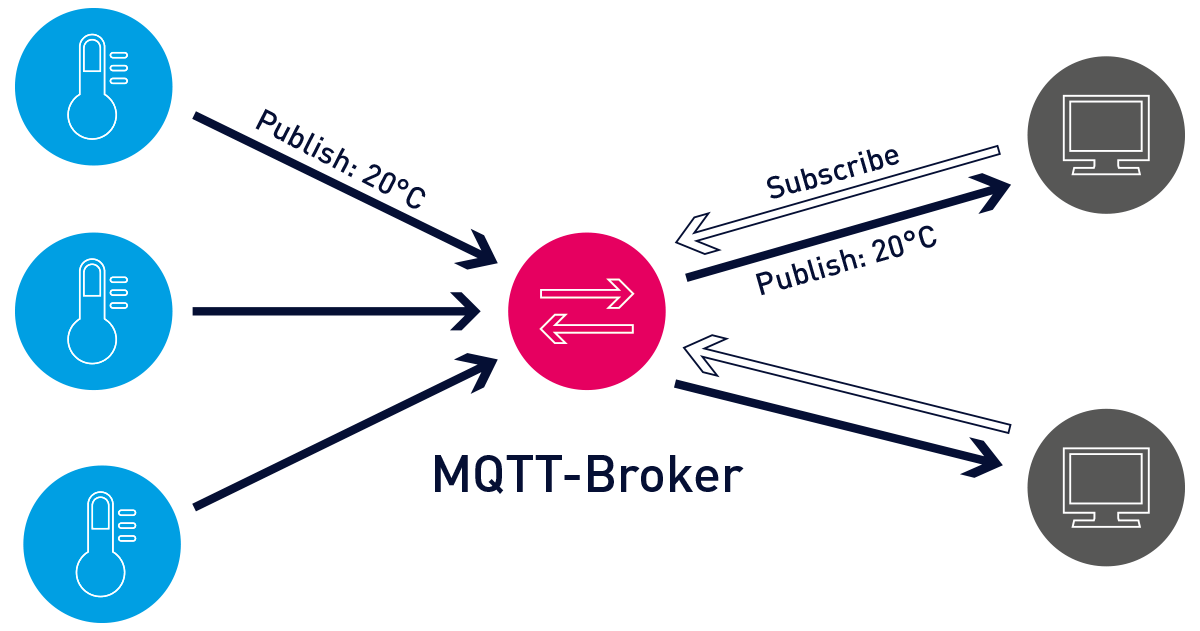
Os administradores podem obter a ultima leitura de um sensor com identificador X, ver os sensores resgistados no broker obtendo assim as informações sobre o local, tipo de sensor e versão firmware.

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.

# Decisões tomadas

Obtamos 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 contem 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 validos 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.

Obtamos 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 obtamos 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.

# 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.