



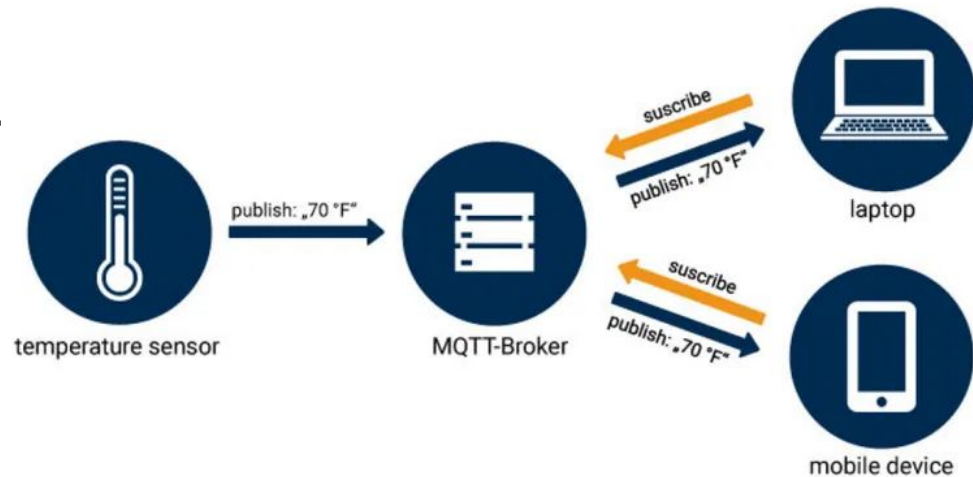
Servidor MQTT

MAC0352 - Redes de Computadores e Sistemas Distribuídos
Instituto de Matemática e Estatística • USP

Arthur Font

Protocolo MQTT

- Protocolo de mensagens
- Padrão publisher/subscriber onde as mensagens são identificadas por tópicos
- Broker: administra as informações





Funcionalidades

- Conexão de vários clientes simultaneamente (cada cliente simultâneo pode publicar ou requisitar mensagens do mesmo tópico ou de tópicos distintos);
- Inscrição de cliente em um tópico e consequente envio das mensagens deste tópico para o cliente;
- Publicação de mensagem em um tópico;
- Desconexão de cliente.



Comandos MQTT implementados

- CONNECT
- CONNACK
- SUBSCRIBE
- SUBACK
- PUBLISH
- PINGREQ
- PINGRESP
- DISCONNECT



Protocolo MQTT

- Fixed Head (Presente em todos os comandos)
 - Control Packet
 - Remaining Length
- Variable Head (Depende do comando)
 - Informações sobre a identificação do pacote
- Payload (Depende do comando)
 - Mensagem em si

Estrutura do servidor

- Para as requisições vindas de clientes, o servidor seleciona qual foi o comando recebido através de um *switch-case*, dado que cada comando MQTT possui um pacote de controle.
- Para responder aos clientes, o servidor envia a mensagem correspondente à requisição feita:
 - CONNECT -> CONNACK
 - SUBSCRIBE -> SUBACK
 - PINGREQ -> PINGRESP

Pontos principais

- CONNECT (0x10)
 - O cliente envia uma requisição para conectar-se
 - O servidor lê a requisição recebida e responde através do CONNACK
- CONNACK (0x20)
 - O servidor responde ao cliente que ele se conectou com sucesso através do campo <Return Code>
 - <Return Code> = 0x00 (Sucess)

Pontos principais

- PUBLISH (0x30)
 - O cliente publica uma mensagem no tópico especificado
 - O servidor lê a mensagem recebida e publica esta mensagem no tópico especificado
- Implementação:
 - Foi utilizado um sistema de arquivos nomeados por tópicos para realizar a comunicação interprocessos
 - O servidor escreve no arquivo especificado caso já exista, ou cria um novo arquivo caso o mesmo não exista.

Pontos principais

- SUBSCRIBE (0x82)
 - O cliente envia uma requisição para inscrever-se no tópico especificado
 - O servidor lê a mensagem recebida e responde através do SUBACK
- Implementação:
 - Checa se o arquivo existe e se foi recentemente modificado. Assim, o cliente só lê a mensagem quando ocorre alguma mudança no arquivo referente ao tópico inscrito



Pontos principais

- SUBACK (0x90)
 - O servidor responde ao cliente que ele se inscreveu com sucesso no tópico desejado
- PINGREQ (0xC0)
 - O cliente *subscriber* envia requisições de ping ao servidor a cada 60 segundos (por padrão)
 - O servidor responde ao cliente através do PINGRESP



Pontos principais

Para a implementação dos comandos PINGRESP e DISCONNECT, observa-se que:

- Para o cliente Pub, não houve a necessidade da implementação, pois o cliente publica a mensagem e a seguir já se desconecta
- Já para o cliente Sub, houve a necessidade de criar um novo processo para atender ambas as requisições, visto que o processo filho está destinado a ler o arquivo sempre que haja mudanças



Pontos principais

- PINGRESP (0xD0)
 - O servidor confirma ao cliente Sub que ele está conectado
- DISCONNECT (0xE0)
 - O cliente informa sua desconexão ao servidor.

Análise de desempenho

3 cenários:

- 1. Somente o servidor iniciado
 - Rede: 0.000 KB/sec
 - CPU: 0,0% (Sleeping)
- 2. Servidor com 2 clientes conectados simultaneamente
 - Rede: 0.484 Sent KB/sec
 - CPU: 0,0%
- 3. Servidor com 100 clientes conectados simultaneamente
 - Rede: 22.587 Sent KB/sec
 - CPU: 0,0%

Análise de desempenho

O método de avaliação que utilizei para avaliar a carga na rede e na CPU foi:

- CPU:
 - Comando: `$ps -ppid $(pgrep -d',' -f mqtt) -o %cpu,%mem,cmd`
 - Este comando mostra a porcentagem de uso de cada processo filho direto ou indireto do programa principal `mqtt-server`
- Rede:
 - Comando: `$sudo nethogs -a`
 - Este comando exibe a carga da rede utilizada por cada processo. A flag `-a` é essencial para exibir os processos de todas as interfaces da rede, incluindo loopbacks



Ambientes utilizados

- Computacional
 - OS: Ubuntu 20.04
 - Processador: Intel(R) Core(TM) i5-3317U CPU @ 1.70GHz
 - CPU(s): 4
 - Memória RAM: 4 GB
- Rede
 - Largura de banda contratada: 100 Mbps
 - Velocidade de Download \approx 100 Mbps
 - Velocidade de Upload \approx 100 Mbps



Referências

- <https://mosquitto.org/>
- <https://packages.debian.org/bullseye/mosquitto-clients>
- <https://docs.oasis-open.org/mqtt/mqtt/v5.0/mqtt-v5.0.pdf>



Obrigado pela atenção!