

IoT 환경에서 대량 접속처리를 위한 MQTT 메시지 서버 설계 및 구현

차우석*, 유은국*, 김영준*, 김진순*
 *에이치투오시스템테크놀로지(주)
 e-mail : wscha@h2osystech.com

Design and Implementation of MQTT Message Server for a massive connection processing in IoT Environment

Woosuk Cha*, Eunkuk Yoo*, Yeongjun Kim*, Jinsoon Kim*
 *H2O System Technology

요 약

본 논문은 IoT 환경에서 MQTT 브로커와 대량 클라이언트 간 효율적인 접속을 지원하기 위해 대량 접속처리 기능을 제공하는 CA(Connection Agent)를 포함한 MQTT 메시지 서버를 설계, 구현하였다. MQTT 프로토콜은 MQTT 브로커와 MQTT 클라이언트로 구성되며, 각 MQTT 클라이언트는 MQTT 브로커의 중재를 받아 Pub/Sub 방식으로 메시지를 상호 전송한다. 이를 위해 MQTT 프로토콜은 MQTT 브로커와 MQTT 클라이언트 간 접속기능만을 제공한다. 실험결과에서 MQTT 메시지 서버는 초당 평균 12,500 건의 클라이언트 접속요청을 처리하였고, 20 만건의 접속요청 처리에 16 초가 소요되었다.

1. 서론

최근 IDC(International Data Corporation) 보고서에 따르면 2020 년에 약 220 억 개의 IoT 장치들이 인터넷에 연결될 것으로 예측하고 있다. IoT 환경에서 장치 간 정보를 교환하기 위해 XMPP, CoAP, MQTT 등의 다양한 프로토콜이 연구되고 있다[1].

MQTT(Message Queue Telemetry Transport)은 IoT 장치 간 Pub/Sub 방식으로 양방향 메시지 전송을 지원하는 국제 표준 프로토콜이다[2]. MQTT 는 MQTT 브로커와 MQTT 클라이언트로 구성되며, 각 MQTT 클라이언트는 MQTT 브로커의 중재를 받아 메시지를 상호 전송한다.

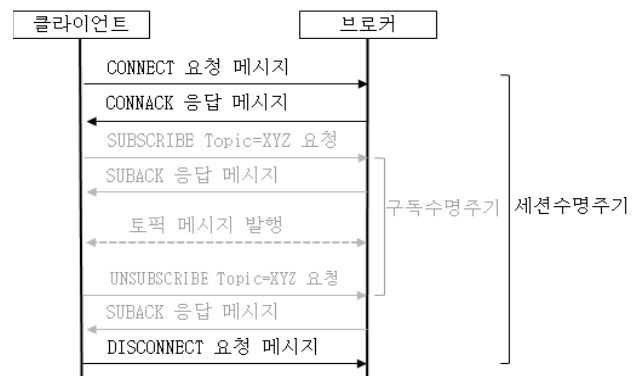
IBM MessageSight 는 MQTT 브로커와 다수의 클라이언트 및 디바이스 간 접속과 다량의 메시지를 처리하도록 디자인된 어플라이언스 기반 메시징 시스템으로써, 100 만 개의 동시 접속을 지원한다[3].

MQTT 프로토콜은 기본적인 브로커와 클라이언트 간 접속기능을 제공하지만, IoT 환경에서 브로커와 대량 클라이언트 간 효율적인 접속을 지원할 수 있는 수단을 필요로 한다.

본 논문은 MQTT 에 메시지 서버에 CA(Connection Agent) 모듈을 추가하여 MQTT 메시지 서버와 대량 클라이언트 간 효율적인 접속처리를 지원할 수 있도록 MQTT 에 메시지 서버를 설계하고 구현한다.

MQTT 브로커와 클라이언트 간 접속
 그림 1 은 MQTT 브로커와 클라이언트 간 접속과 토픽메시지의 Pub/Sub 을 간략하게 도식한 것이다.

MQTT 브로커와 클라이언트 간 접속은 CONNECT 메시지에 의해 접속이 시작하고, DISCONNECT 메시지에 의해 접속이 종료된다. MQTT 브로커와 접속을 요청하는 클라이언트는 TCP/IP 소켓을 연결한 후 명시적으로 종료하거나 네트워크 사정에 의해 접속이 끊어질 때까지 접속상태를 유지한다.



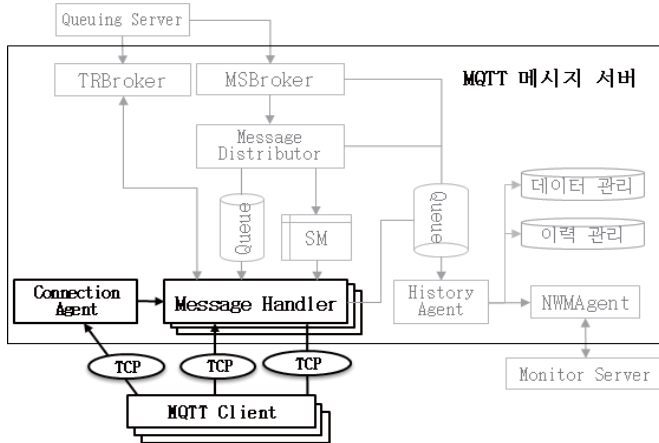
(그림 1) MQTT 접속 절차

2. MQTT 메시지 서버

MQTT 메시지 서버는 MQTT 브로커 역할을 수행하며, MQTT 클라이언트와 통신 접속기능과 메시지 배포기능을 제공한다.

3.1 MQTT 메시지 서버 설계

그림 2 는 MQTT 메시지 서버의 전체 구조를 도식한다. MQTT 메시지 서버에서 CA 와 MH(Message Handler)가 MQTT 브로커와 클라이언트 간 접속기능을 제공한다.



(그림 2) MQTT 메시지 서버 구조

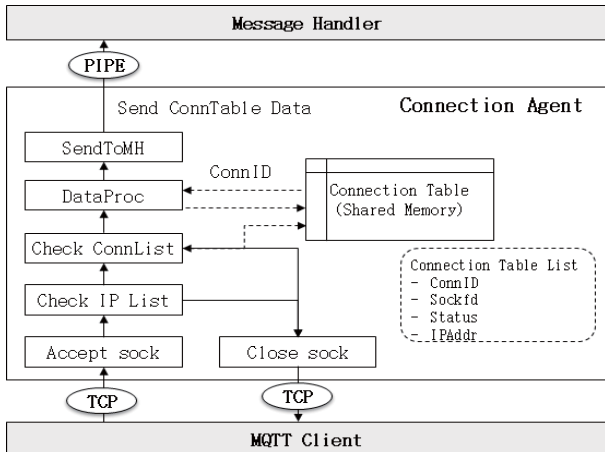
MQTT 메시지 서버의 구성 모듈별 기능은 표 1 과 같다.

<표 1> MQTT 메시지 서버 구성 모듈별 기능

모듈명	기능 내용
Connection Agent	- 클라이언트와의 소켓 연결 - 연결 소켓을 MH로 분배
Message Handler	- MD로부터 배포 메시지 접수하여 접속 클라이언트에게 전송
MSBroker	- 큐잉서버로부터 배포 메시지 수신 - 큐잉서버는 기존 사용자 서비스와 메시지 서버간 인터페이스 기능 제공
Message Distributor	- 배포 메시지를 MH로 분배
TRBroker	- 큐잉서버와의 Request/Reply 트랜잭션 처리
History Agent	- 배포 메시지 이력관리

3.2 CA(Connection Agent) 구조

그림 3 은 CA 모듈의 전체 구조를 도식하고 있다.



(그림 3) MQTT 메시지 서버 구조

CA 모듈은 MQTT 브로커와 클라이언트간 통신 접속을 관리하며, 연결된 TCP 소켓을 MH로 전달하는 프록시 기능을 수행한다. CA 구성 모듈별 기능은 표 2와 같다.

<표 2> CA 구성 모듈별 기능

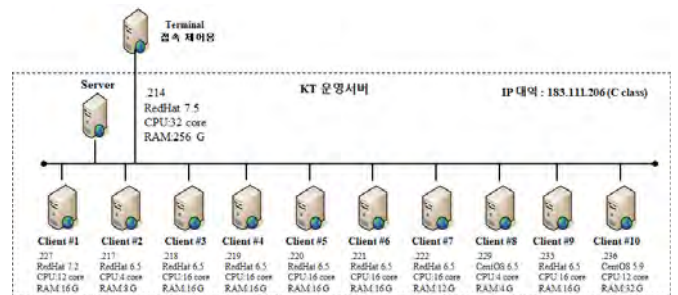
모듈명	기능 내용
Accept sock	- 외부 MQTT 클라이언트와 통신 접속을 수신
Check IP List	- 보안을 위한 MQTT 클라이언트 IP 확인
Check ConnList	- Connection Table을 확인하여 접속 가능여부 확인 및 준비
Close sock	- 통신 접속 해제
DataProc	- Connection Table에서 ConnID를 추출하여 저장 및 데이터 처리
SendToMH	- 접속된 MQTT 클라이언트 정보를 MH에게 PIPE로 전달

CA 모듈이 MQTT 클라이언트로부터 접속 요청을 수신후 처리절차는 다음과 같다.

- MQTT 클라이언트는 TCP 소켓을 열고, CA 모듈의 소켓에 접속한다.
- 소켓 모듈은 DataProc에게 거부 IP 검사를 요청한다.
- Black List에서 거부 IP를 조회한다.
- Black List는 접속 승인 여부를 알려준다.
- 접속 거부시 클라이언트와의 소켓을 닫는다.
- DataProc은 Connect Table을 조회하여 Conn id를 추출한다
- Conn id 생성이 실패하면 클라이언트와의 소켓을 닫는다.
- 접속이 허용되었다면, 접속정보를 SendMH에 전달한다.
- SendMH는 MH 모듈에 접속정보, Socket fd를 전달한다. Socket fd 전달 후 MH 모듈은 소켓에 있는 Connect 패킷 처리를 시작한다.

3.3 MQTT 메시지 서버 구현 및 실험

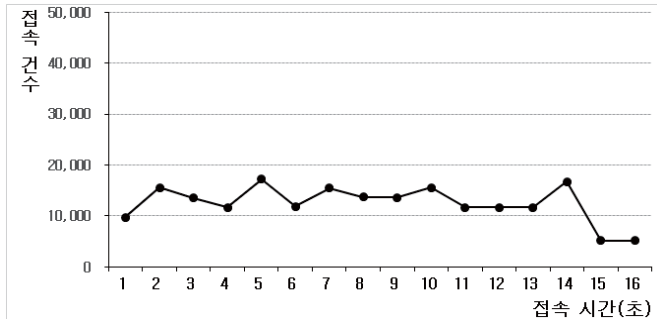
MQTT 메시지 서버는 LINUX 운영체제(CentOS) 환경에서 GCC 4.8.5 도구를 이용하여 C 언어로 구현하였다. 실험은 KT 운용센터에서 서버 1 대와 클라이언트 10 대를 대상으로 그림 4와 같은 실험환경을 구성하였다.



(그림 4) MQTT 메시지 서버 접속실험 환경

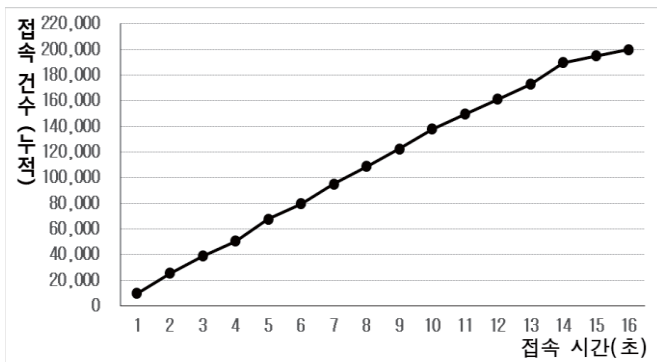
각 클라이언트는 2 만개의 TCP 세션을 동시에 생성하고, MQTT CA 에게 접속을 요청한다. CA 는 각 클라이언트와의 연결 소켓을 MH 에게 분배하고, MH 는 각 클라이언트와의 접속을 유지한다.

그림 5 는 초당 접속처리 건수를 그래프로 도식한다.



(그림 5) 초당 접속처리 건수

그림 6 은 초당 접속처리 건수를 누적하여 그래프로 도식한다.



(그림 6) 초당 접속처리 누적건수

실험결과에서 MQTT 메시지 서버는 초당 평균 12,500 건의 클라이언트 접속 요청을 처리하였고, 20 만건의 접속요청 처리에 16 초가 소요되었다.

3. 결론

본 논문은 MQTT 브로커 역할을 수행하는 MQTT 메시지 서버를 설계, 구현하였다. MQTT 메시지 서버는 CA 모듈을 포함하며, CA 모듈은 MQTT 메시지 서버와 대량 클라이언트 간의 효율적인 접속처리를 지원한다.

실험결과에서 초당 평균 12,500 건의 클라이언트 접속 요청처리와 20 만건의 접속요청 처리에 16 초가 소요됨을 확인하였다. 향후 MQTT 메시지 서버는 100 만건의 동시 접속처리를 목표로 하고 있다.

감사의 글

본 연구는 서울시 산학연 협력사업(FI17 0001) 지원으로 수행되었습니다.

참고문헌

- [1] S.ELHADI, A.Marzak, N.SAEL, S.MERZOUK, "Comparative Study of IoT Protocols" (May 28, 2018), Smart Application and Data Analysis for Smart Cities (SADASC '18), Available at SSRN: <https://ssrn.com/abstract=3186315>
- [2] OASIS Standard. MQTT version 3.1.1. [Online]. Available: <http://docs.oasis-open.org/mqtt/mqtt/v3.1.1/csprd02/mqtt-v3.1.1-csprd02.html>
- [3] IBM MessageSight [Online]. Available: https://www.ibm.com/support/knowledgecenter/ko/SSCGGQ_1.2.0/WelcomePage/ic-homepage.html