

# 지연 민감형 IoT 응용을 위한 GQS 기반 포그 Pub/Sub 시스템의 설계 및 평가

배 인 한<sup>†</sup>

## Design and Evaluation of a GQS-based Fog Pub/Sub System for Delay-Sensitive IoT Applications

Ihn-Han Bae<sup>†</sup>

### ABSTRACT

Pub/Sub (Publish/Subscribe) paradigm is a simple and easy to use model for interconnecting applications in a distributed environment. In general, subscribers register their interests in a topic or a pattern of events and then asynchronously receive events matching their interest, regardless of the events' publisher. In order to build a low latency lightweight pub/sub system for Internet of Things (IoT) services, we propose a GQSFPS (Group Quorum System-based Fog Pub/Sub) system that is a core component in the event-driven service oriented architecture framework for IoT services. The GQSFPS organizes multiple installed pub/sub brokers in the fog servers into a group quorum based P2P (peer-to-peer) topology for the efficient searching and the low latency accessing of events. Therefore, the events of IoT are cached on the basis of group quorum, and the delay-sensitive IoT applications of edge devices can effectively access the cached events from group quorum fog servers in low latency. The performance of the proposed GQSFPS is evaluated through an analytical model, and is compared to the GQPS (grid quorum-based pub/sub system).

**Key words:** Cloud of Things, Fog Computing, Group Quorum System, Internet of Things, Pub/Sub System

### 1. 서 론

발행/구독(Pub/Sub) 모델은 비동기 메시징 패러다임이다. Pub/Sub 모델에서 발신자의 메시지는 특별한 수신자가 정해져 있지 않다. 대신 발행된 메시지는 정해진 범주에 따라, 각 범주에 대한 구독을 신청한 수신자에게 전달된다. 수신자는 발행자에 대한 지식이 없어도 원하는 메시지만을 수신할 수 있다. 이러한 발행자와 구독자의 디커플링은 더 동적인 네트워크 토폴로지와 높은 확장성을 허용한다. Pub/Sub 기술은 메시지 기반으로 데이터 생성자의 위치,

시간, 동기화 등에 대한 제한 없이 데이터에 접근할 수 있게 하는 특징을 갖는다. Pub/Sub 모델에서 구독은 이론적으로 전체 발행된 메시지의 부분집합만을 받는다. 수신자 별로 메시지를 분류하는 프로세스와 처리를 필터링이라 부른다. 필터링에는 일반적으로 2가지 형태: 토픽 기반과 콘텐츠 기반이 있다. 그리고 Pub/Sub 시스템에서 발행자는 중개 브로커에게 메시지를 전달한다[1].

Pub/Sub 시스템은 요즘 인터넷 상에서 폭넓게 사용되고 있다. 그러한 시스템의 몇몇 예는 RSS (Really Simple Syndication) 피드와 같은 뉴스 신디

\* Corresponding Author : Ihn-Han Bae, Address: (18430) Hayang-ro 13-13, Hayang-eup, Gyeongsan-si, Gyeongbuk, Korea, TEL : +82-53-850-2742, FAX : +82-53-850-

2750, E-mail : ihbae@cu.ac.kr

Receipt date : Jul. 6, 2017, Approval date : Jul. 31, 2017

<sup>†</sup> School of IT Eng., Catholic University of Daegu

케이션, 다인용 게임, 트위터 또는 페이스북 북과 같은 사회 연결망, Spotify와 IPTV (Internet Protocol Television)와 같은 미디어 스트리밍 응용들이다. 그러한 시스템의 사용자들은 일일 뉴스, 친구 트위터, 음악 재생 목록, IPTV에 대한 채널이 될 수 있는 다수의 토픽을 구독하여 특정 데이터에서 그것들의 관심을 나타낸다. 만일 어떤 새로운 데이터가 어떤 토픽에 발행되었다면, 그 구독자들에게 그 콘텐츠가 공지되고 제공된다. 응용에 따라 이 서비스는 대역폭 집약적이거나 실시간적이다[2].

일반적인 IoT 서비스들은 복잡한 멀티소스 실시간 데이터 스트리밍과 이벤트 기반 서비스 동적 조정 교차 다중 서비스 도메인이나 구조를 포함하고, 확장성과 낮은 지연은 대형 분산 지연 민감형 IoT 서비스를 위해 중요하다. Pub/Sub 시스템은 대형 분산 느슨한 결합 IoT 환경에서 비동기 주문형 센서 정보 전파를 촉진시킬 수 있다. 그러나 일반적인 Pub/Sub 시스템들은 형편없는 확장성과 높은 지연의 문제를 가지고 있다[3].

본 논문에서는 대형 지연 민감형 사물인터넷 서비스에 적합한 확장 가능한 저 지연 경량 Pub/Sub 시스템 구축을 위하여 GQS 기반 포그 Pub/Sub 시스템인 GQSFPS를 제안한다. 제안하는 GQSFPS에서는 IoT 이벤트들의 효율적인 검색과 낮은 지연 액세스를 위하여 그룹 쿼럼 시스템에 기초하여 포그 컴퓨팅의 서버들에 설치된 다수의 Pub/Sub 브로커들로 응용 수준의 오버레이 네트워크를 구성하고, Pub/Sub 응용과 긴밀히 통합된다. 따라서 IoT의 이벤트들은 그룹 쿼럼을 기반으로 캐시 되고, 에지 컴퓨팅의 다양한 IoT 장치들과 응용들은 그룹 쿼럼 포그 서버로부터 캐시된 이벤트들을 낮은 지연으로 효율적으로 액세스할 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 제안되는 GQSFPS가 실행되는 시스템 환경인 CoT(Cloud of Things)와 포그 서버들로 오버레이 P2P 위상 구축에 사용된 그룹 쿼럼 시스템, 그리고 IoT 응용들을 위한 Pub/Sub 시스템에 대한 관련 연구들을 살펴본다. 3장에서는 대형 지연 민감형 IoT 응용들을 위한 확장 가능한 저 지연 메시지 Pub/Sub 시스템인 GQSFPS를 설계하고, 또한 포그 서버에 설치되는 GQSFPS 브로커의 구조를 설계한다. 그리고 4장에서는 제안하는 GQSFPS의 성능을 분석적 모델로 평

가하고, 그것을 성능을 격자 쿼럼 기반 Pub/Sub 시스템인 GQPS와 비교한다. 마지막으로 5장에서 본 논문의 결론과 향후연구 과제에 대하여 기술한다.

## 2. 관련 연구

2장에서는 제안하는 GQSFPS가 설치되고 실행되는 포그 컴퓨팅과 IoT, 그리고 클라우드 컴퓨팅으로 구성되는 전반적인 CoT의 구조와 GQSFPS가 사용하는 오버레이 P2P 위상을 구축하는데 필요한 그룹 쿼럼 시스템에 대하여 설명하고, 그리고 IoT 응용을 위한 메시지 Pub/Sub 시스템에 대한 관련 연구들을 살펴본다.

### 2.1 CoT

클라우드 플랫폼에서 IoT 응용의 빠른 발전으로 연결된 장치들의 개수는 매우 빠른 속도로 증가하고 있다. 모든 연결된 장치들은 다른 응용들을 위하여 클라우드 플랫폼을 통하여 연결될 것이다. IoT와 클라우드 컴퓨팅을 통합 운영하는 CoT라는 새로운 패러다임이 생성되었다. CoT에서, IoT 객체들은 센서들로부터 인터넷의 모든 프런트엔드 사물들에까지 확장된다. 클라우드 플랫폼과 IoT의 통합으로, CoT는 스마트 지능 응용을 지원하기 위하여 대규모 장치들을 연동하기 위한 강화된 기능이 필요하다. CoT는 다양한 산업과 연구 분야들에서 갈수록 더 중요한 역할을 할 것이다[4].

IoT에서 사물은 일반적으로 단말기를 의미하며 여기에는 단순 기계, 스마트 측정기기, 센서 뿐만 아니라 우리가 일상적으로 활용하고 있는 생필품까지 확대될 수 있다. 이와 같은 IoT 서비스 제공을 위해서는 IoT 접속 기술이 우선적으로 필요하다. IoT 접속 기술은 사람의 인위적인 간섭 없이 기기 간에 자율적으로 정보를 공유할 수 있는 기술을 의미 한다. 이와 같은 정보 공유는 단말기 간, 단말기와 서버 간, 나아가 단말기와 통신 네트워크 간 등 현재 다양한 환경에서 이루어지고 있다. 따라서 물리적 세계의 사이버 세계로 더 많은 직접 통합을 위한 기회가 생성되고 있다. IoT는 IoT 장치의 센서 및 액츄에이터와 통합하여 사이버-물리 시스템이 되고 있다. IoT에서 객체들은 지리적으로 또는 논리적으로 클러스터로 그룹화 될 수 있다. 다양한 IoT 클러스터들은 다양한

위치들로부터 방대한 데이터를 생성하고, 이 데이터를 더 효율적으로 처리하는 것이 필요하다. 이 데이터의 효율적인 처리는 대리 장치들과 클라우드 데이터 센터로 사이트 처리와 분산과 같은 다른 계산 모델들의 조합을 야기시킬 수 있다[5, 6].

클라우드 컴퓨팅(cloud computing)은 다른 컴퓨터들 또는 장치들에 처리기, 저장장치, 서비스, 그리고 응용 등과 같은 구성 가능한 자원들의 공유 풀에 대한 유비쿼터스 주문형 액세스를 제공하는 인터넷 기반 컴퓨팅 패러다임이다. 비록 클라우드 컴퓨팅 패러다임이 IoT 클러스터들로부터의 방대한 데이터를 처리할 수 있지만, 클라우드 컴퓨터와 주고받는 방대한 데이터의 전송은 제한된 대역폭에 어려움을 겪는다. 따라서 데이터 소스 근처에서 데이터를 처리가 필요하고, 포그 컴퓨팅은 이 문제에 대한 유망한 해결책을 제공한다. 포그 컴퓨팅은 최종 사용자 근처의 네트워크 에지에서 클라우드와 유사한 서비스를 제공하는 분산 패러다임이다. Fig. 1에서 보여주는 것처럼 포그 계층은 인터넷에 호스트 된 클라우드 데이터 센터들과 최종 사용자 장치들 사이에 위치한다. 포그 컴퓨팅은 에너지 소비, 지연, 인터넷 상의 데이터 트래픽 감소 등의 측면에서 더 나은 QoS를 제공할 수 있다[7, 8].

### 2.2 그룹 쿼럼 시스템

쿼럼 시스템은 모든 2개의 집합이 교집합을 갖는 집합들의 모임이다. 쿼럼 시스템은 상호배제, 데이터 복사, 정보 전파 등을 포함하는 분산 시스템 분야의 많은 응용들에서 사용되어지고 있다. 그룹 쿼럼 시스템(Group Quorum System, GQS) [9]는 Fig. 2와 같은 펼쳐진 표면 쿼럼 시스템으로부터 구축될 수

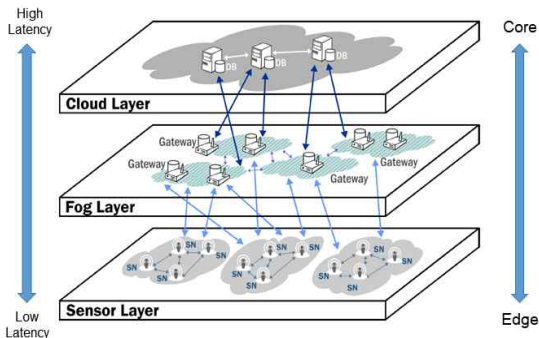


Fig. 1. Overall Architecture of a fog-based IoT platform.

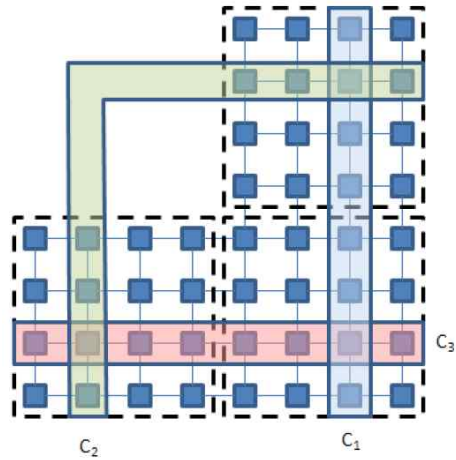


Fig. 2. The unfolded surficial quorum system for  $S_3$ .

있다. 평면상에 입방체의 세 측면을 펼치면 Fig. 2와 같이 3-그룹 쿼럼 시스템  $S_3=(C_1, C_2, C_3)$ 을 구축할 수 있다.  $C_1$ 내의 각 쿼럼은 정사각형의 우측 옆을 가로지르는 수직선에 해당한다.  $C_2$ 내의 각 쿼럼은 상단 정사각형을 가로지르는 수평선과 하단의 왼쪽 정사각형을 가로지르는 수직선에 해당한다. 마지막으로,  $C_3$ 내의 각 쿼럼은 하단의 두 정사각형을 가로지르는 수평선에 해당한다. 일반적으로,  $S_m$  내의 각  $C_i$ 는  $m-1$  정사각형이 필요하다. 그것들의 각각은 다른  $m-1$  연합(cartel)의 하나 이상의 정사각형을 공유하므로 두 연합의 대응하는 선들은 그 정사각형 위의 정확히 하나 이상의 노드에서 교차한다. 전체적으로, 그곳에는  $m(m-1)/2$  개의 정사각형이 있다.

$k$ 를 각 사각형의 너비라 하자. 그러면, 각 정사각형은  $k^2$  노드로 구성되므로  $m(m-1)/2$  정사각형 상의 전체 노드 개수는  $k^2 \frac{m(m-1)}{2}$ 이다. 시스템을 구성하는 모바일 노드들의 전체 개수를  $n$ 으로 나타내면,  $k^2 \frac{m(m-1)}{2}$ 을  $n$ 으로 놓을 수 있다. 따라서  $k = \sqrt{\frac{2n}{m(m-1)}}$ ,  $m > 1$ 이다. 쿼럼 크기는  $|Q| = (m-1)k = \sqrt{\frac{2n(m-1)}{m}}$ 이다.

### 2.3 Pub/Sub 시스템

Pub/Sub 시스템은 분산 환경에서 상호 연결되는 응용들을 위한 간단하고, 사용하기 쉽고 그리고 효율적으로 구현할 수 있는 패러다임이다. Pub/Sub 기반

미들웨어는 금융, 프로세스 자동화, 교통, 그리고 인수합병 등을 포함하는 다수의 응용 분야에서 어플리케이션 통합을 위하여 사용되고 있다. Pub/Sub 시스템은 시스템에 이벤트를 발행하는 정보 제공자와 그 시스템 내의 이벤트들의 특별한 범주를 구독하는 정보 소비자를 포함한다. 그 시스템은 모든 관심 있는 구독자들에게 발행된 이벤트의 즉시 전달을 보장한다. Pub/Sub 시스템은 일반적으로 발행자와 구독자 간의 메시지 라우팅을 전담하는 메시지 브로커를 포함한다.

Pub/Sub 시스템들은 토픽-기반 시스템 또는 콘텐츠 기반 시스템으로 분류할 수 있고, 시스템들은 그것들의 구조 역시 다르다. Pub/Sub 시스템은 푸시-기반, 풀-기반, 또는 혼합-기반일 수 있다. 푸시-기반에서, 메시지들은 구독자들에게 자동적으로 발송된다. 풀-기반 모델은 사용자 요구에 더 민감하게 반응할 수 있다. Pub/Sub 시스템은 구독자의 수와 이벤트들의 효율적인 분산을 위한 구독 관리와 어떤 이벤트의 효율적인 매칭 측면에서 확장성을 해결해야만 한다. 이벤트 시스템은 이벤트 서비스와 유일한 통신 패러다임을 실현하는 소프트웨어 컴포넌트로 구성되는 분산 통신 패러다임이다. Pub/Sub 시스템의 구조는 클라이언트-서버 또는 P2P로 분류된다 [10].

P2P 기반 Pub/Sub 기법들은 어떤 두 노드들 간의 경로의 길이가 일반적으로 길어야  $O(\log_2 n)$ 을 만드는 DHT (Distributed Hash Table) 오버레이 네트워크에 기반 한다[11]. 여기서  $n$ 은 네트워크 내의 노드의 개수이다.

IoT에서 다양한 종류의 데이터를 전달하기 위하여, W. Li 등 [12]은 역할 기반 분산 Pub/Sub 시스템 (RBDS, role-based distributed pub/sub system)을 제안하였다. RBDS는 격자의 계층적 라우팅 정책에 기반 한다. 구독자는 IoT 데이터의 구조에 의해 적절한 역할 브로커 노드로 사상된다. 그리고 브로커 노드들은 역할에 기초하여 다른 그룹들로 나뉜다. 각 그룹에서, 브로커 노드들은 그것들의 기능에 따라 슈퍼 노드와 보통 노드로 나뉜다. 슈퍼 노드들은 다른 그룹들에 이벤트를 분배하기 위하여 상위 격자 구조에 배치되고, 보통 노드들은 같은 그룹에 이벤트를 분배하기 위하여 하위 격자 구조에 배치된다. RBDS는 확장성을 향상시키고 지연시간을 감소시킨다.

대형 지연 민감형 IoT 서비스를 위한 확장 가능한 저 지연 Pub/Sub 시스템을 구축하기 위하여, Y. Sun 등 [3]은 응용 레벨 오버레이 네트워크를 구축하기 위하여 격자 쿼럼을 사용하고, 그리고 Pub/Sub 응용과 통합하는 GQPS를 제안하였다. GQPS에서, 각 노드는  $O(\sqrt{n})$  길이 이웃 리스트를 관리하므로 어떤 두 노드간의 경로는 2 홉을 초과하지 않는다.

### 3. GQSFPS의 설계

제안하는 GQSFPS 설계의 합리적인 배경은 IoT 시스템 자체의 수백만 IoT 객체들의 센서 정보를 분배하는 것이다. 단일 지점 고장을 피하기 위하여, 전체 서비스 광역 정보를 관리하는 단일 Pub/Sub 브로커는 없다. 제안하는 GQSFPS를 지원하는 CoT 기반 전체 시스템 구조는 Fig. 3과 같이 3 가지: 클라우드, 포그, 센서 계층들로 구성된다.

맨 위의 클라우드 계층은 인터넷 상에 전개된 클라우드 플랫폼들로 구성된다. 클라우드 서버들은 일반적으로 데이터 센터 배열을 가지고 있다. 지역 클라우드는 그 서버 상에서 실행되는 클라우드에서 이용 가능한 소프트웨어들로 구성되어지고, 그리고 원격 클라우드와 상호작용을 일반적으로 지원한다. 클라우드 컴퓨팅 계층에서는 동적 의사 결정을 지원하기 위한 대규모 이벤트 탐지, 장기 패턴 인식, 그리고 관계 모델링과 같이 복잡하고, 장기적인 분석, 그리고 도시 감시와 제어 등이 수행될 수 있다.

맨 아래 계층인 센서 계층에서, 스마트 장치들은 센서들과 같은 IoT 장치들이거나 스마트폰과 같은 사용자의 개인 휴대용 장치일 수 있다. 물리적 세계와 연결되어진 스마트 장치들은 제한된 자원들을 가지고 있고, 다른 장치들과 공유될 수 있거나 포그 서버 또는 클라우드 서버에 저장될 수 있는 많은 양의 데이터와 이벤트들을 생성한다. 제안하는 GQSFPS에서 데이터나 이벤트의 발행자와 구독자는 스마트 장치들이다. 그러한 노드들 간의 데이터 전송을 위하여 6LowPAN, IPv4/IPv6, RPL 등과 같은 일반적인 무선 센서 네트워크 프로토콜이 이 계층에서 사용될 수 있다.

포그 컴퓨팅은 클라우드에 의해 제공되는 서비스들을 그 네트워크의 에지로 확장하는 분산 컴퓨팅 패러다임이다. 포그 계층은 최종 사용자에게 가까운 네

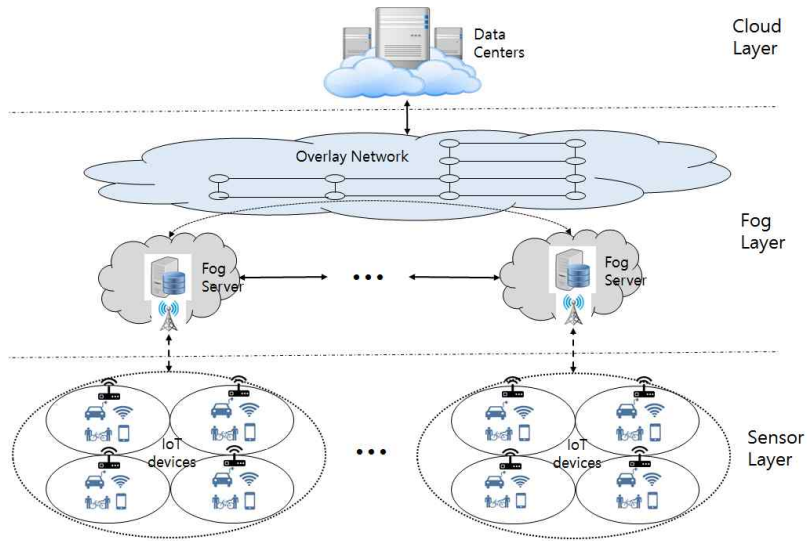


Fig. 3. Overall system architecture for supporting GQSFPS.

트위크 에지에 데이터와 계산이 호스트 되어 진다. 낮은 지연시간, 높은 대역폭, 지리적 분산으로 최종 사용자들에게 응용들과 서비스들의 새로운 유형을 제공한다. 또한 요구되는 데이터의 사본들과 레코드들을 포함하는 포그 서버들은 가용 데이터에 대한 부하 분산과 트랜잭션 고장 감소에 도움을 줄 수 있다. 또한 포그 컴퓨팅 플랫폼은 빅 데이터 분석을 위하여 사용될 수 있다. 빅 데이터 분석은 그 네트워크의 에지에서 수행한 뒤에 그 결과를 코어 네트워크로 전송할 수 있어 대역폭 낭비를 줄이고 네트워크 지연을 향상시킬 것이다. 결과적으로 포그 컴퓨팅 플랫폼은 구독자들의 네트워크 지연과 대역폭 낭비를 줄여 에지 응답성을 증가시킨다.

포그 계층이 갖는 특징은 다음과 같다[7].

- 로컬 네트워크 상태 인식: 포그 컴퓨팅은 에지에 배치되고, 실시간 무선 네트워크와 채널 정보를 액세스할 수 있다.
- 로컬 사용자 상황 인식: 응용들은 국부적으로 위치와 사용자 상황을 이용할 수 있다.
- 분산: 포그 컴퓨팅 자원들, 응용들, 서비스들은 다른 지리적 위치들에 분산될 수 있다.
- 이질성: 포그 컴퓨팅 노드들은 여러 가지 처리 및 저장 자원들뿐만 아니라 여러 가지 네트워크 연결성과 대역폭을 갖는 이질적이다.
- 이동성과 신뢰할 수 없는 액세스: 이동 장치들은 신뢰할 수 없는 무선 액세스를 통하여 포그 서버

스들을 액세스하고, 그 네트워크에 대한 그것들의 접속점이 자주 변한다.

- 극단적으로 낮은 지연: 포그 컴퓨팅은 로봇, 가상 및 증강 현실, 실시간 대화형 산업 제어 시스템과 같은 촉각 인터넷 응용들을 지원하기 위하여 극단적으로 낮은 지연시간을 잠재적으로 이룰 수 있다.
- 중앙 클라우드와 상호작용: 포그 컴퓨팅은 전통적인 중앙 클라우드들을 보완하지만 대체하지 않는다.

이러한 속성들에 기초하여, 포그 컴퓨팅은 매우 낮고 예측 가능한 지연을 요구하는 응용, 지역 분산 응용, 빠른 모바일 응용, 그리고 대형 분산 제어 시스템 등을 포함한다. 또한 포그 서버는 하부의 센서 계층의 센서들로부터 수신한 데이터 스트림에 대한 잠재적인 패턴을 식별하고, 그리고 추가 분석을 위하여 상위 계층의 계산에 필요한 특징 추출을 수행한다. 따라서 포그 서버는 원시 센서 데이터를 클라우드 계층으로 전송하는 대신에 지형공간정보를 포함하는 특징 결과만을 전송한다[13, 14].

본 논문에서는 지연 민감형 IoT 응용으로 센서 데이터를 갖는 이벤트의 전송 지연 시간을 줄이기 위하여, 제안하는 GQSFPS는 포그 서버들에 설치되어지고, 그리고 GQSFPS의 운영을 위하여 포그 계층의 상단에 포그 서버들로 연결된 물리적 네트워크 위에 GQS를 사용하여 가상 오버레이 네트워크를 설립한다. 따라서 많은 IoT 장치들이 짧은 지연시간에

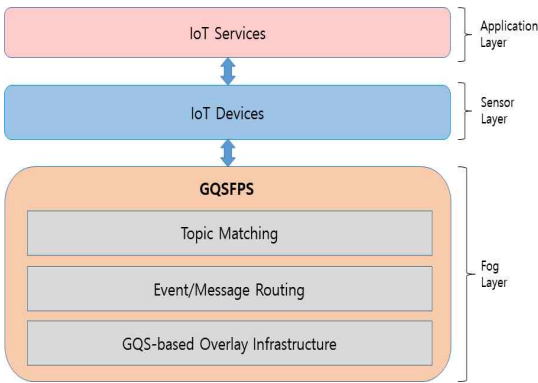


Fig. 4. Internal structure of GQSFPS system.

발행된 이벤트를 구독할 수 있다.

GQSFPS의 내부 구조는 Fig. 4와 같이 3 가지 논리적 계층: GQS 기반 오버레이 인프라, 이벤트/메시지 라우팅, 토픽 매칭으로 구성된다. IoT 장치들은 Pub/Sub 브로커인 GQSFPS에 센서 데이터를 갖는 이벤트와 토픽을 발행한다. GQSFPS는 발행된 이벤트에 대해 토픽 매칭을 실행하여 로컬 구독자들에게 그 이벤트를 포함하는 공지 메시지를 전송한다. 그리고 GQS에 기반 한 쿼럼 내의 포그 서버들에게 그 이벤트와 토픽을 전송한다. 전송한 이벤트가 Pub/Sub 브로커인 다른 쿼럼 서버들에 도착하면, 각 Pub/Sub 브로커의 GQSFPS는 수신한 이벤트를 포그 서버에 저장하고, 그리고 토픽 매칭을 실행하여 그 포그 컴퓨팅의 로컬 지역에 있는 구독하는 지연 민감형 IoT 서비스들에게 공지 메시지로 그 이벤트를 전송한다.

GQS에서 각 정사각형의 너비를  $k$ 라 두면, 각 정사

각형은  $k^2$  노드들로 구성되고, 3개의 정사각형의 전체 노드의 개수는  $3k^2$ 이다. CoT를 구성하는 포그 계층의 전체 Pub/Sub 브로커의 개수를  $n_b$ 로 나타내면,  $3k^2$ 을  $n_b$ 로 놓을 수 있다. 따라서  $k = \sqrt{n_b/3}$  이고, 그리고 쿼럼의 크기는  $q = 2k = 2\sqrt{n_b/3}$  이다. Fig. 5에서 발행자가 이벤트를 전송하는 GQS내의 원격 Pub/Sub 브로커들의 개수는  $\lfloor 4\sqrt{n_b/3} - 1 \rfloor$  이고, Fig. 6에서 구독자가 이벤트를 요청하는 GQS내의 원격 Pub/Sub 브로커들의 개수는  $\lfloor 2\sqrt{n_b/3} \rfloor$  이다. Fig. 5와 Fig. 6는 GQSFPS의 이벤트 발행 과정과 최악 경우 이벤트 구독 과정에 대한 메시지 시퀀스 다이어그램을 각각 보여준다.

GQS의 개념에 기초하여, Pub/Sub 브로커들은 브로커-대-브로커 GQS로 구성된다. GQS에서, Fig. 2에서 보여진 것처럼 표면 쿼럼 시스템의 계단 구조로 3-GQS,  $S_3 = (C_1, C_2, C_3)$ 를 구축할 수 있다. Fig. 7은 12개의 브로커들과 3개의 정사각형으로 구성된 GQS 기반 오버레이 인프라를 보여준다. 여기서 각 정사각형은 4개의 브로커들로 구성되고 그 정사각형의 차원은 2이다. GQS의 모든 정사각형에서, 브로커 노드들은 정사각형 원소에 행 단위로 배치된다.

Fig. 7의 GQS 기반 오버레이에서, 각 브로커 노드는 크기  $O(4\sqrt{n_b/3} - 1)$ 의 브로커 노드들의 집합만을 관리한다. 여기서  $n$ 은 Pub/Sub 시스템내의 전체 브로커 노드의 개수를 나타낸다. Fig. 7에서, 브로커 노드 5의 쿼럼,  $Q_5$ 는  $C_{5,1}$ 과  $C_{5,2}$ 로 구성된다. 여기서  $C_{5,1} = \{1, 3, 5, 7\}$ 이고,  $C_{5,2} = \{5, 6, 9, 11\}$ 이다. 브로커 노드 5는  $Send(d, 5, p), \forall p \in C_{5,1} \cup C_{5,2}$ 를 호출하여  $Q_5$

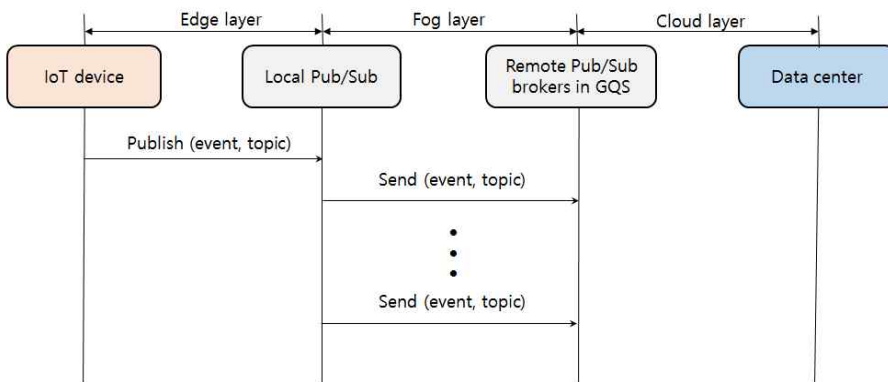


Fig. 5. Message sequence diagram for event publishers in GQSFPS.

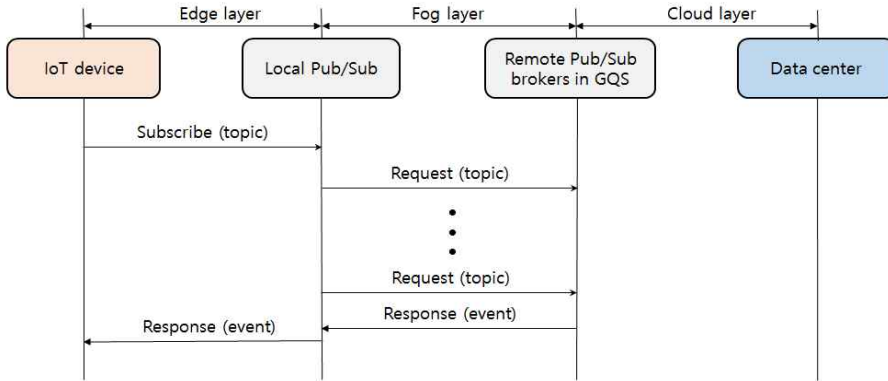


Fig. 6. Message sequence diagram for event subscribers in GQSFPS.

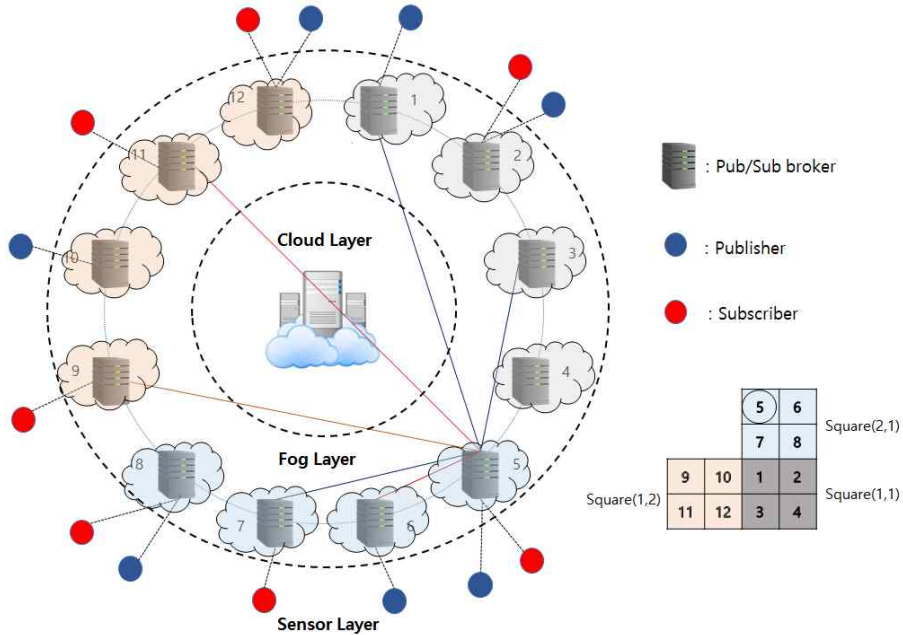


Fig. 7. Example of GQSFPS system having 12 Pub/Sub brokers.

내의 모든 브로커 노드들에게 IoT 장치의 센서 데이터(d)를 갖는 이벤트와 토픽을 전송한다. 브로커 노드 2가 브로커 노드 5의 센서 데이터 수신을 원한다면, 쿼럼  $C_{2,1}=\{2, 4, 6, 8\}$  또는  $C_{2,3}=\{1, 2, 9, 10\}$ 에 토픽을 갖는 *Request* 메시지를 전송한다. 쿼럼  $Q_5$ 와  $C_{2,1}$  또는  $C_{2,3}$ 는 비공백 교집합 {6}과 {9}를 각각 갖기 때문에 발행자인 브로커 노드 5의 센서 데이터는 구독자인 브로커 노드 2에게 전달된다. 따라서 어떤 두 노드간의 경로는 2 홉을 초과하지 않는다.

### 5. GQSFPS의 평가

5장에서는 제안하는 GQSFPS의 성능을 분석적 모델로 평가한다. 여기서는 쿼럼에 기반한 유사한 방식의 Pub/Sub 시스템들: 제안하는 GQSFPS 방법, 그리고 격자 쿼럼을 사용하는 GQPS [3]을 분산 클라우드 환경에 설치한 방법의 성능을 분석하고 평가한다. 우리는 구독/발행률에 따른 전체 Pub/Sub 관리 비용, Pub/Sub 브로커 개수에 따른 평균 이벤트 구독 지연과 구독자 트래픽 부하 등으로 성능을 평가한다. 여기서 토픽에 따른 이벤트 구독은 성능 평가에

Table 1. Parameters and values for analytical performance evaluation

Parameter	Value
Publishing cost to local cloud broker, $C_{plc}$	3
Publishing cost to local fog broker, $C_{plf}$	1
Publishing cost of inter-brokers, $C_{pibs}$	2
Subscribing cost to local cloud broker, $C_{slc}$	2.5
Subscribing cost to local fog broker, $C_{slf}$	1
Subscribing cost of inter-brokers, $C_{sibs}$	1.5

서 고려하지 않았다.

Table 1은 Pub/Sub 시스템의 성능을 분석적으로 평가하는데 사용되는 매개변수와 값을 보여준다. Pub/Sub의 동적 프로세스 비용은 이벤트 발행 비용과 이벤트 구독 비용을 합한 전체 Pub/Sub 관리 비용으로 평가된다. 따라서 제안하는 GQSFPS의 전체 Pub/Sub 관리 비용은 식 (1)과 같다.

$$C_{total} = (C_{plf} + |Q_p| C_{pibs}) + \frac{\lambda}{\mu} \left[ C_{slf} + \left( 1 - \frac{\lfloor 4\sqrt{n_b/3} - 1 \rfloor}{n} \right) |Q_s| C_{sibs} \right] \quad (1)$$

여기서  $|Q_p|$ 와  $|Q_s|$ 는 GQS에서 어떤 Pub/Sub 브로커가 이벤트를 전송하는 쿼럼의 크기와 이벤트를 요청하는 쿼럼의 크기를 각각 나타내고,  $\lambda/\mu$ 는 구독-발행률을 나타낸다.

Fig. 8은 Pub/Sub 브로커의 개수  $n_b = 32$ 일 때, 제안하는 GQSFPS 방법과 GQPS 방법의 전체 Pub/Sub 관리 비용을 보여준다. GQSFPS 방법의 성능이 구독/발행률에 상관없이 GQPS 방법에 비해 항상 우수한 성능을 보였다. 제안하는 GQSFPS는 Pub/

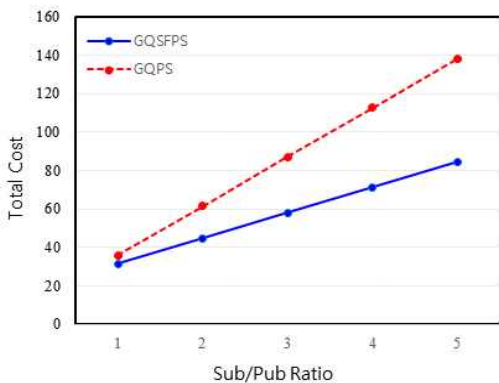


Fig. 8. Total pub/sub management cost over pub/sub ratio ( $n_b = 32$ ).

Sub 시스템을 IoT 장치들이 있는 에지 근처의 포그 계층에 배치하고, 그리고 발행 쿼럼과 구독 쿼럼의 크기가 다른 GQS로 오버레이 네트워크를 구축하여 제안하는 시스템 구조와 지연 민감형 IoT 응용에 맞게 사용하기 때문에 우수한 성능을 보인다.

제안하는 GQSFPS의 평균 이벤트 구독 홉 지연은 식 (2)와 같다..

$$T_{sub\_delay} = 2 \times (\lfloor 4\sqrt{n_b/3} - 1 \rfloor) + 4 \times [n - (\lfloor 4\sqrt{n_b/3} - 1 \rfloor)] = 4n - \lfloor 8\sqrt{n_b/3} \rfloor + 2 \quad (2)$$

Fig. 9는 Pub/Sub 브로커 개수에 따른 평균 이벤트 구독에 대한 홉 지연을 보여준다. GQPS에서는 Pub/Sub 브로커들을 제안하는 CoT 구조에서 코어에 가까운 클라우드에 설치한다. 그러나 제안하는 GQSFPS에서는 Pub/Sub 브로커들을 에지에서 가까운 포그 계층에 설치할 뿐만 아니라 지연 민감형 IoT 응용을 위하여 구독 쿼럼의 크기보다 큰 발행 쿼럼을 사용하여 많은 Pub/Sub 브로커들에 이벤트를 발행하기 때문에 로컬 Pub/Sub 브로커가 요청한 이벤트를 캐시할 확률이 높아진다. 따라서 제안하는 GQSFPS의 평균 이벤트 구독 지연 홉 수가 작아 빠른 시간에 발행된 이벤트를 구독할 수 있기 때문에 지연 민감형 IoT 서비스에 적합한 Pub/Sub 시스템이라는 것을 확인할 수 있었다.

Fig. 10은 Pub/Sub 브로커 개수에 따른 어떤 이벤트 구독자에 의해 최악의 경우에 발생하는 이벤트 요청 메시지의 트래픽 부하를 보여준다. 제안하는 GQSFPS에서는 GQPS의 쿼럼 크기보다 작고, 그것의 발행 쿼럼의 크기보다 작은 구독 쿼럼을 사용기

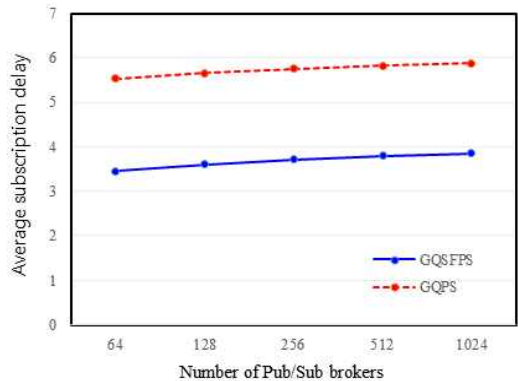


Fig. 9. Average event subscription delay in hops over number of Pub/Sub brokers.



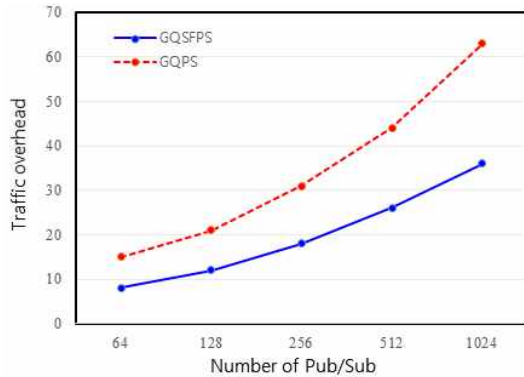


Fig. 10. Traffic overhead of an event subscriber over number of Pub/Sub brokers.

때문에 이벤트 요청 메시지에 의한 트래픽 부하가 작기 때문에 CoT의 전체 망 트래픽 부하가 줄어든다. 따라서 이러한 망 트래픽 부하를 고려하면 평균 이벤트 발행자 지연 시간과 구독자 지연 시간 또한 줄어들 것이다.

## 6. 결 론

본 논문에서는 IoT 장치들에 의해 발생하는 데이터를 갖는 이벤트들을 효율적으로 Pub/Sub하는 GQSFPS를 제안하고, 그것을 분석적 모델로 평가하고, 그리고 격자 쿼럼을 사용하는 GQPS [3]을 분산 클라우드 환경에 설치한 방법과 성능을 비교하였다. 제안하는 GQSFPS에서는 지연 민감형 IoT 응용들을 위하여 에지 계층 근처에 포그 계층의 서버들을 Pub/Sub 브로커로 사용하고, 그리고 포그 계층의 물리적 네트워크 위에 GQS를 사용하여 가상 오버레이 네트워크를 구성한다. GQS의 발행 쿼럼을 사용하여 포그 서버들에 센서 데이터 갖는 이벤트들을 토픽과 함께 전송하여 발행하고, 그리고 IoT 응용들이 GQS에 이벤트 구독 요청을 전송하면, GQS는 구독 쿼럼을 사용하여 포그 서버들에 캐시된 이벤트들을 구독한다. 성능 평가 결과, 제안하는 GQSFPS의 성능이 우수함을 확인하였다. 향후 연구과제로는 모의실험을 통하여 다양한 동적인 변화에 따른 GQSFPS의 성능을 평가하는 것과 CoT 응용을 위하여 Pub/Sub 시스템에 인공지능 기술을 접목한 지능형 상황인식 Pub/Sub 시스템을 연구하는 것 등이 있다.

## REFERENCE

- [1] Wikipedia, Publish and Subscribe Model, <http://wiki.c2.com/?PublishSubscribeModel>, (accessed June, 26, 2017).
- [2] F. Rahimian, S. Girdzijauskas, A.H. Payberth, and S. Haridi, "Vitis: A Gossip-Based Hybrid Overlay for Internet-Scale Publish/Subscribe Enabling Rendezvous Routing in Unstructured Overlay Networks," *Proceeding of International Parallel and Distributed Processing Symposium*, pp. 746-757, 2011.
- [3] Y. Sun, X. Qiao, B. Cheng, and J. Chen, "A Low-Delay, Lightweight Publish/Subscribe Architecture for Delay-Sensitive IOT Services," *Proceeding of International Conference on Web Services*, pp. 179-186, 2013.
- [4] J. Zhou, T. Leppänen, E. Harjula, C. Yu, H. Jin, and L.T. Yang, "CloudThings: a Common Architecture for Integrating the Internet of Things with Cloud Computing," *Proceeding of the International Conference on Computer Supported Cooperative Work in Design*, pp. 651-657, 2013.
- [5] J.H. Kim, "IoT Access Technology and Its Implications," *Journal of Korea Information Society Development Institute, Broadcasting Policy Information and Communication*, Vol. 28, No. 3, pp. 1-17, 2016.
- [6] A. Munir, P. Kansaker, and S.U. Khan, "IFCIoT: Integrated Fog Cloud IoT Architectural Paradigm for Future Internet of Things," *Journal of IEEE Consumer Electronics Magazine*, Vol. 6, No. 3, pp. 74-82, 2017.
- [7] H. Liu, F. Eldarrat, H. Alqahtani, A. Reznik, X. Foy, and Y. Zhang, "Mobile Edge Cloud System: Architectures, Challenges, and Approaches," *Journal of IEEE System*, Issue 99, pp. 1-14, 2017.
- [8] K.P. Saharan and A. Kumar, "Fog in Comparison to Cloud: A Survey," *International Journal of Computer Applications*, Vol. 122, No. 3, pp. 10-12, 2015.

[9] Y. Joung, "Quorum-Based Algorithms for Group Mutual Exclusion," *Journal of IEEE Transactions on Parallel and Distributed Systems*, Vol. 14, No. 6, pp. 463-476, 2003.

[10] Indiana University, *Survey of Publish Subscribe Event System*, Technical Report TR 574, School of Informatics and Computing, IN 47405-7104, 2003.

[11] A. Medvedev, A. Zaslavsky, S. Khoruzhnikov, and V. Grudinin, "Reporting Road Problems in Smart Cities Using OpenIoT Framework," *Proceeding of International Workshop Interoperability and Open-Source Solutions for the Internet of Things*, pp. 169-182, 2015.

[12] W. Li, H. Huang, and L. Zhang, "A Role-Based Distributed Publish/Subscribe System in IoT," *Proceeding of International Conference on Computer Science and Network Technology*, pp. 128-133, 2015.

[13] H.R. Arkian, A. Diyanat, and A. Pourkhalili, "MIST: Fog-Based Data Analytics Scheme with Cost-Efficient Resource Provisioning for IoT Crowdsensing Applications," *Journal of Network and Computer Applications*, Vol. 82, No. C, pp. 152-165, 2017.

[14] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, and Q. Yang, "A Hierarchical Distributed fog Computing Architecture for Big Data Analysis in Smart Cities," *Proceedings of the Academy Science Engineering Big Data and Social Informatics*, pp. 1-6, 2015.

[15] I.H. Bae, "Design and Evaluation of a Hierarchical Hybrid Content Delivery Scheme Using Bloom Filter in Vehicular Cloud Environments," *Journal of Multimedia Society*, Vol. 19, No. 8, pp. 1597-1608, 2016.



배 인 한

1990년 중앙대학교 컴퓨터공학과  
공학박사

1996년~1997년 Department of  
Computer Science and  
Eng., The Ohio State  
University 박사후과정

2002년~2003년 Department of Computer Science, Old  
Dominion University 방문교수

2009년~2010년 Department of Computer Science, Old  
Dominion University 방문교수

1989년~현재 대구가톨릭대학교 IT공학부 교수

관심분야 : 사물인터넷, 차량망, 차량클라우드, 지능스마트  
트랩 등