

사물인터넷에서 분산 발행/구독 구조를 위한 하이퍼큐브 격자 쿼럼의 설계 및 응용

배 인 한[†]

Design and Its Applications of a Hypercube Grid Quorum for Distributed Pub/Sub Architectures in IoTs

Ihnhan Bae[†]

ABSTRACT

Internet of Things(IoT) has become a key available technology for efficiently implementing device to device(D2D) services in various domains such as smart home, healthcare, smart city, agriculture, energy, logistics, and transportation. A lightweight publish/subscribe(Pub/Sub) messaging protocol not only establishes data dissemination pattern but also supports connectivity between IoT devices and their applications. Also, a Pub/Sub broker is deployed to facilitate data exchange among IoT devices. A scalable edge-based publish/subscribe (Pub/Sub) broker overlay networks support latency-sensitive IoT applications. In this paper, we design a hypercube grid quorum(HGQ) for distributed Pub/Sub systems based IoT applications. In designing HGQ, the network of hypercube structures suitable for the publish/subscribe model is built in the edge layer, and the proposed HGQ is designed by embedding a mesh overlay network in the hypercube. As their applications, we propose an HGQ-based mechanism for dissemination of the data of sensors or the message/event of IoT devices in IoT environments. The performance of HGQ is evaluated by analytical models. As the results, the latency and load balancing of applications based on the distributed Pub/Sub system using HGQ are improved.

Key words: Data Dissemination, Edge Computing, Hypercube, Internet of Things, Pub/Sub System, Quorum System

1. 서 론

사물인터넷(IoT)은 일반적인 컴퓨터 기반 인터넷 모델을 연결된 사물들의 지리적으로 분산되고 이질적이며 제한된 모델로 확장한 것이다. IoT에서 사물들은 컴퓨터, 센서, 구동장치 그리고 프로세스일 수 있다. IoT 응용은 인터넷에 연결된 다양한 IoT 장치들로부터 수신된 데이터를 활용하고 결합하여 사람에게 가치를 제공한다. 광범위한 IoT로 인해 연결 기술, 상호 운용성, 관리 및 보안 프로토콜에 대한

문제가 발생한다. 그러한 문제들은 배터리를 교체하지 않고 수년 동안 작동해야 하고, 손실 및 간헐적인 네트워크 조건에서 빈번한 데이터 전송을 수행해야 하는 저전력 제약 장치의 존재로 인해 더욱 복잡해졌다[1].

발행/구독(Pub/Sub, PS) 시스템은 대형 분산 시스템 구축을 위한 강력한 통신 패러다임으로 사용되고 있다. 이 패러다임에서 통신 개체들은 시간, 공간, 동기화 측면에서 느슨하게 결합된다. 이 패러다임은 분산 시스템에서 매우 중요한 익명성, 다대다, 비동

※ Corresponding Author : Ihnhan Bae, Address: (38430) Hayang-Ro 13-13, Hayang-Eup, Gyeongsan-si, Gyeongbuk, Republic of Korea, TEL : +82-53-850-2742, FAX : +82-53-359-6640, E-mail : ihbae@cu.ac.kr

Receipt date : Jul. 20, 2022,
Approval date : Aug. 22, 2022

[†] School of Computer Software, Daegu Catholic University

기성과 같은 유용하고 유연한 특성을 지원할 수 있다 [2]. 분산 PS 시스템에서, PS 브로커들은 응용 단계 오버레이 망으로 상호연결되고, 데이터 소스로부터 싱크까지 발행 메시지를 효율적으로 전송한다. PS 시스템의 오버레이는 시스템 성능과 메시지 라우팅 비용에 직접적인 영향을 미친다. 고급 브로커 오버레이를 구축하는 것은 관심을 받고 있는 분산 PS 시스템을 위한 근본적인 핵심 문제이다[3].

본 논문에서는 분산 PS 시스템에서 IoT 응용을 위한 하이퍼큐브 격자 퀴럼(Hypercube Grid Quorum, HGQ)을 설계한다. HGQ의 설계에서 PS 모델에 적합한 엣지 계층의 하이퍼큐브 망에 메시 구조의 오버레이 망을 임베딩하여 HGQ가 설계된다. 그리고 IoT 환경에서 제안하는 HGQ의 응용으로 엣지 오버레이 망에 구축된 HGQ 플랫폼에 기반한 센서 데이터 전달 메커니즘을 제안한다.

본 부분의 구성은 다음과 같다. 2장에서는 관련 연구로 결합 포용성 및 고가용성 분산 컴퓨팅의 일관성을 보장하기 위한 퀴럼 시스템, HGQ 구축을 위한 하이퍼큐브 구조와 엣지 컴퓨팅, 그리고 메시지를 비동기적으로 멀티캐스트할 수 있는 서버리스 마이크로 통신 구조인 PS 메시징 시스템에 대하여 살펴보고, 3장에서는 분산 PS 시스템을 위한 HGQ의 설계와 제안한 HGQ의 응용 예를 기술하고, 그리고 4장에서는 제안한 HGQ의 성능을 평가한다. 마지막으로 5장에서는 본 연구의 결론과 향후 연구 과제에 대하여 기술한다.

2. 관련 연구

2장에서는 하이퍼큐브 퀴럼 구축에 필요한 퀴럼 시스템, 엣지 컴퓨팅, 하이퍼큐브 구조, 그리고 PS 시스템에 대하여 살펴본다.

2.1 퀴럼 시스템

퀴럼 시스템은 분산 시스템에서 처리기들 간의 조정을 달성하기 위한 균일하고 신뢰적인 방법을 제공하는 기본 도구로 역할을 한다. 퀴럼들의 각 쌍은 비공백 교집합을 가지는 속성을 갖는 퀴럼이라 부르는 노드들의 부분집합들의 모임으로 정의된다.

퀴럼 시스템은 분산 시스템에서 프로세서들 간의 조정을 퀴럼 시스템은 일관된 데이터 복제, 합의 문

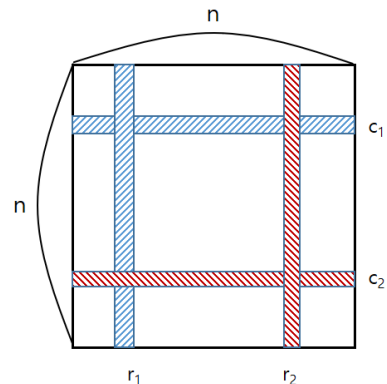


Fig. 1. Basic grid quorum system.

제 해결 또는 정보 보급을 위한 분산 상호배제를 달성하는데 널리 사용된다. 최근에 랭데부 문제나 단말 또는 노드가 네트워크에 존재하지만 동일한 네트워크의 다른 노드와 통신할 수 없는 숨겨진 인컴베트 문제(Hidden Incumbent Problem, HIP)를 해결하기 위해 퀴럼 시스템을 채택하고 있다[4, 5].

격자 퀴럼 방법은 N 노드들을 N_1 행과 N_2 열의 행렬로 배열한다. 여기서 $N_1 \times N_2 = N$ 이고 퀴럼들은 행과 열로 구성된다. $\sqrt{n} \in N$ 이라 가정하면, 격자에서 \sqrt{n} 의 측면 길이를 갖는 정방 행렬에 n 노드들을 배열한다. Fig. 1의 기본 격자 퀴럼 시스템에서, $1 \leq i \leq \sqrt{n}$ 을 갖는 각 퀴럼 Q_i 는 행 i 와 열 j 를 사용한다. 각 퀴럼의 크기는 $2\sqrt{n}-1$ 이고 2개의 퀴럼들은 정확히 2개의 노드들에서 겹친다.

2.2 엣지 컴퓨팅

엣지 컴퓨팅은 다양한 종단 기기에서 발생하는 데이터를 중앙 집중식 데이터 센터와 같은 클라우드로 보내지 않고, 데이터가 발생한 기기 또는 근거리에서 있는 서버에서 실시간으로 처리하여 데이터 처리 시간을 큰 폭으로 단축하고, 인터넷 대역폭 사용량 또한 감소시키는 새로운 산업용 데이터 처리 패러다임이다. 엣지 컴퓨팅 플랫폼은 Fig. 2와 같이 IoT 관리, 데이터 저장, 콘텐츠 캐시, 컴퓨팅 오프로드 및 서비스 전달을 제공할 수 있게 된다[6].

2.3 하이퍼큐브 구조

하이퍼큐브는 컴퓨터 네트워크를 위한 기본적인 모델로 사용되어진다. k -차원 하이퍼큐브는 2^k 노드

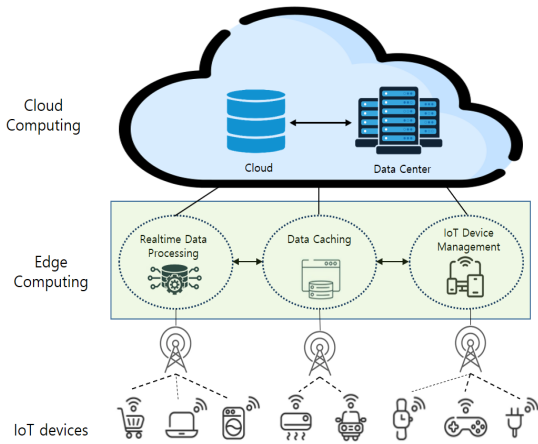


Fig. 2. Edge computing platform.

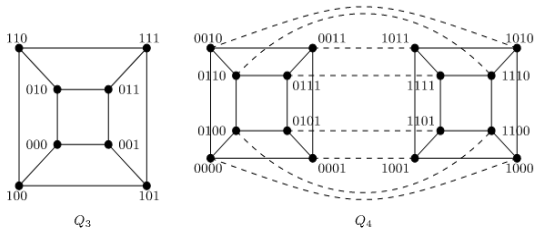


Fig. 3. n-dimensional hypercube structures for $n \in \{3, 4\}$.

들과 $k2^{k-1}$ 간선들을 갖는 무향 그래프 Q_k 이다. Q_k 에서 각 노드는 k -비트 이진 문자열 $(a_{k-1}a_{k-2}...a_1a_0)$ 으로 표현될 수 있다. Q_k 내의 별개의 2개의 노드들 x 와 y 에 대해, 만일 그 노드들의 이진 문자열 표현이 단지한 비트 위치에서만 다르다면 x 는 y 와 인접하다. Fig. 3은 $k \in \{3, 4\}$ 에 대한 Q_k 를 보여준다. 하이퍼큐브에서 두 노드간의 경로의 길이는 간단하게 경로의 이음선의 개수이다. Q_k 에서 어떤 두 노드간의 기껏해야 k 길이의 경로가 존재한다. 2개의 노드 k_1 과 k_2 가 i 비트들만 다르면, 그것들의 해밍 거리 $H(k_1, k_2) = i$ 이고, 경로의 길이는 i 가 될 것이다. 그 하이퍼큐브에는 노드 k_1 과 k_2 사이에 i 보다 더 짧은 경로는 없다[7,8].

2.4 발행/구독 시스템

IoT의 전망은 수십억 개의 장치들을 연결하는 것이다. 그러한 장치들은 일반적으로 네트워크의 엣지에서 작동한다. 따라서 그것들은 배터리로 작동되거나 광역 네트워크에 대한 연결이 느리고 불안정할 수 있다. 장치를 직접 연결하기 보다 IoT에서 통신은

PS 기반 브로커를 통한 비동기적으로 수행된다. 일반적인 PS 시스템은 통신망 상에 분산된 노드들의 집합으로 구성된다. 그 시스템의 클라이언트들은 그것들의 역할에 따라 정보의 생산자 역할을 하는 발행자와 정보의 소비자 역할을 하는 구독자로 나누어진다. 클라이언트들은 그것들 간의 직접 통신할 필요는 없지만 상호작용이 PS 시스템의 노드들을 통하여 일어나므로 분리되어 있다. 이러한 분리는 통신 시스템에 바람직한 특성이다. 그 이유는 응용 프로그램이 통신 문제로부터 더 독립적으로 만들어질 수 있고, 동기화 또는 발행자로부터 구독자의 직접 주소 지정과 같은 측면을 처리하는 것을 피할 수 있기 때문이다.

운영상, 클라이언트 노드들과 PS 시스템 간의 상호작용은 PS 시스템 상의 클라이언트들에 의해 실행될 수 있는 기본 연산들의 집합을 통해 일어난다. Fig. 4는 PS 시스템의 고급 상호작용 모델을 보여준다. 여기서 p 와 s 는 발행자와 구독자를 각각 가리킨다. PS 브로커는 발행 메시지를 수집하고 구독자에게 전송한다. 발행자는 한 가지 정보 e 를 $publish(e)$ 연산을 실행하여 PS 시스템에 제출한다. 구독자는 $subscribe(s)$ 와 $unsubscribe(s)$ 연산을 각각 실행하여 PS 시스템으로부터 구독 s 를 생성하고 제거한다. 공지 e 는 해당 속성에 대해 선언된 모든 제약 조건을 충족하는 경우 구독 s 와 일치한다고 한다[11,12].

PS 시스템은 주로 2가지 방법: 토픽-기반과 콘텐츠-기반으로 설계된다. 보통 이벤트는 속성-값 쌍들의 집합으로 구조화된다. 토픽-기반 PS에서, 구독은 통신 채널의 세션을 만드는데 사용되는 토픽을 표현한다. 토픽은 토픽 이름으로 지정되며, 구독자들은 특정 이벤트에 대한 관심 토픽 이름에 대한 구독을 선언하고, 그리고 발행자들은 토픽 이름이 포함된 메

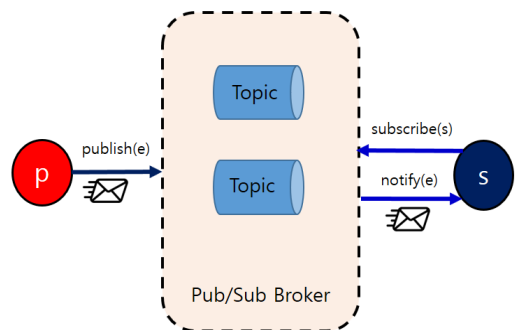


Fig. 4. High-level interaction model of a PS system.

시지를 태그한다. 콘텐츠-기반 PS에서, 구독 s는 구독 언어에 따라 달라지는 일련의 제약 조건을 통해 표현되는 이벤트 콘텐츠의 일부에 대한 필터이다. 따라서 콘텐츠 기반 PS에서 이벤트들은 다소의 미리 정의된 기준에 따라 분류되지 않고 이벤트 자체의 속성에 따라 분류된다.

IoT 응용을 위한 하이퍼큐브 또는 큐립 기반 PS 시스템에 대한 기존 연구들을 살펴보면 다음과 같다. Araujo 등[13]은 회원정보와 발행된 메시지들은 발행자에 루트를 둔 동적으로 구축된 스페닝 트리 상의 토픽 그룹의 구독자에게 방송하기 위하여 가상 하이퍼큐브 유사 위상의 상단에 구축되는 토픽 기반 PS 시스템인 VCube-PS를 제안하였다. VCube에서 노드 i는 다른 N-1 노드들을 d-VCube를 형성하는 $d = \log_2 N$ 클러스터들로 그룹하고, 각 클러스터 $s(s=1, \dots, d)$ 는 2^{s-1} 노드들을 가지고 있다. 각 클러스터 s내의 노드들의 순서화된 리스트는 식 (1)의 함수 $c_{i,s}$ 에 의해 정의되어진다. 초기에 클러스터 s내의 노드 i의 첫번째 이웃이 계산되고, 그리고 그 클러스터내의 나머지 노드들은 하이퍼큐브 이웃의 클러스터들 1,..., s-1의 노드들이다.

$$c_{i,s} = i \oplus 2^{s-1} \parallel c_{i \oplus 2^{s-1}, k} \mid k = 1, \dots, s-1 \quad (1)$$

Fig. 5의 예를 고려해 보면, 여기서 p_2 가 토픽 t_2 와 관련된 메시지 m_2 를 발행하고, 단지 p_0, p_2, p_3, p_5 , 그리고 p_7 이 구독자들이다. 이 경우에, p_2 는 p_6 이 $c_{2,3}$ 의 첫 번째 노드이지만 t_2 를 구독하지 않았기 때문에 t_2 의 구독자들 CHILDREN(2, t_2 , 3)=(3, 0, 7)인 그것의 $\log_2 N = 3$ 클러스터들의 각 자식에게 m_2 를 전송한다. m_2 를 수신한 p_7 은 클러스터 $c_{7,2} = \{5, 4\}$ 에서 p_5 가 t_2

의 구독자인지 CHILDREN(7, t_2 , 2)=(5)를 확인하고, m_2 를 p_5 로 전송한다. 여기서 CHILDREN(i, t, h)는 노드 i에 가상적으로 연결된 모든 노드들의 집합을 반환한다.

Sun 등 [14]는 응용 레벨 오버레이 네트워크를 구축하기 위하여 격자 큐립을 사용하고, 그리고 PS 응용과 통합하는 격자 큐립 기반 PS 시스템(GQPS)을 제안하였다. GQPS에서, 각 노드는 $O(\sqrt{n})$ 길이 이웃 리스트를 관리하므로 어떤 두 노드간의 경로는 2 홉을 초과하지 않는다. 그리고 Bae [15]는 링크 및 노드 장애가 있을 경우 메시지/이벤트를 안정적으로 전달할 수 있도록 큐립 기반 계층적 결합 포용 PS 시스템인 QHFPS를 제안하였다.

3. 하이퍼큐브 큐립의 설계 및 응용

3장에서는 옛지 계층의 하이퍼큐브 망에 메시 구조의 오버레이 망을 임베딩하여 HGQ를 설계하고, 설계된 HGQ의 응용 예로 IoT 환경에서 센서 데이터 전달 방법을 제안한다.

3.1 HGQ의 설계

제안하는 HGQ를 실행하기 위한 전반적인 시스템 구조는 Fig. 6과 같이 3 가지 계층들: 클라우드, 옛지, IoT 장치로 구성된다. 여기서 옛지 계층에 하이퍼큐브 망이 구축되어진다. IoT 장치 계층의 스마트 장치들은 발행자 또는 구독자가 될 수 있고, PS 브로커는 옛지 계층의 모든 노드들에 배치되어지고, 어떤 IoT 장치의 발행자 브로커나 구독자 브로커가 제출한 정보를 옛지 계층의 PS 브로커들에게 멀티캐스트 하기 위하여 옛지 계층의 상단에 옛지 노드들로 하이퍼큐

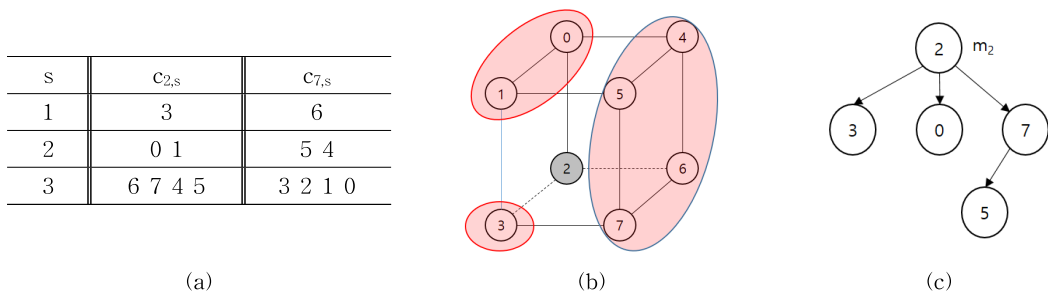


Fig. 5. VCube hierarchical organization and broadcast tree for node 2 [12]. (a) $c_{i,s}$ table for $i \in \{2, 7\}$ and $N=8$ nodes, (b) 3-VCube for node 2, and (c) Broadcast tree for source node 2 and topic t_2 .

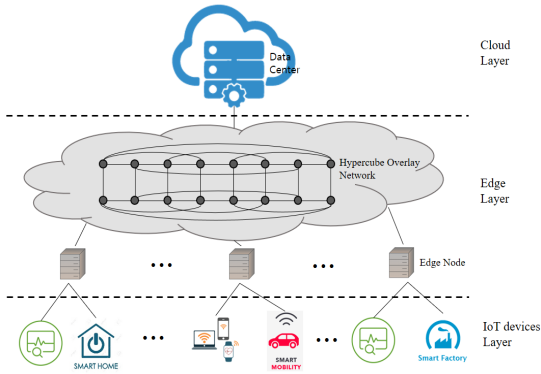


Fig. 6. Overall system structure for executing HGQ.

브 구조의 네트워크를 설립한다.

발행자 브로커나 구독자 브로커는 하이퍼큐브 구조의 네트워크의 발행자 브로커 쿼럼과 구독자 브로커 쿼럼을 각각 사용하여 발행 또는 구독 메시지를 이벤트를 효율적으로 멀티캐스트한다.

분산 PS에서 발행자 브로커 p부터 구독자 브로커 s로의 발행 메시지 전달을 고려하면, 발행자 브로커 p는 자신을 포함한 이웃 PS 브로커 노드들의 집합인 발행자 브로커 쿼럼 $Q_{PB}(p)$ 로 발행 메시지를 멀티캐스트하고, 또한 수신자 브로커 s는 자신을 포함한 이웃 PS 브로커 노드들의 집합인 구독자 브로커 쿼럼 $Q_{SB}(s)$ 로 구독 메시지를 멀티캐스트한다. 쿼럼이 갖는 성질에 의해 $Q_{PB}(p) \cap Q_{SB}(s) \neq \emptyset$ 이기 때문에 교집합에 속하는 PS 브로커가 라우터 포인트 역할을 수행하여 발행 메시지가 s에서 t로 전달된다. 따라서

분산 PS 시스템에서 발행자와 구독자간의 메시지 전달을 위하여 각 PS 브로커 i의 발행자 하이퍼큐브 쿼럼 $HQ_{PB}(i)$ 와 구독자 하이퍼큐브 쿼럼 $HQ_{SB}(i)$ 를 구축해야 한다.

HGQ의 $HQ_{PB}(i)$ 와 $HQ_{SB}(i)$ 를 구축하기 위하여 먼저 엣지 계층의 하이퍼큐브 망에 2-차원 메쉬 구조를 임베딩한다. 하이퍼큐브에 메쉬를 임베딩하는 것은 하이퍼큐브에 링을 임베딩하는 것의 자연스러운 확장이다. 메쉬의 노드 (i, j)를 하이퍼큐브의 $G(i, r-1) \parallel G(j, s-1)$ 에 매핑하여 $2^r \times 2^s$ 메쉬는 2^{r+s} 노드 하이퍼큐브에 삽입할 수 있다. 여기서 \parallel 는 2개의 그레이 코드들의 연결을 나타낸다. 메쉬에서 인접한 이웃들은 레이블이 정확히 한 비트 위치에서 다른 하이퍼큐브 노드로 매핑된다[16].

Fig. 7은 4-차원 하이퍼큐브에 $2^2 \times 2^2$ 메쉬 임베딩을 보여준다. 메쉬를 하이퍼큐브로의 사상은 다음과 같은 유용한 성질을 갖는다. 메쉬의 동일한 행은 레이블에 r, 즉 2개의 동일한 최상위 비트가 있는 하이퍼큐브 노드에 사상되고, 메쉬의 각 행은 2s, 즉 $2 \times 2 = 4$ 노드들을 가지고, 그리고 메쉬의 각 행은 하이퍼큐브의 별개의 서브큐브로 사상된다. 유사하게 메쉬에서 각 열은 하이퍼큐브에서 별개의 서브큐브로 사상된다.

Fig. 7(b)의 Q_4 에서 노드 0111이 분산 PS의 발행자 브로커 노드이면, Fig. 7(a)와 같이 하이퍼큐브에 임베딩된 메쉬 구조의 발행자 브로커 노드 0111의 행 쿼럼으로 발행자 브로커 쿼럼 $HQ_{PB}(0111) = \{0100,$

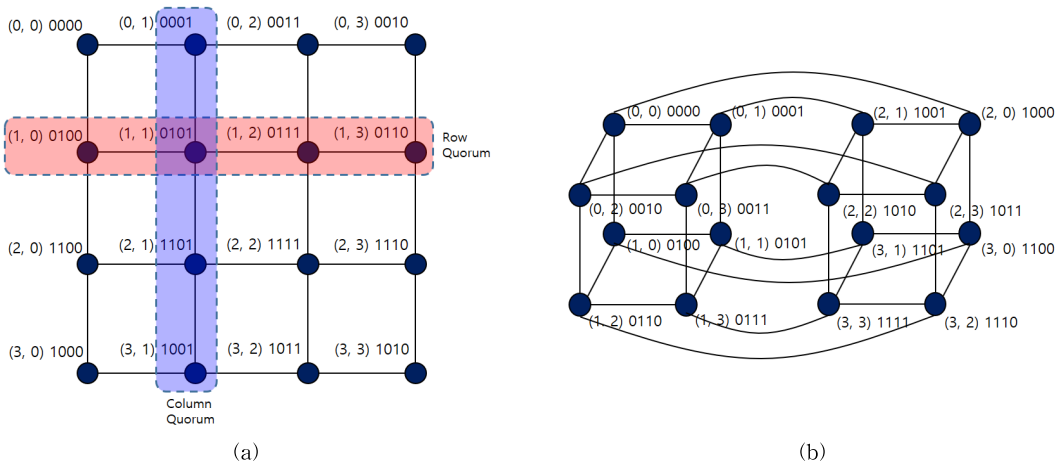


Fig. 7. Embedding 4×4 mesh in Q_4 . (a) 4×4 mesh and (b) Q_4 structure.

0101, 0111, 0110)을 구할 수 있다. 이것을 Q_k 로 일반화하여 발행자 브로커 노드들

$p = p_{k-1} \cdots p_{\lceil k/2 \rceil} p_{\lceil k/2 \rceil - 1} \cdots p_0$ 라 하면, Q_k 로부터 발행자 브로커 노드 p 의 $HQ_{PB}(p)$ 는 식 (2)를 사용하여 구할 수 있다.

$$HQ_{PB}(p) = p_{k-1} \cdots p_{\lceil k/2 \rceil} \parallel x_{\lceil k/2 \rceil - 1} \cdots x_0 \quad (2)$$

여기서 x 는 와일드 카드 매치를 위한 무관 비트(don't care bit)를 나타낸다. 따라서 $HQ_{PB}(p)$ 는 발행자 브로커 노드의 비트열 레이블의 하위 비트열 $p_{\lceil k/2 \rceil - 1} \cdots p_0$ 에서 해밍 거리(H)가 $\lceil k/2 \rceil$ 보다 작거나 같은 PS 브로커 노드들의 집합이다.

Q_4 에서 노드 1001이 분산 PS의 구독자 브로커 노드이면, Fig. 7(a)와 같이 하이퍼큐브에 임베딩된 메쉬 구조의 구독자 브로커 노드 1010의 열 쿼럼으로 구독자 브로커 쿼럼 $HQ_{SB}(1001) = \{0001, 0101, 1101, 1001\}$ 을 구할 수 있다. 이것을 Q_k 로 일반화하여 구독자 브로커 노드를 $s = s_{k-1} \cdots s_{\lceil k/2 \rceil} s_{\lceil k/2 \rceil - 1} \cdots s_0$ 라 하면, Q_k 로부터 발행자 브로커 노드 s 의 $HQ_{SB}(s)$ 를 식 (3)을 사용하여 구할 수 있다.

$$HQ_{SB}(s) = x_{k-1} \cdots x_{\lceil k/2 \rceil} \parallel s_{\lceil k/2 \rceil - 1} \cdots s_0 \quad (3)$$

따라서 $HQ_{SB}(s)$ 는 구독자 브로커 노드의 비트열 레이블의 상위 비트열 $s_{k-1} \cdots s_{\lceil k/2 \rceil}$ 에서 H가 $\lceil k/2 \rceil$ 보다 작거나 같은 PS 브로커 노드들의 집합이다.

Q_4 의 예에서, $HQ_{PB}(0111) \cap HQ_{SB}(1001) = \{0101\}$ 이므로, Fig. 7(a)와 같이 노드 0101는 발행자 노드 0111과 구독자 노드 1001 사이의 랑데부 포인트 역할을 한다. 따라서 발행자 노드 0111가 발행한 모바일 장치들의 메시지나 이벤트가 랑데부 포인트인 노드 0101을 거쳐 구독자 노드 1001에게 전달되어진다.

정리 1. n 노드들로 구성된 Q_k 에서 어떤 발행자 노드에서 HGQ의 랑데부 노드를 거쳐 어떤 구독자 노드 까지의 가장 긴 경로의 길이는 $k = lgn$ 이다.

증명: 발행자 노드 p 에서 $HQ_{PB}(p)$ 의 노드들까지 가장 긴 경로의 길이는 $\lceil k/2 \rceil - 1 + 1 = \lceil k/2 \rceil$ 이고, 또한 구독자 노드 s 에서 $HQ_{SB}(s)$ 의 노드들까지 가장 긴 경로의 길이는 $(k-1) - \lceil k/2 \rceil + 1 = \lfloor k/2 \rfloor$ 이다. 따라서 p 에서 HGQ의 랑데부 노드를 거치는 q 까지의 가장 긴 경로의 길이는 $\lceil k/2 \rceil + \lfloor k/2 \rfloor = k$ 이고, Q_k 에서 $n = 2^k$ 이므로 $k = lgn$ 이다.

3.2 HGQ의 응용

최근에 헬스케어, 보안, 에너지 그리고 산업 자동화와 같은 중요한 분야에서 IoT에 대한 요구가 증가하고 있다. 그것들을 핵심 업무 IoT 응용이라 하고, 그리고 지연, 처리력 또는 신뢰성 측면에서 특정 QoS 보장을 추가적으로 요구한다. 이러한 이유로 IoT 응용들의 성공적인 작업은 센서들과 서버들 사이의 신뢰성 있는 데이터 전송에 의존한다. 따라서 근원지로부터 목적지까지 최소지연으로 데이터의 성공적인 전송은 대형 IoT 시스템의 작동을 위해 필수적이다. 만일 데이터가 요구되는 지연 안에 공유되지 않으면 IoT 응용들은 잘 작동하지 않을 수 있다 [17,18].

본 논문에서는 엣지 계층에 구축된 HGQ에 기반한 분산 발행/구독 모델(HGQ-DPS)을 위한 데이터 전달 방법을 제안한다. Fig. 8는 제안하는 HGQ-DPS 기반 데이터 전달 방법에서 발행 메시지/이벤트와 구독 메시지/이벤트를 분산하고 공지 메시지가 전달되는 과정에 대한 메시지 시퀀스 다이어그램을 보여준다.

Fig. 8에서 IoT 장치 계층의 센서나 모바일 장치의 데이터/이벤트를 발생시키는 발행자 모바일 장치 p 는 그 장치의 엣지 계층의 지역 PS 브로커에게 'publish(tp, m, ts)'를 전송하여 데이터/이벤트 등을 포함하는 메시지를 발행하고, 그 지역 PS 브로커는 발행자 브로커 쿼럼 $HQ_{PB}(p)$ 내의 모든 브로커들에게 'publish(tp, m, ts)'를 멀티캐스트하여 발행 메시지를 분산한다. 여기서 매개변수 tp, m, ts는 발행 메시지의 토픽, 메시지 그리고 타임스탬프를 각각 나타낸다. 그리고 어떤 토픽에 대한 정보의 구독을 원하는 장치 s 는 'subscribe(tp)'를 전송하여 메시지 구독을 신청한다. 만일 그 장치 s 의 엣지 계층의 지역 PS 브로커가 구독을 원하는 장치의 메시지를 가지고 있으면 공지하고, 아니면 그 지역 PS 브로커의 구독자 브로커 쿼럼 $HQ_{SB}(s)$ 내의 모든 브로커들에게 그 구독 메시지를 멀티캐스트하여 관심 메시지를 공지한다.

4. 성능 평가

4장에서는 HGQ의 응용로 제안하는 HGQ-DPS 기반 데이터 전달 방법의 성능을 분석적 모델로 평가한다. 여기서는 분산 발행/구독 방식의 시스템인 제

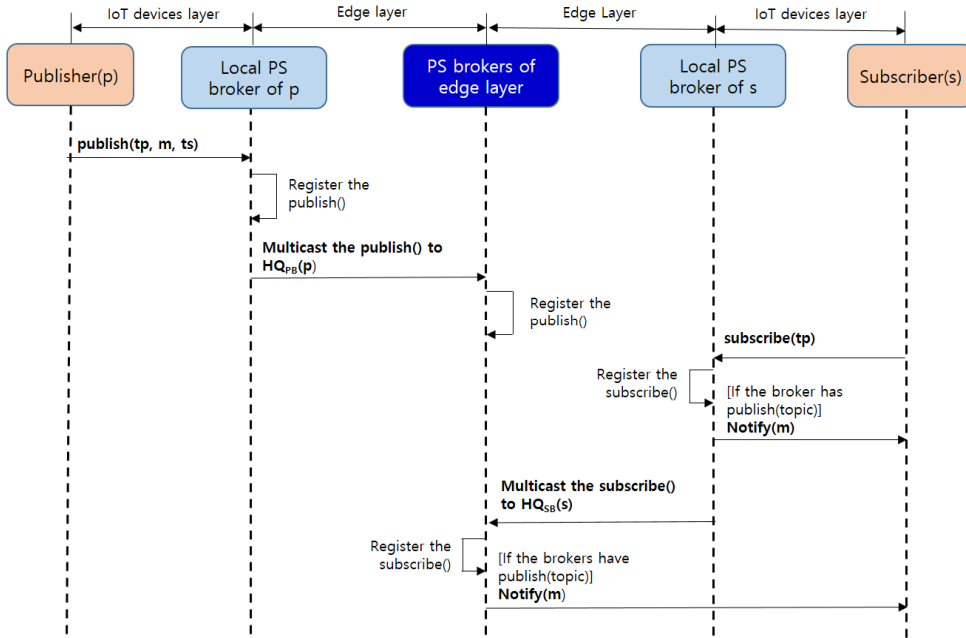


Fig. 8. Message sequence diagram for the HGQ-DPS based data dissemination method.

안하는 HGQ-DPS, VCube-PS, 그리고 GQPS의 성능을 비교하고 평가한다. 우리는 성능 평가 척도로 DPS의 PS 브로커의 개수에 따른 어떤 메시지의 발행에서부터 그 메시지의 구독까지의 구독 지연, 구독자의 수에 따른 전송되는 메시지 발행 비용과 구독 메시지 구독 비용을 합한 전체 관리 비용으로 성능을 각각 평가한다. 여기서 각 평가 항목은 최악 경우의 성능으로 평가한다.

구독 지연은 발행 메시지가 발행자로부터 구독자에게 전달될 때 까지의 경로 상의 홉 개수로 평가한다. n 개의 PS 브로커들로 구성된 DPS에서 HGQ의 최악 경우의 구독 지연(d_{HGQ})을 식 (4)로 구할 수 있다.

$$d_{HGQ} = lgn + 2 \tag{4}$$

여기서 정리 1로부터 발행자 노드에서 HGQ의 랑테부 노드를 거쳐 구독자 노드까지의 가장 긴 경로의 길이는 lgn 이고, 상수 2는 발행자의 IoT 모바일 장치와 지역 PS 브로커를 연결하고 구독자의 IoT 모바일 장치와 지역 PS 브로커를 연결하는 홉스이다.

Fig. 9는 DPS의 PS 브로커의 개수에 따른 구독 지연의 성능을 보여준다. 제안하는 HGQ-DPS의 구독 지연이 VCube-PS와 GQPS 보다 짧아 성능이 우

수함을 확인하였다. 또한 HGQ-DPS는 단 하나의 발행자 PS 브로커를 갖는 VCube에 비해 구독자들은 발행 메시지 사본을 갖는 인접한 이웃 PS 브로커로부터 메시지를 구독할 수 있기 때문에 구독 지연은 더 짧아 질 것이다. 그러나 VCube와 GQPS는 HGQ보다 긴 직경을 가지기 때문에 긴 구독 지연이 발생한다. 따라서 HGQ-DPS는 실시간 또는 시간에 민감한 IoT 응용과 같은 핵심 업무 IoT 응용에 적합한 발행/구독 모델이다.

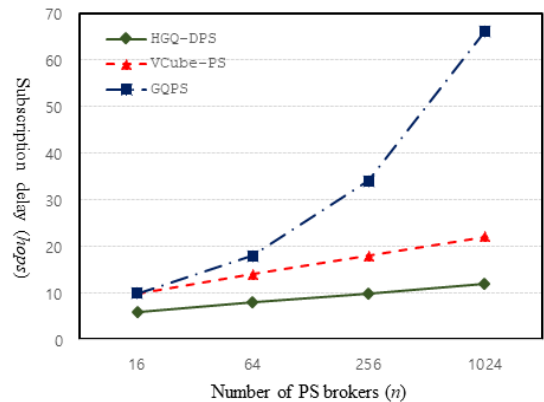


Fig. 9. Subscription delay according to number of PS brokers in the worst case.

DPS의 메시지 발행 비용과 메시지 구독 비용을 합한 식 (5)로 동적 처리 비용인 전체 관리 비용을 평가한다.

$$C_{total} = n_{pb}C_{pub} + r_{sub}n_{sb}C_{sub} \quad (5)$$

여기서 n_{pb} 와 n_{sb} 는 메시지 발행 당 PS 브로커들에게 전송되는 발행 메시지 개수와 메시지 구독 당 PS 브로커들에게 전송되는 구독 메시지 개수를 각각 나타내고, C_{pub} 와 C_{sub} 는 각 메시지의 가장 긴 전송 경로로 최악 경우의 발행 비용과 구독 비용을 각각 나타낸다. 그리고 r_{sub} 는 발행 메시지 당 구독자의 수를 나타낸다.

$2^6 = 64$ 개의 PS 브로커들로 구성된 DPS에서 구독자의 수에 따른 DPS의 전체 관리 비용을 Fig. 10에서 보여준다. 여기서 쿼럼에 기반하는 HGQ-DPS와 GQPS는 쿼럼내의 PS 브로커 노드들에 발행/구독 메시지의 사본을 저장하기 때문에 VCube-PS 보다 전체 관리 비용의 성능이 나쁘다. 그러나 제안하는 HGQ-DPS는 GQPS 보다 쿼럼 크기가 작은 쿼럼을 사용할 뿐만 아니라 직경이 작은 하이퍼큐브를 사용하기 때문에 GQPS 보다 성능이 우수하나 VCube-PS와 거의 유사하지만 관리 비용이 조금 더 든다.

제안하는 HGQ-DPS는 발행/구독 메시지를 쿼럼내의 PS 브로커 노드들에게 전송하여 메시지 사본을 분산 저장하기 때문에 전체 관리 비용은 약간 높지만 메시지 사본을 분산 저장하지 않은 VCube-PS에 비해 다음과 같은 장점을 갖는다.

- HGQ-DPS는 발행자 브로커 쿼럼 $HQ_{PB}(p)$ 내

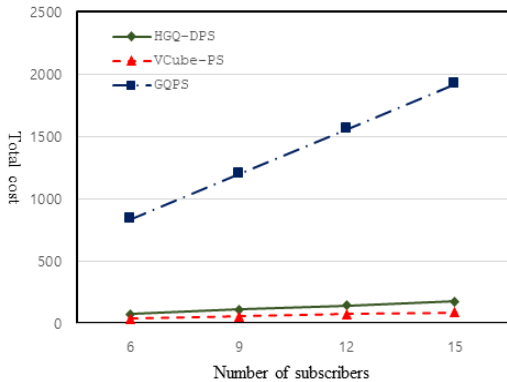


Fig. 10. Total cost according to number subscribers in the worst case.

의 모든 브로커들에 발행 메시지의 사본이 저장되기 때문에 결합 포용성을 제공한다. 반면에 VCube-PS는 발행 메시지를 단지 지역 PS 브로커에만 저장하기 때문에 그 지역 PS 브로커가 고장이면, 그 발행 메시지는 구독할 수 없는 단일 고장점을 갖는다.

- HGQ-DPS는 $HQ_{PB}(p) \cap HQ_{SB}(s) \neq \emptyset$ 이기 때문에 랭데부 포인트 역할을 하는 PS 브로커가 존재한다. 따라서 구독자 s는 발행자 p의 위치 정보를 알 필요가 없다. 반면에 VCube는 랭데부 역할을 하는 PS 브로커 노드가 없기 때문에 구독자 s는 관심 메시지를 발행하는 지역 PS 브로커의 식별자와 같은 위치 정보를 알아야 한다.

- HGQ-DPS는 브로커 쿼럼 $HQ_{PB}(p)$ 내의 모든 브로커들에 발행 메시지의 사본이 저장되기 때문에 구독자들은 그 사본을 가지고 있는 지역 PS 브로커의 인접 PS 브로커들로부터 메시지를 구독할 수 있다. 따라서 구독 요청들이 $HQ_{PB}(p)$ 내의 노드들에 분산되어진다. 반면에 VCube-PS는 발행 메시지를 단지 지역 PS 브로커에만 저장하기 때문에 그 메시지에 관심이 있는 모든 구독자는 메시지를 발행한 지역 PS 브로커에게만 구독 메시지를 전송해야한다. 따라서 모든 구독 메시지가 발행자 지역 PS 브로커에 집중되어 발행자 지역 PS 브로커에 병목 현상이 발생한다.

5. 결론

5G의 과제, 특히 핵심 업무 IoT 통신을 해결하기 위하여 완전한 설계 패러다임 변화가 필요하다는 것은 분명한 사실이다. Pub/Sub 시스템은 실시간으로 처리되어 가입자에게 전달되어야 하는 대용량 데이터를 생성하는 시스템에 적합하고, 다양한 사용 영역에 대한 적응성을 갖는 패러다임이다.

본 논문에서는 분산 PS 시스템에서 IoT 응용을 위한 하이퍼큐브 격자 쿼럼인 HGQ를 설계하고, HGQ의 응용으로 옛지 오버레이 망에 구축된 HGQ 플랫폼에 기반한 센서 데이터 전달 메커니즘을 제안하였다. 그리고 HGQ의 성능을 분석적 모델로 평가하였다, 그 결과, 제한한 HGQ가 기존의 DPS들에 비해 구독 지연이 우수할 뿐만 아니라 결합 포용성, 병목 현상 없음(bottleneck free), 발행자 지역 PS의 위치 인식 필요 없음(no location-aware)을 제공한다는 우수한 DPS 플랫폼이라는 것을 확인하였다.

향후 연구 과제로는 제안하는 HGQ의 구현과 구현된 시스템의 성능평가, 망 성능과 통신 비용 간의 균형을 이루기 위하여 하이퍼큐브의 다소의 변종들: 피비우스 큐브, 꼬임 큐브, 회전 큐브 등에 기반한 쿼럼 구조에 대한 연구 등이 있다.

REFERENCE

- [1] S. Haseeb¹, A.H.A. Hashim¹, O.O. Khalifa¹, and A.F. Ismail, "Connectivity, Interoperability and Manageability Challenges in Internet of Things," *AIP Conference Proceedings*, Vol. 1883, No. 1. pp. 1-7, 2017.
- [2] V.N. Phan, V.D. Nguyen, T.D.T. Nguyen, and E.N. Huh, "Efficient Edge-Cloud Publish/Subscribe Broker Overlay Networks to Support Latency-Sensitive Wide-Scale IoT Applications," *Symmetry* 2020, Vol. 12, Issue 1, No. 3, pp. 1-18, 2020.
- [3] C. Chen, R. Vitenberg, and H.A. Jacobsen, "A Generalized Algorithm for Publish/Subscribe Overlay Design and Its Fast Implementation," *Lecture Notes in Computer Science*, Vol. 7611, pp. 76-90, 2012.
- [4] H. Howard, D. Malkhi, and A. Spiegelman, "Flexible Paxos: Quorum Intersection Revisited," *Proceedings of the 9th Workshop on Principles and Practice of Consistency for Distributed Data*, pp 16 - 23, 2022.
- [5] M. Imani and M.D.T. Fooladi, "S-Grid: A New Quorum-based Power Saving Protocol to Maximize Neighbor Sensibility," *25th Iranian Conference on Electrical Engineering*, pp. 2134-2139, 2017.
- [6] J.H. Hong, K.C. Lee. and S.Y. Lee, "Trends in Edge Computing Technology," *Electronics and Telecommunication Trends*, Vol. 35, No. 5, pp. 78-87, 2020.
- [7] C.-K. Lin, L. Zhang, J. Fan, and D. Wang, "Structure Connectivity and Substructure Connectivity," *Theoretical Computer Science*, Vol. 634, pp 97-107, 2016.
- [8] Y. Saad and M.H. Schultz, "Topological Properties of Hypercube," *IEEE Transactions on Computers*, Vol. 37, No. 7, pp. 867-872, 1988.
- [9] C.-J. Lai and C.-H. Tsai, "Embedding a Family of Meshes into Twisted Cubes," *Information Processing Letters*, Vol. 108, No. 5, pp. 326-330, 2008,
- [10] X. Wang, J. Liang, D. Qi, and W. Lin, "The Twisted Crossed Cube," *Concurrency and Computation: Practice and Experience*, Vol. 28, No. 5, pp. 1507-1526, 2016.
- [11] I. Livaja, D. Skvore, and K. Pripuzic, "Geospatial Publish/Subscribe Systems for the Internet of Things," *International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1-8, 2017.
- [12] B. Garbinato, H. Miranda, and L. Rodrigues, *Middleware for Network Eccentric and Mobile Applications*, Springer-Verlog, Berlin Heidelberg, 2009.
- [13] J.P. Araujo, L. Arantes, E.P. Duarte Jr., and L.A. Rodrigues, "VCube-PS: A Causal Broadcast Topic-based Publish/Subscribe System," *Journal of Parallel and Distributed Computing*, Vol. 125, pp. 18-30, 2019.
- [14] Y. Sun, X. Qiao, B. Cheng, and J. Chen, "A Low-delay, Lightweight Publish/Subscribe Architecture for Delay-sensitive IoT Services," *International Conference on Web Services*, pp. 179-186, 2013.
- [15] I.H. Bae, "Design and Evaluation of a Fault-tolerant Publish/Subscribe System for IoT Applications," *Journal of Korea Multimedia Society*, Vol. 24, No. 8, pp. 1101-1113, 2021.
- [16] A. Grama, V. Kumar, G. Karypis, and A. Gupta, *Introduction to Parallel Computing*, Pearson Education, Harlow England, 2003.
- [17] I.H. Bae and H.T. Noh, "Design and Evaluation of a Quorum-Based Adaptive Dissemination Algorithm for Critical Data in IoTs," *Journal of Korea Multimedia Society*, Vol. 22, No. 3, pp. 913-922, 2019.
- [18] M.Z. Khan, O.H. Alhazmi, M.A. Javed, H.

Ghandorh, and K.S. Aloufi, "Reliable Internet of Things: Challenges and Future Trends," *Electronics*, Vol. 10, No. 19, pp. 1-22, 2021.



배 인 한

1990년 중앙대학교 컴퓨터공학과
공학박사

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

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

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

1989년~현재 대구가톨릭대학교 컴퓨터소프트웨어학부
교수

관심분야 : 사물인터넷, 차량클라우드, 차량통신망, 머신
러닝, 지능스마트 플랫폼 등