

## 모바일 애드혹 환경에서 빠른 경로 복구를 지원하는 Publish/Subscribe 기법

문상철<sup>○</sup> 고양우, 이동만

한국정보통신대학교

[\[sangchul, newcat, dlee}@icu.ac.kr](mailto:{sangchul, newcat, dlee}@icu.ac.kr)

### A Fast Path Recovery Publish/Subscribe Scheme for Mobile Ad-hoc Environments

Sang-chul Moon<sup>○</sup> Yangwoo Ko, Dongman Lee

Information and Communications University

Publish/Subscribe (Pub/Sub)는 애드혹 유비쿼터스 컴퓨팅 환경에서 정보의 발행자(Publisher)와 구독자(Subscriber)를 이벤트 브로커를 매개로 하여 서로 시간적 공간적으로 분리시킨다[1]. 이러한 특징으로 인해 Pub/Sub는 상황의 변화에 따라 상호작용 상대를 계속해서 변화시켜야 하는 유비쿼터스 컴퓨팅 환경에서 즉각적인 상호작용 (Spontaneous Interaction)[2]을 지원하는데 효과적인 상호작용 모델이다.

인프라스트럭처를 가정할 수 없는 모바일 애드혹 환경에 Pub/Sub를 적용하기 위해서 기존 연구는 지정된 이벤트 브로커를 가정하지 않고 모든 모바일 디바이스가 이벤트 브로커의 역할을 나눠서 수행하도록 하였다. 그리고 분산되어 있는 이벤트 브로커들이 서로 정보를 교환하고 협동하여 Pub/Sub기능을 수행하도록 하였다. 여기서 서로 정보를 교환하고 협동하는 방법에 있어 기존의 연구는 크게 두 가지 방법으로 나뉜다. 하나는 구조적인 방법이며 다른 하나는 비구조적인 방법이다 구조적인 방법은 이벤트의 발행자로부터 구독자까지 전달 경로를 설정하고 유지함으로써 이벤트가 최적화된 경로를 통해서만 전달되도록 하여, 이벤트를 전달하는데 드는 비용이 상대적으로 작다 그러나 비구조적인 방법에서는 이동성이 큰 환경에서 이러한 이벤트 전달 경로를 유지하는데 비용이 더 많이 든다는 점에 착안하여 이러한 경로를 가정하지 않고 Flooding (또는 Gossiping 이나 Epidemic 방식)을 통해 이벤트를 경로 없이 전파한 후 각 발행자들이 자신의 관심에 맞게 이벤트의 수신 여부를 결정한다 본 연구에서 가정하고 있는 유비쿼터스 컴퓨팅 환경에서는 서비스의 종류가 다양하지만 서비스들이 개인화되어 있어서 해당 서비스를 이용하는 사용자의 수가 상대적으로 매우 적기 때문에 비구조적인 방법을 통해 서비스를 제공하게 될 경우 불필요한 정보 전송으로 인한 비용이 크게 증가하게 된다 따라서, 구조적인 방법을 통해 정보의 발행자와 구독자간의 이벤트 전달 경로를 설정하고 유지하는 방법이 더 적합하다

구조적 방법을 통해 Pub/Sub를 구성하는 방법인 Huang[3]의 연구는 이동성으로 인한 이벤트 전달 경로 손상을 적절히 복구하지 못하는 단점이 있다 Huang의 연구는 각 노드가 독립적으로 경로 구성 정보를 전파하고 이 정보를 통해 각 노드는 가장 적은 오버헤드로 트리를 구성할 수 있는 부모 노드를 선택하여 이벤트 전달 경로를 설정하고 유지한다 그러나, 기기들이 이동하는 환경에서는 루트로의 최단 경로가 계속해서 변하게 되기 때문에 트리 구조를 유지하기 위해서 각 노드들이 전파하는 경로 구성 정보는 변화된 경로 및 거리를 반영하고 있어야 한다 하지만 Huang의 연구와 같이 각 노드가 독립적으로 트리 정보를 전파하게 되면 각 노드는 루트로부터의 거리 정보를 빠르고 정확하게 알 수 없는 문제가 있다. 루트가 이동함에 따라 루트의 새로운 위치를 반영하는 트리 구성 정보가 전체 네트워크에 퍼지기 위해선 최악의 경우 루트로부터 말단 노드들까지의 홉 수에 트리 정보 전파 주기의 시간을 곱한 만큼 소요된다. 그 이유는 트리 구성 정보가 루트로부터 말단 노드까지 순차적으로 전파되는 것이 아니라 모든 노드가 독립적으로 전파하기 때문이다 따라서 루트 노드의 새 위치를 아직 알지 못한 노드들

은 잘못된 거리 정보를 전파하게 되어 결국 효율적인 이벤트 전달 트리를 구성하는데 실패하게 되고 구독자들은 트리 구조가 다시 복구되지 않는 한 이벤트를 수신할 수 없게 될 수도 있다

따라서 본 연구에서 제안하는 기법은 트리를 구성하기 위해 각 노드가 독립적으로 트리 구성 정보를 전파하는 대신 루트로부터 순차적으로 이 정보를 전파한다각 노드는 루트로부터 전달된 트리 구성 정보를 통해 부모 노드를 결정하고 이를 다시 전파한다이러한 과정을 통해 트리 구성 정보가 말단 노드까지 전파되면 모든 노드들은 매 주기마다 루트로의 정확한 거리 정보를 알 수 있게 되어 이벤트 전달 트리를 더 정확하게 유지할 수 있게 된다 이와 더불어, 이벤트 전달 경로를 구성하는데 있어 Join 메시지 전송이 유실되면 트리 구성에 실패하게 되고 이는 이벤트 수신율 저하를 초래한다따라서 안정적으로 트리를 구성하기 위해 Join 메시지의 전송을 보장해야 한다 따라서 제안하는 기법에서는 Join 메시지에 대한 안정적인 전송을 보장하기 위해 Acknowledgement(Ack) 기법을 도입하여 부모 노드에게 Join 메시지를 보낸 자식 노드가 일정 시간 동안 Ack를 수신하지 못하였을 경우 Join을 재전송하는 방법을 사용하였다

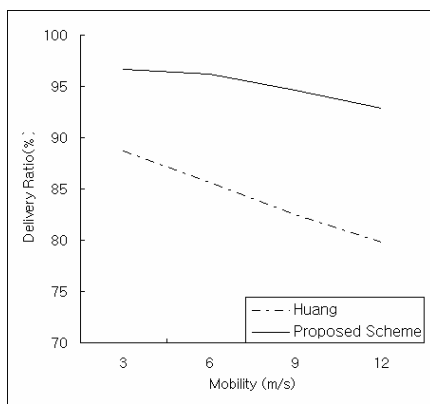


그림 1 이동성의 변화에 따른 비교

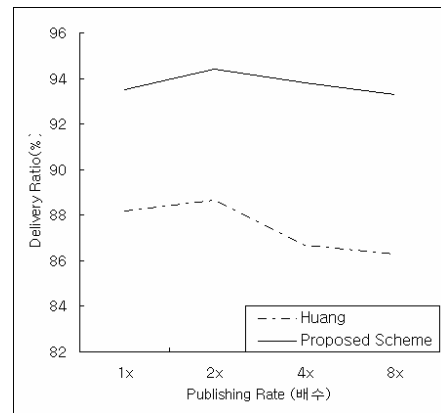


그림 2 이벤트 발행율에 따른 비교

위의 그래프는 기존 연구와 본 연구의 기법을 이벤트 수신율과 오버헤드 측면에서 비교한 결과이다 이 결과는 NS-2를 시뮬레이터로 이용하고 1000x1000m의 공간에 40개의 노드들이 1~12m/s의 속도로 움직이는 환경에서 시뮬레이션을 수행하였다 이벤트 수신율은 각 구독자들이 관심을 가진 전체 이벤트들의 총 발행된 수에서 실제 수신한 이벤트 수의 비율을 나타내며 오버헤드는 하나의 이벤트를 구독자에게 보내기 위해 사용된 총 브로드캐스트의 수를 나타낸다위의 그림 1에서와 같이 본 연구는 기존 연구에 비해서 이동성이 증가함에 따라 8~14% 정도 더 나은 이벤트 수신 성능을 보이고 있다 이는 루트로부터 트리 구성 정보를 순차적으로 모든 노드에게 전파시켜서 트리를 유지하기 때문에 말단 노드부터 루트 노드까지 최적의 경로를 매 전파 주기마다 구성할 수 있어서 이 경로를 통해 이벤트를 전달할 수 있었기 때문이다 그림 2는 트래픽을 증가시킴에 따라 트리 구성 정보 및 Join 메시지의 유실을 유도하여 기존 연구와 제안하는 기법간의 이벤트 수신율을 비교한 결과이다 트래픽이 증가함에 따라 기존 연구는 Join 메시지의 유실로 인해 트리 구성에 실패하게 되어 이벤트의 수신율이 저하된다 그러나 제안하는 기법은 발행율에 따른 이벤트 수신율의 큰 변화가 없다 이는 높은 트래픽에서도 안정적으로 Join 메시지를 전송하여 트리를 구성할 수 있었기 때문이다

본 연구에서는 애드혹 유비쿼터스 컴퓨팅 환경에서 Pub/Sub를 이용하여 기기들이 서로 상호작용을 효과적으로 수행하기 위해 기존 연구들을 분류하고 이들 중 어떠한 Pub/Sub 구성 모델이 적합한지 분석하였다. 그리고 기존 연구의 이벤트 전달 경로 구성 방법의 문제점을 지적하며 좀 더 효과적으로 트리를 구성하여 이벤트 전송율을 높일 수 있는 방법에 대해 설명하였다 그리고 시뮬레이션을 통해 제안하는 기법이 기존 연구에 비해 성능이 더 낫다는 것을 보였다