http://dx.doi.org/10.3745/KIPSTA.2011.18A.5.181

해군 함정 컴퓨팅 체계 정보교환을 위한 Publish Subscribe 모델 기반 아키텍처 및 매칭 알고리즘

김 홍 재[†] · 오 상 윤[††]

요 약

정보우위는 현대전에 있어 승패를 결정할 수 있는 요소로써 효과적인 정보교환을 통해 달성이 가능하다. 현재 해군함정에는 다양한 체계가 탑재되어 운용 중이나 체계 들이 군 독자표준(MIL-STD)과 상용(COTS) 장비들이 혼재 되어 구성이 되어 상호운용성이 제한되고 체계들 간의 정보교환이 어렵다. 이에 본 논문에서는 Publish Subscribe 모델 기반의 함정 체계 간 정보교환 아키텍처를 제안한다. 제안 아키텍처는 Publish Subscribe 모델을 통신 미들웨어로 적용하여 서로 다른 체계들 간의 상호운용성을 향상시키고 정보교환이 가능하도록 하였으며 확장성을 가질 수 있도록 하였다. 그리고 Publish Subscribe Broker의 부하 경감을 위해 개선된 트리 기반의 매칭 알고리즘을 제안하였다. 제안하는 트리 매칭 알고리즘은 트리를 구성하는 각 노드에서 Subscription의 Predicate 정보를 포함하여 이벤트 매칭 시간을 단축할 수 있다. 그리고 본 논문의 성능평가를 통해 제안한 트리매칭 알고리즘이 비교대상 알고리즘에 비해 이벤트 매칭시간을 절감 할 수 있는 것을 확인하였다.

키워드 : Publish Subscribe 모델, 데이터 분산 서비스, 개방형구조, 매칭 트리

A Publish Subscribe Information Exchange Model and A Novel Matching Algorithm for Navy Shipboard Systems

Hongjae Kim[†] · Sangyoon Oh[††]

ABSTRACT

Information superiority is an essential factor in modern warfare and it can be archived by efficient information exchange between systems. Various computing systems are installed on the today's navy vessels. However, it is hard to improve interoperability and efficiency of information exchange since the configurations of installed systems are varying. The military standard and commercial standard are mix-used between systems. In this paper, we propose an information exchange architecture based on Pub/Sub model as a communication middleware to improve interoperability as well as enhancing scalability. We also propose a novel tree matching algorithm to improve a performance of PubSub broker. In the proposed algorithm, each tree nodes have information about predicates of subscription that can reduce event matching time. The performance evaluation results show our proposed algorithm reduces time for matching predicates compare with other algorithms.

Keywords : Publish Subscribe Model, Data Distribution Service, Open Architecture, Matching Tree

1. 서 론

현대전(Modern Warfare)에 있어서 정보 우위(Information Superiority)는 신속한 전장상황 파악, 신속한 의사결정을 통해 전투의 승패를 결정하는 중요한 요소이다. 정보우위는 전투구성원간의 전술정보(적군 또는 아군의 위치, 기동방향 및 속도 등)를 신속, 정확하게 전달함으로써 가능하다. 현재 해군의 함정에는 레이더, 소나 등 다양한 센서 체계와 유도탄, 함포 등 타격 체계 및 전투체계, 지휘통제체계 등 다양한 시스템이 연결되어 복합 시스템(System of System)의 모습을 가진다. 그리고 함정에 탑재되는 체계들이 무인화, 자동화 되는 추세이며 함정 체계의 정보교환 과정이 과거 맨 투 머신(Man to Machine) 형태에서 머신 투 머신 형태로 변하고 있다. 따라서 새로운 체계의 개발에 대한 기간과 비용 문제 해결 이외에도 함정의 정보우위 달성을 위해서 위 체계들 간의 상호운용성 보장 및 효과적인 통신방법이

※ 이 논문은 2011년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원(No.2011-0015089)과 지식경제부 및 정보통신산업진흥원의 대학 IT연구센터 지원사업(NIPA-2011-C1090-1121-0011)의 연구결과로 수행되었음.
† 준 회 원 : 아주대학교 NCW학과 통합과정
†† 정 회 원 : 아주대학교 정보 및 컴퓨터공학부 교수(교신저자)
논문접수 : 2011년 5월 2일
수 정 일 : 1차 2011년 6월 13일
심사완료 : 2011년 6월 29일

요구된다. 그러나 함정에서 사용되는 체계들이 군 독자 표준(MIL-STD) 장비와 일반 상용(COTS) 장비들로 혼재되어 구성되어 상호운용성 및 체계 간의 정보교환이 제한된다.

Publish Subscribe 모델은 비동기적 메시지 기반의 미들웨어로써 기존의 클라이언트 서버 방식 보다 유연한 멀티캐스트 통신능력을 제공한다[1]. 현재 군에서는 Publish Subscribe 모델을 적용하여 전술정보 전파를 연구, 도입 중에 있다. 미 국방성의 NCW(Network Centric Warfare) 능력달성을 위한 가이드라인인 NESI(Net-Centric Enterprise Solutions for Interoperability)[2]에서는 Publish Subscribe 모델 기반의 DDS(Data Distribution Service) 사용을 권장하고 있으며 미 해군의 HiPer-D(High Performance Distributed Computing) 프로젝트에서는 여러 실험을 통해 Publish Subscribe 모델이 다양한 군의 요구사항을 충족하는 것을 확인하였다[3]. 그리고 미 해군에서 함정의 전투체계에 적용하고 있는 개방형구조(Open Architecture)에서는 DDS를 통신 미들웨어 역할로 사용하고 있다[4].

내용 기반(Content based) Publish Subscribe 모델은 메시지를 이벤트 형식으로 전달하게 되며 Broker는 Subscriber의 원하는 이벤트에 대한 Subscription을 Publisher의 이벤트와 매칭(또는 필터링)을 통해 해당하는 Subscriber에게 이벤트를 전달하게 된다. 그러나 참여하는 노드들과 Subscription, 이벤트가 많아질수록 매칭을 수행하는 Broker에 부하가 올라가게 되어 매칭에 소요되는 시간이 늘어나게 되고 전체 Publish Subscribe 모델의 성능에 영향을 미치게 된다[5]. 이에 많은 연구를 통해 트리, DHT 및 DBMS를 이용한 이벤트 매칭 기법이 제안되었다. 그러나 기존의 트리 매칭 알고리즘에 대한 개선점이 존재하여 이를 통한 전체 Publish Subscribe 모델 적용 시스템의 성능향상이 가능하다.

본 논문에서는 함정의 다양한 체계들 간의 상호운용성 확보와 정보교환을 위한 Publish Subscribe 모델 기반 정보교환 아키텍처 및 Broker의 이벤트 매칭 알고리즘을 제안한다. 제안하는 아키텍처는 Publish Subscribe 모델을 통신 미들웨어로 사용하여 상호운용성 및 확장성을 향상하였으며 서로 다른 체계간의 정보교환이 가능하다. 그리고 기존의 Publish Subscribe 모델의 트리매칭 알고리즘에서 각 트리 노드에 추가적인 정보를 저장하여 이벤트 매칭 시 사용하도록 함으로써 기존 알고리즘에 비해 매칭시간을 단축하여 Publish Subscribe 모델의 성능을 향상하도록 하였다.

본 논문의 2장에서 Publish Subscribe 모델, DDS 및 미 해군의 개방형 구조 등을 소개 하고 3장에서는 본 논문에서 제안하는 아키텍처를 제시한다. 4장에서는 Content 기반 Publish Subscribe 모델 이벤트 매칭 알고리즘을 제안한다. 5장에서는 제안한 알고리즘의 성능평가 결과를 제시하고 6장에서 결론을 맺는다.

2. 관련 연구

현재 군 도메인에 Publish Subscribe 모델에 대한 도입이 진행 중에 있다. 본 장에서는 Publish Subscribe 모델에 대한 기본적인 정보와 미 해군 개방형구조 및 개방형구조에 사용되는 DDS를 살펴본다.

2.1 Publish Subscribe 모델

인터넷의 발달로 분산 시스템의 영역이 크게 확장되어 서로 다른 곳에 위치하고 있는 다양한 시스템이 연결될 수 있었다. 그러나 이에 따라 분산 시스템의 복잡성 또한 크게 증가되었는데, 이러한 분산 시스템의 복잡성으로 인해 과거 개별적인 point-to-point방식의 동기식(Synchronous) 통신방식은 현재 또는 미래의 동적이고 대규모 분산 시스템 환경에서는 적합하지 않으며, 비동기식(Asynchronous)의 유연(Flexible) 하고 느슨하게 연결(Loosely coupled)되는 통신방식이 요구된다고 할 수 있다.

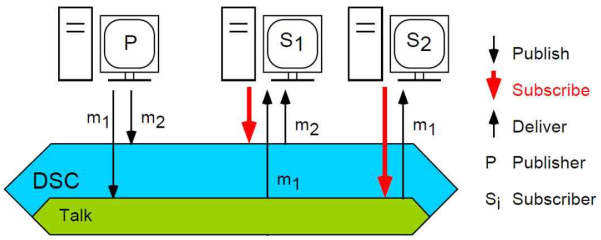
Publish Subscribe 모델은 데이터의 생성이나 삭제로 인해 발생하는 이벤트에 따라서 애플리케이션이 메시지를 네트워크에 주어진 주제 또는 내용에 따라 한 번만 발행, 즉 Publish를 하고 네트워크상의 다수의 애플리케이션들이 해당 주제와 내용에 맞는 메시지를 수신하는 방식이다[6]. Publish Subscribe 모델에서 노드는 이벤트를 생성하는 Publisher, 이벤트를 수신하는 Subscriber 및 중계 역할을 하는 Broker로 구분되며 이벤트를 구독하는 방법에 따라 크게 주제 또는 내용 기반 Publish Subscribe 모델로 나뉜다.

2.1.1 주제 기반 Publish Subscribe 모델

초기 Publish Subscribe 모델은 주제(또는 토픽)을 기반으로 메시지를 교환하도록 고안 되었다. 참여 노드들은 키워드(Keywords)로 구별되는 주제를 통해 이벤트를 발행(Publish)하고 구독(Subscribe)하였으며 여기서 주제는 참여자 그룹, 채널(Channel) 등을 의미한다. Subscriber들은 자신이 관심을 Subscription 형태로 표출하며 해당되는 주제에서 소통되는 모든 이벤트를 수신할 수 있다. 주제 기반 모델에서 Publisher와 Subscriber의 연결은 직접 연결이 아닌 이벤트가 소통되는 채널에 접근함으로써 이루어진다. 그러나 주제 기반 모델에서는 Subscriber가 사전에 이벤트가 소통되는 채널을 인지하여야 하는 문제점이 있다[7]. 주제 기반 Publish Subscribe 모델로는 Scribe[8], Bayuex[9] 등이 있다.

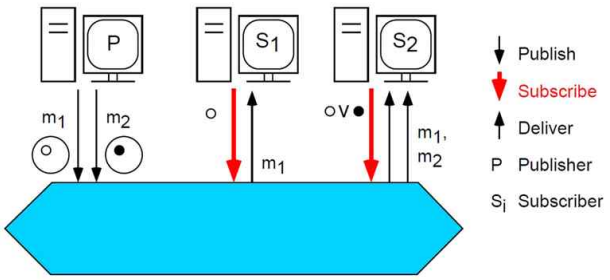
2.1.2 내용 기반 Publish Subscribe 모델

주제 기반 Publish Subscribe 모델은 사전에 주어진 주제



(그림 1) 주제 기반 Publish Subscribe 모델[7]

에 대해 이벤트를 소통을 하도록 되어있어 관심을 정적(Static)으로 표출하게 된다. 내용 기반 Publish Subscribe 모델은 동적(Dynamic)으로 관심을 표출 할 수 있도록 Subscription의 표현성과 유연성을 개선하였다. 내용 기반 Publish Subscribe 모델은 Subscriber의 쿼리문 이나 조건 등에 따라서 이벤트의 내용을 매칭, 일치하는 이벤트를 Subscriber에게 전달함으로써 Subscriber는 이벤트가 소통되는 그룹이나 채널 등에 대해서 알 필요가 없는 장점이 있다[7]. 따라서 이 모델은 주제 기반 모델 보다 유연성, 확장성 등에 있어서 나은 결과를 가져올 수 있다. 그러나 이 모델은 주제 기반 모델에 비해 그 구현이 상대적으로 복잡하며 참여하는 노드의 수와 이벤트 매칭 소요가 증가 시 이벤트 매칭을 수행하는 Broker에 부하를 증가 시킬 수 있어 전체 Publish Subscribe 모델의 성능에 영향을 미칠 수 있다. 내용 기반 Publish Subscribe 모델로는 Siena[10], Narada Brokering[11] 등이 있다.



(그림 2) 내용 기반 Publish Subscribe 모델[7]

2.1.3 내용 기반 Publish Subscribe 모델 이벤트 매칭

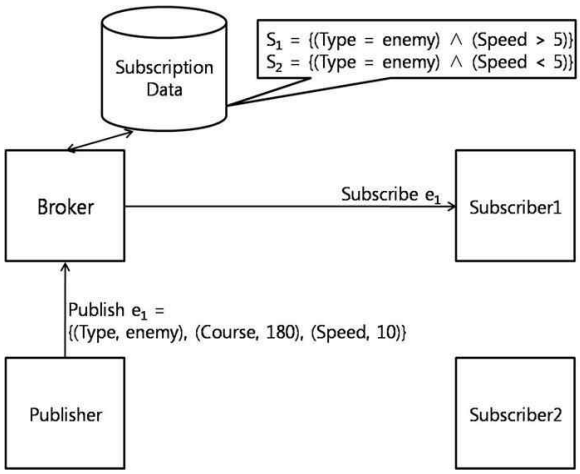
내용 기반 Publish Subscribe 모델의 이벤트 매칭은 Broker가 Publisher의 이벤트를 Subscriber의 Subscription과 비교함으로써 이루어진다. Publisher의 이벤트(e_i)는 {(Type, enemy), (Size, division), (Course, 180), (Speed, 10)}와 같이 어트리뷰트(a_n)와 어트리뷰트 값(v_n)의 쌍 들로 정의된다.

$$e_i = \{(a_1, v_1), (a_2, v_2) \cdot \cdot \cdot (a_n, v_n)\}$$

Subscriber의 Subscription(S_i)은 predicate(Pren)들의 모임이며 predicate는 (type, =, enemy), (speed, >=, 20)과 같이 어트리뷰트, 관계연산자(=, ≠, <, >) 및 어트리뷰트 값으로 정의된다.

$$S_i = \{pre_1 \wedge pre_2 \cdot \cdot \cdot pre_n\}$$
$$Pre_n = (a, op, v)$$

Broker는 Subscriber의 Subscription들을 유지, 이벤트 e_i가 도착하면 비교를 통해 모든 predicate가 이벤트 e_i의 조건에 맞는 Subscription S_i를 찾아내게 된다. 예를 들어 (그림 3)과 같이 Broker가 Subscriber1, 2의 Subscription S_1,



(그림 3) Broker의 이벤트 매칭

S_2를 가지고 있을 때 Publisher가 발행한 이벤트 e_1이 도착하면 S_2의 조건에는 맞지 않으나 S_1의 조건과는 일치, 이벤트 e_1을 Subscriber1에게 전달하며 Subscriber1는 자신이 원하는 이벤트 e_1을 구독하게 된다.

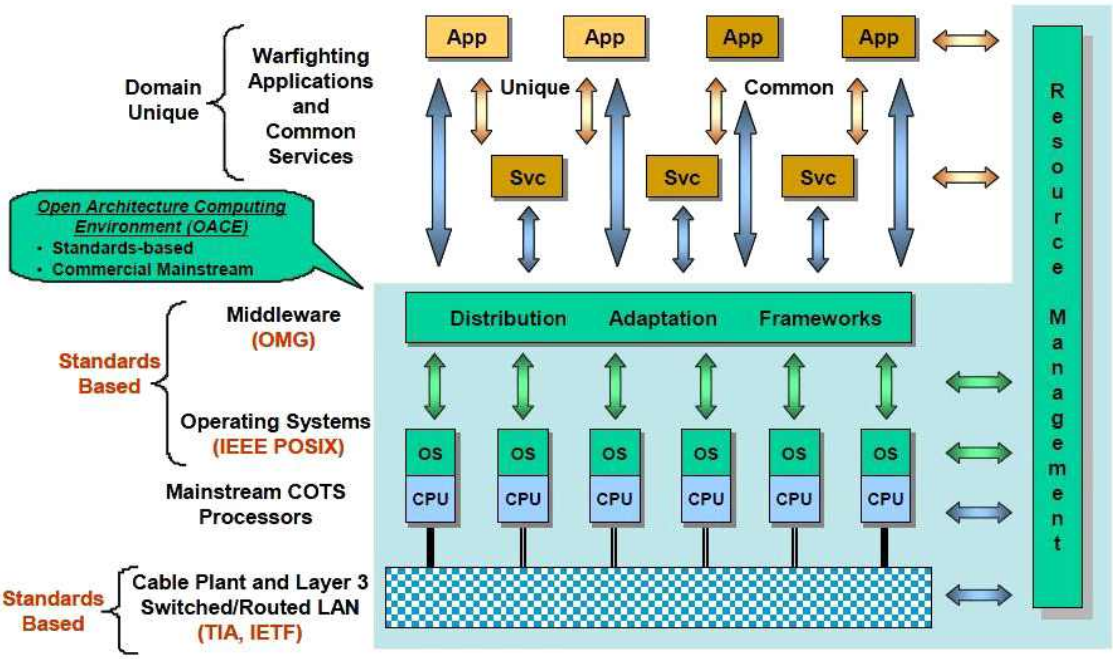
기본적이고 간단한 매칭 알고리즘은 이벤트 도착 시 Broker가 보유하고 있는 모든 Subscription과 순차적으로 비교하는 방법으로 구현이 간단하다. 그러나 이 알고리즘은 참여 노드의 수가 많고 Broker가 유지하고 있는 Subscription, 이벤트 및 어트리뷰트의 수가 늘어날수록 Broker의 CPU, 메모리 사용률이 높아지고 매칭 소요시간이 늘어나게 되어 결국 Publish Subscribe 모델의 전체 성능 저하를 가져오게 된다. 이에 여러 매칭 알고리즘이 연구되었으며 대표적으로 트리[4][12], DHT[13], DBMS 기반 알고리즘[14] 등이 진행되었다.

2.2 군 도메인에서 Publish Subscribe 모델 사용

현재 다양한 신기술의 신속한 적용과 개발비용 절감에 대한 요구조건으로 SOA, 클라우드 컴퓨팅 등 많은 상용 기술들이 군 도메인에 적용되고 있다. 본 절에서는 Publish Subscribe 모델의 군 도메인 적용 사례에 대해 미 해군을 중심으로 살펴본다.

2.2.1 미 해군 개방형구조(Open Architecture)

과거 미 해군 함정에서 사용되는 전투체계는 상용제품이 아닌 군 독자 표준(MIL-STD)에 따라 구현되어 운용되었다. 그러나 군 독자 표준의 특성상 발전하는 상용 신기술의 신속한 적용이 어렵고 개발비용이 과다하게 소요되었으며 새로운 시스템을 함정에 추가로 탑재 시 기존 다른 시스템과의 상호운용성 문제가 발생하였다. 그리고 해군 함정의 수명기간(통상 20~30년) 동안 동일한 시스템을 사용함에 따라 노후화된 기술에 대한 유지보수 문제가 대두되었다. 이에 미 해군에서는 함정에 탑재되는 시스템의 상호운용성 및 유지보수 능력 향상과 신기술의 신속한 도입을 위해 개방형구조(Open Architecture)에 대한 연구를 진행하였다.



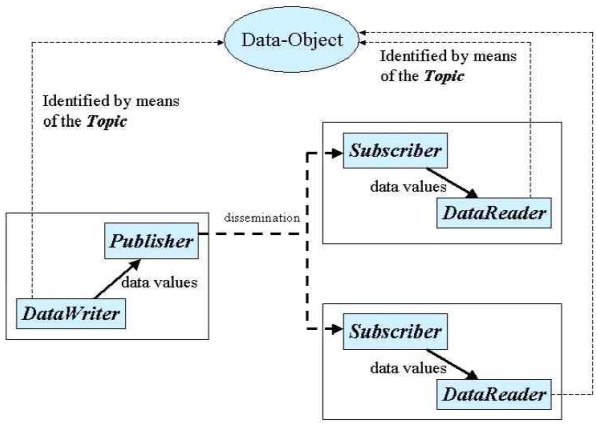
(그림 4) 개방형구조(OA) 계층[4]

미 해군 개방형 구조의 핵심은 군 독자 표준이 아닌 일반 상용표준을 사용하는 것에 있다. (그림 4)는 미 해군 전투체계 개방형구조의 계층을 나타낸 것으로 일반 상용 표준과 제품을 통하여 OACE(Open Architecture Computing Environment)로 불리는 개방형구조 컴퓨팅 환경을 구축하고, OACE 위에서 특정한 애플리케이션 및 서비스를 구축하여 함정 전투체계를 구성한다. OACE의 분산(Distribution) 미들웨어는 OMG의 표준을 따르고 있으며 OACE 기술 및 표준 버전 1.0에서는 분산 미들웨어로 Publish Subscribe 모델을 포함하고 있다[4]. 이와 같이 미 해군의 개방형구조에서는 애플리케이션과 하드웨어를 분리하고 하드웨어 역시 상용 표준기반으로 구성함으로써 미 해군은 앞서 언급한 상호운용성, 유지보수 및 신기술 도입에 대한 문제 해결을 추진 중이며 Publish Subscribe 모델이 데이터 분산에서 중요한 역할을 하는 것을 확인할 수 있다.

2.2.2 DDS(Data Distribution Service)

DDS(Data Distribution Service)는 실시간성이 요구되는 분산 시스템 환경에서 애플리케이션의 통신에 Publish Subscribe 모델을 적용한 미들웨어로써 표준화 기관인 OMG(Object Management Group)에서 2003년 최초 제정했다. 현재 DDS 버전은 DDS 1.2(2007년)[15]이며 DDS 1.3 작업이 진행 중이다. DDS에서는 주제 및 내용 기반 Publish Subscribe 모델 둘 다 지원한다.

(그림 5)는 DDS DCPS(Data-Centric Publish-Subscribe) 오버뷰를 나타낸 것이다. DDS에서는 정보교환을 위해 데이터를 객체 형태(Data-Object)로 교환을 하며 데이터의 송신



(그림 5) DDS DCPS 오버뷰[15]

은 Publisher와 DataWriter 객체가 담당하며 데이터의 수신은 Subscriber와 DataReader 객체가 그 역할을 수행한다. Publisher 객체는 데이터 전파 역할을 담당하며 다양한 데이터 타입을 발행한다. DataWriter 객체는 애플리케이션이 Publisher와 주어진 데이터 객체 타입으로 통신할 수 있도록 한다. 그리고 Subscriber 객체는 발행된 데이터를 수신하여 애플리케이션에 전달하며 애플리케이션은 DataReader 객체를 통해 전달된 데이터에 접근하게 된다.

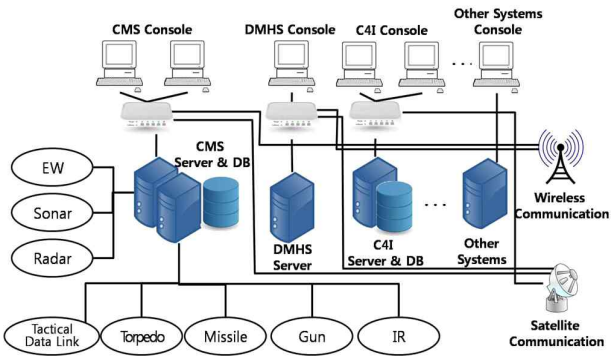
현재 RTI(Real-Time Innovations)사의 NDDS, OCI(Object Computing, Inc)의 OpenDDS, PrismTech의 Open Splice DDS 등 다양한 벤더들이 OMG 표준에 따른 DDS 제품을 생산 중에 있다. 특히 RTI의 DDS 제품은 미국 국방 분야에 많이 도입되어 사용 중이다.

3. Publish Subscribe 모델 기반 정보교환 아키텍처

본 장에서는 본 논문에서 제시하는 Publish Subscribe 모델 기반 정보교환 아키텍처를 제안한다. 제안 아키텍처는 함정 내부의 서로 다른 애플리케이션간의 상호운용성 향상과 효과적인 데이터 교환을 위해 Publish Subscribe 모델을 통신 미들웨어로 사용하여 이벤트 기반의 정보교환이 가능하게 한다.

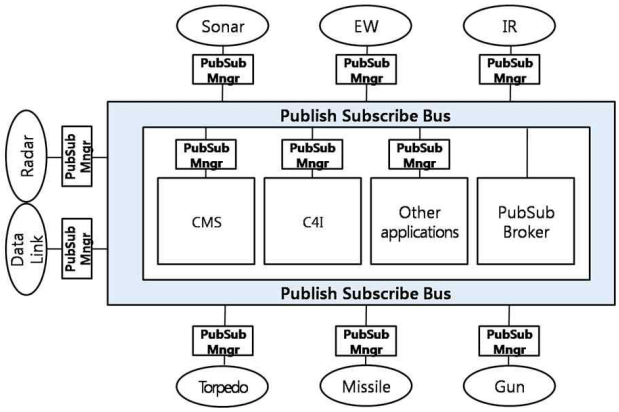
3.1 제안 아키텍처

현재 해군의 함정에는 (그림 6)과 같이 다양한 시스템이 운용 중이며 센서체계(레이더, 소나 등), 타격체계(함포, 유도탄 등), 전투체계 및 C4I체계 등을 포함한 제한된 분산 시스템 형태를 가진다.



(그림 6) 함정 시스템의 시스템 구성

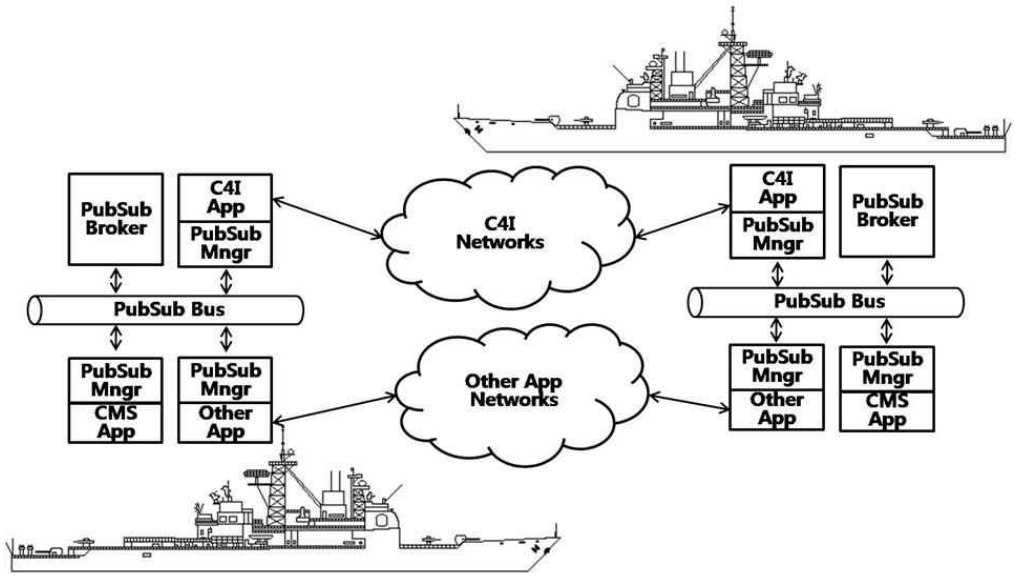
그러나 함정에서 사용되는 체계들이 군 독자 표준(MIL-STD) 장비와 일반 상용(COTS) 장비들로 혼재되어 구성되어 있어서 상호운용성이 제한된다. 그리고 함정의 전투체계 및 일부 센서, 타격체계에서 전술데이터링크[16]를



(그림 7) 제안 정보교환 아키텍처

사용할 경우 전술데이터링크 메시지 변환 등의 추가적인 단계가 요구된다. 또한 현재의 시스템 구성에서 센서나 타격체계 등은 전투체계와 연동이 되며 C4I 및 다른 체계와는 직접적인 연동이 어려워 효과적인 정보교환이 제한된다. 예를 들어 C4I체계의 경우 센서체계의 정보는 전투체계를 거쳐야 함에 따라 정보 교환을 위한 포맷 변경 등이 요구되어 신속한 정보교환이 어렵다. 이는 각 체계가 독립성 및 상호운용성이 제한되어 새로운 체계의 추가 등이 어려운 것을 의미한다.

본 논문에서 제안하는 아키텍처는 여러 애플리케이션이 운용중인 함정에서 상호운용성 향상과 정보교환을 위해 사용이 되며 아키텍처는 (그림 7)과 같다. 제안 아키텍처에서 각 애플리케이션은 Publish Subscribe Bus를 통해 이벤트 형태로 정보를 교환한다. 기존 애플리케이션을 Publish Subscribe Bus와 연결을 위해 PubSub 매니저를 인터페이스로 사용을 하며 PubSub Broker가 Publisher와 Subscriber의 중계 역할을 수행한다. 그리고 PubSub Broker는 다른 함정의 PubSub Broker 들과 PubSub 네트워크를 구성하여 여러



(그림 8) 기존 애플리케이션 네트워크를 통한 정보교환

함정 간에 Publish Subscribe Bus 구현 및 정보교환이 가능하도록 한다.

그리고 제안한 아키텍처는 기존 다른 네트워크와 연결이 이루어진 애플리케이션의 네트워크를 대체하는 것을 의미하지는 않는다. (그림 8)과 같이 기존의 다른 네트워크와 연결된 애플리케이션은 해당 네트워크를 통해 통신이 가능하다.

본 논문에서 제안하는 시스템의 핵심인 PubSub 매니저의 역할은 (그림 9)와 같다. PubSub 매니저는 크게 Publish Subscribe 모듈(PubSub Module)과 전술데이터 링크 메시지 해독 및 변환 모듈(TDL Msg Translate Module) 2개로 구분된다. Publish Subscribe 모듈은 애플리케이션에게 Publisher 및 Subscriber에 대한 인터페이스 및 데이터 변환을 담당한다. 인터페이스 역할은 애플리케이션에서 생산되는 데이터를 Publisher에 전달하며 애플리케이션에서 요구되는 데이터를 Subscriber에게 전달하고 Subscriber로부터 획득한 데이터를 애플리케이션에 전해준다. 데이터 변환은 DataToEvt, EvtToData 및 DataToSub의 세 컴포넌트로 구성이 된다. 먼저 DataToEvt는 애플리케이션에서 발생된 데이터를 Publisher가 이벤트를 발행할 수 있도록 애플리케이션 데이터를 이벤트 형식으로 변환을 하며 EvtToData는 Subscriber로 부터 전달된 이벤트를 애플리케이션 데이터로 변환을 한다. 마지막으로 DataToSub는 애플리케이션에서 요구하는 데이터를 받을 수 있도록 Subscription을 만들어 Subscriber에게 전달한다.

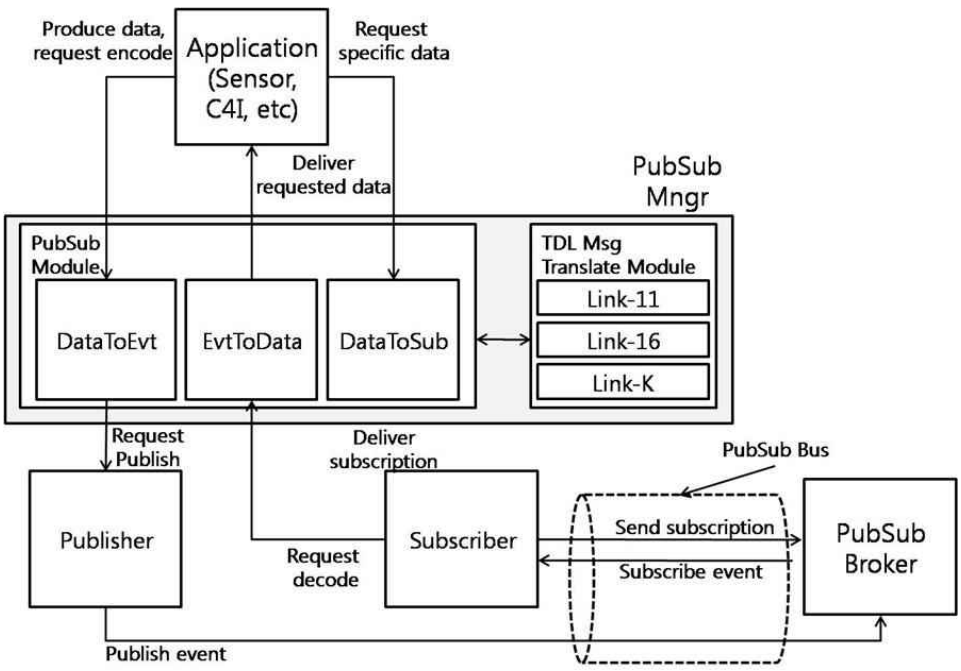
전술데이터 링크 메시지 해독 및 변환 모듈은 Link-11, Link-16 및 Link-K와 같은 전술데이터 링크를 사용하는 어플리케이션에서 발생하는 전술데이터 링크 메시지 데이터를 이벤트 및 Subscription과 변환하는 역할을 수행하여 어플리케이션에서 전술데이터링크 메시지 해독 및 변환 과정을 생략할 수 있도록 한다. 이를 통해 전술데이터링크를 사용하지 않는 어플리케이션과 정보교환이 가능하다.

3.2 제안 아키텍처의 특징

본 논문에서 제시한 아키텍처는 다양한 서로 다른 애플리케이션이 존재하는 함정에서 각 애플리케이션의 효과적인 정보교환을 위해 고안되었으며 다음과 같은 특징을 가진다. 첫째, Publish Subscribe 모델을 사용함에 따라 함정 내에서 다른 애플리케이션간의 효과적인 멀티캐스트 통신이 가능하다. 둘째, 애플리케이션간의 정보교환을 위한 상호운용성의 향상시킬 수 있다. 제안하는 아키텍처에서 Publish Subscribe 모델은 하드웨어나 운영체제 등에서 독립적인 미들웨어 형태를 가진다. 따라서 다양한 하드웨어 및 운영체제에서 운영이 가능하며 상호운용성을 향상할 수 있다. 셋째, 전투체계, 일부 센서 및 타격체계에서 사용되는 전술데이터링크 메시지의 변환을 PubSub 매니저에서 수행함으로써 어플리케이션에서 발생하는 전술 데이터링크 메시지 해독 및 변환 과정을 생략하도록 하였다. 넷째, 제안 아키텍처는 보다 높은 확장성을 가진다. 제안 시스템 구축 후 다른 시스템을 추가나 연동이 보다 쉽게 이루어 질 수 있으며 OMG DDS 표준에 따라 구현 시 보다 높은 확장성을 가질 수 있다.

4. 이벤트 매칭 알고리즘

본 장에서는 Publish Subscribe 모델 기반 정보교환 아키텍처에서 사용된 매칭 알고리즘을 제안한다. 제안 매칭 알고리즘은 Publish Subscribe 모델의 이벤트 매칭에 사용이

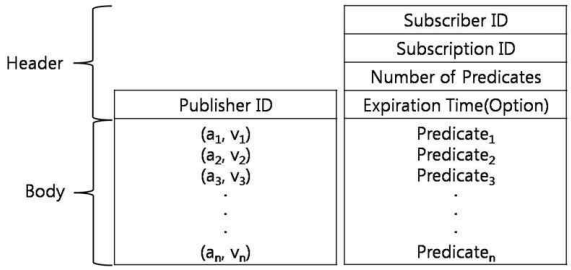


(그림 9) PubSub 매니저 역할

되며 기존 트리매칭 알고리즘에서 각 트리 노드에 추가적인 정보를 저장하여 사용함으로써 기존 알고리즘에 비해 매칭 시간 단축 및 제안 정보교환 아키텍처의 성능 향상이 가능하다.

4.1 이벤트 및 Subscription

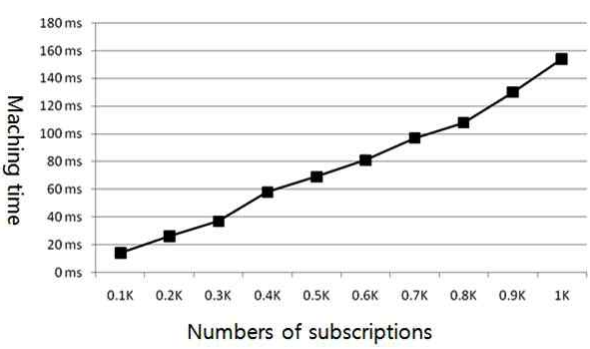
제안 아키텍처에서 애플리케이션의 정보교환을 위해 사용되는 이벤트 및 Subscription의 구조는 (그림 10)과 같다. Publisher가 발행하는 이벤트는 헤더와 바디로 구성되며 헤더는 Publiser의 정보를 가지며 바디는 어트리뷰트(a_n)와 어트리뷰트 값(v_n)의 쌍 들의 집합으로 구성된다. Subscriber의 Subscription 역시 헤더와 바디로 구성이 되며 헤더의 경우 Subscriber 정보 및 Broker의 이벤트 매칭 시 사용하게 될 Number of Predicate 필드와 Broker의 매칭 트리관리를 위한 Expiration Time 필드를 옵션으로 가지며 바디는 Predicate의 집합으로 구성된다.



(그림 10) 이벤트(좌) 및 Subscription(우) 구조

4.2 이벤트 매칭 알고리즘

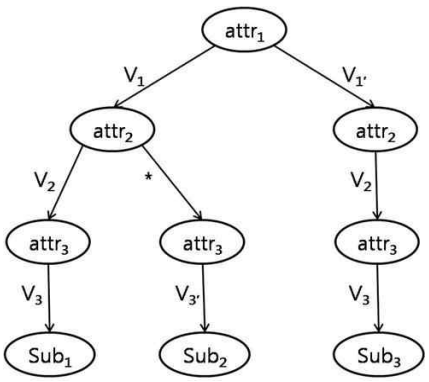
본 논문에서 제안한 아키텍처는 내용 기반의 Publish Subscribe 모델을 적용하였다. 이것은 내용 기반이 주제 기반 Publish Subscribe 모델보다 유연하고 확장성이 높기 때문이다. 내용 기반 모델에서 이벤트의 전달은 Broker가 Subscriber의 Subscription과 Publisher의 이벤트 매칭을 통해 결정이 된다. 기본적이고 간단한 이벤트 매칭방법으로 가지고 있는 Subscription을 이벤트와 차례로 하나씩 비교하는 순차검색을 들 수 있다. 그러나 순차검색은 (그림 11)과 같이 가지고 있는 Subscription이 많아질수록 소요 시간이 늘어나게 된다.



(그림 11) 순차검색을 통한 매칭시간 결과

따라서 효과적인 Broker의 Subscription 데이터 저장 및 매칭을 위해 트리, DHT 및 DBMS 기반 매칭 방법 등이 연구 되었으며 특히 DHT를 이용한 매칭 알고리즘 연구가 활발하다. 그러나 대부분의 DHT를 이용한 매칭 알고리즘은 인터넷 환경에서 많은 노드가 네트워크에 참여하는 분산시스템을 기반으로 고안되었다. 그러나 본 논문에서는 DHT가 아닌 트리 기반의 매칭 알고리즘을 제시한다. 이것은 본 논문의 도메인인 해군 함정이 분산 시스템으로 그 환경이 구성되어 있으나, 그 참여 노드가 상용 환경에 비해 제한되며 전투 임무를 수행하는 함정의 특성상 적의 공격에 의한 시스템의 손상, 파괴 및 네트워크 단절로 인한 DHT 관리가 어려울 수 있기 때문이다. 따라서 해군 함정 환경 고려 시 DHT 보다는 비교적 관리가 용이한 매칭 알고리즘이 요구되며 이에 본 논문에서는 내용 기반 Publish Subscribe 매칭 알고리즘을 제시한다.

내용 기반 Publish Subscribe 모델의 트리 매칭은 크게 두 단계로 구분이 된다[5]. 1단계는 Pre-process 단계로 이벤트 매칭에 앞서 Subscriber로부터 받은 Subscription으로 (그림 12)와 같은 매칭 트리를 생성하는 것이며 2단계는 생성된 트리를 기반으로 이벤트를 매칭하는 단계다.



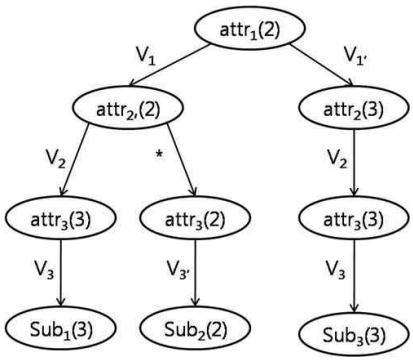
(그림 12) 이벤트 매칭 트리

위 매칭 트리는 Subscription 3개로 생성된 것이다. 여기서 노드($attr_n$)는 Subscription가 가진 Predicate의 어트리뷰트를 의미하며 노드의 edge(V_n, *)는 Predicate의 어트리뷰트 값과 관계연산자를 의미한다. *-edge는 sub2(($attr_1$=v_1)∧($attr_3$=v_3'))와 같이 Predicate에 attr2가 없을 경우 그 값에 상관없이 아래 노드(Sub2의 경우 attr3)로 갈 수 있는 것을 의미한다. 그리고 최하위 노드(sub_1, sub_2, sub_3)가 이벤트를 전달해야할 Subscription이 된다.

Pre-process 단계를 통해 생성된 트리는 Publisher의 이벤트를 매칭하는데 사용한다. 본 논문에서 비교대상으로 선정한 Aguilera의 트리 매칭 알고리즘[5]은 Depth-first order에 따라 최상위 루트 노드에서부터 최하위 노드로 찾아가도록 하였다.

본 논문에서 제안하는 내용 기반 Publish Subscribe 모델의 매칭 알고리즘은 기본적으로 Aguilera 알고리즘을 바탕으로 고안되었다. 제안하는 알고리즘 역시 매칭 전

Subscription으로 트리를 생성하고 트리를 기반으로 매칭을 한다. Aguilera의 알고리즘에 비해 제안하는 알고리즘의 주 차이점은 트리 생성 시 각 Subscription이 가지고 있는 Predicate의 개수정보를 트리 생성 시 각 노드 포함하는 것이다. 이것은 (그림 8)의 Subscription 헤더에 Number of Predicates 필드를 가지고 있어 가능하다. (그림 13)은 본 논문에서 제안하는 매칭트리로 상위 노드는 하위 노드의 Predicate 개수정보(노드의 괄호 안 숫자) 중 가장 작은 값을 가진다. 각 노드가 가지고 있는 Predicate 개수정보는 이벤트 매칭 시 보다 빠른 처리가 가능하게 한다.



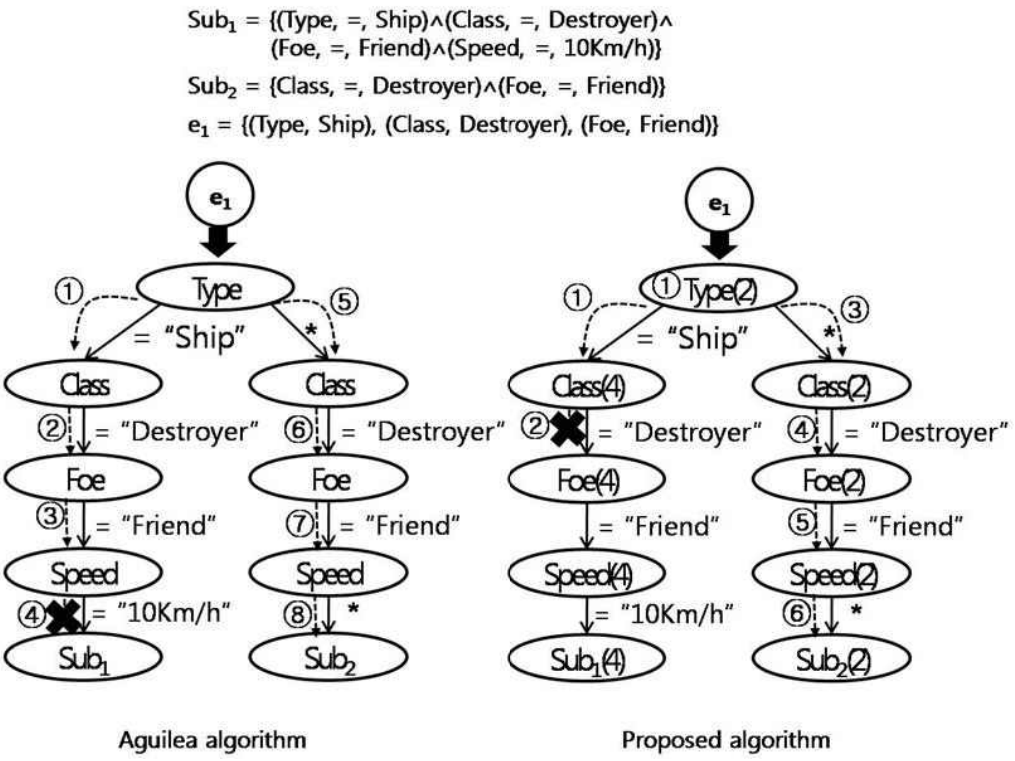
(그림 13) 제안 이벤트 매칭 트리

(그림 14)는 제안하는 매칭 알고리즘 의사 코드로 Depth-first order에 따라 루트 노드에서 최하위 노드로 찾아가도록 하였다. 제안하는 알고리즘에서는 매칭할 이벤트 내의 어트리뷰트 개수와 매칭 트리의 각 노드가 가진 predicate 개수정보(괄호안 숫자)를 비교하여 이벤트의 어트리뷰트 개수가 노드의 predicate 개수정보 보다 같거나 클 때 매칭을 진행하도록 함으로써(라인 6) Aguilera 알고리즘보다 상대적으로 적은 매칭 횟수를 가질 수 있다.

```
1  procedure match(Tree event)
2  visit(Tree, root, event)
3
4  procedure visit(Tree, v, event)
5      if v is a leaf node of Tree then output(v)
6      else if number of pairs in event is bigger than v's
       minimum predicate value
7      then perform test prescribed by v on event
8          if v has an edge e with the result of test
9          then visit(Tree, (child of v at the end point of e in
           Tree), event)
10         if v has a *edge e
11         then visit(Tree,(child of v at the end point of * in
           Tree), event)
```

(그림 14) 제안 매칭 알고리즘 의사 코드

(그림 15)는 3개의 Subscription과 1개의 이벤트에 대한 매칭 순서를 Aguilera 알고리즘과 비교한 것이다. 먼저 Subscription 두 개로 생성한 매칭트리의 모양은 동일하며 제안한 매칭 트리는 각 노드에서 하위 노드의 Predicate 개수정보 중 가장 작은 값을 자신의 개수정보로 가진다. 그림에서 제일하위 노드인 Sub_1, Sub_2는 Predicate 개수정보로



(그림 15) 알고리즘 매칭 순서 비교

(그림 8)에서 설명한 Subscription 헤더에 Number of Predicates 필드 값을 가진다. Aguilera 알고리즘에서 주어진 이벤트 e_1을 매칭은 제일 상위 노드를 기준으로 좌측 노드를 먼저 탐색을 한다. 따라서 ①~③순으로 탐색을 한다. 그러나 e_1이 Speed 어트리뷰트를 가지고 있지 않기 때문에 ④에서 탐색을 중단하고 ⑤ 이후 탐색을 시작, ⑧에서 Sub_2를 찾고 매칭을 종료하게 된다. 제안하는 알고리즘의 트리 탐색은 먼저 이벤트의 어트리뷰트 개수가 트리 노드의 Predicate 개수정보 보다 같거나 클 때 이루어진다. 따라서 ②에서 Class(4) 노드의 Predicate 개수정보 4보다 이벤트의 어트리뷰터 개수정보 3이 적어 탐색을 중단하고 ③ 이후의 탐색을 시작, ⑥에서 Sub_2를 찾고 매칭을 종료한다. 따라서 본 예제에서는 제안하는 알고리즘이 Aguilera 알고리즘에 비해 2회의 트리 노드 탐색 횟수를 줄이는 것을 볼 수 있다. 실제 Publish Subscribe 모델의 매칭에서는 다양한 수의 Subscription과 이벤트가 사용되는 것을 고려 시 제안하는 알고리즘이 트리 노드 방문 횟수를 줄임으로써 기존 Aguilera 알고리즘에 비해 Publish Subscribe 모델의 이벤트 매칭 시간을 감소시킬 수 있다.

5. 성능평가

본 논문에서는 해군 함정 컴퓨팅 체계 정보교환을 위한 Publish Subscribe 모델 기반 아키텍처를 제안하였으며 제안 아키텍처의 성능향상을 위해 개선된 매칭 알고리즘을 제시하였다. 성능평가는 제안 알고리즘의 유효성을 검증하기 위해 수행되었으며 본 성능평가 결과를 통해 제안 아키텍처에서 핵심적인 메시징 시스템의 성능을 확인할 수 있다.

성능평가 대상은 Broker의 이벤트 매칭 알고리즘으로써 비교대상은 Aguilera 알고리즘이다. 본 장에서는 제안 알고리즘과 비교대상 알고리즘을 구현, 시뮬레이션을 통한 결과를 제시한다.

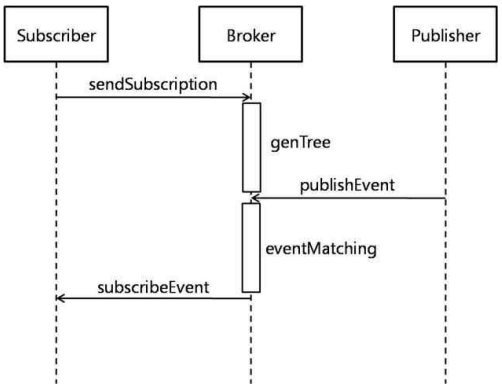
5.1 성능평가 환경

성능평가를 위한 환경은 <표 1>과 같다. 성능평가 시 구현범위는 Publish Subscribe 모델에서 매칭 알고리즘과 이벤트 매칭을 담당하는 Broker의 매칭 트리 생성 기능으로 한정하였다.

〈표 1〉 성능평가 환경

구 분	사 양
CPU	Intel Xeon Quad 1.6GHz X 2
RAM	8GB
OS	Fedora12 64bit
Java version	1.6.0

(그림 16)은 알고리즘 성능평가를 위한 절차를 보여준다. 먼저 Subscriber에서 Subscription을 생성 후 Broker에게 전송하며 Broker는 이를 토대로 매칭 트리를 생성(genTree)



(그림 16) 성능평가 절차

한다. 그 후 Publisher가 이벤트를 발행하고 Broker는 생성된 트리를 이용해 이벤트 매칭(eventMatching) 후 이벤트를 전달하고 종료한다. 성능평가는 Broker의 매칭트리 생성시간 및 매칭 소요 시간을 측정하여 비교하였다.

5.2 성능평가를 위한 데이터 셋

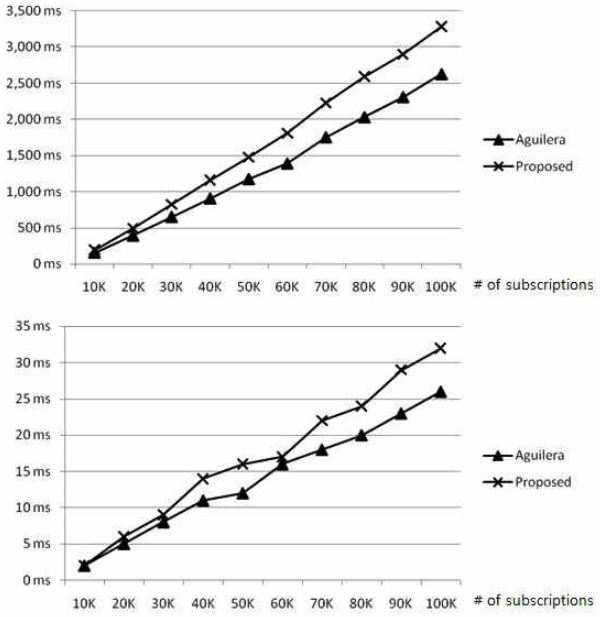
성능평가에 사용한 데이터 셋은 <표 2>와 같다. 본 성능평가에 영향을 미치는 요소는 Subscription과 Subscription이 가진 Predicate의 개수, 어트리뷰트 및 어트리뷰트 값이다. 이에 성능평가 시 Subscription의 개수 변화에 따른 성능평가를 실시하였으며 각 Subscription내 Predicate 개수는 2-10개의 범위 내에서 랜덤 값을 가지도록 하였다. 이벤트 역시 2-10개의 범위 내에서 어트리뷰트와 어트리뷰트 값을 가지도록 하였으며 매칭 시간 오차를 줄이기 위해 이벤트는 500개 단위로 매칭을 실시하였다. 위 데이터 셋은 과거 다른 연구에서 Publish Subscribe 모델의 매칭 알고리즘의 성능을 평가하기 위해 사용한 데이터 셋의 평균적인 값이다 [17][18].

〈표 2〉 데이터 셋

구 분	범 위
Total number of subscriptions	100 - 100,000
Number of predicates per subscription	2 - 10
Number of pairs per event	2 - 10
Number of attribute	10
Number of value	100
Relation operator	<, >, =
Number of event at once	500

5.3 성능평가 결과

먼저 이벤트 매칭을 위한 트리 생성 시간 비교 결과는 (그림 17(상))과 같다. 제안하는 알고리즘이 Aguilera 알고리즘에 비해 시간이 더 소요 되었다. 그리고 (그림 17(하))는 생성된 매칭 트리에서 1% 업데이트 발생 시 소요되는 시간을 비교한 것으로 역시 비교대상 알고리즘 보다 높은 결과값을 가졌다.

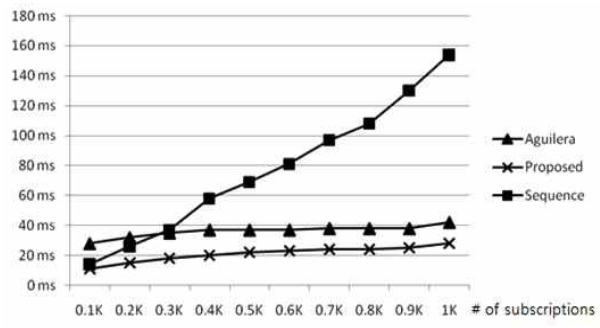


(그림 17) 매칭 트리 생성 시간(상) 및 업데이트 시간(하) 비교

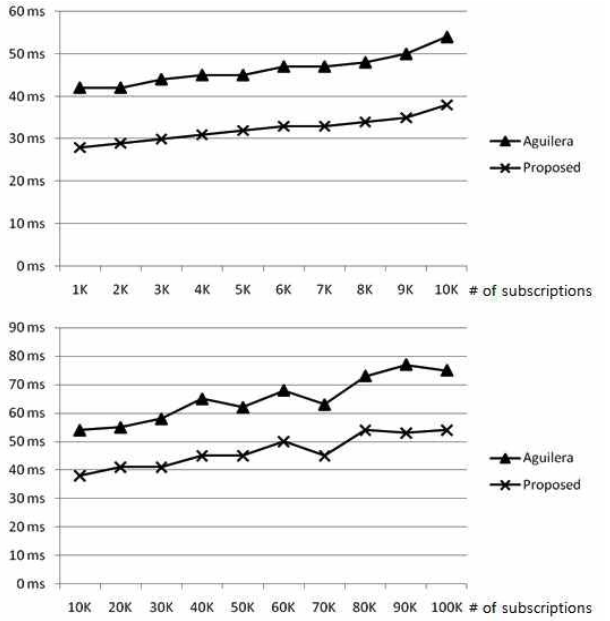
위 결과는 제안하는 알고리즘이 Aguilera 알고리즘과 비교했을 때 매칭 트리 생성 및 업데이트 시 매칭 트리 각 노드에서 Predicate 정보를 반영하는 과정이 추가되었기 때문이다. 비록 트리 생성 시간에 있어 제안 알고리즘이 다소 시간이 더 소요되나 Publish Subscribe 모델에서 트리는 최초 1회 생성 후 지속적으로 사용하는 것을 고려 시 그 영향은 작다고 할 수 있다. 그리고 업데이트 시간은 트리 생성 시간의 100분의 1 이하의 비교적 작은 오버헤드를 가진다. 따라서 본 논문에서 제안하는 군 함정 컴퓨팅 체계 정보교환을 위한 Publish Subscribe 모델 기반 아키텍처의 전체 성능에 큰 영향을 미치지 않는다.

(그림 18)은 순차검색과 매칭 트리 알고리즘(제안 및 Aguilera 알고리즘)의 매칭시간 결과를 비교한 것이다. 비교결과 매칭 트리 알고리즘이 순차검색에 비해 매칭 시간이 줄어드는 것을 볼 수 있으며 Subscription의 개수가 늘어날수록 그 차이는 커지게 된다.

(그림 19)는 제안 알고리즘과 Aguilera 알고리즘의 매칭 시간을 비교한 것으로 제안 알고리즘이 상대적으로 낮은 매칭 시간을 가지는 것을 볼 수 있다. 실험결과 제안 알고리즘이 Aguilera 대비 최소 26%에서 최대 34% 매칭시간을 단축하는 결과를 가져왔다. 이것은 제안하는 알고리즘 매칭 트리의 각 노드가 Subscription의 Predicate 정보를 가지고 있어 이벤트내의 어트리뷰트 개수와 비교하여 이벤트의 어트리뷰트 개수가 노드의 Predicate 개수정보 보다 같거나 클 때 매칭을 진행하여 불필요한 트리 탐색을 줄인 결과이다. 그리고 어트리뷰트 및 어트리뷰트의 범위가 늘어나면 그 차이는 더 커지게 된다.



(그림 18) 순차검색과 매칭 트리 알고리즘 매칭 시간 비교



(그림 19) 제안 알고리즘과 Aguilera 알고리즘 매칭 시간 비교

제안하는 알고리즘의 매칭 트리 생성 시간, 업데이트 시간 및 매칭 시간 비교 결과, 제안하는 알고리즘이 Aguilera 알고리즘에 비해 트리 생성 및 업데이트에 있어 다소 많은 시간이 소요되었다. 그러나 트리 생성 시간 및 업데이트 시간은 본 논문에서 제안하는 아키텍처의 전체 성능에 큰 영향을 미치지 않는다. 그리고 제안하는 아키텍처에서 매칭 트리는 최소 1회 생성 이후에는 계속 사용되나 이벤트 매칭은 지속적으로 발생한다. 따라서 Aguilera 알고리즘에 비해 최소 26%에서 최대 34% 매칭 시간 감소 결과를 가지는 개선된 매칭 알고리즘은 제안한 아키텍처의 성능을 향상 시킬 수 있다.

6. 결 론

현대전에 있어서 정보 우위는 신속한 전장상황 파악, 신속한 의사결정을 통해 전투의 승패를 결정하는 중요한 요소이다. 현재 해군 함정에는 다양한 체계들이 연결되어 복합 시스템(System of System) 모습을 가지고 있어 정보우위를 위해서는 여러 체계 간의 상호운용성 보장 및 효과적인 통신방법이 요구된다. 그러나 함정에서 사용되는 체계들이 군

독자표준 장비와 상용 장비들이 혼재되어 상호운용성 및 체계간의 정보교환이 제한된다.

본 논문에서는 Publish Subscribe 모델 기반의 정보교환 아키텍처를 제안했다. 제안 아키텍처는 여러 체계들이 Publish Subscribe Bus를 통해 이벤트 형식으로 정보 교환이 가능해 보다 효과적인 멀티캐스트 통신이 가능하며 상호운용성을 향상시킬 수 있고 보다 높은 확장성을 가질 수 있다.

그러나 Publish Subscribe 모델에서 Subscription의 증가에 따라 이벤트 매칭을 수행하는 Broker에 부하가 가중될 수 있으며 매칭 시간이 지체되며 결국 Publish Subscribe 모델의 전체 성능에 영향을 미칠 수 있다. 이에 본 논문에서는 트리 기반의 개선된 매칭 알고리즘을 제안하였다. 제안 알고리즘은 매칭 트리 생성 시 Predicate 정보를 각 트리노드에 저장함으로써 매칭 시간을 단축할 수 있도록 하였다.

제안 알고리즘을 비교대상 알고리즘과 성능평가 결과 매칭 트리 생성 및 업데이트 시간에 있어서는 비교대상 알고리즘에 비해 추가적인 시간 및 사용량을 보였으나 매칭 시간에 있어서는 비교대상 알고리즘에 비해 최소 26%에서 최대 34% 매칭 시간을 감소하는 결과를 가져왔다.

따라서 본 논문에서 제안하는 정보교환 아키텍처 및 매칭 알고리즘은 해군 함정과 같이 제한된 분산 시스템 환경에서 효과적인 정보교환에 유용할 것으로 판단한다.

참 고 문 헌

[1] P.T. Eugster, P. Felber, R. Guerraoui, A. Kermarrec, "The Many Face of Publish/Subscribe," ACM Computing Surveys, Vol.35, No.2, pp.114-131, June, 2003.

[2] "NESI (Net-Centric Enterprise Solutions for Interoperability) v3.1," http://nesipublic.spawar.navy.mil

[3] M. Swick, J. White, M. Masters, "A summary of communication middleware requirements for advanced shipboard computing systems," In Proceedings of the IEEE RTAS, 1998.

[4] NSWCDD, "Open Architecture (OA) Computing Environment Technologies and Standards Version 1.0," Aug., 2004.

[5] M. K. Aguilera, R. E. Strom, D. C. Sturman, M. Astley, T. D. Chandra, "Matching Events in a Content-based Subscription System," In Proceedings of the Eighteenth ACM Symposium on Principles of Distributed Computing (PODC'99), pp.53-61, May, 1999.

[6] Y. Liu, B. Plale, "Survey of Publish/Subscribe Event Systems," Indiana University Computer Science Technical Report TR-574, 2003.

[7] P.T. Eugster, R. Guerraoui, J. Sventek, "Type-Based Publish/Subscribe," EPFL Technical report, 2000.

[8] A. Rowstron, A. Kermarrec, M. Castro, P. Druschel, "SCRIBE: The design of a large-scale event notification infrastructure," In Networked Group Communication, pp.30 - 43, 2001.

[9] S.Q. Zhuang, B.Y. Zhao, A.D. Joseph, R.H. Katz, J.D. Kubiatowicz, "Bayeux: An Architecture for Scalable and Fault-tolerant Wide-area Data Dissemination," 2001.

[10] A. Carzaniga, D. Rosenblum, A.L. Wolf, "Design and Evaluation of a Wide-Area Event Notification Service," ACM Transactions on Computer Systems, Vol.19, No.3, pp. 332 - 383, Aug., 2001.

[11] G. Fox, S. Pallickara, "An event service to support grid computational environments," Concurrency and Computation: Practice and Experience, Vol.14, 2002.

[12] P. Triantafillou, I. Aekaterinidis, "Content-based Publish-Subscribe Over Structured P2P Networks," In DEBS, 2004.

[13] Y. Singh, V. Nagar, D.C. Dhubkaryal, "A Multicast Protocol For Content-Based Publish-Subscribe Models," The Global Journal of Computer Science and Technology (GJCST), Vol.10, pp.6-15, Oct., 2010.

[14] G. Ashayer, H. Leung, H.-A. Jacobsen, "Predicate Matching and Subscription Matching in Publish/Subscribe Models," In DEBS, 2002.

[15] OMG, "Data Distribution Service for Real-time Systems Version 1.2," Jan., 2007.

[16] H. Kim, S. Oh, "Interoperable XML Messaging System for Tactical Data Link," 한국컴퓨터정보학회논문지, 제16권, 3호, pp.75-87, 2010.

[17] F. Fabret, A. Jacobsen, F. Llirbat, J. Pereira, K. Ross, D. Shasha, "Filtering algorithms and implementation for very fast publish/subscribe," In Proceedings of the 20th Intl. Conference on Management of Data (SIGMOD 2001), 2001.

[18] S. Bittner, A. Hinze, "On the benets of non-canonical filtering in publish/subscribe systems," In Proceedings of the International Workshop on Distributed Event-Based Systems (ICDCS/DEBS''05), 2005.

김 홍 재

e-mail : carrotbox@ajou.ac.kr

2003년 해군사관학교 전산과학과(학사)

2009년~현 재 아주대학교 NCW학과 통합과정

관심분야 : 미들웨어, 클라우드컴퓨팅, 가상화

오 상 윤

e-mail : syoh@ajou.ac.kr

2006년 미 인디애나대 Computer Science 박사

2006년~2007년 SK텔레콤

2008년~현 재 아주대학교 정보통신대학 정보 및 컴퓨터공학과 조교수

관심분야 : 웹/분산 시스템, SOA, Ad-hoc/P2P 시스템, 클라우드 컴퓨팅