

# 대형실시간분산시스템을 위한 하이브리드 경로탐색기법에 관한 연구

A Study on Hybrid Discovery Mechanism for Large-Scale Realtime  
Distributed Systems

정 용 환\*

Jung, Yong-Hwan

## ABSTRACT

Naval combat management system is a kind of complex weapon systems performing naval ship's warfare mission by integrating various types of system including sensor systems, weapon systems and many kinds of communication system. Recently, there are many studies to apply commercial high-level communication middleware to naval combat system development to integrate and interoperate various kinds of heterogeneous on-board systems efficiently. An initial discovery mechanism is required for application modules to start communication with their relevant modules to apply middleware technology in developing naval combat system, which is characterized by large-scale, complex and real-time system. This paper suggests a fast hybrid discovery mechanism which combines static and dynamic discovery mechanism and confirms its adaptability by an experiment on testing environment.

함정 전투체계는 함정에 탑재되는 다양한 종류의 센서와 무장 및 각종 통신체계들을 통합하여 함정 고유의 작전임무를 수행하는 복합 무기체계로, 급속하게 발전하는 상용 기술을 접목하여 분산형 대형 컴퓨팅 시스템의 특성을 보유하고 있다. 최근에는 함정에 탑재되는 이질적 특성의 다양한 탑재체계들을 효율적으로 통합하고 상호 운용하기 위해 상용 기술인 고수준의 통신 미들웨어를 함정 전투체계 개발에 적용하기 위한 활발한 연구가 수행되고 있다. 본 논문은 대형 복합 실시간 시스템의 특성을 보이는 함정 전투체계 시스템에 통신 미들웨어를 적용하기 위해 선행되어야 하는 다수의 노드에서 많은 수의 전투체계 응용프로그램 개체가 상호 운용을 위해 필요로 하는 초기의 상호 경로 탐색 기법에 대해 연구하여 정적 기법과 동적 기법을 결합한 고속의 하이브리드 경로탐색 기법을 제시하고, 실험을 통해 체계 적용성을 확인한다.

주요기술용어(주제어) : DDS(데이터분배서비스), Publish/Subscribe(배포/가입통신방식), Discovery(경로탐색)

## 1. 머리말

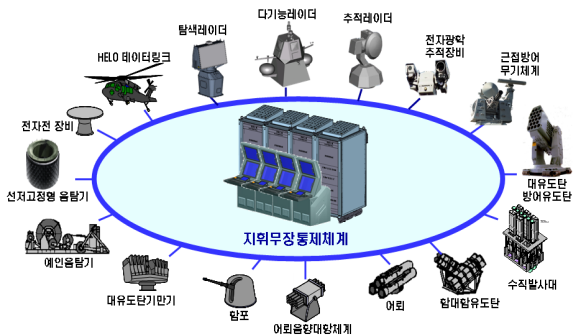
† 2008년 5월 6일 접수~2008년 6월 20일 게재승인

\* 국방과학연구소(ADD)

주저자 이메일 : jungyh06@add.re.kr

함정용 전투체계는 함정에 탑재된 모든 탐지체계에서 탐지 및 수집한 표적정보를 처리하여 지휘관에게 지휘결심을 위한 정보를 제공하고 함정 위협표적에 대해 교전을 할 수 있는 정보를 제공한다. 함정 전투체계는 함정의 전술 운용을 위한 중추적인 역할을 담

당하는 핵심장비이며 함정의 전투력을 결정하는 매우 중요한 장비이다. 또한, 함정 전투체계는 체계를 구성하는 다양한 장비들을 통합하여 함정의 생명주기와 함께 오랜 기간 동안 운용되어야 하는 소프트웨어 중심의 체계이다. 그림 1은 지휘무장통제체계를 중심으로 함정의 규모 및 고유의 작전 임무 유형에 따라 탑재 장비가 연동 및 통합되어 운용되는 일반적인 함정 전투체계의 구성을 보여준다.



[그림 1] 일반적인 함정전투체계 구성

이러한, 함정 전투체계는 최근의 하드웨어 및 소프트웨어적 컴퓨팅 기술과 군사용 특성 등을 고려할 때 다음과 같은 주요 기술적 특징들을 보유하고 있다<sup>[1]</sup>.

- 복합/대형 시스템 : 다양한 종류의 탐지/추적 시스템과 방어용 무기체계 및 지휘통제체계로 구성된 대형/복합 시스템으로 체계 연동/통합 및 상호 운용 능력 확보의 중요성이 높은 체계
- 실시간 시스템 : 급변하는 상용 기술 발전의 체계 적용 확대를 위해, 고속, 고기동 표적에 대한 생존 능력 보장을 위한 고속 실시간 처리 능력 확보가 중요한 체계
- 전력화용 시스템 : 최소 20년 이상의 장기 운용을 위한 기술발전 능력, 운용 안정성 및 신뢰성 확보가 필수적인 체계
- 기술보호 시스템 : 군사용 무기체계로 선진국간 기술 유출 방지를 위한 기술보호가 극심한 체계

이와 같이, 함정 전투체계는 함정 고유의 작전임무 수행을 지원하는 함정 전투력을 결정하는 핵심체계

로, 함정에 탑재되는 센서들과 무장 및 함의 통신장비들을 포함한 다양한 탑재 시스템들을 통합 관리 및 장기간 운용해야하는 실시간 대형 복합체계(Large-Scale Real Time Complex System)의 특성을 보유하고 있는 것으로 규정할 수 있다.

국내 함정 전투체계의 개발 및 운용 현황을 살펴보면, 90년대 말까지 기술도입생산이나 직도입을 통해 체계를 획득하여 운용해 왔기 때문에 국내 함정 전투체계는 운용체계의 기반 환경이 해외업체가 개발한 구조에 종속될 수밖에 없었다. 더욱이 기술도입생산을 하더라도 체계 기반이 되는 기술들은 해외업체에서 기술이전을 기피하여 왔기 때문에 사실상 기술적으로 접근이 불가능한 상태였으며, 이로 인해 신규 무기체계를 연동하거나 체계 기반환경에 대한 정비 및 유지 보수는 전적으로 해외업체에 의존할 수밖에 없었다. 다행히 2000년대 초부터 전투체계 획득방법이 국내 개발로 전환되면서 전투체계의 기본적인 구조를 우리 손으로 선택할 수 있는 가능성이 주어졌고 급속하게 발전하는 높은 성능의 상용 기술들을 전투체계에 적용하여 연구 개발을 추진할 수 있게 되었다.

앞서 언급한 바와 같이 함정 전투체계는 다양한 종류의 함정 탑재 체계들을 연동하여 통합 운용해야 한다. 이렇게 많은 수의 업체에서 자사 고유의 컴퓨팅 기술들을 기반으로 개발된 다양한 탑재 체계들을 통합하여 함정 고유의 작전 임무를 처리하기 위해서는 연동 체계들의 하드웨어 및 소프트웨어적인 다양성을 극복할 수 있는 통합 기반 기술이 필수적이다. 최근 상용의 국제 표준화 그룹에서는 다양한 종류의 컴퓨팅 기술들에 대한 표준화 작업을 진행하고 있어, 함정 전투체계 개발에 있어 상용 기술의 적용 확대를 촉진하고 있으며, 이 중 하드웨어 플랫폼, 운용체제, 소프트웨어 개발 언어 및 네트워크 통신의 이질성을 극복하고 상호 통신 및 상위 수준의 통신 능력을 제공할 수 있는 체계기반통신기술의 일종인 통신 미들웨어 기술은 함정 전투체계 시스템의 이질적 특성을 극복할 수 있는 좋은 기술적 대안이 된다.

국내에서 2000년대 초부터 개발되고 있는 소형 함정용 전투체계는 그 동안의 해외 기술 도입 방식에서 탈피하여 국내 독자 기술로 개발되고 있으며, 고유의 표준화 및 개방형 아키텍처<sup>[2]</sup>를 기반으로 발전된 상

용 기술을 적극적으로 활용하고 있다. 이 소형 합정용 전투체계는 합정 탑재 체계들의 다양성을 극복하고 체계들 간의 안정적인 상호운용성 확보를 위해 상용 기술들에 대한 방대한 분석과 성능 평가 및 선진국 발전 동향 등을 종합하여 데이터분배서비스(DDS : Data Distribution Service) 미들웨어를 체계 기반 통신 기술로 선정<sup>[3~5]</sup>하여 체계개발에 적용<sup>[6]</sup>하고 있다.

최근 국내 기술로 개발되고 있는 중형 체계는 기존 소형 체계에 비해 작전 임무 수행 능력 면에서 복잡해지고, 체계를 구성하는 부체계 시스템의 규모 면에서 약 2배, 송수신 메시지의 종류 면에서 약 3.5배 정도로 크게 증가하였다. 다음 표 1은 중형 체계를 구성하는 컴퓨팅 노드와 송수신 개체 및 송수신 메시지의 종류를 소형 체계와 비교 분석한 결과를 보여준다.

[표 1] 전투체계 시스템 규모 특성

|       | 노드수  | 송신 개체수  | 수신 개체수   | 송수신 메시지수 |
|-------|------|---------|----------|----------|
| 소형 체계 | 20여개 | 3,000여개 | 3,000여개  | 6,000여개  |
| 중형 체계 | 40여개 | 7,000여개 | 14,000여개 | 21,000여개 |

이와 같이 합정 전투체계는 일반적인 상용 컴퓨팅 도메인에 비해 훨씬 복잡하고 많은 수의 장비들이 통합 운용되는 대형 시스템 환경으로 볼 수 있으며, 이러한 복잡한 환경에서 특정한 메시지를 보내는 특정 소프트웨어 송신 개체가 해당 메시지를 수신하는 수신 개체를 실시간적으로 탐색하는 것은 일반적인 동적 경로 탐색 기법으로는 수용하기 힘든 요구사항이며, 기존 체계에 비해 크게 증가한 중형급 이상의 전투체계 개발을 위해서는 소프트웨어 송수신 개체간의 고속 경로 탐색 방안이 연구되어야 한다.

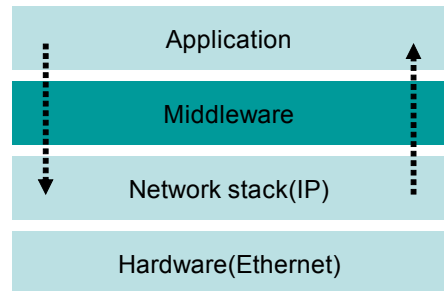
본 논문은 중형급 합정용 전투체계 개발을 위한 기술적 과제들을 분석하고, 기존 체계에 비해 규모면에서 증가된 탑재 및 연동체계들을 통합 운용하기 위해 필요로 하는 체계 구성 소프트웨어 개체들 간의 초기 통신 채널 형성을 원활히 수행하기 위해 기존 체계에서 적용된 일반적인 동적 경로 탐색 기법에서 나타나

는 탐색 지연 시간 증가, 메모리 및 처리 장치 소모율 증가 등의 불확실성을 보완할 수 있는 하이브리드 경로 탐색 기법을 제안한다. 또한, 실 체계와 유사한 시험환경을 구축하여 기술적으로 성능을 평가하여 체계 적용 가능성을 확인한다.

## 2. 데이터분배서비스 및 경로탐색기법

### 가. 데이터분배 서비스

미들웨어는 응용프로그램과 운영체제 사이에 위치하는 소프트웨어 계층이다. 통신 미들웨어는 컴퓨터 구조, 운영체제 및 네트워크 등의 기반 계층(Stack)의 세부 규격으로 부터 응용프로그램을 분리시킨다. 통신 미들웨어는 그림 2와 같이 소켓(Socket)이나 TCP 또는 UDP/IP와 같은 낮은 단계의 프로토콜을 사용하여 프로그램할 필요가 없이 응용프로그램이 정보를 송수신 할 수 있도록 하여 분산 시스템의 개발을 단순화시킨다.



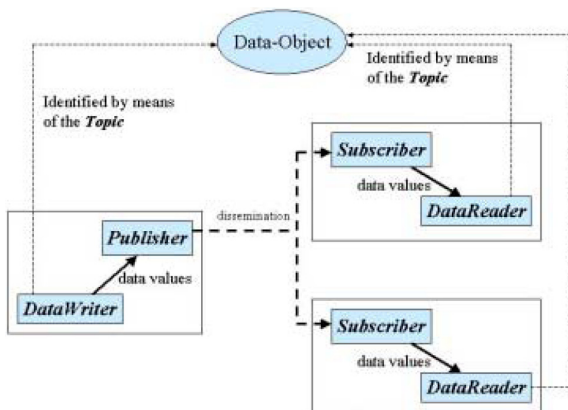
[그림 2] 통신 미들웨어

데이터분배서비스(DDS) 미들웨어는 하드웨어 플랫폼과 네트워크 통신 및 소프트웨어적 이질성을 극복하고 실시간 내장형 시스템을 위한 배포/가입(Publish/Subscribe) 방식의 미들웨어로, 전세계 800여개 이상의 주요 컴퓨팅 업체들이 구성한 국제 표준화 그룹인 객체관리그룹(OMG : Object Management Group)에서 표준화한 미들웨어 표준<sup>[7]</sup>으로, 다음과 같은 특징을 가진다.

- 배포/가입 방식의 간결한 통신 지원

- 종말 송수신 개체 간의 동적 경로탐색
- 예측 가능한(Deterministic) 실시간 데이터 전송
- 통신 서비스 규모의 동적 확장 및 축소 가능
- 일대일(One-To-One), 일대다(One-To-Many), 다대일(Many-To-One), 다대다(Many-To-Many) 통신 지원
- 메시지 특성에 따른 다양한 통신품질 제어 파라미터(QoS) 제공

그림 3은 데이터를 전송하는 측의 배포자(Publisher)와 송신개체(DataWriter), 데이터를 수신하는 측의 가입자(Subscriber)와 수신개체(DataReader)로 구성되는 데이터분배서비스의 각 개체(Entity) 간의 정보흐름을 보여준다.



[그림 3] DDS 구성 개체들의 관계

응용 프로그램들 간의 메시지 교환을 위해서는 메시지를 송신하는 측의 배포자 및 송신개체와 메시지를 수신하는 측에서의 가입자 및 수신개체가 필요하다.

배포자는 데이터 분배를 담당하는 개체로서, 각 배포자는 토픽(Topic)으로 구분되는 여러 종류의 데이터를 배포할 수 있으며, 토픽의 종류별로 여러 개의 송신개체를 가질 수 있다. 송신개체는 실제 메시지를 배포하기 위한 응용프로그램 인터페이스를 제공하는 개체로서 응용프로그램에서 요구하는 특정 데이터의 값을 배포자에게 전달하는 역할을 수행한다. 배포하고자 하는 데이터가 배포자에게 전해지면, 배포자는 자신에게 설정된 통신 품질 제어 파라미터(QoS)에

따라 연관된 가입자에게 데이터를 배포한다.

가입자는 배포된 데이터를 수신하는 개체로서, 가입자의 QoS 설정에 따라 각 배포자들로부터 배포되는 데이터들을 수신한다. 특정 토픽으로 지정된 데이터를 응용프로그램이 접근하기 위해서는 해당 토픽에 대한 수신개체를 생성하여 가입자에 등록하여야 한다. 하나의 가입자는 다수의 수신개체를 가질 수 있으며, 이를 통해 토픽과 데이터타입으로 구분되는 여러 종류의 메시지를 수신할 수 있다.

토픽은 배포자와 가입자들간의 송수신 데이터를 구분하기 위한 개념적인 식별자이다. 토픽은 배포자와 가입자간의 메시지 교환시 메시지의 종류를 식별하기 위한 유일한 이름으로서, 토픽에는 메시지 이름 외에도 메시지에 대한 데이터의 타입과 QoS가 설정된다. 배포자와 가입자는 토픽을 통해 송수신 메시지에 대한 종류를 식별하며, 동일 토픽을 보유한 배포자와 가입자만이 메시지의 송수신이 가능하다.

도메인(Domain)은 서로 통신을 수행하는 응용프로그램 집합의 개념으로서, 한 시스템에 논리적으로 여러개의 분리된 통신 그룹을 구현할 수 있게 한다. 도메인 참여개체(DomainParticipant)는 시스템 내에 설정된 여러개의 도메인들 중에서 하나의 특정 도메인에 참여하기 위한 도메인 참가 개체이며, 도메인 내부에서의 통신에 참여하는 배포자와 가입자 및 송수신 개체들을 보유한다.

#### 나. 경로 탐색

##### 1) 경로탐색의 정의

경로 탐색(Discovery)은 서로 다른 노드에 존재하는 데이터분배서비스 개체들이 서로에 대한 정보를 수집하여 데이터를 송신하는 개체와 수신하는 개체간의 연관 관계를 형성하기 위한 과정이다<sup>[8]</sup>. 이러한 경로 탐색 과정을 통해 각 도메인 참여개체는 동일한 도메인 내에서 활동하고 있는 모든 송신개체와 수신개체에 대한 정보를 저장하는 메타 데이터베이스를 구성하게 되며, 송신개체와 수신개체는 이 메타 데이터베이스 정보를 통해 자신이 통신해야 하는 다른 개체의 경로를 알 수 있게 된다.

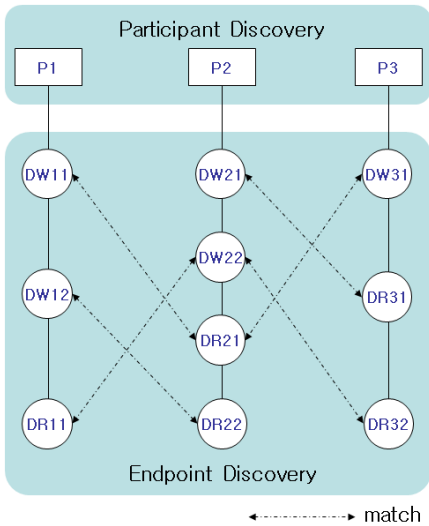
응용프로그램은 이러한 메타 데이터베이스를 생성 및 갱신하기 위해서, 도메인 참여개체들에 대한 탐색

과정인 참여개체 탐색(Participant Discovery) 단계와 도메인 참여개체 내부에 존재하는 송수신 개체에 대한 탐색 과정인 종말개체 탐색(Endpoint Discovery) 단계로 구성되는 두 단계의 경로 탐색 과정을 수행한다. 그림 4는 데이터분배서비스 개체간 경로 탐색 단계를 보여준다.

이러한 두 단계를 거치는 경로탐색의 목적은 각 도메인 참여개체가 시스템 내의 모든 노드들에 존재하는 다른 도메인 참여개체들에 대한 정보와 각 참여개체들 내부에 존재하는 모든 개체들에 대한 상황을 완전히 파악하기 위한 것이다.

도메인 참여개체 탐색(Participant Discovery) 단계는 동일 도메인에 속하는 시스템 내의 모든 참여개체들의 활동 상태에 대한 정보를 수집하는 단계이다.

종말개체 탐색단계(Endpoint Discovery)는 도메인 참여개체에 포함되는 실제 데이터 송수신 개체인 모든 송신개체와 수신개체들 중에서 상호 메시지가 요구되는 특정 송신 개체와 수신 개체간의 연관관계여부를 확인(Matching)하기 위한 탐색 단계이다.



[그림 4] DDS 개체간 경로 탐색

본 논문에서는 도메인 참여개체에 대한 탐색과 송수신개체에 대한 탐색을 수행하는 방법을 경로탐색 기법으로 설정하고 탐색 절차에 따라 동적인 방법과 정적인 방법으로 분류하였다.

## 2) 경로탐색기법

### 가. 동적 경로탐색 기법(Dynamic Discovery)

동적 경로탐색 기법은 데이터분배서비스의 기본적인 경로탐색 기법으로 도메인 참여개체에 대한 탐색 단계와 참여개체 내부에 존재하는 송수신 개체에 대한 종말개체 탐색단계를 동적으로 수행하는 방법이다.

도메인 참여개체들 간의 활동 정보에 대한 동적 경로 탐색은 다음과 같은 과정을 통해 이루어진다.

응용프로그램이 실행되면 프로그램 내에 존재하는 도메인 참여개체에 대한 세부 정보는 동일 도메인 내에 속하는 모든 다른 도메인 참여개체들에게 전송된다. 도메인 참여개체에 대한 세부 정보에는 고유 식별자 정보(GUID : Globally Unique ID), 송수신 포트 정보(주소 및 포트 번호), QoS 등이 포함된다.

각 도메인 참여개체들은 이러한 참여개체 세부 정보에 대한 주기적 교환을 통해 상호간의 활동 상태를 유지 및 관리하며, 상대 참여개체의 생성, 변경 및 삭제 정보를 획득한다.

송수신 개체에 대한 종말개체 탐색 단계는 각 도메인 참여개체들에 존재하는 송수신 개체들 간의 세부 정보(GUID, QoS 등)를 교환 및 비교하는 단계로 다음과 같은 과정을 통해 이루어진다.

각 도메인 참여개체는 자신의 내부에 정의된 송수신 개체에 대한 세부정보를 탐색된 모든 도메인 참여개체들에게 전송함으로써 각 참여개체들에 존재하는 송수신 개체에 대한 정보를 교환한다. 도메인 참여개체는 다른 도메인 참여개체에 존재하는 송수신개체 및 수신개체가 발견되면, 자신이 가지고 있는 송수신 개체와 관련이 있는지를 비교한다. 도메인 참여개체들에 존재하는 송수신 개체들 간의 관련성은 각 송수신 개체들에 설정된 토픽 종류와 데이터 타입 및 QoS 정책의 비교를 통해 이루어지며, 송수신 개체들 간의 관련성이 있는 경우 메타 데이터베이스에 저장되어 데이터 송수신시 사용되게 한다. 이러한 과정은 동일 도메인 내에 포함되는 모든 도메인 참여개체들에 존재하는 송수신 개체들이 모두 서로 간의 관련성을 확인할 때까지 수행된다.

모든 참여개체들에 존재하는 송수신 개체들에 대한 탐색이 종료되면 메타 데이터베이스는 완성되며, 이후 탐색 단계는 안정 상태(Steady State)로 전환된다.



안정 상태에서는 도메인 참여개체와 송수신 개체들의 활동 상태를 유지 및 관리하기 위한 주기적인 메시지 교환을 통해 이루어지며, 새로운 참여개체의 발견이나 기존 참여개체의 변경 또는 삭제 등이 발생하면 메타 데이터베이스는 수정이 이루어지게 된다.

그림 5는 동적 경로탐색 기법에서의 도메인 참여개체 탐색 단계와 종말개체 탐색 단계를 수행하기 위한 일반적인 탐색 절차를 시간 기준 내부 메시지 흐름을 통해 보여주며, 세부 사항은 참고문헌<sup>[9]</sup>을 참고한다.

이와 같이 동적 경로탐색 기법은 시스템 내에 존재하는 응용프로그램의 실행 위치나 도메인 참여개체 및 참여개체 내부의 송수신 개체들의 세부 정보들의 변경에 관계없이 동적으로 데이터의 송수신 경로를 설정할 수 있어 유연성과 확장성 측면에서 상당히 효율적이다. 그러나, 전체 시스템 환경에서 개별 도메인 참여개체 및 송수신개체에 생성, 수정, 삭제 등에 대한 정보를 동적으로 감지하고 경로 탐색 정보를 저장하는 메타데이터베이스를 모든 도메인 참여개체들이 생성 및 갱신 등의 유지 관리 작업을 수행하기 위해서는 일정 이상의 시간과 프로세스 자원을 요구하게 된다.

따라서, 함정전투체계와 같이 함정에 탑재되는 다양한 부체계 장비들을 통합하여 실시간적으로 운용해야 하는 중대규모 이상의 컴퓨팅 시스템에 직접 적용하기 위해서는 연동 대상 장비의 변경이나 체계 구성 장비의 다양한 운용 형상 등의 변경에 관계없이 예측 가능한 처리 성능을 보장해야 하며, 개별 구성 요소의 상태 변화에 따라 전체 시스템의 상태를 예측 불가능하게 할 수 있는 동적 경로탐색 기법의 적용성은 개선될 필요가 있다.

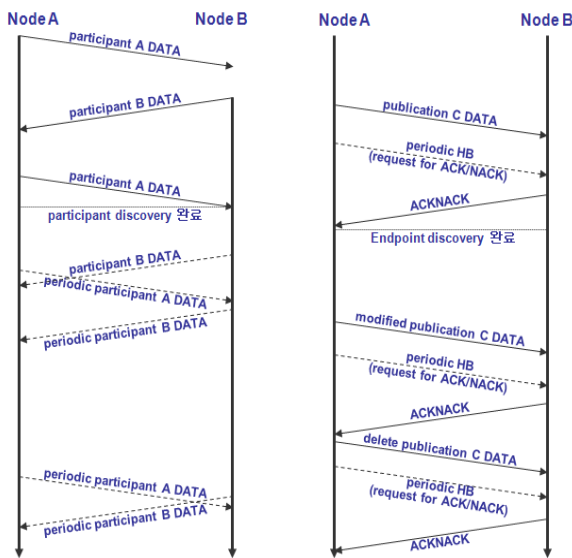
### 나. 정적 경로탐색 기법(Static Discovery)

정적 경로탐색 기법은 시스템 내에 존재하는 도메인 참여 개체와 송수신 개체들에 대한 모든 정보를 사전에 정의함으로써, 도메인 참여개체에 대한 탐색 단계와 참여개체 내부의 송수신 개체에 대한 탐색 단계를 별도로 수행하지 않는 기법이다.

정적 경로탐색 기법은 통신 네트워크 상의 통신 개체들에 대한 정보를 사전에 정적으로 선언하고, 이러한 통신 경로 정보를 통신 개체 자체의 메모리 공간에 미리 적재(Load)함으로써, 프로그램 시작시 별도의 경로 탐색을 수행할 필요가 없기 때문에 시스템 시작과 동시에 예측 가능한 시스템 안정화 시간을 기할 수 있다.

이러한, 정적 경로 탐색 기법은 전체 시스템에서의 변경이 없어 통신 노드 및 규모의 동적 확장성이 상대적으로 덜 중요한 시스템과 같이 특수한 경우에 시스템을 안정적으로 동작시킬 수 있어 적합한 방식이다.

정적 경로탐색 기법에서는 도메인 참여개체 및 종말개체에 대한 경로 정보를 저장하는 메타데이터베이스는 송수신 개체들 간의 모든 정보들이 사전에 정의되어 각 응용 프로그램에서 별도의 경로 탐색 단계가 수행되지 않는다. 이러한 경로 정보는 파일을 통해 제공되거나 별도의 경로 정보를 제공하는 서버를 통해 제공될 수 있으며, 시스템 내의 개체들에 대한 정보를 사전에 정의하는 것 외에도 시스템 내에서 송수신하는 개체들에 대한 연관성을 미리 파악하여 응용 프로그램 마다 관련되는 원격 개체들에 대한 정보만을 제공함으로써 탐색 단계에서 개별 개체들이 상대 개체들에 대한 상호 연관성을 별도로 비교하지 않게 할 수 있다.



[그림 5] 동적 경로탐색 메시지 흐름

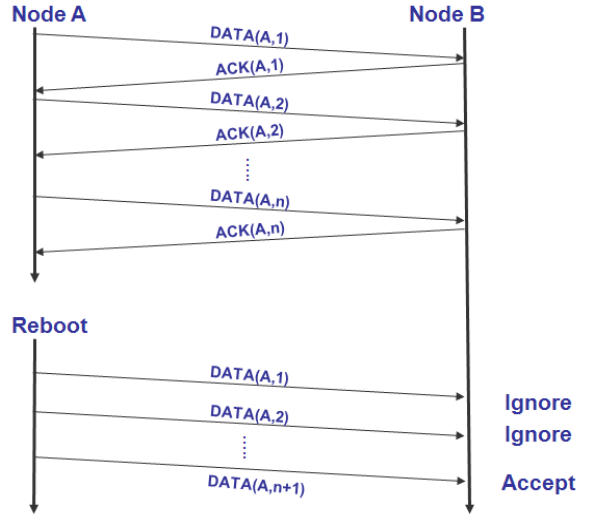
(1) Participant discovery (2) Endpoint discovery

따라서, 정적 탐색기법 적용시 응용프로그램에 상호 연관성이 없는 원격 개체들에 대한 정보를 배제하고 상호 통신이 수행되어야 하는 개체들에 대한 경로 탐색 정보를 사전에 제공함으로써, 경로 탐색을 위한 시간과 경로 탐색을 위해 할당되는 메모리 자원 및 프로세스 자원 등을 최소화할 수 있어 시스템의 예측성과 안정성 측면에서 상당히 효과적인 방법이라 할 수 있다.

그러나, 정적 경로탐색 기법은 시스템 내의 모든 통신 개체들에 대한 정보를 사전에 정의하여 사용함으로써, 시스템 운용중의 구성 요소 변경이나 통신 개체들의 운용 노드 이동과 같은 운용 상태의 동적인 변경 등을 미리 예측할 수 없기 때문에, 시스템의 적은 형상 및 운용 상태 변화에도 상당히 취약하다고 할 수 있다.

또한, 데이터분배 서비스의 송수신 개체간의 통신 특성에 의해 송신개체를 포함하는 도메인 참여 개체의 재부팅 등과 같은 상태 변경 시 데이터 송수신에 영향을 미친다. 이러한 현상은 데이터분배서비스에서의 도메인 참여개체에 대한 고유 식별 번호를 이용한 송수신 개체 구분 방식과 송수신 메시지 식별을 위한 메시지 순차번호(Sequence Number)를 이용하는 특성에 의해 발생한다.

정적 경로탐색기법에서 사전에 정의된 도메인 참여 개체들에 대한 식별 정보는 GUID를 통해 구분되며, 특정 GUID가 할당된 하나의 도메인 참여개체는 타 도메인 참여개체와의 경로 정보 인식을 위해 재부팅 등과 같은 상태 변경에 관계없이 항상 동일한 GUID를 할당받게 된다. 데이터분배서비스에서는 송수신 개체간 메시지의 순차적 전송과 메시지 분실에 대한 재전송 등을 위해 도메인 참여개체별, 데이터 종류별로 개별 데이터 전송시마다 메시지 식별을 위해 순차 번호를 증가시킨다. 데이터분배서비스에서의 수신개체는 특정 송신개체로부터 메시지가 도착시 할당된 순차 번호를 확인하여 해당 메시지에 대한 수신, 무시, 이전 메시지에 대한 분실 확인 및 재전송 요구 등을 수행한다. 특히, 수신 메시지의 순차번호가 해당 송신개체로부터 수신한 이전 메시지의 순차번호보다 오래된 경우 중복된 메시지로 간주하고, 메시지를 수신하지 않는다.



[그림 6] 정적 경로탐색 기법의 데이터전송문제

그림 6은 송신개체를 포함하는 특정 도메인 참여개체가 일부 메시지 전송 후 동일한 GUID로 재부팅된 경우 메시지 전송시 재부팅 전에 전송한 메시지의 순차번호를 유지하지 못하여 순차번호를 처음부터 다시 증가시킬 경우 수신개체에서 정상적으로 메시지를 수신하지 못하는 문제를 보여준다.

이와 같이 정적 경로탐색 기법은 실시간 시스템에서의 예측 가능성을 보장하고 시스템을 안정적으로 운용할 수 있는 장점을 제공하지만, 다수의 구성요소를 보유하는 중대형의 실시간 시스템에서는 구성요소에 대한 변화나 응용 프로그램 개체의 재부팅, 실행 위치 변경 등과 같은 구성요소에 대한 운용 상태 변경에 따라 큰 영향을 받는 유연성 측면에 있어서의 상당한 취약점을 가지고 있다.

### 3. 하이브리드 경로탐색 기법

하이브리드 경로탐색 기법은 경로 탐색 단계별로 동적 경로탐색과 정적 경로탐색 기법을 분리하여 적용함으로써, 동적 경로탐색 기법에서 발생가능한 과도한 자원 소모 및 탐색 시간을 줄이고 정적 경로탐색 기법에서의 응용 프로그램 재부팅에 따른 메시지 손실이나 체계 구성 변경에 따른 유연성 문제 등을 해

결할 수 있는 방법이다.

본 논문에서 제안하는 하이브리드 경로탐색 기법은 그림 4에서 도메인 참여개체간의 경로 탐색은 동적으로 수행하고, 탐색된 도메인 참여개체간의 내부 송수신 개체들에 대한 경로탐색은 사전 정의된 정보를 이용하는 정적인 방법을 적용하는 것이다.

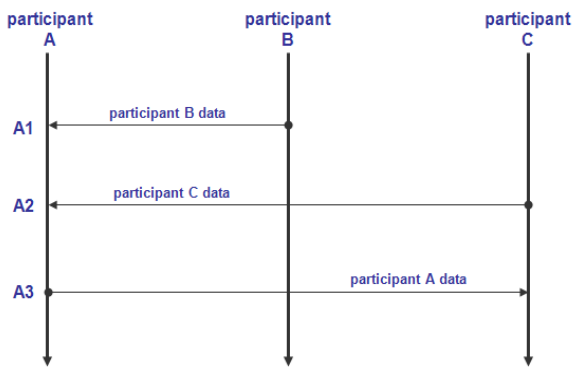
다음은 제안하는 경로탐색기법에 대해서 경로탐색 단계별 세부적인 제안 사항을 보여준다.

### 가. 하이브리드 경로탐색 단계

#### 1) 도메인 참여개체 경로탐색 단계

도메인 참여 개체들의 경로탐색 단계에서는 데이터 분배서비스의 기본적인 경로탐색 기법인 동적 경로탐색 기법을 적용하되 사전 정의된 경로 정보에 각 도메인 참여개체에 대해 관련 있는 다른 도메인 참여개체들에 대한 참여개체 이름만을 기술하여 동적으로 참여개체에 대한 변경 정보를 수집하되 사전 정의된 경로 정보에 포함되지 않는 참여 개체에 대해서는 불필요한 메시지 교환을 수행하지 않는 수정된 동적 참여개체 탐색 방법을 적용한다.

그림 7은 제안 하이브리드 경로탐색기법에서의 도메인 참여 개체에 대한 탐색 과정을 보여준다.



[그림 7] 하이브리드 경로탐색 기법의 도메인 참여개체 탐색 절차

참여개체 A와 C의 경로만 사전 정의된 경우, 그림 7에서 참여개체 B는 참여개체 A의 사전 정의된 경로 정보에 포함되지 않기 때문에, 참여개체 B에 대한 활성화 정보가 참여개체 A에 전달(A1)되어도 참여개체

A에서는 별도의 메시지 교환을 수행하지 않는다. 참여개체 C의 경우 참여개체 A의 사전 정의된 경로 정보에 이름이 포함되어 있기 때문에 참여개체 A는 참여개체 C의 활성화 정보가 전달되면(A2) 경로 정의 정보를 참조하여 참여개체 C의 상호관계를 확인하고 자신의 활성화상태 정보를 참여개체 C로 전송하여(A3) 참여개체 A와 C간의 참여개체 경로 탐색을 완성하게 된다.

이러한 방법은 새로운 도메인 참여 개체에 대한 정보가 동적으로 수신되는 경우 이 참여 개체에 대한 경로 정보가 사전 정의된 경로 정보에 포함된 참여 개체에 대해서만 동적으로 서로의 상태 정보 교환을 수행하고, 사전 정의된 경로 정보에 참여개체의 이름이 포함되지 않는 경우 참여개체들에 대한 상태 정보 교환을 수행하지 않는다. 이를 통해 시스템 내에 존재하는 도메인 참여개체들이 사전 정의된 서로 관련된 개체들만이 상태 정보를 동적으로 교환함으로써, 상호 관련이 없는 개체들 간의 경로 탐색 및 유지 관리를 위한 불필요한 메시지의 송수신을 방지할 수 있다.

사전 정의된 경로 정보에는 참여개체 자신과 관련 있는 다른 참여개체들에 대한 개체 이름을 명시함으로써, 참여개체를 포함하는 응용 프로그램이 재부팅되더라도 항상 새로운 GUID가 할당되기 때문에 이와 통신하던 다른 모든 참여개체들이 재부팅된 참여개체의 상태를 알 수 있게 되며, 이를 통해 프로그램 재부팅으로 인한 메시지 손실을 방지할 수 있다.

#### 2) 종말개체 경로탐색 단계

종말 개체들에 대한 경로탐색 단계에서는 정적 경로 탐색기법에서의 종말 개체들에 대한 탐색 방법을 적용한다.

송수신 개체를 포함하는 도메인 참여개체가 다른 도메인 참여개체로부터 활성화 상태 정보를 수신한 후 상호 연관성이 있는 참여개체인 경우, 자신의 종말 개체와의 직접 통신을 수행해야 하는 개체인지 확인하기 위해 별도의 경로 탐색 정보를 다른 도메인 참여개체에 전송하지 않고, 경로 정보 내부에 저장된 해당 종말 개체의 정보로부터 정적으로 경로 정보를 생성한다. 표 2는 도메인 참여개체(P1)에 대해 사전 정의된 경로 정보에 대한 예를 보여준다.



[표 2] 경로 관계 정보 정의 테이블 예

| 참여개체(P1) | 종말 개체 |      |      |
|----------|-------|------|------|
|          | DW11  | DW12 | DR11 |
| P3       | DR34  | -    | DW32 |
| P5       | -     | DR51 | -    |
| P9       | -     | DR92 | -    |

표 2에서 참여개체 P1이 2개의 송신개체(DW11, DW12)와 1개의 수신개체(DW11)을 보유하고 있으며, 3개의 참여개체(P3, P5, P9)와 연관성이 있음을 보여준다. 참여개체 P1은 초기 구동시 경로탐색정보를 저장하기 위한 메타데이터베이스에 자신과 연관성이 있는 3개의 참여개체들에 대한 공간만을 확보한다. 참여개체 P1이 경로 정보에 사전 정의되지 않은 참여개체 P2로부터 활성 상태정보에 대한 변경 메시지를 수신하더라도 아무런 후속처리를 수행하지 않는다. 만약 참여개체 P1의 경로 정보 테이블에 연관성이 있는 참여개체 P5로부터 활성상태가 전송되면, 참여개체 P1은 자신의 활성상태를 참여개체 P5로 전송하고 자신의 경로정보 테이블로부터 참여개체 P5의 종말개체들과 자신의 종말개체들간의 관계 정보를 확인하여 메타데이터베이스를 완성해 나간다.

시스템 내에서 모든 종말 개체들은 항상 1개의 도메인 참여개체에 포함되며, 운영체제 및 응용프로그램 구조에 따라 응용프로그램은 1개에서 수십 개의 도메인 참여개체들로 구성된다. 시스템 특성에 따라 차이는 있으나 종말개체들의 수가 도메인 참여개체들의 수보다 항상 수배 내지 수십배 이상 많기 때문에, 종말개체들에 대한 정적인 탐색만으로도 필요한 시스템 자원의 소모량이나 경로탐색 시간을 크게 줄일 수 있다.

나. 하이브리드 경로탐색기법 분석

1) 경로탐색 시간 복잡성(Complexity) 분석

시스템 내에 존재하는 모든 참여개체의 수를 N, 각 참여 개체에 존재하는 종말개체의 수를 M이라고 가정한다.

정적 경로탐색 기법의 경우 모든 참여개체 및 종말

개체에 대한 탐색 정보들이 사전 정의되므로, 모든 응용 프로그램들은 구동시 연관되는 다른 개체들에 대한 탐색 과정을 수행하지 않는다. 따라서, 정적 경로 탐색 기법에서의 경로 탐색을 위한 전체 시스템에서의 시간 복잡성은 O(1)으로 볼 수 있다.

동적 경로탐색 기법은 시스템 내에 존재하는 모든 참여개체와 종말개체에 대한 정보를 응용 프로그램 구동과 함께 탐색 메시지 송수신을 통해 동적으로 수행해야 한다. 하나의 참여 개체는 시스템 내에 존재하는 (N-1)개의 다른 참여개체들에 대해 동적 경로 탐색을 수행해야 하며, 시스템 내에 총 N개의 참여개체가 존재하므로, 참여개체들간의 경로탐색을 위해 O(N<sup>2</sup>)의 시간 복잡성이 필요하게 된다. 또한, 하나의 참여개체 내에 존재하는 하나의 종말개체는 타 참여개체내에 존재하는 M개의 종말개체에 대한 경로탐색이 필요하며, 참여 개체 내부에는 M개의 종말개체가 존재하므로, 두개의 참여개체를 사이에서 종말개체들간의 탐색을 위한 시간 복잡성은 O(M<sup>2</sup>)이 된다. 따라서 동적 탐색기법에 대한 복잡성은 O(N<sup>2</sup>)×(M<sup>2</sup>)이 되며, 결과적으로 O(N<sup>4</sup>)으로 볼 수 있다.

하이브리드 경로탐색기법은 동적 경로탐색기법에서의 참여개체들간의 경로 탐색을 위한 O(N<sup>2</sup>)의 복잡성이 소요되며, 종말개체들에 대한 경로탐색은 수행하지 않고 사전 정의된 정보를 이용하므로 전체적으로 O(N<sup>2</sup>)의 복잡도를 가진다.

따라서, 위의 복잡도 분석 결과와 같이 경로 탐색 시간의 관점에서 볼 때 사전 정의된 경로 정보를 이용하는 정적 탐색 기법이 가장 빠르며, 동적 경로탐색 기법이 가장 느리다고 할 수 있다.

2) 처리자원 소모량 분석

일반적으로, 동적 경로탐색 기법에서 종말개체에 대한 경로탐색을 위해 할당되는 메모리의 크기(M<sub>discovery</sub>)는 다음과 같다.

$$M_{discovery} = N_{LP} \times DB_{RP} = N_{LP} \times N_{RP} \times N_E \times M_{DB}$$

여기서,

N<sub>LP</sub> : 지역 노드의 참여개체(Local Participant)의 수

$DB_{RP}$  : 원격 참여개체(Remote Participant)의 데이터베이스 크기

$N_{RP}$  : 탐색 원격 참여개체(Remote Participant)의 수

$N_E$  : 원격 참여개체(Remote Participant) 당 종말개체(endpoint)의 수

$M_{DB}$  : 데이터베이스 정보의 크기

이에 반해, 참여개체별로 미리 식별된 연관성 있는 종말개체(matching endpoint)들에 대한 경로 정보만을 이용하는 하이브리드 경로탐색 기법에서 할당되는 메모리( $M_{discovery}^{match}$ )는 다음과 같다.

$$M_{Discovery}^{match} = N_{LP} \times DB_{RP} \\ = N_{LP} \times N_{RP} \times N_{ME} \times M_{DB}$$

여기서,

$N_{ME}$  : 원격 참여개체(Remote Participant) 당 연관 종말개체(Matching Endpoint)의 수

$N_{ME} \leq N_E$  이므로  $M_{discovery}^{match} \leq M_{discovery}$  가 되어 연관된 종말개체만을 탐색하는 시스템이 메모리 소모가 더 적으며, 합정 전투체계와 같은 규모의 대형 시스템에서 종말개체 경로탐색에 할당된 메모리는 수십배 이상의 차이를 보인다.

#### 4. 시험 및 결과 분석

##### 가. 시험환경

합정 전투체계는 다양한 정보처리 시스템으로 구성되나, 일반적으로 전술 운용자와의 인터페이스를 제공하는 운용자 인터페이스 장비(HU : HMI Unit)와 센서나 무장과 같은 외부의 다양한 장비들을 연동하기 위한 연동장비(IU : Interfacing Unit), 그리고 수집된 데이터를 가공하여 전술 정보를 생성 및 처리하는 정보처리 장비(PU : Processing Unit) 등으로 구성된다.

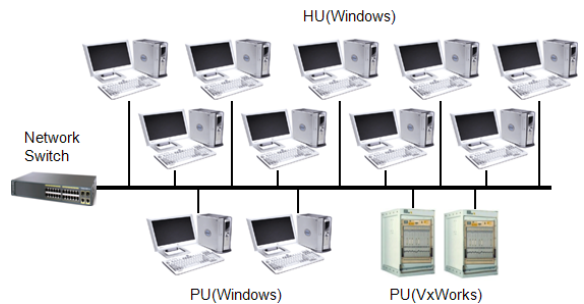
HU는 운용자의 명령을 입력받아 PU로 전달하며, PU로부터 처리된 전술 정보를 운용자에게 전시한다. 각 HU 노드는 실제로 할당된 운용자의 임무에 따라

다른 전술 운용 소프트웨어를 이용하나, 본 시험에서는 동일한 전술 임무를 수행하는 것으로 가정한다.

PU는 다수의 IU들과 통신하고 데이터를 처리하며 처리한 데이터를 운용자가 볼 수 있도록 HU로 전송한다. PU는 HU, IU 뿐만 아니라 분산 처리를 위해서도 다른 PU와도 통신을 수행하나, 본 시험구성에서는 PU들 간의 경로탐색은 고려하지 않는다.

IU는 PU와 탑재 장비 사이에서 데이터 변환 등을 수행하며, 그 외에 데이터 전송은 하지 않고 전투체계의 전체 데이터를 기록하는 노드가 있다.

시험의 구성은 그림 8과 같이 전체 시스템에서 경로탐색 중 실패 확률이 가장 높은 PU들과 HU들 간의 통신을 시험하도록 구성하였다. 이러한 두 종류의 노드에는 다수의 종말 개체들이 존재한다. 특히 HU는 많은 수의 응용프로그램을 가지고 있어서 경로탐색 소요시간 및 자원 소모 등 경로탐색에서 위험도가 가장 크다고 할 수 있다.



[그림 8] 시험 구성

##### ○ 시험 구성

- 구성1 : HU(Windows)×9, PU(Windows)×2
- 구성2 : HU(Windows)×9, PU(VxWorks)×2
- 구성3 : HU(Windows)×9, PU(Windows)×2, PU(VxWorks)×2

##### ○ HW 및 SW 규격

- HU : Windows XP MCE, Intel Core2 duo 2.13GHz, 2GB memory
- PU : Windows XP MCE, Intel Core2 duo 2.13GHz, 2GB memory
- PU : VxWorks 5.5.1, MVME6100-0163,

MPC7457 processor 1.267GHz, 2GB memory

- DDS 미들웨어 : RTI DDS 4.2<sup>[9]</sup>

나. 시험방법

경로탐색 성능 분석을 위해 경로탐색을 위한 시간과 이를 위해 소모된 시스템 자원량을 측정 하였다. 이러한 경로탐색 시간과 자원소모량은 전체 시스템에 존재하는 모든 종말개체들이 연관되는 모든 상대 종말개체들에 대한 탐색을 완료하는 시점을 기준으로 한다. DDS에서는 송수신 종말개체에 대해 상대 종말개체에 대한 탐색 정보를 제공하기 위해, 상대 종말개체가 탐색되는 경우 특정 콜백 함수를 호출해 준다. 본 시험에서는 이 콜백 함수를 이용하여 시스템 내에 존재하는 종말 개체들에 대해 상대 종말 개체들에 대한 탐색 개수를 측정하였으며, 예정된 전체 시스템의 탐색 개수와 콜백 함수를 통해 수집된 탐색 개수가 동일한 시점을 시스템 내의 모든 개체들이 경로 탐색을 완료한 시점으로 판단하였다.

또한, 시험에서 개별 종말 개체들이 경로 탐색을 완료한 후 실제 통신 메시지를 송수신함으로써, 다른 종말 개체들의 경로 탐색 단계에 대한 영향성을 배제하기 위해서 모든 종말 개체들에 대한 경로 탐색이 완료되기 이전에는 통신 메시지를 송신하지 못하도록 설정하였다.

시험 구성에 따른 종말개체의 수는 다음과 같다.

- 시험구성1/시험구성2
  - 전체 종말개체의 수 : 15,400개
  - HU : 13,860개(송수신 개체 각 6,930개)
    - 22 송수신 개체 × 70 참여개체 × 9 노드
  - PU : 3,080개(송수신 개체 각 1,540개)
    - 1,540 송수신 개체 × 1 참여개체 × 2 노드
- 시험구성3
  - 전체 종말개체의 수 : 20,020개
  - HU : 13,860개(송수신 개체 각 6,930개)
    - 22 송수신 개체 × 70 참여개체 × 9 노드
  - PU : 6,160개(송수신 개체 각 3,080개)
    - 1,540 송수신 개체 × 1 참여개체 × 4 노드

다. 시험결과

동적 경로탐색 기법과 하이브리드 경로탐색 기법에 대해서 위의 시험 구성으로 각각 시험한 결과는 다음과 같다. 정적 경로탐색 기법의 경우 장비 손상 및 결합 등에 의한 재 구동시 일정 시간 동안의 메시지 전송이 보장되지 않는 특성으로 인해 신뢰성 및 즉시성이 요구되는 합정 전투체계 시스템에 적용이 힘들기 때문에 시험에서 제외하였다.

1) 동적 경로탐색 기법 시험 결과

표 3은 HU(Windows) 9대, PU(Windows) 2대인 시험구성 1에 대한 동적 경로탐색 기법 적용의 시험 결과이다. 경로탐색 시간은 시험환경에 포함되는 모든 구성 장비를 구동한 시점으로부터 모든 탐색이 완료되는 시점을 기준으로 측정하였으며, 경로탐색에 소요되는 메모리 양을 개략적으로 비교하기 위해서 개별 구성장비에서 할당되는 전체 메모리 사용량을 기준으로 측정하였다.

[표 3] 동적 경로탐색 시험결과(시험구성1)

|                | PU(Windows)×2                   | HU×9                               |
|----------------|---------------------------------|------------------------------------|
| Memory usage   | 211MB                           | Total : 1.5GB<br>Each App : 21.4MB |
| Discovery time | PU1 : 4min 50sec<br>PU2 : 13min | 70~140sec                          |

시험 구성 2와 3의 경우 경로 탐색을 위한 메타트래픽 량과 자원 소모량 등이 급격하게 증가되어 전체 시스템에 대한 경로 탐색이 상당히 지연되었으며, 시험 진행 중 구성 장비들의 상태가 상당히 불안정함을 확인할 수 있었다. 또한, 이로 인해 충분한 시간이 경과한 이후에도 경로탐색이 완료되지 못하여 시험을 종료하였다.

2) 하이브리드 경로탐색 기법 시험결과

하이브리드 경로탐색 기법을 사용했을 경우 시험구성 1, 2, 3에 대한 시험 결과는 각각 표 4, 5, 6과 같다. 동적 경로탐색 기법의 단점인 경로탐색 소요시간과 메모리 사용이 현격히 줄었음을 알 수 있다. 메모

리의 소모량은 응용프로그램의 수가 많은 HU의 경우 사용이 많이 줄었고, 경로탐색 시간은 PU과 HU 모두 1분 이내로서 경로탐색의 고속화가 가능함을 보여 준다. 특히 표 5의 결과와 같이 전체 PU과 HU를 모의한 시험구성 3에서도 시험구성 1, 2의 결과와 비교 시 경로탐색 완료 시간과 메모리 사용량에서 큰 차이를 보이지 않았다.

[표 4] 하이브리드 경로탐색 시험결과(시험구성1)

|                | PU(Windows)×2              | HU×9                               |
|----------------|----------------------------|------------------------------------|
| Memory usage   | 184MB                      | Total : 670MB<br>Each App : 9.57MB |
| Discovery time | PU1 : 40sec<br>PU2 : 38sec | 2.2sec~2.5sec                      |

[표 5] 하이브리드 경로탐색 시험결과(시험구성2)

|                | PU(VxWorks)×2              | HU×9                                  |
|----------------|----------------------------|---------------------------------------|
| Memory usage   | 169MB                      | Total : 670MByte<br>Each App : 9.57MB |
| Discovery time | PU1 : 36sec<br>PU2 : 37sec | 480msec                               |

[표 6] 하이브리드 경로탐색 시험결과(시험구성3)

|                | PU(Windows)×2              | PU(VxWorks)×2              | HU×9                               |
|----------------|----------------------------|----------------------------|------------------------------------|
| Memory usage   | 180Mbyte                   | 170Mbyte                   | Total : 676MB<br>Each App : 9.65MB |
| Discovery time | PU1 : 40sec<br>PU2 : 41sec | PU1 : 48sec<br>PU2 : 47sec | 2.0~2.5sec                         |

라. 결과분석

동적 경로탐색 기법을 적용한 시험 결과는 규모가 상대적으로 큰 시스템 환경에서 구성노드간의 실제 메시지 전송이 수행되기 이전에 송수신 개체들간의 탐색을 위해 많은 자원과 시간이 소요됨을 보여준다. 이러한 특성은 시스템의 규모가 증가할수록 더 크게 나타나며, 이로 인해 시스템의 기능을 수행하기 위해

필요한 내부의 실제 메시지 교환에 큰 영향을 주게 된다. 따라서, 함정 전투체계와 같이 다수의 시스템 구성 노드들로 구성된 중대규모의 컴퓨팅 시스템에 동적 경로 탐색 기법을 단순히 적용하는 것은 무리가 있다고 할 수 있다.

하이브리드 경로탐색 기법의 경우 20,000 여개의 종말 송수신 개체의 경로를 탐색하는데에, 경로탐색 시간이 1분 이하이고, 메모리 소모량은 모든 하드웨어에서 절반 이상의 충분한 여유를 허용하고 있다. 그러므로 HU와 PU 이외의 노드를 추가하거나, 시험의 모의 대상인 중형 전투체계 이상의 더 큰 시스템에서도 적용이 가능할 것으로 판단된다.

5. 맺음말 및 향후 연구

대형 복합 실시간 시스템으로서 데이터 중심의 통신 특성을 보이는 함정 전투체계 시스템 개발을 위해서는 DDS와 같은 상위수준의 통신 메커니즘을 지원하는 통신 미들웨어의 적용이 필요하다. 또한, 다수의 노드에서 많은 수의 단위 응용 프로그램들 간의 상호 운용을 위해서는 메시지 송수신 개체들간의 복잡한 상호 관계를 효과적으로 식별할 수 있는 경로탐색 기법이 요구된다. 본 논문에서는 시스템 내에 존재하는 메시지 송수신 개체들에 대한 상호 관계 식별을 위한 동적 경로탐색 기법과 정적 경로탐색 기법을 분석하고, 각 기법들의 문제점을 보완할 수 있는 고속의 하이브리드 경로탐색 기법을 제안하였다. 또한, 고속의 경로 탐색 및 적은 메모리 사용 등의 장점을 시험을 통해 확인하였으며, 함정 전투체계와 같이 복잡한 시스템에 대한 적용성을 확인하였다.

현재 단계에서는 경로탐색 실패 가능성이 높고 종말개체 수가 많아 경로탐색 과정에서 중요성이 가장 높은 HU와 PU을 모의한 시험을 통하여 하이브리드 경로탐색 기법의 성능을 확인하였으나, 추후 단계에서는 IU 등 다른 노드들을 추가하여 성능 평가를 수행하고, 최종적으로 실제 전투체계 개발 장비 및 응용프로그램을 이용하여 성능을 입증할 예정이다.

향후 발전될 함정 전투체계를 대비하여 하이브리드 경로탐색 기법 사용시 종말개체들에 대한 수동 경로

정보 설정의 복잡성을 해소하기 위한 방안이 수립되어야 할 것으로 판단된다. 또한, 수동 경로 정보 설정의 복잡성과 시스템의 유연성 및 재사용성 저하 등의 특성을 보이는 제안 경로 탐색 기법의 한계를 보완하기 위해서는 동적 경로 탐색 기법에 대한 성능 개선, 전투체계 소프트웨어 구조 개선 및 시스템 규모와 경로탐색 시간 간의 세부 상관관계 등과 같은 별도의 연구가 지속적으로 이루어져야 할 것으로 판단된다.

### 참 고 문 헌

- [1] C&CA Working Group, "Report of High Performance Distributed(HiPer-D) Computing Program", NSWC, Sep., 1998.
- [2] 노정호, "개방형 아키텍처 기반의 함정 전투체계 기술 발전 방향", 제10회 해군해양과학기술 심포지엄 논문집, pp. 161~176, 해군사관학교 해군해양연구소, 2005.
- [3] 이재승, 김점수, 박신배, 이균정, "함정 전투체계를 위한 통신 미들웨어 평가방안 연구", 제5회 해상무기체계 발전세미나, 2004.
- [4] 이재승, 김은지, 김점수, "NDDS 기술 분석 보고서", 국방과학연구소, NSDC-115-040473, 2004.
- [5] 이재승, "함정 전투체계를 위한 통신 미들웨어 성능 적합성 평가 보고서", 국방과학연구소, NSDC-525-051049, 2005.
- [6] 이재승, "NDDS 적응 계층의 설계와 구현", 국방과학연구소, NSDC-115-060614, 2006.
- [7] OMG, Data Distribution Service for Real-Time Systems Version 1.2, 2007.
- [8] OMG, The Real-time Publish-Subscribe Wire Protocol DDS Interoperability Wire Protocol Specification, 2007.
- [9] RTI, RTI<sup>®</sup> Data Distribution Service The Real-Time Publish-Subscribe Middleware User's Manual Version 4.2.