

효율적인 비행계획 연관을 위한 CWP MRT 데이터 처리 모듈 구현

Implementation of CWP MRT Data Processing Module for Efficient Correlating Flight Plan

김강희 · 최상방*

인하대학교 전자공학과

Kanghee Kim · Sangbang Choi*

Department of Electronics Engineering, Inha University, Incheon, 402-751, Korea

[요 약]

비행계획 연관은 안전하고 신속한 관제 업무를 위해 매우 중요한 기능이다. 본 논문에서는 레이더 감시자료 중 MRT 데이터와 비행계획을 연관시키는 과정에서 발생하는 네트워크 부하를 최소화함으로써 전체적인 항공교통 자동화 시스템의 성능을 향상시킬 수 있는 CWP MRT 처리 모듈을 구현하였다. 구현 모듈은 분산형 연관 방식을 적용함으로써 항공교통관제 자동화 시스템의 효율성과 안전성을 동시에 확보하며, 인코딩 방식의 비행계획 연관 MRT 데이터 포맷을 사용함으로써 네트워크 사용량을 감소시킨다. 구현 모듈은 독립형 연관 방식에 비해 네트워크 사용량이 8.54~12.11% 낮았으며, 네트워크 공정성이 8.37~11.27% 높은 것을 확인하였다.

[Abstract]

It is very important to correlate flight plan for safe and prompt ATC(air traffic control) operation. In this paper, we design CWP(controller working position) MRT(multi radar tracking) processing module to achieve improvement of overall ATC automation system's performance by minimizing network traffic overload when correlating MRT with flight plan. This implemented module can guarantee efficiency and safety of ATC automation system by applying distributed correlated manner, and reduce network usage by using encoded flight plan correlated MRT data format. We found that this module has 8.54~12.11% lower network usage and 8.37~11.27% higher network traffic fairness than the module using standalone manner.

Key word : ATC automation system, MRT, Correlated MRT, ASTERIX, Network traffic fairness.

<http://dx.doi.org/10.12673/jant.2014.18.4.268>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 10 July 2014; Revised 22 August 2014

Accepted (Publication) 18 August 2014(30 August 2014)

*Corresponding Author; Sangbang Choi

Tel: +82-32-860-7417

E-mail: sangbang@inha.ac.kr

1. 서론

항공교통관제의 목적은 항공기의 이·착륙시 안전을 보장하고 원활한 항공교통흐름이 이루어지도록 하는 것이다. 항공교통관제 자동화 시스템(air traffic control automation system)을 이루고 있는 서브시스템 중 CWP(control working position)는 이러한 항공교통관제의 목적 달성을 위한 관제사와의 시스템 간의 인터페이스(HMI; human machine interface)를 제공하며 실시간으로 변화하는 항공상황을 보여줌으로써 항공기간의 충돌을 미연에 방지하고 효율적으로 항공교통흐름을 통제할 수 있도록 도와주는 역할을 담당한다[1],[2].

CWP의 기능은 크게 화면 표시 기능, 교통량 관리 기능, 비행자료 관리 기능, 관제자료 관리 기능, 항공기 관리 기능이 있다[3]. 이 중 가장 중요한 기능은 운항 중인 항공기의 정보를 가진 레이더 감시자료를 통해 현재 항공교통상황을 표시하는 화면 표시 기능이다. 이 기능을 통해서 관제사는 지상에서 관제 공역의 모든 항공기를 모니터 화면 하나만으로 살펴볼 수 있으며 항공기의 위치, 속도, 고도를 확인할 수 있다. 그러나 항공상황에 적합한 관제 명령을 내리기 위해서는 해당 항공기가 어느 항공사의 비행기인지, 몇 시에 출발 또는 도착 예정인지, 어느 공항을 향해서 비행하고 있는지 등의 정보를 포함한 비행계획이 있어야 한다. 따라서 CWP는 수신한 비행계획을 관제사가 한 눈에 보기 쉽게 화면에 표시하는 역할을 수행해야 한다. 비행계획 표시 방법 중 표시화면의 항적 데이터블록에 관한 비행계획의 일부 필드를 추가하는 방법이 있다. 이를 레이더 감시 자료와 비행계획의 연관이라고 한다. 관제사는 비행계획 전문을 열람하지 않고도 비행계획이 추가된 데이터블록을 통하여 해당 항공기에 대한 정보를 파악함으로써 신속한 관제 업무를 수행할 수 있다. 레이더 감시자료와 비행계획의 연관을 처리하는 모듈의 최적화는 항공교통관제 자동화 시스템의 성능 향상을 위해 매우 중요하다. 연관된 비행계획을 탐색하는 데 긴 시간이 소요되거나 연관 작업이 지연될 경우 비행계획이 포함된 데이터블록을 생성할 수 없기 때문에 관제 업무에 불편을 초래하게 된다.

본 논문에서는 레이더 감시 자료 중 MRT(multi radar tracking)와 비행 자료의 결합을 위해 사용되는 네트워크의 부하를 최소화함으로써 증가하는 네트워크 트래픽을 안정적으로 수용할 수 있는 CWP MRT 데이터 처리 모듈을 설계하였다. 구현한 MRT 데이터 처리 모듈은 네트워크 부하를 줄이기 위해 ASTERIX로 인코딩된 데이터 포맷을 사용하며, SDP(surveillance data processing system)로부터 수신한 MRT 데이터와 FDP(flight data processing system)로부터 수신한 비행계획 연관 MRT 데이터를 디코딩하고 항적과 데이터블록을 표시하는 데에 필요한 데이터를 추출한다.

본 논문은 다음과 같이 구성된다. 2장에서는 CWP MRT 데이터 처리 모듈의 구현과 관련한 연구를 소개하고, 3장에서는 본 논문에서 구현한 CWP의 감시자료처리 모듈에 대하여 설명

한다. 4장에서는 측정을 통해 구현 모듈의 성능을 평가하였으며, 5장에서 결론을 맺는다.

II. 관련연구

이 장에서는 본 논문에서 구현한 CWP MRT 데이터 처리 모듈의 성능을 평가하기 위해 항공교통관제 자동화 시스템의 구조와 각 서브시스템의 기능에 대하여 설명한다. 그리고 MRT와 비행계획의 연관 방식에 대하여 설명한다.

2-1 항공교통관제 자동화 시스템

항공교통관제는 공항 또는 항로를 비행하는 항공기의 안전 보장, 항공기 간의 질서 유지, 신속한 항공 교통 흐름을 위해 지상의 관제사들에 의해 제공되는 서비스로, 조종사들에게 적절한 명령을 내려 항공기 간의 안전한 분리 간격을 유지시키는 것이 주목적이다. 항공교통관제가 처음 시행된 1960년도에는 교통량이 많지 않았기 때문에 충분히 사람에 의한 서비스 제공이 가능하였다. 그러나 해가 갈수록 항공교통량이 크게 증가하고 관제 공역이 세분화되면서 사람에 의한 관제업무의 수용력을 초과하게 되었고, 이로 인해 인력 부족, 업무 사고에 의한 안전성 문제, 관리 비용 문제 등의 추가 문제들이 발생하였다. 이러한 문제들을 해결하기 위해 항공교통관제 자동화 시스템이 개발되기 시작하였다[4]. 항공교통관제 자동화 시스템은 항공교통관제시스템 구성 요소 중에 하나로 항공교통관제를 운영하기 위한 자료를 자동으로 처리하고 이를 표시하는 역할을 수행한다.

그림 1은 항공교통관제 자동화 시스템의 구조를 나타내며 다수의 서브시스템들로 구성되어 있다. SIF(surveillance interface)는 항공교통관제 자동화 시스템과 외부 감시레이더와 기상레이더들을 연결하는 다리 역할을 수행하는 시스템으로, 감시레이더가 생성한 raw sensor data와 기상레이더가 생성한 raw weather data를 수신하면 데이터 길이 검증과 로그 기록 이후에 ATC 내부 network interface로 데이터를 전달한다. EIF(external interface)는 항공교통관제 자동화 시스템과 외부 비행자료 시스템들을 연결하는 통로 역할을 수행하는 시스템으로, AFTN(aeronautical fixed telecommunication network)으로부터 ATS(air traffic service) 전문(비행계획, 출·도착, 지연, 수정, 취소, 경고 등), MCRC(master control and report center)로부터 군 관련정보, AIS(aeronautical information services)로부터 NOTAM(notice to airmen)을 수신하여 network interface를 통해 내부 ATC 시스템으로 전달한다. SMC(system management control system)는 항공관제시스템을 구성하는 서브시스템을 제어하는 시스템으로, network interface를 통해 지속적으로 서브시스템의 상태를 보고 받으며 직접 서브시스템에 명령을 내려 어플리케이션의 on/off를 수행한다. SDP는 CWP에 표시할 항적데이터를 생성하는 시스템으로, network interface를 통해

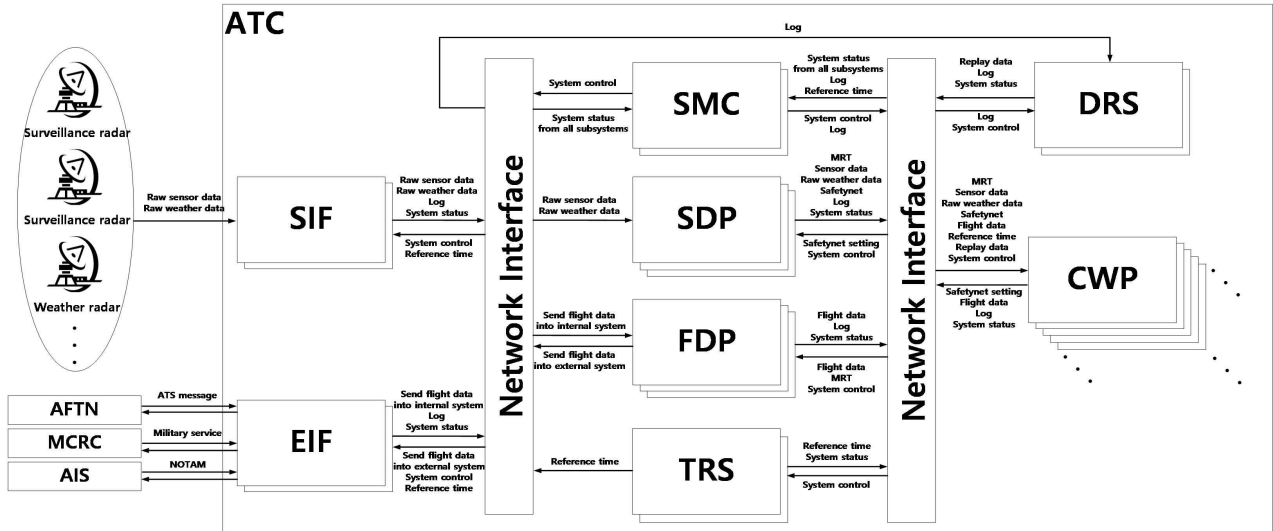


그림 1. 항공교통관제 자동화 시스템의 구조
 Fig. 1. Structure of ATC automation system.

수신한 raw sensor data를 조합하여 하나의 MRT(multi radar tracking) 항적을 생성하며 생성 항적 간의 충돌이 예상되거나 위험 지역으로 진입할 경우 경고/경보메시지를 생성하여 network interface를 통해 CWP로 전달한다. FDP는 수신 비행자료를 처리하는 시스템으로, 비행자료의 오류 검증을 실시하며 검증을 통과한 비행자료를 내부 데이터베이스에 저장한다. 내부 데이터베이스의 비행자료 중 이륙을 준비 중이거나 관제 구역에 들어오는 항공기와 관련되어 있으면 해당 비행자료를 사용 가능한 상태로 활성화시킨다. TRS(time reference system)는 항공교통관제시스템의 시간 서버로 서브시스템 간의 시간 동기화를 위해 사용된다. DRS(data recoding system)은 항공교통관제 자동화 시스템의 모든 서브시스템에서 전송하는 로그 데이터를 저장하는 시스템으로, 관제기록참조 또는 사고조사 시 저장된 데이터를 SMC에서 조회할 수 있으며, 특정기간동안의 감시자료와 비행자료를 추출하여 당시 관제상황을 CWP에서 재생할 수 있다. CWP는 항공교통관제시스템의 서브시스템들이 처리한 데이터를 표시하는 시스템으로, network interface로부터 들어오는 SMC·SDP·FDP·TRS·DRS의 데이터를 수신하여 모니터 화면에 표시한다.

항공교통관제 자동화 시스템을 이루는 서브시스템들과 네트워크는 시스템 장애에 대비하여 이중화 또는 삼중화 구성되어 있다. 평시에는 주 시스템을 운영하며 나머지 시스템들은 대기 상태로 둔다. 만약 주 시스템에 장애가 발생한 경우 대기 상태로 놓여 있던 예비 시스템을 주 시스템으로 전환하여 운영하며 전환 시 다른 서브시스템에 영향을 주지 않도록 해야 한다. CWP는 다중화 구성을 하지 않는데 이것은 CWP들 중 하나의 장애가 서브시스템들의 운영 또는 다른 CWP의 관제 업무에 영향을 주지 않기 때문이다.

2-2 비행계획 연관 MRT

CWP는 관제사가 보는 영역에 SDP로부터 수신한 MRT를 표시하며, 표시된 MRT는 식별부호(SSR-code), 위치, 고도, 속도, 상승/하강 지시자 등의 정보들을 포함하고 있다. 그러나 더 정확한 관제 업무를 수행하기 위해서는 레이더 감시자료가 가리키는 항공기의 항공사 정보, 기체의 크기, 출/도착 공항, 출/도착 시간, 배정 고도, 사용 활주로 등의 추가 정보가 필요하다. 국제민간항공기구 ICAO(international civil aviation organization)에서는 위와 같은 정보를 포함하는 표준화된 전문 형식을 PANS(procedures for air navigation services) air traffic management doc.4444에서 정의하고 있으며 이를 비행계획(FPL; filed flight plan)이라고 한다[5-6]. 작성된 비행계획은 AFTN(aeronautical fixed telecommunication network)을 통해 전세계의 항공교통관제 자동화 시스템으로 전달된다.

CWP를 개발하는 현대의 모든 ANSP(air navigation service provider)들은 표시된 MRT에 관련 비행계획을 약식으로 표시할 수 있는 기능을 제공하고 있다. 이를 비행계획 연관 MRT라고 한다. 항공교통관제 자동화 시스템은 MRT와 관련된 비행계획을 가지고 있는 경우 자동으로 비행자료 연관 MRT를 생성하며, 비행계획이 존재하지 않는 경우에도 관제사의 입력을 통해 수동으로 비행계획을 생성하여 MRT와 연관시킬 수 있다. CWP에 표시되는 MRT와 비행계획 연관 MRT의 예는 그림 2와 같다.

MRT는 그림 2(a)와 같이 데이터블록에 식별부호(7240), 실제고도(086; 8600 FL), 속도(21; 210 knot), 상승/하강 지시자(V; 하강비행)만이 표시되지만, 연관 비행자료가 존재할 경우 그림 2.(b)와 같이 데이터블록에 편명(ESR218; 이스타항공 218편), 항적난기류(M; 7톤 이상 136톤 이하의 중량급), 관제

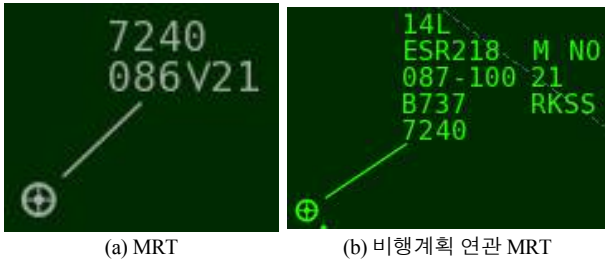


그림 2. MRT와 비행계획 연관 MRT의 표시
 Fig. 2. Display of MRT and flight plan correlated MRT.

섹터(NO), 실제고도(087; 8700 FL), 상승/하강 지시자(-; 배정 고도 200 ft 외부에서 하강비행), 배정고도(100; 10000 FL), 기종(B737; boeing 737), 도착공항(RKSS; 김포공항), 식별부호(7240), 사용 활주로(14L) 등이 추가로 표시된다.

2-3 비행계획 연관 방식

비행계획을 MRT에 연관시키는 방식은 비행계획 연관 MRT를 생성하는 서브시스템이 무엇인지에 따라 독립형 연관 방식과 분산형 연관 방식으로 구분할 수 있다. 독립형 연관 방식은 CWP가 자체적으로 내부에 저장되어 있는 비행계획과 수신 감시자료를 비교하여 비행계획을 항적과 연관시켜 데이터블록에 표시하는 방식이며, 분산형 연관 방식은 FDP가 비행계획 연관 MRT를 생성하고 이를 수신한 CWP가 비행계획 연관 MRT를

데이터블록에 표시하는 방식이다.

CWP 내부에는 비행계획 연관 MRT의 생성을 위한 기능 블록인 CIOB(CWP in/out block), CAMB(CWP aeronautical data management block), CFMB(CWP flight data management block), CDMB(CWP display management block)가 있다. CIOB는 CWP와 외부 서브시스템 간의 모든 데이터 입출력을 담당하는 블록으로 수신한 데이터를 CWP 내부 블록 또는 CWP 외부 network interface로 전달한다. CAMB는 감시자료를 처리하는 블록으로 수신한 감시자료를 디코딩하여 표시에 필요한 정보를 추출한다. CFMB는 비행자료를 처리하는 블록으로 수신한 비행자료를 표시 가능한 데이터로 변환한다. CDMB는 실제 모니터 화면에 데이터를 보여주는 블록으로 CWP에서 처리한 데이터를 관제화면의 구성에 맞추어 표시한다.

그림 3은 독립형 연관 방식의 흐름도를 나타낸다. SDP와 FDP는 생성한 감시자료와 비행자료를 network interface를 통해 지속적으로 CWP의 CIOB에 전달한다. CIOB는 수신데이터를 감시자료와 비행자료로 구분하여 각각 CAMB와 CFMB에 전달한다. CAMB는 수신 감시자료의 유효성을 검증한 뒤 표시에 필요한 데이터를 추출하여 CDMB에 전송하며, 이 중 MRT는 비행계획 연관 MRT 생성을 위해 CFMB로 전송한다. CFMB는 수신한 MRT와 내부에 저장된 비행계획을 비교하여 항공기 정보가 일치하면 비행계획 연관 MRT를 생성하여 다시 CAMB로 전달한다. CAMB는 CFMB로부터 비행계획 연관 항적을 수신한 경우 이를 CDMB로 전달하며, 그렇지 않은 경우 수신한

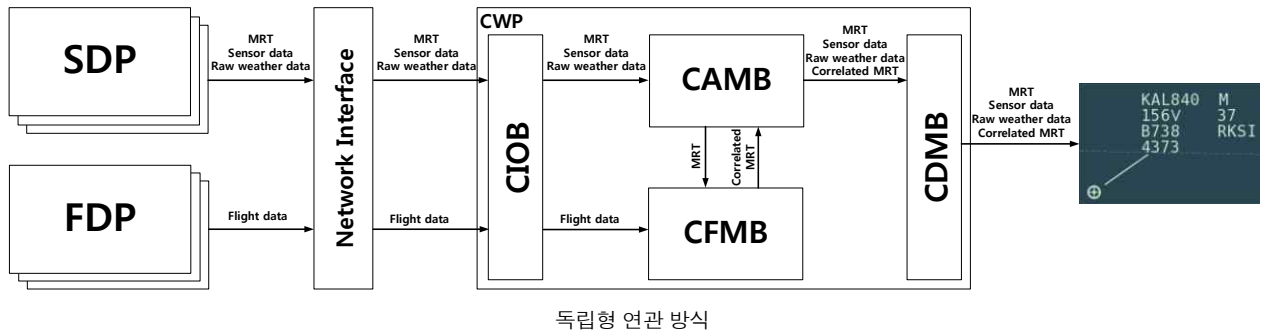


그림 3. 독립형 연관 방식의 흐름도
 Fig. 3. Flow chart of standalone correlation.

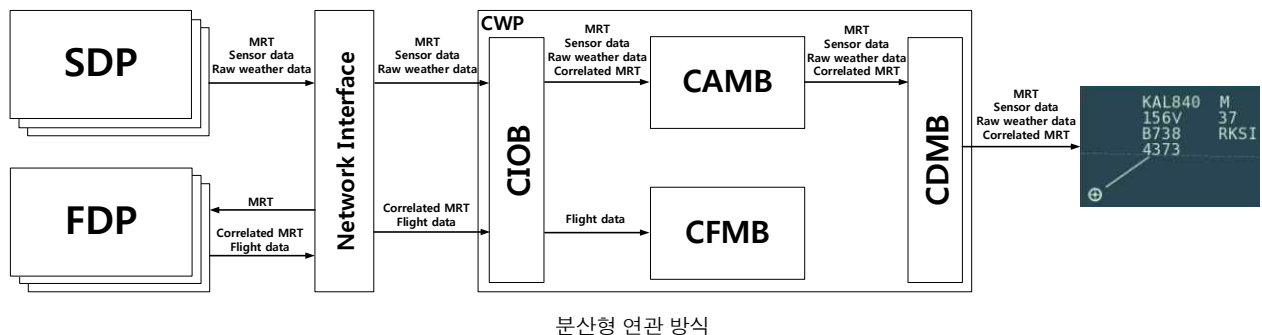


그림 4. 분산형 연관 방식의 흐름도
 Fig. 4. Flow chart of distributed correlation.

MRT를 그대로 CDMB에 전달한다.

그림 4는 분산형 연관 방식의 흐름도를 나타낸다. SDP는 생성한 감시자료를 network interface를 통해 CWP의 CIOB에 전달하며 동시에 MRT 정보를 FDP에 전송한다. FDP는 수신한 MRT를 분석하여 내부에 저장된 비행계획과 항공기 정보가 일치하면 비행계획 연관 MRT를 생성하여 network interface를 통해 CWP의 CIOB로 전달한다. CIOB는 수신한 감시자료 중에 비행계획 연관 MRT가 존재하면 이를 CFMB가 아닌 CAMB로 전달한다. CAMB는 수신한 MRT와 비행계획 연관 MRT를 비교하여 항공기 정보가 일치하면 CDMB로 비행계획 연관 항적을 전송한다. 반면, 일치하지 않은 경우 비행계획 연관 MRT를 폐기한다.

독립형 연관 방식은 CWP 내부에서 작업이 이루어지기 때문에 CWP의 구현 복잡도가 증가하며, 생성한 비행계획 연관 MRT를 다른 모든 CWP에 적용하기 위해서는 추가로 CWP와 FDP간 또는 CWP들 간의 동기화가 필요하다. 그에 비해 분산형 연관 방식은 FDP가 비행계획 연관 MRT 생성을 맡기 때문에 CWP의 구현 복잡도가 크게 줄어들며 CWP와 FDP간의 동기화 작업 없이 모든 CWP에 동일한 비행계획 연관 MRT를 현시할 수 있다. 따라서 본 논문에서는 독립형 연관 방식보다 효율적으로 비행계획과 MRT의 연관을 수행할 수 있는 분산형 연관 방식을 기반으로 한 항공교통관제 자동화 시스템을 사용하였다.

III. CWP MRT 데이터 처리 모듈 구현

이 장에서는 본 논문에서 구현한 CWP MRT 데이터 처리 모듈에 대하여 설명한다. 우선 비행계획 연관 MRT를 위한 데이터 포맷을 정의하고, MRT 데이터와 비행계획 연관 MRT 데이터를 결합하는 동작 과정에 대하여 설명한다.

3-1 비행계획 연관 MRT 포맷 정의

AFTN을 통해 교환되는 비행계획 전문은 작성자가 입력한 원본에 대한 인코딩 또는 변환 과정 없으며 ASCII code 텍스트 포맷을 사용한다. 그러나 비행계획 전문 포맷에는 MRT와의 연관을 표현할 수 있는 필드가 존재하지 않기 때문에 연관 정보를 포함할 수 있는 새로운 메시지 포맷이 필요하다. 따라서 FDP는 비행계획 연관 MRT 데이터를 어떤 메시지 포맷을 이용하여 CWP로 전송할 것인지를 결정해야한다.

FDP가 비행계획 연관 MRT 데이터를 생성하는 방식은 무인코딩 방식과 인코딩 방식이 있다. 무인코딩 방식은 가장 단순한 구현 방법으로 수신 비행계획 전문의 원본을 그대로 두고 MRT 연관 여부를 판단할 수 있는 ASCII code 텍스트 데이터를 전문의 앞 또는 뒤에 추가로 덧붙이는 방식이다. 무인코딩 방식은 텍스트 기반의 데이터로 문자당 1 octet의 크기를 차지하기 때문에 항공교통관제 자동화 시스템 내부망의 통신량을

을 증가시킬 수 있다. 인코딩 방식은 수신 비행계획과 MRT와의 연관 정보를 인코딩한 새로운 포맷을 생성하는 방식으로, 정확한 비행계획 연관 MRT 데이터의 전달을 위하여 시스템 간에는 미리 정의된 인코딩 규약에 따라 데이터를 처리하는 모듈이 필요하다. 이 방식은 비행계획 전문에서 비행계획 연관 MRT에 필요한 데이터만을 추출하여 인코딩하고 이를 CWP에 전송하기 때문에 네트워크 사용량을 크게 줄일 수 있다.

항공교통관제 자동화 시스템은 모든 서브시스템들이 network interface를 통해 연동되어 있기 때문에 각 서브시스템들이 발생하는 네트워크 트래픽이 network interface에 집중된다. 만약 과도한 네트워크 트래픽이 발생하면 network interface의 데이터 처리가 지연되어 전체 시스템의 장애를 가져올 수 있다. 따라서 가능한 한 불필요한 데이터를 제거하고 데이터 크기를 줄임으로써 네트워크 부하를 최소화하는 것이 항공교통관제 자동화 시스템의 성능 향상을 이끌 수 있다. 이에 본 논문에서는 CWP MRT 데이터 처리 모듈의 비행계획 연관 MRT 데이터 포맷으로 인코딩 방식을 이용한다.

항공교통관제 자동화 시스템의 구현 방식과 관계없이 인코딩 방식의 비행계획 연관 MRT 데이터를 다루기 위해서는 표준화된 인코딩 규약이 필요하다. 국제민간항공기구 ICAO에서는 이러한 문제를 해결하기 위해 표준 감시자료 데이터 포맷인 ASTERIX(all purpose structured eurocontrol surveillance information exchange)[7]를 만들었으며, 이 중 ASTERIX CAT.062[8]는 인코딩 방식의 비행계획 연관 MRT 데이터의 전송을 지원할 수 있는 포맷을 가지고 있다. ASTERIX CAT.062는 항공기의 ID, 위치, 속도, 고도 등 항적의 비행상태를 알려주는 필드정보를 포함하고 있으며, 특별히 관련 비행자료를 실어 전송할 수 있는 I062/390 flight plan related data 필드가 존재한다.

그림 5는 I062/390 필드의 primary subfield를 나타내는 그림이다. I062/390의 primary subfield는 3 octets의 크기를 가지고 있으며, 각 bit는 subfield의 존재 여부를 의미한다. 예를 들어 TAG bit가 1이라면 primary subfield 다음에 TAG의 subfield가 뒤이어 붙는다는 것을 말한다. FX는 데이터를 확장하기 위해

Octet no.1							
24	23	22	21	20	19	18	17
TAG	CSN	IFI	FCT	TAC	WTC	DEP	FX
Octet no.2							
16	15	14	13	12	11	10	9
DST	RDS	CFL	CTL	TOD	AST	STS	FX
Octet no.3							
8	7	6	5	4	3	2	1
STD	STA	PEM	PEC	0	0	0	FX

그림 5. I062/390 비행계획 관련 데이터의 primary subfield
Fig. 5. Primary subfield of I062/390 flight plan related data.

표 1. I062/390 subfield들의 정의와 데이터크기
Table 1. I062/390 definition and data size of subfields.

Octet	Primary subfield	Subfield	Size(octet)
24	TAG	레이더 ID	2
23	CSN	편명	7
22	IFI	비행계획 ID	4
21	FCT	비행범주	1
20	TAC	기종	4
19	WTC	항적난기류	1
18	DEP	출발공항	4
17	FX	- (사용하지 않음)	-
16	DST	도착공항	4
15	RDS	사용활주로	3
14	CFL	배정고도	2
13	CTL	현재관제위치	2
12	TOD	예정시간	5+
11	AST	계류장 이름	6
10	STS	계류장 상태	1
9	FX	-	-
8	STD	SID	7
7	STA	STAR	7
6	PEM	SSR-CODE	4
5	PEC	예비편명	7
4	0	-	-
3	0	-	-
2	0	-	-
1	FX	-	-

```
(FPL-ESR218-IS-B737/M-S/C-RKPC0535
-N0400F100 GUKSU Y722 OLMEN Z52 KAKSO G585
SEL-RKSS0105 RKSI RKTU-0)
```

그림 6. 비행계획 전문 예
Fig. 6. Example of flight plan.

사용하는 필드이다. 만약 octet no.1의 FX 필드의 값이 1이면 다음에 octet no.2가 확장됨을 의미하며 그렇지 않으면 octet no.1에서 primary subfield가 종료됨을 의미한다.

표 1은 각 subfield들의 정의와 데이터 크기를 나타낸다. I062/390 subfield에는 항공기의 편명, 기종 항적난기류, 출발/도착공항, 사용 활주로, 허가고도, SSR-code 등의 비행자료를 포함한다. CWP는 I062/390 subfield들 중 레이더 ID와 SSR-code를 이용하여 MRT 데이터를 탐색하며 레이더 ID와

SSR-code가 일치하는 데이터를 발견 시 해당 정보들을 가진 데이터블록을 생성하여 항적과 결합한다.

그림 6은 그림 2.(b)의 비행계획 연관 MRT를 생성을 위해 사용한 비행계획 전문의 예이다. FPL은 해당 전문이 비행계획이라는 것을 의미하고, ESR218은 이스타항공의 218편, I는 계기 비행, S는 스케줄된 정기비행, B737은 보잉 737기, M은 항공기가 7톤 이상 136톤 이하의 중량을 가지고 있다는 것, S는 항로비행을 위한 표준 장비(VHF, VOR, ILS)를 갖추고 있다는 것, C는 SSR Mode A와 Mode C 트랜스폰더를 갖추고 있다는 것, RKPC0535는 김해공항에서 UTC(universal time coordinated) 시간으로 5시 35분에 출발한다는 것을 의미한다. 그리고 N0400은 400 knot로 비행, F100은 배정고도가 10000 ft라는 것을 의미한다. 그 다음 GUKSU Y722 OLMEN Z52 KAKSO G585 SEL은 항공기의 운항이 계획된 항로로, 알파벳 문자로만 쓰여진 것은 통과 지점을 의미하며 숫자가 포함된 것은 항로의 번호이다. RKSS0105는 도착공항이 RKSS이고 총 예상비행시간이 1시간 5분이라는 것을 의미하고, RKSI와 RKTU는 도착공항에 문제가 발생한 경우 사용가능한 예비공항명을 의미한다. 마지막 0은 추가정보가 작성되는 부분으로 0인 경우 추가 정보가 존재하지 않는다는 것을 의미한다.

그림 7은 그림 6의 비행계획을 그림 5와 표 1을 통해 설명한 I062/390을 이용하여 인코딩한 예이다. CSN, TAC, WTC, DST, RDS는 ASCII형 subfield로 첫 octet에서부터 차례로 데이터를 집어넣고 남은 octet은 0으로 설정한다. TAG는 레이더 ID로 수신 MRT 데이터의 레이더 ID와 동일하게 설정함으로써 CWP가 비행계획 연관 MRT라는 것을 인식할 수 있게 한다. IFI는 비행계획 ID로 4 octets의 크기를 가지고 있으며 총 4억 개의 비행계획을 다룰 수 있다. IFI는 비행계획 처리 속도 향상을 위하여 매우 유용한 필드이다. 관제화면에 현시 중인 항공기에 대한 비행계획을 조회하고자 할 때 IFI를 사용하지 않을 경우 CSN과 PEM 두 개를 비교하여 비행계획을 검색해야 한다. FDP에 저장된 비행계획 개수가 n 이라고 할 때, CSN과 PEM을 찾기 위해 두 번 검색을 수행해야 하므로 검색에 드는 비용의 worst case는 $O(n^2)$ 이 된다. 뿐만 아니라 CSN과 PEM이 같은 비행계획이 다수 검색될 경우 이를 CWP에 모두 보여 줄 수 있는 메시지를 생성해야 한다. 그러나 IFI를 사용할 경우, 일치하는 IFI만을 찾아내면 되므로 검색에 소비되는 비용의 worst case는 $O(n)$ 이며 중복되는 비행계획이 존재하지 않는다. CFL은 비행계획 전문의 배정고도를 그대로 사용한다. CTL은 관제색터를 의미하며 구역 설계를 통해 지정한 관제색터코드를 기록한다. PEM에는 수신 MRT의 SSR-code를 기록한다.

수신 비행계획을 그대로 사용할 경우 총 103 octets 크기의 데이터를 전송해야 하지만 I062/390 필드를 이용하면 34 octets로 데이터 크기를 2배 이상 감소하므로 네트워크 사용량을 크게 줄일 수 있다. 따라서 본 논문에서는 I062/390 필드로 인코딩한 데이터 포맷을 MRT와 비행계획의 연관에 이용한다.

I062/390 primary field																																																
Offset(octet)	0											1								2																												
Data(hex)	D				D							F				1				2							0																					
Data(binary)	1	1	1	0	1	1	0	1	1	1	1	1	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0																	
Subfield	TAG	CSN	IFI	TAC	WTC	DST	RDS	CFL	CTL	PEM																																						
Subfield	TAG	CSN				IFI				TAC				WTC	DST				RDS	CFL	CTL	PEM																										
Data type	number	ASCII								number				ASCII				ASCII	ASCII				ASCII	number	number	number																						
Offset (octet)	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33																	
Data (hex)	64	64	45	53	52	32	31	38	00	00	00	7D	5F	42	37	33	37	4D	52	4B	53	53	31	34	4C	01	90	01	01	0E	A0																	
Data	100	100	E	S	R	2	1	8	-	32095				B	7	3	7	M	R	K	S	S	1	4	L	F100	1	1	7240																			

그림 7. I062/390 인코딩 예
 Fig. 7. Example of encoding I062/390.

3-2 CWP MRT 데이터 처리 모듈

분산형 연관 방식을 사용할 경우 SDP는 MRT 데이터를, FDP는 비행계획 연관 MRT 데이터를 각각 독립적으로 network interface에 전송하므로 MRT 데이터와 비행계획 연관 MRT 데이터는 정확히 시간 동기화되어 CWP에 수신되지 않는다. 만약 비행계획 연관 MRT 데이터가 MRT 데이터보다 먼저 들어오면 CWP 내부에는 MRT 정보가 존재하지 않기 때문에 비행계획 연관 작업을 수행할 수 없다. MRT가 다음에 들어온다 하더라도 비행계획 연관 MRT 데이터를 저장하고 있지 않으면 다음 비행계획 연관 MRT가 수신될 때까지 기다려야 한다. 따라서 이러한 문제들을 해결하기 위해 본 논문에서는 MRT 데이터 처리부와 비행계획 연관 MRT 데이터 처리부를 각각 개별 프로세스로 분리하고 수신되는 모든 MRT 데이터를 보관하며 새로운 정보가 들어올 때마다 갱신을 수행하도록 CWP MRT 데이터 처리 모듈을 설계하였다.

표 2는 CWP MRT 데이터 처리 모듈에서 사용하는 변수들이다. *NetworkInterface*는 CIOB로부터 데이터 수신을 대기하는 오브젝트이며, 데이터를 수신한 경우 내부 큐에 데이터를 저장한다. *CorrelatedMRT*는 I062/390으로 인코딩된 비행계획 연관 MRT를 저장하는 변수이고 *stCorrelatedMRT*는 *CorrelatedMRT*를 디코딩하여 추출한 필드값을 저장하는 구조체이며 *CorrelatedMRTbuffer*는 *stCorrelatedMRT*들을 저장하는 버퍼이다. *threshold*는 비행계획 연관 MRT의 보관 시간으로 *threshold* 이상의 시간 동안 보관되어 있는 *stCorrelatedMRT*를 삭제하기 위한 조건 변수이다. *MRT*는 ASTERIX로 인코딩된 데이터를 저장하는 변수이고 *stMRT*는 MRT를 디코딩하여 필드값을 저장하는 구조체이며 *MRTbuffer*는 *stMRT*들을 저장하는 버퍼이다. *stCorrelatedMRT* 구조체는 I062/390의 필드 중 MRT 현시에 반드시 필요한 TAG, CSN, IFI, TAC, WTC, DEP, DST, RDS, CFL, CTL, PEM 필드만을 추가하며 비행계획 연관 MRT 데이터의 수신 시간을 기록하는 *InTime* 변수를 추가한다. *stMRT* 구조체는 ASTREIX에서 MRT 현시에 필요한 위치, 고도, 속도 정보 등을 포함하며 *stCorrelatedMRT*를 추가하여

표 2. CWP MRT 데이터 처리 모듈 변수
 Table 2. Variables for CWP MRT data processing module.

변수명	설명
<i>NetworkInterface</i>	네트워크 인터페이스 오브젝트
<i>CorrelatedMRT</i>	비행계획 연관 MRT (ASTERIX)
<i>stCorrelatedMRT</i>	비행계획 연관 MRT 저장 구조체
<i>CorrelatedMRTbuffer</i>	<i>stCorrelatedMRT</i> 저장 버퍼
<i>threshold</i>	비행계획 연관 MRT 보관 시간
<i>MRT</i>	MRT (ASTERIX)
<i>stMRT</i>	MRT 저장 구조체
<i>MRTbuffer</i>	<i>stMRT</i> 저장 버퍼

```

// Procedure for CorrelatedMRT
1  if ((CorrelatedMRT=read(NetworkInterface))==true)
2  then
3  if(isValid(CorrelatedMRT)==false) then
4  delete CorrelatedMRT
5  continue
6  else
7  stCorrelatedMRT=ParsingCorrelatedMRT(CorrelatedMRT)
8  stCorrelatedMRT.InTime=currentTime
9  CorrelatedMRTbuffer.insert(IF1, stCorrelatedMRT)
10 delete CorrelatedMRT
11 endif
12 endif
    
```

그림 8. 비행계획 연관 MRT 데이터 처리를 위한 의사코드
 Fig. 8. Pseudo code for processing FPL correlated MRT.

비행계획 연관 MRT 데이터블록을 현시하는 데에 이용한다. 그림 8은 비행계획 연관 MRT 데이터 처리부의 의사코드이다. *NetworkInterface*를 통해 비행계획 연관 MRT 데이터

```

// Procedure for removing stCorrelatedMRT
1 for i to CorrelatedMRTbuffer.count() do
2   stCorrelatedMRT=CorrelatedMRTbuffer[i]
3   if(currentTime - stCorrelatedMRT.InTime
4     >= threshold)
5     delete stCorrelatedMRT
6   endif
7 endif
    
```

그림 9. stCorrelatedMRT 삭제를 위한 의사코드
 Fig. 9. Pseudo code for removing stCorrelatedMRT.

```

// Procedure for MRT
1 if((MRT=read(NetworkInterface))==true)
2 then
3   if(isValid(MRT)==false) then
4     delete MRT
5     continue
6   else
7     stMRT=ParsingMRT(MRT)
8     if (isCorrelated(MRT)==true)
9       stMRT=Correlation(stMRT,stCorrelatedMRT)
10    MRTbuffer.insert(stMRT)
11  endif
12  endif
    
```

그림 10. MRT 데이터 처리를 위한 의사코드
 Fig. 10. Pseudo code for MRT.

CorrelatedMRT를 수신하면 3을 통해 데이터 길이, ASTERIX category 번호 등의 간단한 데이터 검증을 수행한다. 검증을 통해 잘못된 데이터로 판단되는 경우 4를 통해 폐기하고 수신 대기 상태를 유지한다. 데이터 검증을 통과한 경우 7을 통해 디코딩을 수행하여 필드값을 추출하고 데이터 수신 시간 *InTime* 을 포함하여 *stCorrelatedMRT* 구조체에 저장한다. *CorrelatedMRTbuffer*는 MRT 수신 시 비행계획 연관 MRT를 검색하는 데에 사용되며 *InTime*은 타임아웃 시 *stCorrelatedMRT*를 제거하는 데에 사용된다. 9를 통해 *CorrelatedMRTbuffer*에 *stCorrelatedMRT*를 저장할 때 동일 항공기에 대한 *stCorrelatedMRT*가 존재하는지 *IFI*를 이용하여 확인한다. 동일 *IFI*가 존재하지 않는 경우 *stCorrelatedMRT*를 추가하고 존재하는 경우 *stCorrelatedMRT*를 갱신한다.

그림 9는 *CorrelatedMRT*의 삭제를 위한 의사코드이다. 더 이상 사용하지 않는 비행계획 연관 MRT 데이터는 메모리 확보를 위해 제거되어야 한다. 3과 같이 현재 시간 *currentTime*과 *stCorrelatedMRT*의 *InTime*의 차이를 *threshold*와 비교하고 *threshold* 이상 시간이 지난 경우 3을 통해 *stCorrelatedMRT*를 삭제한다.

그림 10은 비행계획 연관 MRT 데이터 처리부의 의사코드이다. *NetworkInterface*를 통해 MRT 데이터를 수신하여 3을 통해 데이터를 검증한다. 오류 데이터를 발견 시 4를 통해 *MRT*를 제거한다. 검증을 통과하면 7을 통해 *MRT*를 디코딩하여

MRT 데이터의 필드값을 *stMRT* 구조체에 저장한다. 그리고 수신한 MRT의 필드 값 *TAG*, *PEM*을 이용하여 비행계획 연관 MRT 데이터가 저장된 *CorrelatedMRTbuffer*를 검색한다. 만약 *TAG*와 *PEM*이 일치하는 *stCorrelatedMRT*를 찾으면 8을 통해 *stCorrelatedMRT*를 *stMRT*에 추가하여 *MRTbuffer*에 저장하며 그렇지 않은 경우 *stCorrelatedMRT* 추가없이 *stMRT*를 저장한다. CWP 현시 모듈은 *MRTbuffer*의 상태를 지속적으로 확인하여 화면을 갱신하며 *stCorrelatedMRT*가 추가된 *stMRT*를 발견하면 그림 2.(b)의 비행계획 연관 MRT 데이터 블록을 현시하며 그렇지 않은 경우 그림 2.(a)의 MRT 데이터 블록을 현시한다.

IV. 성능평가

이 장에서는 본 논문에서 구현한 MRT 데이터 처리 모듈에 대한 성능을 평가한다. 우선 비행계획 연관 방식에 따른 네트워크 사용률과 트래픽의 공정성에 대하여 평가한다.

4-1 실험 환경

본 논문에서 구현한 CWP MRT 데이터 처리 모듈을 평가하기 위해 AICC(airport integrated communication center)의 항공교통관제 자동화 시스템 테스트 베드를 사용하였다. 총 5일간 테스트를 수행하면서 CWP MRT 데이터 처리 모듈의 비행계

표 3. 테스트 베드 서브시스템의 제원

Table 3. Specification of sub systems in testbed.

SDP, FDP	
모델	HP DL380 G7 server
CPU	인텔 Xeon 프로세서 X5670
MEM	12GB
HDD	1TB
OS	RHEL 5.8
Network interface	
모델	Cisco catalyst 3750-x
Ethernet	100BASE-T
CWP	
모델	HP Z800 workstation
CPU	인텔 Xeon 프로세서 E5640
MEM	8GB
HDD	250GB
VGA	NVIDIA quadro 2000
OS	RHEL 5.8

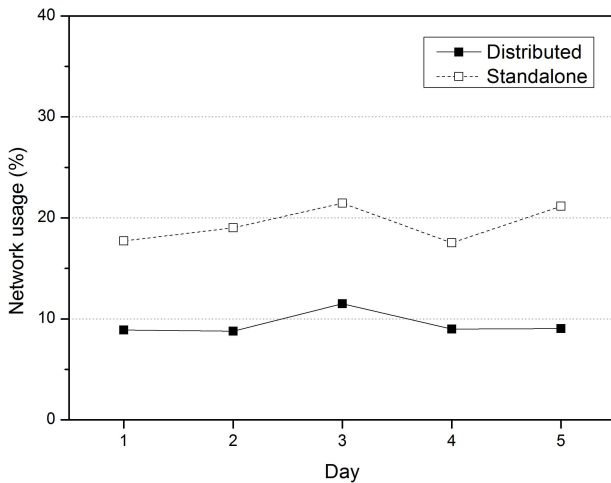


그림 11. 네트워크 사용률 비교
 Fig. 11. The comparison with network usage.

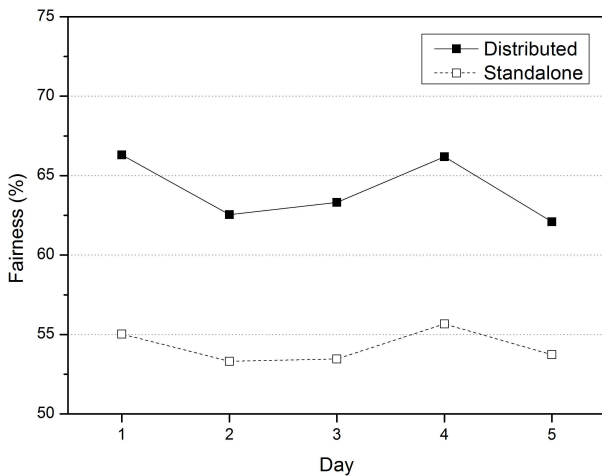


그림 12. 네트워크 트래픽 공정성 비교
 Fig. 12. The comparison with network traffic fairness.

획 연관 방식에 따른 네트워크 사용률과 트래픽 공정성을 평가하였다. CWP에 수신되는 네트워크 트래픽의 측정을 위해 패킷 분석 도구인 Wireshark[9]를 사용하였다. 테스트 베드의 구성도는 그림 1과 동일하며 비행계획 연관 MRT와 관련된 서브시스템인 SDP, FDP, network interface, CWP의 제원은 표 3과 같다.

4-2 네트워크 사용률과 트래픽 공정성 비교

그림 11은 독립형 연관 방식과 분산형 연관 방식의 네트워크 사용률을 비교한 것이다. 측정된 결과, 분산형 연관 방식이 독립형 연관 방식에 비해 약 2배 이상 네트워크 사용률이 낮았다. 독립형 연관 방식은 약 100 octets 길이를 가진 텍스트 기반의 비행계획을 사용하기 때문에 network interface를 통해 전달하는 데이터양 또한 크게 증가하게 된다. 그러나 분산형 연관

방식은 약 40 octets 길이를 가진 1062/390으로 인코딩된 비행계획 연관 MRT 데이터를 사용하므로 네트워크 사용률을 2배 이상 크게 낮출 수 있다. 따라서 비교를 통해 분산형 연관 방식이 독립형 연관 방식에 비해 네트워크 트래픽을 추가 수용할 수 있는 능력이 높으며 향후 추가되는 감시자료 및 비행자료를 효과적으로 지원할 수 있다는 것을 보여준다.

그림 12는 독립형 연관 방식과 분산형 연관 방식의 네트워크 트래픽 공정성을 비교한 것이다. 네트워크 트래픽의 공정성은 식 (1)을 통해 구할 수 있다[10].

$$Fairness = \frac{(\sum d_i)^2}{N \times \sum (d_i)^2} \quad (1)$$

N 은 네트워크 트래픽을 발생시키는 source의 개수, d_i 는 source i 의 throughput을 의미한다. 네트워크 트래픽 공정성은 각 source i 의 throughput의 편차가 클수록 낮아진다. 즉 다른 트래픽에 비해 상대적으로 높은 네트워크 점유율을 가지는 source가 존재하여 자원 공유의 공정성이 침해되는 상황이 발생한다는 것을 말한다.

측정 결과, 분산형 연관 방식은 독립형 연관 방식에 비해 8.3~11.27% 높은 트래픽 공정성을 가졌다. 이는 분산형 연관 방식이 독립형 연관 방식에 비해 네트워크 트래픽의 throughput의 편차가 크지 않기 때문이다. 독립형 연관 방식의 경우 비행계획 데이터가 전체 네트워크 사용률의 약 70%가 넘는 높은 점유율을 보이고 MRT 데이터는 전체 네트워크 사용률의 약 25%를 보였다. 반면 분산형 연관 방식의 경우 비행계획 연관 MRT 데이터가 전체 네트워크 사용률의 약 35%를 차지하고 MRT 데이터는 전체 네트워크 사용률의 약 60%를 차지하여 독립형 연관 방식에 비해 편차가 적은 것을 확인하였다. 따라서 비교를 통해 분산형 연관 방식이 독립형 연관 방식에 비해 모든 네트워크 트래픽이 공정하게 대역폭을 공유할 수 있다는 것을 알 수 있으며, 특히 비행계획 연관 MRT 데이터에 치중된 네트워크 트래픽을 줄임으로써 항공상황을 표현하는 MRT 데이터가 더 원활하게 처리될 수 있도록 함을 알 수 있다.

V. 결론

MRT와 비행계획의 연관은 안전하고 정확한 관제 업무를 위해 매우 중요한 CWP의 기능 중 하나이다. 그러나 매년 늘어나는 항공교통량으로 항공교통관제 자동화 시스템의 network interface에 부하가 증가하게 되고 이로 인해 신속하게 MRT와 비행계획의 연관 작업을 수행할 수 없는 문제가 발생한다. 본 논문에서 구현한 CWP MRT 데이터 처리 모듈은 이러한 문제를 해결하기 위해 네트워크 사용률을 줄이고 각 트래픽 간의 공정성을 확보할 수 있는 1062/390으로 인코딩된 비행계획 연

관 MRT 데이터 포맷을 사용하며 MRT와 비행계획 연관을 위해 분산형 연관 방식을 사용하였다.

구현한 CWP MRT 데이터 처리 모듈을 평가하기 위하여 인천공항 서울접근관제소 내에 구성된 항공교통관제 자동화 시스템의 테스트 베드를 이용하였다. 분산형 연관 방식은 독립형 연관 방식에 비해 네트워크 사용률이 8.54~12.11% 적었으며, 8.37~11.27% 높은 네트워크 트래픽 공정성을 가졌다.

향후 차세대 감시시스템 (ADS-B, MLAT, Mode-S 등)의 발달과 항공교통량의 증가로 관제자료의 양이 크게 증가할 것으로 예상된다. 따라서 이를 효과적으로 처리하여 항공교통관제 자동화 시스템의 전반적인 성능 향상에 기여할 수 있는 항공교통관제용 현시시스템에 대한 지속적인 연구가 필요하다.

참고문헌

[1] U. Ahlstrom, J. Rubinstein, S. Siegel, R. Mogford, and C Manning: Display Concepts For En Route Air Traffic Control, FAA, DOT/FAA/CT-TN01, 2001.

[2] T. Yuditsky, F. Friedman-Berg, and A. Smith: Design of Information Display Systems for Air Traffic Control, FAA, DOT/FAA/CT-TN04/33, 2004.

[3] S. I. Na, J. W. Lee, I. S. Won, S. B. Choi, H. D. Park and D. S. Jeong, "The research of the Control Work Position for developing ATC," in *Conference of Institute of Electronics Engineers of Korea*, Pyeongchang: Korea, pp. 1197- 1198,

2008.

[4] R. Li and Z. Zhou, "Research on the operational condition evaluation of ATC automation system," in *Quality, Reliability, Risk, Maintenance, and Safety Engineering (QR2MSE) 2013 International Conference*, Chengdu: China, pp. 2022-2025, 2013.

[5] Procedures for Air Navigation Services: Air Traffic Management, Doc 4444 ATM/501, Fifteenth Edition, ICAO, 2007.

[6] Procedures for Air Navigation Services: Air Traffic Management Amendment No.1, Doc 4444 State letter AN 13/2.1-08/50, ICAO, 2008.

[7] Eurocontrol Standard Document for Surveillance Data Exchange Part 1 : All Purpose Structured Eurocontrol Surveillance Information Exchange (ASTERIX), European Organization for The Safety of Air Navigation, SUR.ET1.ST05.2000-STD-01-01, Edition 1.30, Nov, 2007.

[8] Eurocontrol Standard Document for Surveillance Data Exchange Part 9 : SDPS Track Messages, European Organization for The Safety of Air Navigation, SUR.ET1.ST05.2000-STD-09-01, Edition 1.7, Nov, 2007.

[9] Wireshark[Internet]. Available: <http://www.wireshark.org/>.

[10] S. Y. Lee and C. K. Jeong, "Improving Fairness in an ATM-based TCP Network," in *Conference of The Korean Institute of Communications and Information Sciences*, Yangyang: Korea, pp. 905-908, 1999.



김 강 희 (Kanghee Kim)

약력

2011년 2월 : 인하대학교 전자공학과 (공학사)

2013년 2월 : 인하대학교 전자공학과 (공학석사)

2013년 3월~현재 : 인하대학교 전자공학과 박사과정

※ 관심분야 : 항공교통관제 시스템, 차세대항행시스템, 컴퓨터 네트워크



최 상 방 (Sangbang Choi)

약력

1981년 2월 : 한양대학교 전자공학과 (공학사)

1988년 : University of washington (공학석사)

1990년 : University of washington (공학박사)

1991년~현재 : 인하대학교 전자공학과 교수

※ 관심분야 : 컴퓨터 구조, 컴퓨터 네트워크, 무선통신, 병렬 및 분산처리 시스템