

효율적인 다중 버전 ADS-B 타깃 리포트 메시지 처리를 위한 모듈 설계 및 구현

Design and Implementation for Efficient Multi Version ADS-B Target Report Message Processing

김강희¹ · 장은미¹ · 송인성¹ · 조태환² · 최상방^{1*}

¹인하대학교 전자공학과

²공군사관학교 전자공학과

Kanghee Kim¹ · Eunmee Jang¹ · Inseong Song¹ · Taehwan Cho² · Sangbang Choi^{1*}

¹Department of Electronics Engineering, Inha University, Incheon 402-751, Korea

²Department of Electronics Engineering, Republic of Korea Air Force Academy, Chungcheongbuk-do 360-849, Korea

[요 약]

ADS-B (automatic dependent surveillance-broadcast)는 차세대 항행시스템 CNS/ATM (communication, navigation and surveillance/air traffic management)의 핵심기술로 위성항법시스템 정보를 이용하여 자신의 위치 정보를 자동으로 방송하며, 기존 레이더 시스템에 비해 지형에 의한 제약이 적고 갱신 속도가 빠르다는 장점을 가지고 있다. EUROCONTROL에서는 항공관제시스템에서 ADS-B 메시지를 교환하기 위하여 표준 감시자료 포맷 ASTERIX (all purpose structured eurocontrol surveillance information exchange) CAT.021을 정의하였다. 그러나 ASTERIX CAT.021은 업데이트 주기가 빠르고 이전 버전과 호환이 되지 않는 문제가 있다. 본 논문에서는 ASTERIX CAT.021의 버전과 관계없이 효율적으로 메시지를 처리하는 모듈을 설계 및 구현하였다. 구현 모듈은 네트워크로부터 수신한 메시지를 수집하여 패킷을 생성하고, 이후 수신 메시지는 패킷에 따라 정의된 루틴으로 처리한다.

[Abstract]

Automatic dependent surveillance-broadcast (ADS-B) is the core technology of communication, navigation and surveillance/air traffic management (CNS/ATM), automatically broadcasts its own position information using GNSS and has an advantage of lower geological constraints and faster update speed compared with legacy radar systems. EUROCONTROL defined all purpose structured eurocontrol surveillance information exchange (ASTERIX) CAT.021. ASTERIX CAT.021 is modified several times, but it has compatibility issues with previous version of it. In this paper, we have designed an efficient message processing module regardless of the version of ASTERIX CAT.021. This implemented module generates patterns to collect messages received from the network, after that, received messages are processed in the routine that is defined in accordance with the patterns.

Key words : ADS-B, CNS/ATM, All purpose structured eurocontrol surveillance information exchange (ASTERIX), Packet classification, Binary prefix search tree.

<http://dx.doi.org/10.12673/jant.2015.19.4.265>



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 18 June 2015; Revised 24 July 2015

Accepted (Publication) 24 August 2015 (30 August 2015)

*Corresponding Author; Sangbang Choi

Tel: +82-32-860-7417

E-mail: sangbang@inha.ac.kr

1. 서론

2014년 연간 항공교통량은 국제유가하락으로 인한 여객수요증가와 중국인 관광객의 증가로 2013년 585,017대에서 626,066대로 7% 증가하였다. 그 중 국제선은 380,743대에서 412,743대로 8.4% 증가, 국내선은 204,274대에서 213,343대로 4.4% 증가하여 특히 국제선의 증가 추세가 두드러졌다. 지난 10년간의 연간교통량을 보면 세계적으로 경제불황을 겪었던 2009년도를 제외하고 연 6%의 꾸준한 증가 추세를 보였으며, 최근 5년 동안 연 6.6% 이상 증가하면서 세계교통량 평균 예측치 4.7%보다 높은 수치를 보였다. 국내 항공 교통량이 집중되는 시간은 오전 10시로 FIR (flight information region) 내에 시간당 139대로 전체 하루 평균 1715대의 12.3%를 기록하였다. 그리고 Airbus 사의 세계항공시장 전망보고서 2014-2033은 유상여객킬로미터 (RPK; revenue passenger kilometer)가 향후 20년간 연 4.7% 증가하여 2028년에는 2013년의 2배가 될 것으로 전망하였다[1]-[2].

이처럼 해마다 늘어나는 항공교통량에 대비하여 항공교통을 효율적, 경제적, 안정적으로 처리할 수 있는 기술이 필요하게 되었고 이에 1983년 국제민간항공기구 (ICAO; international civil aviation organization)에서는 FANS (future air navigation system)라는 특별 위원회를 만들어 관련 연구를 수행하였으며, 1991년 10차 항행회의에서는 FANS에서 제안한 위성항법기술과 데이터링크 통신 기술을 기반으로 한 새로운 개념의 차세대 항행시스템 CNS/ATM(communication, navigation and surveillance/air traffic management)을 승인하고 회원들이 개발 및 구축하도록 권고하였다.

ADS-B (automatic dependent surveillance-broadcast)는 CNS/ATM의 핵심 기술로 위성항법시스템 (GNSS; global navigation satellite system)으로부터 얻은 항공기 위치 정보와 항공기 탑재 장비로부터 얻은 항공기 속도, 고도, 방향 등의 정보를 합하여 무선 데이터링크를 통해 주변에 방송한다. 그러므로 지상에서 항공교통업무를 수행하는 관제사뿐만 아니라 ADS-B를 장착한 모든 항공기의 조종사 또한 정보를 공유할 수 있어 관제업무의 효율성을 극대화할 수 있다. 세계 각국에서는 이미 2025년 항행시스템 전환을 목표로 ADS-B 설치 의무화를 추진 중에 있다. 우리나라 또한 국토해양부 주관 하에 체계적인 계획 수립과 지속적인 기술 개발을 수행하고 있다[3].

EUROCONTROL은 관제시설 간에 감시자료를 교환하기 위하여 ASTERIX (all purpose structured eurocontrol surveillance information exchange)라는 메시지 포맷을 표준으로 제정하였다. ASTERIX는 기존의 레이더 감시자료를 포함한 여러가지 감시자료를 표현할 수 있다. 그 중 ADS-B 타깃 리포트 메시지는 ASTERIX 카테고리 번호 21(CAT.021)을 가지며 고유 식별 ID, 위치, 속도, 고도, 방향, 센서 정확도, 탑재 장비 유무 및 상태 정보, 경고 상태 등 다양한 정보를 포함한다. ASTERIX CAT.021은 1999년 12월 처음 발표되었으며 ADS-B 기술의 발

전에 따라 자연스럽게 업데이트가 이루어졌다. 그러나 약 15년 동안 총 34번의 수정 버전이 발표되는 등 업데이트가 잦고 최신 버전과 이전 버전이 호환되지 않는 문제가 있다. 이러한 ASTERIX CAT.021 버전의 호환 문제로 인해 ADS-B 지상국과 관제시스템을 구축할 경우 미리 사용할 ASTERIX CAT.021 버전을 정의한다. 그러나 ASTERIX CAT.021 버전 업데이트 시 기존 장비 및 소프트웨어를 그대로 사용할 수 없기 때문에 장비 교체 또는 소프트웨어 업그레이드를 수행해야 한다. 또는 수신한 ASTERIX CAT.021 메시지를 기존 시스템과 호환되는 버전으로 변환하는 중개 시스템을 새로 구축해야 하는데 이 방식을 사용할 경우 추가 설치비용이 발생하며 처리지연시간이 증가하여 전체 시스템의 성능이 저하될 수 있다.

본 논문에서는 ASTERIX CAT.021 ADS-B 타깃 리포트 메시지의 버전 호환 문제를 효과적으로 해결하고 중개 시스템으로 인한 처리 지연을 줄임으로써 수신 메시지를 효율적으로 처리할 수 있는 다중 버전 ADS-B 타깃 리포트 메시지 처리 모듈을 설계 및 구현하였다. 설계 모듈은 수신 ASTERIX CAT.021 메시지를 학습하여 패턴을 생성한다. 이후 수신하는 ASTERIX CAT.021 메시지를 패턴에 의해 분류하여 해당 버전과 호환되는 디코더로 전송함으로써 버전 검사를 위해 필요한 시간을 크게 줄일 수 있다. 그리고 설계한 모듈은 우선순위 기반의 결정 트리를 사용함으로써 패턴 검색에 걸리는 시간을 단축하였다.

본 논문에서 설계한 다중 버전 ADS-B 타깃 리포트 메시지 처리 모듈을 평가하기 위하여 ASTERIX CAT.021을 교환하는 관제시스템을 모의한 테스트베드를 구성하고 처리지연시간, CPU 및 메모리 사용률을 측정하였다. 실험결과, 설계한 모듈은 ADS-B 지상국으로부터 수신한 다른 버전의 ASTERIX CAT.021을 효과적으로 처리하는 것을 확인하였다.

본 논문은 다음과 같이 구성된다. II장에서는 ADS-B와 관련된 연구를 소개하고, III장에서는 본 논문에서 설계한 다중 버전 ADS-B 타깃 리포트 메시지 처리 모듈의 구조와 동작 과정을 설명한다. IV장에서는 실험을 통해 설계 모듈의 성능을 평가하고, V장에서는 결론을 맺는다.

II. 관련연구

2-1 ADS-B

ADS-B는 항공기와 항공기간 또는 항공기와 지상국간에 항공기의 정보를 방송하는 감시 시스템이다. 질문 신호를 요구하는 기존의 레이더 시스템인 이차감시레이더 (SSR; secondary surveillance radar)와는 달리 자동으로 자신의 위치, 속도, 고도, 방향 등의 정보를 생성한다. Automatic은 사람이 아닌 기계에 의해 자동으로 감시자료가 보고되는 것을 의미하고, dependent surveillance는 감시자료가 가진 정보는 탑재 장비에 의해 결정된다는 의미하며, broadcast는 일정 주기를 가지고 감시자료를 무선으로 송출한다는 것을 의미한다.

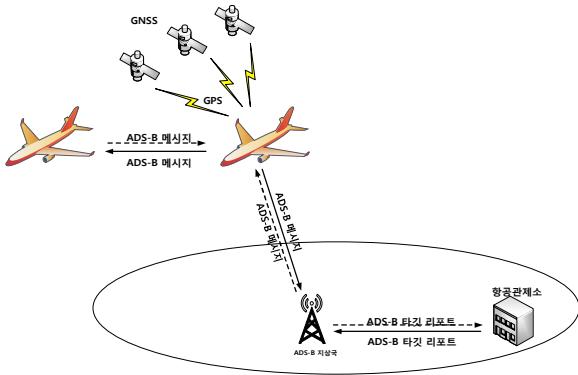


그림 1. ADS-B 기본 개념도
Fig. 1. A concept of the ADS-B.

그림 1은 ADS-B의 기본 개념을 보여주는 그림이다. 항공기에 장착된 ADS-B 메시지 송신기는 1090ES (extended squitter), UAT (universal access transceiver), VDL mode 4 (very data link mode 4) 등과 같은 무선 데이터 링크를 이용하여 주기적으로 ADS-B 메시지를 방송하며, 주변 항공기들은 ADS-B 메시지 수신기로 ADS-B 메시지를 수신한다. 이 때 항공기 안의 조종사는 CDTI (cockpit display of traffic information)를 통해 자신의 위치 뿐만 아니라 주변에 있는 항공기에 대한 위치, 고도 및 비행 의도를 쉽게 인지할 수 있다. 지상에서는 ADS-B 메시지를 송수신하는 ADS-B 지상국이 있다. 지상국은 ADS-B 메시지를 수신하여 관제업무에서 사용 가능한 ADS-B 타겟 리포트 메시지로 변환하고 이를 항공관제소로 전송한다[4].

국제민간항공기구는 급격히 증가하는 항공교통량에 의한 항행안전에 대비하여 2025년을 목표로 ADS-B의 구축 완료를 권고하고 있다. 이에 미국의 NextGen (next generation air transportation system)[5],[6], 유럽의 SESAR (single european sky ATM research) 등은 ADS-B를 차세대 항행시스템의 핵심 기술로 정하여 시설 구축 및 설치 의무화를 적극 추진하고 있다. 우리나라의 경우 1998년 인천 항공교통센터에 ADS-B 시스템을 구축하여 시험 운용을 시작하였으며, 국토해양부의 주관으로 차세대 항행안전시설 종합발전계획과 국가 위성항법시스템 기본발전계획을 각각 수립함으로써 ADS-B를 포함한 차세대 항행시스템 개발을 적극 추진하였다. 이러한 노력의 결과로 2014년에 개발된 1090ES ADS-B 지상국이 김포공항에 실제 설치되어 시범운영 중에 있다[7].

2-2 ASTERIX CAT.021 ADS-B 타겟 리포트 메시지의 생성, 처리, 공유

ASTERIX는 항공교통관제 자동화시스템에서 감시자료를 교환하기 위하여 사용하는 이진화 메시지 포맷으로 EUROCONTROL에 의해 개발되고 표준화되었다. 그 중 ASTERIX CAT.021은 ADS-B 타겟 리포트 메시지를 공유하기 위해 전 세계적으로 채택된 표준 데이터 포맷으로 ADS-B 타겟

리포트 메시지를 ADS-B 지상국에서 항공관제자동화시스템으로 전달하기 위해 사용한다. 이 메시지 포맷은 항공기의 위치, 속도, 고도, 방향, 상승하강률 뿐만 아니라 ID, 플랫폼 종류(지상/공중), 중량, 탑재 센서의 측량 정확도, 경고 상태와 같은 항공기 고유의 정보를 포함한다.

ASTERIX CAT.021은 최초 발표된 1999년 12월부터 지속적으로 업데이트되었으며 최신 버전은 2015년 1월에 발표된 v2.3이다[8]. 버전 번호는 소수점을 구분자로 하여 “2”는 주버전, “3”은 부버전이다. ASTERIX CAT.021은 약 14년간 총 34번의 업데이트가 이루어졌으며 레이더 기반의 감시자료를 포함하는 ASTERIX CAT.062의 약 15년간 총 38번의 업데이트 수준과 비슷하다[9]. 이처럼 ASTERIX CAT.021은 ADS-B 기술 발전과 추가 요구사항에 맞추어 활발한 수정 작업이 진행되어 왔으나 이로 인하여 이전 버전과 호환되지 않는 문제가 발생하였다. 따라서 ADS-B 지상국 서버 또는 ADS-B 타겟 리포트 메시지를 다루는 감시자료처리시스템 및 모니터링 시스템을 개발하는 ANSP(air navigation service provider)는 ASTERIX CAT.021의 빈번한 버전 업데이트와 호환성 문제를 해결하기 위한 별도의 추가 개발 작업을 수행하여야 한다.

국제민간항공기구의 ADS-B 연구개발 특별위원회 SITF(study and implementation task force)의 11번째 세미나에서는 ASTERIX CAT.021의 생성, 처리, 공유에 대한 지침이 제시되었다. 1090ES ADS-B 데이터 포맷을 정의하는 RTCA(radio technical commission for aeronautics) DO260 시리즈에 초점을 맞춘 가이드라인은 시스템 설계 및 생산 시 ASTERIX CAT.021의 버전 선택, 지상국 및 항공관제자동화시스템에서의 ASTERIX CAT.021 처리, ASTERIX CAT.021 메시지 변환 및 필터링 시스템의 필요성에 대한 내용을 포함하고 있다[10].

2008년 8월 발행된 ASTERIX CAT.021 v1.0이 DO260A를 포함하고 있으며, 2011년 5월에 발행된 v2.1이 DO260B를 포함하고 있다. 국제민간항공기구 아태지부에서는 아태 지역 내 ADS-B 지상국의 상호 운용을 위하여 2005년 8월에 개최한 16번째 APANPIRG(asia/pacific air navigation planning and implementation regional work group) 회의에서 DO260 표준을 포함하고 있는 ASTERIX CAT.021 v0.23을 ADS-B 지상국의 설치와 ADS-B 데이터를 공유하기 위한 기준 ADS-B 데이터 포맷으로 채택하였다. v0.23은 관제업무 시 항공기 3-5NM 간격 분리를 가능하게 해주며, DO260, DO260A, DO260B를 함께 사용할 수 있어 활용도가 높다. 그러나 DO260A와 DO260B에만 포함되어 있는 몇 가지 특수한 데이터 아이템들은 v0.23이 포함할 수 없다는 단점이 있다. 따라서 ASTERIX CAT.021에 대하여 자국 내 관제시설에서의 사용, 타 관제시설 또는 ANSP 제품과의 데이터의 공유를 고려할 때 취할 수 있는 버전 선택 방안은 표 1과 같다.

아태 지역에서는 v0.23을 기본 표준으로 사용하기로 합의하였기 때문에 자국 내에서 v0.23을 사용한다면 타 관제시설과의 데이터 공유를 위해 별도의 조치를 취할 필요가 없다. 그러나 v2.1을 자국 내에서 사용하기 원한다면 타 관제시설과 ANSP

표 1. ASTERIX CAT.021 데이터 공유를 위한 버전 선택 방안
Table 1. Choices of ASTERIX CAT.021 version for data sharing.

선택	자국 내 사용	데이터 공유	고려사항
1	v0.23	v0.23	기본 표준. 고려사항없음
2	v2.1	v0.23	v2.1 사용을 위하여 ADS-B 포맷 변환 또는 필터링 시스템이 요구됨. 만약 자국에서 특수한 DO260A/B 데이터 아이템들을 요구한다면 v0.23을 사용하는 외부 소스의 데이터 스트림으로부터 해당 아이템을 추가하는 것이 불가능하기 때문에 공유가 어렵다.
3	v2.1	v2.1	의도하는 목적을 달성하기 한 정확한 버전 사용을 위해 반드시 상호 협의가 있어야 한다.

제품과의 데이터 공유를 위해 외부로 송신하는 데이터는 v2.1에서 v0.23으로, 외부로부터 수신한 데이터는 v0.23에서 v2.1로 변환하는 시스템이 필요하다. 만약 데이터 변환 없이 v2.1을 그대로 공유하기를 원한다면 상대방 시스템과의 협의를 통해 사용 버전을 통일해야 한다.

ADS-B 지상국 설계자는 ADS-B 지상국에서 항공기로부터 받은 DO260, DO260A, DO260B를 어떻게 처리하여 ASTERIX CAT.021에 넣을 것인지, ASTERIX CAT.021의 데이터 항목 또는 필드가 어떤 것이 사용되고 얼마나 사용할 것인지를 결정해야 한다. 항공관제자동화시스템 설계자는 ADS-B 지상국으로부터 수신한 ASTERIX CAT.021의 어떤 데이터 항목과 필드를 선택적으로 처리할 것인지, 어떤 데이터가 시스템에 요구되는지를 결정해야 한다. 그리고 ADS-B 지상국, 항공관제자동화시스템, 백업시스템, 외국의 ADS-B 관련 시스템 간에 ADS-B 타깃 리포트 메시지를 교환하기 위해 어떤 버전을 사용할 것인지를 결정해야 한다. ASTERIX CAT.021은 필수 필드와 다수의 옵션 필드로 구성되어 있으며, 옵션 필드는 항공기 탑재장비로부터 데이터를 수신한 경우 또는 지상국에서 옵션 필드의 추가를 요구하는 경우에만 사용할 수 있다. 항공기 탑재 장비가 ADS-B 지상국에 송신하는 데이터는 부가적인 사용 목적과 녹화분석 용도를 위해 반드시 항공관제자동화시스템에 전달되어야 한다. 만약 업데이트 주기 동안에 항공기 탑재 장비가 송신하는 데이터를 수신하지 못하여 항공관제자동화시스템에 전달할 데이터가 없는 경우 대역폭을 줄이기 위해 해당 항공기에 대한 ASTERIX CAT.021 메시지를 전송 스트림에 포함시키지 않는다.

지침서에서는 중요도에 따라 ASTERIX CAT.021 데이터 항목을 표 2와 같이 4가지 그룹으로 분류하였다. 그룹 1은 필수 데이터 항목으로 데이터 소스 ID, 시간, 품질지표, 위치 정보 등

표 2. ASTERIX CAT.021 데이터 항목 그룹
Table 2. ASTERIX CAT.021 data item group.

그룹	데이터 항목
그룹1 (필수 데이터 항목)	데이터 소스 ID, 시간, 타깃 리포트 기술자, 타깃 주소, 품질지표, 위치 정보
그룹2 (가치있는 데이터 항목)	항공기 운영 상태, 방사체 범주, 고도, 상승하강률, 지상벡터, 타깃 ID, 타깃 상태
그룹3 (부가데이터 항목)	리포트 전송 시간, 날짜 정보 정확도, 속도 정보 정확도, 속도 정보 가용 시간, 속도 정보 수신 시간, 트랙 번호, 궤적 정보, 서비스 유형, 위치 정보 정확도, Mode 3/A Code, 회전 각도, 속도, 기상 리포트 등
그룹4 (미래 데이터 항목)	타깃 헤딩, GPS 오프셋

을 포함하고 있으며 ASTERIX CAT.021 메시지 포맷 안에 반드시 포함되어 있어야 한다. 만약 그룹 1의 데이터 항목이 하나라도 존재하지 않는다면 해당 메시지를 전송하지 않는다. 그룹 2는 가치 있는 데이터 항목으로 필수 항목은 아니지만 항공관제 자동화시스템에서 유용하게 사용되는 탑재 장비, 플랫폼 정보, 고도, 경고 상태 등을 포함하고 있다. 그룹 3은 부가 데이터 항목으로 데이터의 사용 능력 또는 특별한 요구에 따라 전달될 필요가 있는 탑재 장비 센서의 정확도, 사용 무선 링크 종류, SSR-CODE, 속도 등을 포함하고 있다. 그룹 4는 미래 데이터 항목으로 DO260A나 DO260B 표준 안에서 정의되어 있으나 ASTERIX CAT.021에 정의되지 않은 타깃 헤딩이나 GPS 오프셋과 같은 항목들이다. 지침서에서는 ADS-B 지상국을 설계할 때, 그룹 1과 함께 그룹 2의 모든 데이터를 ASTERIX CAT.021에 포함할 것을 권장하고 있으며 그룹 3의 경우 요구에 따라 설정 가능해야 한다고 언급하고 있다.

2-3 ADS-B 포맷 변환과 필터 시스템

ADS-B 기술은 미래에도 지속 발전할 것이며 ANSP들은 추가되는 기능들을 제공하기 위해 꾸준히 제품을 업그레이드 할 것이다. 그 결과 ASTERIX CAT.021 표준 또한 계속 진화할 것이다. 그러므로 새로운 기능 요구 사항과 새로운 메시지 포맷 표준에 대처할 수 있는 능력이 반드시 필요하다. 이러한 문제를 해결하기 위하여 SIFT의 ASTERIX CAT.021 생성, 처리, 공유에 대한 지침은 그림 2와 같이 ASTERIX CAT.021 메시지 공유를 위한 변환 및 필터링 시스템 추가를 제안하고 있다.

ANSP “A”와 ANSP “B”간에 다른 ASTERIX CAT.021 버전을 사용하고 있더라도 중간에 ADS-B 포맷 변환 및 필터 시스템이 각 ANSP의 항공관제자동화시스템 및 ADS-B 지상국이 사용하는 버전으로 변환하기 때문에 외부 환경의 변화와 관계 없이 독립적으로 환경을 구성할 수 있다는 장점이 있다. 그러나 데이터의 변환 및 필터를 위한 오버헤드로 인해 추가 지연시간

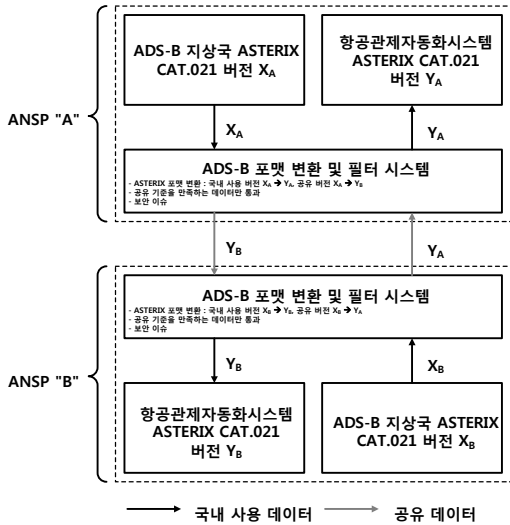


그림 2. ASTERIX CAT.021의 생성, 처리, 공유
 Fig. 2. Generation, processing and sharing of ASTERIX CAT.021.

이 발생하여 전체적인 시스템 성능을 떨어뜨릴 수 있다. 또한 내외부 ANSP의 시스템들이 독립적으로 운영 가능하더라도 ASTERIX CAT.021의 버전이 변경되는 경우 ADS-B 포맷 변환 및 필터 시스템의 업그레이드 및 교체가 반드시 필요하므로 이에 대한 추가 비용이 발생한다. 따라서 본 논문에서는 변환 및 필터 시스템 설치로 인한 비용 및 전송 지연을 감소시키기 위해 별도의 중계시스템 없이 자체적으로 패킷을 분류하는 방식을 사용한다.

2-4 패킷 분류 알고리즘

패킷 분류란 수신 패킷을 정의된 클래스로 분류하는 것을 말하며 주로 인터넷 라우터에 의해 서비스된다. 인터넷 라우터는 패킷을 수신하면 헤더 정보를 읽어 자신이 가지고 있는 헤더 테이블에 일치하는 클래스가 존재하는지 검색한다. 만약 수신 패킷의 헤더 정보가 저장된 클래스의 헤더 정보와 일치하면 해당 클래스에 정의된 서비스를 제공한다. 패킷 분류는 단순히 패킷 포워딩을 위한 정보 제공 뿐만 아니라 외부 공격을 차단하기 위한 방화벽 설정이나 허가된 사용자만이 접근할 수 있는 리스트 생성 등과 같은 다양한 어플리케이션에서 사용할 수 있다[11].

일반적으로 패킷 분류는 어려운 문제이다. 패킷 헤더는 다수의 필드를 가지고 있기 때문에 수신 패킷이 클래스와 일치하는지를 알아내려면 패킷이 가지고 있는 필드 정보를 테이블에서 모두 검색하여 일치 여부를 확인해야 한다. 그리고 패킷 헤더의 필드별 검색 방식이 다르다. 예를 들어 IP 주소 필드는 최장 길이 일치, 포트번호 필드는 영역 일치, 프로토콜 필드는 완벽하게 값이 일치해야 한다. 만약 이러한 필드 정보들이 다수의 클래스와 일치한다면 각 클래스의 우선순위를 추가 비교해야 한

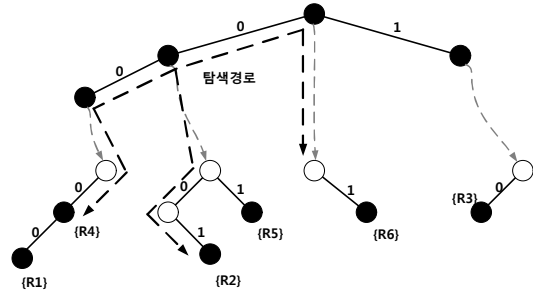


그림 3. 계층적 트라이
 Fig. 3. Hierarchical trie.

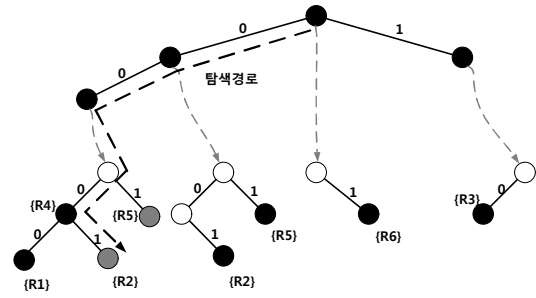


그림 4. 계층적 셋-프루닝 트라이
 Fig. 4. Hierarchical set-pruning trie.

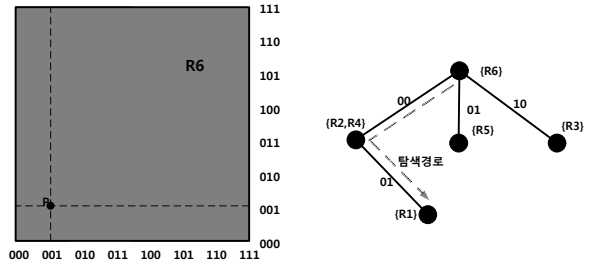


그림 5. 영역 분할 사분 트라이
 Fig. 5. Area-based quad trie.

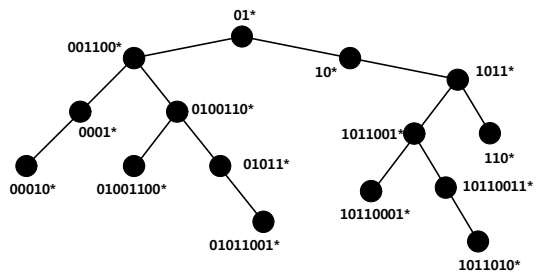


그림 6. 이진 프리픽스 검색 트리
 Fig. 6. Binary prefix search tree.

다. 최근 인터넷 속도가 기가 비트, 테라 비트 등 고속 인터넷으로 진화하고 있으므로 이에 맞추어 패킷 분류 성능 또한 향상시켜 전송속도를 유지할 수 있도록 해야 한다[12],[13].

이러한 패킷 분류의 어려움을 해결하기 위하여 다양한 알고리즘이 연구되어 왔다. 대부분의 알고리즘은 빠른 패킷 분류를

위하여 근원지 IP 주소와 목적지 IP 주소, 이 두 개의 필드만을 이용하여 계층화된 트라이를 구성하고 패킷 수신 시 트라이를 통해 클래스를 찾아내는 방식을 사용한다. 이는 IP 주소에 의한 분류가 나머지 필드에 의한 분류보다 확률적으로 더 많은 경우의 수를 가지고 있음에 기인한다.

그림 3은 IP 주소 필드를 이용한 계층적 트라이의 구조이다. 계층적 트라이는 근원지 IP에 대한 트라이와 목적지 IP에 대한 트라이를 별도로 구성하고 이를 연결한 구조를 가지고 있다. 각 트라이는 MSB(most significant bit)에서부터 한 비트씩 관찰하여 0이면 왼쪽, 1이면 오른쪽으로 노드를 확장하는 구조를 가지고 있다. 예를 들어 IP 주소 값이 “100*”이면 루트에서 오른쪽(1)으로, 다음 노드에서 왼쪽(0)으로, 다음 노드에서 왼쪽(0)으로 노드를 이어나가면 마지막 노드는 프리픽스 노드가 되며 목적지 IP 트라이와 연결된다. 검색을 통해 근원지 IP 트라이에서 근원지 IP와 프리픽스 값이 일치하는 노드를 발견하면 그 프리픽스 노드와 연결된 목적지 IP 트라이로 진입하여 목적지 IP 검색을 수행한다. 목적지 IP 트라이에서 일치하는 프리픽스를 찾으면 나머지 3개 필드값(근원지 포트번호, 목적지 포트번호, 프로토콜 번호)을 비교하여 최종적으로 클래스를 결정한다. 만약 최종 노드 위치에서 여러 개의 클래스가 동시에 존재하는 경우에는 우선순위가 높은 것을 따른다. 계층적 트라이는 트라이의 생성과 탐색 알고리즘이 단순하다는 장점이 있으나 트라이 생성 시 발생하는 빈 노드와 상위 트라이로의 역추적으로 인해 메모리 접근이 늘어나 검색 속도가 떨어진다.

그림 4는 IP 주소 필드를 이용한 계층적 셋-프루닝 트라이의 구조이다. 계층적 셋-프루닝 트라이는 계층적 트라이에서 역추적에 의한 검색 시간 지연을 제거하여 성능을 향상시킨 것이다. 계층적 트라이에서 상위 트라이의 프리픽스 노드와 연결된 모든 하위 트라이를 복사시킨다. 이로 인해 상위 트라이에서 일치하는 프리픽스 노드가 다수 존재하는 경우에서 상위 트라이로 올라갈 필요 없이 하위 트라이에서 검색을 가능하게 해 준다. 프리픽스 노드 중 회색으로 나타낸 노드가 복사된 노드이다. 검색 성능은 계층적 트라이보다 향상되지만 하나의 규칙에 대하여 다수의 복사가 이루어져 노드가 많아질수록 메모리 사용량이 크게 증가하며 여전히 빈 노드가 존재하여 불필요한 메모리 접근이 발생한다.

영역 분할 사분 트리 (AQT; area-based quad tree)는 근원지 IP와 목적지 IP로 새로운 프리픽스인 튜플을 이용하여 사분 트라이 구조를 만든다. 각 클래스들은 하나의 평면에 분할된 영역을 차지하게 된다. 그림 5는 영역 분할 사분 트라이의 구조를 나타낸다. W를 프리픽스의 최대 길이라고 할 때, 루트 노드는 $2W*2W$ 의 정사각형 평면으로 표현되며, 자식 노드 4개는 정사각형을 사분할한 평면에 각각 대응된다. 이와 같은 과정을 반복하면 그림 5와 같은 이차원 트라이를 완성할 수 있다. 근원지 IP 프리픽스 길이와 목적지 IP 프리픽스의 길이가 같을 경우 영역은 정사각형으로 표현된다. 예를 들어 근원지 IP와 목적지 IP의 튜플이 00,01,10,11 조합을 가질 때 그려지는 영역은 R1에 속하므로 수신 패킷을 규칙 R1에 따라 처리한다. 만약 근원지 IP 프

리픽스 길이와 목적지 IP 프리픽스의 길이가 다른 경우 교차곱(crossproduct)을 수행하여 직사각형 영역으로 표현한다. 영역 분할 사분 트라이 구조는 계층적 트라이 구조보다 검색이 효율적이나 여전히 빈 노드가 존재하며 프리픽스의 길이가 증가함에 따라 메모리 효율성이 떨어지게 된다.

Yazdani의 이진 프리픽스 검색 트리는 본디 문자열을 검색하는 용도로 만들어졌으나 이진 데이터의 프리픽스값을 사용하므로 패킷 분류 알고리즘에 적용 가능하다[14]. 이진 프리픽스 검색 트리를 생성하려면 우선 프리픽스의 길이가 다른 문자들을 크기순으로 정렬한 후 중간값으로 프리픽스 리스트를 양분한다. 이후 중간값 정의, 리스트 양분을 반복적으로 수행하여 트리를 완성시킨다. 그림 6은 이진 프리픽스 검색 트리의 구조를 나타낸다.

Yazdani의 이진 프리픽스 검색 트리는 문자 값의 크기를 이용하므로 프리픽스의 길이와 상관없이 조희가 가능하고 역추적 없이 빠른 검색을 가능하게 한다. 이러한 장점 때문에 다른 분류 알고리즘에 비해 약 2~4배 검색 성능이 높다는 것을 [14]의 실험결과를 통해 확인할 수 있다. 따라서 본 논문에서는 이진 프리픽스 검색 트리를 ASTERIX 포맷 특성에 맞게 수정하여 ASTERIX CAT.021의 버전을 판단하는 결정트리로 사용한다.

2-5 ADS-B 타깃 리포트 메시지 처리 방법

ADS-B 타깃 리포트 메시지로부터 시스템이 필요로 하는 데이터를 추출하기 위해서는 ASTERIX 표준에서 정의하는 규칙에 따라 디코딩을 수행해야 한다[15]. asterixinspector[16], vog asterix[17], croatia control asterix[18]와 같은 opensource들은 ADS-B 타깃 리포트 메시지 처리 방법을 보여주는 좋은 예이다. 이들은 본래 ASTERIX 데이터를 분석하기 위한 도구로 개발된 것이지만 ASTERIX CAT.021 포맷을 가지는 ADS-B 타깃 리포트 메시지를 처리하는 시스템에 적용 가능하다.

Asterixinspector는 Qt로 개발되었으며 raw data 저장 파일을 읽어 분석 결과를 GUI 창에 테이블과 웹문서로 보여준다. vog asterix는 Volker Grabsch라는 개발자가 개인 git 저장소를 통해 공개한 소스이다. 프로그래밍 언어는 python이며 raw data 저장 파일을 읽어 처리한 결과를 터미널 창에 출력한다. croatia control asterix는 croatia control Ltd.라는 회사에서 공개한 소스로 C/C++로 개발되었으며 패킷 캡처 데이터와 멀티캐스트로 수신한 데이터를 입력 데이터로 사용하며 분석 결과를 text, xml(extensible markup language), json(javascript object notation)으로 출력한다. 이들은 사용한 프로그래밍 언어는 각각 다르지만 xml 파일을 이용하여 ASTERIX 포맷을 정의한다는 공통점을 가지고 있다. xml 파일에는 데이터 항목의 순서와 크기, 유형(숫자, 문자, 문자열 등)의 정보가 포함되어 있다. 프로그램 시작 시 xml 파일 안에 포함된 메타데이터들을 이용하여 카테고리 번호를 고유키로 가지는 탐색 테이블을 생성한다. 이후 입력 데이터의 카테고리 번호와 키값이 일치하는 테이블을 찾으면

세부 테이블로 들어가 데이터 항목들을 추출한다.

이들의 ADS-B 타깃 리포트 메시지 처리 방법은 xml의 사용으로 인해 ASTERIX 포맷이 변경되거나 새로운 카테고리가 추가되더라도 프로그램을 변경할 필요가 없다는 장점이 있다. 그러나 카테고리 번호를 고유키로 가지는 테이블로는 ADS-B 타깃 리포트 메시지와 같이 카테고리 번호가 같지만 호환이 되지 않는 버전을 동시에 처리할 수 없다. 호환 문제를 해결하기 위해서는 수신 ADS-B 타깃 리포트 메시지의 버전이 변경될 때마다 시스템 관리자가 매번 xml 파일을 바꾸고 프로그램을 재시작해야 한다. 이는 상시 시스템이 가동되어야 하는 항공교통관제 자동화시스템에 적합하지 않다. 이에 본 논문에서는 xml 파일의 변경이나 시스템의 재기동 없이 다중 버전 ADS-B 타깃 리포트 메시지를 처리할 수 있는 방법을 제안한다.

III. 다중 버전 ADS-B 타깃 리포트 메시지 처리 모듈 설계 및 구현

3-1 ASTERIX CAT.021 포맷 특징

ADS-B 지상국 또는 ADS-B 감시자료처리시스템 등의 ANSP 제품들은 프로그램 되어 있는 규칙에 따라 ASTERIX CAT.021을 생성한다. 이러한 특징 때문에 ASTERIX CAT.021 데이터는 일정한 비트값을 가지는 필드가 반복적으로 나타나게 되며 이 값들을 추출하여 패턴으로 사용할 수 있다. ASTERIX CAT.021 데이터 포맷의 구조는 그림 7과 같다[15].

본 논문에서는 카테고리번호, 길이, 필드상세(FSPEC; field specification)를 패턴을 형성하는 특징으로 정의한다. 카테고리번호는 데이터의 사용 목적 또는 어플리케이션을 정의하는 것으로 ADS-B 타깃 리포트 메시지는 고유번호 21번을 가진다. 길이는 데이터의 전체 길이를 정의하는 것으로 0~2⁸-1 범위를 가진다. 필드상세는 ASTERIX 포맷 내부의 데이터 항목 구조를 정의하는 것으로 비트 값의 유무에 따라 데이터 항목의 존재 유무가 결정되며 이로 인해 데이터의 길이가 가변하므로 데이터를 구별하는 좋은 특징이 된다. EUROCONTROL에서 공개한 ASTERIX CAT.021 중 v0.1에서 v0.26까지는 v0.26[19]과 호환되며, v1.4에서 v2.1까지는 v2.1[20]과 호환된다. 표 3은 v0.26과 v2.1의 데이터 항목들을 필드상세 순으로 정렬하여 비교한 것이다.

표 3을 보면 v0.26과 v2.1은 I021/010 데이터 소스 ID와 I021/040 타깃 리포트 기술자를 제외한 모든 데이터 항목이 다른 순서로 이어져 있음을 확인할 수 있다. 그리고 I021/040 타깃 리포트 기술자는 같은 위치에 놓여 있지만 데이터 크기가 다른 것을 확인할 수 있다. 데이터 항목 오른쪽 상단의 *는 ASTERIX CAT.021 표준 문서에 필수 데이터 항목으로 분류된 것을 표시한 것이다. v0.26의 경우 I021/010 데이터 소스 ID, I021/040 타깃 리포트 기술자, I021/030 시간, I021/080 타깃 주

길이	Octets : 1	2	가변길이	가변길이
항목	카테고리 번호(21)	길이	필드상세	데이터 항목

그림 7. ASTERIX CAT.021 데이터 포맷

Fig. 7. Data format of ASTERIX CAT.021.

표 3. ASTERIX CAT.021 v0.26과 v2.1 비교표

Table 3. Table for comparison ASTERIX v0.26 with v2.1.

번호	v0.26			v2.1		
	데이터 항목	정보	길이	데이터 항목	정보	길이
1	I021/010*	데이터 소스 ID	2	I021/010*	데이터 소스 ID	2
2	I021/040*	타깃 리포트 기술자	2	I021/040*	타깃 리포트 기술자	1+
3	I021/030*	시간	3	I021/161	트랙번호	2
4	I021/130	위치	8	I021/015	서비스 ID	1
5	I021/080*	타깃 주소	3	I021/071	위치정보이용 가능시간	3
6	I021/140	기하고도	2	I021/130	위치	6
7	I021/090	감도 지수	2	I021/131	위치(고해상도)	8
FX	-		-	-		-
8	I021/210*	링크 기술	1	I021/072	속도정보이용 가능시간	3
9	I021/230	롤 각도	2	I021/150	속도	2
10	I021/145	고도	2	I021/151	진 속도	2
11	I021/150	속도	2	I021/080*	타깃 주소	3
12	I021/151	진 속도	2	I021/073	위치정보수신 시간	3
13	I021/152	자북기준방향	2	I021/074	위치정보수신 시간(고정밀)	4
14	I021/155	기압수직률	2	I021/075	속도정보수신 시간	3
FX	-		-	-		-
15	I021/157	기압수직률	2	I021/076	속도정보수신 시간(고정밀)	4
16	I021/160	지상벡터	4	I021/140	기하고도	2
17	I021/165	방향전환률	1+	I021/090*	품질지표	1+
18	I021/170	타깃 ID	6	I021/210	MOPS 버전	1
19	I021/095	속도 정확도	1	I021/070	Mode 3/A	2
20	I021/032	시간 정확도	1	I021/230	롤 각도	2
21	I021/200	타깃 상태	1	I021/145	고도	2
FX	-		-	-		-
22	I021/020	방사체 범주	1	I021/152	자북기준방향	2
23	I021/220	기상정보	1+	I021/200	타깃 상태	1
24	I021/146	중간상태선택 고도	2	I021/155	기압수직률	2
25	I021/148	최종상태선택 고도	2	I021/157	기압수직률	2
26	I021/110	궤도계획	1+N*15	I021/160	지상벡터	4
27	I021/070	Mode 3/A	2	I021/165	방향전환률	2
28	I021/131	신호세기	1	I021/077	리포트전송시간	3
FX	-		-	-		-

29			I021/170	타깃 ID	6
30			I021/020	방사체 범주	1
31			I021/220	기상정보	1+
32			I021/146	선택고도	2
33			I021/148	최종상태선택 고도	2
34			I021/110	궤도계획	1+
35			I021/016	서비스 관리	1
FX			-		-
36			I021/008	항공기운영상 태	1
37			I021/271	지상운영능력	1+
38			I021/132	메시지 세기	1
39			I021/250	Mode S MB 데이터	1+N *8
40			I021/260	ACAS 동작지시 레포트	7
41			I021/400	수신기 ID	1
42			I021/295	데이터 수명	1+
FX			-		-

소, I021/210 링크 기술 총 5개의 필수 데이터 항목이 있으며 필드상세는 2 octets이 필요하다. 반면 v2.1의 경우 I021/010 데이터 소스 ID, I021/040 타깃 리포트 기술자, I021/080 타깃 주소, I021/090 품질 지표 총 4개의 필수 데이터 항목이 있으며 필드상세는 3 octets이 필요하다. v0.26의 I021/210 링크 기술은 ADS-B 타깃 리포트 전송을 위해 사용되는 링크(1090ES, UAT, VDL Mode 4 등)가 무엇인지를 알려주는 데이터 항목으로 필수 데이터 항목으로 지정되어 있으나 v2.1에서는 I021/210 MOPS (minimum operational performance standards) 버전의 세부 데이터에 링크 기술이 포함되어 있으며 필수 데이터 항목으로 지정되어 있지 않다. 이처럼 v0.26과 v2.1의 데이터 포맷은 매우 다른 구조를 가지고 있기 때문에 이러한 구조를 정의하는 필드상세 또한 다른 값을 가지게 된다. 따라서 필드상세는 데이터 항목의 구성에 대한 정보뿐만 아니라 ASTERIX CAT.021 버전을 구별할 수 있는 특징이 된다.

3-2 ASTERIX CAT.021 패턴 생성

3-1에서 언급한 특징인 길이와 필드상세를 기반으로 ASTERIX CAT.021 메시지의 버전을 구별할 수 있는 패턴을 정의하기 위해 데이터를 수집한다. 하루 단위로 반복되는 비행계획에 의해 운행되는 국내선과 국제선의 항공교통상황을 반영하기 위해 데이터를 24시간동안 수집한다. ASTERIX CAT.021 데이터를 수신하면 우선 디코딩을 수행하고 해당 메시지의 버전을 확인하면 길이, 필드상세, 버전 정보를 표 4와 같이 검색 테이블에 함께 저장한다. 그리고 발생 빈도수가 가장 많은 패턴이 최상위에 위치하도록 하여 빠른 검색이 가능하도록 한다.

표 4. ASTERIX CAT.021 데이터 패턴 검색 테이블
Table 4. ASTERIX CAT.021 data pattern search table.

패턴번호	길이	필드상세	버전	빈도수
P0	39	FFA150	v0.26	39%
P1	50	CD9B3319C0	v2.1	36.5%
P2	48	C51B331BC0	v2.1	9%
P3	35	FF8150	v0.26	3.5%
P4	59	CD9B3B69D8	v2.1	2.5%
P5	33	FF8110	v0.26	2%
P6	44	FFA1D120	v0.26	1%
P7	35	F9A150	v0.26	1%

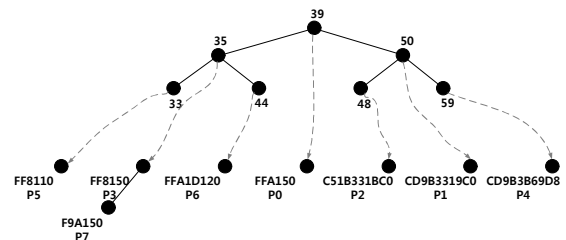


그림 8. ASTERIX CAT.021 버전 결정 트리
Fig. 8. ASTERIX CAT.021 version decision tree.

3-3 우선순위 기반의 버전 결정 트리 생성과 검색 알고리즘

본 논문에서는 Yazdani의 이진 프리픽스 검색 트리를 ASTERIX CAT.021 메시지 포맷의 특성에 맞게 수정한 결정 트리를 사용한다. 그림 8은 표 4의 데이터 패턴 검색 테이블을 통해 생성한 ASTERIX CAT.021 버전 결정 트리의 구조이다. 결정 트리는 길이 검색 트리와 길이 검색 트리 하위의 필드상세 검색 트리로 구성되어 있다. 이것은 계층적 트라이 또는 셋-프루닝 트라이 구조에서의 근원지 IP 트라이와 목적지 IP 트라이와 유사하다. 그러나 본 논문에서 사용하는 결정 트리는 계층적 트라이 또는 셋-프루닝 트라이와는 달리 빈 노드가 존재하지 않으며 프리픽스 매칭이 아닌 값의 크기 비교를 통해 데이터가 속한 클래스를 찾기 때문에 상위 노드로의 역추적 없이 정확한 패턴을 찾아낼 수 있다.

표 4의 패턴들 중 길이 값의 빈도수가 39%로 가장 많은 39가 결정 트리의 루트 노드가 되며, 나머지 길이 값들은 길이 39의 하위 자식 노드 후보가 된다. 그리고 결정 트리의 균형을 맞추기 위하여 후보들의 평균으로 후보들을 양분한다. 최상위 루트인 39를 제외한 나머지의 평균은 44.8이므로 좌측 자식 트리에는 33, 35, 44가 포함되며 우측 자식 트리에는 48, 50, 59가 포함된다. 양분된 후보들에 대하여 발생 빈도수가 높은 길이 값을 루트로 정하고 나머지를 평균으로 양분하는 과정을 반복하면 그림 8의 상위 트리인 길이 검색 트리를 완성할 수 있다. 길이 검색 트리가 완성되면 각 노드의 하위 트리인 필드상세 검색 트리를 생성한다. 길이 35를 가진 노드는 FF8150, F9A150 두 개


```

BuildVersionDecisionTree(Patterns)
1  if Patterns is empty then return endif
2  Set root = the maximum frequent value in the Patterns
3  if the datatype of root is length then
4    Let fspecs contains all fspecs of root
5    root.fspecTree = BuildVersionDecisionTree(fspecs)
6  endif
7  Set root.m = the average value of Patterns except root
8  while Patterns is not empty do
9    A = Patterns.pop()
10   if A ≤ root.m then
11     leftPatterns.add(A)
12   else
13     rightPatterns.add(A)
14   endif
15 endwhile
16 root.leftChildTree =
17   BuildVersionDecisionTree(leftPatterns)
18 root.rightChildTree =
19   BuildVersionDecisionTree(rightPatterns)
20 end BuildVersionDecisionTree
    
```

그림 9. 우선순위 기반의 버전 결정 트리 생성 프로시저의 의사코드

Fig. 9. Pseudo code for BuildVersionDecisionTree procedure based on priorities.

의 필드상세 값을 가지고 있으며, 길이 검색 트리 생성과 마찬가지로 발생 빈도수가 가장 많은 FF8150이 루트가 되며 나머지 값인 F9A150은 FF8150의 자식 노드가 된다. 35를 제외한 나머지 길이 검색 트리의 노드들은 하나의 필드상세 값을 가지므로 루트 노드 하나만을 가진 필드상세 검색 트리를 가지게 된다.

그림 9는 우선순위 기반의 버전 결정 트리를 생성하는 과정을 나타내는 BuildVersionDecisionTree 프로시저의 의사코드이다. 먼저 입력되는 패턴들이 존재하지 않으면 프로시저를 종료하며, 패턴들이 존재하면 패턴들 중에 가장 빈도수가 많은 길이 값 또는 필드상세 값을 root로 선정한다. 만약 root값의 형태가 길이이면 해당 길이 값을 가지는 필드상세 값들을 fspecs에 추가하고 5번의 과정을 통해 BuildVersionDecisionTree 프로시저에 입력하여 하위 필드상세 검색 트리 root.fspecTree를 생성한다. root.fspecTree를 완성하면 7번을 통해 얻은 평균값 root.m을 이용하여 루트 값을 제외한 나머지 패턴들을 8~15번 과정을 통해 leftPatterns와 rightPatterns로 양분한다. 10번의 조건과 같이 평균값보다 작거나 같으면 11번을 통해 패턴을 leftPatterns에 추가하며, 그렇지 않으면 13번을 통해 패턴을 rightPatterns에 추가한다. 그리고 leftPatterns와 rightPatterns를 16, 17번을 통해 각각 프로시저에 입력하여 좌측 자식 트리 root.leftChildTree와 우측 자식 트리 root.rightChildTree를 완성한다.

그림 10은 수신 ASTERIX CAT.021 데이터를 data라고 할 때, data의 버전을 찾아내는 SearchPattern 프로시저의 의사코드

```

SearchPattern(tree, data)
1  if tree = NIL then return -1endif
2
3  // check length
4  if tree is lengthTree then
5    if tree.root is equal to data.length then
6      return SearchPattern(tree.root.fspecTree,data)
7    endif
8  if data.length ≤ tree.root.m then
9    return SearchPattern(tree.root.leftChildTree,data)
10 else then
11   return SearchPattern(tree.root.rightChildTree,data)
12 endif
13
14 // check FSPEC
15 if tree is fspecTree then
16   if tree.root is equal to data.fspec then
17     return tree.root.version
18   endif
19   if data.fspec ≤ tree.root.m then
20     return SearchPattern(tree.root.leftChildTree,data)
21   else then
22     return SearchPattern(tree.root.rightChildTree,data)
23   endif
24 endif
25
26 end SearchPattern
    
```

그림 10. 패턴 검색 프로시저의 의사코드

Fig. 10. Pseudo code for SearchPattern procedure.

드이다. 만약 1번과 같이 더 이상 검색할 수 있는 자식 트리가 존재하지 않으면 -1을 반환함으로써 data의 버전을 찾을 수 없음을 알려준다. 2~11번 과정은 상위 트리인 길이 검색 트리에서의 검색 과정을 나타낸다. 만약 3번 조건과 같이 tree의 root 값이 data의 length와 일치하면 4번을 통해 해당 root의 하위 트리인 fspecTree를 추가 검색한다. 그러나 일치하지 않는 경우에는

6번을 통해 data의 length 값을 tree의 root에 저장되어 있는 평균값 m과 비교하여 m보다 작거나 같으면 7번을 통해 좌측 자식 트리 leftChildTree를, m보다 크면 9번을 통해 우측 자식 트리 rightChildTree를 추가 검색하여 길이 값이 일치하는 노드를 찾아낸다. 12~21번 과정은 하위 트리인 필드상세 검색 트리에서의 검색 과정을 나타낸다. 만약 13번 조건과 같이 tree의 root 값이 data의 fspec과 비교하여 일치하면 최종적으로 14번을 통해 data의 버전 값을 반환한다. 13번 조건을 만족하지 않는 경우 16번 조건을 통해 root에 저장된 평균값 m과 data의 fspec을 비교하여 m보다 작거나 같은 경우 17번을 통해 좌측 자식 트리 leftChildTree를, 큰 경우 19번을 통해 우측 자식 트리 rightChildTree를 추가 검색하여 필드상세 값이 일치하는 노드

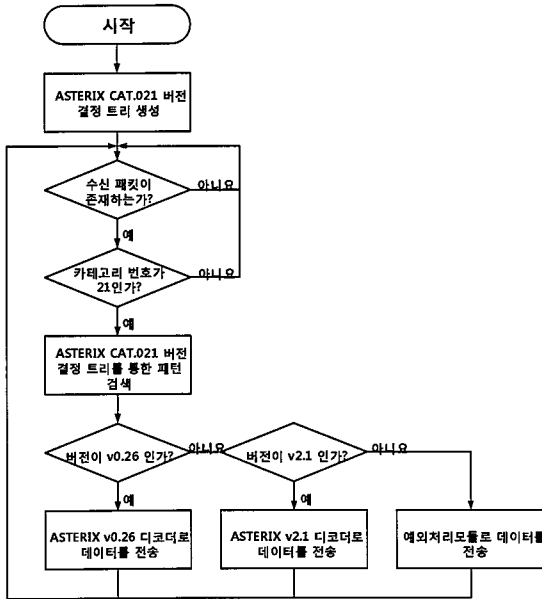


그림 11. 다중 버전 ADS-B 타깃 리포트 메시지 처리 흐름도
 Fig. 11. A flowchart representing a process for multi version ADS-B target report message.

를 찾아낸다.

그림 11은 그림 9에서 생성한 버전 결정 트리와 그림 10의 패턴 검색 프로시저를 이용하여 ADS-B 타깃 리포트 메시지를 처리하는 과정을 그림으로 나타낸 것이다. ASTERIX CAT.021 버전 결정 트리가 준비되면 네트워크로부터 패킷을 수신한다. 수신한 데이터의 카테고리 번호가 21이면 ASTERIX CAT.021 버전 결정 트리를 이용하여 패턴을 검색하여 해당 데이터의 ASTERIX CAT.021 버전을 찾아낸다. 이후 찾아낸 버전에 맞는 디코더로 데이터를 전송함으로써 ADS-B 타깃 리포트 메시지를 시스템이 활용할 수 있도록 한다.

IV. 성능 분석

4-1 실험 환경

그림 12는 구현한 다중 버전 ADS-B 타깃 리포트 처리 모듈의 성능을 평가하기 위한 시뮬레이션 환경 구성도이다. 서버 A와 B는 ADS-B 지상국을 모의한 시스템으로 항공기로부터 수신한 ADS-B 메시지를 이용하여 ASTERIX CAT.021을 생성하고 이를 항공관제자동화시스템으로 전송한다. 이 때 서버 A는 ASTERIX CAT.021 v0.23을 사용하고 서버 B는 ASTERIX CAT.021 v2.1을 사용한다. 그리고 서버 C는 항공관제자동화시스템 내부의 ADS-B 모니터링 시스템을 모의한 것이다. 서버 C는 서버 A와 B로부터 ASTERIX CAT.021을 수신하여 메시지의 무결성 확인 및 분석을 수행하며 논문에서 구현한 다중 버전

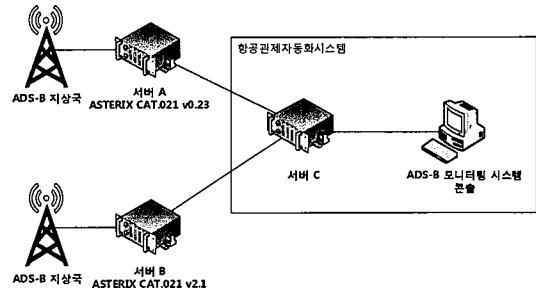


그림 12. 시뮬레이션 환경 구성도
 Fig. 12. Diagram of simulation environment.

ADS-B 타깃 리포트 메시지 처리 모듈이 설치되어 있다. 서버 C에서 처리한 ASTERIX CAT.021 타깃 리포트 메시지 정보는 ADS-B 모니터링 시스템 콘솔의 화면을 통하여 조회할 수 있으며 이로 인한 성능 지연은 없다고 가정한다. 입력데이터는 실제 김포공항에서 운용 중인 ASTERIX CAT.021 메시지를 저장한 로그파일을 사용하였다. 일단위로 반복되는 항공상황을 반영하기 위해 24시간 주기(하루 약 10만개의 메시지)로 실험을 10회 반복 수행하였으며, 처리지연시간, CPU 사용률, 메모리 사용률의 평균값을 산출하였다. 그리고 항공교통량 증가를 모의하기 위해 평시 교통 상황(약 50대)을 반영하는 ASTERIX CAT.021 메시지를 원하는 배수만큼 복사하고, 식별자인 타깃 주소와 위치를 나타내는 위경도 값을 교체하여 사용하였다.

4-2 실험 결과

구현 모듈의 평가를 위해 구현 모듈과 다른 방식으로 ADS-B 버전을 판단하는 순차 모듈을 비교군으로 포함하였다. 순차 모듈은 ASTERIX CAT.021 메시지를 수신하면 우선 v0.26 전용 디코더로 메시지 파싱을 수행한다. 만약 에러가 발생하지 않으면 해당 메시지는 v0.26과 호환되는 메시지로 판단한다. 그러나 파싱 수행 중에 값에 에러가 존재하거나 길이 필드에 정의된 길이와 일치하지 않는 경우 v2.1 전용 디코더로 메시지 파싱을 추가 수행하고 에러가 존재하지 않으면 최종적으로 v2.1과 호환되는 메시지로 판단하는 방식을 사용한다.

그림 13은 ASTERIX CAT.021 메시지의 버전을 판단한 후 디코딩 정보를 외부로 출력하는 데에 걸리는 평균지연시간을 비교한 것이다. 시뮬레이션 데이터의 양을 항공기 대수 50대에서 450대까지 증가시키면서 측정된 결과, 구현 모듈은 지연시간을 순차 모듈 대비 19~30 us 단축하였다. 순차 모듈의 경우 ASTERIX CAT.021 메시지가 v0.26과 호환되지 않는 경우 v2.1과의 호환여부를 다시 확인해야 하므로 이에 따른 오버헤드가 발생한다. 반면 구현 모듈은 미리 정의된 결정 트리에 의해 정확하게 버전을 분류하기 때문에 버전 판단 실패로 인한 오버헤드 없이 신속하게 ADS-B 정보를 얻어낼 수 있다. 항공기 대수의 증가에 따라 처리지연시간이 감소하는 이유는 반복 작업이 늘어날수록 캐시 히트가 많아져 수행 속도가 향상되는 CPU의 특

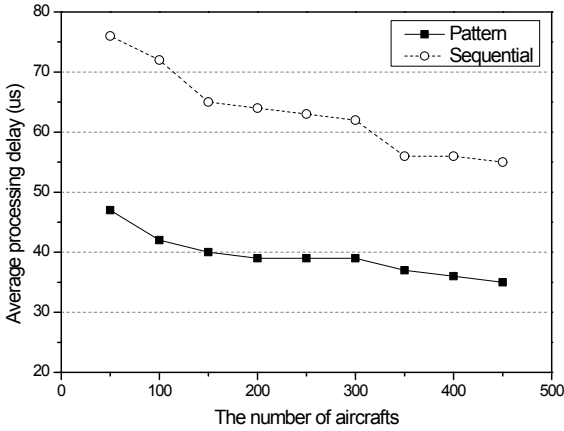


그림 13. 평균 처리지연시간 비교
Fig. 13. The comparison with average processing delay.

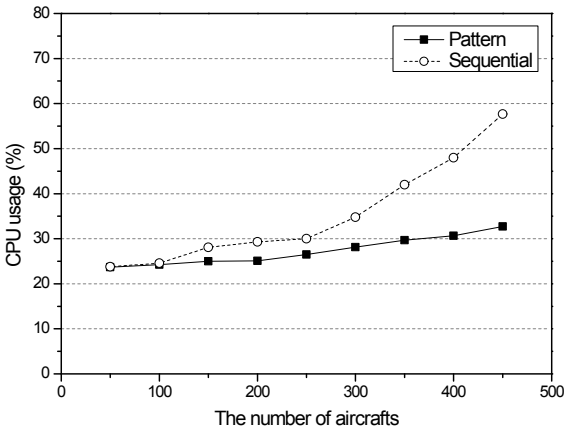


그림 14. 평균 CPU 사용률 비교
Fig. 14. The comparison with average CPU usage.

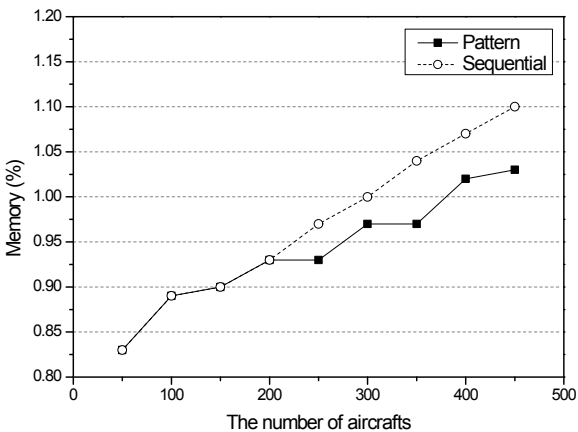


그림 15. 평균 메모리 사용률 비교
Fig. 15. The comparison with average memory usage.

성 때문이다.

그림 14는 각 모듈의 평균 CPU 사용률을 비교한 것이다.

50~100대 정도의 낮은 수준의 오버헤드에서는 두 모듈의 CPU 사용률에 차이가 없으나, 100대 이상 항공기 대수가 많아짐에 구현 모듈은 순차 모듈에 비해 약 3~25% 낮은 CPU 사용률을 보였다. 이는 데이터양이 증가할수록 버전을 분류하기 위해 결정트리를 검색하는데 드는 작업량보다 버전 분류 실패로 인해 다른 버전을 재검색하는데 드는 작업량이 더 많아지기 때문이다.

그림 15는 각 모듈의 평균 메모리 사용률을 비교한 것이다. 항공기 50~200대 사이의 낮은 수준의 오버헤드에서는 평균 메모리 사용률이 거의 동일하였으나, 그 이상 증가함에 따라 구현 모듈은 순차 모듈 대비 0.04~0.07% 낮은 메모리 사용률을 보이는 것을 확인하였다. 이는 순차 모듈이 하나의 ASTERIX CAT.021 메시지에 대하여 두 개의 디코더를 동시에 사용하기 때문에 하나의 디코더를 사용하는 구현 모듈에 비해 더 많은 메모리 공간을 사용하기 때문이다.

V. 결론

ADS-B는 기존 레이더 시스템과 달리 위성항법시스템을 사용하므로 성능 및 지형 한계를 극복할 수 있어 증가하는 항공교통량을 효과적으로 처리할 수 있는 차세대 항행 시스템의 주요 기술 중 하나이다. ADS-B 지상국 및 항공관제자동화시스템에서 ADS-B 정보를 공유하기 위해 표준으로 사용하는 ASTERIX CAT.021 메시지는 다수의 버전을 가지고 있기 때문에 메시지 교환을 위해서는 변환 또는 필터링을 수행하는 중계 시스템이 필요하다. 그러나 이러한 중계 시스템의 추가로 인해 설치비용이 증가하고 전송 지연 및 처리 지연이 발생한다.

본 논문에서 설계한 다중 버전 처리 ADS-B 타깃 리포트 메시지 모듈은 중계 시스템 없이 ASTERIX CAT.021 버전을 구분할 수 있어 지연 시간을 효과적으로 줄일 수 있다. 수신 ASTERIX 데이터로 생성한 패턴 정보를 이용하며 패턴 검색 시간을 최소화하기 위해 프리픽스 이진 트리를 수정한 우선순위 기반의 결정 트리를 사용한다.

구현 모듈의 성능을 평가하기 위해 ADS-B 타깃 리포트 메시지를 공유하는 지상국 및 항공관제자동화시스템의 구성을 모의한 시뮬레이션 환경을 구성하고 구현 모듈과 순차 모듈의 성능을 비교분석하였다. 그 결과 구현 모듈은 순차 모듈보다 평균 19~30 us 짧은 처리지연시간을 가졌으며, CPU와 메모리 사용률은 각각 3~25%, 0.04~0.07% 낮은 사용률을 보였다. 따라서 구현 모듈은 ADS-B 지상국 및 ADS-B 타깃 리포트 메시지를 활용하는 항공교통관제자동화시스템의 성능 향상에 크게 기여할 수 있다.

참고 문헌

[1] Aviation market trend and analysis, Ministry of Land,

- Infrastructure and Transport, Korea, Vol. 33, Mar, 2015.
- [2] Ministry of Land, Infrastructure and Transport, "Press : Breaking highest air traffic in 2014," Available: http://acc.molit.go.kr/USR/N0201/m_16156/dtl.jsp?lcmspa ge=1&id=95075153
- [3] H. S. Jung, D. M. Kim, and C. H. Yeom, "Analysis of CNS/ATM technology trend," *Airspace Industrial Technology*, Vol. 8, No. 2, pp. 113-123, Dec. 2010.
- [4] CNS today, Ministry of Land, Infrastructure and Transport, Korea, Vol. 2, Oct, 2014.
- [5] M. Strohmeier, M. Schäfer, V. Lenders, and I. Martinovic, "Realities and challenges of nextgen air traffic management: the case of ADS-B," *IEEE Communications Magazine*, Vol. 52, No. 5, pp. 111-118, May. 2014.
- [6] D. McCallie, J. Butts, and R. Mills, "Security analysis of the ADS-B implementation in the next generation air transportation system," *International Journal of Critical Infrastructure Protection*, Vol. 4, Issue 2, pp. 78-87, Aug. 2011.
- [7] C. S. Yoo, B. S. Song, A. Cho, K. J. Sung, and S. O. Koo, "Implementation and operational test of ADS-B system in Goheung aeronautical center," *Airspace Technology*, Vol. 13, No. 1, pp. 1-9, Jul. 2014.
- [8] Eurocontrol Specification for Surveillance Data Exchange ASTERIX Part 12 Category 21 ADS-B Taarget Reports, European Organization for The Safety of Air Navigation, EUROCONTROL-SPEC-0149012, Edition 2.3, Jan. 2015.
- [9] Eurocontrol Standard Document for Surveillance Data Exchange Part 9 : Category SDPS Track Messages, European Organization for The Safety of Air Navigation, SUR.ET1.ST05.2000-STD-09-01, Edition 1.17, Dec. 2014.
- [10] Guidance Material on generation, processing and sharing of ASTERIX Category 21 ADS-B Messages provided in Appendix H to the Report on agenda item 3.4, APANPIRG/23, Bangkok:Thailand, Sep. 2012.
- [11] M. Dixit, B.V. Barbadekar, and A.B. Barbadekar, "Packet classification algorithms," in *Industrial Electronics on IEEE International Symposium*, Seoul: Korea, pp. 1407-1412, Jul. 2009.
- [12] P. Gupta, and N. McKeown, "Algorithms for packet classification," *IEEE Network*, Vol. 15, No. 2, pp. 24-32, Mar/Apr. 2001.
- [13] S. Lee and H. Lim, "A Hierarchical Packet Classification Algorithm Using Set-Pruning Binary Search Tree," *Journal of The Korean Institute of Information Scientists and Engineers : Information and Communication*, Vol. 35, No. 6, pp. 482-496, Dec. 2008.
- [14] N. Yazdani and P. S. Min, "Prefix trees: new efficient data structures for matching strings of different lengths," *Database Engineering and Applications on International Symposium*, Grenoble:France, pp. 76-85, Jul. 2001.
- [15] 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.
- [16] AsterixInspector [Internet]. Available: <http://asterix.sourceforge.net/>
- [17] vog/asterix [Internet]. Available : <https://github.com/vog/asterix>
- [18] CroatiaControlLtd/asterix [Internet]. Available : <https://github.com/CroatiaControlLtd/asterix>
- [19] Eurocontrol Standard Document for Surveillance Data Exchange Part 12 : Category 021 ADS-B Messages, European Organization for The Safety of Air Navigation, SUR.ET1.ST05.2000-STD-12-01, Edition 0.26, Jun. 2005.
- [20] Eurocontrol Standard Document for Surveillance Data Exchange Part 12 : Category 021 ADS-B Reports, European Organization for The Safety of Air Navigation, SUR.ET1.ST05.2000-STD-12-01, Edition 2.1, May. 2011.



김강희 (Kanghee Kim)

2011년 2월 : 인하대학교 전자공학과 (공학사)

2013년 2월 : 인하대학교 전자공학과 (공학석사)

2013년 3월~현재 : 인하대학교 전자공학과 박사과정

※관심분야 : 항공교통관제 시스템, 차세대항행시스템, 컴퓨터 네트워크



장 은 미 (Eunmee Jang)

2002년 2월 : 인하대학교 전자전기컴퓨터공학부 컴퓨터공학전공 (공학사)
2005년 2월 : 인하대학교 정보·컴퓨터 교육학과 (교육학석사)
2010년 9월 ~ 현재 : 인하대학교 전자공학과 박사과정
※ 관심분야 : 컴퓨터 네트워크, 컴퓨터 아키텍처, ADS-B



송 인 성 (Inseong Song)

2009년 2월 : 인하대학교 전자공학과 (공학사)
2011년 2월 : 인하대학교 전자공학과 (공학석사)
2011년 3월 ~ 현재 : 인하대학교 대학원 전자공학과 박사과정
※ 관심분야 : 병렬 및 분산 처리 시스템, 컴퓨터 아키텍처, ADS-B



조 태 환(Taehwan Cho)

2001년 2월 : 인하대학교 항공우주공학과 (공학사)
2014년 2월 : 인하대학교 전자공학과 (공학박사)
2014년 12월 ~ 현재 : 공군사관학교 전자공학과 조교수
※ 관심분야 : 항공전자 시스템, 항공통신



최 상 방 (Sangbang Choi)

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