Research Paper
Journal of Aerospace System Engineering
Vol.16, No.1, pp.1-11 (2022)
EISSN 2508-7150
http://dx.doi.org/10.20910/JASE.2022.16.1.1

AFDX Tap과 AFDX 프로토콜 분석기를 이용한 AFDX 네트워크 인증 기술

박부식[1,†], 손명환[1], 이정도[1], 윤종호[2]

[1]한국전자기술연구원 지능정보연구본부 모빌리티플랫폼연구센터
[2]한국항공대학교 항공전자정보공학과

Evaluation of AFDX Certification Support System by both AFDX Tap and AFDX Analyzer

Pusik Park[1,†], Myeonghwan Son[1], Jeongdo Lee[1] and Jongho Yoon[2]

[1]Korea Electronics Technology Institute
[2]Korea Aerospace University

Abstract

Avionics Full-DupleX Ethernet (AFDX) is a next-generation avionics network interface technology that is widely applied in the latest aircraft to replace ARINC429 and MIL-STD-1553B. However, the criteria for authenticating an avionics network consisting of AFDX are very scarce. Using AFDX Protocol Analyzer developed by the Korea Electronics Technology Research Institute and AFDX Tap developed by the Korea Aerospace University, we proposed a technology of certification practicality that can verify the normal functioning of avionics equipment with AFDX network interface. Our proposed technology provided the ability to collect precision packets, to verify AFDX specification compliance, and perform automatic tests to reduce the time and cost of authentication of AFDX avionics devices.

초 록

항공용 이더넷 기술 AFDX (Avionics Full-DupleX Ethernet)는 ARINC429와 MIL-STD-1553B를 대체할 최신 항공기들에 널리 적용되고 있는 차세대 항공전자 네트워크 인터페이스 기술이다. 하지만 AFDX로 구성된 항전 네트워크를 인증하기 위한 솔루션은 매우 부족한 상황이다. 본 논문은 한국전자기술연구원에서 개발한 AFDX 프로토콜 분석기와 한국항공대학교에서 개발한 AFDX Tap을 이용해 AFDX 네트워크 인터페이스를 탑재한 항전 장비가 정상적으로 기능하는지 검증할 수 있는 인증 실용화 기술을 소개한다. AFDX 항전 장비의 기능을 인증할 수 있도록 정밀 패킷 수집, AFDX 규격 적합성 검증, 자동 테스트 수행 기능 등을 제공하여 AFDX 항전장치의 인증 시간과 비용을 단축시킬 수 있다.

Key Words: Avionics (항공전자), Ethernet, AFDX, ARINC664 (항공용 이더넷), Certification(인증), Protocol analyzer (프로토콜 분석기), DO-178, DO-254 (인증 프로세스)

1. 서 론

국내 항공산업의 지속적인 발전과 국산 항공기의 해외 수출시장 확대를 위해 군용항공기 감항인증관련법, 시행령 및 시행규칙, 업무규정 제정, 표준감항 인증 기준 고시, 감항인증 교육 등 본격적인 감항인증 업무가 추진 중이나 체계장비 위주로 제도 및 절차가 구축되어 있다. 또한 국내 대기업을 포함한 중소기업체에서 헬기, 고정익 항공기에 대한 부품, 구성품, 훈련체계가 개발 진행 중이나 이에 대한 인증제도 및 절차 부재로 우수한 국내 부품, 구성품 및 훈련체계의 사용이 제한되고 해외구매에 의존함으로써 국가 신경제 성장 동력으로서 항공 산업이 제기능을 발휘하지 못하고 있는 실정이다 [1].

그럼에도 불구하고, 국내 항공 산업은 KT-1 기본훈련기, T-50 고등 훈련기, FA-50 경공격기, KUH (Korean Utility Helicopter) 한국형 기동 헬기 등 여

Received: Dec. 15, 2020 Revised: Dec. 14, 2021 Accepted: Dec. 14, 2021

† Corresponding Author
Tel: +82-31-739-7507, E-mail: pusik.park@keti.re.kr

러 군용 항공기를 개발하여 운영하고 있어 군용 항공기 개발은 어느 정도 수준에 도달해 있다고 할 수 있다. 이에 반해 민수 항공기 분야는 아직 많은 경험이 축적되지 않은 상황이다. 4인승 소형항공기 KC-100은 미국 연방항공청(FAA, Federal Aviation Association)와 한국 국토교통부의 인증과정 (BASA IPA)을 거쳐 개발된 국내 최초의 민항기이며 이를 훈련기 (KT-100)로 개조하여 공군에서 활용되고 있다 [2]. 하지만 수리온의 경우 12년 도입되었지만 지금까지도 FAA 혹은 EASA (European Union Aviation Safety Agency)로 부터 민수용 항공 감항 인증을 획득하지 못해 민수용 해외 수출에 어려움을 겪고 있다. 민간 항공기용 부품의 감항 인증을 받기 위해서는 대상 항공기가 필요하기 때문에 부품 개발사가 감항 인증 경험을 획득하기 어려우며 항공 감항 인증으로 인해 추가되는 개발 비용과 시간이 적지 않기 때문에 국내에서 항공기 인증 경험을 축적하기엔 아직 시간과 비용이 더 필요한 실정이다.

다행히도 최근 수행되고 있는 다양한 항전 프로젝트를 통해 국내 기업들은 꾸준히 항공기 인증 프로세스를 체득하고 있다. 특히, 군용 항공기의 감항 인증 프로세스가 최근 민간 항공기 인증 프로세스를 준용하기 시작하면서 DO-178C와 DO-254 프로세스[3] 등을 항전 장비 개발 프로세스로 정착시키고 있는 중이다. 한국형 전투기 KFX 개발 사업과 LAH (Light Armed Helicopter) 개발 사업 모두 LRU (Line Replaceable Unit)단위부터 항공기 체계까지 모두 민간 항공기 인증 프로세스를 일부 준용하며 진행 중이다.

본 논문에서는 항공기의 여러 요소 기술 중 항전 장비 간의 데이터 교환을 위해 사용되는 항공용 이더넷 AFDX™ 기술을 탑재한 LRU를 효과적으로 인증 받을 수 있도록 하는 실용화 기술을 제안하겠다.

2. AFDX 개요와 AFDX 인증 고려사항

90년대 후반 이더넷과 TCP/IP 기반의 인터넷 관련 제품이 데이터 통신기술을 주도함에 따라 저렴한 가격과 고속 전송의 특성을 항공기에 적용하기 위한 차세대 ADN (Aircraft Data Network)의 개발이 시작되어 그 결과 Airbus에 의해 AFDX 기술이 A380의 주로 제어통신망에 사용되었다. AFDX는 아래 그림과 같이 AFDX 스위치에 여러 엔드 시스템이 연결되는 스타 구조의 네트워크 기술이다.

A380 기종에 성공적으로 적용된 AFDX 기술은 ARINC 664 파트 7으로 표준화되고 Airbus와 Boeing의 최신 항공기와 더불어 AugustaWestland의 회전익 항공기 등 다양한 최신 항공기에 폭넓게 적용되고 있다. 국내의 경우에도 LAH에 시범적으로 AFDX 스위치와 소수의 LRU를 AFDX로 연결해 데이터를 교환하고 있다. Fig. 2의 목록은 AFDX 항전 네트워크가 적용된 항공기들이다 [4].

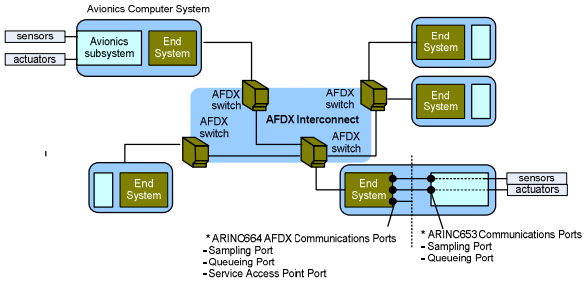

Fig. 1 General AFDX Network Architecture.

Fig. 2 AFDX-installed Aircrafts.

Fig. 3은 A380 기내의 항전장비 간 AFDX 연결 구조이다. 18개의 이중화 된 AFDX 스위치 장비와 80개 이상의 AFDX 엔드 시스템으로 구성된다. 모두 이중화를 지원하고 각 스위치에는 20여개의 포트를 지원하고 있다 [5].

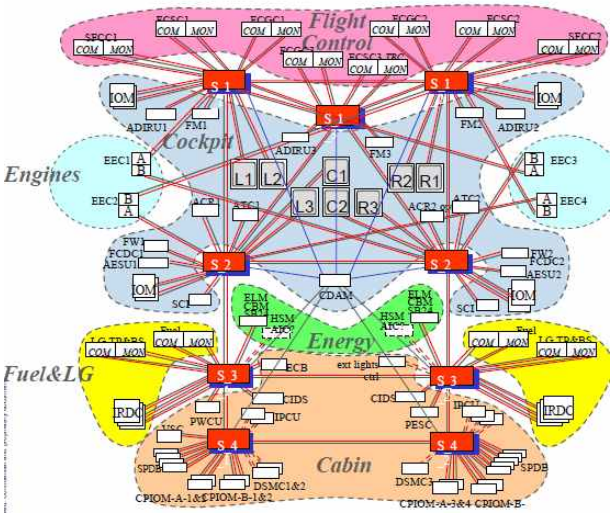

Fig 3 A380 AFDX Network Architecture.

기존의 항공기 제어 네트워크 기술로는 ARINC 429와 ARINC 629가 있으며 군용기의 경우 MIL-STD-1553B 등의 저속의 기술이 사용되었다. 그러나 최근 복잡해지는 항전 시스템들의 데이터 처리량을 수용하기에는 기존 기술의 대역폭은 최대 2Mb/s 수준으로 매우 낮으며 AFDX 기술이 최근 신형 항공기에 장착되는 이유는 100Mb/s의 대역폭과 향후 1Gb/s로의 확장 가능성이다.

AFDX의 대역폭이 넓다 보니 여러 개의 응용 시스템이 넓은 대역폭을 공유할 수 있게 되었다. AFDX는 효율적인 대역폭 활용을 위해 가상 링크 VL (Virtual Link) 개념을 도입하였고 여러 개의 케이블로 통신하던 것을 하나의 케이블로 통신 가능하게 하여 케이블의 수를 줄임으로써 항공기를 경량화 하였다.

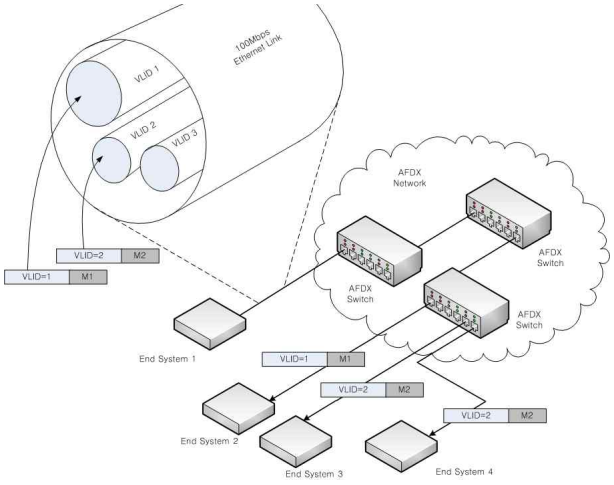

Fig. 4 AFDX Virtual Link.

하지만, 많은 대역폭이 무조건 장점만 있는 것은 아니다. 가상의 링크로 데이터를 통신하는 방법은 개발 시 물리적으로 신호를 측정하거나 분석하는 것을 불가능하게 한다. 여러 개의 LRU가 하나의 케이블을 공유해 각자의 가상 링크 기반으로 통신하기 때문에 LRU의 통신 기능을 검증하기 위해서는 이 가상 링크를 분석할 수 있는 별도의 분석 솔루션이 필요하다.

또한, 항공기 안전 확보와 인증 통과를 위해 AFDX 규격에서 요구하는 다양한 성능 지표가 있다. BAG (Bandwidth Allocation Gap)은 각 가상 링크를 통해 데이터를 전송할 수 있는 데이터 양과 그 사이 간격에 관한 제한 사항인데 이 BAG 규정을 지키지 않을 경우 LRU에서 전송한 이더넷 프레임이 AFDX 스위치에서 버려질 수 있다. 그리고 AFDX 규격에서는 각 LRU는 최대로 허용되는 지터 값을 정하고 있으며 이 지터 값을 초과하여 송신할 경우에는 정상적인 송신을 보장할 수 없게 된다.

저속의 네트워크 기술을 사용하던 예전에는 데이터 양이나 종류가 많지도 다양하지도 않았기 때문에 엔지니어가 개발을 진행하거나 기능을 검증하기가 수월했다. 하지만, 최신 항공기의 데이터 교환 속도와 양이 매우 증가함으로 인해서 과거의 방식으로 데이터와 고장을 분석하기는 역부족인 상태다. 또한 개발 과정에서 수시로 변경되는 ICD (Interface Control Document) 규격과 네트워크 설정 정보 등을 사용자가 실수하지 않도록 자동화하는 기술을 제공하지 않는다면 인증을 통과하는데 많은 시간과 비용이 소요될 것이다.

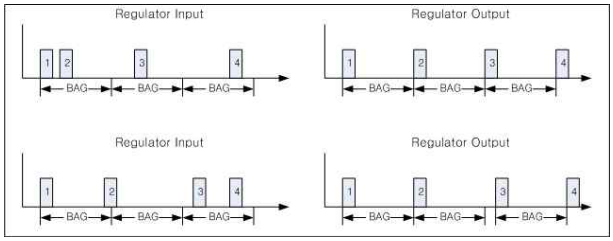

Fig. 5 AFDX Bandwidth Allocation Gap.

3. AFDX 정밀 패킷 수집 기술

AFDX Tap은 100Mbps급 이상의 이더넷 시스템 간에 연결된 전송선로 사이에 개입하여 전-이중 통신과정에서 전송되는 모든 양방향 AFDX 프레임을 추출하여 AFDX 프로토콜 분석기에 전달하는 AFDX 프레임 전용 트래픽 추출 기능을 수행한다 [6]. 또한 액티브 방식의 수집 시스템으로서 신호 감쇄 및 중계 트래픽의 손실없이 두 단말 장치 간 전-이중 이더넷 트래픽을 추출하면서 각 프레임의 송신개시 시점을 100 나노 초 단위로 시간 정보를 해당 프레임에 부착하여 전달할 수 있는 타임스탬핑 기능을 제공한다. 기존의 Network Tap이 수십 마이크로 초 수준의 타임스탬핑을 제공하므로 AFDX Tap은 기존 솔루션 대비 수십 배의 패킷 수집 시간 정밀도를 제공할 수 있다.

게다가 기존의 Network Tap은 IEEE 1588v2 [6]를 지원하지 않기 때문에 서로 다른 Network Tap으로부터 수집된 패킷의 시간 정보가 동기화되지 않아 AFDX 프로토콜 분석기에서 정확한 트래픽 분석이 불가능하다. 또한 스위치의 미러링 방식을 통한 패킷 분석은 미러링 포트의 대역폭 한계로 여러 포트를 동시에 분석하는 것이 제한적이다. 그리고 대역폭의 제한으로 패킷 수집 시간의 지연 발생과 패킷 손실로 인해 정밀한 패킷 모니터링이 힘들어 인증을 획득하는데 어려움이 있을 수 있다[6].

Fig. 6은 한국항공대학교에서 개발한 정밀 시간동기 기능을 탑재한 Quad-포트 AFDX Tap 모듈의 모습이다.

Fig. 6 AFDX Tap module.

3.1 AFDX Tap을 이용한 패킷 수집

이중화 된 항전 장비는 AFDX 프레임을 이중화 전송 링크를 통해 송수신한다. 그리고 각 링크는 전-이중 방식으로 송/수신이 독립적으로 작동한다. AFDX Tap은 이중화 된 전-이중 방식의 항공용 이더넷을 분석하기 위해 총 4개의 이더넷 포트를 이용해 패킷을 수집한다.

Fig. 7은 항전 장비와 AFDX Tap의 연결 구성을 보여준다. AFDX Tap은 총 8개의 이더넷 포트를 가지고 있다. 앞서 설명한 4개의 포트는 항전장비와 AFDX 스위치 사이를 탭핑 (Tapping)하는데 사용되고 나머지 4개 포트는 모니터링을 위해 PC에 연결된다. 탭핑을 통해 복사된 패킷은 4개의 이더넷 포트로 PC로 전달된다. PC로 연결되는 4개의 이더넷 포트는 각각 A 네트워크의 업-링크 패킷, 다운-링크 패킷 그리고 B 네트워크의 업-링크 패킷, 다운-링크 패킷을 전달하는 개별 경로로 이용된다.

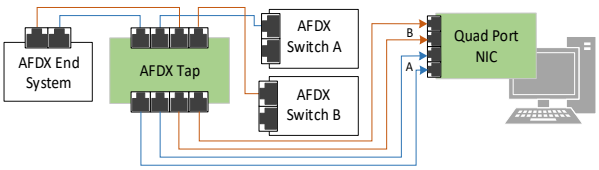

Fig. 7 AFDX Tap configuration.

3.2 분산 AFDX Tap연결을 통한 통합 패킷 수집

항전 장비는 단독으로 동작하는 단순 시스템이 아니다. 여러 항전장비가 AFDX 네트워크로 연결되어 데이터를 교환하며 작동한다. 기존 Network Tap을 이용해 복잡한 항전 시스템을 분석할 경우에는 동기화 이슈가 발생한다.

시간 동기화를 지원하지 않는 다수의 Network Tap에서 PC로 AFDX 프레임을 전달하면 서로 다른 Network Tap으로부터 수집된 패킷의 시간 정보가 동기화되지 않아 수집된 패킷들의 순서를 비교할 수 없다. 기존 시스템의 이런 문제점은 AFDX로 구성된 항전 네트워크의 인증을 획득하는데 어려움을 야기한다. 이런 문제를 해결하기 위해 AFDX Tap은 IEEE 1588v2 시간 동기 기능을 이용해 모든 Tap들의 시각을 동기화하고 동기화된 타임스탬프를 AFDX 프레임에 삽입하여 AFDX 프로토콜 분석기에게 전달한다. 서로 동기화된 시간 정보를 갖는 패킷은 한 곳에 모아 정밀 분석이 가능하다.

Fig. 8은 4개의 항전 장비가 AFDX Tap을 통해 AFDX 스위치와 PC로 연결되는 그림이다. AFDX Tap은 여러 대의 항전 장비가 AFDX 스위치에 연결될 경우 IEEE 1588v2 기능을 통해 모든 항전 장비의 데이터 흐름을 정확하게 확인할 수 있다.

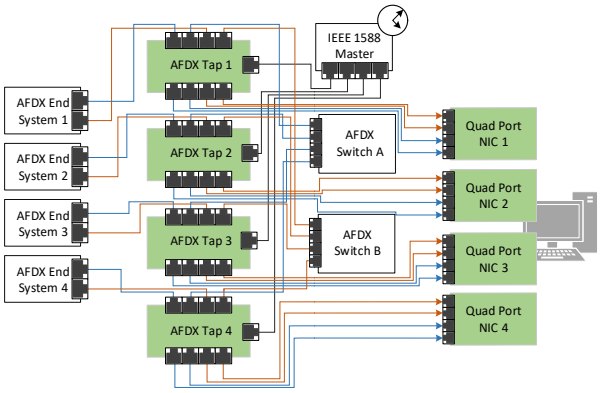

Fig. 8 Multiple AFDX Taps configuration.

4. AFDX 기능 검증

항전 장비를 인증함에 있어 장비의 입출력 인터페이스의 기능 검증은 큰 부분을 차지한다. 그러나 이 검증을 지원하기 위한 전용 소프트웨어 도구는 부재하거나 값비싼 외산 기술에 의존한다. 때문에 AFDX 장비의 개발 초기에는 AFDX 프로토콜을 지원하지 않는 기존의 이더넷 프로토콜 분석 도구를 사용하는 경우가 많다. 하지만 AFDX 기술은 독자적인 주소 체계와 Deterministic QoS (Quality of Service)와 관련하여 망 구성 방법 및 독특한 고유의 송수신 규칙을 가지고 있기 때문에 기존의 분석 도구로 AFDX의 장비를 검증하기에는 한계가 있다. 특히 AFDX의 패킷 쉐이핑 (Shaping), 이중화 관리 (Redundancy Management) 기능의 검증은 정밀한 시간 정보와 다양한 고장 시나

리오를 필요로 하므로 AFDX 장비의 인증을 획득함에 있어서는 AFDX 프로토콜 분석기와 같은 AFDX 전용 프로토콜 분석 도구가 필수적이다.

4.1 AFDX 엔드 시스템의 송신 기능 검증

AFDX 프로토콜 분석기는 AFDX 엔드 시스템의 데이터 송신 기능을 검증할 수 있는 패킷 수집 도구를 제공한다. 패킷 수집 도구는 다수의 포트에서 동시에 패킷을 수집하고 해석하여 검증에 필요한 다양한 정보를 제공한다. 또한 AFDX Tap과 결합하면 하드웨어 타임스탬프를 사용하여 나노-초 단위로 패킷의 송신 타이밍의 정확도를 검증할 수 있다. 패킷 브라우저를 통해 기본적으로 수신된 메시지를 해석하여 원하는 인터페이스에서 원하는 데이터의 송신이 정상적으로 수행되었는지 확인할 수 있으며 이를 통해 AFDX 프로토콜의 가상 링크 별 송신 및 이중화와 관련된 송신 기능을 검증할 수 있다. 또한 추가적으로 다음과 같은 검증을 수행할 수 있다.

첫번째로 AFDX 프로토콜 분석기는 패킷 스트림에 대해 다양한 필터를 적용하여 대역폭, 길이 분포, 전송 간격 등의 통계 수치를 차트를 통해 실시간으로 제공하며 이를 통해 AFDX의 패킷 쉐이핑 기능을 검증할 수 있다. AFDX 엔드 시스템은 VLID (Virtual Link IDentifier) 별로 고정된 BAG (Bandwidth Allocation Gap) 값을 가지고 주기적으로 패킷을 송신한다. 이러한 BAG 관련 기능이 제대로 동작하지 않으면 AFDX 스위치에서 패킷이 폐기되는 경우가 발생한다. 따라서 엔드 시스템 (End-system)의 송신단에서 이 기능이 검증되지 않으면 비정상적인 BAG에 의해 패킷이 폐기된 것인지 AFDX 스위치의 오류에 의해 폐기된 것인지 파악하기가 어렵다.

AFDX 프로토콜 분석기는 필터링을 통해 VLID 별로 패킷을 분류하고 전송 간격을 관측하여 AFDX 스트림의 BAG 정합성을 판단할 수 있다. Fig. 9는 두 VLID 패킷 스트림의 BAG의 분포를 차트로 표시한 예를 보여준다. 분포도의 중심-값 주변의 노이즈와 같이 발생하는 분포 값은 지터 (Jitter)로 인한 전송 간격의 미세한 변화를 의미한다.

두번째로 수신되는 패킷 데이터를 바이트 단위로 분류하여 모니터링 할 수 있다. AFDX 엔드 시스템은 대부분 실시간으로 변하는 고정된 종류의 데이터를 주기적으로 송신한다. 그렇기 때문에 기존의 프로토콜 분석 도구와 같이 패킷 데이터를 순차적으로 나열해 놓는 것 보다는 송신지가 동일한 패킷 내에서 같은 종류의 데이터끼리 분류하여 변화를 관측하는 것이 장비의 기능을 검증하는 데에 더 유리하다.

AFDX 프로토콜 분석기의 실시간 데이터 모니터링 기능을 통해 주기적으로 송신되는 데이터에 대해 Fig. 10과 같이 관심 메시지 종류, 수신 여부 및 지연 시간과 메시지 값의 변동을 동시에 관측하여 특정 데이터 필드에 대한 실시간 검증을 수행할 수 있다. 또한 정해진 규칙에 맞춰 전송되고 있는지를 녹색 아이콘과 적색 아이콘을 이용해 실시간으로 모니터링할 수 있다.

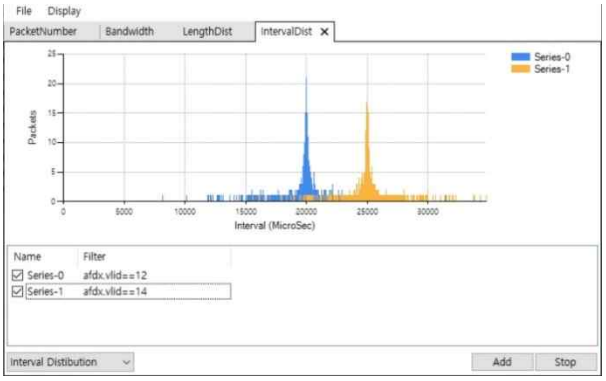

Fig. 9 BAG distribution chart.

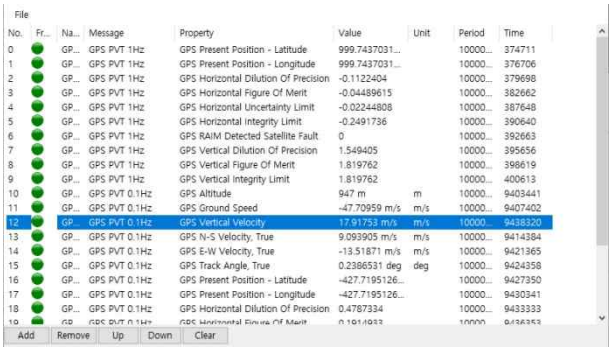

Fig. 10 Real-time monitoring.

마지막으로 AFDX 프로토콜 분석기는 패킷 브라우저 컨트롤과 상호작용하는 항공기의 3D 모델을 제공함으로써 사용자가 네트워크의 상태 정보를 시각적으로 확인할 수 있도록 한다. 이는 다수의 엔드 시스템이 연결된 복잡한 항전 네트워크 환경에서 보이지 않는 패킷 데이터의 이동을 직관적으로 이해할 수 있도록 한다. 이를 통해 사용자는 패킷을 선택하는 것으로 패킷의 송신지와 목적지 및 경로, 그리고 관련 장비의 이름과 위치까지 파악할 수 있고 어떤 엔드 시스템에서 네트워크 결함이 발생했는지 쉽게 파악할 수 있다. Fig. 11은 선택된 패킷이 타겟 항공기의 어떤 부품에서 어떤 경로로 전달되는지를 표시하는 기능을 보여준다.

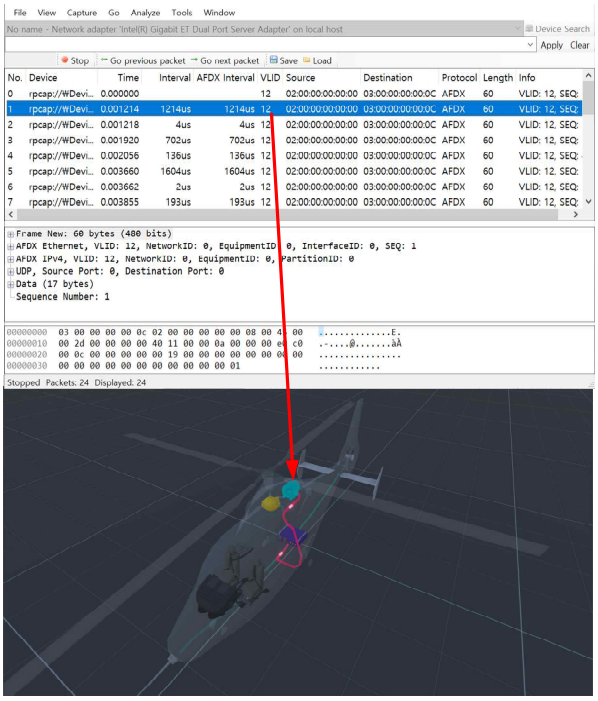

Fig. 11 Packet browser and 3D modeling.

4.2 AFDX 엔드 시스템의 수신 기능 검증

AFDX 프로토콜 분석기는 AFDX 엔드 시스템의 VL 단위의 패킷 수신 기능 및 이중화 복구 기능 검증에 필요한 패킷을 생성하는 도구를 제공한다. 패킷 생성 도구는 AFDX 패킷 및 패킷 스트림을 생성하여 엔드 시스템에게 송신할 수 있으며 사용자는 이를 활용하여 엔드 시스템의 여러 수신 기능들을 검증할 수 있다.

첫번째로 송신하는 메시지 길이, VLID, 포트 번호, 송신 디바이스 등을 설정함으로써 엔드 시스템의 수신단의 패킷 필터링 기능 및 이중화 기능을 검증한다. AFDX는 독자적인 주소 체계와 이중화 기능을 가지고 있어 이를 지원하는 패킷 생성 도구가 필요하다. AFDX 프로토콜 분석기는 Fig. 12과 같은 AFDX 패킷 스트림 생성기를 제공한다.

두번째로 AFDX 프로토콜 분석기는 BAG 와 지터를 적용한 스트림 생성하고 패킷을 삭제하거나 개별적으로 수정하여 패킷 손실, 시퀀스 넘버, BAG와 지터의 오류가 발생하는 등의 시나리오를 구성할 수 있다. 이를 통해 사용자는 다양한 고장 발생 시나리오에 대해 AFDX 엔드 시스템 수신단의 이중화 복구 기능을 검증할 수 있다.

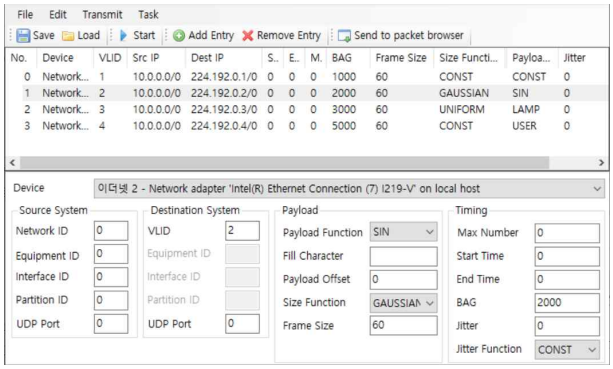

Fig. 12 AFDX packet generation.

4.3 AFDX 최대 지터 (Max Jitter) 검증

AFDX의 Fig. 13은 여러 개의 어플리케이션이 하나의 엔드 시스템에서 동시에 패킷을 송신하고자 할 경우 패킷의 송신 지연에 의한 지터가 발생하는 경우의 예를 보여준다. APP1이 BAG1인 VL1 스트림을 송신하고, APP2가 BAG2인 VL2 스트림을 송신하면 송신 링크에서는 그림과 같은 지터가 발생하게 된다. AFDX 규격에는 발생하는 지터가 최대 지터를 초과하면 안 된다는 규칙이 있고 이 규칙을 준수하지 못하면 해당 AFDX 네트워크는 정상적으로 동작하지 못하므로 AFDX 장비들은 이 지터 제한을 준수해야 한다. 최대 지터는 AFDX 네트워크의 구성에 따라 AFDX 표준에 정의된 식에 의해 계산된다.

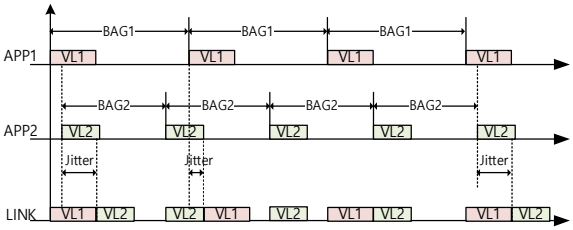

Fig. 13 Example of AFDX Jitter.

AFDX 프로토콜 분석기는 AFDX 네트워크에서 발생하는 지터의 유효성을 검증하기 위한 도구를 제공한다. AFDX 네트워크에서 발생하는 지터의 유효성을 검증하기 위해서는 해당 네트워크의 최대 지터와 BAG에 대한 정보가 필요하며 그 정보는 AFDX 설계 지원 SW를 통해 설정할 수 있다. AFDX 설계 지원 SW에서는 네트워크 구성을 정의하기 위해서 Fig. 14과 같은 모습의 AFDX 네트워크를 그려볼 수 있다.

사용자는 엔드 시스템과 AFDX 스위치를 마우스 드래그-앤-드롭으로 연결하여 AFDX 네트워크를 구성하고 각 엔드 시스템 마다 사용되는 VL을 추가하며 각 VL마다 BAG, 지터, 패킷의 최소 길이 (L_{Min})와 최대 길이(L_{Max})를 지정한다. 구성이 완료된 후에 룰 체크

(Rule Check)를 수행하여 각 경로 별로 최대 지터가 계산된다. AFDX 구성 내용은 XML 파일로 저장거나 불러올 수 있다.

AFDX 프로토콜 분석기는 BAG 측정을 통해 가상 링크 별로 지터를 계산하고 최대 지터를 초과하는지 확인하여 지터의 유효성을 검사한다. 패킷의 지터가 최대 지터를 초과하면 그 패킷의 지터는 무효한 것이며 초과하지 않으면 유효한 것으로 판단한다. 이때 분석기가 BAG의 정확한 옵셋(Offset) 위치를 알지 못하기 때문에 초기 옵셋에서 점차적으로 옵셋을 수정하면서 지터의 유효성을 계산하는 알고리즘이 사용된다.

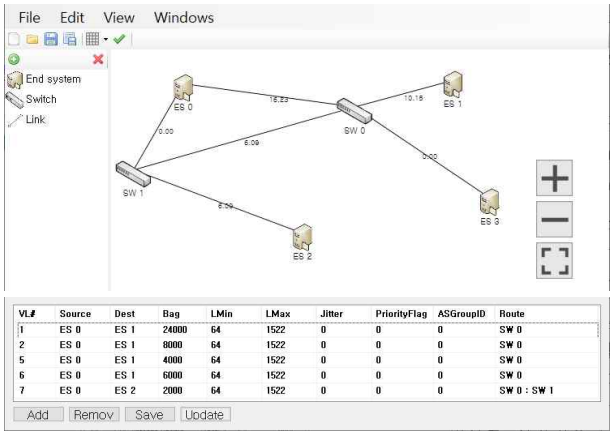

Fig. 14 AFDX Network Configurator.

5. AFDX 인증 지원 기능

5.1 AFDX ICD 편집 기능

AFDX 메시지의 구조는 ICD (Interface Control Document) 또는 ADN (Aircraft Data Network) 메시지로 정의된다. ICD는 그 인터페이스의 입출력에 대한 필수적이고 중요한 정보이며 다수의 네트워크 장비가 연결되는 항전 시스템에서는 ICD의 구조가 복잡하고 양이 방대하여 관리가 까다롭다. 특히 항전 장비를 개발하는 과정에서 빈번하게 발생하는 ICD의 수정은 많은 시간을 소비하여 개발 참여자들 간에 갈등을 유발하기도 한다.

기존의 프로토콜 분석기들은 Fig. 15-(a)와 같이 ICD를 XML, ASN.1, UML, JSON 같은 언어를 통해 추상화하고 이 데이터를 기반으로 분석 코드를 생성하여 ICD를 분석하거나 Fig. 15-(b)와 같이 Lua 같은 스크립트 언어를 사용하여 직접 메시지를 해석하는 추가 기능을 제공한다. 전자와 같은 방식은 추상화 결과를 코드로 변환하여 응용 소프트웨어를 다시 빌드해야 하는 불편함이 있고, 후자의 경우에는 스크립트 언어를 배워야 하는 어려움이 있다. 또한, 두가지 방법 모두 방대한 양의 ICD를 정의하기 위해서는 많은 파일이 필요하여 유지 보수하기가 불편하다.

AFDX 프로토콜 분석기는 Fig. 16과 같이 객체-관계형 모델을 기반으로 한 ICD 디자이너 (Designer) 및 분석기를 제공한다. ICD 디자이너는 ICD 편집 UI를 제공하며 사용자는 이것을 사용하여 기존의 방법보다 직관적이고 편리하게 ICD를 추가/편집/삭제할 수 있다. 정의된 ICD는 데이터베이스에 저장되며 프로토콜 분석기가 메시지를 수신했을 때 메시지 분석기는 해당 메시지의 해석에 필요한 ICD 정의 객체를 검색하여 메시지를 해석하고 그 결과를 GUI를 통해 출력한다.

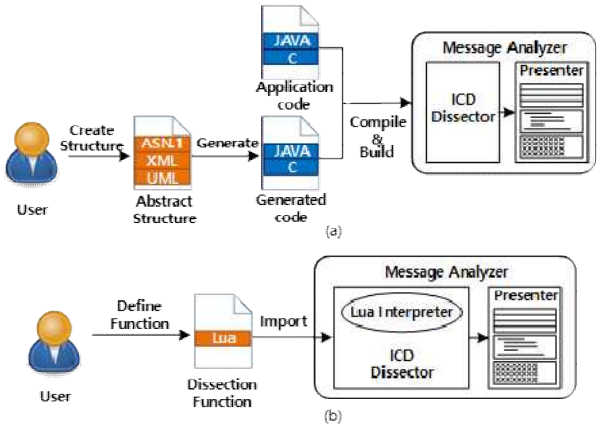

Fig. 15 Legacy ICD message analyzer.

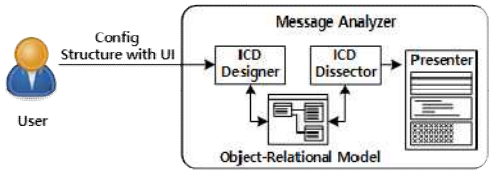

Fig. 16 Proposed ICD message analyzer.

Fig. 17은 ICD 디자이너의 컨트롤 중 하나를 보여준다. 사용자는 메시지 구조를 특정 언어를 통해 설계하고 기존의 프로토콜 분석기에 추가하는 과정 없이 AFDX 프로토콜 분석기가 제공하는 ICD 디자이너의 UI 컨트롤을 통해 직관적으로 메시지 구조를 정의하고 수정할 수 있다.

Fig. 18은 이전 그림의 ICD 디자이너에서 정의된 GPS PVT 메시지가 ICD 메시지 분석기를 통해 해석되어 GUI에 표시된 것을 보여준다.

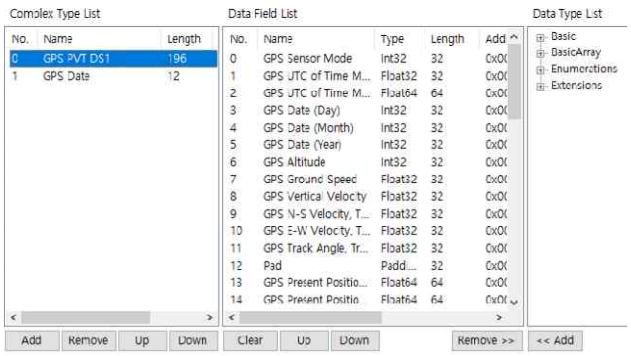

Fig. 17 Example of User-defined ICD Design.

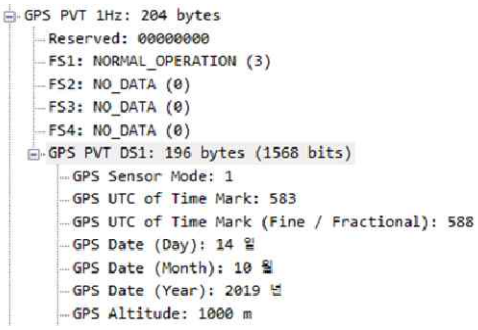

Fig. 18 Example of User-defined GPS PVT ICD.

표 1은 기존의 방법들로 구현된 메시지 분석기들과 AFDX 프로토콜 분석기의 차이점을 비교하여 보여준다. 방법 A는 그림 6-(a)와 같이 XML, ASN.1 등을 사용하여 메시지 구조를 정의한 뒤 코드를 생성하고 메시지 분석기와 함께 빌드하는 방법을 가리키고 방법 B는 그림 6-(b)과 같이 Lua를 사용하여 Dissector Function을 작성하여 메시지 분석기에 입력하는 방법을 가리킨다.

Table. 1 Comparison of implementation methods

	방법 A	방법 B	AFDX 프로토콜 분석기
요구되는 언어	1개 (XML)	1개 (Lua)	없음
코드 수정	어플리케이션 수정	플러그인 수정	필요 없음
Dissector 알고리즘	분석기에 포함	구현 필요	분석기에 포함
Designer 도구	별도의 편집 도구 활용 (XML, MS Access)	별도의 편집 도구 활용 (Lua, Code Editor)	전용 GUI Designer 제공
참조 무결성	보장하지 않음	보장하지 않음	보장
절차 단계 수	4	3	2

먼저 방법 A는 메시지 구조를 추상화하기 위해 XML을 사용하고 방법 B는 바이트 배열로부터 메시지를 분석하는 Dissector Function을 작성하기 위해 Lua를 사용한다. 반면에 AFDX 프로토콜 분석기는 사용자가 어떠한 컴퓨터 언어도 구사할 필요가 없으므로 사용자의 편의성과 접근성이 높다.

방법 A으로 구현된 기존의 한 시스템에서는 사용자가 XML 파일을 직접 작성하는 대신 추가적인 도구를 사용하는 방법이 사용되기도 하였는데 그것은 MS Access를 통해 데이터 구조를 정의하고 DB파일로 저장한 뒤 그것을 다시 별도의 도구를 사용하여 XML파일을 생성하는 것이다. 이 방법에서는 사용자가 XML를 구사해야할 필요는 없어졌지만 MS Access라는 도구와 DB파일을 XML파일로 변환하는 도구를 추가적으로 사용해야한다.

또한 방법 A는 생성한 추상화 파일을 분석기 어플리케이션에 적용하기 위해서 어플리케이션을 다시 빌드하여 수정해야 한다. 그리고 방법 B는 작성한 Lua를 어플리케이션에 플러그인으로 적용시키기 위한 코드 작업과 플러그인 삽입 작업이 필요하다.

한편 각 방법에 따라 Dissector 알고리즘의 구현 방법에도 차이가 있다. 방법 B는 사용자가 Lua를 사용하여 바이트 배열을 분해하고 Presenter에 표시하는 Dissector 알고리즘을 Dissector 함수에 정의한다. 반면에 방법 A와 AFDX 프로토콜 분석기는 메시지의 구조만을 메시지 분석기가 요구하는 규칙에 따라 정의하며 Dissector 알고리즘은 메시지 분석기에 포함되어 있다. 그리고 방법 A와 B는 메시지 구조 또는 Dissector Function을 정의하기 위해 별도의 편집기나 개발 도구를 사용해야한다. 그러나 AFDX 프로토콜 분석기는 메시지 구조를 정의하기 위한 전용 GUI Designer를 구현된 어플리케이션에서 제공한다.

5.2 테스트케이스 생성 및 자동화 기능

기존에는 테스트케이스를 설계한 뒤 테스트케이스별로 검증 장비의 코드를 직접 작성해야 했다. 이러한 작업은 시간과 비용이 많이 소요되며 유지보수가 어렵다. 하지만 AFDX 인증 지원 시스템은 미리 프로그래밍 된 동작을 체계적으로 조직화하는 테스트케이스 생성 도구를 제공한다.

이를 사용하면 테스트 동작을 쉽게 구성 가능하다. 미리 만들어 놓은 테스트케이스를 복사하여 다른 입력을 반복적으로 수행하는 테스트를 쉽게 추가할 수 있으며 복잡한 테스트케이스도 체계적으로 구성하고 관리할 수 있다. 또한 항공기 안전 공학에 따르면 사용자 과실 (Human Factor)은 안전을 위협하는 요인 중 하나이다 [8]. 테스트케이스를 통해 검증하는 과정에서 시스템의 자동화는 테스트 설계자 및 테스트 실행자의 실수를 최소화할 수 있다.

AFDX 인증 지원 시스템은 AFDX의 송수신 기능 검증 및 지터 검증을 자동화하는 AFDX 테스터 (Tester) 도구 및 테스트 항목 편집기 (Test Operation Editor) 도구를 제공한다. 사용자는 AFDX 테스터에서 송수신 포트와 패킷 송수신 방식 및 검증할 데이터 등의 파라미터를 정의함으로써 테스트케이스에서 수행할 테스트 항목을 정의한다. AFDX 테스터는 테스트 항목이 수행되었을 때 해당 테스트 항목의 종류와 파라미터 값 그리고 수행 결과를 비교하여 성공/실패를 결정한다.

테스트 항목이 1개 이상 모여 테스트케이스를 구성하며 테스트케이스를 실시하면 포함되는 모든 테스트 항목이 동시에 수행된다. 테스트 항목이 모두 종료되면 테스트케이스가 종료되며 모든 테스트 항목의 결과가 성공일 때만 테스트케이스의 수행 결과가 성공으로 된다. 테스트케이스의 테스트 항목 중 하나라도 실패이면 그 테스트케이스의 결과는 실패가 된다.

Table 2는 AFDX의 송수신 기능을 테스트하기 위해 구성한 테스트 항목의 종류와 각 테스트 항목이 가지는 파라미터를 보여준다.

첫번째, TxPacketsOp는 패킷 목록을 정해진 시간안에 송신하는 동작이다. 이 동작은 TxStartTime부터 TxEndTime까지의 시간 동안 모든 패킷의 송신을 완료하고 사용자가 최종적으로 결과를 확인하면 결과는 Pass가 된다. 패킷 목록의 모든 패킷들은 송신할 시간이 지정되어 있어야 한다.

두번째, RxCountOp는 정해진 시간안에 정해진 개수만큼의 패킷을 수신하는 동작이다. StartTime부터 EndTime까지 수신한 패킷 중 필터를 통과한 패킷의 개수가 RxCountMin이상이고 RxCountMax 이하이면 성공이다. Echo는 필터를 통과한 패킷을 수신했을 때 수신한 포트로 재송신하는 옵션이다. RxEvents는 패킷을 수신했을 때 추가적인 테스트 항목을 수행하기 위한 파라미터이다.

세번째, RxJitterOp는 수신한 패킷의 지터 유효성 검증을 수행하는 동작이다. 수신 동작은 RxCountOp와 유사하나 패킷을 수신했을 때 가상 링크 별로 분류하여 지터 검증을 수행하며 모든 패킷의 지터가 RxMaxJitter보다 작으면 성공이다.

네번째, LoopbackOp는 송신한 패킷을 다시 수신하는 동작이다. 송신한 모든 패킷을 주어진 시간 안에 모두 다시 수신하면 성공이다.

마지막 ComplexOp는 여러 개의 테스트 항목을 결합한 복합 테스트 항목이다. ComplexOp의 하위에 포함하는 모든 테스트 항목의 결과가 성공이면 해당 ComplexOp의 결과도 성공이 되며 하위 항목 중 하나라도 실패이면 해당 ComplexOp의 결과도 실패가 된다.

Table. 2 Peformance evaluation test matrix.

	Parameters	Description
TxPacketsOp	TxStartTime	송신 시작 시간
	TxEndTime	송신 종료 시간
	TxDevice	송신 디바이스
	TxPackets	송신할 패킷 목록
	AutoResult	송신 완료시 Pass
RxCountOp	RxStartTime	수신 시작 시간
	RxEndTime	수신 종료 시간
	RxDevice	수신 디바이스
	RxFilter	수신할 패킷을 위한 필터
	RxCountMin	수신해야 할 패킷의 최소 개수
	RxCountMax	수신해야 할 패킷의 최대 개수
	Echo	수신한 패킷의 재송신 여부
	TxDevice	재송신 시 송신 디바이스
	RxEvents	수신 시 추가 Operation 동작
RxJitterOp	RxStartTime	수신 시작 시간
	RxEndTime	수신 종료 시간
	RxDevice	수신 디바이스
	Filter	수신할 패킷을 위한 필터
	RxBAG	수신할 패킷의 지터 계산에 사용할 BAG
	RxJitterMax	수신해야 할 패킷의 최대 개수
	Echo	수신한 패킷의 재송신 여부
	RxEvents	수신 시 추가 동작
LoopbackOp	TxStartTime	송신 시작 시간
	RxEndTime	수신 종료 시간
	TxDevice	송신 디바이스
	RxDevice	수신 디바이스
	Packets	송신할 패킷 목록
ComplexOp	Operatoins	하위 Operation 목록

Fig. 19은 네트워크 장비를 위한 몇 가지 테스트케이스의 구성 방법을 도식화한 것이다. 생성한 테스트케이스를 실행하는 테스트 장비와 DUT (Device Under Test)가 AFDX 링크를 통해 연결된다.

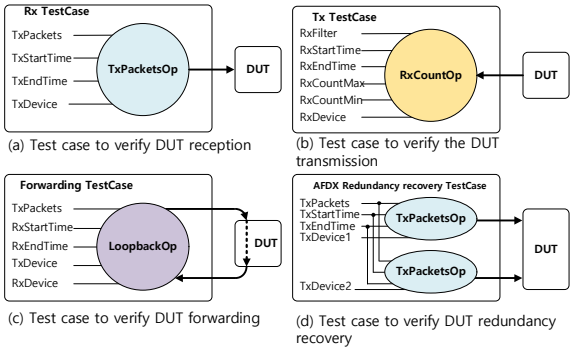

Fig. 19 Examples of test case configuration.

(a)는 DUT의 수신 기능을 검증하는 테스트케이스를 보여준다. 인증 지원 시스템에서는 패킷을 송신하는 동작을 하는 TxPackets 테스트 항목이 필요하다. 송신 인터페이스 (TxDevice)를 지정하고 DUT와 연결시킨 뒤 송신 패킷의 목록을 송신한 뒤 DUT에서 패킷의 수신을 확인하여 DUT의 수신 기능을 검증한다. DUT 수신단의 AFDX 프로토콜과 관련된 모든 기능을 검증하기 위해서는 송신 패킷 목록을 적절히 변화시켜야 한다.

예를 들어 특정 가상 링크에 대해서 수신 동작을 확인하기 위해서는 해당 가상 링크 ID를 가지는 패킷을 송신해야 하며 수신단의 패킷 손실 복구 기능을 검증하기 위해서는 정상적인 AFDX 시퀀스를 가지는 패킷 목록에서 시퀀스 패킷을 강제로 누락시켜야 할 것이다.

(b)는 DUT의 송신 기능을 검증하는 테스트케이스를 보여준다. 인증 지원 시스템에서는 DUT가 송신한 패킷을 수신하는 테스트 항목이 필요하다. 수신 인터페이스 (RxDevice)를 지정하고 DUT와 연결한 뒤 DUT가 송신한 패킷을 수신함으로써 DUT의 송신 기능을 검증할 수 있다.

(c)는 DUT의 포워딩 기능을 검증하는 테스트 케이스를 보여준다. TxPacketsOp와 RxCountOp를 함께 사용하여 구성할 수 있지만 송신 패킷과 수신 패킷이 동일한 경우 LoopbackOp를 사용하여 테스트케이스를 구성할 수도 있다. 송신 인터페이스 (TxDevice)를 DUT의 인터페이스와 연결하고 수신 인터페이스 (RxDevice)를 DUT의 다른 인터페이스와 연결한 뒤 포워딩 되어야할 패킷을 송신하고 그것을 다시 수신함으로써 DUT의 포워딩 기능을 검증할 수 있다. 이는 AFDX 스위치의 기능을 검증하는 데 사용할 수 있다.

(d)는 AFDX 엔드 시스템의 이중화 복구 기능을 검증하기 위한 테스트케이스를 보여준다. AFDX는 송신단에서 두 개의 인터페이스를 통해 동시에 패킷을 전송하며 수신단에서는 중복된 패킷을 폐기하고 먼저 수신된 하나의 패킷 만을 상위 시스템으로 전달하여 이중화 된 패킷을 복구한다. 이 기능을 검증하기 위해 인증 지원시스템에서는 두개의 TxPacketsOp이 필요하다. DUT에 해당하는 AFDX 엔드 시스템의 두 포트를 TxPacketsOp의 송신 인터페이스 (TxDevice1, 2)와 연결하고 동일한 패킷 목록을 두개의 인터페이스를 통해 동시에 송신함으로써 AFDX 패킷 이중화 송신을 재현한다. 그리고 DUT에서 이중화 복구의 결과를 확인함으로써 이중화 복구 기능을 검증할 수 있다.

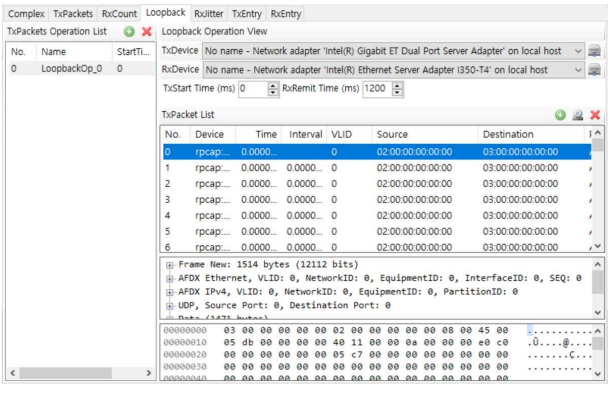

Fig. 20 Example of operation editor.

Fig. 20은 LoopbackOp를 정의하기 위해 AFDX 인증 지원 시스템에서 제공하는 테스트 항목 편집기 컨트롤을 보여준다.

6. 결 론

항전장비의 고도화로 인해 AFDX와 같은 신기술이 기존의 ARCIN429 및 MIL-STD-1553B 기술을 대체하고 있다. 하지만, AFDX 기술을 이용해 개발한 항전 시스템의 네트워크 기능을 효과적으로 검증하기 위한 실용 기술이 부족하였다. 본 논문에서는 이런 문제를 극복하기 위해 여러 네트워크 분석 솔루션을 통합한 AFDX 항전 시스템 인증 실용화 기술을 제안하였다.

첫번째 AFDX 인증 실용화 기술은 시험 장비의 개조 없이도 100-ns 단위의 타임스탬핑을 제공하는 정밀 AFDX 패킷 수집 시스템이다. 기존의 수십 μs 단위의 캡처 솔루션보다 수십 배 정밀하게 분석이 가능하다.

두번째 AFDX 인증 실용화 기술은 AFDX 정합성을 검증할 수 있는 프로토콜 분석기이다. AFDX 고유의 규격인 BAG 전송 간격 및 최대 지터 등의 성능 지표 준수 여부를 분석할 수 있다. 기존에는 100% 해외 제품에 의존하던 것을 국산화하였다.

마지막으로 AFDX 인증 실용화 기술은 개발 및 검증 업무의 편의성을 개선시키기 위한 추가 기능으로서, ICD 규격의 변경을 수월하게 할 수 있는 ICD 사용자 편집기와 자동으로 테스트 케이스를 생성, 편집 및 자동 수행할 수 있는 테스트 케이스 편집기이다. 인간의 개입을 최소화하고 테스트 과정을 자동화하여 검증의 신뢰도를 향상시키고 검증을 위한 시스템의 구현과 실행에 걸리는 시간을 절약할 수 있다.

따라서 본 논문에서 제안하는 AFDX Tap과 프로토콜 분석기 통합 인증 지원 솔루션을 이용할 경우 AFDX 항전 장비의 개발 시간과 비용을 줄이고 항전 제품의 품질을 높일 수 있다.

Acknowledgment

본 연구 중 네트워크 모니터링 소프트웨어 개발은 국토교통부가 지원하는 철도기술연구사업(과제번호: 20RTRP-B123310-06)의 지원을 받았으며 AFDX Tap 및 AFDX 프로토콜 분석기 개발은 산업통상자원부가 지원하는 항공우주부품기술개발사업 (과제번호: 20003238)의 일환으로 수행되었습니다.

References

[1] J. Lee, Y. An, C. Kim, D. Park, S. Yun, S. Lee and J. Park, “A Study on Aircraft Parts, Components and Training System Certification System,” Defense Agency for Technology and Quality, Sep. 2010.

[2] D. Ko, N. Choi, M. Kang, K. Kim and G. Ryu, “Development of Civil Aircraft Used in KC-100 Aircraft Experience,” Cheongmoongak (Gyomoonsa), Dec. 2013.

[3] Vance Hilderman and Tony Baghai, Avionics certification – A complete guide to DO-178 (Software) DO-254 (Hardware), Avionics Communications Inc., USA, 2008.

[4] Peter Heise, Iris Gaillardet, Haseeb Rahman, Vijay Mannur, “Avionics Full Duplex Ethernet and the Time Sensitive Networking Standard,” IEEE 802.1 Interm Meeting, May 2015.

[5] MEN, AFDX From Component to Application, Feb. 2017.

[6] J. Eo, “Performance Evaluation and Implementation of AFDX Packet Monitoring System with IEEE 1588 Precision Time Protocol,” Korea Aerospace University, July 2012.

[7] IEEE Std 1588-2008 − IEEE Standard for a Precision Clock Synchronization Protocol for Networked Measurement and Control Systems, IM/ST − TC9 − Sensor Committee Technology, Mar. 2008.

[8] I. Melnyk, P. Yadav, M. Steinbach, J. Srivastava, V. Kumar and A. Banerjee, "Detection of Precursors to Aviation Safety Incidents Due to Human Factors," 2013 IEEE 13th International Conference on Data Mining Workshops, 2013, pp. 407-412, doi: 10.1109/ICDMW.2013.55.