

네트워크 시뮬레이터(NS-3)에서 정보통신설비 BIM 라이브러리의 속성정보를 활용한 시뮬레이션 방법

김현덕, 심상균, 김승수, 김한별

한국정보통신산업연구원

force80@kici.re.kr, sksim@kici.re.kr, sskim@kici.re.kr, khanb@kici.re.kr

A Simulation Method Utilizing Property Information from ICT Facility BIM Libraries in Network Simulator (NS-3)

Hyunduk Kim, Sangkyun Sim, Seungsuh Kim, Seul Lee, Hanbyeol Kim

Korea Information & Communication industry Institute

요약

초연결사회로 진입함에 따라 건축물 내 정보통신설비의 중요성과 복잡성은 급격히 증가하고 있다. 스마트 빌딩, 데이터센터, 병원, 공항 등과 같은 주요 시설에서는 고성능의 네트워크 인프라 구축이 필수적이며, 이에 따라 설계 초기부터 운영 단계까지 전 생애주기에 걸쳐 체계적인 관리가 요구된다. 특히, 네트워크 연결 구조에 대한 사전 검토와 성능 예측은 설비의 안정성과 효율성을 확보하는 핵심 요소로 부각되고 있다. BIM은 건축물과 설비의 3차원 형상정보와 속성정보를 통합하여 설계, 시공, 유지관리 등 전 과정에 걸쳐 통합적인 정보관리 체계를 제공하는 기술이다. 그러나 현재까지 BIM은 건축 및 기계·전기·배관(MEP) 중심으로 활용되고 있으며, 정보통신설비 분야와의 연계는 미흡한 실정이다. 본 논문에서는 정보통신설비의 특성을 반영한 BIM 라이브러리를 개발하고, 해당 라이브러리에서 IFC 형식으로 추출한 속성정보를 네트워크 시뮬레이터인 NS-3에 적용하는 연동 방안을 제안한다. 이를 통해 실제 건축 공간 내 네트워크 성능을 설계 단계에서 사전 검증하고, 시공 전 최적화할 수 있는 디지털 기반 시뮬레이션 워크플로우를 구현할 수 있음을 확인하였다.

I. 서론

BIM(Building Information Modeling)은 건축물과 설비의 3차원 형상 정보와 속성 데이터를 통합하여 설계, 시공, 유지관리 등 건물의 전체 생애주기를 포괄하는 정보관리 체계를 제공한다. 정보통신설비 BIM 라이브러리는 케이블, 분배기, 라우터, 스위치 등의 형상과 속성정보를 구조화함으로써, 디지털 환경에서 실제와 유사한 네트워크 인프라의 설계와 시각화를 가능하게 한다[1]. 그러나 다양한 BIM 저작도구 간의 데이터 호환성과 확장성을 보장하기 위해서는 개방형 표준의 적용이 필수적이며, 이러한 역할을 수행하는 것이 buildingSMART에서 제정한 개방형 데이터 포맷인 IFC(Industry Foundation Classes)이다[2]. IFC는 설비의 물리적 형상뿐 아니라 속성, 시스템 연결 관계 등을 구조화하여 표현할 수 있는 플랫폼 독립적인 데이터 모델로, 건축 및 설비 정보를 다양한 시스템 간에 연계할 수 있게 한다. 특히 정보통신설비의 배치 위치, 연결 구조, 대역폭, 주파수, 전력 등의 기술 속성을 IFC 속성(Property Set)으로 정의함으로써, 향후 시뮬레이션 및 성능 분석에 활용할 수 있는 데이터 기반을 제공한다.

한편, NS-3(Network Simulator 3)는 현실적인 네트워크 구성과 트래픽을 가상 환경에서 재현하고 분석할 수 있는 네트워크 시뮬레이터로, 네트워크 설계 및 성능 평가에 널리 활용되고 있다. 하지만 기존 NS-3 기반 시뮬레이션은 네트워크 토폴로지나 장비의 속성정보를 사용자가 수동으로 정의해야 하며, 실제 설비의 위치나 물리적 특성을 정확히 반영하는 데 한계가 존재하였다.

이에 본 연구에서는 IFC 기반 정보통신설비 BIM 라이브러리로부터 설비의 형상 및 속성정보를 자동으로 추출하고, 이를 NS-3에 연동하여 실제 건축물 내 설치 환경을 반영한 네트워크 시뮬레이션 모델을 자동화하는 방법을 제안한다. 이를 통해 BIM 기반의 네트워크 설계 정보가 시뮬레이션 도구에서 직접 활용될 수 있도록 하고, 나아가 다양한 BIM 라이브러리

리를 시뮬레이션에 적용함으로써 그 활용성과 확장 가능성을 검토하였다. 연구 결과, BIM 라이브러리에 포함된 속성정보를 기반으로 NS-3 시뮬레이션을 수행할 수 있음을 확인하였으며, 이를 바탕으로 네트워크 시뮬레이션의 가시화 및 디지털 기반의 사전 성능 검증이 가능함을 입증하였다. 이는 정보통신설비의 설계 정합성을 향상시키고, 실제 구축 전 최적화를 위한 유효한 분석 도구로서의 활용 가능성을 제시한다.

II. 본론

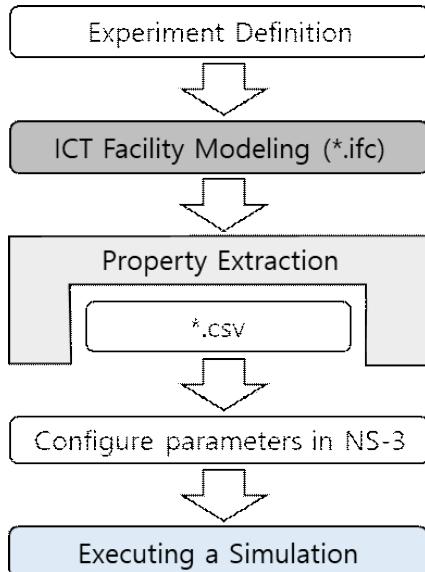
정보통신설비 BIM은 정보 통신의 설계부터 시공, 유지보수까지 필요한 다양한 형태의 속성정보 값을 가지고 있다. 특히 정보 통신에 특화된 다양한 응용 애플리케이션을 사용하기 위해 각각의 라이브러리의 속성정보에 디바이스의 물리적 특성, 네트워크 채널 구성, 사용 프로토콜 등의 대한 정보가 포함될 수 있다.

Property		
Level 2.	Level 3.	Expression
Wireless Communication	WirelessConnectionType	Wi-Fi (str)
	WirelessInterface	802.11g (str)
	MaximumThroughput	56 Mbps (int)
	Frequencyband	2.4 Ghz (int)
	TransmissionPower	19 dBm (int)
	NumberOfAntenna	4 (int)
	AntennaGain	5 dBi (int)
	MinimumReceiveSensitivity	-80 dBm (int)
	CommunicationTechnology	MIMO (str)

[그림 1] 무선 AP의 BIM라이브러리 속성정보(예시)

이러한 정보통신설비 BIM 라이브러리의 경우 다양한 플랫폼에서의 데이터를 교환·공유하기 위해 buildingSMART International(bSI)에서 표준화한 IFC를 참고하여 라이브러리를 생성하였기 때문에, 다양한 환경에서 활

용할 수 있는 Open-BIM 형태로 구성되었으며, 이러한 ifc 파일의 경우 쉽게 파일에 대한 형상 및 속성정보를 활용할 수 있다는 장점이 있다. 따라서 정보통신설비 BIM 라이브러리에 포함된 다양한 속성정보를 네트워크 시뮬레이터의 파라미터 값으로 활용하게 되면 실제 현장에 설치되는 설비의 특성을 고려하여 네트워크에 대한 시뮬레이션을 수행할 수 있을 것으로 판단된다. 그러나 현재 NS-3에서 직접 ifc 파일을 읽거나 처리할 수 있는 모듈이 존재하지 않기 때문에 그림2와 같은 프로세스에 따라 ifc 파일에서 필요한 속성정보 값을 추출하여 csv 파일로 저장하고, 저장된 csv 파일을 NS-3의 C++ 코드에서 읽어온 후 파라미터값으로 적용할 수 있다.



[그림 2] 속성정보 추출 및 적용 프로세스

이러한 속성정보 추출 및 적용 프로세스를 바탕으로, 무선AP의 전송범위에 대한 시뮬레이션이 진행하기 위해 무선AP에 대한 모델링을 진행하였으며, 필요한 속성정보를 입력하였다. 이후 그림3과 같이 속성정보를 추출할 수 있는 코드를 적용하였다.

```

# PropertySets
for rel in device.IsDefinedBy:
    if rel.is_a("IfcRelDefinesByProperties"):
        pset = rel.RelatingPropertyDefinition
        if pset.is_a("IfcPropertySet"):
            for prop in pset.HasProperties:
                if prop.is_a("IfcPropertySingleValue"):
                    pname = prop.Name.lower()
                    val = prop.NominalValue.wrappedValue if prop.NominalValue else None

                if "txpowerstart" in pname:
                    tx_start = val
                elif "txpowerend" in pname:
                    tx_end = val
                elif "txgain" in pname:
                    tx_gain = val
                elif "rxgain" in pname:
                    rx_gain = val
                elif "energydetectionthreshold" in pname or "energythresh" in pname:
                    energy_thresh = val

DeviceId,TxPowerStart,TxPowerEnd,TxGain,RxGain,EnergyDetectionThreshold
AP01,18.0,18.0,1.0,0.0,-96.0
AP02,15.0,15.0,2.0,1.0,-95.0
  
```

[그림 3] 속성정보 추출 코드(일부) 및 저장된 csv파일 구조

실제 무선AP를 모델링한 ifc 파일에서 추출한 데이터는 그림3과 같은 형태의 csv 파일로 저장하였다. 각 무선AP의 식별을 위한 ID, 송신 거리를 위한 전력 범위(dBm), 송신/수신 안테나 이득(dB), 무선 신호 감지 임계

값(dBm)을 추출하였다. 이렇게 추출된 csv 파일을 NS3의 PHY계층 파라미터로 적용하기 위해서 그림4와 같은 코드를 통해 시뮬레이션에서 활용될 AP마다 해당 파라미터값을 적용하였다.

```

phy.Set("TxPowerStart", DoubleValue(phyParams[i].txStart));
phy.Set("TxPowerEnd", DoubleValue(phyParams[i].txEnd));
phy.Set("TxGain", DoubleValue(phyParams[i].txGain));
phy.Set("RxGain", DoubleValue(phyParams[i].rxGain));
phy.Set("EnergyDetectionThreshold", DoubleValue(phyParams[i].energyThresh));
  
```

[그림 4] csv기반 파라미터 적용 코드 예시(일부)

이후, 그림5와 같이 NS-3의 시뮬레이션 순서에 따라, 현재 설치된 AP의 무선전송 범위 및 거리에 따른 신호 세기 등을 측정할 수 있는 시뮬레이션을 수행하였으며, 시뮬레이션 결과 각 파라미터값이 AP마다 제대로 반영된 것을 확인할 수 있었다.

