

내장형 BACnet 프로토콜 스택 및 제어기의 개발

BACnet은 1995년 미국 ANSI/ASHRAE에 의해 개발되어 한국 산업표준(KS) 및 ISO 국제표준(IS)으로 인준된 빌딩자동화 및 제어를 위한 표준 통신 프로토콜로서 최근 빌딩자동화 분야의 핵심 기술로 부각되고 있다. 한양대학교 자동화통신망 연구실, 바스코리아, 삼성SDS, 아이콘트롤스 컨소시엄은 내장형 BACnet 프로토콜 스택 및 제어기를 개발하였다.

한양대학교 자동화통신망 연구실, 바스코리아, 삼성SDS, 아이콘트롤스

최근의 지능형 빌딩에서는 공기조화, 조명, 전력, 위생, 수송, 방재, 방범 및 주차 등의 다양한 설비들에 대한 자동화시스템의 요구가 높아지고 있다. 따라서 첨단 빌딩자동화시스템은 빌딩 내에 설치되는 각종 설비들을 실시간으로 감시하고 제어할 수 있어야 하며, 이를 위하여 빌딩 설비들간의 제어 및 자동화 관련 정보가 통신망을 통하여 실시간으로 전달 및 처리되어야 한다. 그러나 기존의 빌딩자동제어 분야에서는 장비 제조업체마다 서로 다른 통신 프로토콜을 사용하거나, 자체적으로 개발한 독자적인 통신 프로토콜을 적용하였기 때문에 서로 다른 제조업체의 장비들간에 통신 및 상호운용이 불가능하였다. 이로 인해 사용자는 장비 공급업체에 기술적으로 종속되어 시스템의 구축이나 운용 및 확장에 제한을 받을 수밖에 없었다. 이러한 문제점을 해결하고, 빌딩 자동화 시스템을 사용자 중심의 기술로 발전시키기 위하여 1980년대 중반부터 빌딩 자동화 통신망의 표준 기술에 대한 필요성이 제기되었다.

BACnet(Building Automation and Control network)은 빌딩 자동화 및 제어용 통신 기술을 공급 업체에 의존적인 기술에서 벗어나 사용자 중심의 기술로 발전시키고 서로 다른 제조업체의 장비들 사이에 상호운용성(InterOperability)을 보장하기 위하여 1995년에 ASHRAE(American Society of Heating, Refrigerating

and Air-conditioning Engineers)에 의해 개발된 빌딩 자동화 전용 표준 통신망 프로토콜이다. BACnet은 2003년 ISO에서 빌딩 자동화 및 제어 시스템의 국제 표준 통신 프로토콜(ISO16484)로 채택되었으며, 국내에서도 이미 1999년 빌딩자동화 통신망의 KS 표준 규격(KS X 6909)으로 채택되었다.

그러나 국내의 경우 BACnet에 대한 원천기술이 아직 확보되지 않아 많은 업체들이 외국 기술이나 제품에 의존적일 수밖에 없는 실정이었다. 이러한 문제점을 극복하기 위해 한양대학교 자동화통신망 연구실(지도교수: 홍승호)은 바스코리아, 삼성SDS, 아이콘트롤스와의 컨소시엄을 구성하여 2년여의 연구 끝에 Embedded BACnet 프로토콜 스택과 제어기를 개발하였다. 이는 2003년 국내 최초로 윈도우 운영체제 기반의 Operator Workstation용 BACnet 프로토콜 스택을 자체 개발한데 이어 내장형 제어기를 위한 BACnet Protocol 스택과 제어기의 프로토타입을 개발한 것으로서 빌딩자동제어 분야에서 표준 BACnet 기술을 통한 국내 자체 기술력과 국제 경쟁력을 확보하는데 크게 기여할 것으로 기대된다.

BACnet 개요

BACnet은 그림 1과 같이 물리계층, 데이터링크계



층, 네트워크계층 및 응용계층으로 구성되는 4계층 구조를 가지고 있다. BACnet은 빌딩 자동화 설비들 간에 교환되는 다양한 정보들을 응용계층의 객체(Application Object)로 추상화 하고 있으며, 이러한 객체들을 다루고 조작하기 위한 다양한 응용계층 서비스(Application Service)를 표준에 포함하고 있다.

또한 BACnet은 기존 빌딩자동화 시스템에서 널리 사용되는 Ethernet, ARCnet, MS/TP, PTP, LonTalk 등의 통신기술을 데이터링크 계층의 5가지 선택사항으로 포함함으로써 시스템 통합에 대한 확장성 및 유연성을 확보하고자 하고있다. 따라서 사용자는 원하는 시스템의 성능, 비용 및 기능요구사항 등을 고려하여 보다 다양하고, 효율적인 통합 빌딩 자동화 시스템을 구성할 수 있다.

앞서 기술한 바와 같이, BACnet에서는 빌딩 자동화 시스템에서 사용되는 다양한 정보들을 객체를 사용하여 추상화한다. 이러한 객체는 단순히 빌딩 자동화 시스템에서 사용되는 입·출력 정보뿐만 아니라, 자동제어를 수행하기 위한 제어 알고리즘과 원격 감시를 위한 이벤트 알고리즘 등을 포함한다. 표 1은 BACnet에서 정의하고 있는 표준 객체의 종류를 보여준다. 이와 같은 표준 BACnet 객체를 사용하여 네트워크상의 모든 장비와 장비 내부의 정보를 표현하기 때문에, 서로 다른 제조업체에서 공급된 장비들이라 할지라도 그 내부 동작에 관계 없이 동일한 방식으로 정보에 접근할 수 있다.

이와 더불어, BACnet에서는 여러 범주의 표준 서비스를 정의하고 있다. BACnet 표준 서비스란, 네트워크상에 존재하는 다양한 객체들로부터 정보를 얻어내고 그 값을 조작하여 전체 빌딩 자동화 및 제어 시스템을 운용하기 위한 수단이다. 표 2는 BACnet에

BACnet Layers				Equivalent OSI Layers
BACnet Application Layer				
BACnet Network Layer				Network
ISO 8802-2 (IEEE 802.2) Type 1	MS/TP	PTP	LonTalk	Data Link
ISO 8802-3 (IEEE 802.3)	ARCNET	EIA - 485 EIA - 232		Physical

[그림 1] BACnet 프로토콜 계층 구조와 OSI 참조 모델

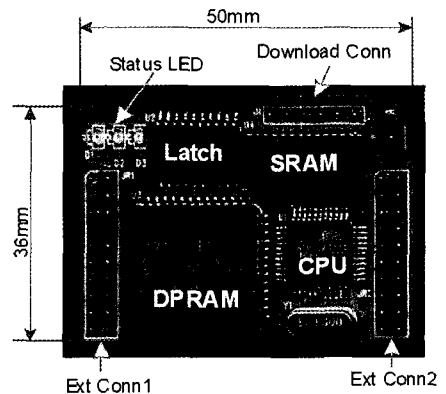
서 정의하고 있는 5가지 범주의 표준 BACnet 서비스와 제공하는 기능을 간략히 보여준다.

내장형 BACnet 제어기 개발

본 연구를 통해 개발된 Embedded BACnet 제어기의 하드웨어는 크게 CPU모듈, MS/TP 통신 모듈, 그리고 하드웨어 입/출력 부분으로 구분된다. CPU 모듈로는 Motorola사의 MPC860SAR CPU가 탑재되어 있으며, 하나의 Ethernet 포트와 하나의 시리얼 포트, 그리고 모든 어드레스버스와 데이터 버스, 컨트롤 시그널 라인이 커넥터를 통해 연결된다.

그림 2는 본 프로젝트를 통해 개발된 MS/TP 통신 모듈의 사진이다. MS/TP 통신 모듈은 BACnet MS/TP 데이터링크 계층의 통신 기능을 제공하는 모듈로, ATmel사의 ATmega8515 CPU를 사용하여 BACnet MS/TP 계층의 기능만을 담당하는 Duather Board 형태로 개발하였다. BACnet 제어기는 일반적인 SRAM 접근 방식으로 MS/TP 통신 모듈의 DPRAM(Dual Port RAM) 영역을 통하여 데이터를 주고 받으며, 데이터 교환을 위해 하나의 인터럽트를 사용한다.

그림 3은 본 연구를 통해 개발된 BACnet 제어기 프로토타입의 사진이다. 제어기의 프로토타입은 MS/TP모듈 및 CPU모듈을 Duather Board의 형태로 탑재하고 있으며, 전원부, AI, AO, DI, DO 하드웨어 입출력부, Ethernet 및 시리얼포트 등의 통신포트들



[그림 2] MS/TP 통신 모듈

<표 1> BACnet에서 제공하는 표준 객체

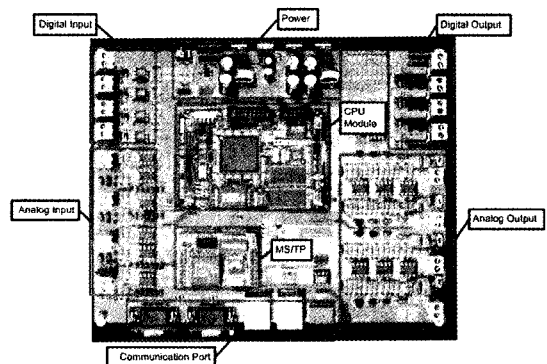
객체명	이날로그 센서 입력값 등을 표현.
Analog Input	아날로그 센서 입력값 등을 표현.
Analog Output	아날로그 액추에이터의 출력 값을 표현.
Analog Value	내부적으로 사용되는 아날로그 변수를 표현.
Binary Input	스위치 등 온/오프 입력 값을 표현.
Binary Output	릴레이와 같은 온/오프 출력 값을 표현.
Binary Value	내부적으로 사용되는 이진 변수를 표현.
Multi-state Input	여러 단계로 동작하는 장비의 상태를 표현. 예) 다중 속도 팬
Multi-state Output	여러 단계로 동작하는 장비의 출력 값을 표현.
Multi-state Value	정수 값을 가지는 내부 변수를 표현.
Average	여러 입력 객체의 평균값을 유지하기 위하여 사용.
Device	네트워크 상에서 특정 장비를 대표하며, 장비의 기본 정보를 표현.
Group	여러 객체를 한번의 서비스로 읽어 들이기 위해 그룹화하는데 사용.
Command	한번에 여러 객체의 값을 변경시키기 위해 그룹화하는데 사용.
Program	장비 내부에서 동작하는 프로그램의 현재 상태를 표현.
Event Enrollment	현재 등록된 algorithmic reporting 이벤트 목록을 표현.
File	장비 내에 존재하는 파일에 접근 가능하도록 하기 위하여 사용.
Life Safety Point	화재 감지 포인트와 같이 매우 중요한 곳의 관제점을 표현.
Life Safety Zone	여러 Life Safety Point을 그룹화 하는데 사용.
Loop	표준화된 PID 제어 알고리즘을 제공.
Calendar	스케줄에 사용되는 시간 정보를 표현.
Schedule	특정 시간에 자동으로 수행되는 명령을 미리 등록하는데 사용.
Trend log	지정된 시간 동안의 이력 정보를 저장하기 위하여 사용.
Notification Class	이벤트 발생시 이를 알릴 대상 장비를 지정하기 위하여 사용.

<표 2> BACnet에서 제공하는 표준 서비스

BACnet 서비스 범주	제공하는 기능
객체 접근 서비스	BACnet 객체의 값을 읽거나 변경하는 기능 BACnet 객체를 생성시키거나 삭제하는 기능
원격 장비 관리 서비스	BACnet 장비를 검색/인식하기 위한 기능 원격으로 BACnet 장비를 초기화하는 기능
파일 접근 서비스	BACnet 네트워크를 통한 파일 조작 기능
경보 및 이벤트 서비스	경보 및 알람 발생을 보고하는 기능
가상 터미널 서비스	가상 터미널 기능

탑재하고 있다.

임베디드 시스템의 경우, 소프트웨어적으로 사용할 수 있는 리소스가 한정되어 있기 때문에 간단한 시스템의 경우에는 운영체제를 사용하지 않는 것이



[그림 3] BACnet Controller

일반적이다. 그러나 BACnet과 같이 매우 복잡한 형태의 통신 프로토콜의 송수신 기능을 기본적으로 처



리하면서 동시에 입출력제어, 제어로직 수행 등 여러 가지의 어플리케이션 기능을 수행하기 위해서는 멀티태스킹 환경이 반드시 필요하다. 따라서 본 프로젝트에서는 이러한 기능요구사항과 추후 상용화 및 확장성을 고려하여 임베디드 리눅스의 일종인 HardHat Linux를 운영체제로 채택하여 적용하였다.

또한 BACnet 제어기에는 입출력 접점으로 표 3과 같이 각각 4채널의 AI, AO, DI, DO 포인트를 수용하고 있다. 각각의 하드웨어 입출력 접점들은 표준 BACnet Object들에 연결되어 관리된다.

프로토콜 스택 및 BACnet 소프트웨어 개발

임베디드 BACnet 소프트웨어는 그림 4와 같이 크게 BACnet 어플리케이션, 프로토콜 스택, 임베디드 리눅스 드라이버의 3부분으로 구성된다. 가장 상위에 있는 부분은 BACnet 어플리케이션으로 BACnet 통신을 이용하는 사용자 계층의 부가기능이 탑재된다. BACnet 프로토콜 스택 소프트웨어에서는 이러한 BACnet 어플리케이션에서 쉽게 BACnet 프로토콜 스택을 이용할 수 있도록 다양한 형태의 API를 제공한다.

BACnet 프로토콜 스택 부분은 다양한 종류의 BACnet 서비스를 제공하며 BACnet 프로토콜을 실행하는 가장 핵심적인 부분으로 BACnet 오브젝트 및 서비스의 관리 기능과 인코딩(Encoding) 디코딩(Decoding) 기능 등 모든 프로토콜 처리 기능이 구

현되어 있다.

임베디드 리눅스 디바이스 드라이버는 각종 하드웨어 자원을 사용할 수 있도록 하드웨어와 소프트웨어의 중간 매개체 역할을 담당한다. 이는 임베디드 리눅스 개발환경에서 구현되는 모든 소프트웨어에 필수적으로 요구되는 사항으로 하드웨어 사양에 의존적일 수 밖에 없게 된다. 디바이스 드라이버로는 임베디드 리눅스 에서 기본적으로 시리얼통신 포트 드라이버와 이더넷 통신 드라이버 외에 추가로 MS/TP 통신모듈을 구동하기 위한 드라이버와 각종 입출력 하드웨어를 구동하기 위한 드라이버가 모듈 형태로 탑재된다. 따라서 하드웨어 인터페이스가 바뀌더라도 상위 프로토콜 스택과 BACnet 어플리케이션을 수정할 필요없이 변경된 하드웨어 대한 리눅스 드라이버만 수정하면된다.

그림 5는 BACnet 프로토콜 스택의 내부 구조이다. BACnet 프로토콜 스택은 유연성 및 확장성을 고려하여 ANSI C/C++를 이용하여 객체지향적으로 구현되었으며, 운영체제에서 제공하는 시스템함수의 사용을 최소화하고 대부분 필요한 기능을 직접 구현함으로써 향후 개발환경의 변화에 대한 이식성을 최대한 보장하고자 하였다.

그림에서 알 수 있듯이, 임베디드 BACnet 프로토콜 스택 소프트웨어는 내부적으로 독립적인 여러 프로세스(스레드)로 동작하며, 각각의 프로세스는 길이에 제한이 없는 별도의 송/수신 버퍼를 가지고 있다. 이렇게 함으로서 한쪽 프로세스의 시간 지연이

<표 3> BACnet 제어기의 하드웨어 입출력 사양

하드웨어 입출력	채널수	신호범위	대응 BACnet 객체
아날로그 입력	4	전압 : 0~10 V 전류 : 4~20 mA RTD : PT1000옴	Analog Input Object
아날로그 출력	4	전압 : 0~10 V 전류 : 4~20 mA	Analog Output Object
디지털 입력	4	Dry Contact	Binary Input Object
디지털 출력	4	Relay : (SPST) 5A 250 VAC / 5A 30 VDC	Binary Output Object
직렬통신	2	9600 bps ~ 25600 bps	
Ethernet	2	10/100 M	

다른 쪽 프로세스의 동작에 직접적으로 영향을 미치지 않도록 할 수 있으며, 각 프로세스의 동작 알고리즘을 독립적으로 설계할 수 있다. 또한 향후 기능 추가가 필요한 경우, 추가된 기능이 영향을 미치는 범위를 최소화하여 수정시간을 단축시킬 수도 있다. 이러한 이유로 인하여, 내부적으로 독립적인 여러 프로세스가 병렬로 처리하는 방식을 사용하였다.

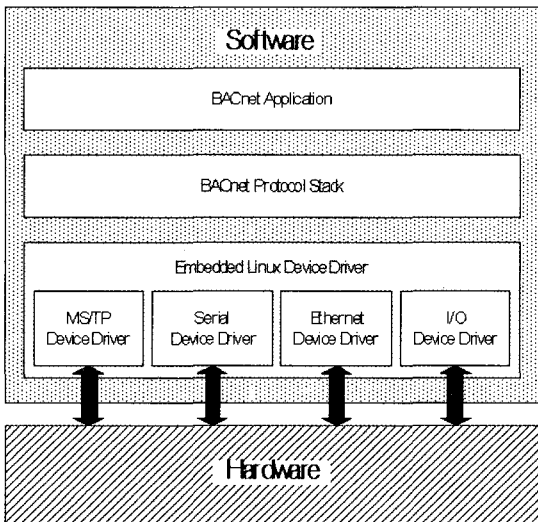
하부 계층에는 MS/TP 통신 모듈과 이더넷 포트를 통하여 메시지를 송/수신하기 위한 프로세스가 존재한다. 이들 프로세스는 MS/TP 통신 모듈의 경우 별도로 개발된 MS/TP 통신 모듈 드라이버(리눅스 드라이버)를 통하여 메시지를 교환하며, Ethernet의 경우에는 Socket을 사용하여 교환한다.

하부 계층과 BACnet 어플리케이션 계층 사이에는 라우팅 기능을 수행하는 프로세스가 별도로 존재한다. 이 프로세스에서는 송/수신되는 메시지의 어드레스를 분석하여, 어플리케이션 계층 프로세스로 전달하거나 곧바로 다른 통신 포트로 전달하는 역할을 수행한다.

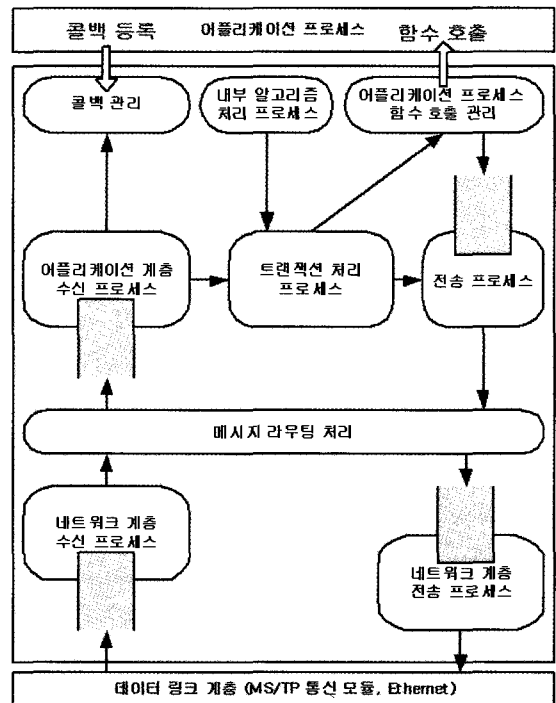
어플리케이션 계층 프로세스는 어플리케이션 계층에서 생성된 메시지를 하부 계층으로 송신하거나, 하부 계층에서 수신한 메시지를 적당한 처리 프로세스로 전달하는 역할을 수행한다. 특히 트랜잭션 처리

프로세스는 서비스 요청을 한 후, 지정된 시간동안 타임 아웃이 발생하는지 여부를 감시하여 이를 처리한다. 또한 동시에 여러 개의 서비스 요청을 관리하여, 하나의 서비스 처리가 지연된다 하더라도 다른 서비스 메시지가 계속 처리될 수 있도록 하였다.

BACnet 프로토콜 스택 소프트웨어는 어플리케이션 프로세스와 크게 두 가지 방법으로 인터페이스를 한다. 먼저 API함수를 직접 호출 방법이 있는데, 이는 어플리케이션 프로세스에서 필요한 경우 BACnet 라이브러리에서 제공하는 함수를 호출하여 BACnet 통신 기능을 사용하는 방법이다. 이는 주로 클라이언트 기능(서비스를 요청하는 기능)에 해당한다. 이와 반대로 콜백 등록 방식은 사용자가 특정 BACnet 서비스 요청이 수신될 경우, 이를 처리할 알고리즘을 미리 프로토콜 스택에 등록해 놓으면, 프로토콜 스택에서 자동으로 필요한 시기에 이를 호출하여 수행해주는 기능을 말한다. BACnet 프로토콜 스택 소프트웨어는 개발되는 모든 BACnet 서비스에 대하



[그림 4] Embedded BACnet 소프트웨어 구조



[그림 5] BACnet 프로토콜 스택 구조



여, 함수 호출 기능과 콜백 기능을 모두 개발하였다.
표 4, 5는 본 개발 프로젝트를 통해 개발된 BACnet 프로토콜 스택이 제공하는 표준 BACnet 서비스와 오브젝트들이다.

기능검증 및 시스템 통합 시험

BACnet 제어기의 기능검증을 위해서는 제어기의 자체 동작에 대한 검증 외에 반드시 BACnet 상호운용성에 대한 기능검증이 필요하다. 또한 시스템의 통합시험을 통해 통신규격에서 정의된 다양한 응용서비스들이 유기적으로 동작하여 시스템이 정상적으로 운용되는 지에 대한 검증도 필요하다. 따라서 본 기능검증 및 시스템 통합시험에서는 다음과 같은 장치와 도구들을 활용하여 기능검증시험을 수행하였다.

VTS (Visual Test Shell) for BACnet

VTS는 미국 표준기술연구소 NIST(National Institute of Standard and Technologies)의 후원으로 개발되어 무상으로 배포되는 BACnet을 위한 표준 기능검증 도구로 기본적으로 BACnet 프로토콜 어날라이저 기능과 메

<표 4> BACnet Service 개발 내역

BACnet Service	Initiate	Execute
Who-Is	○	○
I-Am	○	○
Who-Has	○	○
I-Have	○	○
TimeSynchronization	○	○
Read Property	○	○
Read Property Multiple	○	○
Write Property	○	○
Write Property Multiple	○	○
Unconfirmed COV Notification	○	○
Confirmed COV Notification	○	○
Unconfirmed Event Notification	○	○
Confirmed Event Notification	○	○
Acknowledge Alarm	○	○
Subscribe COV	○	○
Subscribe COV - Canceling	○	○

시지 제너레이터 기능을 제공한다. 그리고 시험결과를 로그파일로 기록하여 분석할 수 있어 대부분의 BACnet 개발자들이 공통적으로 활용하는 기능검증 도구이다. **그림 6**과 **그림 7**은 VTS를 이용하여 기능을 검증하는 화면과 테스트 로그파일의 예이다.

ORCAView 및 DSC 1616 Controller

ORCA시스템은 캐나다의 Delta Controls사에서 개발한 BACnet 기반의 빌딩자동제어 시스템으로서 국내에는 (주)바스코리아를 통해 HVAC, 조명제어, 출입통제 시스템을 공급하고 있다. ORCAView는 이 시스템에서 사용되는 HMI 소프트웨어로서 BACnet 제어/감시 기능과 시스템관리기능 등을 제공한다. 본 시험에서는 개발된 BACnet 제어기를 ORCAView를 이용해 제어/감시하고 관리함으로써 서로다른 공급업체의 제품간에 상호운용성이 보장됨을 검증하였다. 또한 DSC1616 제어기는 BACnet Ethernet과 MS/TP를 동시에 지원하면서 추가적으로 Ethernet-MS/TP 라우터 기능도 함께 제공한다. 따라서 DSC1616 제어기를 BACnet 제어기 및 BACnet 라우터로 활용하였다. **그림 8**과 **그림 9**는 각각 ORCAView에서 BACnet 시스템을 인식하고 관리하기 위한 BACnet Navigator 화면과 HMI 사용자 인터페이스 화면의 예이다.

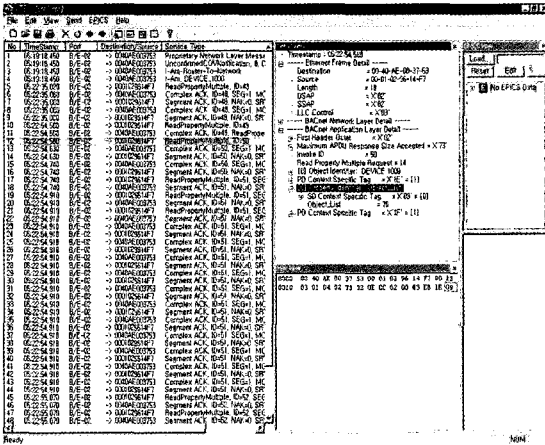
ControlCity

ControlCity는 삼성SDS에서 공급하는 통합 빌딩자

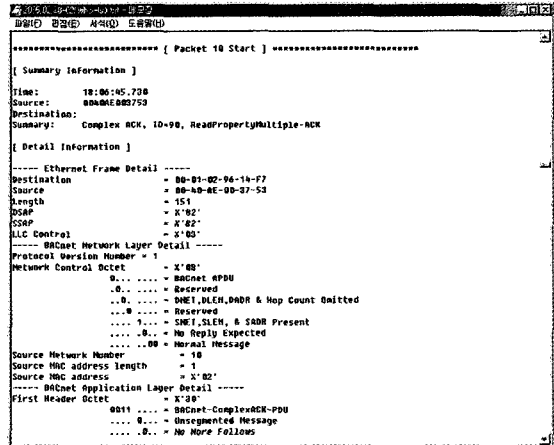
<표 5> BACnet 오브젝트 개발 내역

BACnet Object	R/W	COV	Intrinsic	Runtime
Analog Input	○	○	○	-
Analog Output	○	○	○	-
Analog Value	○	○	○	-
Binary Input	○	○	○	○
Binary Output	○	○	○	○
Binary Value	○	○	○	○
Device	○	-	○	-
Notification Class	○	-	-	-
Calendar	○	-	-	-
Schedule	○	-	-	-

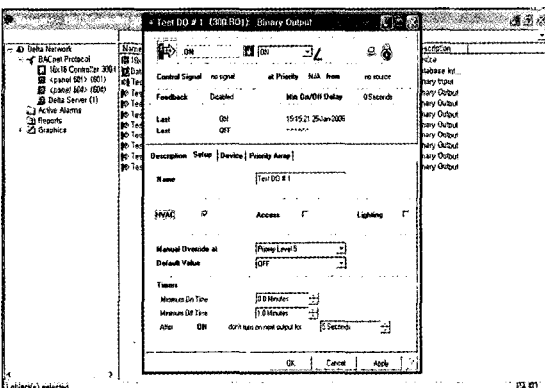
- : 규격에서 기능 지원하지 않음



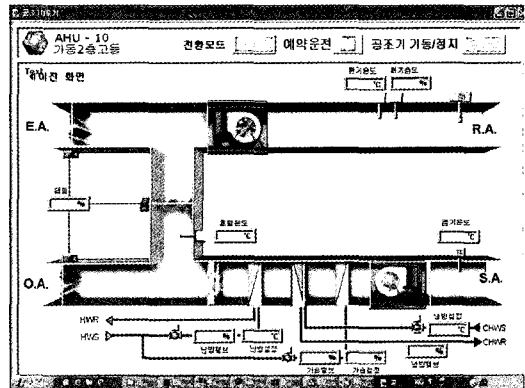
[그림 6] VTS를 이용한 기능검증 화면



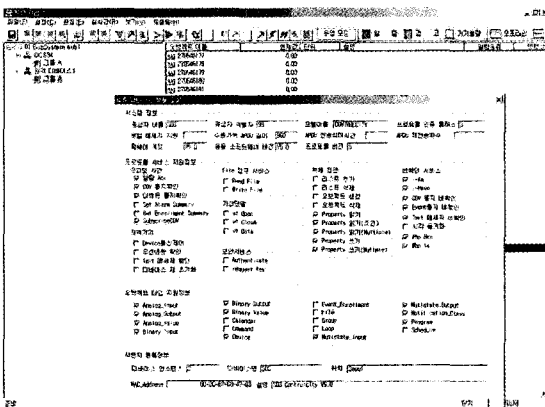
[그림 7] VTS 테스트 로그 파일의 예



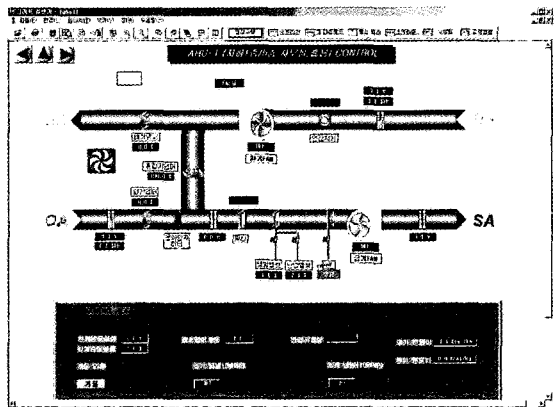
[그림 8] ORCAview의 BACnet Navigator 화면



[그림 9] ORCAview의 HMI 사용자 인터페이스 화면



[그림 10] ControlCity의 BACnet 관리기 화면



[그림 11] ControlCity의 HMI 사용자 인터페이스 화면



동제어 시스템으로 고유 통신방식과 함께 BACnet 통신기능을 선택적으로 추가할 수 있다. 따라서 앞서 언급한 ORCAview와 같이 BACnet 제어가 ControlCity 시스템과도 표준 BACnet 통신을 통해 상호 운용이 가능한지 시험하였다. 그림 10과 그림 11은 각각 ControlCity의 BACnet 관리기 화면과 HMI 사용자 인터페이스 화면의 예이다.

emdDDC

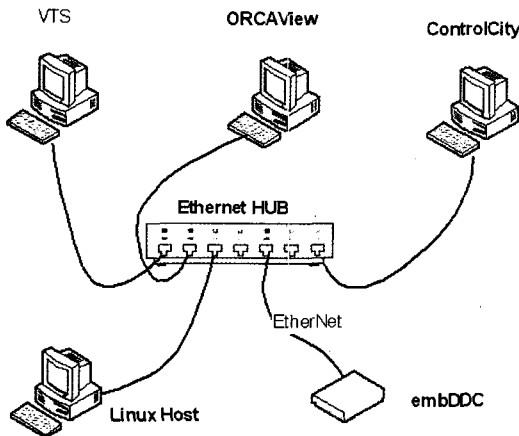
본 프로젝트를 통해 개발된 내장형 BACnet 제어기로 BACnet Ethernet 제어기, BACnet MS/TP 제어기 및 BACnet 라우터 기능을 제공한다.

Linux Host

내장형 BACnet 제어기인 emdDDC의 임베디드 리눅스 실행환경을 설정하고 제어기의 내부동작을 모니터링 하기 위해 사용된다.

테스트결과

본 프로젝트를 통해 개발된 임베디드 BACnet 제어기는 크게 BACnet Ethernet 제어기, BACnet MS/TP 제어기 그리고 BACnet Ethernet-MS/TP 라우터의 기능을 수행할 수 있다. 따라서 개발된 BACnet 제어기의 기능을 검증하고 타 시스템과의 기능 연동을 검



[그림 12] BACnet Ethernet 제어기 테스트 시스템 구성도

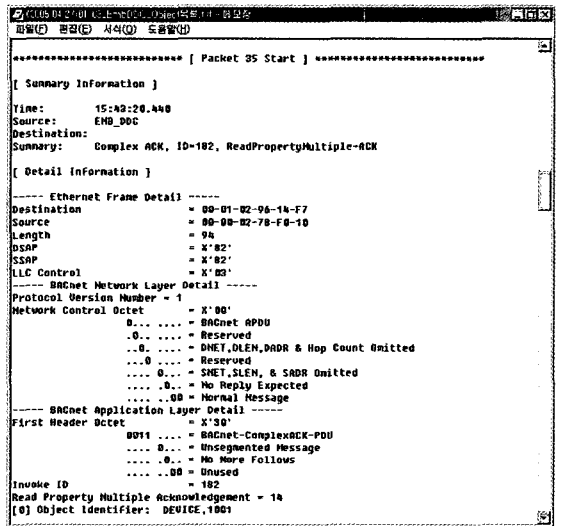
증하기 위하여 다음 그림 12와 같이 3가지 형태의 테스트 시스템을 구성하여 시험하였다.

(1) BACnet Ethernet 제어기

BACnet Ethernet 제어기는 BACnet Ethernet을 통해 연결되는 단일 제어기로 동작한다. 이를 위해 제어기에는 앞서 소개한 BACnet 프로토콜 스택과 간단한 제어 및 감시가 가능한 데모 어플리케이션 프로그램을 탑재하였다. 그림 12는 BACnet Ethernet 제어기의 기능을 검증하기 위한 테스트 시스템의 구성도이다.

우선 기능검증을 위해 VTS(Visual Test Shell)를 이용하여 표준 BACnet 서비스들이 정상적으로 수행되는지 확인하였다. 그림 13은 VTS를 이용해 BACnet 제어기의 Object 목록을 확인하고 Device를 인식하기 위한 시험의 결과의 로그파일이다. 로그 파일에서 볼수 있듯이 emdDDC가 BACnet Ethernet을 통해 수신된 Read Property Multiple 서비스에 대해 Complex ACK로 응답하고 있음을 알수 있다.

또한 BACnet HMI 소프트웨어인 ORCAview와 ControlCity 시스템과의 제어 및 감시 등 상호운용에 대해서도 검증하였다. 우선 BACnet Who-IS/I-Am



[그림 13] VTS를 이용한 BACnet Ethernet 제어기 기능시험의 예

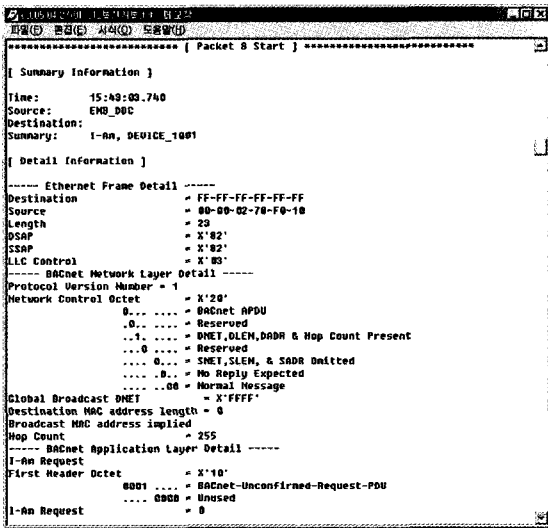
서비스를 이용해 Delta ORCAView와 삼성SDS ControlCity에서 BACnet 제어기를 자동으로 인식함을 확인하였으며, ORCAView의 개별제어 및 일괄제어 기능을 이용해 BACnet 제어기의 AO 및 DO Object를 제어한뒤 그 값이 ORCAView와 ControlCity의 제어감시 화면에 동일하게 모니터링 되고 동시에 이에 해당하는 BACnet 서비스들이 정상적으로 교환되는지를 VTS를 통해 확인하였다. **그림 14**는 Who-Is/I-Am 서비스를 통해 BACnet 제어기를 인식하는 동작에 대한 VTS 로그 화면이며, **그림 15**는 ORCAView와 ControlCity에서 BACnet 제어기를 제어/감시할 때 발생하는 WriteProperty 및 ReadProperty 서비스를 VTS로 확인한 로그파일의 예이다.

(2) BACnet MS/TP 제어기

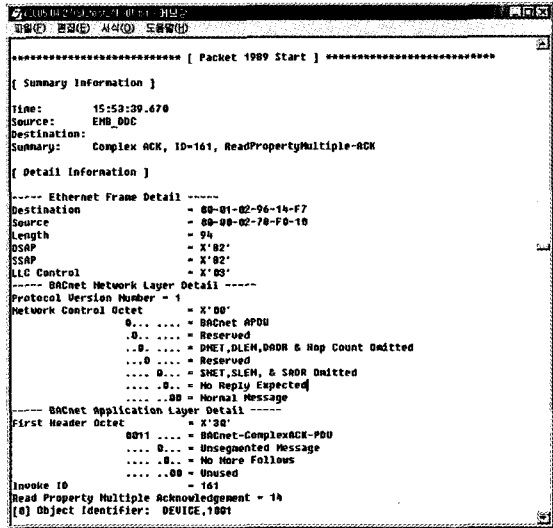
BACnet MS/TP제어기는 BACnet MS/TP를 통해 연결되는 단일 제어기로 동작한다. 앞서 소개한 BACnet Ethernet 제어기와 마찬가지로 BACnet MS/TP 제어기에도 BACnet 프로토콜 스택과 간단한 제어 및 감시가 가능한 동일한 데모 어플리케이션 프로그램을 탑재하였다. DSC1616 제어기는 BACnet Ethernet과 MS/TP를 동시에 지원하면서 Ethernet-

MS/TP 라우터 기능을 제공한다. 따라서 본 시험에서 BACnet MS/TP제어기를 BACnet Ethernet에 연결하기 위해 DSC1616 제어기의 라우터 기능을 이용하였다. 따라서 상위 VTS 및 ORCAView, ControlCity 등 BACnet Ethernet에서 발생된 메시지들은 DSC1616 제어기를 통해 MS/TP 네트워크로 라우팅되며, 반대로 MS/TP 네트워크에서 발생된 메시지는 1616 제어기를 통해 BACnet Ethernet으로 라우팅된다. **그림 16**은 BACnet MS/TP 제어기 기능을 검증하기 위한 테스트 시스템의 구성도이다.

역시 기본적인 상호운용성 검증을 위해 VTS를 이용하여 표준 BACnet 서비스에 대한 요청 및 응답이 DSC1616 제어기의 라우터 기능을 거쳐 BACnet MS/TP 제어기에의해 정상적으로 수행되는지 시험하였다. 또한 ORCAView와 ControlCity 시스템과의 제어 및 감시 등 상호운용에 대해서도 검증하였다. 우선 BACnet Who-IS/I-Am 서비스를 이용해 ORCAView와 ControlCity에서 BACnet 제어기를 자동으로 인식함을 확인하였으며, ORCAView의 개별 제어 및 일괄제어 기능을 이용해 BACnet 제어기의 AO 및 DO Object를 제어한뒤 그 값이 ORCAView와 ControlCity의 제어감시 화면에 동일하게 모니터링



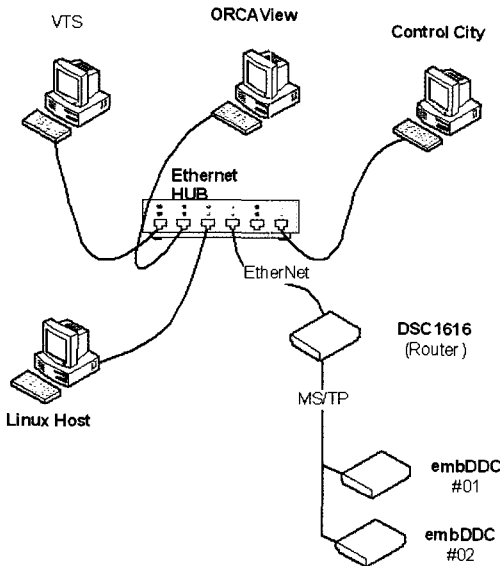
[그림 14] Who-Is/I-am 서비스를 통한 BACnet Ethernet 제어기의 인식



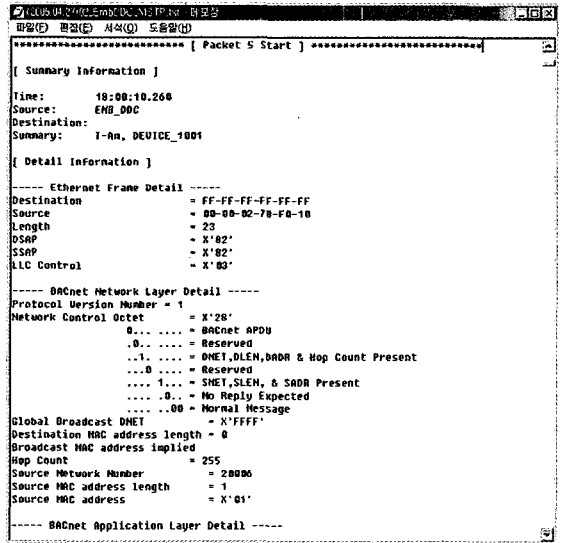
[그림 15] BACnet Ethernet 제어기의 제어/감시 기능시험 로그파일

되고 동시에 이에 해당하는 BACnet 서비스들이 정상적으로 교환되는지를 VTS를 통해 확인하였다. 그림 17은 Who-Is/I-Am 서비스를 통해 BACnet MS/TP 제어기를 인식하는 동작에 대한 VTS 로그 화

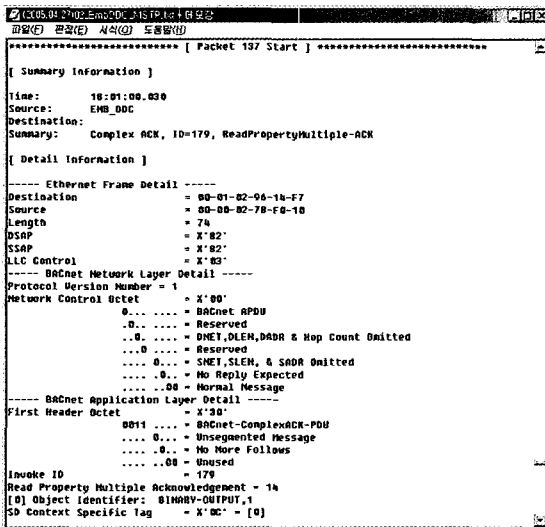
면이며, 그림 18은 ORCAView와 ControlCity에서 BACnet MS/TP 제어기를 제어/감시할 때 발생하는 WriteProperty 및 ReadProperty 서비스를 VTS로 확인한 로그파일의 예이다.



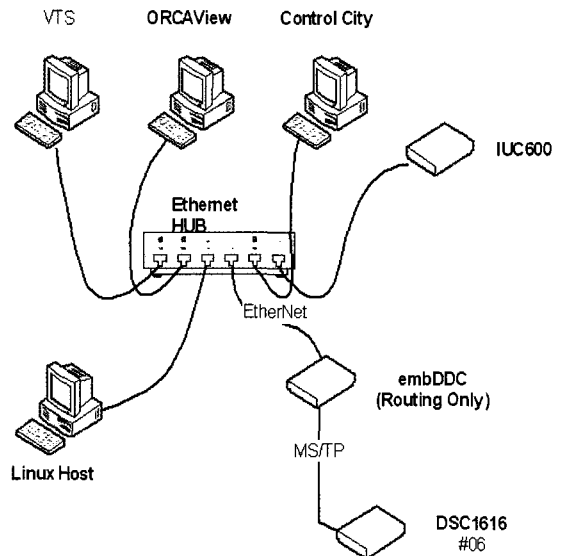
[그림 16] BACnet MS/TP 제어기 테스트 시스템 구성도



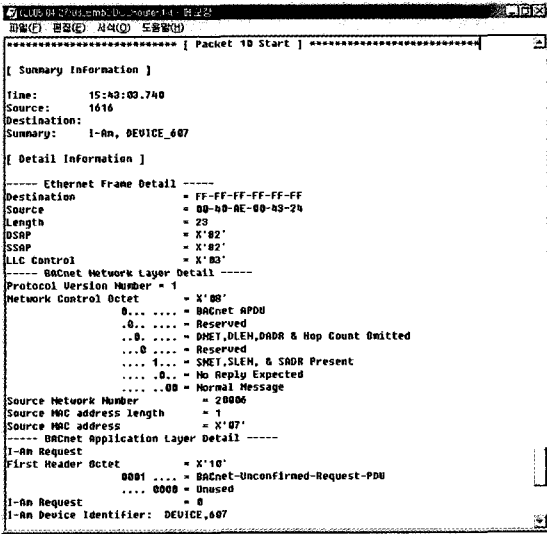
[그림 17] Who-Is-I-am 서비스를 통한 BACnet MS/TP 제어기의 인식



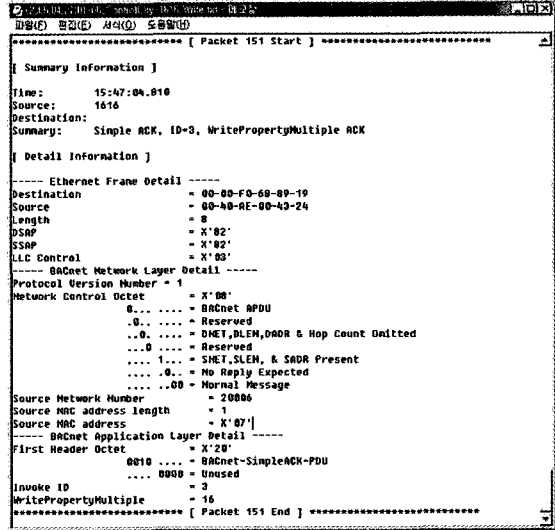
[그림 18] BACnet MS/TP 제어기의 제어/감시 기능시험 로그파일



[그림 19] BACnet Router 테스트 시스템 구성도



[그림 20] 라우터를 통해 전달되는 Who-Is/I-am 서비스의 VTS 로그화면



[그림 21] 라우터를 통해 전달되는 제어/감시 기능의 VTS 로그화면

(3) BACnet EtherNet - MS/TP 라우터 기능

본 프로젝트를 통해 개발된 BACnet 제어기 프로토타입은 Ethernet 통신기능과 MS/TP 통신 기능을 동시에 지원할 수 있으며, BACnet Ethernet-MS/TP 라우터 기능이 독립적으로 구현되어있다. 따라서 본 시험에서는 BACnet 어플리케이션 계층의 기능 없이 순수하게 BACnet 네트워크 계층에서 동작하는 BACnet Ethernet과 BACnet MS/TP 네트워크간의 라우터 기능에 대한 검증을 수행하였다. BACnet 라우터에는 상위 네트워크로 BACnet Ethernet이 연결되고 하부에는 DSC1616 제어기가 BACnet MS/TP를 통해 연결된다. 그림 19는 BACnet 라우터 기능을 검증하기 위한 테스트 시스템의 구성도이다.

우선 기본적인 상호운용성 검증을 위해 VTS를 이용하여 표준 BACnet 서비스에 대한 요청 및 응답이 embDDC의 라우터 기능을 거쳐 DSC1616 제어기에 정상적으로 전달되고, DSC1616 제어기에 의한 응답 역시 embDDC의 라우터 기능을 거쳐 BACnet MS/TP 네트워크로부터 BACnet Ethernet 네트워크로 전달됨을 확인하였다. 또한 ORCAview와 ControlCity 시스템과의 제어 및 감시 등 상호운용에 대해서도 검증하였다. 우선 ORCAView 및

ControlCity에서 BACnet 제어기를 자동으로 인식하기 위해 발생하는 Who-Is/I-Am 서비스들이 embDDC의 라우터 기능을 통해 BACnet Ethernet 네트워크와 BACnet MS/TP 네트워크 간에 라우팅되어 전달됨을 확인하였다. 그리고 ORCAView 및 ControlCity의 개별제어 및 일괄제어 기능을 이용한 제어/감시 기능 역시 embDDC의 라우터 기능을 통해 BACnet MS/TP 네트워크 상의 DSC1616 제어기에 전달되어 정상적으로 수행됨을 확인하였다. 그림 20은 Who-Is/I-Am 서비스 요청 및 응답이 embDDC의 라우터 기능을 통해 DSC1616 제어기를 인식하는 동작에 대한 VTS 로그 화면이며, 그림 21은 ORCAView와 ControlCity에서 BACnet MS/TP 제어기를 제어/감시할 때 발생하는 WriteProperty 및 ReadProperty 서비스를 VTS로 확인한 로그파일의 예이다.

결론

이번 프로젝트를 통해 BACnet 프로토콜의 5가지 데이터링크 옵션 중 활용도가 가장 높은 BACnet Ethernet과 BACnet MS/TP를 동시에 지원할 수 있는 프로토콜 스택과 함께 응용계층 및 응용기능을 포함



하는 내장형 BACnet 제어기의 프로토타입을 함께 개발한 것으로서, 표준 BACnet 기술을 개발하기 위해 가장 절실히 요구되었던 내장형 BACnet 프로토콜의 구현기술과 이를 제어기에 적용하기 위한 응용 기술을 함께 확보할 수 있게 되었다.

Embedded BACnet 프로토콜 스택은 향후 이식성을 고려하여 표준 C/C++로 구현되었으며, 한국 산업표준 KS X 6909 및 ANSI/ASHRAE 135-1995에서 규정하고 있는 대부분의 응용 객체와 응용 서비스를 제공한다.

또한 본 연구에서는 VTS 뿐만 아니라 실제 상용제품으로 사용되고 있는 ORCA 시스템 및 ControlCity

시스템과의 다양한 상호운용성 테스트를 통해 그 기능을 검증함으로써 본 연구의 결과물이 바로 제품화에 적용될 수 있음을 확인하였다.

본 프로젝트는 최근 빌딩자동화 분야의 핵심기술로 부각되고 있는 BACnet 원천기술을 개발하기 위하여 바스코리아, 삼성SDS, 아이콘트롤스 등 관련 업체와 한양대학교가 컨소시엄을 구성하여 기술을 공동 개발함으로써 개발에 대한 부담을 줄이고 개발된 기술을 공유함으로써 산학협력을 통한 국내 원천기술의 확보 및 해외 기술경쟁력 확보의 계기를 만들었다는 점에서 그 의의 깊다고 할 수 있다. ㉓