

# 네트워크 기반 제어를 위한 LonWorks/IP 라우터의 설계 및 구현

## Design and Implementation of LonWorks/IP Router for Network-based Control

현진욱\*, 최기상\*\*, 최기흥\*\*\*  
 Jin Wook Hyun, Gi Sang Choi, Gi Heung Choi

**Abstract** - Demand for the technology for access to device control network in industry and for access to building automation system via internet is on the increase. In such technology integration of a device control network with a data network such as internet and organizing wide-ranging DCS(distributed control system) is needed, and it can be realized in the framework of VDN(virtual device network). Specifications for device control network and data network are quite different because of the differences in application. So a router that translates the communication protocol between device control network and data network, and efficiently transmits information to destination is needed for implementation of the VDN(virtual device network). This paper proposes the concept of NCS(networked control system) based on VDN(virtual device network) and suggests the routing algorithm that uses embedded system.

**Key words** : VDN(virtual device network), DCS(distributed control system), NCS(networked control system)

### 1. 서론

언제든지 장소와 관계없이 네트워크에 접속할 수 있는 유비쿼터스 시대가 도래하면서 산업현장과 빌딩의 전자기계장비뿐만 아니라 가정의 가전제품까지 네트워크에 연결되어 사용자가 원격으로 제어 및 관리, 감시할 수 있는 기술에 대한 수요가 생성되고 있다. 그래서 디바이스 제어 네트워크는 기존의 폐쇄적인 기술 기반에서 보다 지능적이고 개방적인 기술 기반으로 즉, 기존 IT 인프라와 통합되는 방향으로 진화하고 있다. 이를 위해서는 산업현장 기기들의 정보교환에 사용되는 디바이스 제어 네트워크와 인터넷에서 사용되는 데이터 네트워크의 통합이 필요하고 이렇게 두 네트워크가 통합된 형태를 가상 디바이스 네트워크(virtual device network)라 한다. 가상 디바이스 네트워크로의 통합에 있어 중요한 문제는 디바이스 제어 네트워크와 데이터 네트워크에서 요구되는 사양이 상당부분 다르다는 것이다. 그래서 두 네트워크 간의 데이터의 프로토콜을 변환하고 통신 패킷을 원격지까지 전송할 수 있는 라우터가 필요하다.

본 연구에서는 가상 디바이스 네트워크를 구성하기 위해 반드시 필요한 라우터의 개념과 틀을 제안한다.

### 2. 가상 디바이스 네트워크

가상 디바이스 네트워크(virtual device network)는 산업현장이나 빌딩 자동화 시스템(BAS, building automation system) 등에서 기기들 간의 정보교환에 사용되는 필드 버스 네트워크, 혹은 디바이스 제어 네트워크와 비즈니스 네트워크를 위한 인터넷에서 사용되는 데이터 네트워크가 통합된 네트워크이다. 디바이스 제어 네트워크의 사양은 데이터 네트워크의 사양과는 차이가 있다. 데이터 네트워크는 대량의 정보를 고속 전송하는 데 초점이 맞추어져 있지만 디바이스 네트워크는 분산 제어, 신뢰성 있는 통신, 데이터의 실시간 전송, 간편한 시스템 구성과 설치 등과 같은 산업 현장에서 적합한 응용에 초점이 맞추어져 있다. 이러한 차이점 때문에 현재 인터넷의 발달에도 불구하고 디바이스 제어 네트워크가 존재한다. 하지만 디바이스 제어 네트워크의 특성상 거리의 제한이 있고 사용자가 쉽게 접근할 수 없기 때문에 서로 다른 네트워크들로 이루어진 광범위한 분산 제어 시스템, 즉 가상 디바이

스 네트워크를 구현하기 위해서는 거리상의 제약이 없고 어디서나 접근이 가능한 데이터 네트워크와 통합해야 한다.

TCP/IP를 포함하는 인터넷 프로토콜 집합인 IP 네트워크는 기업을 위한 통합 네트워크이다. IP 네트워크를 통하여 서로 떨어져 있는 디바이스 제어 네트워크를 연결함으로써 많은 지역에 퍼져 있는 디바이스 제어 네트워크들이 이음새 없는 가상 디바이스 네트워크로 간단히 통합될 수 있다. 다시 말해 분리되어 있는 디바이스 제어 네트워크들이 데이터 네트워크를 매개로 하여 정보를 교환함으로써 가상적으로 하나의 네트워크처럼 구성되어 마치 바로 옆에 있는 것처럼 관리되는 네트워크(peer-to-peer network)를 말한다.

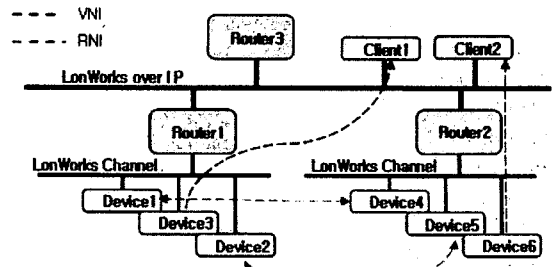


그림 1 가상 디바이스 네트워크의 일반적인 구조

그림 1에 가상 디바이스 네트워크의 일반적인 구조를 나타내었다. 가상 디바이스 네트워크의 핵심은 피어 투 피어(peer-to-peer) 네트워크라는 것이다.

데이터 네트워크를 통하여 원격지의 콘웍스 디바이스 제어 네트워크를 운영하고 관리하는 도구의 하나로 마이크로소프트사의 윈도우 호환 LNS(lonworks network services) 응용 프로그램을 들 수 있다. 이것은 미국 에셀론사에서 제공하는 LNS 응용 프로그램 개발 도구에 의해 개발되어 인터넷에 연결된 인터넷 클라이언트 PC에 설치된다. LNS 응용 프로그램은 가상 디바이스 네트워크를 구성하고 네트워크를 구성하는 개별 디바이스들 간의 정보 교환을 가능하게 하며 가상 디바이스 네트워크를 운영하고 관리하는데 이용된다. 이를 통하여 근거리는 물론 원격지에서 개별 디바이스를 제어하고 감시할

\* 현진욱 : 서울시立大學校 電子電氣컴퓨터工學部 碩士課程  
 \*\* 최기상 : 서울시立大學校 電子電氣컴퓨터工學部 教授  
 \*\*\* 최기흥 : 漢城大學校 機械시스템工學科 教授

수 있다. 이와 같은 방법으로 가상 디바이스 네트워크를 운영하고 관리하기 위해서는 디바이스 네트워크와 데이터 네트워크를 연결하고 있는 라우터가 필요하다.

### 3. LonWorks/IP 라우터

#### 3.1 LonWorks/IP 라우터의 개요

전체적인 LonWorks/IP 라우터 시스템의 기능은 서로 분리된 디바이스 제어 네트워크를 연결하여 가상 디바이스 네트워크 환경을 만들어 가상으로 연결된 디바이스 제어 네트워크 사이의 통신이다. LonWorks/IP 라우터의 하드웨어적인 구성은 데이터 네트워크 부분을 담당할 마스터 프로세서와 디바이스 제어 네트워크 부분을 담당할 슬레이브 프로세서로 나뉜다. 마스터 프로세서는 인텔사의 PXA250을 사용하였고, 슬레이브 프로세서는 에셀론사의 뉴런 프로세서 FT3150을 사용하였다. 그림 2에는 위에서 설명한 마스터 부분과 슬레이브 부분의 구성을 볼 수 있다.

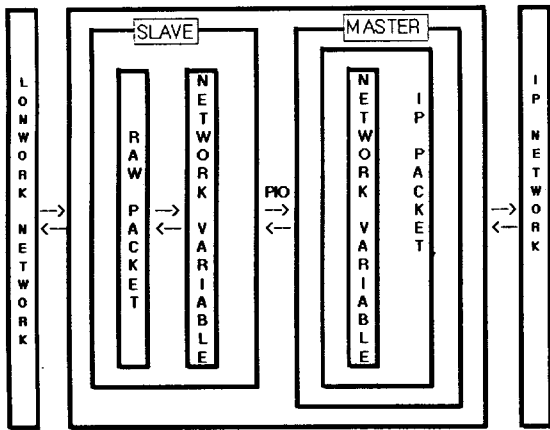


그림 2 LonWorks/IP 라우터 시스템 블록도

슬레이브 프로세서는 그림 3.2에서처럼 24개의 아날로그 입출력, 32개의 디지털 입출력, 총 56개의 론웁스 네트워크 인터페이스를 가지며 론웁스 네트워크의 로 패킷(raw packet)을 네트워크 변수(network variable)의 형태의 데이터로 변환하여 마스터 프로세서로 전송하거나 마스터 프로세서에서 네트워크 변수의 데이터를 받아 로 패킷으로 변환하는 역할을 한다. 마스터 프로세서와 슬레이브 프로세서가 통신하는 방식은 슬레이브 뉴런 프로세서의 펌웨어 라이브러리에서 제공하는 핸드셰이크 프로토콜(handshake protocol)의 병렬 I/O 통신 방식을 사용한다.

마스터 프로세서는 슬레이브 프로세서로부터 네트워크 변수의 데이터를 읽어 IP 패킷으로 변환하여 IP 네트워크를 통하여 목적지의 라우터로 보내거나 IP 네트워크에서 받은 IP 패킷으로부터 네트워크 변수의 데이터를 추출하여 슬레이브 프로세서로 전송한다. 슬레이브 프로세서로부터 얻은 패킷을 어떤 라우터로 보낼지는 라우팅 테이블에 기반을 두어 결정하게 되며 좀 더 빠른 라우팅을 위해 라우팅 알고리즘이 수행하게 된다. 또한, 라우터는 웹 기반 설정을 위해 웹 서버로서의 기능도 수행하게 된다. 마스터 프로세서의 소프트웨어는 이러한 여러 일들을 한꺼번에 효율적으로 수행하기 위해 멀티 쓰레드 환경이 요구된다. 이를 위해서는 운영체제가 필요하며 본 논문에서는 마이크로소프트사의 운영체제인 Windows CE.NET을 선택하였다. Windows CE는 멀티 쓰레드 환경을 제공하며, 라우터에 필요한 웹 서버도 간단히 구현할 수 있도록 라이브러리가 제공된다. 그림 3은 LonWorks/IP 라우터를 실제로 구현한 사진이다.

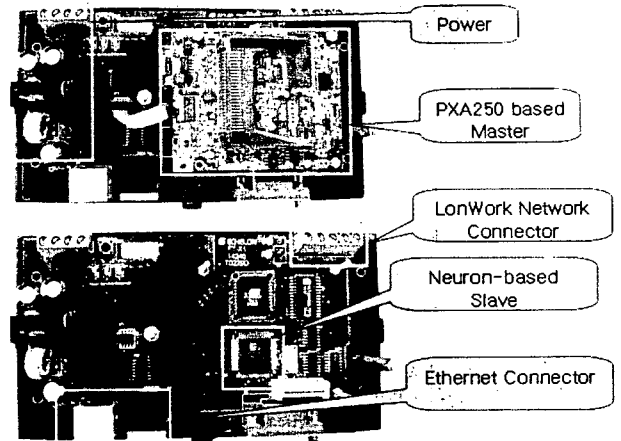


그림 3 개발된 라우터 사진

#### 3.2 LonWorks/IP 라우터 계층 간의 프로토콜

LonWorks/IP 라우터는 그림 4에서 보는 것처럼 CNL(component network layer), INL(intermediate network layer), IPL(IP network layer)의 3개의 계층으로 나누어 구현하였으며, CNL 계층은 마스터 프로세서와 슬레이브 프로세서를 모두 포함한다.

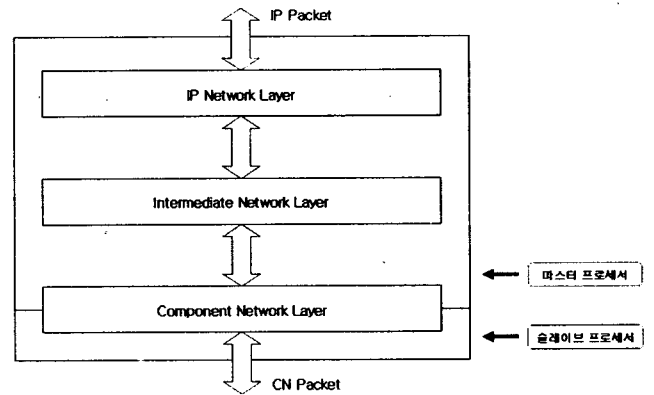


그림 4 LonWorks/IP 라우터 프로토콜 스택

마스터 프로세서의 소프트웨어에서 각 계층 간 데이터의 이동은 계층 간 공유된 데이터 저장 공간을 통하여 이루어진다. 다른 계층으로 데이터를 전달할 때는 데이터 저장 공간에 데이터를 넣고, 이 데이터를 받는 계층에서는 데이터 저장 공간에서 데이터를 꺼내서 처리를 하면 된다. 하지만 공유된 데이터이기 때문에 여러 계층에서 동시에 데이터 저장 공간에 접근 하는 경우가 생기게 되며 데이터가 깨질 수 있다. 그래서 안전한 데이터 처리를 위해 동기화 기법이 요구된다. 동기화란 복수의 쓰레드가 보조를 맞추어 실행하도록 함으로써 경쟁 상태(race condition)나 교착상태(deadlock)를 해소하는 것이다. 라우터의 마스터 프로세서는 하나의 프로세스로 구현되었으므로 동기화를 위해 크리티컬 섹션을 선택하였다. 크리티컬 섹션은 공유 자원에서의 접근과 같은 다른 쓰레드에 의해 방해받지 말아야 할 작업을 할 때 이 영역을 크리티컬 섹션으로 둘러싸 주어 공유 자원의 독점권을 보장해주는 것이다.

#### 3.3 라우팅 알고리즘

가상 디바이스 네트워크에서의 라우터의 역할은 서로 분리되어 있지만 가상으로 연결된 디바이스 제어 네트워크 사이의 CNL PDU의 교환이다. 라우터의 CN 계층에서 만들어지는 CNL PDU의 주 데이터는 네트워크 변수(network variable)라 불리는 룬웁스 네트워크에서 돌아다니는 일종의 변수이며, 라우터의 기본적인 역할은 이러한 네트워크 변수를 하나의 라우터에 속한 룬웁스 네트워크와 다른 라우터에 속한 룬웁스 네트워크와 주고받을 수 있도록 하는 것이다. 그러므로 라우터는 네트워크 변수 데이터에 기반을 두어 라우팅을 수행하게 된다. 라우팅을 하기 위해서는 가상으로 연결된 네트워크 간 네트워크 변수의 가상 연결 상태 정보가 필요하며, 이 정보는 라우터의 웹서버나 윈도우 기반 응용 프로그램을 통해 설정이 가능하다. 가상 연결을 위해서는 연결되는 라우터의 IP 주소와 룬웁스 네트워크의 디바이스들의 네트워크 변수 정보가 필요하다.

SOURCE NV	SOURCE IP	TARGET NV	TARGET IP	TYPE
SOURCE NV	SOURCE IP	TARGET NV	TARGET IP	TYPE
SOURCE NV	SOURCE IP	TARGET NV	TARGET IP	TYPE
...	...	...	...	...
SOURCE NV	SOURCE IP	TARGET NV	TARGET IP	TYPE

그림 5 연결 테이블 구조

라우터로 연결된 가상 디바이스 네트워크는 네트워크 변수 기반 가상 연결이기 때문에 라우팅을 수행하기 위해서는 네트워크 변수만 검사를 하면 된다. 네트워크 변수 기반 라우팅 과정을 그림 6에 나타내었다. 하나의 네트워크 변수가 두개 이상의 가상 네트워크 변수 연결을 가질 수 있기 때문에 하나의 CNL PDU에 대하여 연결 테이블을 처음부터 끝까지 검색할 필요가 있다.

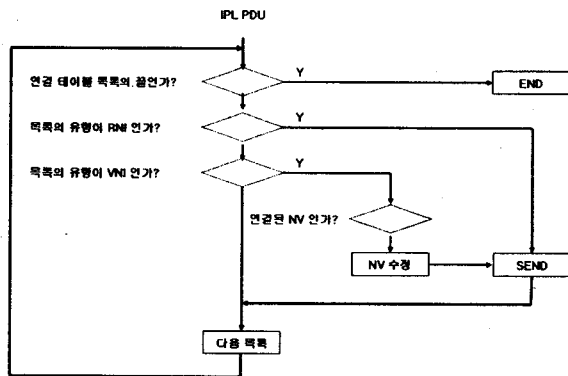


그림 6 라우팅 과정 블록도

VNI로 연결된 네트워크 변수의 경우에는 네트워크 변수의 번호를 타겟의 네트워크 변수 번호로 바꿔서 보냄으로서 패킷을 받은 쪽에서 네트워크 변수 값 수정을 위해 다시 한 번 연결 리스트를 검색할 필요가 없게 한다.

#### 4. 라우터의 데이터 구조

라우터의 주요한 동작은 라우터 내부에서 각 계층 간에 공유된 데이터 저장 공간을 읽고 쓰는 것이기 때문에 그로 인해 발생하는 복사 시간에 따라 라우터의 성능이 좌우된다. 디바이스 제어 네트워크와 데이터 네트워크로의 변환 과정에서 발생한 여러 계층 간의 빈번한 데이터 복사는 자칫 라우터의 성능을 저하시킬 수 있기 때문에 실시간으로 제어되는 네트워크 기반 제어 시스템에서 데이터 구조에 대한 중요성은 더욱 강조된다. 데이터 저장 방식은 큐(queue)형태의 동적 데이터

구조와 배열 형태의 정적 데이터 구조 방식이 있다.

#### 4.1. 큐(queue)

큐는 First-Input/First-Output방식의 데이터 구조로 기존의 라우터에서 취하는 데이터 구조이며 일종의 데이터의 버퍼 역할을 할 수 있다. 데이터가 추가 될 때마다 새로 메모리를 할당하여 큐의 리스트에 추가시키므로 메모리 용량이 정적인 배열 구조의 데이터 형식보다 유연성이 있지만 데이터를 큐에 저장하거나 꺼낼 때마다 데이터를 복사해야 한다. 즉, 네트워크로부터 실시간으로 읽혀지는 데이터를 라우터가 허용하는 메모리 공간까지 계속 저장할 수 있지만 그로 인해 지연시간이 발생할 수 있다. 그림 3.14의 블록도를 보면 라우터 내부의 계층 간에 공유되는 큐 형태의 데이터 구조를 알 수 있다.

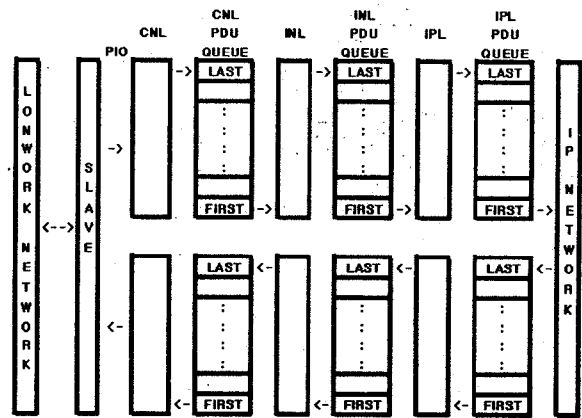


그림 7 큐를 이용한 데이터의 이동

그림 7에서처럼 슬레이브 프로세서와 마스터 프로세서간의 큐, CN 계층과 IL 계층 간의 큐, IL 계층과 IP 계층 간의 큐, IP 계층과 IP 네트워크간의 큐 등 3 종류의 큐가 필요하며 데이터를 받는 계층 간의 이동과 데이터를 보내는 계층 간의 이동에서의 큐는 따로 관리 되어야 하므로 두 배의 큐, 즉 총 6 개의 큐가 필요하게 된다. 쓰레드가 큐에 접근하면 크리티컬 섹션에 의해 다른 쓰레드의 프로세스는 중단되기 때문에 큐에 의한 빈번한 데이터 복사는 서로 다른 쓰레드의 프로세스에 영향을 주어 전체적인 프로세스의 성능을 저하시키게 되고 그것은 다른 계층 간의 데이터 이동 속도를 떨어뜨리게 되는 등 성능 저하의 악순환이 반복된다. 이런 식으로 데이터가 큐에 계속 누적되어 라우터로 인해 발생하는 지연시간이 점점 길어진다면 실시간 네트워크 기반 제어 시스템의 의미가 퇴색될 수 있다. 그래서 큐를 다루는 프로그램 최적화에 세심한 주의가 필요하다.

#### 4.2 배열 구조

큐 형태의 데이터 구조의 개선안으로 배열 형태의 데이터 구조를 제안한다. 여기서 말하는 배열 구조란 큐와는 동일한 기능을 하면서 메모리 특성이 정적인 배열 구조와는 좀 다르다. 그것은 슬레이브의 뉴런 프로세서 네트워크 인터페이스의 특성에 따른 배열 구조이다. 슬레이브의 네트워크 인터페이스는 아날로그 입력 12개와 출력 12개, 디지털 입력과 16개와 출력 16개의 네트워크 변수로 이루어져 있기 때문에 그것을 기반으로 큐와는 다른 데이터 구조를 만들어 낼 수 있다. 그 데이터 구조의 기본은 마스터 프로세서에서 각각의 네트워크 인터페이스만큼 CNL PDU를 저장할 수 있는 배열을 할당하는 것이다. 그리고 각 배열의 요소마다 갱신되었음을 표시하

는 플래그 형태의 변수도 할당하여 병렬 I/O 를 통해 들어온 네트워크 변수의 정보를 갱신하고 갱신되었다는 것을 플래그 변수에 표시하고 데이터를 사용했을 경우 갱신되었다는 표시를 삭제한다. 이것의 단점은 갱신된 데이터를 라우터의 상위 계층에서 읽어 들이지 못한 상태에서 새로 갱신된 데이터는 이전 데이터 위에 덮어쓰이기 때문에 데이터의 손실이 발생한다는 것이다. 그 구조는 그림 8과 같다.

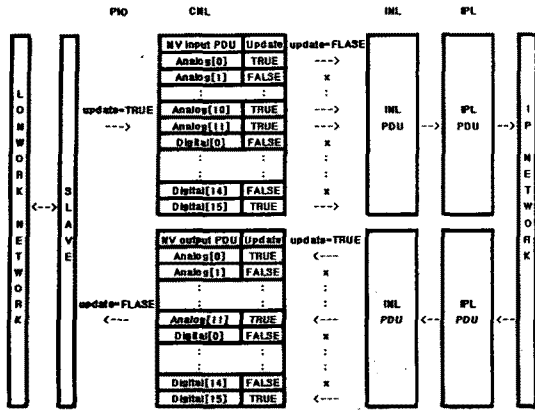


그림 8 배열 구조의 데이터 이동

### 5. 결론

본 논문에서는 가상 디바이스 네트워크의 필요성을 설명하고 가상 디바이스 네트워크 기반 제어 시스템을 구현하기 위하여 임베디드 시스템을 이용한 라우터의 개념과 알고리즘을 제안하였다. 또한 실제로 LonWorks/IP 라우터를 구현하였다. LonWorks/IP 라우터와 IP 네트워크를 통해 원격의 제어 네트워크들을 하나의 네트워크로 통합하는 목적을 달성하기 위해서는 차후에 실험실이 아닌 실제 현장에서 큰 규모의 가상 디바이스 네트워크를 구성하여 네트워크 기반 제어에 관한 실험이 필요하다.

### 참고 문헌

[1] G. H. Choi, G. S. Choi, and J. S. Kim, "Lonworks Based Virtual Device Network(VDN) for Predictive Maintenance", ICMIT'01, Yamaguchi Seminar Park, Japan, pp. 372-376, 2001.

[2] Feng-Li Lian, James Moyne, and Dawn Tilbury, "Network Design Consideration for Distributed Control Systems", IEEE, pp 297-307, March 2002.

[3] M. S. Branicky, S. M. Phillips, and Wei Zhang, "Stability of Networked Control Systems: Explicit Analysis of Delay", Proc. ACC, Chicago, Illinois, pp. 2352-2357, June, 2000.

[4] Wei Zhang, M. S. Branicky, and S. M. Phillips, "Stability of Networked Control Systems", IEEE Control Systems Magazine, pp. 84-99, February 2001.

[5] Saverio, "Smith's Predictor for Congestion Control in High-Speed Data Networks", IEEE, pp 358-364, February 2000.

[6] Weidong Zhang and Viaomong Xu, "Simple Predictor for Processes with Time Delay", Proc.AACC, San Diego, California, pp. 822-826, June, 1999.

[7] H. Shahnasser and Q. Wang, "Controlling Industrial Devices over TCP/IP by Using LonWorks," Proc. IEEE, pp. 1309-1312, 1998.

[8] O. Vahamaki, A. Allen and J. Gaff, "High Speed Peer-to-Peer Communication System for Integrated Protection and Control in Distribution Network", Development in Power System Protection, IEE, pp.243-246, 1997.

[9] H.Reiter, C. Kral, "Interaction Between JAVA and LonWorks," Proc. IEEE, pp. 335-339, 1997.

[10] Echelon, Engineering Bulletin, 1999.

[11] Peter Palensky, D. Dietrich and H. Reiter, "Demand Side management in Private Homes by Using LonWorks", Proc. IEEE, pp.341-347, 1997.

[12] Keiji Watanabe and Masami Ito, "A Process ? Model Control for Linear Systems with Delay ", IEEE Trans. Automatic Control., Vol. ac-26, no.6, pp. 1261-1269, dec. 1981.

[13] Saverio Mascolo "Smith's predictor for congestion control in TCP internet protocol", AACC 1999, pp. 4441-4445.

[14] C. C. Hang, B. W. Chong, "On Methods of Treating dc Levels in an Adaptive Digital Smith predictor", IEEE Trans. on Automatic control, Vol.35, No.1, pp. 65-66,1990.