

IPv6 상에서의 실시간 처리용 DCP 프로토콜

이영준^o 조인휘^o
한양대학교 정보통신대학원
efreeti^o@ihanyang.ac.kr, iwjoe@hanyang.ac.kr

Device Configuration Protocol over IPv6 for Real-time Processing

Youngjun Lee^o Inwhhee Joe^o
The Graduate School of Information and Communications, Hanyang University

요 약

DCP (Device Configuration Protocol)는 인터넷을 기반으로 하여, TCP/IP 표준 프로토콜상에서 돌아가는 응용프로토콜로서, 특히 응용계층에서 Client와 Server의 역할을 교환하여, 다양한 필드장치들의 인터페이스를 단일화하기 위해 제안한 프로토콜이다. 이 프로토콜에 실시간 처리 알고리즘과 IPv6에서 제공되는 QoS의 기능을 연동하여 적용하게 된다면, 긴급한 실시간 처리가 요구되는 데이터를 상황에 맞게 처리 할 수 있다. 만약, 이러한 실시간 처리가 보장되는 DCP 인터페이스 방식을 현장에 적용하게 된다면, Device의 소형화에 의한 비용절감 효과, 인터페이스 단일화에 의한 효율적 장치 제어 및 관리, 그리고 실시간 처리로 인한 서비스의 질적 향상이 기대된다.

1. 서 론

IT(Information Technology)산업의 발전이 우리 생활에 가져온 변화는 일일이 헤아릴 수 없을 정도로 많으며, 산업 전반에 걸쳐 사용되고 있는 필드 장치들과 응용 소프트웨어에 있어서도 예외는 아니다.

최근의 추세는 이러한 필드 장치들을 원격 접속하여 제어 할 수 있는 인터넷 기반의 인터페이스 연구가 활발히 진행되고 있으나, 표준화된 인터페이스가 없으므로 개발 업체들마다 다양한 방식의 독립적인 인터페이스가 사용되어지고 있고, 기본적으로 디바이스와 서버가 결합되어 있는 방식으로, 이를 사용자(관리자)가 클라이언트 프로그램으로 네트워크를 통해 접근하여 제어하는 방식이 대부분이다.

이러한 방식의 가장 큰 단점은 디바이스가 서버와 결합되어 있기 때문에 디바이스의 크기 증가로 인한 비용 문제와 현재의 추세인 기기의 소형화 및 경량화에 있어서도 역하며, nPNP (Networking Plug and Play)의 지원 불가, 서버 확장 어려움 등이 있다. 이런 문제들을 해결하고자, 인터넷을 기반으로 하여, TCP/IP 표준 프로토콜상에서 돌아가는 응용 프로토콜로서, 특히 응용계층에서 Client와 Server의 역할을 교환하여, 다양한 필드장치들의 인터페이스를 단일화하기 위해 제안된 것이 DCP (Device Configuration Protocol) 프로토콜이다. 그러나 DCP 프로토콜은 기본적으로 단일 서비스 클라이언트/서버에 맞추어 제안되었기에 보다 효율적인 처리가 가능한 다중처리의 통합서버 환경에 적용하기에는 개선이 요구된다.

그러므로 본 논문에서는 기존의 DCP 프로토콜을 산업 제어 시스템(PLC, DCS) 및 응용 소프트웨어 전반에 걸쳐 통합적 서버의 기능을 수행하기 위해, Real-time Processing의 적용과 앞으로 차세대 인터넷 환경의 대안으로 제시된

IPv6의 QoS 기능과의 연동으로 다양한 실시간적 다중처리가 가능한 보다 확장 개선된 DCP 프로토콜을 제안한다.

2절에서는 DCP 프로토콜의 인터페이스 개요 및 DCP 프로토콜의 패킷 구성을, 3절에서는 Real-time 처리기법과 IPv6의 QoS 기능을 연동한 확장 개선된 DCP 프로토콜을, 마지막으로 4절에서는 시뮬레이션한 결과를 보여준다.

2. Device Configuration Protocol

2.1 DCP 프로토콜 개요

Device Configuration Protocol (DCP)은 TCP/IP를 기반으로 등록된 디바이스를 인식하고, 디바이스에 인터넷 상의 존재하는 서버를 알려줌으로써 연결시키는 중계자 역할을 하는 프로토콜이다.

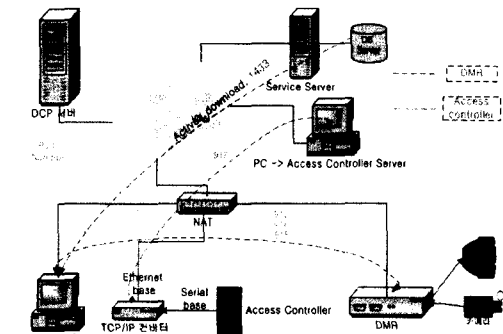


그림 1. DCP Service

그림 1은 다양한 종류의 Device들 중에서 테스트 환경을 구축하여 직접 서비스를 해 본 DMR Service와

Access Controller Service의 전체적인 구성도이다. 기존의 필드 기기 인터페이스와는 반대로 Device가 Client가 되고, 관리자의 PC가 Server가 되는 방식으로 Device와 Server가 분리되어 있다. 그러므로 DCP Server가 필요하며, DCP Server에 등록된 Device는 먼저 DCP Server로 접속후, 다시 DCP Server가 알려준 주소의 Service Server로 접속하게되어 Service Server와 Device(Client) 간의 커넥션이 확정되어 Service가 가능해 진다.

이러한 DCP Interface를 사용하여 서비스를 구현하게 될 경우 Device와 Server의 분리로 인하여 서비스의 소형화 및 경량화, 서버이전 및 확장용의 용이성과 비용 절감의 효과, nPNP(Networking Plug and Play)의 지원, 동시에 여러 Device들이 서버에 접속되어 통합적 관리가 가능함으로써 인터페이스의 단일화로 보다 편하게 종류별 Device 제어가 가능하다는 장점이 있다.[1]

2.2 DCP 프로토콜의 패킷 구성

DCP packet은 TCP/IP 프로토콜을 기반으로 확장된 형태로 구성되어있다.

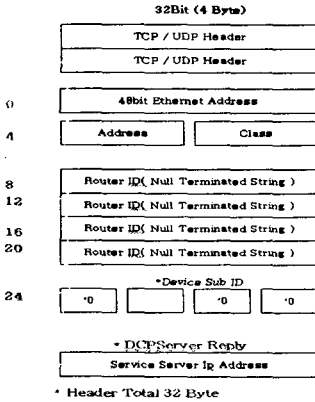


그림 2는 DCP packet의 구성을 보여주고 있다. 전체 packet의 크기는 32 Byte이다. 헤더는 TCP/UDP 헤더를 수정없이 사용하며, 0 Byte에서 7 Byte는 인증을 위한 Device의 인터넷 주소와 각 Device의 다양한 서비스의 지원을 위한 종류별 구분 코드에 해당하는 Class 코드(5 Bit)가 포함된다.

8 Byte에서 24 Byte는

그림 2. DCP packet

예전에 사용되어지던 장치들의 라우터 표시 관련 기능의 호환성을 지원하기 위해 포함되어 있으나, 현재 Device들의 방식에서는 사용되지는 않는다.

마지막으로 DCP Server가 새로운 Device로부터 서비스 요청을 받았을때 인증후 Service Server의 주소를 알려주기 위한 DCP Server Reply에 사용되는 "Service Server Ip Adress" 필드가 포함되어 있다.[1]

3. 실시간 처리용 DCP 프로토콜

3.1 Real-time Processing 구현

기존의 DCP 프로토콜은 기본적으로 단일 서비스 클라이언트/서버에 맞추어 제안되었기에 보다 효율적인 실시간 처리가 가능한 멀티 클라이언트들의 다중처리를 위한 통합서버 환경에 적용하기에는 개선이 요구된다.

그림 3은 실시간적 다중처리가 가능한 개선된 DCP에 적용된 Real-time Processing Algorithm이다.

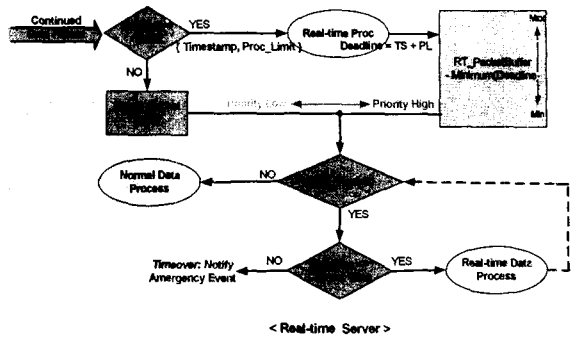


그림 3. Real-time Processing Algorithm

실시간 처리가 요구되는 데이터는 패킷의 전송시점의 time stamp와 처리 우선순위에 따른 Proc_limit 값을 포함하여 전송하게 됨으로서 서버에서는 처리 우선순위에 따라 RT_PacketBuffer에 저장되어 최대한 요구시간인 Deadline을 넘어서지 않도록 보장된다. 이는 산업제어 기기나 응용 소프트웨어에서 사용시 긴급 상황이나 실시간 처리가 요구되는 데이터의 경우 효율적인 처리로 인한 서비스의 질을 향상 시킬 수 있으며, 다중 클라이언트의 처리시 우선순위에 따른 처리가 가능하므로 효과적인 통합서버의 구성이 가능하다.

그림 4은 기존의 DCP Packet Format에 Real-time Processing Algorithm이 포함된 보다 진보된 DCP Packet Format이다.

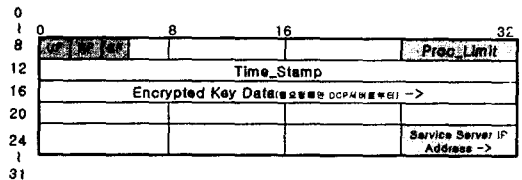


그림 4. 개선된 DCP Packet Format

기존의 DCP 환경과의 호환성을 고려하여, 현재 사용되어지지 않는 잉여 필드인 8 byte에서 24 byte를 활용하였다. UF는 확장필드의 사용여부, RF는 리얼타임패킷의 사용여부, SF는 암호화 패킷 사용여부, Proc_Limit는 한계 처리요구시간, Time_Stamp는 전송 시간, Encrypted Key는 암호화된 Key데이터를 의미한다. 위에서 보았듯이 개선된 DCP Packet Format은 실시간 처리기능 외에도 보안기능 또한 포함되어 있다.

3.2 IPv6 QoS 기능과의 연동

현재의 인터넷은 IPv4를 기반으로 하고 있으나 폭발적인 수요 증가로 인해 주소 공간 부족과 다양한 멀티미디어 응용에 따른 서비스 품질 개선 요구 등과 같은 문제에 직면해 있다. 주소 공간 문제는 현재 인터넷의 IPv4를 자연스럽게 수용하면서 IPv6으로 진화해나감으로써 해결할 수 있을 것이며, 인터넷에서 QoS 보장은 In:serv WG과 Diffserv WG에서 정의하고 있는 종합 서비스모델

(Flowlabel)과 차등화 서비스모델(Traffic class)을 수용함으로써 가능할 것이다. Intserv의 종합서비스모델은 RSVP를 이용하여 기존의 최선형서비스 이외에 보장형서비스와 부하제어형서비스를 제공하는 IPv6의 Flowlabel field를 활용하는 것이다.

3.2.1 RSVP (Resource Reservation Protocol)

RSVP 프로토콜은 네트워크의 종단간의 경로를 형성함으로써 네트워크 자원을 예약한다. 전송측에서는 필요한 대역폭을 명시한 PATH 메시지를 전송하여 QoS의 레벨을 요청하고, 수신측에서는 전송측에 RESV 메시지를 전송하여 경로에 대한 자원예약을 알린다. RESV 메시지를 통하여 수신측과 전송측 경로의 중간에 있는(RSVP가 가능한)장치들은 네트워크 자원을 사용할 권한과 대역폭을 결정하게 되고, 자신의 네트워크 정책과 사용 가능한 대역폭을 만족한다면 RESV 메시지를 전송측에 전달하게 된다. 전송측이 RESV 메시지를 수신하게 되면 자원예약이 완료되어 경로에 따라 QoS 데이터가 전송될 수 있다.

3.2.2 IPv6상에서의 DCP 프로토콜

IPv6의 Flowlabel을 통해 RSVP를 사용하여 원하는 자원을 예약하여 적절하게 사용하게 된다면, 실시간 처리가 요구되는 데이터를 훨씬 신속하게 처리할 수 있으므로, Flowlabel에서 제공되는 자원예약기능과 실시간 처리용 DCP 프로토콜의 연동으로 인한 상호보완은 서비스의 질적 향상의 측면에서 큰 의미가 있다.

4. Simulation

본 논문의 Real-time Processing과 IPv6의 Flowlabel의 측정에서 사용되는 RSVP의 시뮬레이터는 ns2를 기초로 하여 Java 언어를 사용하여 제작하였다.

기본적인 구성은 Send Node, Receive Node, 자원 예약 기능을 가진 중간 Router1, 2 및 다른 Other node들로 구성된다.

표 1. Simulation Parameters Value

Simulation Parameter	Parameter Value
Packet Type	TCP
Packet Size	1 Kbyte
Bandwidth	1.8 Mb, 2 Mb
Transfer Rate	1 Mbps
Delay	1 ms
Offered Load	100%(full), 120%(excessive)

Simulation Parameters는 표 1과 같으며, 대역폭은 중간 Router1,2의 링크에 따라 각각 다르게 설정되었고, 일반 패킷과의 비교를 위하여 대역폭을 초과한 과부하 트래픽을 인가하였다.

IPv6의 Flowlabel(RSVP) 기능에 의한 자원예약 전송과 전송후 Real-time Processing에 따른 처리시간을 패킷수와 초과된 트래픽 비율에 따라 일반적인 패킷의 전송과 비교하여 측정하였다.

그림 5는 전송된 패킷량에 따른 처리시간을 측정하여

나타낸 시뮬레이션 결과이다.

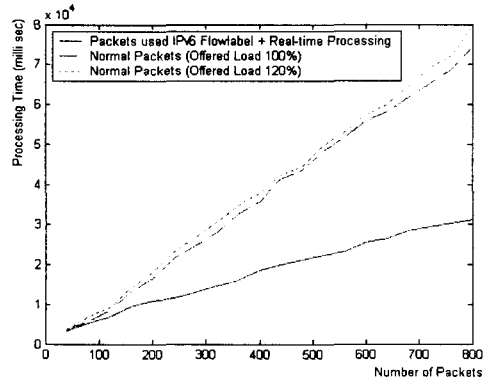


그림 5. 전송된 패킷수에 따른 처리시간

위 결과와 같이 IPv6의 Flowlabel 기능과 Real-time Processing 기법을 동시에 적용하여 사용함으로써 트래픽의 집중도와 상관없이 항상 일반적인 패킷들보다 월등히 신속한 처리가 가능함을 알 수 있다.

5. 결론

본 논문에서는 TCP/IP 표준 프로토콜 상에서 돌아가는 응용 프로토콜로서, 특히 응용계층에서 기존 장치들의 Client와 Server의 역할을 교환하여, 공통된 원격 Server의 통합적 제어에 의한 많은 장점을 지닌 새로운 방식의 단일화된 장치 인터페이스인 DCP를 확장 개선하여 Real-time Processing의 적용과 IPv6 QoS의 연동으로 다양한 실시간 다중처리가 가능한 통합적 서버기능을 수행할 수 있도록 함으로써, 한층 진보된 DCP 인터페이스 방식을 제안하였다.

이러한 실시간 처리가 보장되는 DCP 인터페이스 방식을 현장에 적용하게 된다면, Device의 소형화에 의한 비용절감 효과와 효율적 장치 제어 및 관리 그리고 실시간 처리로 인한 서비스의 질적 향상이 기대된다.

[참고 문헌]

- [1] 이영준, 조인휘, "필드장치 인터페이스 단일화를 위한 인터넷 기반의 DCP 프로토콜 연구," 제30회 추계학술발표회, 한국 정보과학회 2003
- [2] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", IETF RFC-1889, January 1996
- [3] S. Deering, R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", December 1998
- [4] Lloyd, B., Susnik, M., "Web embedded field devices", Conference Record of the 2002 Annual Pulp and Paper Industry., pp. 199-202, June 2002