

내장형 Network Agent를 위한 새로운 TCP state의 제안

정진규, 안광혁, 진선일, 홍석교, *유영동
아주대학교, *테크이즈
전화 031-219-2489 / 핸드폰 017-513-3566

The new TCP state for the embedded network agent

Jin-Kyu Jung, Gwang-Hyuk Ahn, Sun-Il Jin, Suk-Kyo Hong
Dept. of electronics, Ajou University
E-mail : fire0715@madang.ajou.ac.kr
*Young-Dong Yoo
*Techiz (<http://www.techiz.com>)
*E-mail : vdyoo@dreamwiz.com

Abstract

This paper describes an experiment on the embedded network agent. The embedded network agent is the device that will enable other devices to communicate over the Internet. Many devices have no way to communicate with other devices over the Internet. The embedded network agent extends the stand-alone devices that have no TCP/IP protocol suite to the Internet, the global networks, without any hardware and software. To support the better efficiency, this paper proffers new TCP state diagram of the embedded network agent. This TCP state on the embedded network agent considers the status of the common devices behind the embedded network agent. Because of this consideration, the embedded network agent tenders an improved result to user.

I. 서론

최근에 들어, 독립 장비들을 인터넷에 연결하고자 하는 연구에 많은 관심이 고조되고 있다.[4] 인터넷에 연결되기 위해서는 TCP/IP protocol suite를 갖추고 있어야 하는데, 이전에 제작되어진 대부분의 독립 장비들은 TCP/IP protocol suite와 같은 복잡한 통신 규격보다는 RS-232와 같은 단순한 Serial protocol만을 갖추고 있는 경우가 많다. 이러한 독립 장비들을 인터넷에 연결하기 위해서는 독립 장비들을 TCP/IP protocol suite를 갖춘 네트워크 시스템에 연결하여 도움을 받아야만 한다.

원거리에 위치한 Client system에서 네트워크망에 연결되어져 있는 독립 장비의 상태를 예측하는 것은 불가능하다. 원거리의 네트워크망의 상태는 TCP/IP protocol suite를 통하여 동작을 확인할 수 있지만, 네트워크 시스템에 가려져 있는 독립 장비의 상태는 데이터 전송에 관한 응답이 오거나 Timeout을 통한 에러 처리 과정을 통하여 확인할 수밖에 없다. 이러한 근본적인 이유는 TCP/IP protocol suite가 독립 장비의 상태를 고려하지 않기 때문이다.

본 논문에서는 단순한 Serial protocol만을 갖추고 있는 독립 장비들을 인터넷에 연결시켜 줄 수 있는 Embedded System에 적합한 새로운 TCP state를 제안하고자 한다. 새로운

TCP state는 Server와 연결되어진 독립장비를 인식하고 이를 TCP/IP protocol suite에 반영함으로써 기존 TCP/IP protocol suite를 사용하는 프로토콜보다 네트워크의 사용 효율이 높은 것을 보여줄 것이다. 실험을 위하여 PLC를 독립 장비로 사용하였으며, Server는 PLC와 통신이 가능하며 Ethernet을 통하여 인터넷과 연결할 수 있는 Embedded system으로 구성하였다. 이를 Embedded network agent라 부른다.

II. 본론

1. Embedded network agent의 하드웨어 구조

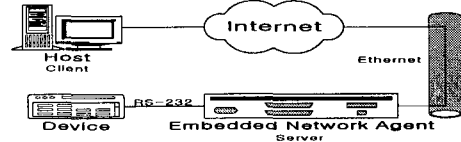


그림 3. 내장형 네트워크 대행자의 토폴로지

Fig. 3. The topology of embedded network agent

Embedded network agent의 Network Topology는 그림 1과 같이 구성된다. 본 논문에서 Embedded network agent는 RS-232 통신을 가지고 독립 장비인 PLC와 통신을 하며, 자체의 Ethernet device를 통하여 인터넷에 연결되어진다.

1) PLC

Embedded network agent에서 사용되어진 TCP/IP protocol suite는 독립 장비의 인식을 기초로 하여 이루어진다. 이러한 인식의 과정은 Embedded network agent와 독립 장비 사이의 통신에 근간 한다. 그러므로 독립 장비의 통신 규격을 이해하는 것이 Embedded network agent에게 필수적으로 요구되어지는 사항이다.

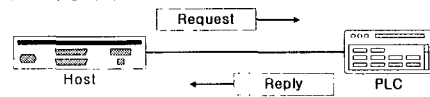


그림 4. PLC의 통신규격

Fig. 4. PLC protocol

실험을 위하여 사용한 PLC의 통신 규격은 그림 2와 같다.

2) The Embedded network agent

본 논문에서 사용한 Embedded network agent는 표 1과 같이 구성되어 있다.

표 5. 내장형 네트워크 대행자의 하드웨어 구성

Table 1. Hardware specification of embedded network agent

항목	사양
CPU	AMD Am188™ER-40KC
RAM	128 Kbytes
ROM	64 Kbytes
Network Interface	10 Base-T Ethernet Port
Serial Device	3 * RS-232 Ports

2. TCP/IP protocol suite

Embedded network agent 상에는 NIC(Network Interface Card)를 지원하기 위한 Device driver와 ARP(Address Resolution Protocol), IP(Internet Protocol), ICMP(Internet Control Message Protocol), TCP(Transmission Control Protocol) 등이 구현 되어졌다. 이러한 Protocol들은 Embedded 환경에 적합하도록 코드가 최적화 되어졌으며, 최소한의 기능을 수행하도록 작성 되어졌다. 예를 들어, ICMP의 경우, Echo request와 Echo reply 기능만을 지원하도록 작성되어 졌다.

3. TCP state diagram

현재 대부분의 Server가 사용하는 TCP/IP protocol suite는 Service를 제공하는 Server와 Service를 요청해 오는 Client사이만의 정보 전송을 목적으로 개발되었기 때문에 Server와 통신을 주고받는 독립 장비에 대한 고려를 하지 않는다.[1][3] 이러한 일반적인 TCP/IP protocol suite를 사용하는 Server를 통하여 독립 장비를 인터넷에 연결하고자 할 때, Server와 독립 장비 사이의 통신 상태에 따라서 Client로부터 재전송 요구가 발생할 수 있다. 이러한 재전송 요구는 전체 네트워크의 효율을 떨어뜨릴 뿐만 아니라, Service를 요청한 Client의 사용자 입장에서 RTT(Round Trip time)이 길어지는 결과를 가져오게 된다.

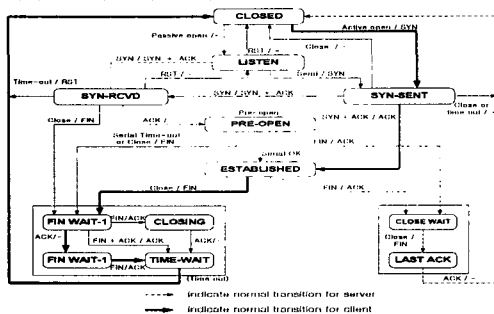


그림 5. 제안된 내장형 네트워크 대행자의 TCP 상태 천이도

Fig. 5. Proposed TCP state transitions diagram of embedded network agent

그림 3은 Embedded network agent에서 사용되는 TCP/IP protocol suite의 새로운 TCP state diagram이다.

Embedded network agent에서는 Service를 제공하는 Embedded network agent와 독립 장비 사이의 통신 상태를 고려하기 위하여 Pre-open state라 부르는 새로운 state를 제안한다.

4. The Pre-open state.

그림 2에서와 같은 방식으로 Embedded network agent와 독립장비가 통신을 하는 경우, 양자간의 통신은 TCP protocol과 Server application program 사이의 통신과 같은 방식으로 이루어진다. 일반적으로 TCP protocol과 Server application사이의 통신은 하나의 Computer system안에서 이루어지므로 두 계층간의 통신에서 문제의 발생을 고려하지 않는 반면에 Embedded network agent와 독립 장비 사이의 통신에서는 양자간의 통신 상태를 고려하여야 한다.

Client와 독립 장비 사이의 통신은 TCP connection이 이루어지기 전과 후라는 조건과 양자간의 통신에서의 문제 발생 여부를 하는 조건이 존재한다. 따라서 전체적으로 고려해야 할 상황은 4 가지 경우이다. 전형적인 TCP/IP protocol suite에서는 모든 경우에 Timeout mechanism을 사용하여 이 문제를 처리한다. 이러한 Timeout mechanism은 수 차례의 재전송 요구 후에 처리되어지므로 재전송으로 인한 네트워크의 효율 저하와 RTT(Round Trip time)의 증가를 초래한다. [2]

TCP connection이 이루어진 후에는 TCP protocol 자체에서 Timeout mechanism을 사용하므로 두 계층간의 통신에서 문제가 발생하더라도 이를 해결할 수 있다. 그러나 TCP connection이 이루어지기 전에 원격리에 위치한 Client가 독립 장비 사이의 통신 문제를 인식하는 것은 쉽지 않은 문제이다. 독립 장비가 하나의 시스템으로 국한 되어질 경우에는 통신 상태를 확인하기 위하여 Dummy data를 전송하고 이에 관한 응답을 확인하는 방법을 사용할 수 있지만, 이기종의 독립 장비가 연결되어질 경우에는 Dummy data를 정의하기가 어렵다.

본 논문에서 제안한 TCP state에서는 독립 장비와의 통신 규격과는 상관없이 가능한 빠르게 Embedded network agent가 통신의 문제점을 인식하기 위하여 Pre-open이라는 state를 정의하였다. Pre-open state는 Client connection 요청을 수락하여 Client에게는 TCP connection이 이루어진 것처럼 보이지만 독립장비와 Embedded Network agent와 독립장비 사이의 통신 상태는 확인이 이루어지지 않은 상태를 의미한다. Pre-open state에서는 Event 발생에 따라 다음의 3 가지 다른 state로 전환되어진다.

- Established state
- Fin wait-1 state
- Close wait state

Close wait state로의 전환은 Client로부터 Fin flag를 통한 연결 종료 요청 Event와 TCP protocol의 Timeout Event에 의하여 발생하므로 Embedded Network agent와 독립장비사이의 통신 상태와는 무관하게 진행되어진다 그림 4는 Established state로의 전환 과정으로 Embedded network agent와 독립장비 사이의 통신 상에 아무런 문제도 발생하지 않은 경우로서 Embedded network agent는 정상적인 Service를 Client에게 제공해 줄 수 있으므로 TCP state를 Established state로 전환한다.

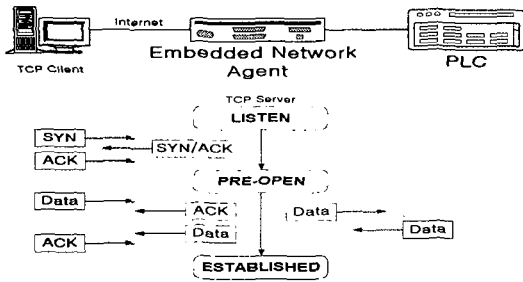


그림 6. Established 상태로의 전환

Fig. 6. State transition to the established

그림 5는 Fin Wait-1 state로의 전환 과정으로 Embedded network agent와 독립장비 사이의 통신에 문제가 발생한 경우이다. 이러한 경우 TCP protocol의 Error recovery 기능에 의하여 Client로부터 재전송 요구가 발생한다. Embedded network agent는 Client로부터 전송된 Data를 독립장비로 송신한 후 Client측의 Timeout event에 의한 Passive close operation을 기다리지 않고 독립장비로부터의 응답시간을 이용하여 Active close operation을 수행한다. 독립장비로부터의 응답시간은 Client의 Timeout event의 발생시간보다 상대적으로 훨씬 적은 시간이 소모된다. 뿐만 아니라 Client로부터의 재전송 과정이 사라지므로 네트워크의 효율이 높아지게 된다.

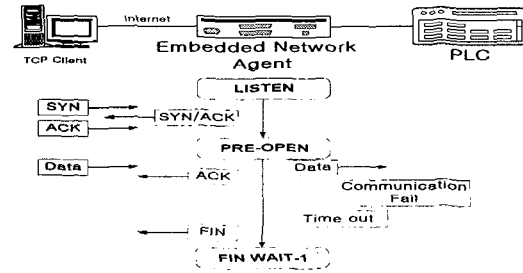


그림 7. Fin Wait-1 상태로의 전환

Fig. 7. State transition to the Fin Wait-1

독립장비의 응답시간은 Embedded network agent와 독립 장비 사이의 Data의 송수신시간과 독립 장비 내부에서 Data processing에 걸리는 시간의 합보다 크고, Client의 재전송 요구시간보다는 작아야 한다.

그림 2에서처럼 명령을 내리는 Client와 명령을 수행하는 PLC사이에서 데이터 전송 에러가 발생할 경우, 데이터 재전송을 통하여 에러를 처리한다. TCP/IP protocol suite를 사용하는 원격지 제어 방식에서는 Application layer에서 이와 같은 재전송을 지원해 주어야 한다. 즉, TCP layer의 재전송은 TCP protocol 내부에서 자체적으로 이루어지며, 실제 작성되어진 응용 프로그램에서도 이와는 별도의 재전송 방식을 사용 하여야 한다.

Service를 요청하는 Client와 Embedded network agent 사이의 Data 전송에 소모되는 시간을 T_i 라 하고, Embedded network agent와 독립 장비 사이의 Data 전송에 소모되는 시간을 T_s 라 하자. 전송이 성공한 경우에 소모되는 시간을 $T_{success}$, 실패한 경우를 T_{fail} 이라 할 때, Fail은 Timeout으

로 발견되어지며, $T_{fail} > T_{success}$ 인 특성을 갖는다.

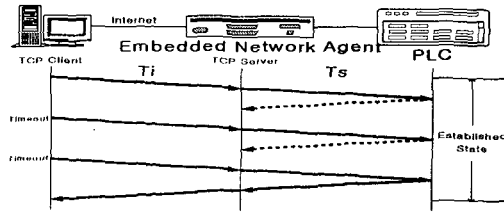


그림 8. 일반적인 TCP protocol 상에서의 재전송 방식

Fig. 8. Retransmission mechanism on the general TCP protocol

그림 6은 기존의 TCP protocol을 사용하는 Embedded network agent에서 Data 전송이 실패한 후 재전송 방법을 사용하여 2번째 재전송에서 성공한 경우를 보여주는 그림이다. 그림 7에서의 실선은 전송의 성공을 점선은 전송의 실패를 의미한다. 기존의 TCP protocol에서의 재전송을 통한 에러 복구 모델에서의 RTT(Round Trip Time)를 수식으로 표현하면 식 1과 같다.

$$T_{RTT} = (T_{i_{success}} + T_{s_{success}} + T_{s_{fail}} + T_{i_{fail}}) \times i + 2(T_{i_{success}} + T_{s_{success}}) \quad (1)$$

여기서,

- $T_{i_{success}}$: Client-Agent 간의 전송성공시간
- $T_{i_{fail}}$: Client-Agent 간의 전송실패시간
- $T_{s_{success}}$: Agent-PLC 간의 전송성공시간
- $T_{s_{fail}}$: Agent-PLC 간의 전송실패시간
- i : 재전송 회수

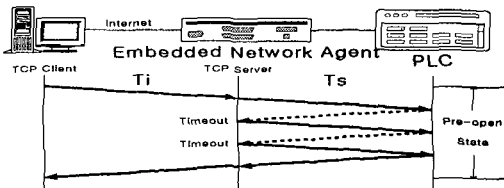


그림 9. 제안된 TCP protocol 상에서의 재전송 방식

Fig. 9. Retransmission mechanism on the proposed TCP protocol

그림 7은 그림 6과 같은 경우를 본 논문에서 제안된 TCP protocol로 구현했을 경우의 모델이다. 제안된 TCP protocol에서의 RTT를 수식으로 표현하면 식 2와 같다.

$$T_{RTT} = (T_{s_{success}} + T_{s_{fail}}) \times i + 2(T_{i_{success}} + T_{s_{success}}) \quad (2)$$

여기서,

- $T_{i_{success}}$: Client-Agent 간의 전송성공시간
- $T_{i_{fail}}$: Client-Agent 간의 전송실패시간
- $T_{s_{success}}$: Agent-PLC 간의 전송성공시간
- $T_{s_{fail}}$: Agent-PLC 간의 전송실패시간
- i : 재전송 회수

III. 모의 실험

모의 실험을 위하여 다음과 같은 조건을 가정하였다.

- PLC와 Client사이의 통신 방식은 그림 2와 같이 Request와 Reply로 구성 되며, Request와 Reply Packet의 길이가 같다.
- PLC의 응답시간은 Embedded network agent와 PLC사이의 Data의 송수신 시간과 PLC 내부에서 Data processing에 걸리는 시간의 합보다 크고, Client의 재전송 요구시간보다는 작아야 한다.
- $T_{i_{success}} = 10 \text{ sec}$, $T_{i_{fail}} = 20 \text{ sec}$,
 $T_{s_{success}} = 2 \text{ sec}$, $T_{s_{fail}} = 3 \text{ sec}$ 이라고 가정한다.

그림 8은 2가지 TCP/IP protocol model에 따른 RTT를 표시한 것이다. 그림에서 알 수 있듯이, 재전송 횟수가 증가함에 따라 기존의 TCP protocol을 사용할 경우 RTT가 급격히 증가하지만 본 연구에서 제안된 TCP protocol에서는 RTT의 증가는 아주 작다는 것을 알 수 있다.

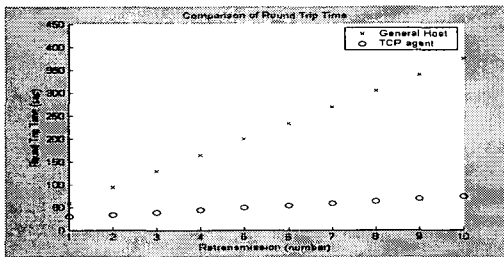


그림 10. Round Trip Time의 비교
 Fig. 10. The comparison of the round trip time

IV. 실험

그림 9는 실험을 하기 위한 장비의 구성을 보여준다. 실험에서 사용한 독립 장비인 PLC는 TCP/IP protocol suite에서 Server로서 동작하는 Embedded network agent를 통하여 인터넷에 연결되어졌으며, Microsoft Window를 OS로 사용하는 Client가 다른 Router를 통하여 인터넷에 연결 되어졌다.

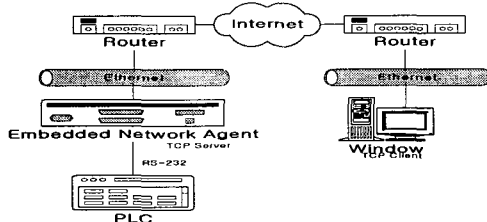


그림 11. 실험 장비의 구성도
 Fig. 11. The block diagram of the experimental set

그림 10은 실험에 사용된 Embedded network agent와 PLC를 촬영한 사진이다. PLC가 RS-232 Serial Port를 통하여 Embedded network agent와 연결되어져 있으며, PLC의 동작은 LED를 통하여 쉽게 확인할 수 있다. 본 논문에서 제시한 TCP/IP protocol suite를 사용하는

Embedded network agent에서는 전송 Error 발생시에 기존의 TCP protocol을 사용하는 방식보다 RTT이 현격하게 감소함을 확인할 수 있었으며, 실제 인터넷상으로 재전송되어지는 Packet의 감소도 확인할 수 있었다.

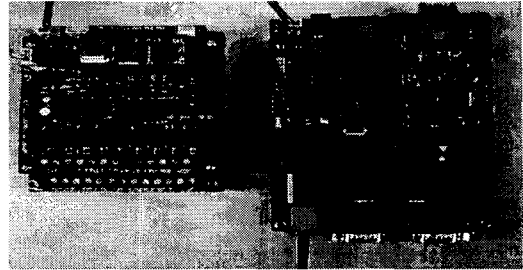


그림 12. 실험 장비의 실제 사진
 Fig. 12. The photograph of the experimental set

V 결론

본 논문에서는 TCP/IP protocol suite를 갖고 있지 않은 독립 장비들을 인터넷에 연결할 수 있는 Embedded network agent의 필요성과 이러한 Embedded network agent들이 갖추어야 할 기능을 살펴보고, 실제로 Embedded network agent를 독립 장비와 연계하여 인터넷 상에서 실험을 하였다. TCP/IP protocol suite를 작성하여 실험하였으며, 기존의 TCP/IP protocol suite에서는 고려되지 않던 Embedded network agent와 독립 장비 사이의 통신 상태를 고려한 Pre-open state를 정의하여 이를 Embedded network agent의 TCP/IP protocol suite에 반영하였다. Embedded network agent와 독립 장비 사이의 통신 상태를 고려하는 TCP/IP protocol suite를 사용함으로써 원거리 인터넷 통신상에서의 RTT를 줄였으며, 통신 에러 발생시의 인터넷상으로 재전송되어지는 통신 Packet의 수를 줄임으로써 네트워크의 부하를 감소시킬 수 있음을 보였다.

본 논문에서 제안한 Embedded network agent는 인터넷에 연결될 수 없었던 기존의 장비를 인터넷과 연결시켜줌으로써 원격지에서도 장비들을 효율적으로 감시·제어할 수 있는 기반을 제공한다. 실제 실험에서 Embedded network agent와 독립 장비인 PLC의 연결은 RS-232를 사용하였으나, Embedded network agent와 독립 장비 사이의 네트워크는 그림 2와 같은 통신 방식이라면 네트워크의 통신 방식에 구애 받지 않고 제시한 TCP state의 적용이 가능하다고 본다.

참고문헌(또는 Reference)

- [1] Behrouz A. Forouzan, "TCP/IP Protocol suite," McGRAW-HILL Higher Education, Singapore, July, 1999.
- [2] Jon Postel, "Transmission Control Protocol," RFC 793, Internet Engineering Task Force, Sep., 1981.
- [3] W. Richard Stevens, "TCP/IP Illustrated, Volume I: The Protocol," Addison-Wesley Publishing Company, Inc., Massachusetts, Mar., 1996.
- [4] 안광혁, 유영동, "원격 감시를 위한 내장형 소형 웹 서버" 제어·자동화·시스템 공학회 합동 학술 발표회 논문집, pp.116, Oct. 2000.