안정적인 데이터 공유를 위한 일방향 전송 알고리즘에 대한 연구

강 민 경*, 고 근 호**, 안 성 진***

요 약

국네트워크를 통한 데이터의 공유로 인해 각종 분야에서 첨단 응용 연구 성과를 보여 왔다. 그만큼 이에 적합한 안정적 데이터 공유 환경의 필요성 역시 꾸준히 제기되었다. 본 연구에서는 이러한 필요성에 따라 등장한 여러 성과 중 일방향 통신 지향 프로토콜에 대해 다룬다. 이 프로토콜은 오직 일방향만으로 데이터를 전송토록 하는 체계이다. 이는 불확실하고 잠재적인 위협 요소로부터 시스템을 지켜낼 가능성이 높아지지만, 데이터에 대한 확인/응답(ACK) 패킷을 전혀 받을 수 없다. 따라서 TCP, ICMP와 같은 양방향 프로토콜이 제대로 작동하면서 실질적인 통신이 이루어지도록 해야 한다. 이에 따라 본 연구에서는 인터페이스 계층, 인터넷 계층, 전송 계층에 대해서 각각 존재하는 프로토콜별 가능한 시나리오에 대해 분석하고 각각의 해결방법에 대해 접근하였다.

A Study on the Algorithms of One-way Transmission for Stable Data Sharing

Min Kyung Kang*, Keun Ho Koh**, Seong Jin Ahn***

ABSTRACT

As many high-tech researches are shown in various field by sharing data via computer network, a necessity for stable data-sharing environment has suggested steadily. This research covers one-way communication protocol which is newly developed through the necessity. In this protocol, data is transmitted only in one way. One-way transmission has strong possibility to protect system from uncertain and potential risk, but it is impossible to receive ACK(Acknowledge Character) packet about data. Therefore, we need to find a way which duplex protocol such as TCP, ICMP works properly and practical communication falls under one-way transmission environment. To conclude, we analysed possible scenario in each protocol from three layers - interface layer, internet layer and transmission layer and approached how to resolve the problem in each way..

Key words : One-Way Transmission, Duplex Protocol, TCP, ICMP, ACK

접수일(2016년 8월 24일), 수정일(1차: 2011년 12월 1일), 게재확정일(2016년 12월 16일)

★ 본 논문은 미래창조과학부 및 정보통신진흥센터 의 SW컴퓨팅산업원천기술개발 사업의 일환으로 수행하였음. [R0126-15-1095, 사이버·물리시스템에서의 물리적 단방향 보안 게이트웨이 개발]

* 성균관대학교 통계학과
** 성균관대학교 컴퓨터교육과
*** 성균관대학교 컴퓨터교육과

1. 서 론

네트워크를 통한 데이터 통신 과정을 통하여 데이터를 공유하고 이를 바탕으로 각종 분야에서 첨단 응용 연구 성과를 보여 왔다. 이렇듯이 네트워크와 이를 바탕으로 한 데이터 공유 환경의 구축은 이전에는 쉽지만은 않았던 것들을 가능하게 해왔다. 이에 따라 네트워크를 통한 데이터 통신이 중요시되면서 데이터를 주고받기 위한 환경과 규약, 체계 역시 복잡해진 것 또한 사실이다. 따라서 이에 적합한 안정적 데이터 공유 환경의 필요성 역시 꾸준히 제기되어왔다. 실제로 데이터를 공유하는 과정에서 여러 가지 보안을 위협할만한 요소들이 꾸준히 등장해왔기 때문이다. 이러한 필요성에 따라 여러 보완책들과 노력의 결과가 나타났다.

본고에서는 일방향으로 데이터를 전송하면 불확실하고 잠재적인 위협 요소로부터 시스템을 지켜낼 가능성이 높아진다는 점에 착안해 새로 개발된 일방향 통신 지향 프로토콜에 대해 다룬다. 이 프로토콜은 응용 계층에서 작동하는 프로토콜로 오직 일방향으로만 데이터를 전송하는 체계이다. 하지만 이 방식으로 데이터를 전송할 경우, 데이터에 대한 확인/응답(ACK) 패킷을 전혀 받을 수 없으므로 TCP와 같은 연결 지향성인 프로토콜이 제대로 작동할 수 없다[1]. 따라서 본고에서는 TCP/IP Layer 모델 중 네트워크 인터페이스 계층, 인터넷 계층, 전송 계층에 대해 각각 프로토콜별 가능한 시나리오 분석 및 해결방법에 대해 접근하였다.

2. 네트워크 계층 및 프로토콜 분석

본고에서는 오직 유선 통신이고 일반적으로 많이 사용되는 이더넷(Ethernet) 프로토콜 기반 환경과 2개의 노드가 하나의 회선으로 연결되어 있는 간단한 Topology에 대해서만 다룬다.

2.1 네트워크 인터페이스 계층

2.1.1 LLC, MAC 부계층

LLC 부계층 프로토콜은 무잡음 채널(Noiseless Channel)과 잡음이 있는 채널(Noisy Channel)로 나눌 수 있다. 우선 무잡음 채널에서는 Simplest Protocol과 Stop-and-Wait Protocol이 존재한다. Simplest Protocol은 흐름 제어나 오류 제어를 하지 않고 데이터 프레임만 보내는 방식이다. Stop-and Wait Protocol의 경우 송신 노드가 한 개의 프레임을 전송한 후에 수신자로부터 응답 신호를 받아야만 다음 프레임을 전송한다. 잡음 채널에서는 Stop-and-Wait Protocol ARQ(Automatic Repeat Request), Go-Back-N ARQ, Selective Repeat ARQ가 존재한다. 먼저 Stop-and-Wait ARQ는 앞서 알아본 Stop-and-Wait Protocol에 오류 제어 기능이 추가된 것이다. Go-Back-N ARQ는 Stop-and-Wait ARQ의 데이터 프레임을 한 개씩 전송한다는 단점을 보완한 것이다. 이는 Sliding Window를 통해 여러 개의 데이터 프레임을 동시에 보낼 수 있지만, 수신 노드에서 오류 발생 이후의 프레임들을 받아들이지 못한다는 단점이 있다. 이에 따라 한 단계 더 나아간 Selective Repeat ARQ는 오류가 발생한 이후에도 프레임 자체에 오류가 없으면 수신 노드에서 받아들인다[2].

MAC 부계층과 관련해서 여러 매체의 접근 제어 방법들이 있는데 Polling, Token Passing 등이 존재한다. 우선 폴링(Polling)의 경우 주국은 종국에게 전송할 데이터 프레임이 있는지를 물어보기 위해 Poll 프레임을 보낸다. 종국이 ACK 프레임으로 응답을 하면 주국은 데이터를 받을 준비를 하고 종국은 데이터 프레임을 전송한다[2]. Token Passing은 토큰(Token)이라고 불리는 패킷을 통해 이루어진다. 토큰을 붙잡게 되는 지국은 채널 접근 권한을 갖게 되고 데이터를 전송하게 된다[3].

2.2 인터넷 계층

2.2.1 IP/ICMP과 ARP/RARP

IP 패킷 프레임에 의해서 데이터가 전송될 경우

확인/응답 절차가 없어서 패킷이 확실히 전송된다는 보장이 없다. 따라서 오류 통지 기능과 진단용 문답 메시지를 보내는 기능을 하는 ICMP를 통해서 보완한다. ICMP 메시지의 종류는 오류 보고 메시지(Error Reporting Message)와 질의 메시지(Query Message)로 나뉜다. 오류 보고 메시지의 경우, 송신지로부터 수신지까지 IP 패킷을 전달하는 과정에서 문제가 발생할 때 이러한 사실을 송신 노드에게 전달해준다. 질의 메시지의 경우 네트워크에 존재하는 호스트가 다른 호스트, 라우터로부터 특정한 정보를 얻고자 하는 경우 사용된다. 이는 Request와 Reply로만 구성된다[4].

ARP(Address Resolution Protocol)는 네트워크 계층에서 사용되는 논리적 주소인 IP 주소를 물리적 주소인 MAC 주소로 매핑(Mapping)시켜주기 위한 프로토콜이다. RARP는 반대로 물리적 주소에 해당하는 IP 주소를 얻고자 하는 것이다[5].

2.3 전송 계층

2.3.1 TCP

TCP는 연결 지향형 프로토콜이고 데이터를 전송하기 전에 미리 가상 경로(Virtual Path)를 설정하고 이 경로를 통해서 데이터를 전송한다. 이때 연결을 설정하는 방식은 3-Way Handshaking이다. 처음에 송신 노드 측에서 SYN 패킷을 보내면 수신 노드는 이에 대한 응답으로 SYN+ACK 패킷을 보내고, 이에 대한 응답으로 송신 노드는 다시 ACK 패킷을 보냄으로써 연결 설정 과정이 이루어진다. 데이터 전송 과정에서는 수신 측에서 데이터를 받을 때마다 ACK를 전송하고, 데이터 전송이 모두 끝나면 연결 설정이 종료된다[6].

3. 문제 상황 진단

앞선 내용을 통해서 TCP/IP Layer Model 중 계층별 프로토콜에 대한 정보를 제시하였다. 그런데 이러한 프로토콜은 모두 양방향 통신 하에 만들어진 것이다 보니 일방향 통신을 할 경우에는 제대로 작동하지 않는다. 이에 따라 본 연구 과제에서는 이러한 문제 상황들을 계층별로 분석하고 진단하였다. 응용 계층의 경우 단순히 사용자가 선택하는 서비스에 대한 개념이므로 일방향 전송과는 직접적인 상관이 없기 때문에 연구 대상에 포함하지 않는다.

<표 1> 계층별 문제 상황

계층	프로토콜	문제상황
네트워크 인터페이스	-	MAC 부계층의 경우 이더넷은 피드백이 필요하지 않지만, LLC 부계층에서는 ARQ나 HDLC를 기반으로 Node-to-Node 구간의 오류 제어 및 흐름 제어의 피드백을 수행한다.
인터넷	ARP/ RARP	송신 노드가 ARP Request 패킷을 브로드캐스트로 전송하게 되면 해당 수신 노드만이 Reply 패킷을 보낸다.
	IP/ ICMP	ICMP는 전송한 패킷의 이상 상황에 대한 정보를 송신 노드에 전달한다.
전송	TCP	3-Way Handshaking과 같은 연결 설정 과정이 필요하고 송신 노드에서 전송한 데이터 패킷에 대해서 ACK와 같은 확인/응답 절차가 필요하다.

4. 계층별 일방향 전송 알고리즘

4.1 네트워크 인터페이스 계층

MAC 부계층의 경우 본 연구에서는 이더넷 환경만을 다루기로 하였다. 그런데 이더넷(Ethernet)의 경우 별도의 피드백이 필요하지 않기 때문에 MAC 부계층에서는 일방향 통신으로 인한 문제가 발생하지 않는다. LLC 부계층에서는 ARQ나 HDLC를 기반으로 Node-to-Node 구간의 오류 제어와 흐름 제어를 하는 기능을 수행한다. 따라서 본 연구 과제에서는 LLC 부계층에서의 오류 제어와

흐름 제어 기능이 이루어지지 않도록 하였다. 그 이유는 이미 전송 계층에서의 TCP를 통한 양 종단 간의 오류 제어, 흐름 제어와 ICMP를 통한 피드백 절차가 있기 때문이다.

4.2 인터넷 계층

4.2.1 ARP/RARP

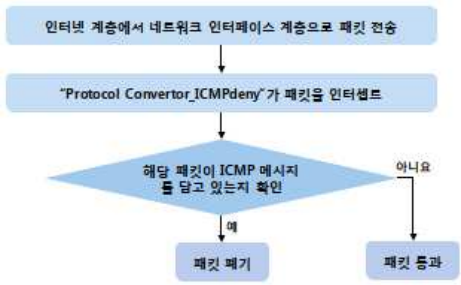
1. ARP_static/RARP_static

각각의 노드의 ARP Cache Table에서 동적(Dynamic)으로 설정되어 있는 주소에 대해서 송신 노드가 ARP Request 패킷을 브로드캐스트로 LAN 전체에 전송하게 되면 ARP Request 헤더의 'Target Protocol Address' 필드의 값과 IP 주소가 일치하는 노드만이 Reply 패킷을 보내고 나머지 노드들은 Request 패킷을 무시한다[9]. 그런데 이때 ARP Reply 패킷의 경우 일방향 통신과는 거리가 멀다. 따라서 수신 노드가 ARP Reply 패킷이 전송될 필요가 없도록 ARP Cache Table에 존재하는 목록들을 모두 정적(Static) 모드로 설정하는 방안을 고려했다. RARP 역시 ARP와 비슷한 방식으로 동작하기 때문에 이 역시 Static 모드로 설정해놓는다.

4.2.2 IP/ICMP

1. ICMP_DENY

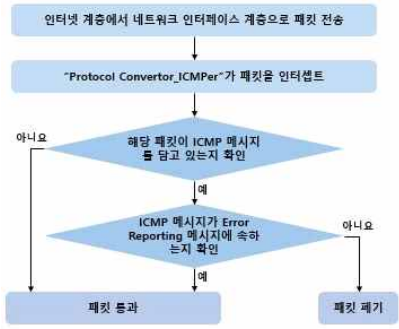
본 방안은 모든 ICMP 메시지가 발생되지 않도록 하는 방안이다. 이렇게 접근한 이유는 ICMP 메시지를 통한 오류 파악 과정이 생략되어도 실질적 데이터 전송에는 문제가 되지 않기 때문이다. 이를 위해서는 양쪽 노드 모두에 ICMP 메시지를 폐기해주는 별도의 모듈 "Protocol Convertor_ICMPdeny"이 구축되어야 한다. 이 모듈은 어떤 서비스와 관련된 패킷을 전달하는지를 나타내는 Protocol 필드를 이용하여 ICMP 서비스(ICMP의 경우 이 필드에 대한 값은 1이다.)에 대한 패킷만을 찾아내어 폐기한다.



<그림 1> Protocol Convertor_ICMPdeny의 알고리즘

2. ICMP_PermitER

본 방안은 오류 보고를 위한 메시지는 통과시켜 송신 노드가 오류 정보를 전달받을 수 있도록 하는 관점이다. 다만, 이 경우에 ICMP Query 메시지는 작동되지 않도록 하기 위해 송수신 노드에서 모듈 "Protocol Convertor_ICMPer"을 구축하여 Query Request, Reply 패킷을 필터링한다. 이 모듈은 인터넷 계층으로부터 전해 내려오는 패킷을 받아 해당 패킷이 ICMP 메시지를 담고 있는지 여부를 확인하고 ICMP 메시지를 담고 있는 경우에 ICMP Header 중에서 Type 필드의 값이 ICMP Error Reporting 메시지에 속하면 통과시키고 그렇지 않다면 폐기시킨다.



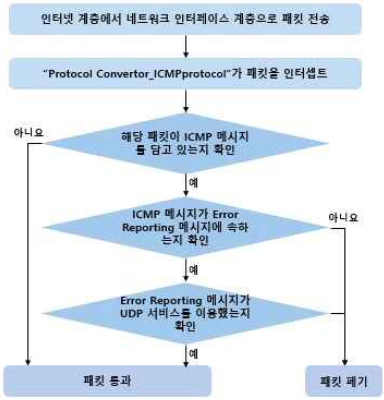
<그림 2> Protocol Convertor ICMPer의 알고리즘

3. ICMP_Protocol

본 방안은 앞서 언급한 "ICMP_PermitER" 방

안에 다른 조건을 추가한 것이다. 모듈 “Protocol Convertor_ICMPprotocol”을 구축하여 IP Header 부분의 Protocol 필드의 값에 따라 ICMP Error Reporting 메시지를 Source Node로 보낼지 말지 결정한다.

Protocol 필드의 값이 6이면 전송 중 문제가 발생한 패킷이 TCP 서비스, 17이면 UDP 서비스를 이용했다는 것을 알 수 있다. 본 방안에서는 이 점에 착안하여 UDP인 경우에만 ICMP Error Reporting 메시지가 정상적으로 전송되도록 한다.



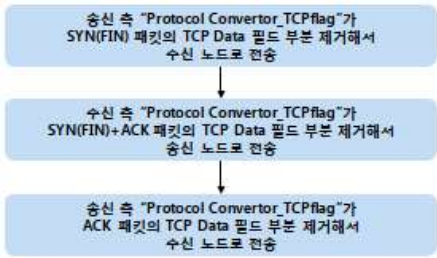
<그림 3> Protocol Convertor ICMPprotocol의 알고리즘

4.3 전송 계층

4.3.1 TCP

1. TCP_MinFunc

본 방안은 TCP에서의 3-Way Handshaking 과정에서 꼭 필요한 형태의 패킷만이 회선을 통해서 전달될 수 있도록 하는 관점에서 제시된 방안이다. 본 방안도 마찬가지로 송수신 노드에 독립적으로 작동하는 모듈 “Protocol Convertor_TCP Flag_asf”을 구축하여야 한다. 각각의 모듈에서는 피드백 패킷을 이용하여 의미 있는 정보가 일방향 통신 방향의 반대 방향으로 전송되는 것을 방지하기 위해 연결 설정 시와 연결 종료 시에는 TCP Data 필드를 폐기한다. 또한, 이상한 패킷이 유입되지 못하도록 ACK, SYN(연결 설정), FIN(연결 종료) 패킷을 받을 때마다 항상 Sequence Number와 Acknowledgement Number를 검사한다.



<그림 4> TCP_MinFunc의 알고리즘

2. TCP_Permit_ack

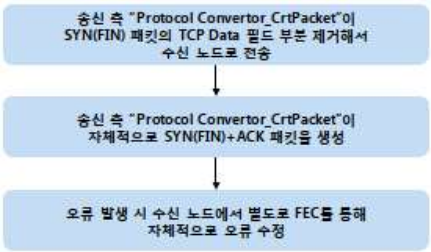
"TCP_Permit_ack"는 목적지에 해당하는 노드로부터 송신 노드에 해당하는 노드로 전송되는 패킷을 ACK 플래그만으로 한정시키는 것이다. 하지만, ACK만으로는 TCP에서의 연결 설정 또는 연결 종료 과정이 전혀 이루어질 수 없기 때문에 양쪽 노드에 별도의 모듈을 구축해야 한다. 송신 노드에는 자체적으로 SYN+ACK와 FIN 패킷을 생성하는 “Protocol Convertor_CrtPacket” 모듈을 구축하고, 수신 노드에는 ACK 패킷만을 허용하는 "Protocol Convertor_Permit_ack"를 구축한다.

<그림 5> TCP_Permit_ack의 알고리즘

3. TCP_NoFeedback_FEC

본 방안 “TCP_NoFeedback_FEC”는 전송 계층인 프로토콜 상에서 아무런 피드백이 이루어지지 않도록 하는 것으로, 송신 노드로 어떠한 패킷도 전달되지 못하도록 필터링 하는 “Protocol Con

vertor_DenyAll" 모듈과 피드백에 필요한 패킷을 자체 생산해내는 "Protocol Convertor_CrtPacket " 모듈을 구축해야 한다. 이때, 수신 노드에서 별도로 FEC(Forward Error Control)를 통해 수신 노드 자체에서 오류를 정정한다.

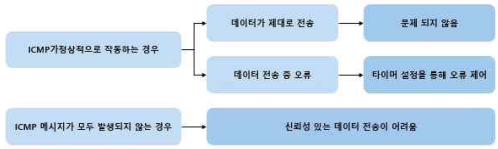


<그림 6> TCP_NoFeedBack_FEC의 알고리즘

4. TCP_NoFeedback_icmp

본 방안 "TCP_NoFeedback_icmp"는 "TCP_NoFeedBack_FEC"와 마찬가지로 수신 노드에서 구축된 모듈을 통해서 모든 TCP 패킷이 송신 노드로 전달되지 못하도록 필터링한다. 또한 동시에 송신 노드의 모듈에서는 연결 설정과 종료에 필요한 SYN+ACK, FIN 패킷과 ACK 패킷을 자체 생산한다. 본 방안은 인터넷 계층에서의 ICMP가 정상적으로 작동하는 경우와 ICMP 메시지가 모두 발생되지 않도록 하는 경우로 세분화할 수 있다. 전자부터 살펴보면, 데이터 전송 과정에서 오류가 발생한 경우를 대비하여 송신 측 모듈인 "Protocol Convertor_CrtPacket"에서 송신 노드로부터 데이터 전송이 시작되어 모듈을 통과할 때 별도의 타이머를 설정해 놓는다. 이 타이머는 ICMP Error Reporting 메시지에 대한 타이머이고, 이는 TCP 프로토콜의 타이머보다 "모듈에서 ICMP Error Reporting 메시지 발생 여부를 판단하여 이에 따라 모듈에서 자체적으로 ACK를 생산하여 송신 노드의 상위 계층으로 전달하는데 필요한 시간"만큼 짧아야 한다. 또한 오류 발생 시 ICMP 메시지가 송신 노드까지 다시 되돌아오는데 충분한 시간이어야 한다. 이 조건 하에서 데이터 전송 중에 위·변조가 일어날 경우에는 수신 노드에서의 FEC를 이용하여 오류를 제어한다.

후자의 경우에는 TCP에서의 ACK 패킷을 통한 일련의 피드백 절차뿐만 아니라 ICMP Error Reporting 메시지도 제대로 작동하지 않기 때문에 사실상 신뢰성 있는 데이터 전송을 기대하기 어렵다. 실제로는 데이터 전송 과정에서 오류가 발생했어도 송신 노드의 모듈에서 자체적으로 ACK를 생산하여 송신 노드로 전달하기 때문에 실제 데이터 전송의 결과와는 전혀 다른 잘못된 확인/응답 결과가 나타날 수 있다. 따라서 본 방안을 선택할 때는 앞서 언급한 ICMP 관련해서 제시한 방안 중에서 모든 ICMP 메시지를 차단하는 것과 같이 사용되지 않도록 유의해야 할 필요가 있다.



<그림 7> TCP_NoFeedback_icmp의 알고리즘

5. 결론

본 논문에서는 일방향 전송장치 표준모델 마련을 위한 연구를 통해서 세 가지의 기대할 수 있는 성과와 두 가지의 활용 방안을 제시하였다. 본 연구의 성과로서 첫째, 안정적인 데이터 통신 환경을 구축했다는 점, 둘째 보안 강화를 위한 네트워크 설계에 도움이 되었다는 점, 셋째 네트워크 관리 정책 수립을 위한 가이드라인을 제시하였다는 점을 들었다. 그리고 이 연구결과가 어떤 식으로 활용될 수 있는지 다음 두 가지 정도로 정리해 볼 수 있다. 첫째, 일방향 통신 네트워크 환경 구축 시 자율적 환경을 설정할 수 있고, 둘째 일방향 통신 환경 관리 정책을 수립하는 데에 도움을 줄 수 있을 것으로 보인다.

참고문헌

[1] 박항규, "물리적 연결선 차단을 이용한 고품질 단방향 보안 통신방법 구현", 고려대학교 정보보호

대학원, 2012.
[2] 박기현, “데이터 통신과 컴퓨터 네트워크”, 한빛아카데미, 2013.
[3] 진강훈, “후니의 쉽게 쓴 시스코 네트워킹”, 성안당, 2010.
[4] 최선철・차현철, “ICMP 프로토콜을 사용하는 네트워크 침입의 탐지 구현”, 동양대학교, 2001.
[5] 진혜진, “네트워크 개론”, 한빛아카데미, 2014.
[6] 김혁, “UNIX운영체제에서의 TCP/IP 프로토콜의 전송 및 수신 성능 분석”, 한림대학교 대학원, 1997.
[7] 양대일, “정보 보안 개론과 실습 - 네트워크 해킹과 보안”, 한빛미디어, 2010.
[8] 박상호, “단방향 링크 Ad-hoc 망을 위한 라우팅 프로토콜”, 정보·보안 논문지(Journal of Information and Security) Vol. 7, 한국사이버테러정보전학회, 2007.
[9] 윤종호, “TCP/IP와 라우팅 프로토콜”, 교학사, 2003.
[10] 강원종, “이더넷 환경에서 신뢰적인 Self-configuration을 통한 IP주소 할당 메커니즘”, 서울산업대학교 산업대학원, 2004.
[11] J.Pstel, “Internet Protocol”, RFC791, 1981.
[12] D.D. Clark・V.jacobson, “An analysis of TCP processing overhead”,1989.
[13] Behrouz A. Forourzan, "Data Communications and Networking", McGraw-Hill, 2007.

[著者紹介]

고 근 호 (Keun Ho Koh)
성균관대학교 컴퓨터교육과 소속
email : keuno0923@gmail.com

강 민 경 (Min Kyung Kang)
성균관대학교 통계학과 소속
2016년 2월 이학사
email : mnmnaldma@naver.com

안 성 진 (Seong Jin Ahn)
1988년 2월 학사
1990년 2월 석사
성균관대학교 대학원
정보공학과 공학석사
1998년 8월 박사
성균관대학교 대학원
정보공학과 공학박사
email : sjahn@skku..edu