Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.26, NO.6, Dec. 2016 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2016.26.6.1571

신뢰성 있는 단방향 데이터 전송 시스템 설계

김 동 욱,† 민 병 길‡
ETRI 부설연구소

Design of a Reliable Data Diode System

Dongwook Kim,† Byunggil Min‡
The Attached Institute of ETRI

요 약

단방향 전송 기술에서 해결해야 할 이슈들 중에 한 가지는 TCP 기반의 데이터 전송에서 발생하는 패킷 손실을 줄이는 것이다. 잘 알려진 에러 수정 기법들을 활용해서 패킷 손실을 줄일 수 있다. 하지만, 기존의 여러 기법들을 활용한다고 하더라도, 링크 에러와 버퍼 오버플로우에 의한 패킷 손실은 여전히 발생할 수 있다. 본 논문에서는 신뢰성 있는 단방향 데이터 전송 시스템(RED, REliable Data diode)을 제안한다. RED는 기존의 단방향 전송 기술과 마찬가지로 TCP 기반의 데이터 전송을 지원하기 위해 TCP 프록시 기법을 활용한다. RED 송신시스템은 TCP 패킷의 지연 전송을 활용하여 버퍼 오버플로우에 의한 패킷 손실을 줄일수 있다. 또한, RED 송신시스템은 패킷의 중요도 및 여유 자원을 고려하여, RED 수신시스템에게 다수의 동일한 패킷을 복제 및 전송함으로써, 단방향 전송 링크에서의 링크 에러에 의한 패킷 손실을 줄일 수 있다.

ABSTRACT

One of the issues, which is dealed with in undirectional data transmission technology, is reducing the packet loss in TCP based data transfer. We can decrease the packet loss by using several well known error correction approaches. Although we utilize those previous approaches, the packet loss by both link errror and buffer overflow could be occurred. In this paper, we propose the RED(REliable Data diode). RED also uses the TCP proxy approach for supporting the TCP based data transfer which is similar with the existing unidirectional data transmission technologies. The RED transmission system could alleviate the packet loss caused by buffer overflow by exploiting the delaying transmission of TCP packets. Furthermore, in order to reduce the packett loss caused by link error in the unidirectional transmission link, the RED transmission system transmits one or more duplicated packets to the RED reception system by considering both the remaining resources and packet importance.

Keywords: Unidirectional Data Transmission System, Data Diode, packet loss, link error, buffer overflow

I. 서 론

데이터 다이오드(data diode)라고도 불리는 단방향 전송 기술 또는 시스템([1-13])은 한 네트워크(A)에서 다른 네트워크(B)로 물리적인 단방향 연결을 제공하는 기술이다. 이 기술은 A→B 방향으로만 데이터 전송이 가능하다. 최근에 단방향 전송 기술은 화학 발전 등 산업 제어시스템 네트워크에 널리 적용 및 사용되고 있다. 널리 알려진 단방향 기술로는 Waterfall 사의 Unidirectional Security Gateway[5]와 Owl사의 Dual-Diode[6] 및 Fox-IT사의 FFHDD(Fort Fox Hardware Data Diode)[12]가 있다.

단방향 전송 기술은 백워드 링크(backward link)가 없기 때문에 패킷 손실은 불가피하며, 이는 단방향 전송 기술에서 해결해야 할 주요 이슈 중에

Received(09. 01. 2016), Modified(10. 18. 2016), Accepted(10. 18. 2016)
† 주저자, dwkim1980@nsr.re.kr
‡ 교신저자, bgmin@nsr.re.kr(Corresponding author)

하나이다[10]. 패킷 손실의 주요 원인은 크게 링크 에러와 네트워크 혼잡이다. 과거의 연구에서, FEC(Forward Error Correction)[19-21] 및 패킷 결합(packet combining)[22-25]과 같은 다양한 에러 정정 기술이 링크 에러에 의한 패킷 손실을 줄이기 위해 제안되었다. 이러한 접근법들은 명시적으로 수신측으로부터 어떠한 정보도 요청하지 않으므로, 백워드 링크를 가지지 않는 단방향 적용기술에 효과적으로 활용될 수 있다. 상용 이더넷 NIC(Network Interface Card)에서는 일반적으로 이러한 에러 정정 메커니즘을 지원하지 않으므로, 상용 이더넷 기반 단방향 전송 기술에 적용을 위해서는 NIC의 펌웨어 수정이 필요할 수 있다. 또한 네트워크 혼잡에 의해 버퍼 오버플로우가 발생할 수 있는데, 이로 인한 패킷 손실은 라우터나 목적지 단말과 같은 노드의 버퍼 크기를 높이면 완화될 수 있다. 하지만, 네트워크의 혼잡 정도가 일정 임계치를 초과할 경우, 버퍼 오버플로우에 의한 패킷 손실은 여전히 발생할 수 있다.

본 논문에서는 단방향 전송 기술에서 발생할 수 있는 링크 에러 또는 버퍼 오버플로우에 의한 패킷 손실을 완화할 수 있는 방안을 제시한 신뢰성 있는 단방향 전송 시스템(RED, Reliable Data Diode)을 설계하였다. RED는 기존의 단방향 전송 기술과 마찬가지로 TCP 기반의 데이터 전송을 지원하기 위해 TCP 프록시(TCP Proxy) 기법을 사용한다. RED는 커널레벨에서 TCP 패킷의 지연 전송을 활용하여 버퍼 오버플로우에 의한 패킷 손실을 줄일 수 있다. 또한 RED 송신기는 여유 자원이 있거나 스케줄에 의해 RED 수신기에게 다수의 동일한 패킷을 생성 및 전송하여, 단방향 전송 링크에서의 링크 에러에 의한 패킷 손실을 줄일 수 있다. 해당 과정은 유저레벨에서 구현이 가능하므로 상용 NIC의 펌웨어를 수정할 필요는 없다.

본 논문의 나머지 구성은 다음과 같다. 2장에서는 배경, 해결해야 할 이슈 및 관련 연구에 대해서 살펴본다. 3장에서는 제안하는 신뢰성 있는 단방향 데이터 전송 시스템(REliable Data diode, RED)에 대해서 기술한다. 4장에서는 제안하는 시스템과 기존 시스템에 대해서 성능을 비교 분석하며, 5장에서는 결론을 맺는다.

II. 배경, 이슈 및 관련 연구

2.1 배경

Fig.1.은 단방향 전송 시스템(혹은 데이터 다이오드)가 적용된 네트워크의 예시를 나타낸다. Fig.1.에서 볼 수 있듯이, 단방향 전송 시스템은 송신시스템과 수신시스템으로 구성되며, 송신시스템에서 수신시스템으로만 물리적으로 데이터 전송이 가능한 시스템이므로, 네트워크1의 A1~A3가 네트워크2의 B1~B3로 데이터 전송이 가능하며, B1~B3는 A1~A3로 데이터 전송은 불가능하다.

이러한 단방향 전송 시스템은 적용 환경에 따라, Table 1.의 기능을 High Security(높은 보안) 네트워크에 제공할 수 있다. 예를 들어, 단방향 적용 시스템이 case1에 적용되면, 네트워크1은 네트워크2를 통한 사이버공격은 불가능하므로, 가용성(availability)과 무결성(Integrity)을 보장받을 수 있다. 전력 발전소와 같은 제어시스템에 case1을 적용하면, 단방향 전송 시스템은 제어시스템이 포함된 네트워크(네트워크1)를 외부 네트워크(네트워크2)로부터 안전하게 보호할 수 있다. case2에 적용되면, 단방향 적용 시스템은 네트워크2의 자료가 외부로 세어나가지 않게 기밀성(confidentiality)을 보장할 수 있다. case2는 투표시스템에 활용되어, 투표 시간동안은 외부로 정보가 공개되지 않도록 기밀성을 제공하는데 활용될 수 있다.

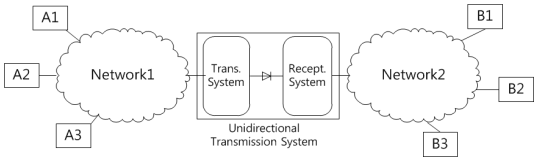

Fig. 1. An example of networks applying uni-directional data transmission system

Table 1. The support function of uni-directional data transmission system according to the usage case

	Network1	Network2	Function
case1	High Security	Low Security	Integrity, Availability
case2	Low Security	High Security	Confidentiality

2.2 이슈

단방향 전송 시스템은 일반적으로 백워드 링크를 사용하지 않기 때문에 패킷 손실의 발생이 나타날 수 밖에 없다[10]. 하지만, 파일 전송과 같은 무손실 전송이 필요하므로, 단방향 전송 시스템에서 패킷 손실은 해결해야 할 주요 이슈중 하나이다. 단방향 전송 시스템에서 패킷 손실이 발생하는 원인은 크게 링크 에러에 의한 손실과 버퍼 오버플로우에 의한 손실로 나눌수 있다. 세부 절에서 각 손실에 대해서 자세히 살펴본다.

2.2.1 링크 에러에 의한 패킷 손실

단방향 전송 시스템은 송신시스템에서 수신시스템으로 단방향 물리 링크를 사용하여 전송한다. 이 때 IEEE 802.3 MAC[15]과 같은 링크 계층을 사용할 수 있다. 802.3의 경우, MAC 헤더의 FCS(Frame Check Sequence)를 활용하여 수신한 데이터의 오류검사를 수행하는데, MAC 계층에서는 단일 비트의 에러만 발생해도 수신한 패킷을 버린다. 예를 들어, BER(Bit Error Rate)가 10e-8일 경우, 1bit의 에러가 100Mbit의 트래픽을 전송할 때마다 발생할 수 있다. FEC[19-21] 및 패킷 결합[22-25]과 같은 기법들을 단방향 전송 시스템의 수신시스템에 구현하여 사용할 수 있지만, 이는 상용 이더넷 NIC의 펌웨어 수정이 필요할 수도 있다.

2.2.2 버퍼 오버플로우에 의한 패킷 손실

단방향 전송 시스템의 경우, TCP 기반의 데이터 전송을 지원하기 위해 TCP 프록시 기법을 활용한다. 먼저, TCP 프록시[26, 27]를 살펴보면 Fig.2.와 같다. TCP 프록시는 TCP 송신지(TCP Sender)와 TCP 목적지(TCP Destination)간에 중간 서버로서의 역할을 수행한다. TCP 송신지가 TCP 목적지로 연결(connection)을 맺을 때, TCP 프록시는 비밀리(TCP 송신지와 TCP 목적지가 판단하지 못하게)에 연결 설립 절차에 참여하며, 결론적으로는 2개의 TCP 세션(Fig.2.의 TCP 연결1과 TCP 연결2)이 만들어진다. TCP 송신지가 전송하는 모든 TCP 트래픽은 TCP 프록시를 통해서 TCP 목적지로 전달된다.

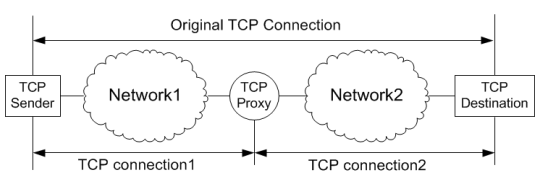

Fig. 2. An example using a TCP proxy

단방향 전송 시스템은 앞서 설명한 TCP 프록시를 기반으로 수정된 TCP 프록시를 적용하고 있다. 이는 Fig.3.과 같다. 단방향 전송 시스템에서 송신시스템은 TCP 목적지 프록시(TCP Destination Proxy) 역할, 수신시스템은 TCP 송신지 프록시(TCP Sender Proxy) 역할을 각각 수행한다. 즉, TCP 송신지가 전송하는 모든 트래픽은 TCP 목적지 프록시(송신시스템)로 전달된다. 이제 TCP 목적지 프록시는 단방향 전송을 통해 트래픽을 수신하면 실시간으로 TCP 송신지 프록시로 전달한다. 마지막으로, TCP 송신지 프록시는 TCP 목적지로 수신한 트래픽을 전송한다.

이러한 단방향 전송 시스템에서는 TCP 송신지 프록시가 TCP 목적지 프록시로부터 수신한 데이터를 TCP 목적지로 실시간으로 전달할 경우, 송신시스템의 NIC 또는 수신시스템의 TCP 송신지 프록시에서 버퍼 오버플로우에 의한 패킷 손실이 발생할 수 있다. 단방향 전송 시스템의 경우 버퍼 오버플로우에 의한 패킷 손실이 발생해도 백워드 링크가 없어 TCP의 흐름 제어(flow control)[17]나 혼잡 제어(congestion control)[18]를 적용할 수 없는 한계가 존재한다. TCP 연결1이 X Mbps의 속도를 지원하고, TCP 연결2가 Y Mbps의 속도를 지원하며, 송신시스템에서 수신시스템으로의 단방향 링크가 Z Mbps를 지원한다고 가정하자. X 〉 Y 이고 Z 〉 X 인 경우, TCP 송신지 프록시가 TCP 목적지 프록시로부터 수신한 데이터를 실시간으로 TCP 목적지로 전달하면서 기 할당된 버퍼를 모두 소진하게 되면, TCP 송신지 프록시(응용프로그램)에서 병목현상으로 인한 버퍼 오버플로우가 발생할 수 있다. 예를 들어, X가 10Mbps, Y가 5Mbps이며, Z가

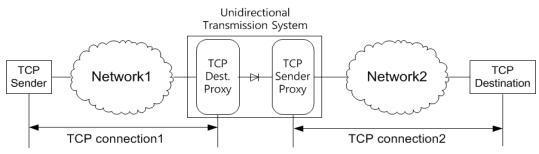

Fig. 3. An example using a TCP proxy in an unidirectional transmission system

100Mbps인 경우, TCP 송신지 프록시는 초당 5Mbits의 데이터가 버퍼에 쌓인다. 만약 기 할당된 버퍼가 100Mbits면 20초 후에 버퍼 오버플로우가 발생할 수 있다. X 〉 Z인 경우, 송신시스템이 수신 시스템과 연결된 NIC에서 병목현상으로 인한 버퍼 오버플로우가 발생가능하다. 하지만, 최근 물리적 단방향 링크(Z)는 1~10Gbps의 속도를 제공[5,6]하므로, X 〉 Z의 상황은 현실적이지 않기에, 본 논문에서는 고려하지 않는다.

2.3 관련 연구

기존의 단방향 전송 시스템의 경우, 링크 에러 및 버퍼 오버플로우에 의한 패킷 손실을 줄이기 위한 기술을 적용하고 있다. 본 절에서는 이와 관련한 기존 연구들의 동향에 대해서 살펴본다.

2.3.1 링크 에러에 의한 패킷 손실 완화와 관련한 연구

미리 정의된 목적으로만 사용가능한 백워드 채널을 활용하여 링크 에러에 의한 패킷 손실을 없애는 기술들이 제안되었다. Owl사의 OSAE(Owl Secure Acknowledgement Engine)[6]의 경우, 피드백 엔진을 탑재하여, 수신시스템이 수신한 패킷 중 손실이 발생한 패킷을 재전송하여 링크 에러에 의한 패킷 손실을 처리하였다. 이와 유사하게 Kim 등[14]은 GPIO(General Purpose Input Output)을 이용하여 링크 에러에 의한 패킷 손실을 해결하였다.

OSAE 및 Kim의 연구와는 달리 단방향 기조를 그대로 유지하면서, 패킷 손실을 완화하는 기술들이 제안되었다. Waterfall 사의 제품[5]은 송신시스템에서 데이터를 전송할 때, sequence number 및 에러 정정 코드를 포함하여 전송함으로써, 패킷 손실을 일부 정정할 수 있도록 하였다. Fox-IT사의 FFHDD(Fort Fox Hardware Data Diode)[12]도 waterfall 사의 기술과 유사하게 오류 정정을 수행한다. 이와 유사하게, Heo 등의 연구[13]는 단방향 전송 데이터의 신뢰성을 제공하기 위해 reed solomon(255, 239)을 활용하여 패킷 손실의 에러 발생 시 정정할 수 있도록 하였다.

이외에도 802.11[16] 등의 무선 네트워크에서 적용될 수 있는 FEC[19-21] 및 패킷 결합[22-25]들을 단방향 전송 시스템에 적용하여 패킷 손실을 줄일 수도 있다.

2.3.2 버퍼 오버플로우에 의한 패킷 손실 완화와 관련한 연구

버퍼 오버플로우에 의한 패킷 손실을 제거하기 위한 가장 효과적인 방안은 수신시스템에서 TCP 송신지가 보낸 데이터를 모두 저장해 두는 것이다. 즉, Fig.3.에서 1) TCP 목적지 프록시가 TCP 송신지로부터 데이터를 수신하고, 2) 송신시스템은 이를 수신시스템으로 단방향으로 전송하며, 3) 수신시스템은 이를 자신의 저장 공간에 저장해두고, 4) 수신시스템은 송신시스템으로부터 수신한 데이터에 대한 저장을 완료한 뒤에, TCP 목적지로 전달한다.

이 방안을 사용하면, TCP 송신지 프록시에서 버퍼 오버플로우는 제거할 수 있으나, 패킷 전송에 대한 지연시간이 증가할 수 있다. 예를 들어, 2.2.2절에서 X가 10Mbps, Y가 5Mbps이며, Z가 1Gbps인 경우에, 1Gbits의 트래픽을 전송한다고 가정하자. 이 때, TCP 연결1과 TCP 연결2에서 전송에 소요되는 시간은 각각 100초와 200초이다. 또한 송신시스템에서 수신시스템으로 전송하는데 소요되는 시간은 1초이다. 앞서 기술한 TCP 송신지 프록시에 저장해두는 방안을 활용하면, 프로세싱 시간 등을 고려하지 않고 전달 시간만 고려할 경우 총 301초의 시간이 소요된다. 전달 시간을 줄이기 위해서는, 버퍼에 저장하면서 전송하는 방안을 사용할 수 있으나, 이는 앞서 설명한 버퍼 오버플로우에 의한 패킷 손실이 발생할 수 있다.

Waterfall 및 Owl사 등의 제품 브로셔 등에서는 세부 기술에 대해서는 확인할 수 없었으나, 버퍼 오버플로우에 의한 패킷 손실이 발생하지 않도록 설계된 것으로 판단된다.

III. 신뢰성 있는 단방향 전송 시스템 설계

본 장에서는 제안하는 신뢰성 있는 단방향 전송 시스템(RED, REliable Data diode)에 대해서 설명한다. RED는 기존의 단방향 전송 기술과 마찬가지로 송신시스템과 수신시스템으로 구성되어 있으며, TCP 기반의 데이터 전송을 지원하기 위해 TCP 프록시 기법을 사용한다. RED는 수신시스템에서 발생할 수 있는 링크 에러에 의한 패킷 손실을 줄이기 위해, RED 송신시스템에 여유 자원이 있을 경우, 다수의 동일한 패킷을 생성 및 전송하는 방안을 사용한다. 그리고, 버퍼 오버플로우에 의한 패킷 손실을 줄

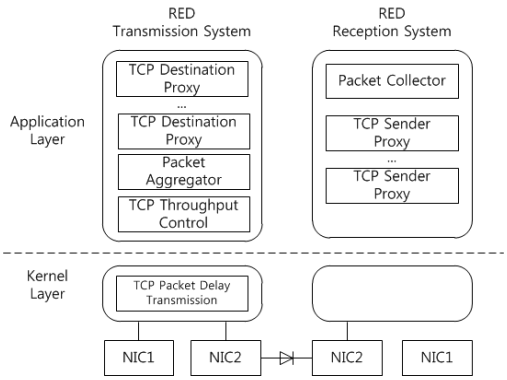

Fig. 4. RED Architecture

이기 위해, RED 송신시스템의 TCP 목적지 프록시는 TCP 송신지에게 지연된 TCP 패킷의 전송하는 방법을 활용한다. 세부 절에서는 RED의 상세에 대해서 살펴본다. 이에 앞서, 단방향 전송 시스템이 사용되는 네트워크는 운영자가 트래픽 사용량 및 트래픽 전송 속도 등에 대한 정보를 파악할 수 있다고 가정한다.

3.1 RED 구조

RED의 기본 구조는 Fig.4.와 같다. RED 송신시스템(RED Transmission System)은 NIC1과 NIC2 및 응용 계층의 TCP 목적지 프록시(TCP Destination Proxy) 기능, TCP 처리량 조절(TCP Throughput Control) 기능과 패킷 집합기(Packet Aggregator) 기능으로 구성되어 있다. RED 수신시스템(RED Reception System)은 NIC1과 NIC2 및 유저 레벨의 TCP 송신지 프록시(TCP Sender Proxy) 기능, 패킷 수집기(Packet Collector) 기능으로 구성되어 있다.

3.1.1 RED 송신시스템

우선 RED 송신시스템에 대해서 살펴보자. RED 송신시스템 NIC1은 Fig.2.의 네트워크1과 연결되어 있으며, NIC2은 RED 수신시스템과 연결되어 있다. 즉, RED 송신시스템은 NIC1을 이용하여 네트워크1에 연결되어 있는 기기들로부터 TCP 또는 UDP 패킷을 수신하고, 이를 처리한 후에 NIC2를 통해서 RED 수신시스템으로 전달하는 과정을 수행한다.

RED 송신시스템의 TCP 목적지 프록시 기능은 TCP 목적지를 모사한 응용프로그램으로 TCP 송신지에게 마치 자신이 TCP 목적지인 것처럼 판단하게 한다. TCP 기반의 응용프로그램마다 별도의 TCP 목적지 프록시가 구동되며, 이는 기존의 TCP 프록시 기능과 동일하므로 자세한 설명은 생략한다. TCP 목적지 프록시는 수신한 패킷 로그, DB 등의 파일 데이터는 저장 공간에 임시로 저장한다. 이는 TCP 목적지의 동작과 동일하다. 이와 동시에 TCP 목적지 프록시는 TCP 송신지로부터 수신하는 모든 TCP 패킷을 패킷 집합기로 전송한다. 또한, TCP 목적지 프록시는 주기적으로 TCP 송신지와의 RTT(Round Trip Time) 값(RTT_1)을 측정한다. 이는 TCP 송신지와 TCP 목적지 프록시 간에 송·수신되는 트래픽을 통해 측정 가능하다[28]. 측정된 RTT_1는 TCP 속도 조절 기능으로 전달된다.

패킷 집합기는 TCP 목적지 프록시로부터 TCP 패킷을 받으며, 이와 동시에 NIC1으로부터 UDP 패킷을 캡쳐한다. 또한 패킷 집합기는 주기적(예, 10ms)으로 NIC2의 버퍼상태를 확인하여 패킷의 중요도 및 잔여 자원의 정도에 따라 패킷을 복제하여, NIC2로 전송한다. 예를 들어, TCP 패킷의 중요도는 상, UDP 패킷의 중요도는 하로 구분하고, 상의 중요도를 가진 패킷은 0~2개 복제하고, 하의 중요도를 가진 패킷은 0~1개 복제할 수 있다. NIC2의 버퍼 잔여 공간에 따른 패킷 복제 수는 Table 2.와 같이 정의할 수 있다. 이는 하나의 예시이며, 중요도 및 복제 패킷의 수는 관리자가 설정할 수도 있다.

TCP 처리량 조정 기능은 TCP 패킷의 지연전송을 이용하여 TCP 송신지의 최대 TCP 처리량(throughput)을 조절하는 기능으로, 버퍼 오버플로우에 의한 패킷 손실을 완화할 수 있다. 일반적으로 TCP에서 최대 처리량은 다음과 같이 계산할 수 있다[29].

Table 2. The number of packet duplication according to the remaining buffer size

Remaining buffer size	The number of packet duplication
0~10%	TCP:0, UDP:0
10~30%	TCP:1, UDP:0
30~70%	TCP:2, UDP:0
70%~	TCP:2, UDP:1

$$throughput = bufsize/RTT \quad (1)$$

bufsize는 TCP 목적지의 receive window size를 의미하며, RTT는 round trip time을 의미한다. 따라서, 측정된 RTT 및 bufsize를 기반으로 계산된 예상되는 처리량(Th_1)을 의도한 처리량(Th_2)으로 조정하기 위해 패킷 집합기는 해당 TCP 세션의 RTT값을 제어하거나, TCP 목적지 프록시의 receive window size를 변경할 수 있다.

$$a = \frac{bufsize_1}{Th_2} - RTT1 \quad (2)$$

수식(2)는 $Th_1 > Th_2$ 인 경우, Th_1을 Th_2로 조정하기 위해 필요한 추가 RTT값을 나타낸다. $bufsize_1$ 및 RTT_1은 각 기존에 측정된 버퍼 receive window size와 round trip time을 의미하며, Th_2는 조정 후 처리량을 나타낸다. 수식(2)에 따라 기존의 RTT_1에 a만큼의 RTT를 추가하면, Th_2의 처리량으로 낮출수 있다. RTT 제어를 통해서는 처리량을 낮추는 기능만 지원한다. receive window size 조정을 통해 Th_2의 처리량으로 낮추기 위해서는 receive window size를 기존의 $bufsize_1$에서 Th_2*RTT_1으로 변경하면 된다. 이 또한 bufsize의 초기값이 최대치로 잡혀 있는 경우 처리량을 낮추는 기능만 지원한다.

관리자는 미리 각 TCP 세션에 대해 원하는 처리량(Th_2)를 고정으로 할당하여 사용할 수 있다. 관리자는 각 TCP 세션(source IP, source port, destination IP, destination port)별로 처리량을 할당할 수 있다. 이 값은 주기적으로 RED 수신시스템에게 전달될 수 있다. 예를 들어 보자. 관리자가 한 TCP 세션의 처리량을 5Mbps로 설정하였다고 가정하자. 이 때, 해당 세션의 receive window size와 측정된 RTT값이 각각 64Kbytes(65,536bytes) 및 20ms 인 경우 수식(1)에 따라 최대 처리량은 25Mbps로 계산된다. 이를 5Mbps로 줄이기 위해서는 RTT를 100ms로 만들거나, receive window size를 13,107bytes로 변경하면 된다. 하지만, receive window size를 세션이 맺어진 후에 변경하는 것은 데이터 수신에 영향을 미칠수 있다. RTT를 100ms으로 만들기 위해서는 현재의 RTT값 20ms에 추가로 80ms의 지연시간을 주면 된다. 이를 위해 커널 계층에 TCP 패킷 지연 전송 기능에 특정 세션에 대한 지연 시간 값을 알린다.

TCP 패킷 지연 전송 기능은 특정 세션에 대해 RED 송신시스템의 TCP 목적지 프록시가 TCP 송신지로 전송하는 TCP 패킷(커널에서의 TCP ACK 패킷 포함)을 일정 시간 지연시킨 후에 전송하는 기능이다. 이를 이용해 특정 TCP 세션을 관리자가 정의한 처리량으로 설정할 수 있다. 앞의 예시를 활용하면 TCP ACK 패킷 등의 TCP 패킷을 80ms 동안 지연시켰다가 전송을 하면 25Mbps의 처리량을 5Mbps의 처리량으로 제한할 수 있다.

3.1.2 RED 수신시스템

이제 RED 수신시스템에 대해서 살펴보자. RED 수신시스템 NIC1은 Fig.2.의 네트워크2와 연결되어 있으며, NIC2은 RED 송신시스템과 연결되어 있다. 즉, RED 수신시스템은 NIC2을 이용하여 송신시스템으로부터 TCP 또는 UDP 패킷을 수신하고, 이를 처리한 후에 NIC1를 통해서 네트워크2로 전달하는 과정을 수행한다.

RED 수신시스템의 패킷 수집기기능은 RED 송신시스템이 단방향으로 전송하는 패킷을 RED 수신시스템의 NIC2에서 캡쳐하고, 수신한 패킷 중 동일한 패킷을 제거한 다음, TCP 패킷인 경우 적합한 TCP 송신지 프록시로 전달하며 UDP 패킷인 경우 NIC1로 전달하는 기능을 수행한다.

캡쳐되는 패킷은 IP헤더, UDP/TCP 헤더, 응용 데이터를 모두 포함한다. 앞서 RED 송신시스템의 패킷 집합기에서는 RED 송신시스템 NIC2의 버퍼 상태 및 패킷 중요도에 따라 0~2개의 패킷을 복제하여 RED 수신시스템으로 전송한다고 설명하였다. 802.3 기반 NIC인 경우, 패킷 수집기에서 캡쳐되는 패킷은 MAC에서 비트에러가 발생되지 않았다고 판단된 패킷이다. 그러므로, 패킷 수집기에서는 MAC에서 비트에러가 발생하지 않은 동일 패킷이 중복해서 캡쳐되는 상황이 발생할 수도 있다. 따라서, 패킷 수집기에서는 중복 패킷을 걸러내는 필터링 작업이 필요하다. 중복 패킷 필터링 위해서 Table 3.의 알고리즘1을 제안하였다.

알고리즘1에서 패킷 수집기가 새로운 패킷(P_1)을 캡쳐하면, 기존에 캡쳐한 패킷(P_2)과 먼저 패킷 길이를 비교한(라인5~라인7)다. 패킷 길이가 다르다면, P_1과 P_2는 다른 패킷임을 의미하며, 라인14 의

Table 3. Duplicated Packet Filtering Algorithm

```
Algorithm1. Duplicated Packet Filtering
1: Input
2:   P1:a new packet captured in NIC2
3:   P2:an existing packet captured in NIC2
4: Start
5:   if P1 packet length != P2 packet length then
6:     goto post-processing
7:   end if
8:   P3 = P1 XOR P2
9:   P2 = P1
10:  If all bits in P3 are 0 then
11:    delete P1
12:  else goto post-processing
13:  end if
14:  post-processing:
15:    if P1 protocol == UDP
16:      forwarding P1 to NIC1
17:    else if P1 protocol == TCP
18:      forwarding P1 to a suitable TCP sender proxy
19: End
```

'후처리'단계를 수행한다. 패킷 길이가 같을 경우, P_1과 P_2에 대해 XOR(라인8)을 수행하며, 그 결과(P_3)의 비트들 중 한 비트 이상이 0이 아닌 경우 다른 패킷을 의미하며, 라인14 의 '후처리(post-processing)'단계를 수행한다. P_3의 모든 비트들이 모두 0인 경우 이는 완전히 동일한 프레임을 의미하므로, 수신한 패킷 P_1을 삭제하고 종료한다. '후처리'에서는 패킷 P_1의 IP 헤더의 프로토콜 필드가 TCP 혹은 UDP 인지를 식별하여, UDP 인 경우 NIC2로 바로 전달하며, TCP 패킷인 경우 적합한 TCP 송신지 프록시로 전달한다.

TCP 송신지 프록시 기능은 기본적으로 TCP 송신지를 모사한 응용프로그램으로 TCP 목적지에게 마치 자신이 TCP 송신지인 것처럼 판단하게 한다. TCP 목적지 프록시처럼 TCP 기반의 응용프로그램마다 별도의 TCP 송신지 프록시가 구동되며, 이는 기존의 TCP 프록시 기능과 동일하므로 자세한 설명은 생략한다. TCP 송신지 프록시가 패킷 수집기로부터 수신한 패킷 중 로그, DB 등의 파일 데이터는 저장 공간에 임시로 저장한다. 이는 TCP 송신지 프록시가 TCP 송신지의 프로토콜을 모두 알고 있으므로 로그, DB 등의 파일 데이터가 어떤 프로토콜을 사용하는지 파악할 수 있다. 예를 들어, 응용 프로그램에서 파일 데이터를 교환하기 전에 TCP 목적지와 "접속 요청(connection request)" 메시지 및 "접속 승인(connection accept)" 메시지 등을 교환하는

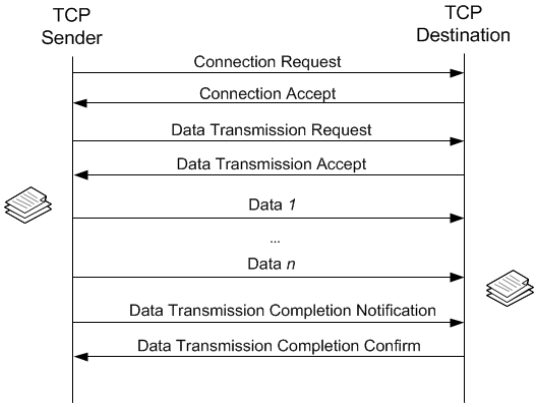

Fig. 5. An example of a file transfer process based on TCP protocol

프로토콜을 사용할 수 있다. TCP 송신지 프록시는 수신한 패킷의 IP 헤더 및 TCP 헤더로부터 여러 정보를 획득하여 TCP 목적지와 TCP 세션을 설립하고, 패킷을 교환한다. 이는 3.2에서 좀 더 상세히 설명한다.

3.2 패킷 전송 과정

3.2.1 TCP 패킷 전송 과정

RED가 적용된 환경에서 TCP 기반의 데이터를 전송할 때, RED에서 패킷이 전송되는 과정에 대해서 살펴본다. TCP 기반의 데이터 파일을 전달할때 Fig.5.와 같은 프로토콜을 사용하여 진행된다고 가정하자. 이때 TCP 계층에서 교환되는 패킷은 그림에 나타내지 않았다. 일반적으로는 TCP 송신지가 접속 요청 메시지를 전송하기 전에 TCP 목적지와의 TCP 세션이 설립될 것이며, 데이터전송 완료 확인을 TCP 송신지가 받은 후, TCP 세션이 종료될 것이다. RED가 적용 될 경우, Fig.5.의 과정은 Fig.6.의 과정으로 진행된다. RED 송신시스템이 TCP 목적지 프록시 역할을 수행하며, TCP 송신지의 여러 응용계층 메시지에 응답한다. 즉, TCP sender가 보내는 접속 요청 메시지에 대한 접속 승인 메시지를 보내며, 데이터전송 요청 메시지에 대한 데이터전송 수락 메시지를 보낸다. 이 과정에서 측정된 TCP 송신지와 TCP 목적지 프록시간의 RTT를 기반으로 RED 송신시스템의 TCP 처리량 조정 기능은 관리자가 설정한 해당 세션의 초당 처리량으로 처리량을 조정하기 위해서 수식(2)을 기반으로 필요

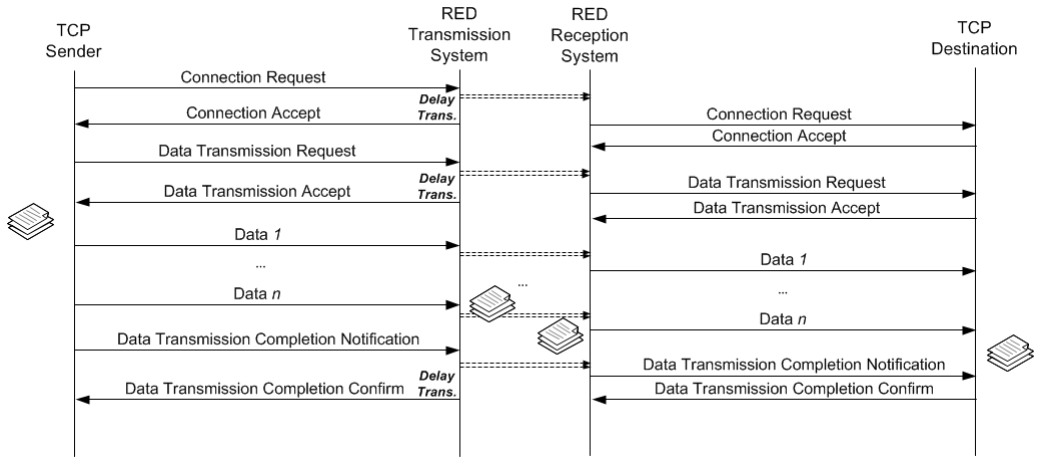

Fig. 6. An example of a file transfer process in an environment applying RED

한 지연시간 값을 계산하고, 이를 TCP 패킷 지연 전송 기능을 이용하여, 패킷을 지연 전송한다. 이는 Fig.6.에서 확인할 수 있으며, RED 송신시스템이 전송하는 TCP ACK 패킷들에 대한 지연 전송은 Fig.6.에 나타내지 않았다.

Fig.6.에서 볼 수 있듯이, RED 송신시스템은 데이터 1부터 데이터 n까지 모두 TCP 송신지로부터 성공적으로 수신하면, TCP 목적지처럼 파일을 임시 공간에 저장할 수 있다. RED 송신시스템은 TCP 송신지로부터 응용 계층 패킷을 수신하면, 수신한 패킷을 RED 수신시스템으로 전달한다. 이 때, RED 송신시스템의 패킷 집합기는 NIC2의 버퍼 상태 및 패킷의 중요도에 따라 0~2개의 패킷을 복제하여 RED 송신시스템의 NIC2를 통해 RED 수신시스템으로 전달하는데, Fig.6.에서는 1개의 패킷을 복제하여 전달하는 것을 확인할 수 있다.

RED 수신시스템의 패킷 수집기는 RED 송신시스템으로부터 1개 이상의 동일한 패킷을 수신할 것이며, 패킷 수집기는 알고리즘1을 기반으로 중복 패킷을 제거한다. RED 수신시스템이 데이터1부터 데이터n까지 성공적으로 수신하면 TCP 목적지처럼 파일을 임시공간에 저장할 수 있다. RED 수신시스템의 TCP 송신지 프록시는 RED 수신시스템의 패킷 수집기로부터 접속 요청패킷을 수신하면, 해당 패킷의 IP헤더 및 TCP헤더로부터 IP목적지주소 및 TCP 목적지 포트(port) 정보등을 획득하고, 이를 기반으로 TCP 목적지에게 TCP 세션 설립을 요청한다. TCP 세션 설립이 완료되기 전까지는 접속 승인 요청 메시지의 전송을 보류하며, 해당 패킷 및 이후 수신한 패킷은 버퍼에 임시 보관한뒤, TCP 세션 설립이 완료되면, TCP 수신시스템의 TCP 송신지 프록시는 접속 요청 메시지를 버퍼에서 추출하여, TCP 목적지로 전달한다. 이후 접속 승인 메시지를 수신하면, 버퍼에 저장된 데이터전송 요청 등의 메시지를 차례로 추출하여 TCP 목적지로 전달한다. TCP 수신시스템에서는 TCP 목적지로의 전송이 성공적으로 완료되면, 임시 저장된 파일을 삭제할 수 있다.

3.2.2 UDP 패킷 전송 과정

RED가 적용된 환경에서 UDP 패킷이 전송되는 과정에 대해서 살펴본다. Fig.1.의 A1이 B1으로 UDP 패킷을 전송한다고 가정하자. A1이 보내는 UDP 패킷은 RED 송신시스템의 패킷 집합기에서 캡쳐된다. 패킷 집합기는 NIC2의 버퍼 상태 및 UDP 패킷의 중요도에 따라 0~1개의 패킷을 복제하여 RED 송신시스템의 NIC2를 통해 RED 수신시스템으로 전달할 수 있다. RED 수신시스템의 패킷 수집기는 동일 패킷이 아닌 경우에, NIC1로 전달한다.

IV. 성능 평가

본 장에서는 RED와 기존 방안에 대해 성능 평가를 한 결과를 기술한다. RED에서는 1) RED 송신시스템에서 TCP 송신지로의 TCP 패킷의 지연 전송과 2) RED 송신시스템에서 패킷의 중복전송을

활용하여 기존의 단방향 시스템에서 발생할 수 있는 패킷 손실 문제를 완화할 수 있다. 각 방안에 대해 기존 방안 대비 얼마만큼의 성능 개선이 나타나는지 살펴본다.

4.1 지연 전송 방안의 성능 개선

RED의 1)을 적용하였을때와 2.3.2에서 기술한 수신시스템에서 TCP 송신지가 보낸 데이터를 모두 저장해 두는 방안에 대한 성능을 비교한다. RED에서처럼 단방향 송신시스템 및 수신시스템이 받은 파일을 바로 포워딩하는 시스템은 버퍼 오버플로우 문제가 발생할 수 있기 때문에 비교 대상에서 제외한다.

이 때, TCP 연결1이 X Mbps의 속도를 지원하고, TCP 연결2가 Y Mbps의 속도를, 송신시스템에서 수신시스템으로의 단방향 링크가 Z Mbps를 각각 지원하며, 전송하려는 데이터 크기는 A Mbits라고 가정한다. 또한 관리자는 RED 송신시스템에서 TCP 연결1의 속도 제한을 B Mbps (B ≤ 10)로 설정했다고 가정한다. 또한 관리자는 TCP 연결1의 속도 제한을 정확한 값으로 결정할 수 있다고 가정한다. 즉, 관리자는 X〉10인 경우 TCP 연결1의 속도를 10으로 제한 한다. Fig.7.은 A=500, Y=10, 및 Z=1000인 환경에서, X(TCP 연결1 처리량)의 속도변화 (1 ~ 20)에 따른 기존의 방안과 RED에서의 파일 전송 소요 시간은 다음과 같다. 이 때, RED 수신시스템이 TCP 송신지가 보내는 첫 번째 패킷을(Fig.6.의 경우는 접속 요청) 수신하고 TCP 목적지와 TCP 세션을 설립하는데 까지 0.5초가 걸린다고 가정하였다. Fig.7.에서 볼 수 있듯이, TCP 연결1의 처리량에 관계없이 RED의 전송 시간이 줄어드는 것을 확인할 수 있다.

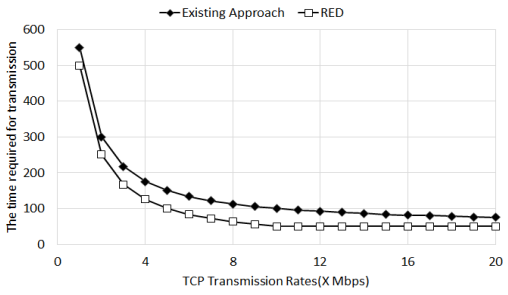

Fig. 7. Comparison of the time required for transmission according to the various X(TCP transmission rates)

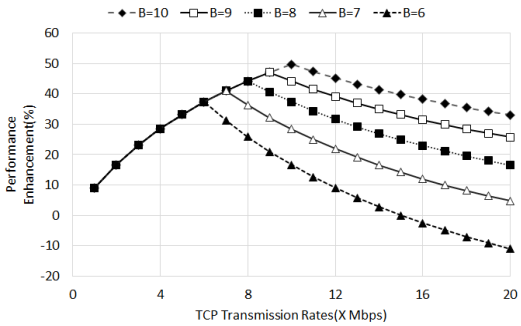

Fig. 8. Comparison of Performance Enhancement in RED according to the B

Fig.8.은 A=500, Y=10, 및 Z=1000인 환경에서 B의 값에 따라 기존 방안 대비 RED의 성능 개선 정도(전송 소요시간)를 나타내는 그림이다. B=10인 전송 소요시간이 기존 방안 대비 9% ~ 50% 정도 단축되는 것(성능향상)을 확인할 수 있다. TCP 연결1의 속도 제한이 동작을 하지 않는 X ≥B 인 환경에서는 X의 값이 증가할 때, RED의 성능향상 정도가 좋아진다. B가 8인 경우, X가 1일 때 RED는 기존 방안 대비 9%의 성능향상을 X가 6일때는 37%의 성능향상을 각각 보여준다. 관리자가 B를 너무 낮은 값으로 설정할 때, X가 클 경우 되려 기존 방안보다 성능이 떨어질 수 있다. 예를 들어, B가 6 이고 X가 20일 때, 성능은 -11%로 기존방안보다 좋지 않다. X가 커질수록 더 나빠지는데, 그림에서는 나타내지 않았다. 이를 통해 적절한 B값을 설정하는 것이 중요함을 확인할 수 있다.

4.2 중복 전송 방안의 성능 개선

RED의 2)를 적용하였을 때와 단일 전송을 하는 경우에 대한 성능을 비교한다. 본 제안 방안은 FEC[19-21] 및 패킷 결합[22-25]과 함께 사용될 수 있으므로, 해당 방안들과는 비교하지 않는다.

BER(bit error rate)가 b이고 패킷 크기가 p bytes 일 때, PER은 수식(3)과 같이 표현될 수 있다.

$$PER = 1-(1-b)^{8p} \quad (3)$$

따라서, RED의 n (n≥0)번 중복전송하는 방안을 사용할 경우, RED 수신시스템 패킷 수집기에서의 최종 PER(fPER)은 수식(4)와 같이 계산될 수 있다.

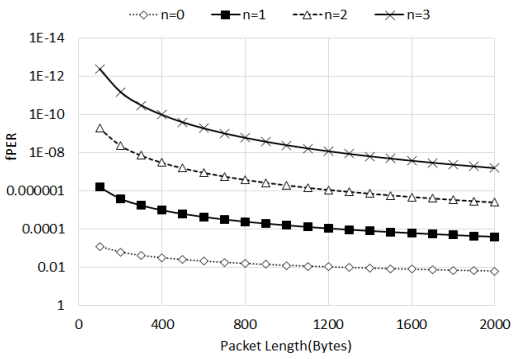

Fig. 9. fPER according to the degree of duplicated transmission(n value) in various packet length (RED reception system)

$$fPER = PER^{n+1} = (1-(1-b)^{8p})^{n+1} \quad (4)$$

수식 (4)를 기반으로, RED의 송신 시스템이 0회에서 3회까지 중복 전송을 할 때, RED 수신시스템의 패킷 수집기에서의 최종 PER(fPER)은 Fig.9.와 같다. n=0일때는 RED는 기존의 단방향 시스템과 같다고 볼 수 있다. 실험환경은 b는 1e-6이며, p는 100 bytes에서 2000 bytes로 100bytes씩 증가하는 환경이다. 패킷 크기가 증가할수록 fPER이 커지며, 중복 전송 횟수가 커질수록 fPER이 작아지는 것을 그림에서 확인할 수 있다. 예를 들어, n=0이고 p=500일 때 fPER은 0.004이나, p=2000일 때 fPER은 0.016으로 p=500일 때보다 거의 4배가 높다.중복전송의 효과를 예로 들면 다음과 같다. p=1000이고 n=0일 때 fPER은 0.008이며, n=3일 때 fPER은 4.03e-9로, n=0일 때와 비교하여 5.06e-7만큼 fPER이 낮아진다. 즉, n이 커질수록 RED의 패킷 복원 효과는 커지는 것을 확인할 수 있다.

4.3 고려 사항

성능평가 4.1과 4.2에서 살펴보았듯이, RED는 지연 전송 및 중복 전송을 통해 성능이 개선될 수 있다. 중복 전송은 중복전송 정도에 따라 항상 PER이 개선되지만, 지연 전송은 Table 4.에 기술된 것처럼 환경에 따라 개선여부가 다르게 나타난다. RED는 X 〉 Y 이고 B 〈 Y 인 경우를 제외하고는 기존 방안(TCP 송신지가 보낸 데이터를 모두 저장해 두는 방안) 대비 모두 전송 소요시간을 단축할 수 있다. 특히, X 〉 Y 이고 B 〈 Y인 경우도 Fig.8.에서 볼 수 있듯이, B와 X의 차이가 크지 않을 경우 여전히 RED가 성능이 좋음을 확인할 수 있다. 예를 들어, B가 6이고 X가 12일 때, RED는 기존 방안 보다 전송시간이 약 10% 단축된다. 하지만 B가 6이고 X가 20일 때, RED는 기존 방안보다 전송시간이 약 11% 늘어난다.

따라서, RED의 지연 전송을 통한 전송시간이 증가되는 문제를 완화하기 위해 Table 5.와 같이 RED와 기존 방안을 혼합하여 사용할 수도 있다.

Table 4. Performance enhancement of RED compared with existing approach according to the environments

environments		performance enhancement
X 〉 Y	B 〉 Y	Always(but may occur buffer overflow at the RED reception system)
	B = Y	Always
	B 〈 Y	Conditionally (RED or Existing approach)
X ≤ Y		Always

Table 5. Performance enhancement of RED compared with existing approach according to the environments

environments	Application
X 〉 Y	if Y could be measured exactly, RED
	otherwise, Existing Approach
X ≤ Y	RED

V. 결 론

본 논문에서는 단방향 전송 기술에서 비트 에러 및 버퍼 오버플로우에 의해 발생할 수 있는 패킷 손실 문제를 완화하기 위해 신뢰성 있는 단방향 전송 시스템(RED)를 제안하였다. RED는 비트 에러에 의한 패킷 손실을 줄이기 위해, 패킷의 중요도 및 여유 자원을 고려하여 동일 패킷의 중복 전송을 활용하였다. 또한 RED는 버퍼 오버플로우에 의한 손실을 줄이기 위해, RED 송신시스템에서 TCP 패킷의 지연 전송을 활용하였다. 성능 평가를 통해 지연전송이

기존 방안과 동일하게 버퍼 오버플로우에 의한 손실이 발생하지 않지만, 전송 소요시간을 단축할 수 있음을 확인하였다. 향후, RED를 구현하여, 테스트베드 환경에서 성능평가를 수행할 예정이다. 또한, 멀티 TCP 세션이 있는 환경에서 정확한 Y값을 측정할 수 방안에 대해서도 연구할 예정이다.

References

[1] Wiki: Unidirectional Network, http://en. wikipedia.org/wiki/Unidirectional_network

[2] M.H. Kang, I. Moskowitz, and S.Chincheck. "The pump: A Decade of Covert Fun," Proceedings of 21st Annual Computer Security Applications Conference, pp. 352-360, Dec. 2005

[3] David M. Goldechlag, "Several Secure Store and Forward Devices," Proceedings of the Third ACM Conference on Computer and Communications security, pp. 129-137, 1996

[4] D. G. Gomez, "Receive-only UTP cables and Network Taps," http://www.infosecwriters.com, 2004

[5] Waterfall One-Way. [Online]. Available: http://www.waterfallsecurity.com

[6] Dual diode. [Online]. Available: http://www.owlcti.com

[7] H. Okhravi and F. T. Sheldon, "Data diodes in support of trustworthy cyber infrastructure," in Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW), pp. 231-234, Apr. 2010

[8] R. Mraz and J. Hope, "Concurrent Data Transfer Involving Two or More Transport Protocols Over a Single One-Way Data Link," US Patent No. 8,139,581, Mar. 21, 2012

[9] C. A. Nilsen, "Method for Transferring Data from an Unsecured Computer to a Secured Computer," US Patent No. 5,703,562, Dec. 30, 1997

[10] Lin Honggang, "Research on Packet Loss Issues in Unidirectional Transmission," Journal of Computers, vol. 8, no. 10, pp. 2664-2671, Oct. 2013

[11] J. Menoher, "All Data Diodes Are Not Equal", White Paper, Sep. 2013

[12] Fox-IT, "Fox DataDiode: A Pereferred Solution for high-security real-time electronic unidirectional data transfer between networks," White paper, Jan. 2008

[13] Y. Heo, B. Kim, D. Kang, S. Shon, and J. Na, "A Design of Unidirectional Security Gateway for Enforcement Security and Reliability for Transfer Data," The Korean Institute of Communications and Information Sciences, pp827-828, Jan. 2016

[14] K. Kim, J. Yun, H. Kim, M. Jung, W. Kim, E. Park, and S. Park, "Physical One Direction Communication Device and Method Thereof," Korea Patent No. 10-1593168, Feb. 02, 2016

[15] IEEE 802.3-2012. The Standard for Ethernet, Dec. 2012

[16] IEEE 802.11-2007. IEEE Standard-part 11: wireless LAN medium access control and physical layer specifications, Jun. 2007

[17] Jon Postel, "Transmission Control Protocol," RFC 793, 1981

[18] M. Allman, V. Paxson, and W. Stevens, "TCP Congestion Control," RFC 2581, 1999

[19] M. Elaoud and P. Ramanathan, "Adaptive Use of Error-Correcting Codes for Real-time Communication in Wireless Networks," Proceedings of IEEE Infocom'98, pp. 548-555, Mar. 1998

[20] J.S. Ahn and J. Heidemann, "An adaptive FEC algorithm for mobile wireless networks," The KIPS Transactions, vol9C, no. 4 pp. 563-572, Aug. 2002

[21] S. Choi, Y. Choi, and I. Lee, "IEEE 802.11 MAC-level FEC scheme with retrans-

mission combining," IEEE Transactions on Wireless Communications, vol. 5, no. 1, pp. 203-211, Jan. 2006
〔22〕 S. S. Chakraborty, E. Yli-Juuti, and M. Liinaharja, "An ARQ scheme with packet combining," IEEE Communications Letters, vol. 2, no. 7, pp. 200-202, Jul. 1998
〔23〕 S. S. Chakraborty, M. Liinaharja, and E. Yli-Juuti, "An adaptive ARQ scheme with packet combining," IEEE Communications Letters, vol.3, no.2, pp. 52-53, Feb. 1999
〔24〕 H. Dubois-Ferriere, D. Estrin, and M. Vetterli, "Packet combining in sensor networks," Proceedings of ACM SenSys, pp. 102-115, Nov. 2005
〔25〕 P. S. Sindhu, "Retransmission error control with memory," IEEE Transactions on Communications, vol. COM-25, no.5, pp.473-479, May. 1977
〔26〕 A. Bakre and B. Badrinath, "I-TCP: Indirect TCP for mobile hosts," Proceedings of 15th International Conference on Distributed Computing Systems (ICDCS), pp. 136-143, May. 1995
〔27〕 R. Cohen and S. Ramanathn, "Using proxies to enhance TCP performance over hybrid fiber coaxial networks," Elsevier Computer Communications, vol. 20, pp. 1502-1518, Jan. 1998
〔28〕 H. Jiang and C. Dovrolis, "Passive Estimation of TCP Round-Trip Times," ACM Computer Communication Review, vol. 32, no. 3, pp. 75-88, Aug. 2002
〔29〕 B. Constantine, G. Forget, R. Geib, and R. Schrage, "Framework for TCP Throughput," RFC 6349, 2011

〈저자소개〉

사 진

김 동 욱 (Dongwook Kim) 정회원
2005년 2월: 경북대학교 컴퓨터공학과 학사
2012년 2월: 포항공과대학교 컴퓨터공학과 박사
2012년 4월~현재: ETRI 부설연구소 선임연구원
〈관심분야〉 통신공학, 제어시스템 보안, 정보보호

사 진

민 병 길 (Byunggil Min) 정회원
2002년 2월: 충북대학교 컴퓨터공학과 학사
2004년 2월: 포항공과대학교 컴퓨터공학과 석사
2004년 3월~현재: ETRI 부설연구소 선임연구원
〈관심분야〉 제어시스템 보안, 침입탐지 시스템, 취약성 분석