

# 상용 NIC 기반 단방향 통신 방법에 관한 연구\*

김진홍,<sup>1\*</sup> 나중찬,<sup>2</sup> 이성현<sup>2\*</sup><sup>1</sup>과학기술연합대학원대학교, <sup>2</sup>한국전자통신연구원

## A Study on One-way Communication using Commercial Off The Shelf Network Interface Card\*

Jin-hong Kim,<sup>1\*</sup> Jung-chan Na,<sup>2</sup> Seoung-Hyeon Lee<sup>2\*</sup><sup>1</sup>Korea University of Science and Technology,<sup>2</sup>Electronics and Telecommunications Research Institute

### 요 약

상용 NIC 기반 단방향 통신 방법은 저비용으로 고속 단방향 통신을 지원한다는 점에서 장점을 갖는다. 본 논문에서는 보안 강화를 위한 단방향 통신 방법을 소개하고, 상용 NIC 디바이스 드라이버의 수정을 통해서 단방향 통신을 제공할 수 있는 기법을 제시한다. 그리고 상용 NIC 기반의 단방향 통신 방법의 활용을 확인하기 위하여 인텔 82580 NIC 예제 구현을 제시하고 성능과 신뢰성 측정실험을 통해 단방향 통신 방법 실현에 기여할 수 있음을 보인다.

### ABSTRACT

Commercial Off The Shelf(COTS) based one-way communication is advantageous in that support a low cost communication and high speed one-way communication. This paper introduce a one-way communication method, and provides a implementation method of one-way communication through modified device driver for COTS NIC. Then, To verify the advantage of the COTS based one-way communication method, We present a sample implementation using Intel 82580 NIC, and present a possibility that can contribute to the realization of one-way communication through experiments on performance and reliability.

**Keywords:** One-way Communication, Data Diode, Unidirectional network, DPDK, Device Driver

## 1. 서 론

국가 공공기관 망 분리 사업은 해킹 등 사이버 공격으로부터 국가기밀 등 중요자료의 유출을 근본적으로 차단하기 위해 도입되었다. 보안등급이 높은 영역을 보안등급이 낮은 영역과 분리하면 업무관련 중요 정보에 접근하는 것이 차단되므로 안전한 업무 환경을 구축할 수 있다[1][2].

이렇게 보안등급이 다른 두 영역에서의 데이터 연계를 위한 기술로 물리적 단방향 통신과 방화벽이 있다. 하지만 방화벽의 경우, 특정 서비스에 대한 허용 및 차단 정책을 적용하기 때문에 우회 공격이 가능한 상황에서 외부위협에 대한 원천봉쇄가 불가능하기 때문에, 최근에는 물리적인 단방향 통신 기술의 적용을 권고하고 있다[3].

또한, 기존의 국내 단방향 통신 시스템은 사전 정의된 소켓 기반 데이터 스트리밍 중계 방식과 공유 스토리지를 이용한 중계 방식을 필드에 적용하고 있으나, 중계 데이터 유형과 중계 경로 구현 방식에 따라 보안 취약점이 존재하여 이에 대한 대응방안이 추가되어야 한다[4].

본 논문에서는 상용 NIC의 동작을 제어하는 레지

Received(09. 01. 2016). Accepted(09. 28. 2016)

\* 본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보보호핵심원천기술개발사업[R0126-15-1095, 사이버·물리시스템에서의 물리적 단방향 보안 게이트웨이 개발]의 일환으로 수행하였습니다.

† 주저자, [jhkim3624@naver.com](mailto:jhkim3624@naver.com)

‡ 교신저자, [duribun2@etri.re.kr](mailto:duribun2@etri.re.kr)(Corresponding author)

스터 수정을 통해 보안성을 강화시키고, 단방향 통신 성능을 향상시키기 위한 DPDK(Data Plain Development Kit) 단방향 통신 기법을 제시한다. 본 논문의 구성은 2장에서 단방향 통신 기술을 설명하고, 3장에서 레지스터 수정을 이용한 DPDK 단방향 통신 기법의 설계와 구현 결과를 제시한다. 4장에서 본 논문에서 제시한 기법을 적용한 성능 향상 결과를 분석하고, 마지막으로 5장에서 결론과 향후 연구 방향을 제시한다.

## II. 관련 연구

### 2.1 네트워크 디바이스 드라이버

#### 2.1.1 커널 기반 패킷 처리

커널 기반 패킷 처리는 네트워크 디바이스 드라이버가 필요한데 이는 LKM(Loadable Kernel Module)형태로 컴파일되며, 문자나 블록 디바이스 드라이버가 파일 오퍼레이션 구조체 형식으로 처리되는 것과 달리 네트워크 오퍼레이션 구조체라는 *net\_device*를 사용한다. 또한 네트워크 디바이스 드라이버는 커널 부팅 과정이나 커널에 삽입하는 과정에서 모듈을 커널에 등록하기 위해 *module\_init*을 호출하게 되는데, 이 과정에서 *pci\_device\_id*를 참고하여 하드웨어의 존재 여부를 확인하게 된다. 이후 *probe*라는 디바이스 초기화 루틴을 실행하여 *net\_device* 구조체를 할당하게 되고 이 구조체의 설정을 통해 기본적인 하드웨어를 초기화 하게 된다. 모든 초기화가 끝나면 하드웨어 제어에 필요한 함수 필드를 커널에 등록하게 되고 상위 계층에서 디바이스 드라이버에 등록되어 있는 함수를 호출하여 사용할 수 있게 된다.

리눅스가 정상적으로 수행되기 시작하면 리눅스 초기화 스크립트는 *ifconfig* 명령을 이용해 하드웨어를 검출해, 커널에 등록된 네트워크 인터페이스를 활성화 시킨다. 그리고 네트워크 인터페이스가 활성화 될 때는 디바이스 열기를 담당하는 함수를 호출하고, 네트워크 환경에 맞추기 위한 디바이스 드라이버의 제어 상태를 변경한다.

#### 2.1.2 DPDK 기반 패킷 처리

DPDK는 Intel을 중심으로 고성능 네트워크 장

비를 개발하는 몇몇 회사들(WindRiver, 6Wind 등)이 함께 오픈소스로 개발하고 있는 고성능 네트워크 패킷 처리를 위한 입출력 드라이버 및 라이브러리 모음집이다[5].

드라이버뿐만 아니라, 고속 패킷 처리를 구현할 때 유용하게 쓸 수 있는 x86 계열 CPU에 최적화된 메모리 관리 및 데이터구조 라이브러리들도 다수 포함하고 있다. 비슷한 목적과 구조를 가지는 경쟁자로는 PF RING ZC(Zero-Copy), netmap 등이 있다[6][7].

단점으로는 여러 사용자 애플리케이션(프로세스)가 하나의 네트워크 인터페이스를 서로 공유하기 어렵고, 기존의 소켓 API를 사용하는 네트워크 애플리케이션들은 모두 DPDK의 자체 API(Application Programming Interface)를 사용하도록 코드를 변경하고 TCP/IP에 해당하는 기능을 직접 만들어 써야 그 성능 이득을 누릴 수 있다는 점이다. DPDK 자체적으로 패킷 처리 파이프라인을 보다 쉽게 작성할 수 있도록 추가적인 API를 제공하고자 하는 시도가 있다. 또한, 기존 운영체제에 비해 드라이버 종류가 적어 사용할 수 있는 네트워크 카드 모델이 제한된다는 것도 고려해야 한다.

### 2.2 기존 단방향 통신 방법 분류

단방향 통신기술은 물리적 전송 매체에 따라 UTP(Unshielded Twisted Pair) 케이블을 이용한 방식과 광섬유 케이블을 이용한 방식으로 나눌 수 있다. UTP 케이블을 이용한 구현 방식은 Kyung-Pyo Hong[8], Diego Gonxalez Gomez[9]등의 논문에서 찾아 볼 수 있으며, 단방향 통신 기술의 초기 버전이라고 할 수 있다. 하지만, 최근 고속 전송에 사용되는 NIC은 UTP 케이블의 감쇠와 간섭 현상 때문에 대부분 광섬유 케이블을 사용하고 있다.

다음으로, 구현 방식에 따라 물리적 연결선의 되먹임을 이용한 구현 방식과 상용 NIC에서 디바이스 드라이버 수정을 통한 구현 방식, 그리고 단방향 통신 전용 NIC을 만드는 방식으로 구분 할 수 있다.

첫 번째로 물리적 연결선의 되먹임을 이용한 구현 방식은 Lin[10]의 논문에서 사용한 방식으로 상용 네트워크 인터페이스 카드를 장착 한 뒤 반송파 손실 상태를 극복하고자 Fig.1 과 같이 구현 하였다. 장 점으로는 구현이 쉽고 비용이 적게 들며 범용성이 뛰

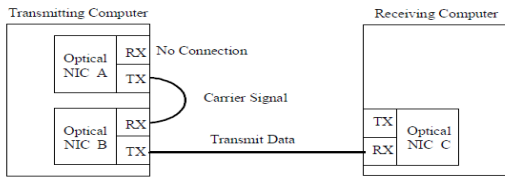


Fig. 1. Unidirectional transmission using carrier signal feedback

어나다는 점이지만, 반송파의 되먹임을 위한 라인을 정상적으로 연결하면 양방향 통신이 된다는 점에서 100% 완벽하다고 할 수 없다.

두 번째로 단방향 통신용 NIC을 만드는 구현 방법[11]은 하드웨어를 직접 구현하기 때문에 Fig. 2와 같다. 장점으로는 전송 NIC의 수신기와 수신 NIC의 송신기를 물리적으로 제거 할 수 있으며 디바이스 드라이버와 NIC을 교체하지 않는 이상 단방향 통신이 가능하다는 점이다. 하지만 단방향 통신용 하드웨어를 직접 구현하기 위한 비용이 많이 들어가고 범용성이 떨어진다는 단점이 있다.

마지막으로 상용 NIC에서 디바이스 드라이버 수정을 통한 구현 방법은 본 논문에서 제안하는 통신 방식으로 상용 NIC을 장착하고 리눅스 커널의 디바이스 드라이버를 수정하여 단방향 통신 기술을 구현하는 방식으로 Fig. 3와 같다. 장점으로는 비용이 적게 들고 범용성이 뛰어나며 NIC의 리시버가 작동되지 않는다는 점에서 Tx, Rx 선을 모두 연결 하더라도 단방향 통신이 가능하다는 장점이 있다. 하지만

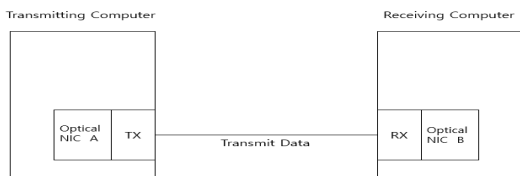


Fig. 2. Unidirectional transmission using exclusive network interface card



Fig. 3. Unidirectional transmission using modified device driver

공격자가 OS에 침입하였을 경우 정상적인 디바이스 드라이버를 올리고 케이블을 연결 하면 양방향 통신이 가능하다는 단점이 있다.

### III. 상용 NIC 기반의 단방향 통신 방법 설계

본 논문에서는 범용성이 좋고 구현이 용이하며, 저비용으로 고속의 단방향 통신을 할 수 있는 방법인 상용 NIC에서의 단방향 통신 방법을 중심으로 범용성이 뛰어난 LKM을 사용한 구현방법과 범용성이 조금은 떨어지지만 데이터센터등에서 고속 통신에 초점을 둔 DPDK 라이브러리를 사용한 구현 방법의 실험을 통해 단방향 통신 기법 실현에 기여할 수 있음을 보인다.

먼저, 구현을 위해 사용된 NIC은 Intel 82580 Gigabit Ethernet Controller를 사용하며, SerDes(Serializer/Deserializer) 인터페이스를 사용하는 4포트 카드이다. 또한 igb 디바이스 드라이버를 사용하여 상위계층의 프로토콜 스택과 연결된다. 사용된 카드의 구조는 Fig. 4와 같이 구성되어 있다.

Intel 82580 계열의 NIC을 선택한 이유는 이 장에서 설명된 레지스터들이 읽기/쓰기 권한을 가지고 있어 레지스터 값의 수정이 가능하며, DPDK를 지원하는 NIC이기 때문이다[5][12].

물리적 절체를 통한 단방향 통신 방법은 통신 케이블의 한쪽 라인을 제거하여 물리적으로 통신 케이블을 단절 시켜 구현하게 되는데, 물리적으로 통신 케이블이 단절되면 NIC의 링크 상태는 링크가 끊어졌다는 disable 상태가 된다. 이 상태는 NIC의 레

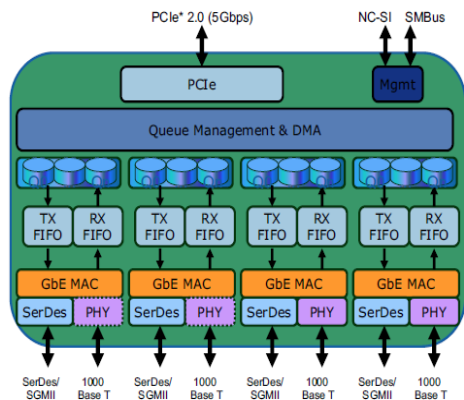


Fig. 4. Intel 82580 GbE Controller

Table 1. Registers about communication link

Offset	Abbreviation	Name	RW
General			
0x0000	CTRL	Device Control Register	RW
0x0018	CTRL_EXT	Extended Device Control Register	RW
Receive			
0x0100	RCTL	RX Control Register	RW
PCS			
0x4208	PCS_LCTL	PCS Link Control Register	RW

지스터 상태를 바꾸고, 그 영향으로 데이터 전송을 위한 드라이버의 코드가 수행되지 않게 된다. 이런 현상을 방지하기 위해 Kyung-Pyo Hong[8], Diego Gonxalez Gomez[9]의 논문에서는 UTP 케이블의 핀(pin) 구성을 바꿔 하드웨어적 방법을 통해 반송과 손실 상태를 극복하여 링크 상태를 유지하도록 만들었다.

그러나, 본 논문에서는 소프트웨어적인 방법으로 NIC의 동작을 제어하는 레지스터를 수정하여 링크 상태를 enable 시키게 된다.

### 3.1 단방향 통신을 위한 레지스터 설정

단방향 통신 환경에서 링크 끊김 현상을 극복하기 위한 소프트웨어적인 방법으로 디바이스 검출과 초기화 루틴이 완료되는 시점에 NIC의 레지스터를 수정하면 링크를 유지할 수 있다. 링크 상태 변경을 위해 참고 되어야 하는 레지스터는 Table 1과 같다.

본 논문에서는 상용 NIC에서의 단방향 통신 방법을 중심으로 범용성이 뛰어난 LKM을 사용한 구현 방법과 범용성이 조금은 떨어지지만 데이터센터등에서 고속 통신에 초점을 둔 DPDK 라이브러리를 사용한 구현 방법을 설명한다.

이를 위해 상용 디바이스 드라이버와 데이터 전송 DPDK 애플리케이션에 공통적으로 다음 두 가지 기능을 추가하여 LKM 버전과 DPDK 버전의 단방향 통신 플랫폼을 구현할 것이다.

#### 3.1.1 물리적 단방향 통신을 위한 링크 상태 활성화

먼저, 단방향 통신을 위해선 네트워크 링크 상태를 변경 시켜줘야 하는데 이를 위해 기본적으로 양방향 통신에서 사용되는 자동협상(Auto-Negotiation)기능을 정지시키고 정적으로 설정을 해 주어야 한다.

자동협상이란 상대방의 전송 속도나 통신 방식을

모른 채로 상대 장비와의 통신을 위한 준비를 자동으로 맞추어 주는 기술을 말하며, 요즘 개발되는 네트워크 장비의 기본적인 사양이 되고 있으며 커넥터에 케이블을 연결하면 자동으로 속도와 통신 방식을 맞추어 설정이 된다.

자동협상에서 가장 중요하게 다루는 것이 바로 속도와 통신 방식(Half duplex or Full duplex)인데, 속도의 경우에는 오늘날 Gigabit Ethernet의 상용화로 10/100/1000Mbps의 세 가지 방식이 존재하고 통신 방식은 Half duplex와 Full duplex의 두 가지 방식이 있다. 또한, Duplex와 속도와의 관계를 따진다면 10/100Mb/s 모드의 인터페이스에서는 반이중 모드와 전이중 모드가 가능하다. 그러나 1000Mb/s 연결에서는 전이중 모드만 가능하다.

전이중 모드는 전송과 수신이 독립적으로 구성되어 있어 패킷 송신과 수신이 동시에 일어날 수 있다. 전송이 시작하면 TX\_DATA 버스를 통해 유효한 데이터가 MAC에서 PHY로 전달되고, 반대로 유효한 수신 데이터가 MAC으로 가는 RX\_DATA 라인에 존재한다는 RX\_DV 신호가 발생되면서 수신이 진행된다.

본 논문에서 예제로 구현한 플랫폼은 1000Mb/s의 속도와 전이중방식으로 정적 설정하여 사용한다.

#### 3.1.1.1 이중통신방식과 링크속도의 설정

단방향 통신을 위해서는 아래와 같이 인터페이스 종류, 속도와 통신 방식 등의 정적 변수 레지스터를 변경해야 한다.

이를 위해 Device Control Register와 PCS(Physical Coding Sublayer) Link Control Register, Extended Device Control Register를 수정한다.

- Device Control Register

디바이스의 주요 작동 모드를 제어하는 32bit 길

이의 값을 갖는다. 디바이스의 속도, 통신 방식과 흐름제어를 변경하기 위해, DPDK 등을 이용하여 값을 재할당할 경우 Duplex, Speed와 같은 특징 bit가 변경된다. 본 논문에서 제시한 기법을 적용하기 위해 수정되어야 할 값은 Table 2와 같다.

디바이스에서 불러온 Device Control Register값에서 음영이 들어가 있지 않은 부분은 OR을, 음영이 들어간 부분은 And Not을 취하여 디바이스에 Fig. 5와 같은 값을 적용할 수 있다.

Table 2. Device Control Register setting list

Define	offset
E1000_CTRL_SPD_1000	0x0200
E1000_CTRL_FRCPSPD	0x0800
E1000_CTRL_SLU	0x0040
E1000_CTRL_FD	0x0001
E1000_CTRL_FRCDPX	0x1000
E1000_CTRL_ILOS	0x0080
E1000_CTRL_RFCE	0x8000000
E1000_CTRL_TFCE	0x10000000

```
CTRL
Before : 0000 1000 1001 0000 0001 1010 1100 0001
After  : 0000 0000 1001 0000 0001 1010 0100 0001
```

Fig. 5. Initial value and modified register value

• PCS Link Control Register

이 레지스터는 물리계층의 SerDes, SGMII(Serial Gigabit Media-Independent Interface)와 1000BASE-KX PCS 인터페이스에서 링크와 관련된 부분을 제어하기 위해 사용된다. 본 논문에서 사

```
PCS_LCTL
Before : 0000 0010 0001 0100 0000 0000 0000 1100
After  : 0000 0000 0000 0000 0000 0000 1011 1101
```

Fig. 6. Initial register value and modified register value

Table 3. PCS Link Control Register setting list

Define	offset
E1000_PCS_LCTL_FLV_LINK_UP	0x0001
E1000_PCS_LCTL_FSV_1000	0x0004
E1000_PCS_LCTL_FDV_FULL	0x0008
E1000_PCS_LCTL_FSD	0x0010
E1000_PCS_LCTL_FORCE_LINK	0x0020
E1000_PCS_LCTL_FORCE_FCTRL	0x0080
E1000_PCS_LCTL_AN_RESTART	0x20000

용된 NIC은 SerDes 인터페이스를 사용하기 때문에 PCS Link Control Register의 링크와 속도, 이중 통신 방식을 조절하는 레지스터를 수정한다. 수정되어야 할 값은 Table 3과 같다.

3.1.1.2 사용된 NIC의 인터페이스 정의

본 논문에서 사용된 NIC은 SFP 모듈을 연결하여 사용되는 SerDes 인터페이스이므로 Extended Control Register의 Link.MODE bit를 SerDes라고 정의 해주어야 한다. 마찬가지로 수정되어야 할 값은 Table 4와 같다.

Table 4. Extended Device Control Register setting list

Define	offset
E1000_CTRL_EXT_LINK_MODE_PCIE_SERDES	0xC00000

```
CTRL_EXT
Before : 0000 0010 1101 0100 0000 1000 1000 0000
After  : 0000 0010 1101 0100 0000 1000 1000 0000
```

Fig. 7. Initial register value and modified register value

3.1.2 단방향 통신 장치의 수신 포트 비활성화

물리적 거리를 통한 단방향 통신시 사용자의 부주의로 인해 Tx/Rx 포트를 모두 연결 시켰을 경우를 고려하여, 전송하는 주체의 수신기를 정지시켜 패킷이 네트워크 계층으로 올라가지 않게 막아 주어야 한다. 이를 위해 RX Control Register의 내용을 수정하여 수신된 패킷이 즉시 폐기 될 수 있도록 만든다.

3.1.2.1 RX Control Register

이 레지스터는 82580 컨트롤러의 모든 수신 기능을 제어하며 본 단방향 플랫폼에서는 Receiver를

```
RCTL
Before : 0000 0000 0000 0000 0000 0000 0000 0000
After  : 0000 0000 0000 0000 0000 0000 1100 0000
```

Fig. 8. Initial register value and modified register value

Table 5. Receive Control Register setting list

Define	offset
E1000_RCTL_RXEN	0x02

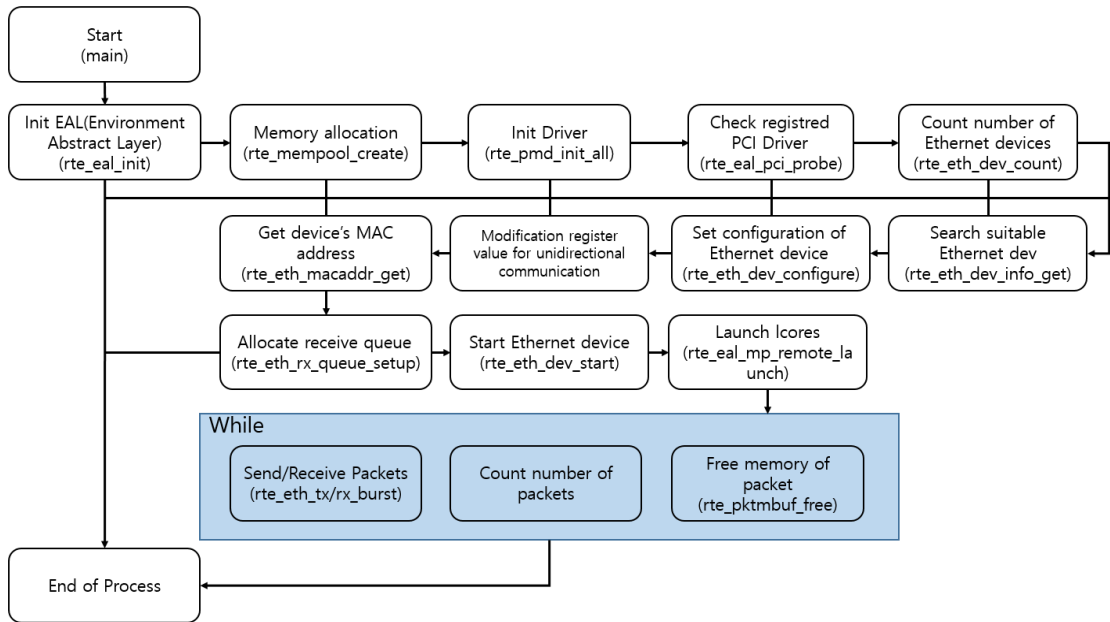


Fig. 9. Flow of execution in DPDK one-way communication application

활성화 시키는 bit를 AND NOT 연산으로 수정하여 적용한다.

## 3.2 설 계

### 3.2.1 LKM 기반의 단방향 통신 방법 설계

LKM 기반의 단방향 통신 방법을 구현하기 위해 기존의 igb 드라이버를 수정하여 사용한다. 수정은 디바이스 드라이버의 코드 중 커널에서 NIC을 검출하고 초기화 시키는 루틴이 완료되는 시점에 3.1에서 언급한 17개의 레지스터를 변경 시켜주면 된다.

### 3.2.2 DPDK 기반의 단방향 통신 방법 설계

DPDK 기반의 단방향 통신은 커널의 네트워크 스택을 사용하지 않기 때문에, PCI 검출과 데이터 전송에 필요한 네트워크 기술들을 DPDK API를 이용하여 추가로 구현해야 한다. 또한, DPDK 응용 프로그램의 실행을 위해서 커널에서 사용하는 기존의 igb 드라이버를 제거하고, DPDK API의 igb\_uio 드라이버를 설치하여 실행시켜야 한다.

본 논문에서 제시한 기법을 적용한 단방향 통신 절차는 Fig. 9와 같다.

또한, 리눅스 시스템 내에서 LKM 기반의 단방향 통신방법과 DPDK 기반의 단방향 통신방법의 데이터 이동 방향을 비교해보면 Fig. 10과 같다. LKM 기반의 단방향 통신과 달리 DPDK기반의 단방향 통신 방법은 리눅스 커널의 네트워크 스택을 거치지 않고 User Space에서 미리 생성되어 Network Hardware로 직접 복사 된다는 차이점이 있다.

## IV. 실 험

이 장에서는 이전에 언급한 상용 NIC 기반의 단방향 통신 방법을 활용할 수 있음을 확인하기 위해 실제 단방향 파일 전송을 통해 데이터 신뢰성과 성능 평가를 수행하였다.

### 4.1 실험 환경

단방향 통신은 Fig. 11.과 같이 디자인 된 플랫폼을 사용한다. 전송 컴퓨터와 수신 컴퓨터에 각각 1개의 NIC을 장착하며, 단방향 통신을 위한 LKM과 DPDK 애플리케이션을 올려 플랫폼을 구성한다.

상용 NIC은 Intel 82580 EB가 사용되었고, 드라이버로는 igb와 igb\_uio가 사용되었다. 실험에 쓰인 환경은 다음과 같다.

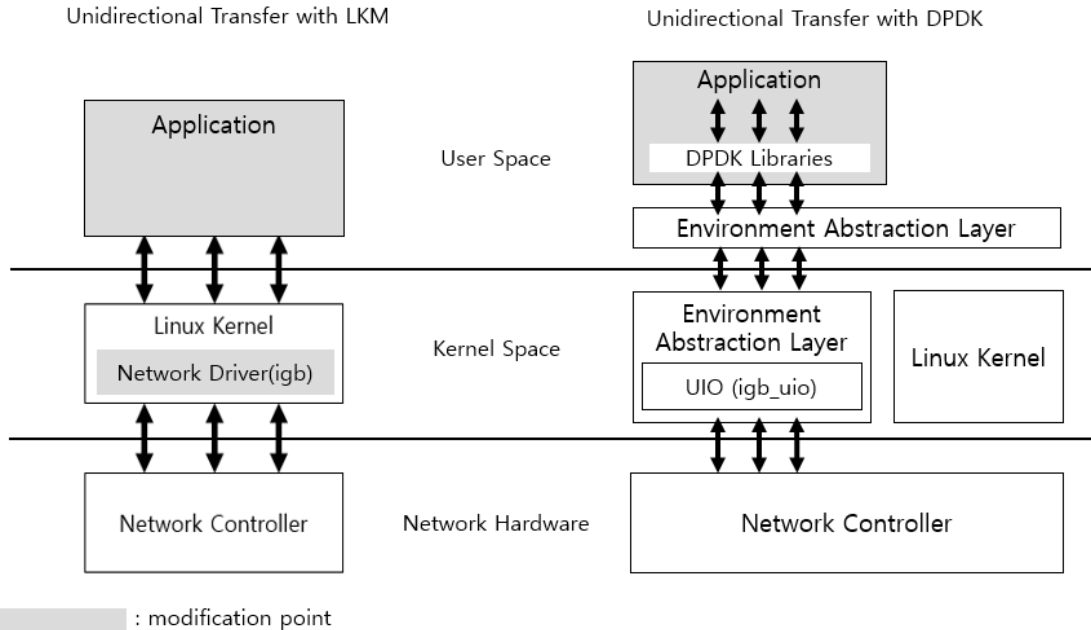


Fig. 10. Comparison of data path in LKM and DPDK

- CPU : Intel Core i5 - 4590 Processor (3.3Ghz)
- OS : Ubuntu 14.04 LTS
- RAM : 8GB (4GB \* 2) Hynix Original
- NIC : Intel 82580 EB
- Driver : igb, igb\_uio(DPDK 2.1.0)
- Ethernet Header : 14 bytes
- IP Header : 20 bytes
- UDP Header : 8 bytes
- DATA : 1024 bytes

4.2 실험 결과

또한, 물리적으로 차단된 사항이기 때문에 양 단말 간 연결을 맺고 데이터를 전송하는 TCP와 같은 신뢰 프로토콜을 사용할 수 없다. 따라서 비 연결 지향성 프로토콜인 UDP를 사용하여 단방향 통신 응용 프로그램을 구현한다. 실험에 쓰인 패킷 구조와 크기는 다음과 같다.

상용 NIC 기반의 단방향 통신 방법을 활용할 수 있음을 확인하기 위해 단방향 데이터 전송 상의 신뢰성과 성능을 측정하였다.

첫 번째로 신뢰성과 관련하여 파일 사이즈와 전송 속도에 대한 손실률을 측정하였다. 실험결과 DPDK 버전의 단방향 통신 플랫폼이 LKM 버전과 비교하여 50Mbps 에서는 손실률이 최대 15배 적었으며, 130Mbps 에서는 손실률이 최대 26배 더 적었다. 또한, 손실률은 파일사이즈가 작을수록, 속도가 낮을수록 줄어드는 경향을 볼 수 있었다.

다음으로, 128MB, 256MB, 512MB, 768MB, 1024MB 크기의 파일을 전송하는데 걸린 시간을 측정하였다. DPDK는 사용자 영역에서 커널로의 복사 없이 NIC으로 직접 복사가 이뤄지기 때문에 같은 크기의 파일을 전송 하는데 LKM 버전의 드라이버 보다 짧은 시간이 걸렸다.

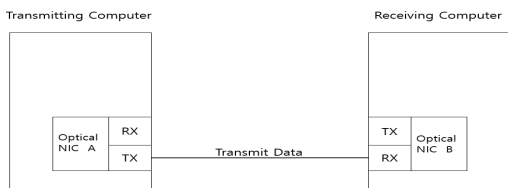


Fig. 11. Environment for unidirectional communication using COTS NIC

Table 6. Results of packet loss test (50 Mbps)

Size (MB)	Speed (Mb/s)	Loss Percent ( % )	
		LKM	DPDK
128	50	0	0
256	50	0	0
512	50	0.2	0
768	50	0.4	0
1024	50	1.2	0.08

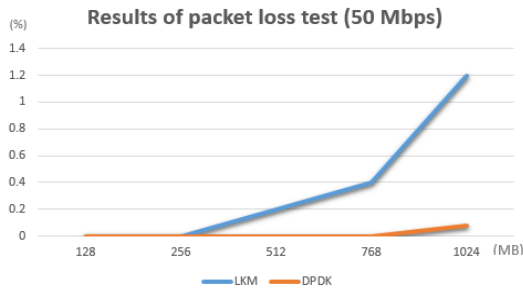


Fig. 12. Results of packet loss test (50 Mbps)

Table 7. Results of packet loss test (130 Mbps)

Size (MB)	Speed (Mb/s)	Loss Percent ( % )	
		LKM	DPDK
128	130	0.05	0
256	130	0.20	0
512	130	1.20	0
768	130	1.73	0
1024	130	3.95	0.15

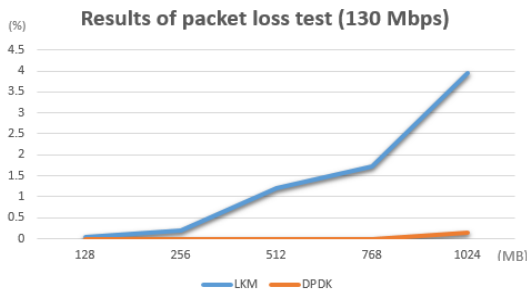


Fig. 13. Results of packet loss test (130 Mbps)

Table 8. Comparison of transmission complete time

Size (MB)	Speed (Mb/s)	Transmission time (Sec)		Result Percentage (%)
		LKM	DPDK	
128	151	4.36	3.33	23.5
256	151	13.01	8.19	37.0
512	151	26.27	15.96	39.3
768	151	43.93	24.42	44.4
1024	151	57.71	32.69	43.4

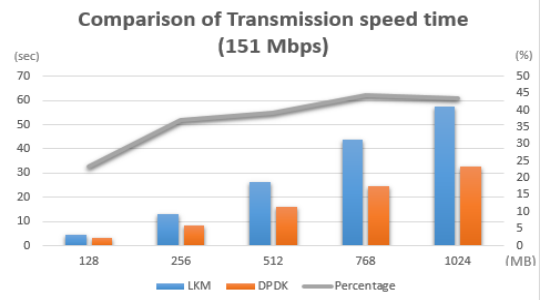


Fig. 14. Comparison of transmission complete time

### V. 결 론

본 논문에서는 업무망 영역을 인터넷과 물리적으로 분리하기 위한 기술로 단방향 통신 방법에 대해 구분하고 Intel 82580 계열의 NIC의 강제 링크 상태 변경과 전송 방식, 전송 속도, 수신기의 설정을 통해 LKM 방식과 DPDK를 사용한 방식의 상용 NIC 기반 단방향 통신 방법이 가능하다는 것을 보였다.

또한, LKM버전과 DPDK버전의 실험을 통해 DPDK버전의 단방향 통신 플랫폼의 효율이 데이터 손실 관점에서 최대 26배, 전송 속도 관점에서 최대 1.8배 더 좋다는 것을 보였다.

향후 계획으로는 단방향 플랫폼의 안전성 향상의 관점에서 전용 하드웨어를 구현하여 성능 실험을 진행해 볼 것이고, 데이터 손실을 극복하고자 하는 관점에서 순방향 오류 정정 기법의 적용에 대한 연구를 실시할 것이다.



## References

- [1] Sung-Hoon, Lee "A Study on Separate Plan of Efficient Information System Network in Partitioned Network Environment", Soongsil University, pp. 1-46, Jun, 2011.
- [2] Se-Chul Park, In-Sik Jang, Yong Lee, Byung-Chul Kim, Myung-Sin Lee, Dae-Hwan Hyun and Dae-Won Chung, "Security Association and Testbed Implementation for Separated Business and Organizational Networks", Journal of the Institute of Electronics Engineers of Korea, Vol 12, pp 42-53, Dec, 2011.
- [3] NIS, "Network deployment guide for secure information and communication environment implementation", 2009
- [4] Jae-bum Park, Huy-Kang Kim "Implementation of secure data relay mechanism removing the streaming relay vulnerability : focusing on the linkage between separate physical networks.", pp. 1-55, Korea University, Aug, 2013.
- [5] "Intel DPDK: Data Plane Development Kit." <http://dpdk.org/>.
- [6] L. Deri, Netikos S P A Via, Brennero Km and Loc La Figuretta. "Improving Passive Packet Capture: Beyond Device Polling." In Proceedings of 4th International System Administration and Network Engineering Conference (SANE), pp. 85-93, 2004
- [7] L.Rizzo, "netmap: a novel framework for fast packet I/O." In Proc. of USENIX ATC, Jun, 2012
- [8] Kyung-Pyo Hong, "Security improvement and reliability improvement of UDP with an UTP unidirectional cable about monitoring system of NPP", Hanyang University. pp 1-60, Aug, 2011.
- [9] D. Gonzalez Gomez, "Receive-onlyUTP cables and Network Taps", Citeseer, 2003.
- [10] Lin Honggang, "Research on Packet Loss Issues in Unidirectional Transmission", Journal of Computers, , Vol. 8 Issue 10, pp. 2664-2671, Oct 2013
- [11] Kwang-Il Jeong, Joon-Ku Lee and Geun-Ok Park, "Design of Uni-directional Optical Communication Structure Satisfying Defense-In-Depth Characteristics against Cyber Attack", KIPS Transactions on Computer and Communication Systems, Vol 02 pp. 561-568, Dec, 2013.
- [12] Intel Networking Division, "Intel 82580EB/82580DB Gigabit Ethernet Controller DataSheet", pp 1-760, Sep, 2015.

---

 < 저자 소개 >
 

---



김 진 홍 (Jin-hong Kim) 정회원  
 2015년 2월: 전남대학교 컴퓨터공학과 졸업  
 2015년 3월~현재: 과학기술연합대학원대학교 정보보호공학과 석사과정  
 <관심분야> 네트워크 보안, 시스템 보안, 암호 알고리즘



나 중 찬 (Jung-chan Na) 종신회원  
 1986년 2월: 충남대학교 계산통계학과 학사  
 1989년 2월: 숭실대학교 전자계산학과 석사  
 2004년 2월: 충남대학교 컴퓨터과학과 박사  
 1989년 2월~현재: ETRI CPS보안연구실실장/책임연구원  
 <관심분야> 제어시스템보안, 펌웨어 보안 취약성



이 성 현 (Seung-Hyeon Lee) 정회원  
 2001년 2월: 한남대학교 컴퓨터공학과 졸업  
 2003년 2월: 한남대학교 컴퓨터공학과 석사  
 2006년 8월: 한남대학교 컴퓨터공학과 박사  
 2006년 9월~현재: ETRI CPS보안연구실/선임연구원  
 <관심분야> IIoT 보안, 소프트웨어 역공학, 사이버 보안관제