

A SDN for Multi-tenant Data Center Based on IPv6 Transition Method

Sheng Maojia et al.,

Information Technology, Networking, Electronic and Automation
Control Conference, IEEE, May 2016.

이부형 (boohyung@pel.smuc.ac.kr)

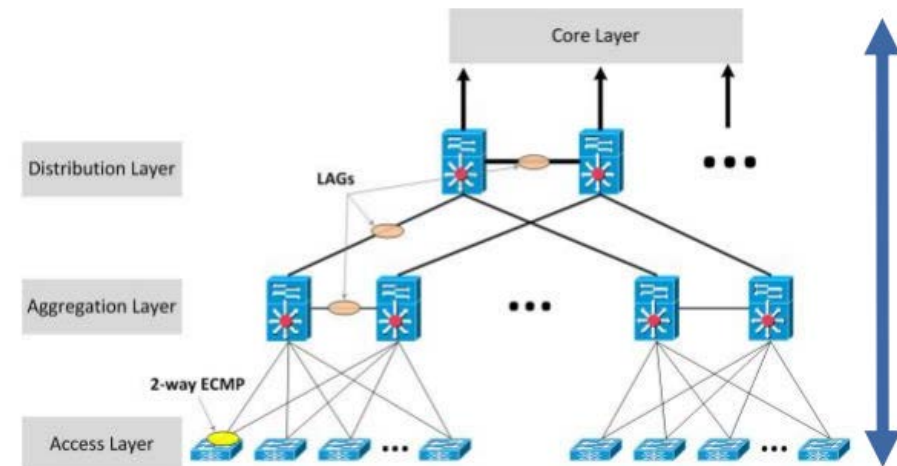
상명대학교 프로토콜공학연구실

Contents

- Introduction
- Related Work
- IPv6 Transition extension in Openflow
- Our Software Defined Multi-tenant Network
- Performance Evaluation

Introduction (1/3)

- Traditional data center network
 - 데이터센터: 서버 컴퓨터 + 네트워크 회선
 - Telco의 기존 장비를 그대로 사용; 랙 (rack)
 - 다계층 구조
 - Core Layer (backbone): 고성능의 백본 스위치나 WAN 라우터 등
 - Distribution Layer: 방화벽, 브리지, AR, LAN 라우터 등
 - Access Layer: 데스크탑, 메일 서버, 파일 공유 서버, 허브 등
 - 단점: 운영 및 설정이 복잡, 비효율적인 링크 대역폭을 사용, 지연 및 부하 관리 어려움



Introduction (2/3)

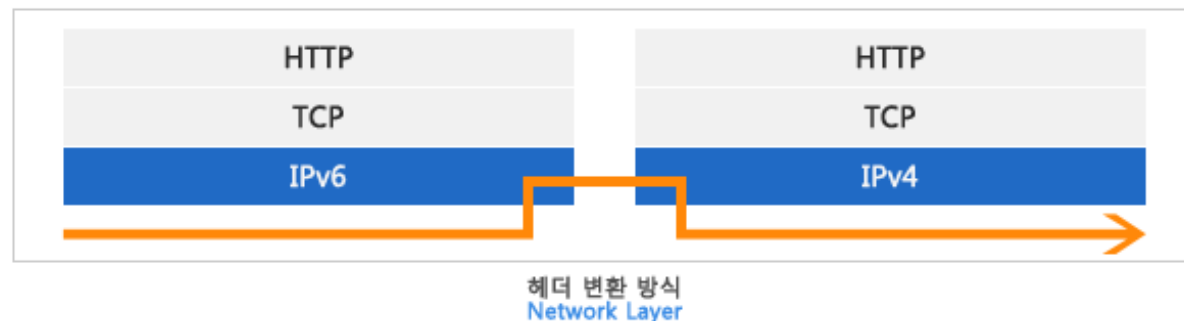
- Multi-tenant data center network
 - 정의: 멀티테넌트로 구성된 클라우드 서비스 센터 네트워크
 - 특징
 - 다수의 테넌트 (가상 컴퓨터, VM)을 포함
 - 테넌트 간의 격리 및 보안을 제공
 - SDN (Software Defined Network)
 - 데이터센터 네트워크에서 데이터 영역과 제어 영역을 분리
 - 네트워크의 확장성을 제공
 - Implementation: OpenFlow, Open vSwitch
 - NFV (Network Function Virtualization)
 - IPv4 주소 공간의 분리
 - 테넌트의 이동성 지원
 - Implementation: FlowVisor

Introduction (3/3)

- Requirements of future data center
 - Agile and programmable
 - API를 사용한 인터페이스 구성
 - 사용하는 기능을 정확하고 빠르게 지원
 - Tenant distinguishable
 - 테넌트 간 독립적인 서비스 사용
 - 보안 및 QoS 보장
 - Supporting transition
 - IPv4-v6 주소 전환 가능
 - 듀얼 스택 모드 지원
 - Unified control strategy
 - 네트워크 모니터링, 오케스트레이션, 스케줄링 및 최적화에 유리

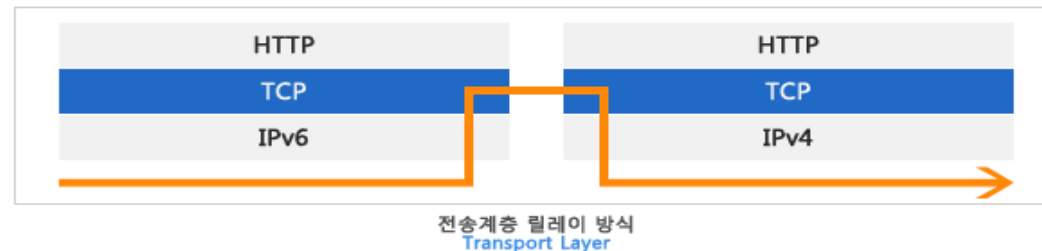
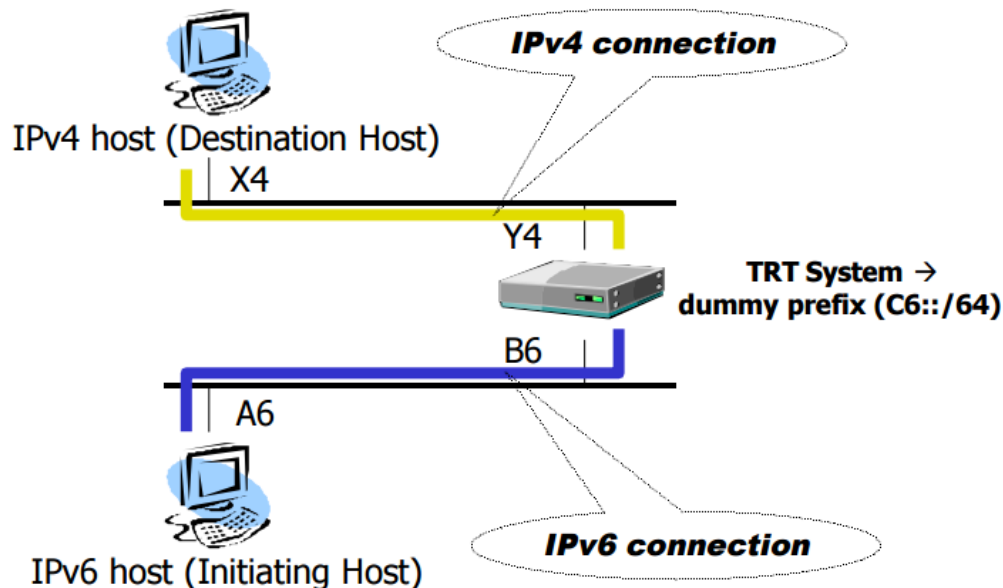
Related Work (1/4)

- Traditional IPv6 transition mechanism
 - 헤더 변환 방식
 - Stateless IP/ICMP Translation Algorithm (SIIT)
 - RFC 2765
 - IPv4와 IPv6 패킷 헤더 간의 변환 규칙, 주소 매핑 방식 정의
 - 패킷 무결성 유지를 위한 체크섬 계산
 - Network Address Transition Protocol Translation (NAT-PT)
 - RFC 2766
 - 헤더 변환: SIIT 방식을 사용
 - NAT 장비에서 미리 할당한 주소 매핑 정보를 활용



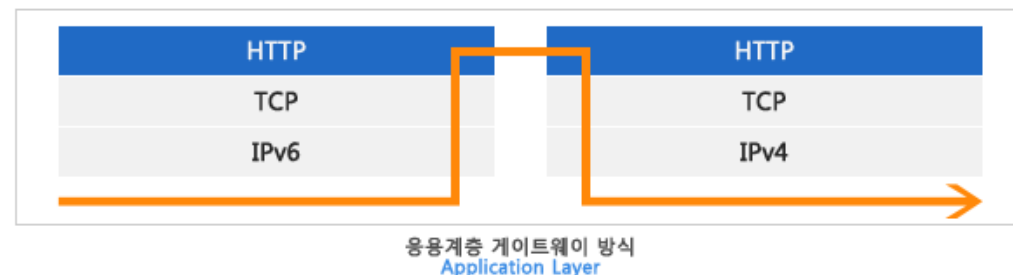
Related Work (2/4)

- Traditional IPv6 transition mechanism
 - 전송계층 릴레이 방식
 - IPv6 to IPv4 Transport Relay Translator (TRT)
 - RFC 3142
 - 전송 계층에서 IPv4-v6 세션 사이에 위치하는 릴레이 (TRT System)를 이용하여 변환
 - dummy prefix (C6:: /64): TCP를 사용하는 IPv6 세션



Related Work (3/4)

- Traditional IPv6 transition mechanism
 - 응용계층 게이트웨이 방식
 - 응용계층에서 주소를 변환
 - 응용 수준의 게이트웨이 (ALG)가 IPv4와 IPv6 프로토콜을 모두 지원할 수 있어야 함
 - DNS, FTP와 같은 응용 프로토콜에 내장된 IP 계층 주소를 변환하는데 사용
 - DNS-ALG, FTP-ALG



Related Work (4/4)

- IPv6 transition in SDN
 - 주로 터널링이나 듀얼 스택 기술을 사용
 - 터널링: IPv6 패킷을 IPv4 패킷에 캡슐화하여 전송하는 방식
 - 듀얼 스택: IPv4와 IPv6 프로토콜을 둘 다 지원
- Non-tunneling model을 제안, Ryota Kawashima, 2013.
 - OpenFlow를 사용
 - 프레임의 헤더를 재작성하여 캡슐화한 헤더를 압축
- 제안하는 기법: 터널링을 이용하지 않고, 기존에 사용하는 IP 헤더에 테넌트 네트워크 ID를 임베딩하는 방법을 사용

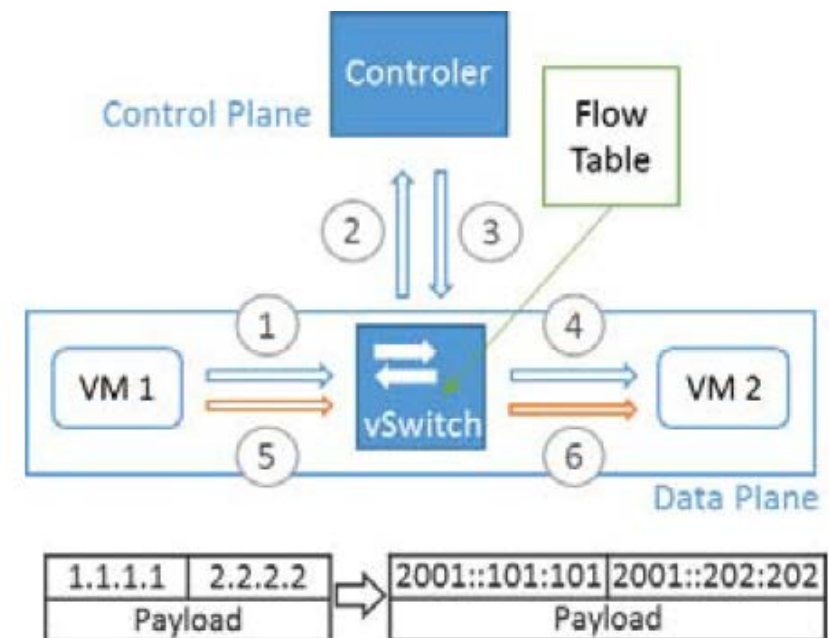
IPv6 Transition extension in Openflow (1/4)

- V4V6 and V6V4 translating actions
- Flow table format (extended Openflow)
 - Rule
 - 패킷에서 확인 가능한 정적 정보
 - 포트번호, IP 주소, VLAN ID 등
 - Action
 - Open vSwitch에서 수행하는 기능
 - 기존내용에 2개의 새 항목 추가: V4V6, V6V4 (주소 번역)
 - Statistic information
 - 패킷의 동적 상태 정보
 - 상태 정보는 컨트롤러에게 전송됨

OPENFLOW FLOW TABLE ITEM						
RULE					ACTION	STAT
IN PORT	VLAN	ETH	IP/IP6	TCP/UDP	OUTPUT/V4V6/V6V4/...	PKT/BYTE/TIME

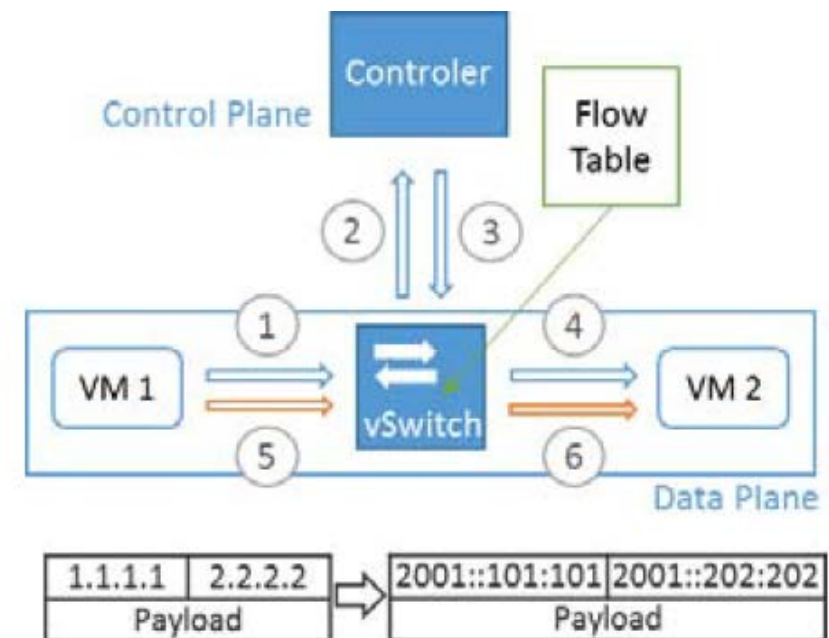
IPv6 Transition extension in Openflow (2/4)

- V4V6 and V6V4 translating actions
- Flow path and translating actions
 - VM 1: IPv4 테넌트, VM 2: IPv6 테넌트
 - Slow path
 - 1) 패킷 전달
 - 2) 전달된 패킷의 정보 확인 후 (Flow table과 비교)
기존정보와 일치하지 않으면 컨트롤러에 전송
 - 3) 컨트롤러는 주소 변환 action을 vSwitch에 할당 후 명령
 - 4) 주소 번역 (V4 -> V6) 후 패킷전달



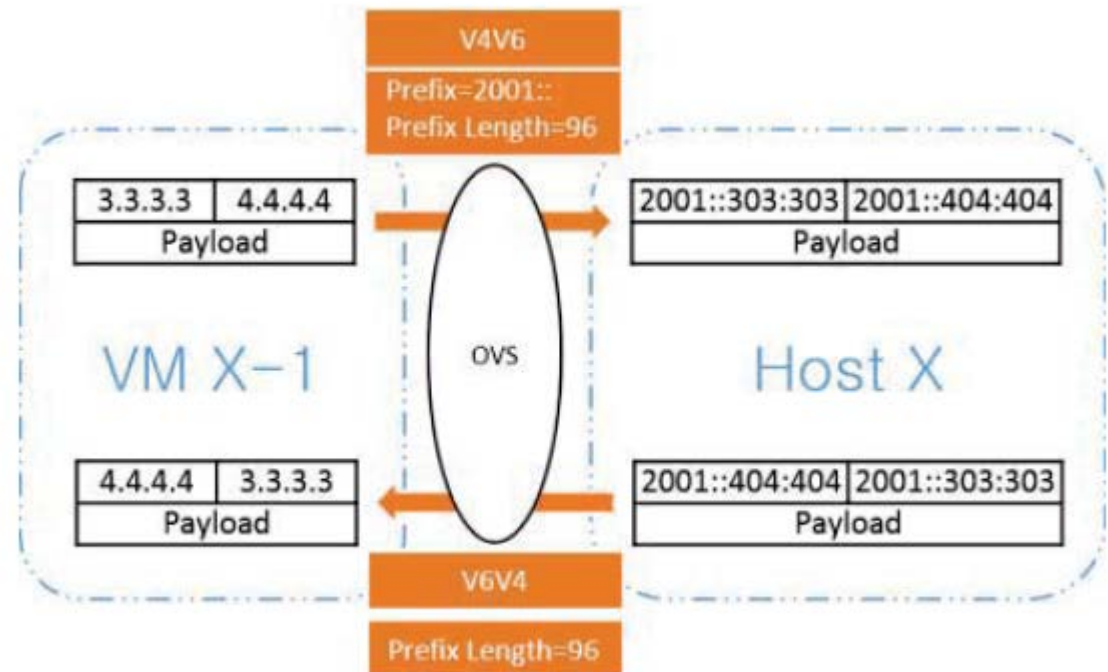
IPv6 Transition extension in Openflow (3/4)

- V4V6 and V6V4 translating actions
- Flow path and translating actions
 - VM 1: IPv4 테넌트, VM 2: IPv6 테넌트
 - Fast path
 - 5) 만약, 같은 경로로 패킷이 전달된다면 등록된 Flow table과 패킷 정보를 비교
 - 6) 주소 번역 후 패킷 전달
 - 컨트롤러를 거치지 않고 주소 변환 후 패킷을 전달



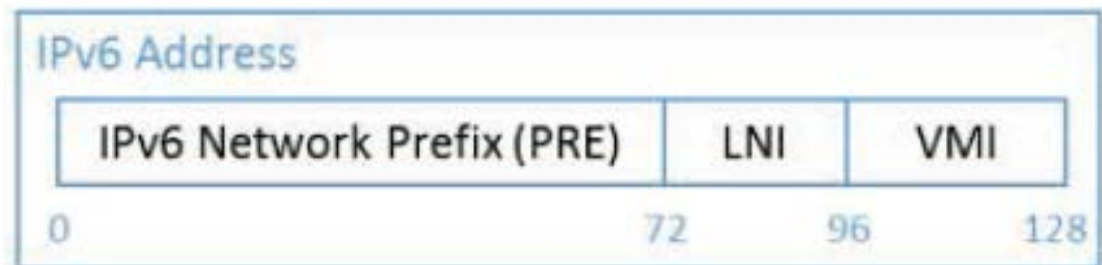
IPv6 Transition extension in Openflow (4/4)

- A unified IPv6 Network with as a service
 - VM's IPv4 traffic -> IPv6 in physical hosts (vice versa)
 - OVS (Open vSwitch)에서의 주소 변환
 - 서브네팅을 위한 IPv6 Prefix의 사용
 - IPv4에서의 서브넷 마스크와 유사



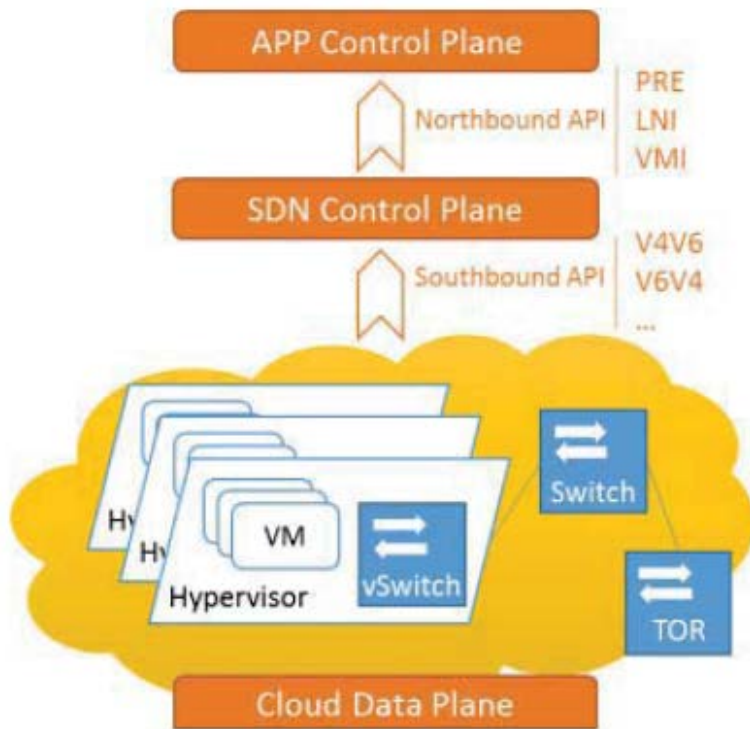
Our Software Defined Multi-tenant Network (1/2)

- IPv6 address format
 - IPv6 Network Prefix (PRE)
 - 한 데이터 센터에는 여러 PRE가 존재할 수 있음
 - 데이터센터 네트워크와 다른 네트워크 (인터넷이나 다른 데이터센터 네트워크)를 분리하기 위함
 - Logical Network ID (LNI)
 - IPv4 or IPv6 테넌트들에게 할당됨
 - 최대 2^{24} 개까지 운영 가능 (약 1600만 개)
 - Virtual Machine ID (VMI)
 - 테넌트가 사용하는 가상 머신의 ID



Our Software Defined Multi-tenant Network (2/2)

- Our SDN scheme for multi-tenant DC
 - (우측 그림) Northbound API의 구성 예



```
/virtual_net_ops/set_default_gateway
{
  "network":{
    "ipv6_pref":"2001:100:200:300::",
    "logic_net_id":7,
    "vm_mask":0
  },
  "direction":"out",
  "gateway":{
    "ipv6_prefix":"2001:100:200:300::",
    "logic_net_id":7,
    "vm_id":1
  }
}
```

Performance Evaluation (1/5)

- OpenFlow에서 V4V6/V6V4를 구현
 - 구현 환경
 - Network: OVS Framework, Mininet
 - CPU: 2.6 GHz
 - Memory: 1GB

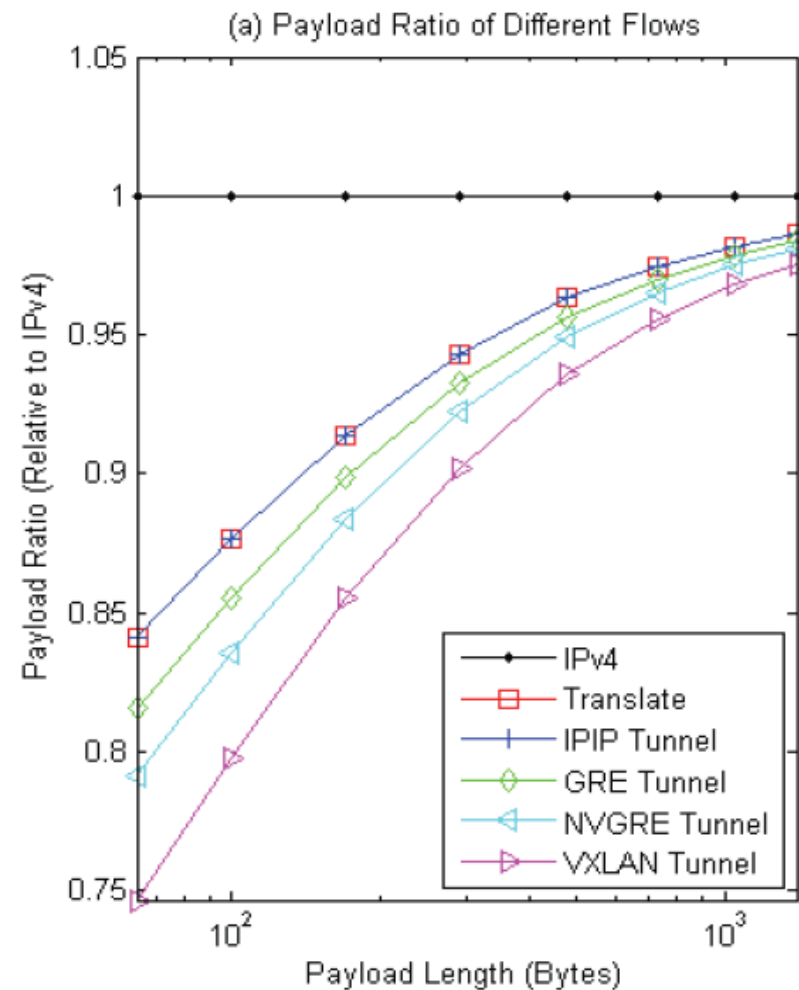
Performance Evaluation (2/5)

- 성능 측정 지표

- Payload Ratio (페이로드 비율) = $\frac{\text{페이로드 길이}}{\text{패킷 길이 (헤더 포함)}}$
- Transfer Rate (전송률) = $\frac{\text{전송한 데이터 양}}{\text{단위 시간}}$
- Relative Transfer Rate (RTR_A) = $\frac{TR_A}{TR_{IPv4forward}}$
- Relative Payload Ratio (RPR_A) = $\frac{PR_A}{PR_{IPv4forward}}$

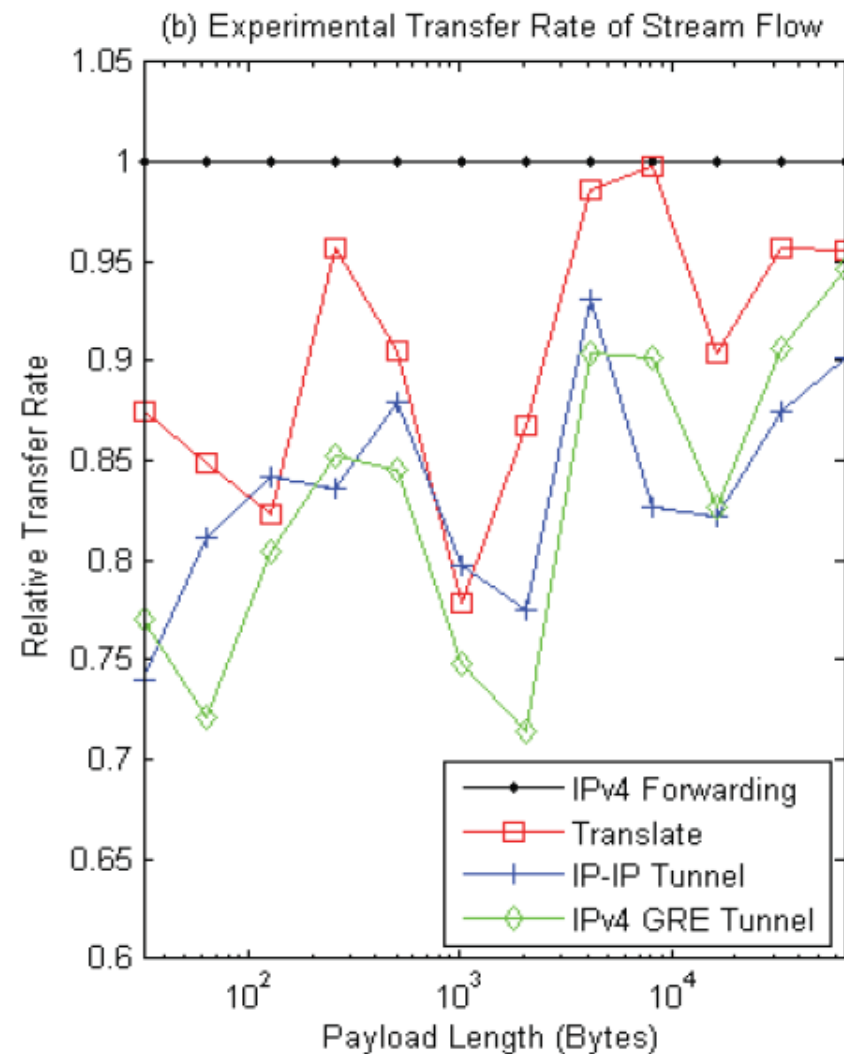
Performance Evaluation (3/5)

- 성능 측정 결과 (1/3)
 - 페이로드 길이에 따른 페이로드 비율
 - IPv4 (비교군)
 - Translate (제안하는 기법)
 - IPIP Tunnel
 - GRE Tunnel
 - NVGRE Tunnel
 - VXLAN Tunnel
 - 결과: IPIP Tunnel, Translate를 사용할 때 오버헤드가 가장 큼



Performance Evaluation (4/5)

- 성능 측정 결과 (2/3)
 - 페이로드 길이에 따른 전송률
 - IPv4 Forwarding
 - Translate (제안하는 기법)
 - IP-IP Tunnel
 - IPv4 GRE Tunnel
 - 결과: 터널링을 사용할 때보다 Translate를 사용할 때 더 나은 전송률을 보임



Performance Evaluation (5/5)

- 성능 측정 결과 (3/3)

- 플로우 타입에 따른 전송률 (요청/응답 시)

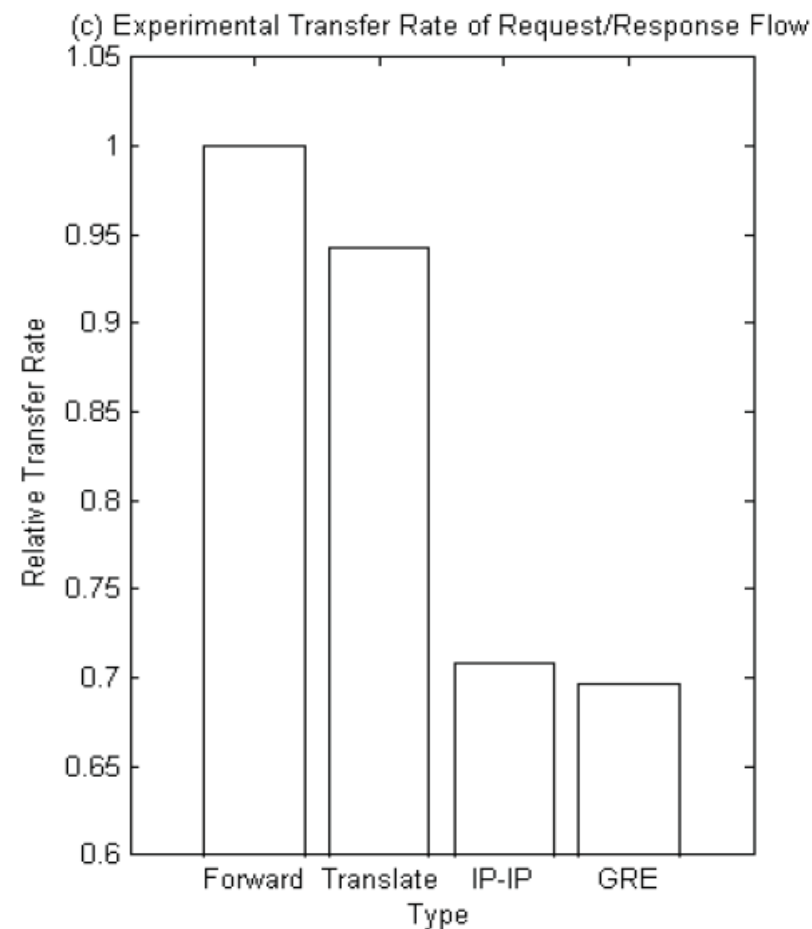
- Forward

- Translate (제안하는 기법)

- IP-IP

- GRE

- **결과: IPv4 Forwarding을 사용할 때와 비슷한 정도의 전송률을 가짐 (터널링을 사용할 때보다 우수)**



감사합니다!

이부형 (boohyung@pel.smuc.ac.kr)