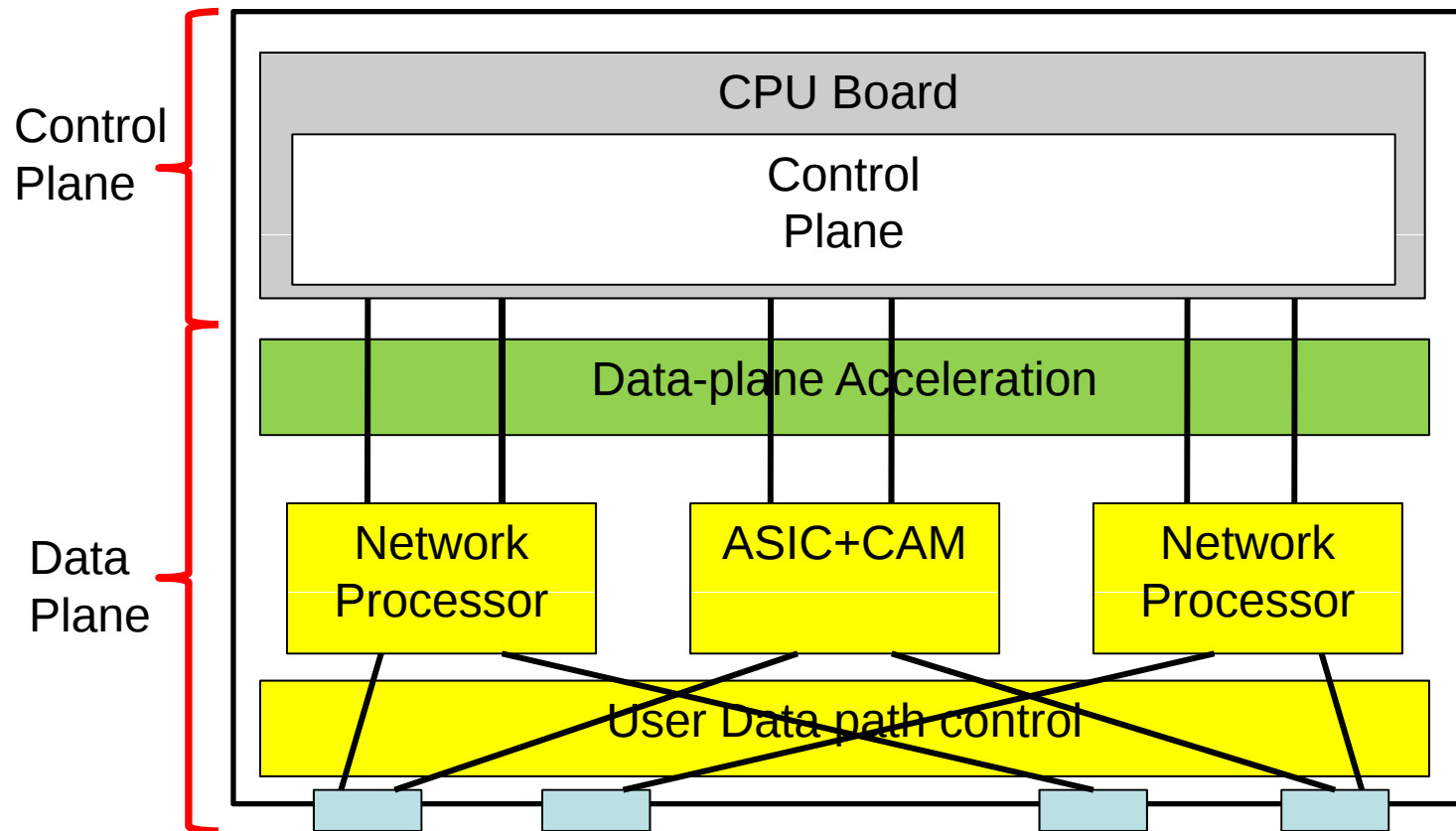


Virtualized Testbed Which is Composed of Network of Programmable Routers

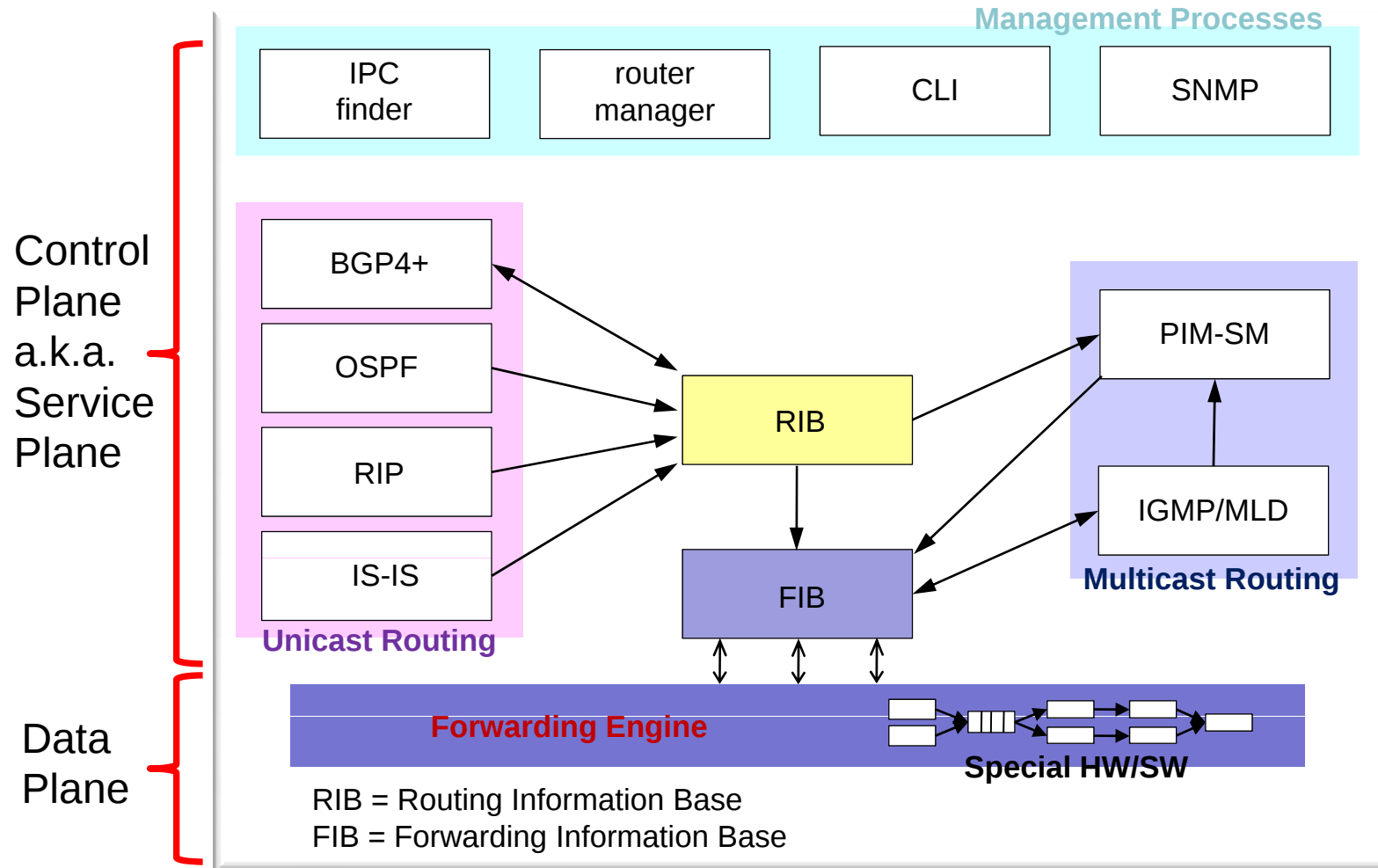
**2010.02.23
Sueng-Y. Park
Yonsei University**

Router Architecture

- Major protocols and other control plane applications run on CPU board.
 - Data plane is optimized for fast forwarding.



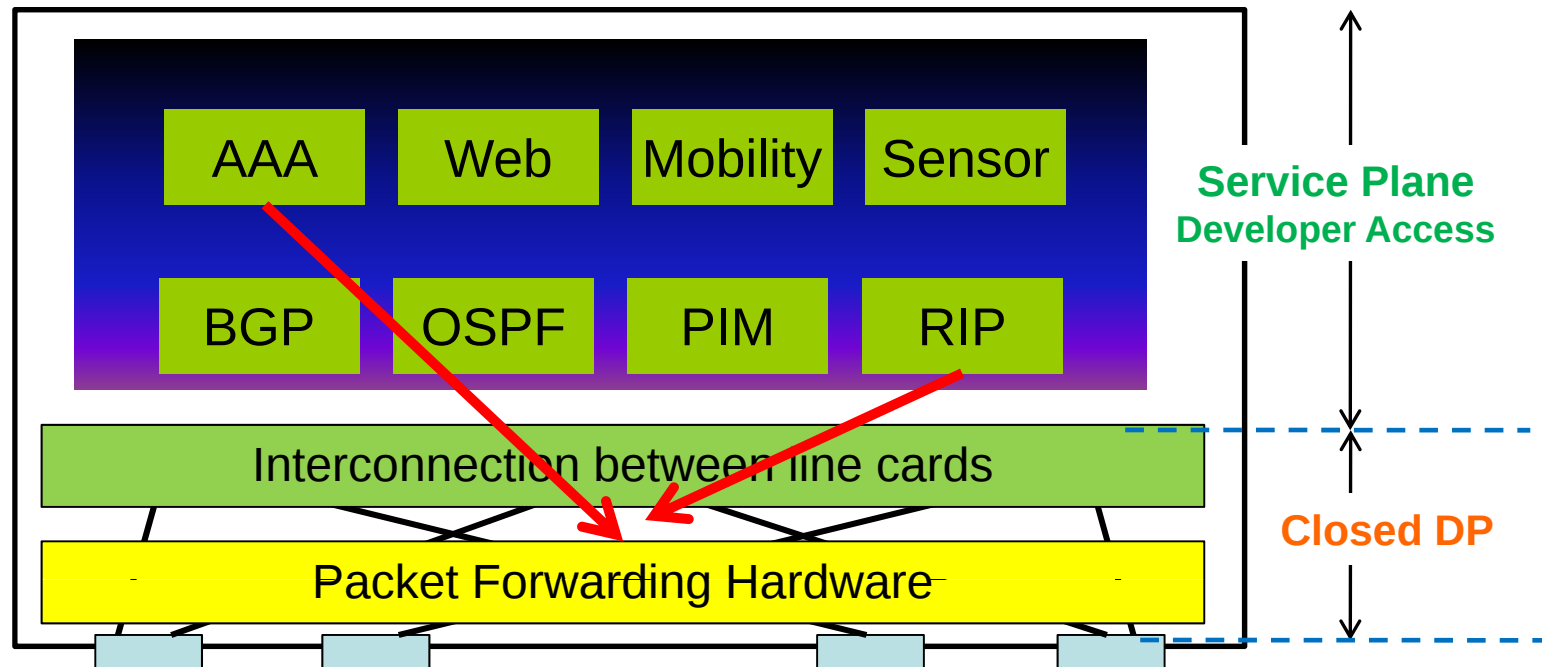
Conceptual View Of Router



Renovate Router

Step 1: Service Plane Communicates with DP

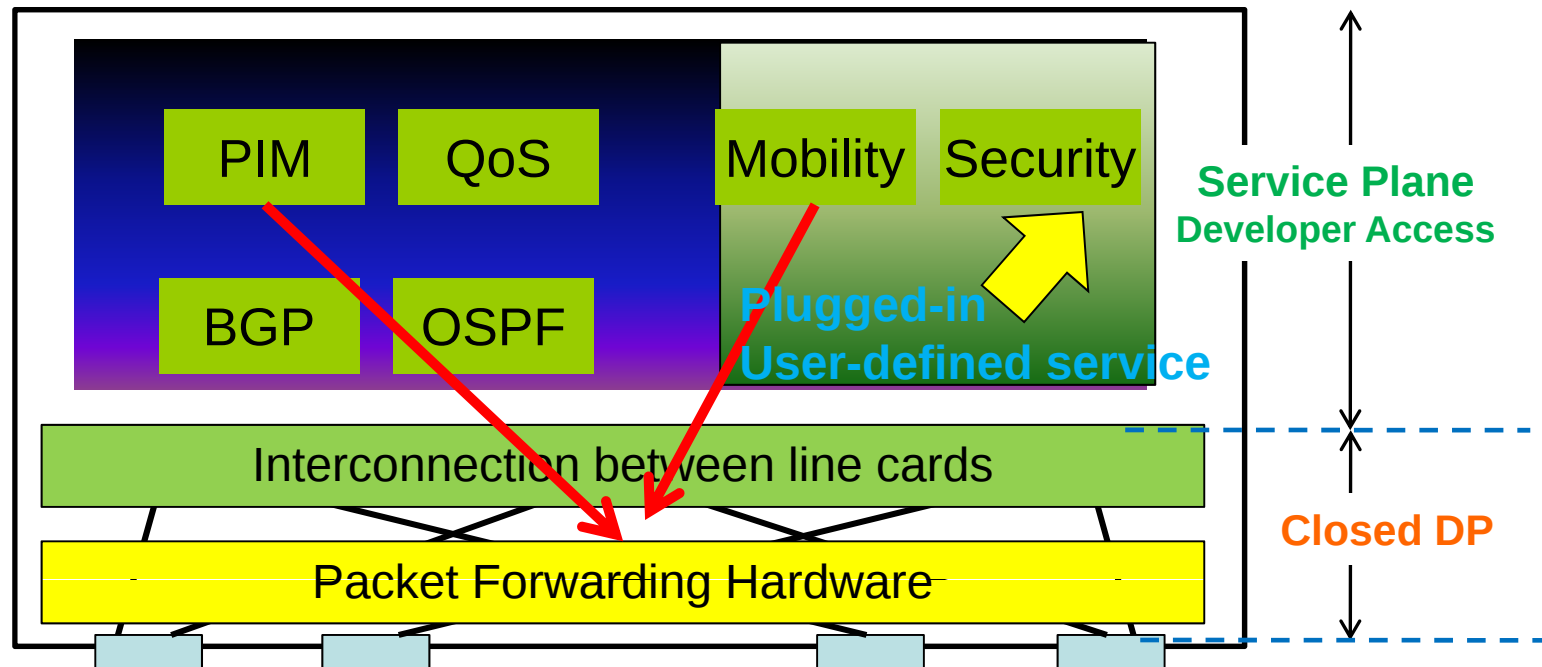
- Some commercial routers already use IPC (inter-process communication) to control DP
 - DMA, Direct memory mapping, etc are viable too.



Renovate Router

Step 2: Provide Isolation between Legacy and User Service

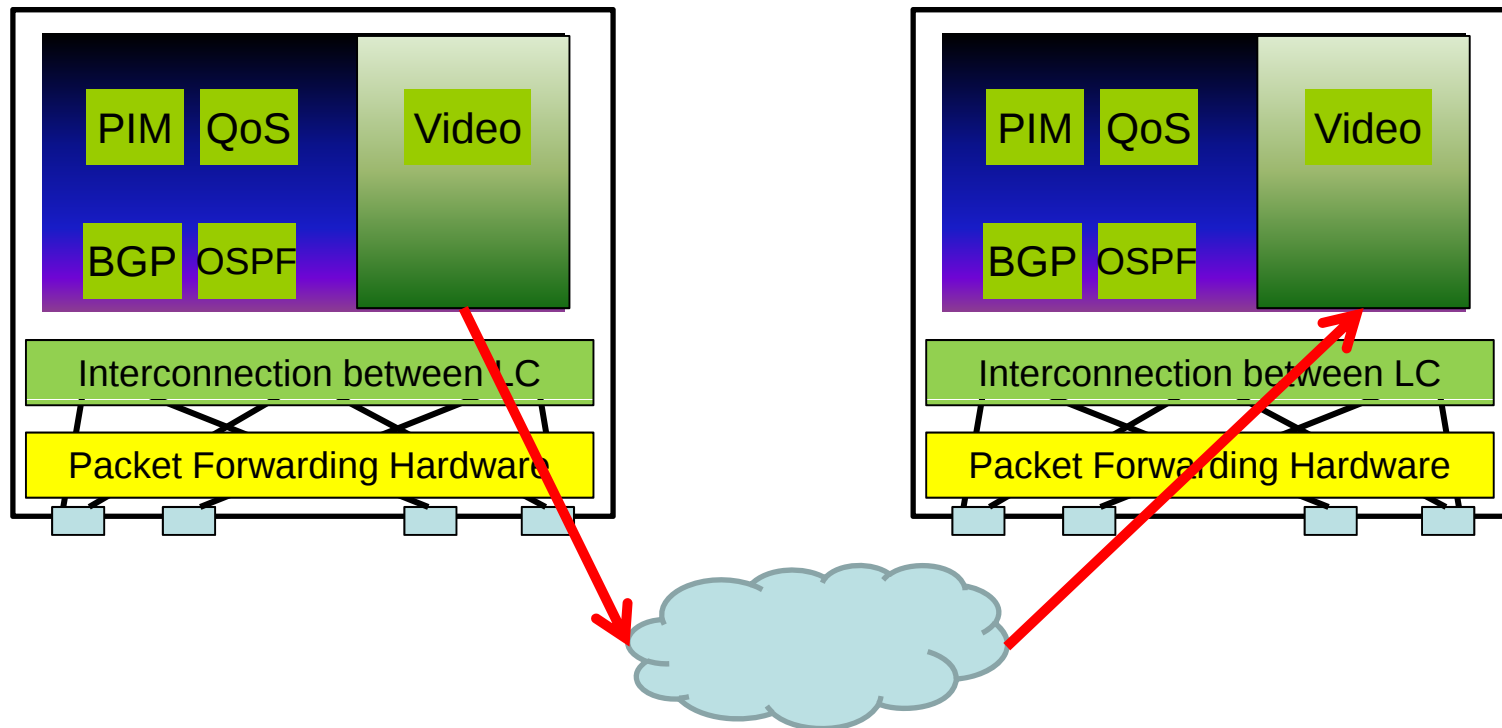
- Legacy service is closed from public access.
- User-defined service is plugged in isolated domain.



Renovate Router

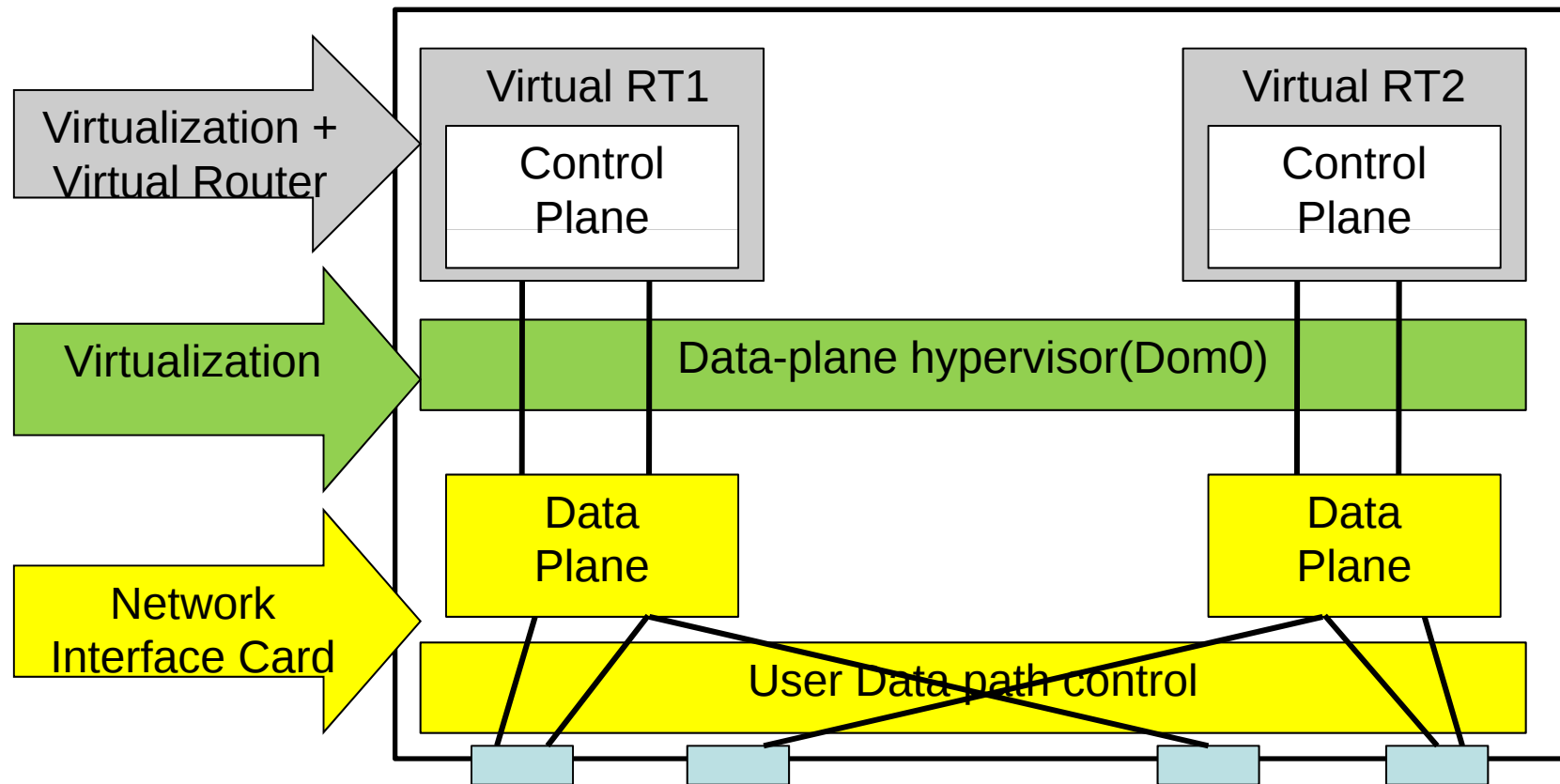
Step 3: Form Network with Programmable Router

- Legacy service is closed from public access.
- User-defined service is plugged in isolated domain.



Virtual Router Architecture

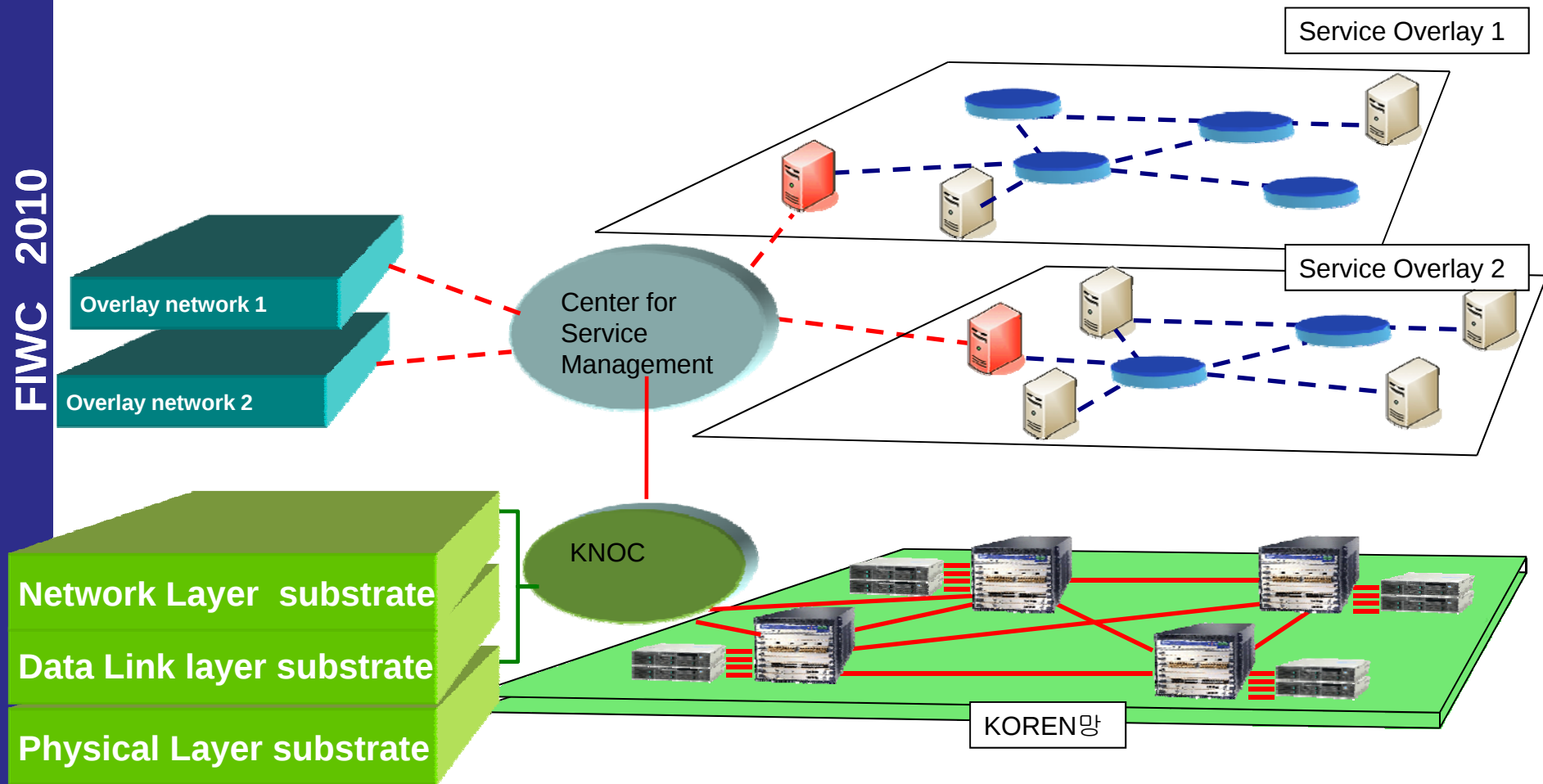
- We have multiple virtual routers in one physical machine.



Resource Monitoring

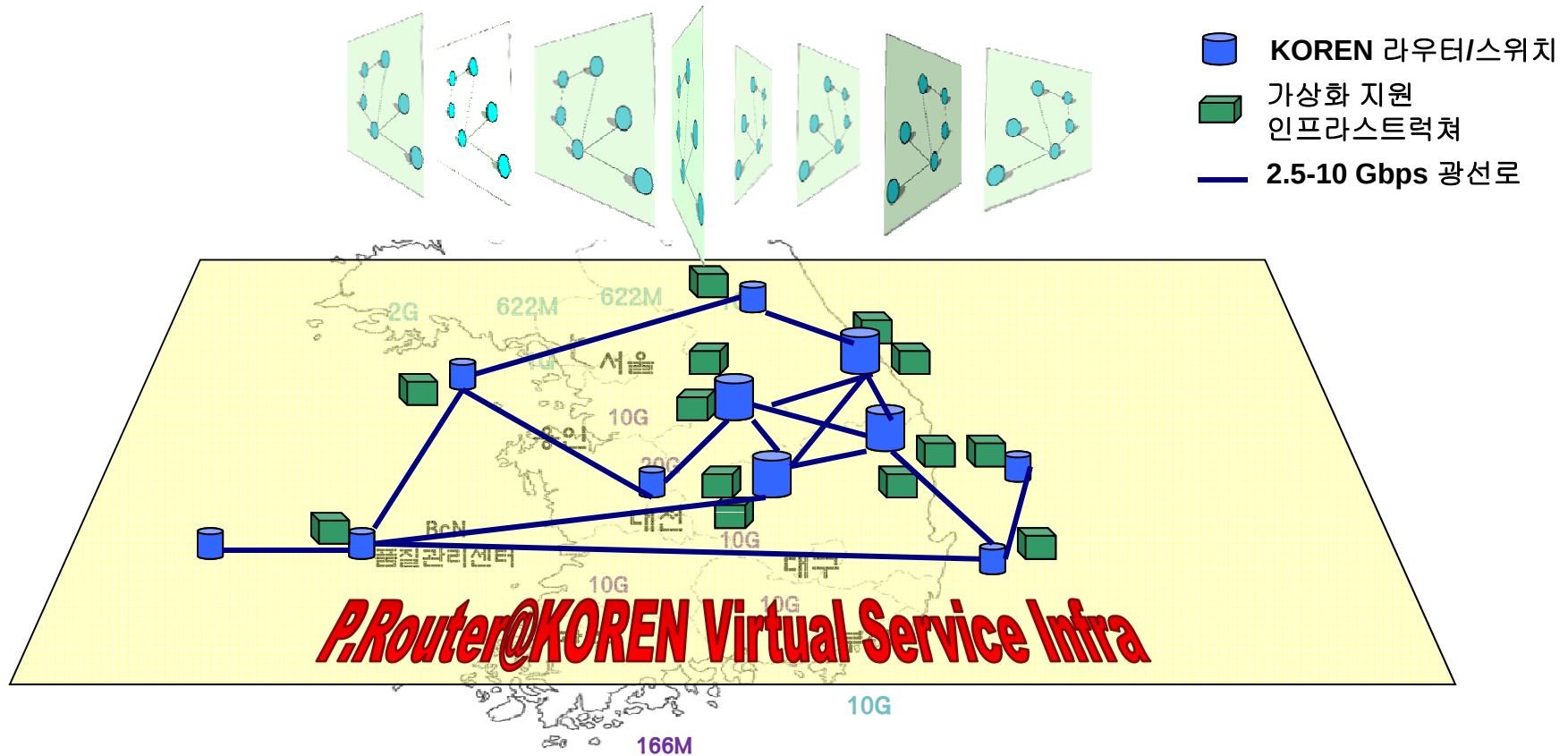
- O.K. We virtualized router. What next?
 - We need the methods of monitoring system if the virtualized router is working properly.
- Especially, the performance of router is very sensitive (as of now) to available CPU resources.

What We Need in Future

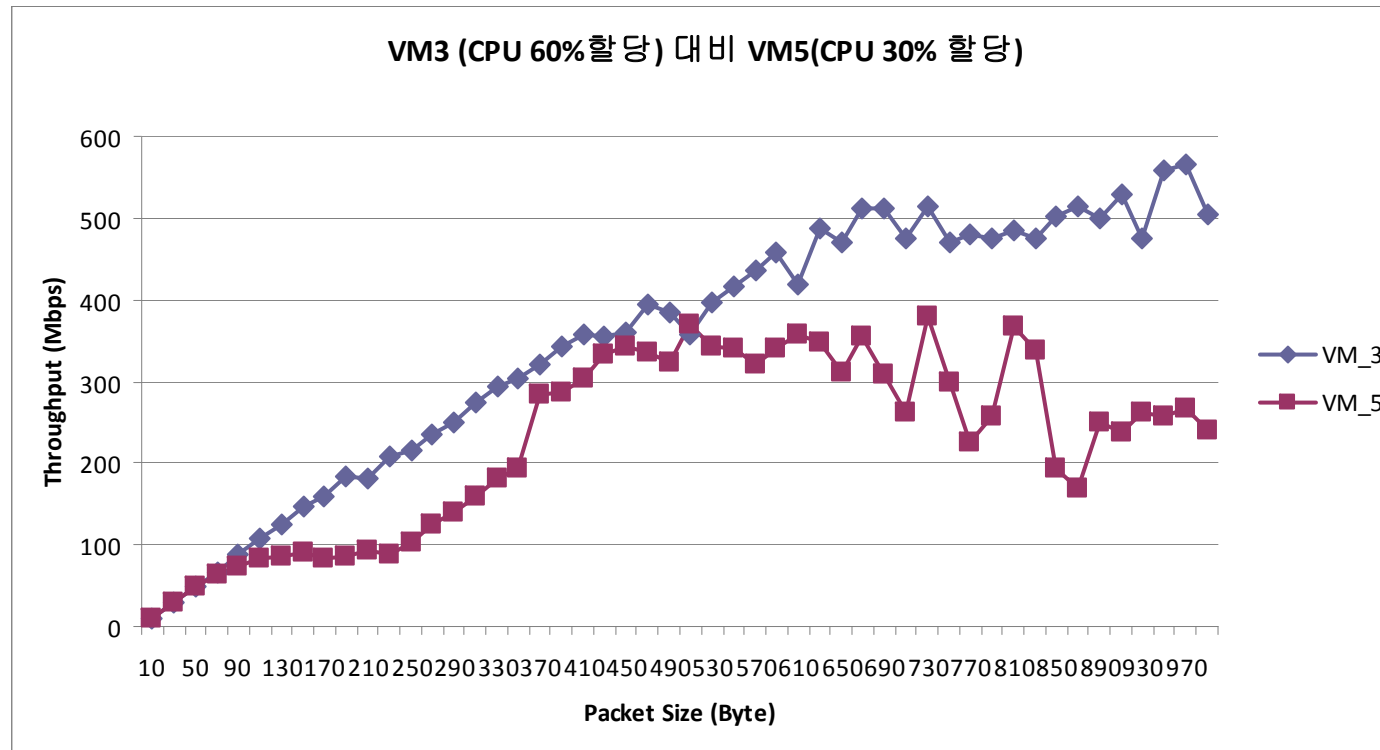


Overlay Network Optimized for Service

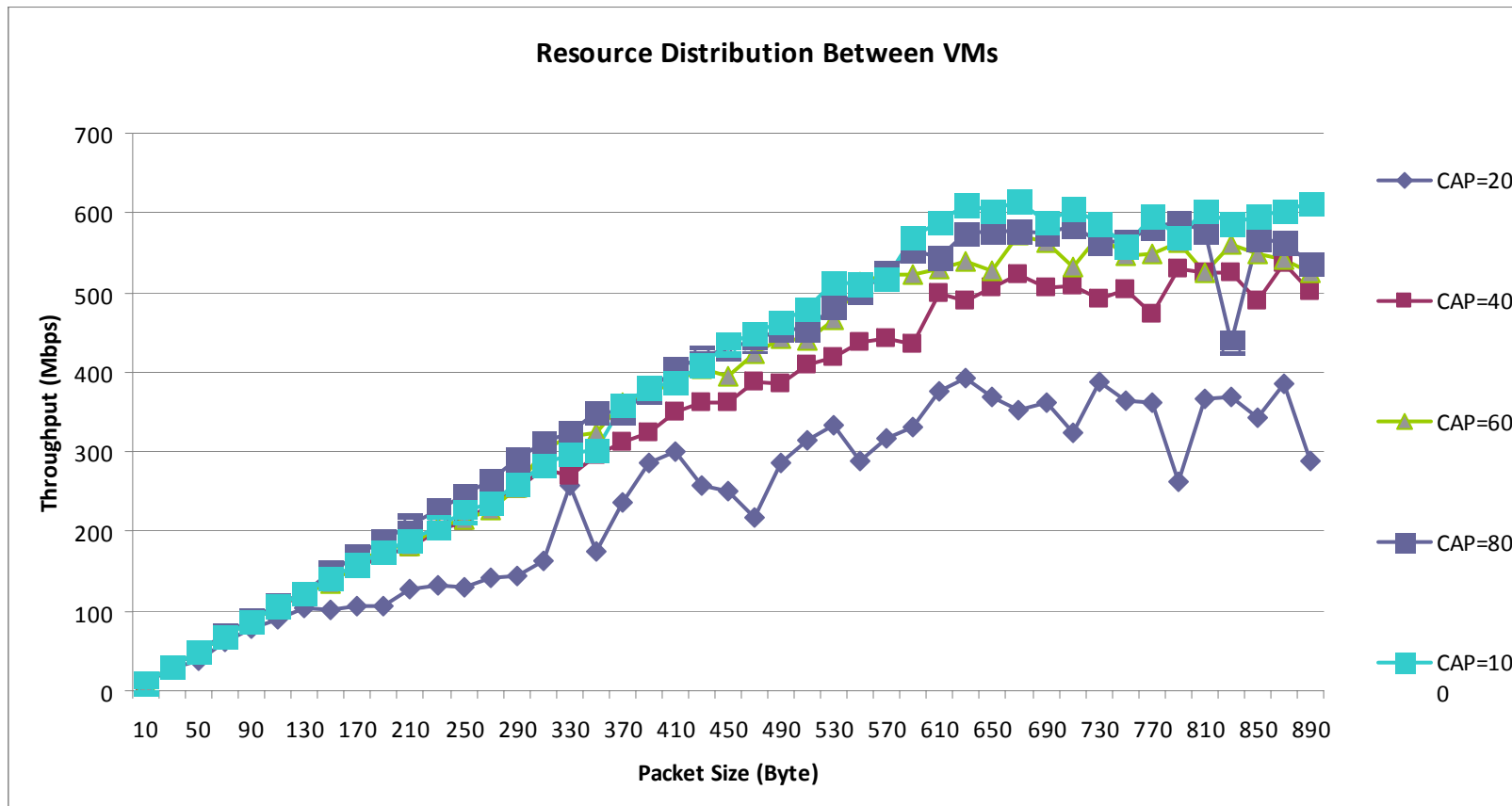
가상화된 서비스 망



Resource Isolation Issue



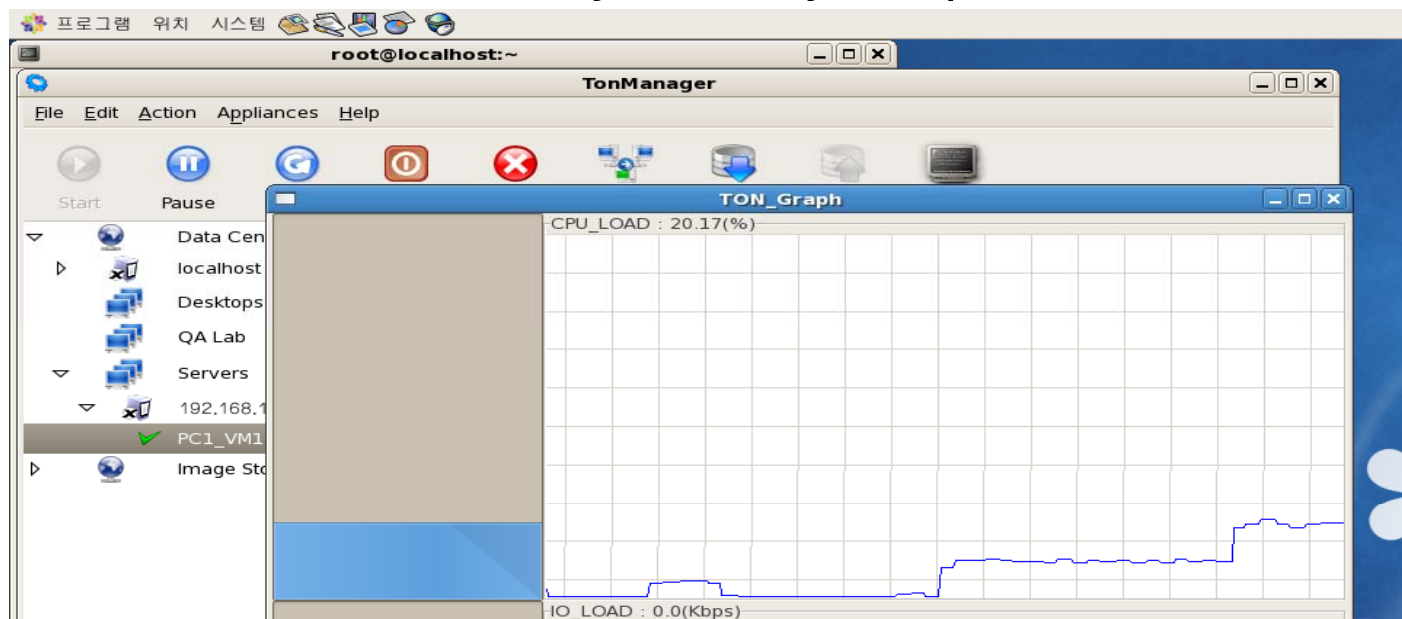
Resource Isolation Issue



Remote Monitoring of V.Router

Real-time monitor

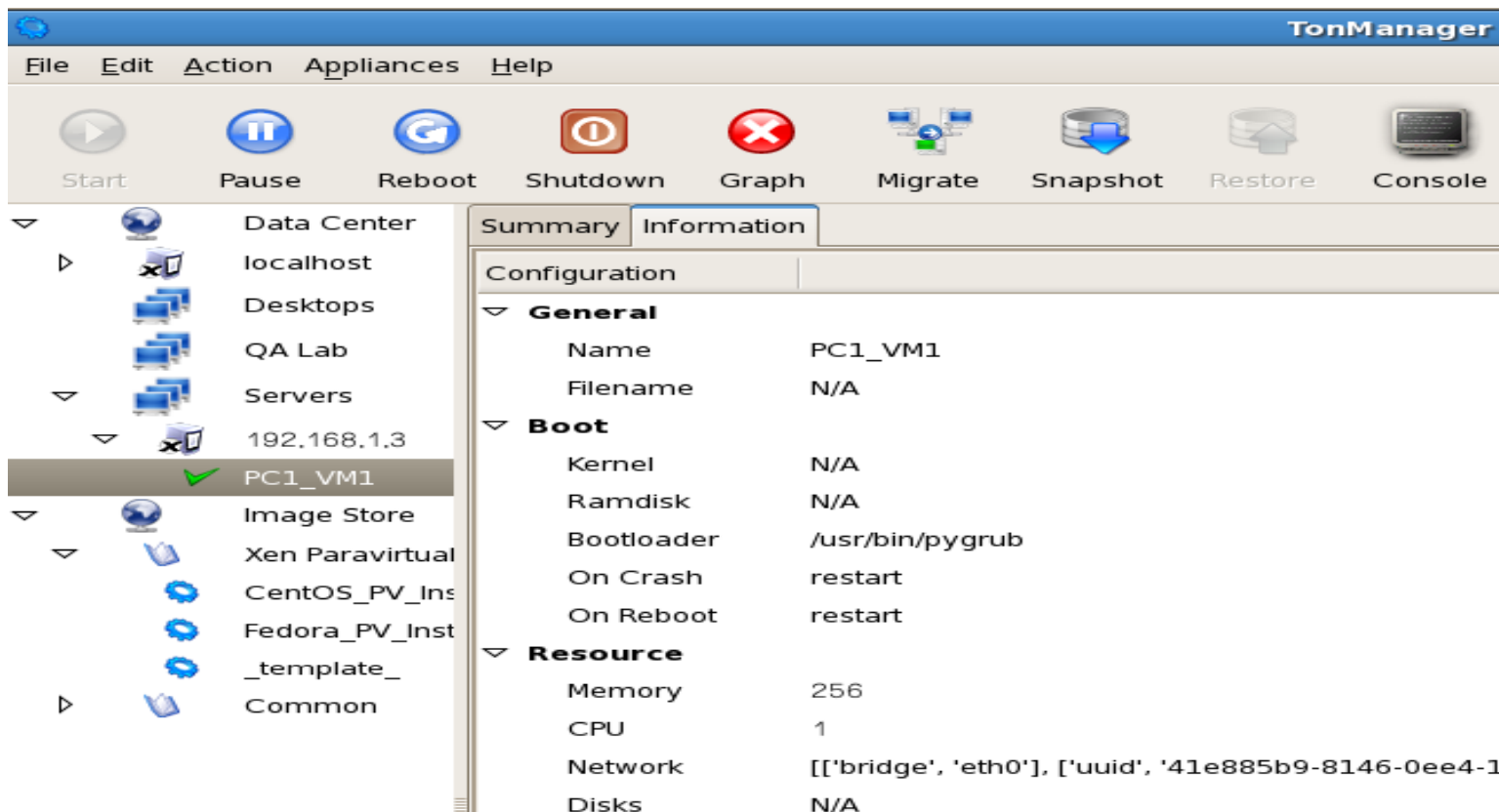
- Real-time monitoring is useful for quick verification of operation .
 - We can verify important parameters such as traffic , I/O status, and CPU usage.
 - We can immediately indentify if experiment starts.



Remote Monitoring of V.Router

Real-time monitor

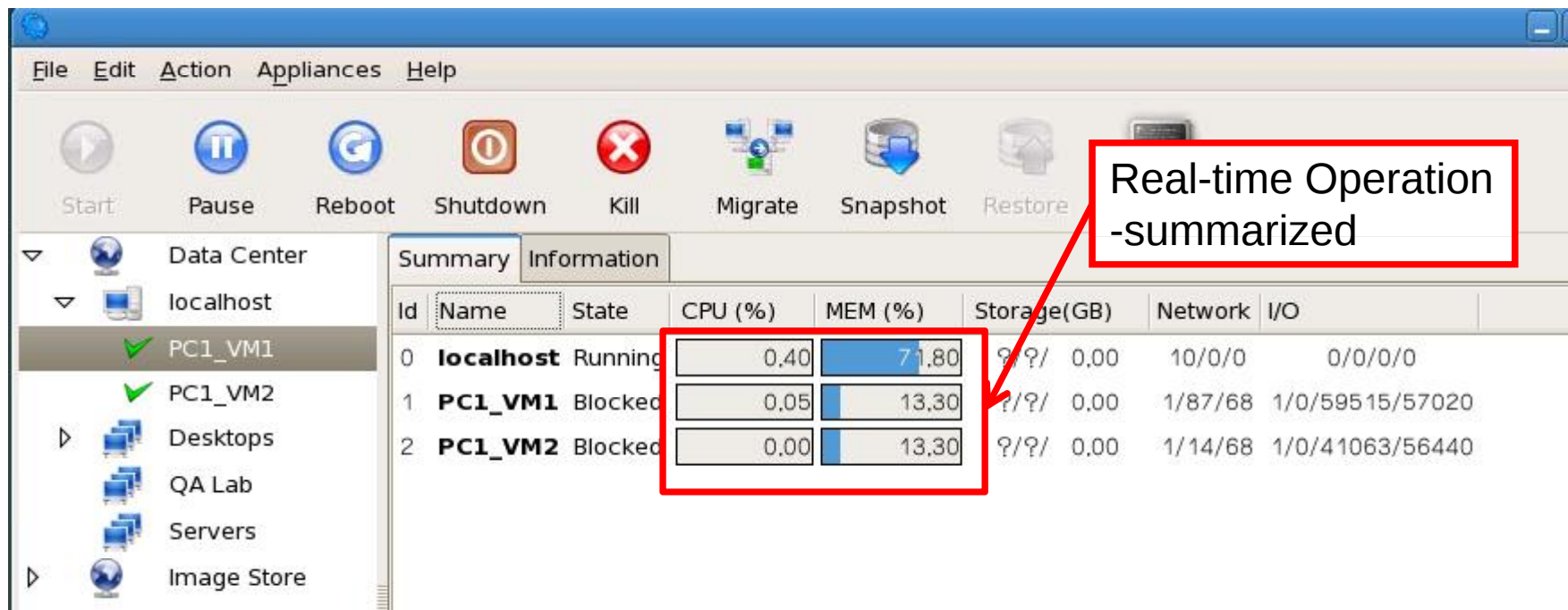
- We can get rough idea of overlay network by real-time monitoring.
 - Accurate data can be obtained by log or dump.



Remote Monitoring of V.Router

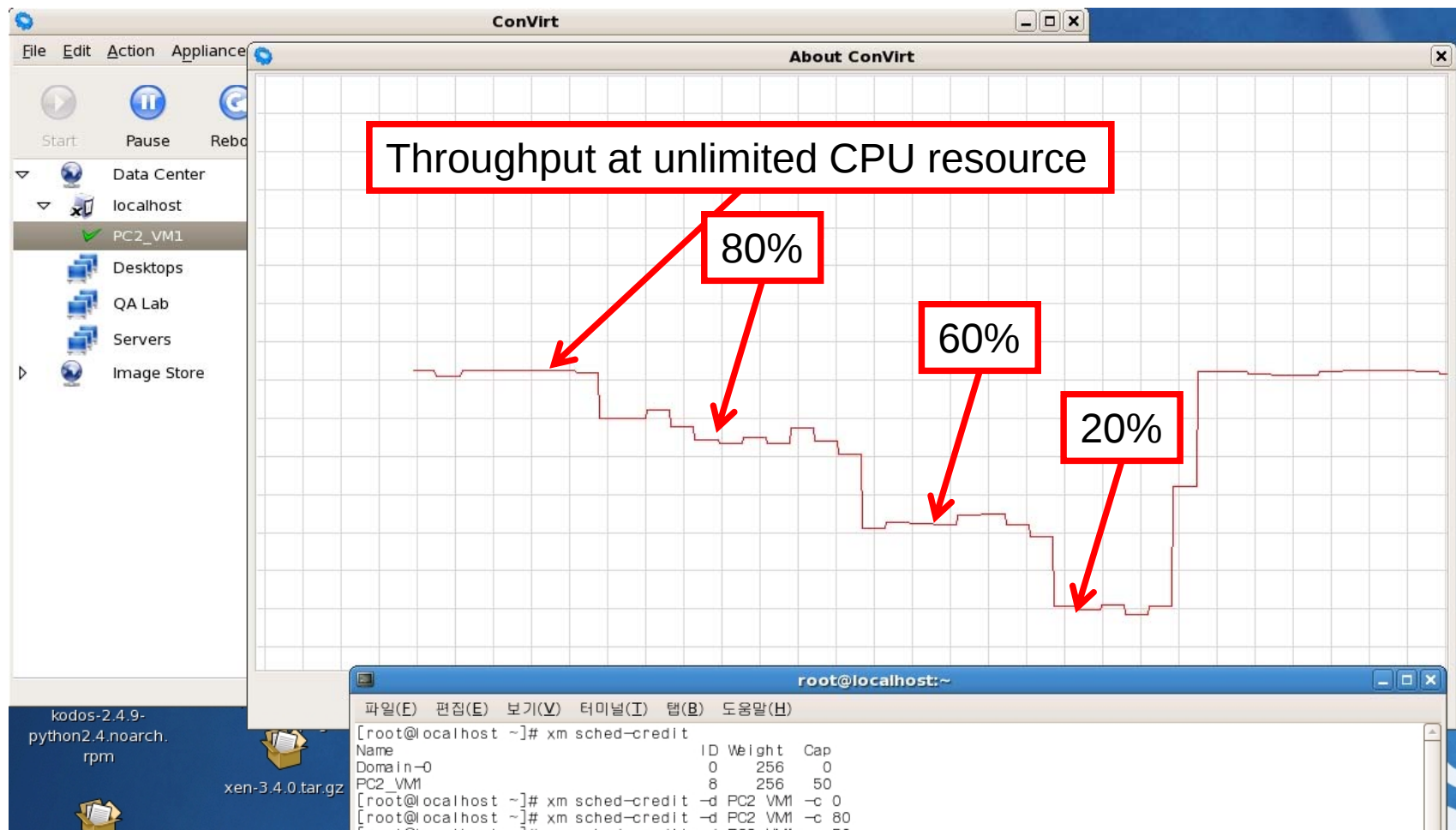
Real-time monitor

- You can define (or program) your own statistics.

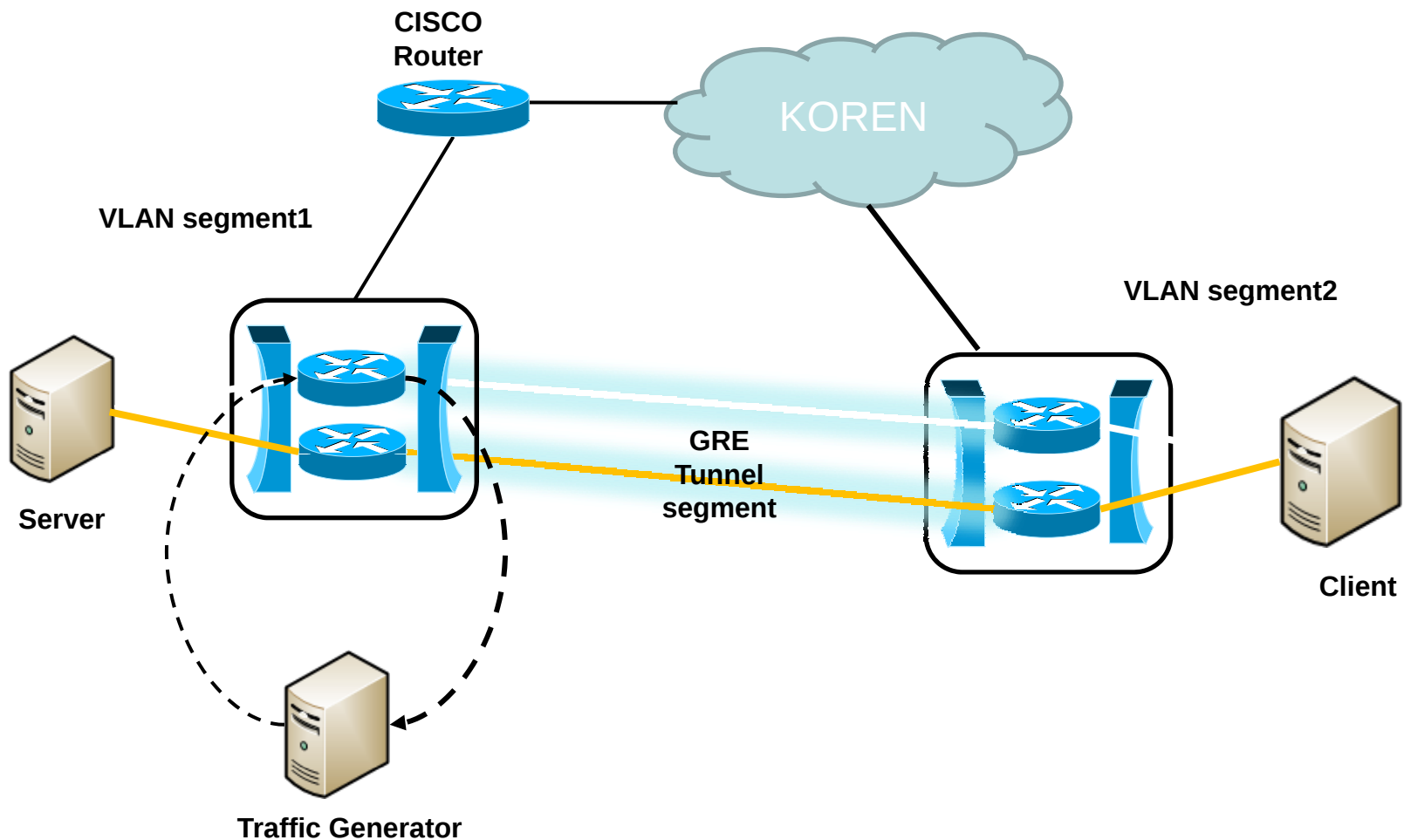


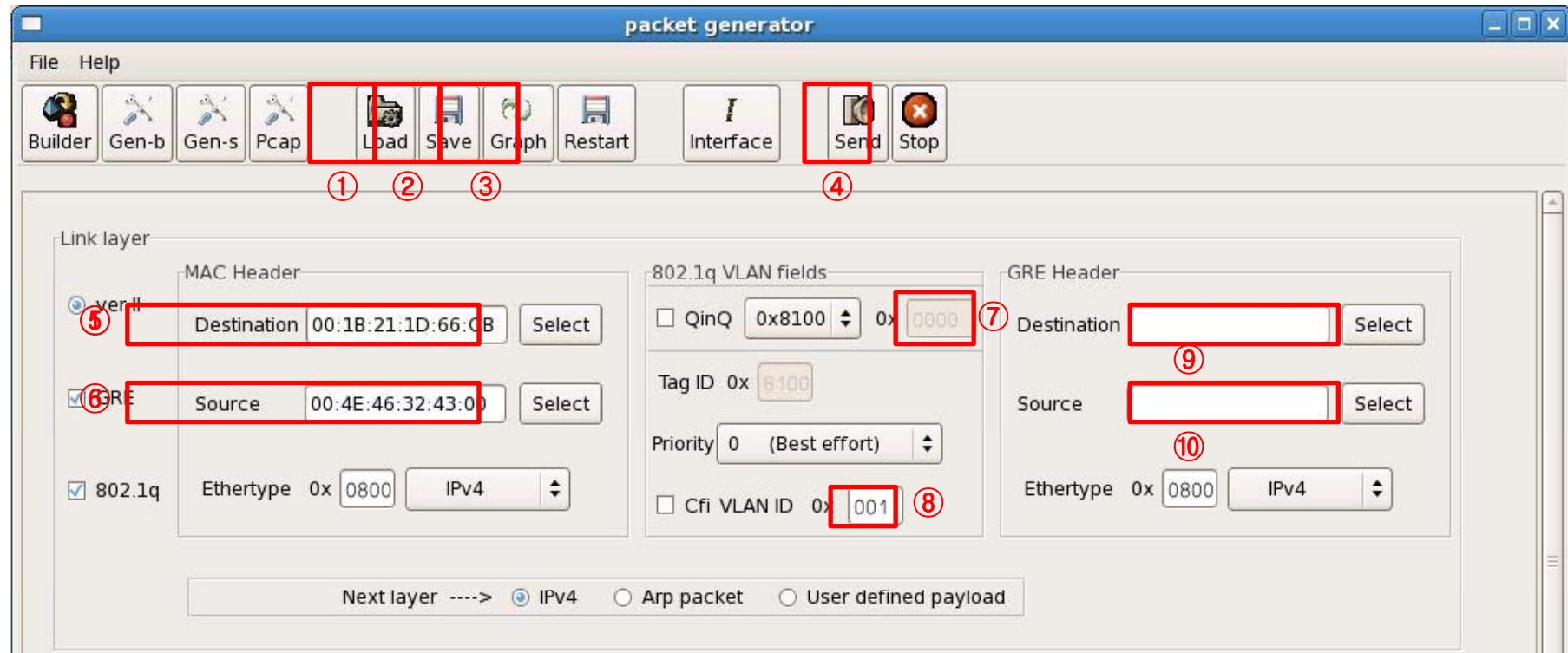
Resource Remote Monitoring

- You can also remotely control the assignment of critical resources (CPU, BW, Priority).



Remote Traffic Generator





1. 사용자가 가지고 있는 특정 pcap 파일을 읽어와서 로드
2. 사용자가 정의 해서 만든 패킷을 pcap으로 저장
3. Bandwidth 측정 결과를 실시간으로 그래프로 나타냄
4. Pcap을 큐에 실어서 특정 포트로 내보냄
5. 사용자가 만들 패킷의 MAC Header의 Destination을 정의
6. 사용자가 만들 패킷의 MAC Header의 Source를 정의
7. QinQ사용 시의 두번째 VLAN ID를 정의
8. 기본적인 첫 번째 VLAN ID를 정의
9. GRE 사용시 Destination의 IP address
10. GRE 사용시 Source의 IP address

프로그램 사용법

- 보내게 될 패킷의 L2, L3에 관한 빨간색 블록의 내용들을 채움.
- Save 버튼을 눌러서 pcap파일을 생성.
- Send 버튼을 눌러서 pcap파일을 NetFPGA기반의 PacketGenerator에 넘겨서 HW레벨에서 해당 패킷을 이용해 Generation 시킴.
- Graph버튼을 눌러서 실시간 Performance를 모니터링

IPv4 data

Version 0x 4 Header length 0x 5 TOS 0x 00 Select Total length 1000 ☒ Auto Identification 0x 1234

Flags 2 Select Fragment offset 0 TTL 255 Protocol 17 Reserved Header cks 0x ☒ Auto

Source IP 192.168.1.2 Select Destination IP 192.168.2.2 Select Options 0x

Next layer ----> ☒ UDP ☐ TCP ☐ ICMP ☐ IGMP ☐ User defined payload

1. 패킷의 사이즈를 조절
2. 패킷의 Source IP Address를 정의
3. 패킷의 Destination IP Address를 정의 상위 Layer를 선택
4. 기본 적인 첫 번째 VLAN ID를 정의

packet generator

File

Help

 Builder

 Gen-b

 Gen-s

 Pcap

 Load

 Save

 Graph

 Restart

 Interface

 Send

 Stop

Link layer

☒ ver II

☒ GRE

☒ 802.1q

MAC Header

Destination00:1B:21:1D:66:CBSelect

Source00:4E:46:32:43:00Select

Ethertype0x0800IPv4

802.1q VLAN fields

☐ QinQ0x81000x0000

Tag ID0x8100

Priority0(Best effort)

☐ Cfi VLAN ID0x001

GRE Header

DestinationSelect

SourceSelect

Ethertype0x0800IPv4

Next layer ----> ☒ IPv4 ☐ Arp packet ☐ User defined payload

IPv4 data

Version0x4Header length0x5TOS0x00SelectTotal length1000AutoIdentification0x1234

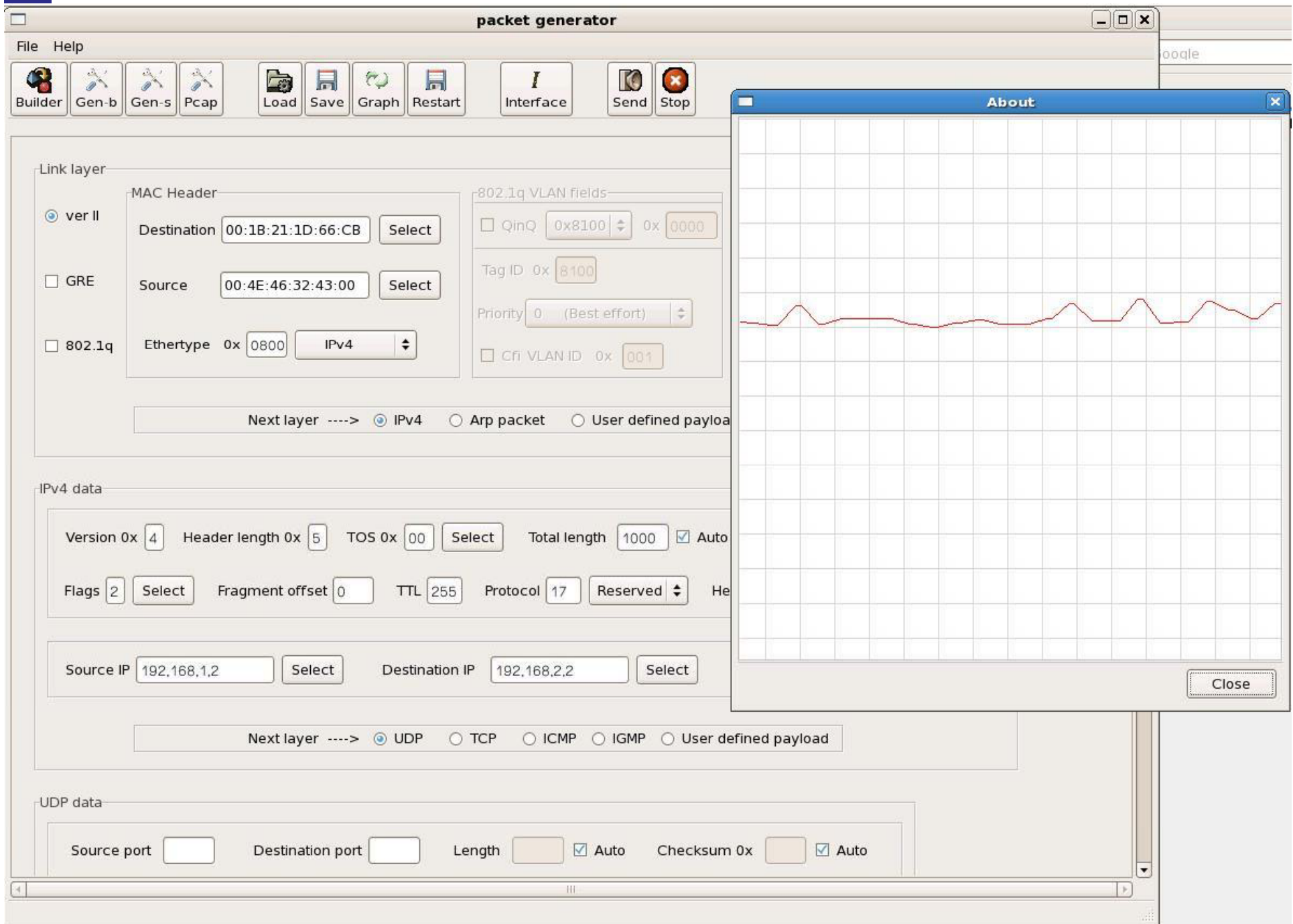
Flags2SelectFragment offset0TTL255Protocol17ReservedHeader cks0xAuto

Source IP192.168.1.2SelectDestination IP192.168.2.2SelectOptions0x

Next layer ----> ☒ UDP ☐ TCP ☐ ICMP ☐ IGMP ☐ User defined payload

UDP data

Source portDestination portLengthAutoChecksum0xAuto



What We Will Have in Future

