최근 네트워크 트렌드와 PacketNgin을 이용한 실습 그리고 전망

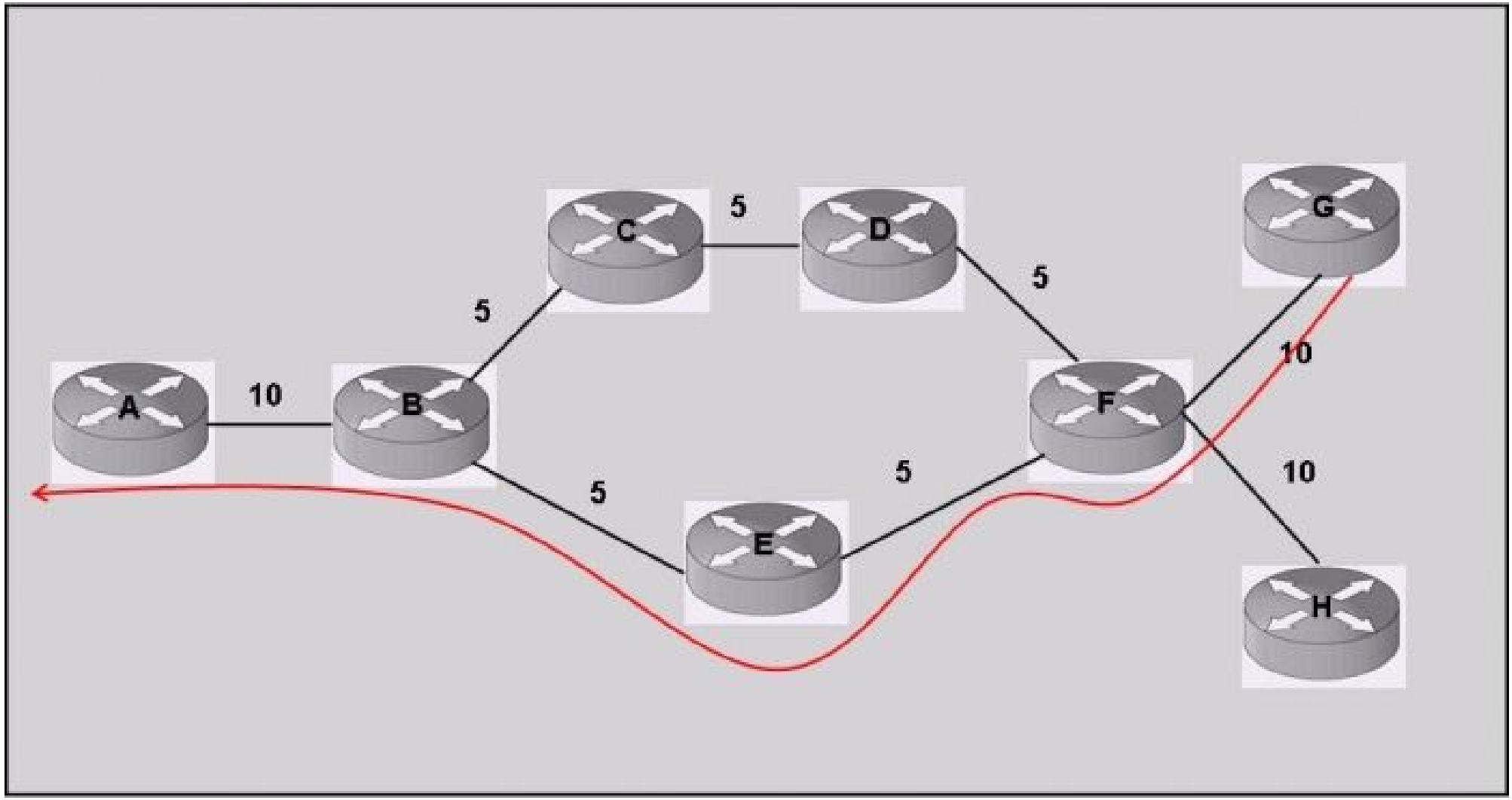
오픈프론티어 1기
김성민 Semih.Kim@gmail.com

목차

- 트렌드
  - Control and Data plane
  - SDN - Software Defined Network
  - NFV - Network Function Virtualization
- 실습
  - SDN - ONOS
  - NFV - PacketNgin
- 전망
  - OpenFlow의 한계
  - NFV의 한계
  - SDN과 NFV의 결합
  - 말랑말랑한 네트워크

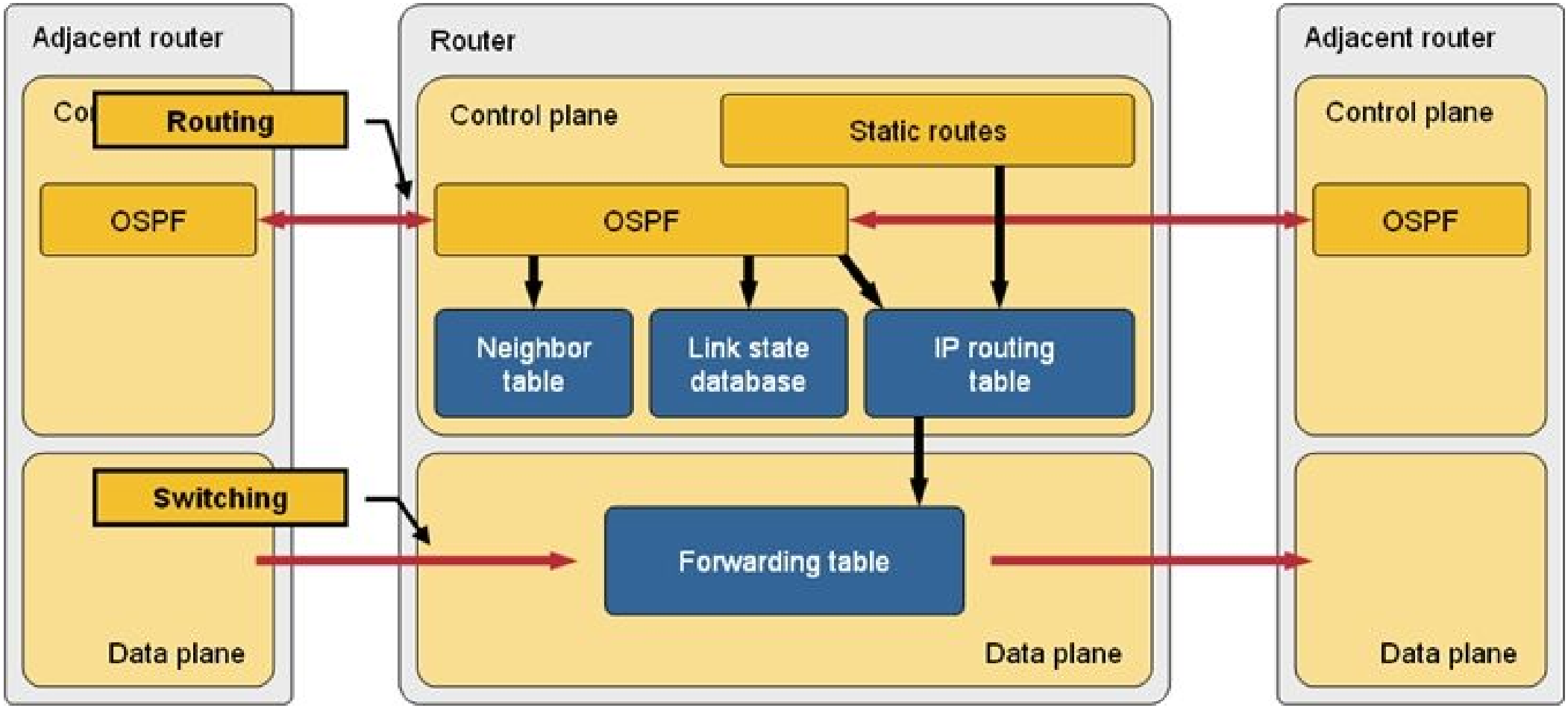
트렌드

Control and Data plane



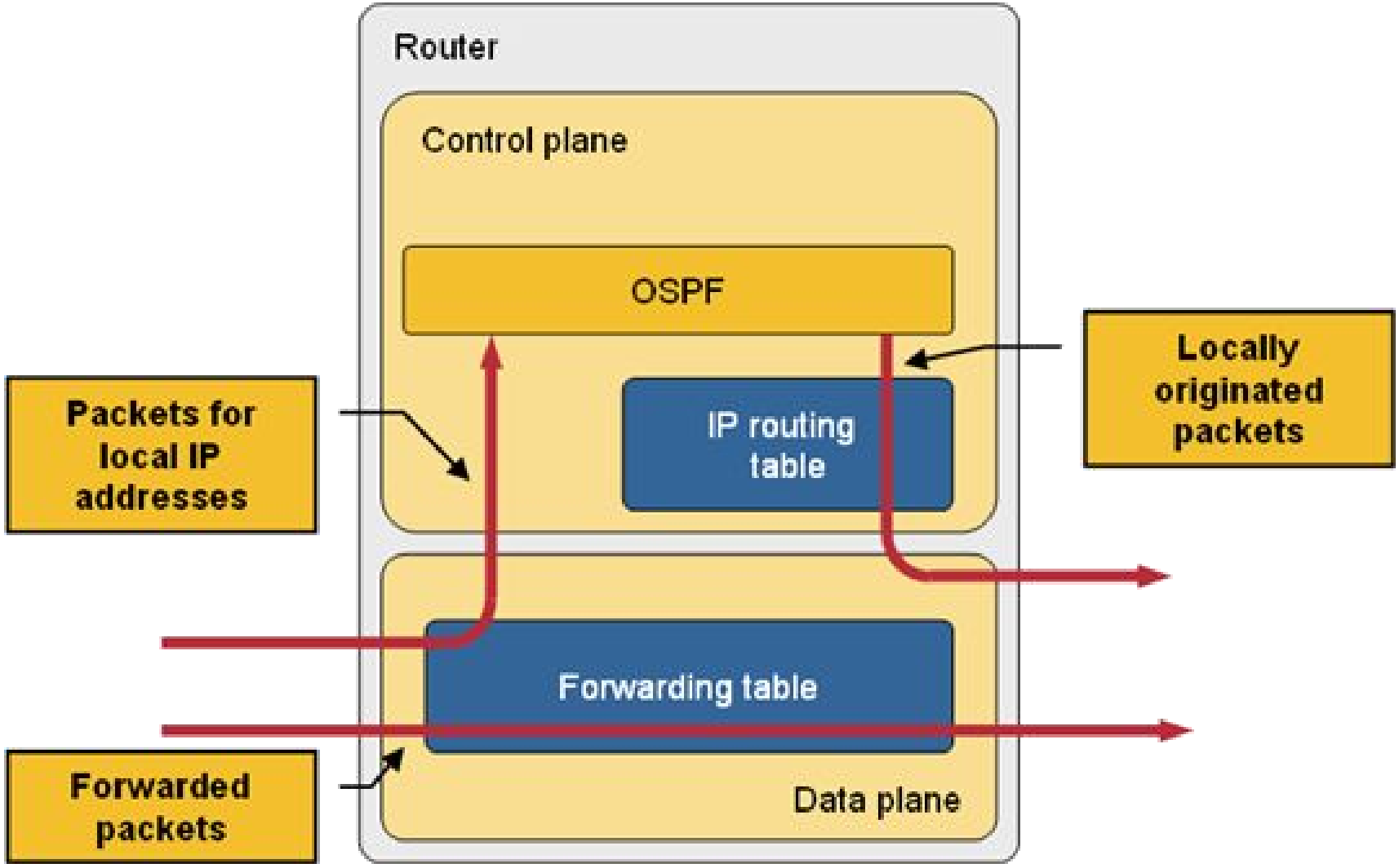
Source: http://gallery.techarena.in/showphoto.php/photo/21765

Control and Data plane

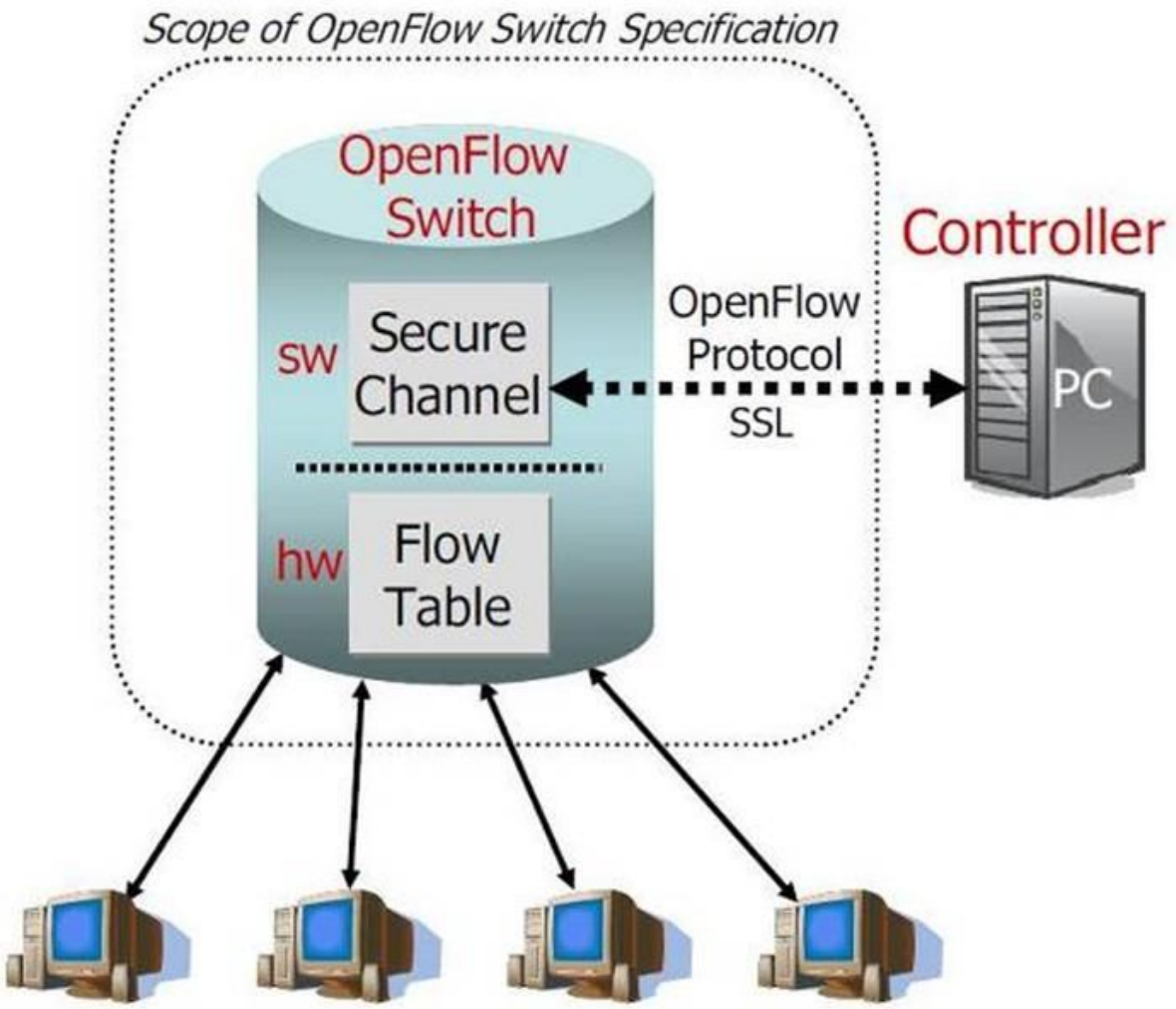


Source: http://wiki.nil.com/Control_and_Data_plane

Control and Data plane

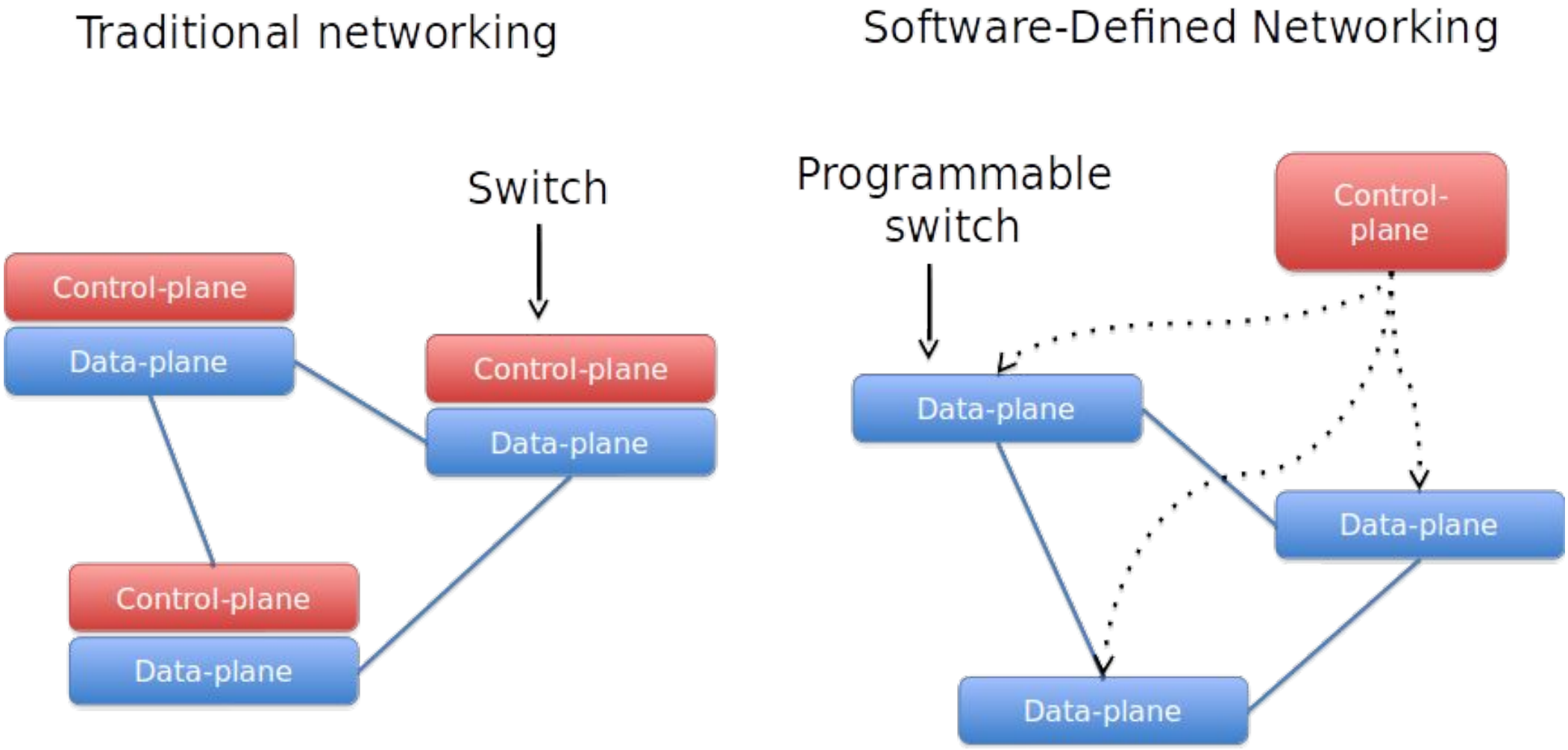


SDN - Software Defined Network



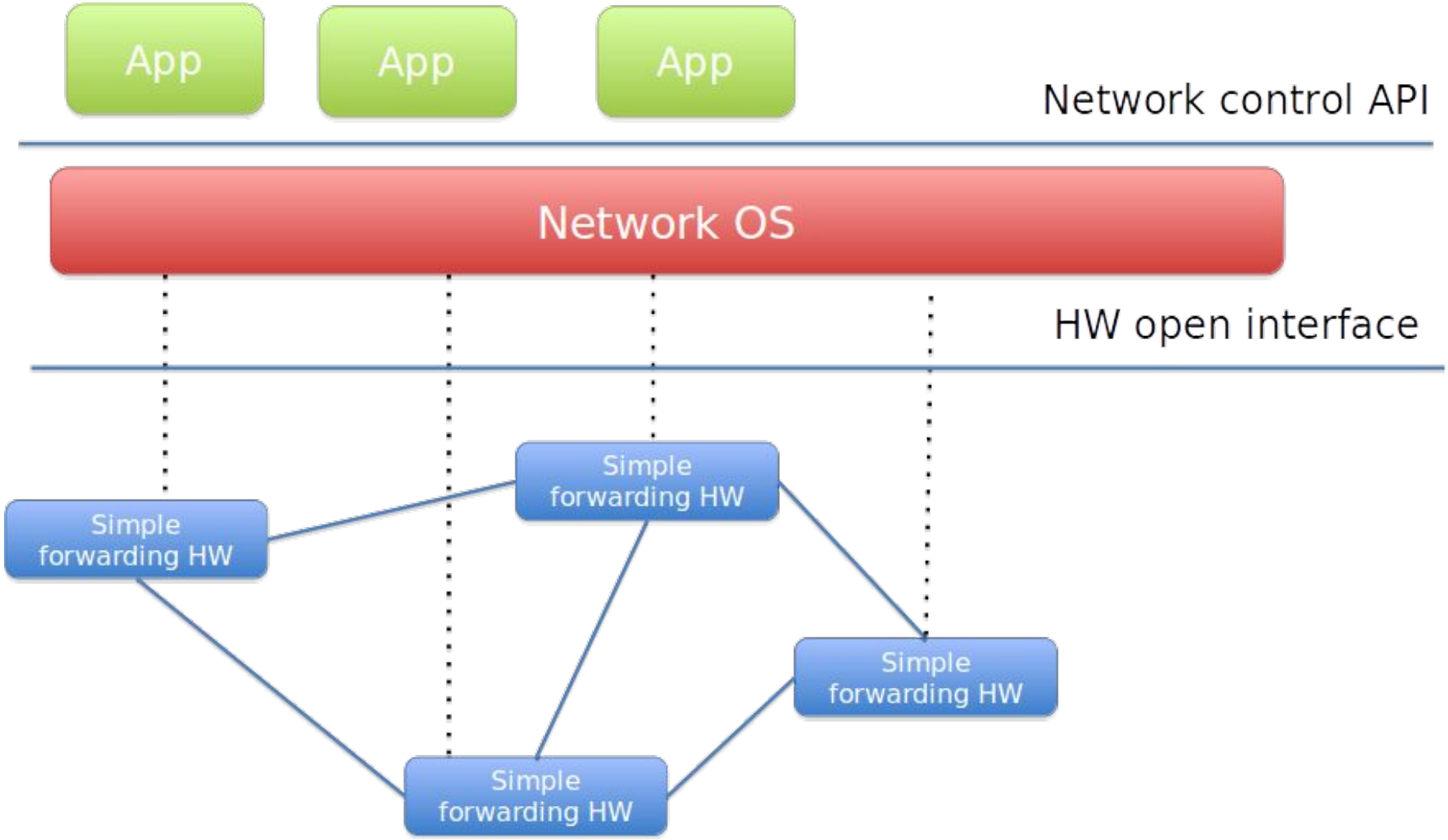
Source: OpenFlow 1.1 Specification

SDN - Software Defined Network



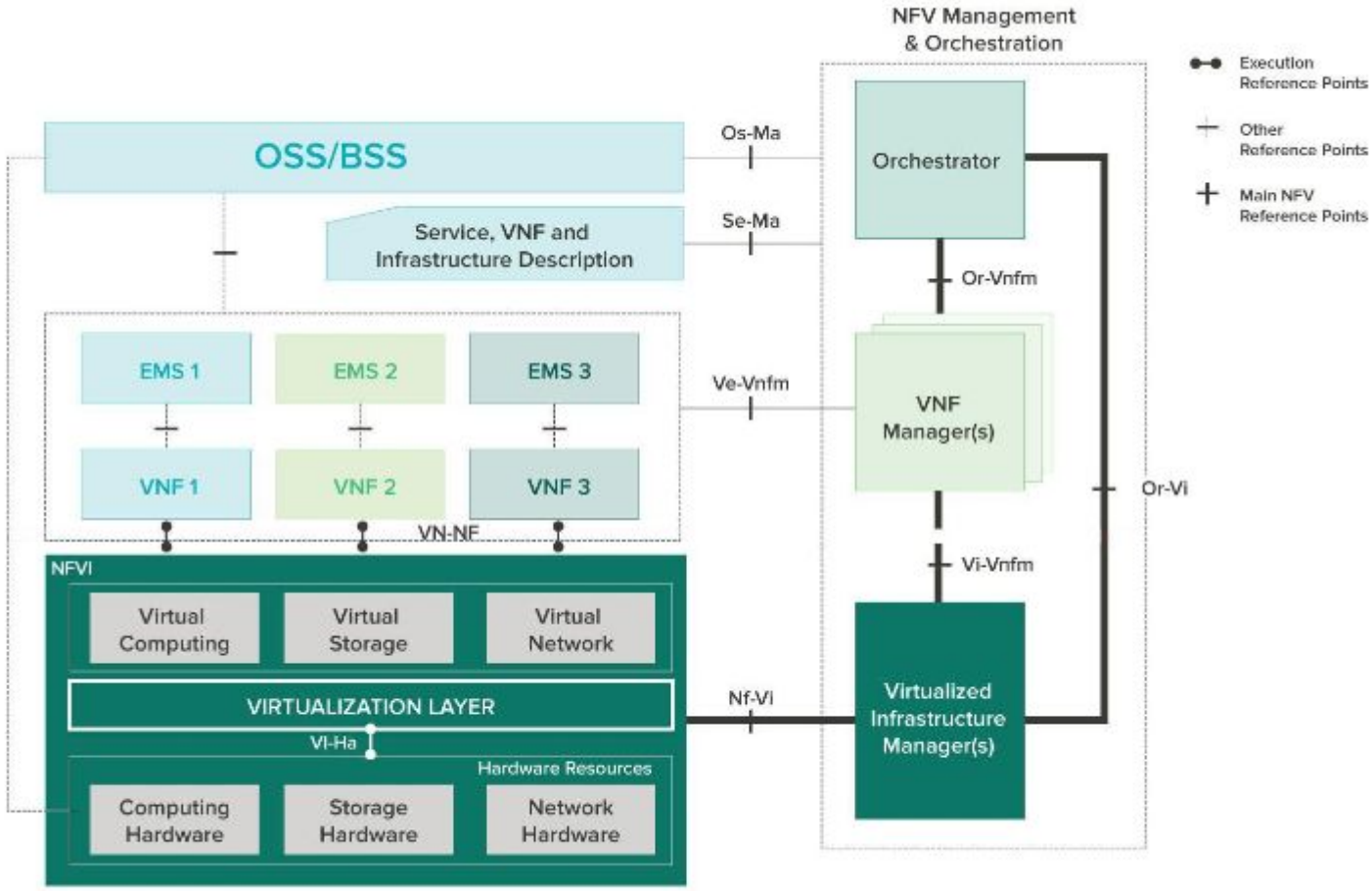
Source: G. Bianchi & A. Capone - SDN tutorial

SDN - Software Defined Network



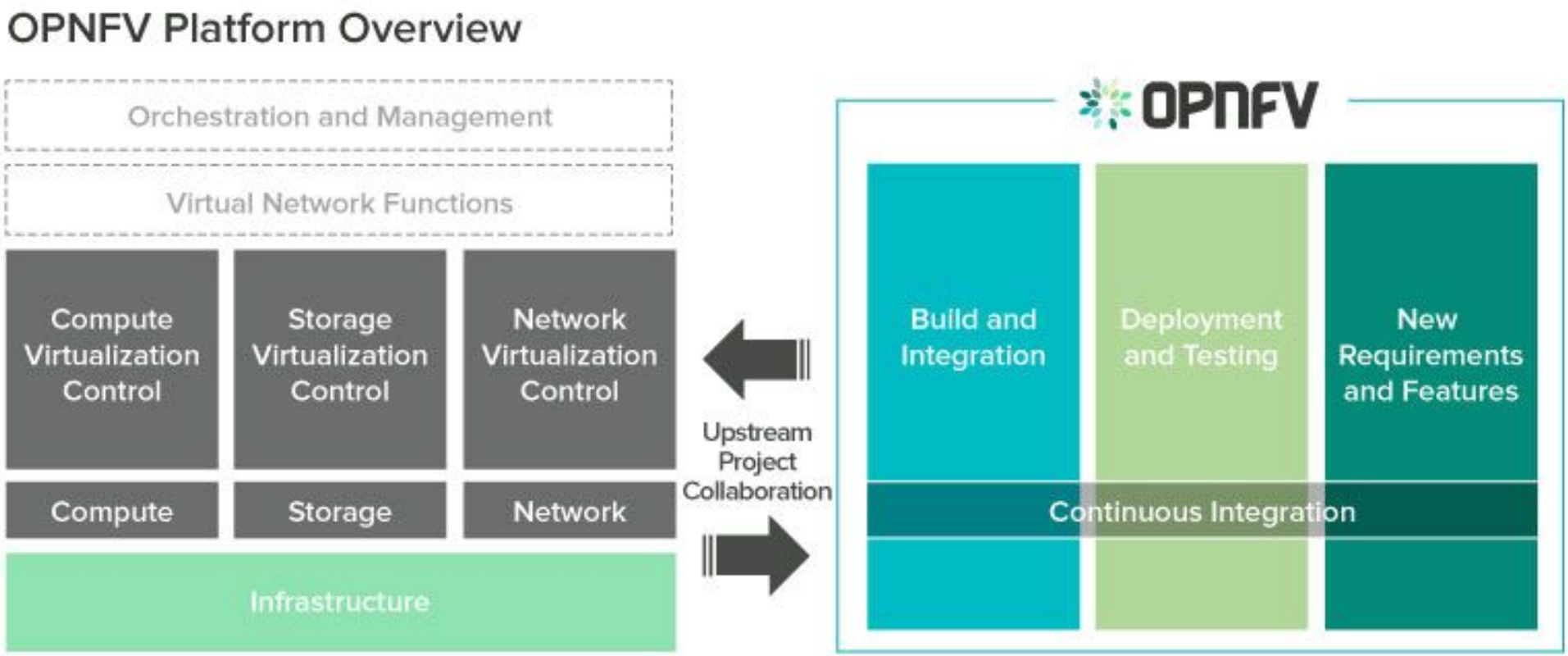
Source: G. Bianchi & A. Capone - SDN tutorial

NFV - Network Function Virtualization



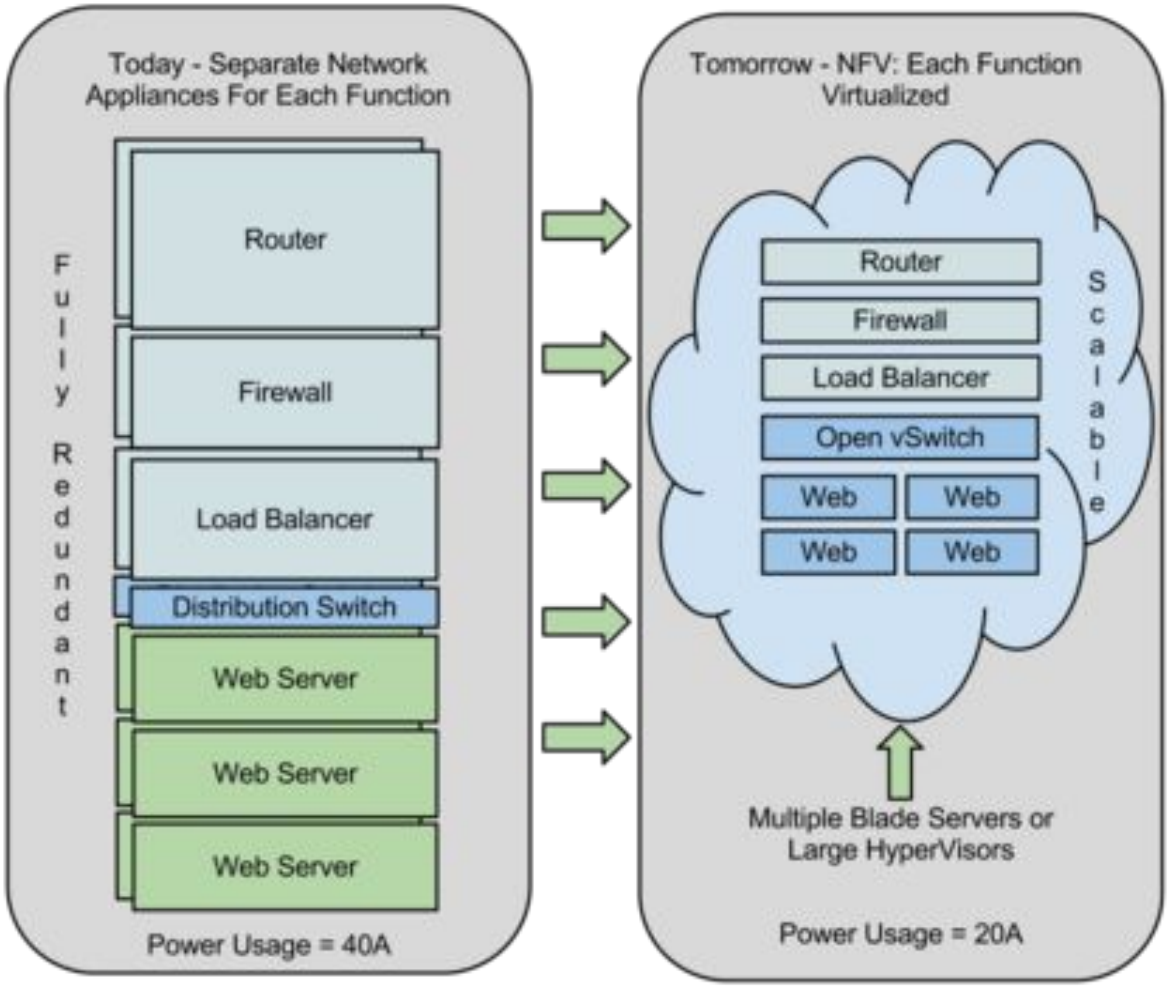
Source: https://www.opnfv.org/software/technical-overview

NFV - Network Function Virtualization



Source: https://www.opnfv.org/software/technical-overview

NFV - Network Function Virtualization



Source: http://wikibon.org/wiki/v/Network_Function_Virtualization_or_NFV_Explained

NFV - Network Function Virtualization

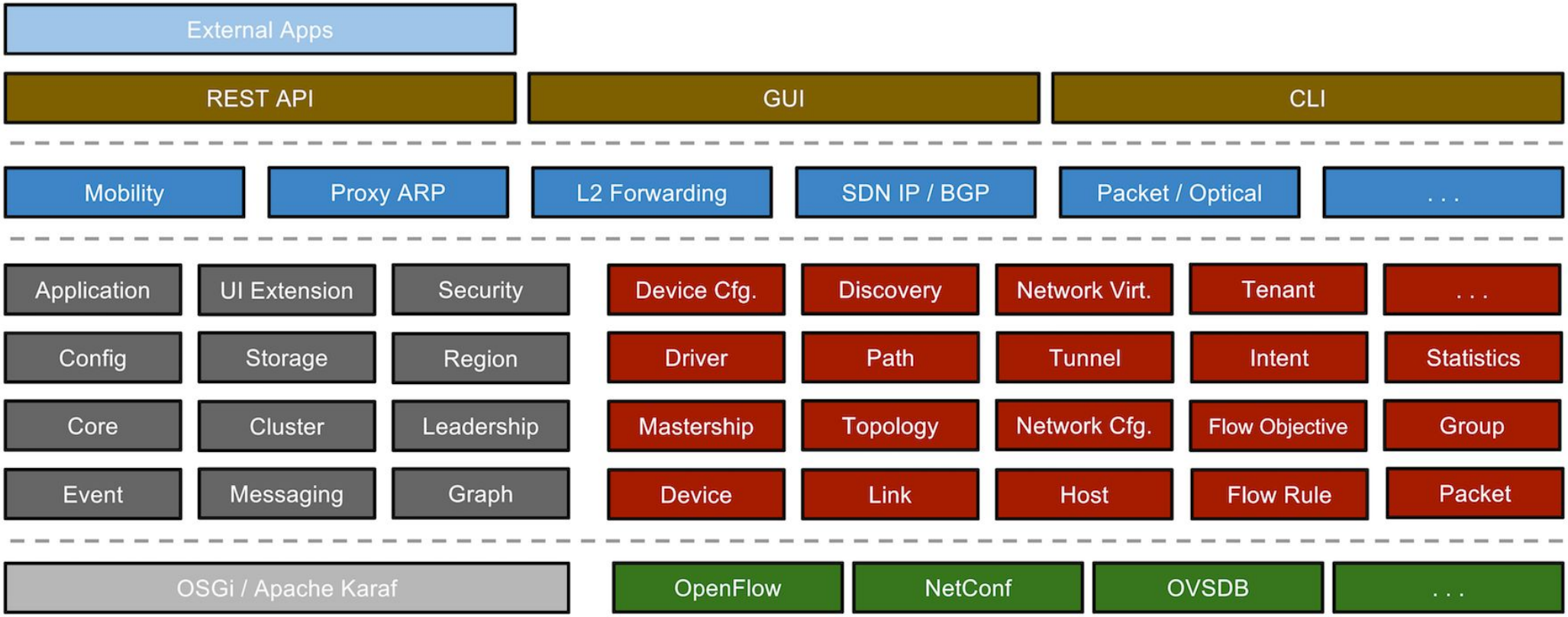
- Mobile Gateway
  - SGW - Serving Gateway
  - PGW - Packet Data Network Gateway
- Tunneling Gateway
  - VPN - Virtual Private Network
- Firewall
  - Web firewall
  - E-mail firewall
  - Intrusion detection system
- Monitoring
  - SLA - Service Level Agreement
- Traffic Control
  - Load balancers
  - WAN accelerators

실습

SDN - ONOS

- Architecture
- Example - ARP Response

ONOS - Architecture



ONOS Example - ARP Response

```
public void programArpRules(DeviceId deviceId, SegmentationId segmentationId,
                            Ip4Address routerAddress, Objective.Operation type) {
    log.info("Install ARP rules");
    TrafficSelector selector = DefaultTrafficSelector.builder()
            .add(Criteria.matchTunnelId(Long
                    .parseLong(segmentationId.toString())))
            .matchEthType(EthType.EtherType.ARP.ethType().toShort())
            .matchArpTpa(routerAddress)
            .build();

    TrafficTreatment treatment = DefaultTrafficTreatment.builder().punt()
            .build();

    ForwardingObjective.Builder objective = DefaultForwardingObjective
            .builder().withTreatment(treatment).withSelector(selector)
            .fromApp(appId).withFlag(ForwardingObjective.Flag.SPECIFIC)
            .withPriority(DEFAULT_PRIORITY);

    if (type.equals(Objective.Operation.ADD)) {
        flowObjectiveService.forward(deviceId, objective.add());
    } else {
        flowObjectiveService.forward(deviceId, objective.remove());
    }
}
```

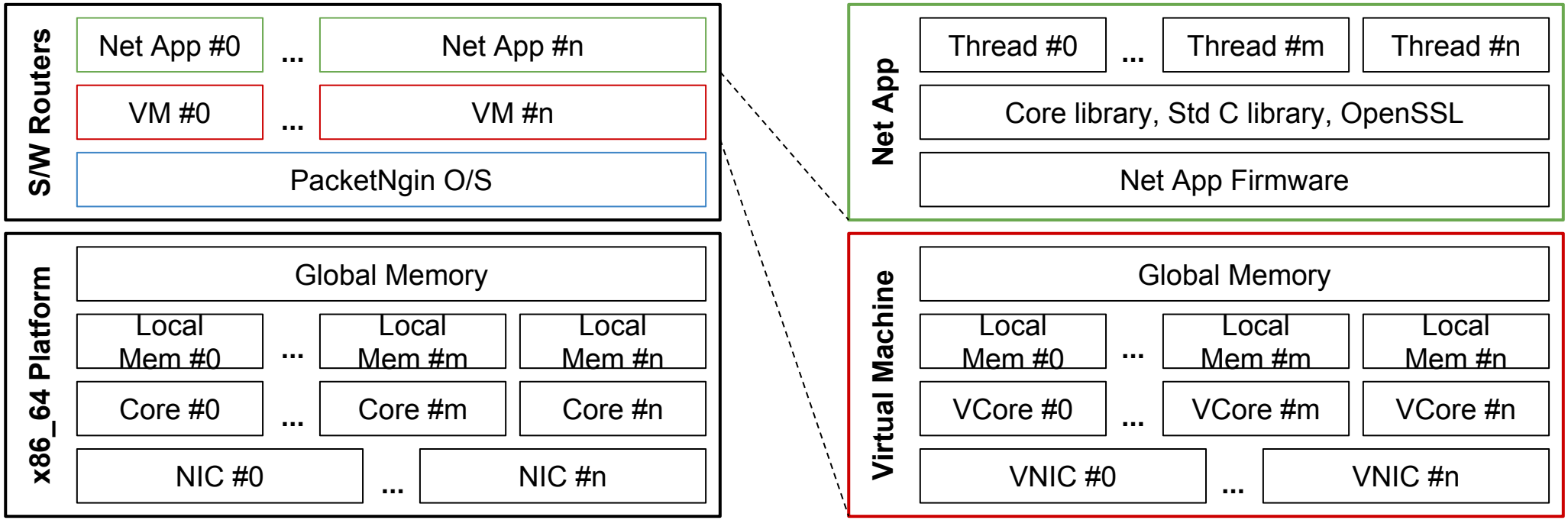
```
private class InternalPacketProcessor implements PacketProcessor {

    @Override
    public void process(PacketContext context) {
        if (context.isHandled()) {
            return;
        }

        InboundPacket pkt = context.inPacket();
        Ethernet ethernet = pkt.parsed();

        if (ethernet == null)
            return;

        if (ethernet.getEtherType() == Ethernet.TYPE_ARP) {

            ARP arp = (ARP) ethernet.getPayload();

            if (arp.getOpCode() == ARP.OP_REQUEST) {
                byte[] srcMacAddress = arp.getSenderHardwareAddress();
                byte[] srcIPAddress = arp.getSenderProtocolAddress();
                byte[] dstIPAddress = arp.getTargetProtocolAddress();

                // Searches the Dst MAC Address based on openstackPortMap
                log.trace("ARP request to : {}", Ip4Address.valueOf(dstIPAddress));

                RouterInterface routerInterface = routerInterfaceStore.values().stream()
                        .filter(e -> {
                            VirtualPort virtualPort = virtualPortService.getPort(e.portId());
                            if (virtualPort == null)
                                return false;

                            for (FixedIp fixedIp : virtualPort.fixedIps()) {
                                IpAddress ipAddress = IpAddress.valueOf(IpAddress.Version.INET, dstIPAddress);
                                if(fixedIp.ip().equals(ipAddress))
                                    return true;
                            }

                            return false;
                        })
                        .findAny().orElse(null);
```

```
MacAddress macAddress = null;
if (routerInterface != null) {
    VirtualPort virtualPort = virtualPortService.getPort(routerInterface.portId());
    macAddress = virtualPort.macAddress();
    log.info("ARP Repose from : {}", macAddress.toString());
} else {
    context.block();
    return;
}

// Creates a response packet
ARP arpReply = new ARP();
arpReply.setOpCode(ARP.OP_REPLY)
        .setHardwareAddressLength(arp.getHardwareAddressLength())
        .setHardwareType(arp.getHardwareType())
        .setProtocolAddressLength(arp.getProtocolAddressLength())
        .setProtocolType(arp.getProtocolType())
        .setSenderHardwareAddress(macAddress.toBytes())
        .setSenderProtocolAddress(dstIPAddress)
        .setTargetHardwareAddress(srcMacAddress)
        .setTargetProtocolAddress(srcIPAddress);

// Sends a response packet
ethernet.setDestinationMACAddress(srcMacAddress)
        .setSourceMACAddress(macAddress)
        .setEtherType(Ethernet.TYPE_ARP)
        .setPayload(arpReply);

TrafficTreatment.Builder builder = DefaultTrafficTreatment.builder();
builder.setOutput(pkt.receivedFrom().port());
OutboundPacket packet = new DefaultOutboundPacket(pkt.receivedFrom().deviceId(),
        builder.build(), ByteBuffer.wrap(ethernet.serialize()));
packetService.emit(packet);
```

NFV - PacketNgin

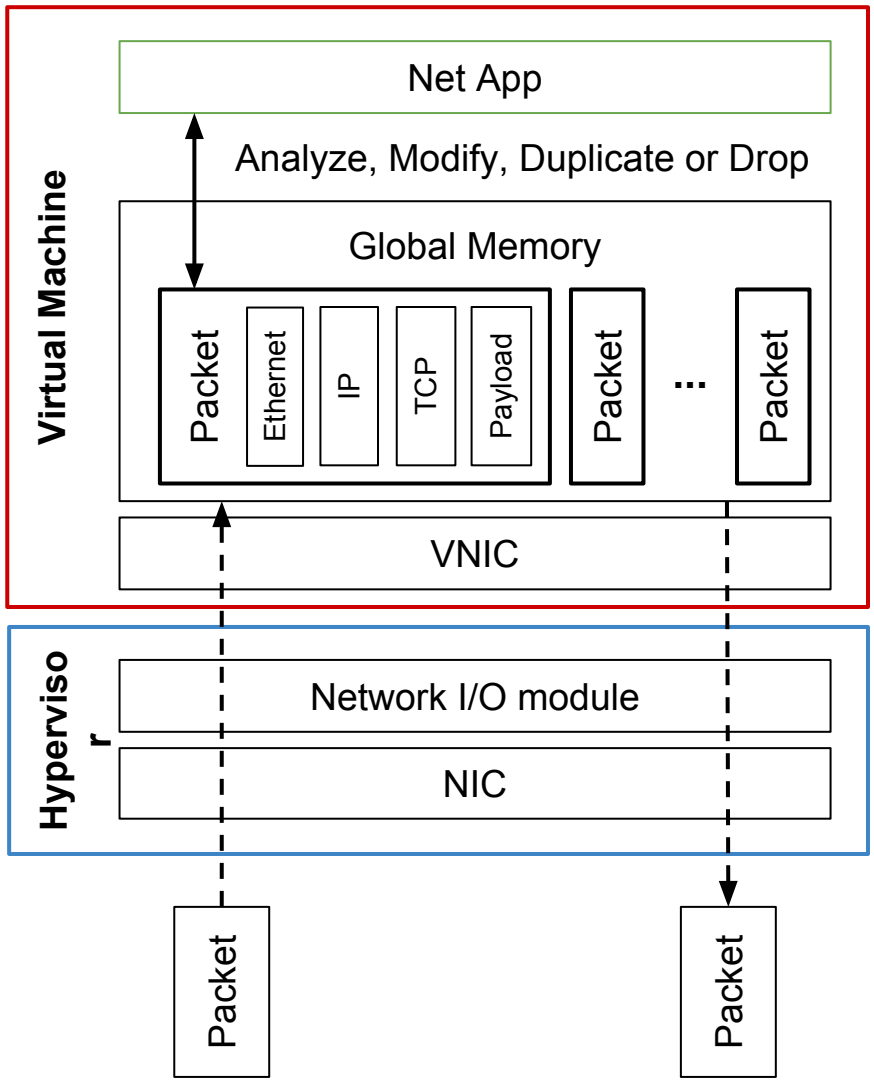
- Concept
- Key Functions
- Example - ARP (Address Resolution Protocol)
- Example - IP (Internet Protocol)
- Example - TCP (Transmission Control Protocol)

PacketNgin Concept



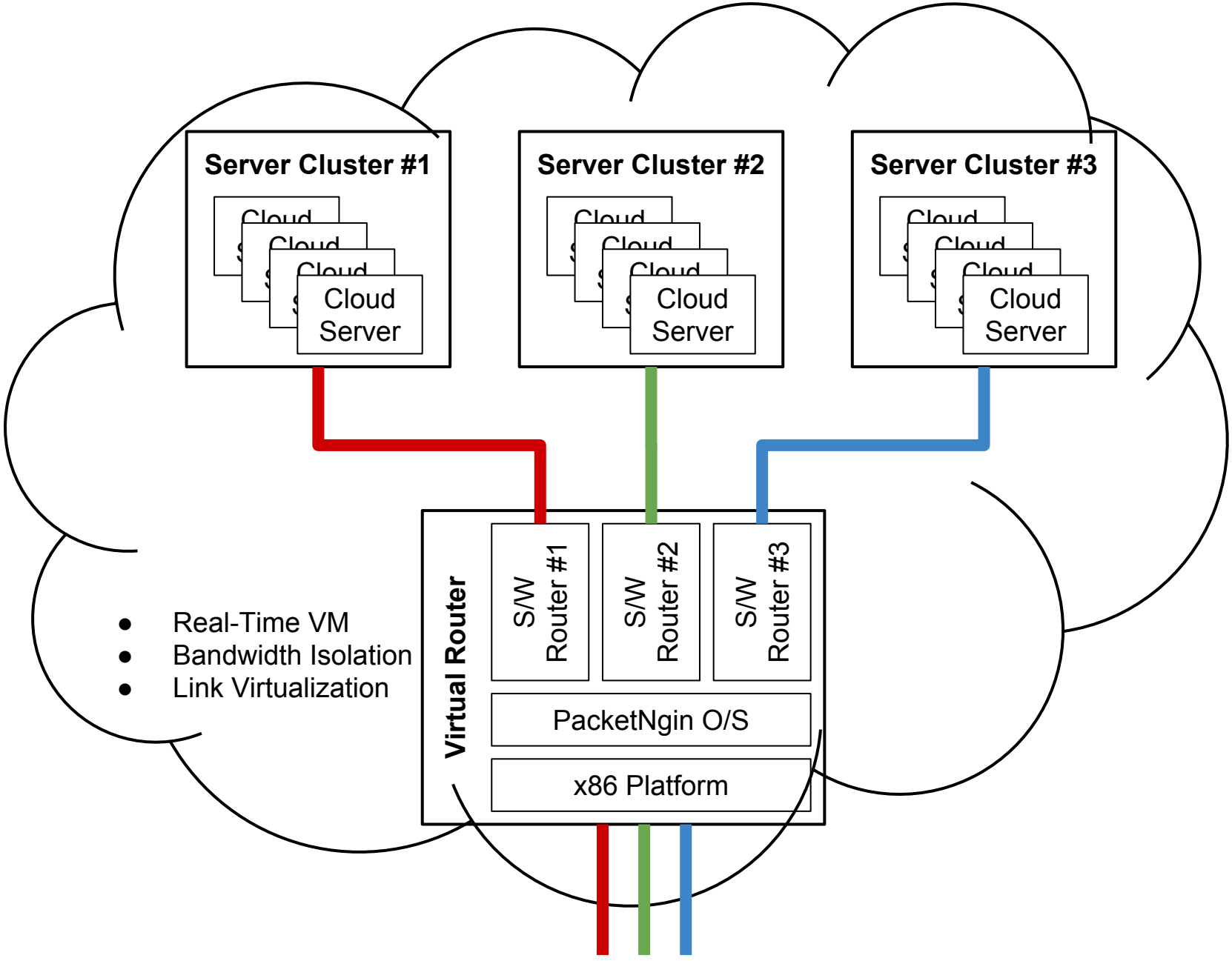
- PacketNgin is an O/S for Network Applications (Net App)
- PacketNgin Virtual Machine is a Real-Time Virtual Machine for Net Apps

PacketNgin - Key Functions

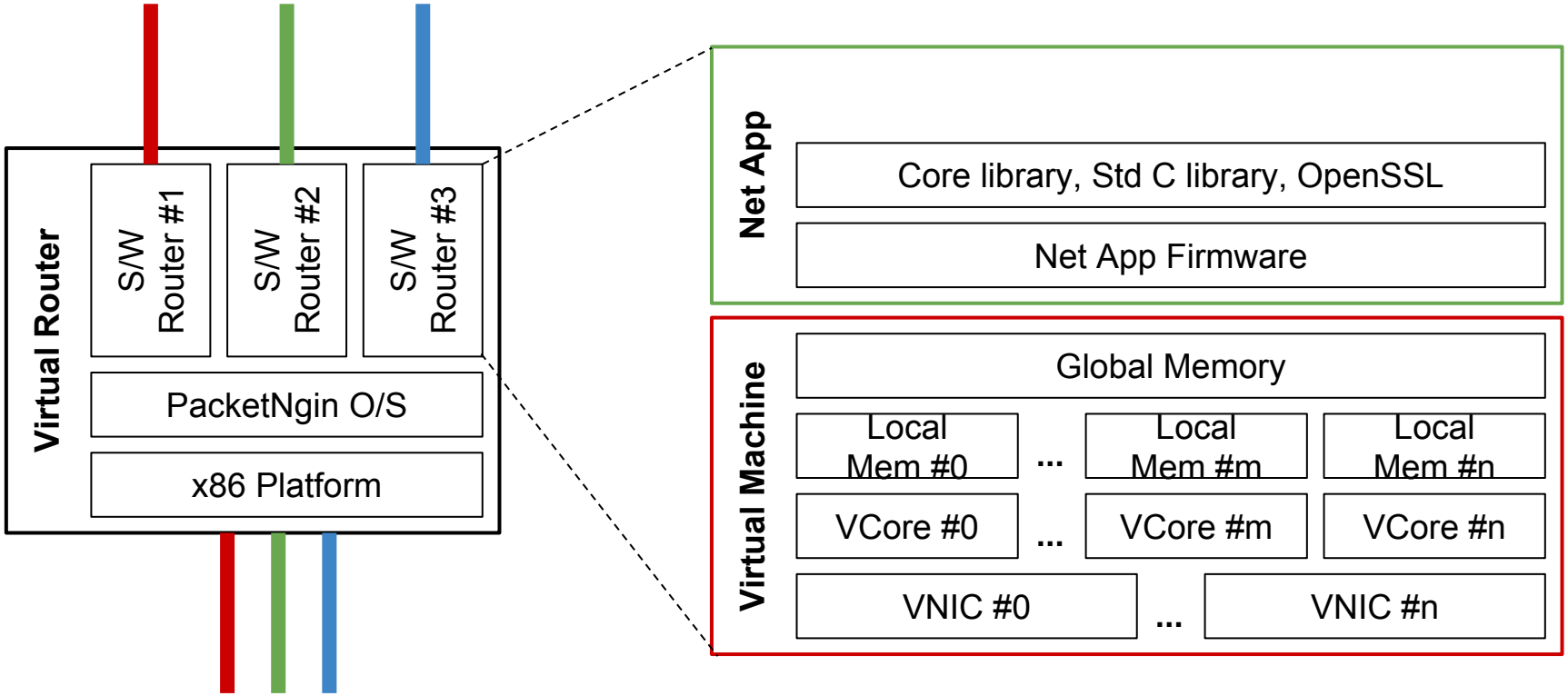


- PacketNgin O/S pass a packet to Net App's memory
- Net App can access the packet from OSI Level 2 to 7 (Ethernet, IP, TCP and payload)
- Net App can analyze, modify, duplicate or drop the packet

PacketNgin - Key Functions



PacketNgin - Key Functions



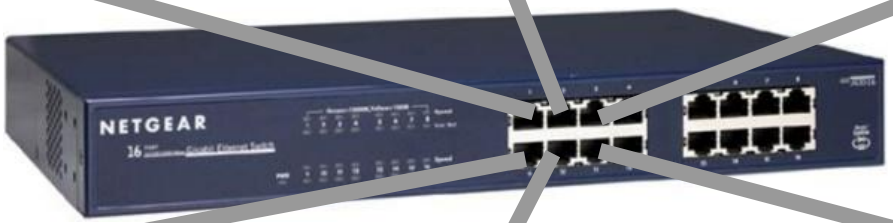
- PacketNgin O/S creates Real-Time VM
- Real-Time VM constists of
  - dedicated CPU Core, memory area
  - dedicated MAC addr, I/O buffer, bandwidth
  - virtualized link (hop-by-hop or tunnel)

PacketNgin Example - ARP

192.168.0.1
00:11:22:33:44:01

192.168.0.2
00:11:22:33:44:02

192.168.0.3
00:11:22:33:44:03



00:11:22:33:44:04
192.168.0.4

00:11:22:33:44:05
192.168.0.5

00:11:22:33:44:06
192.168.0.6

PacketNgin Example - ARP

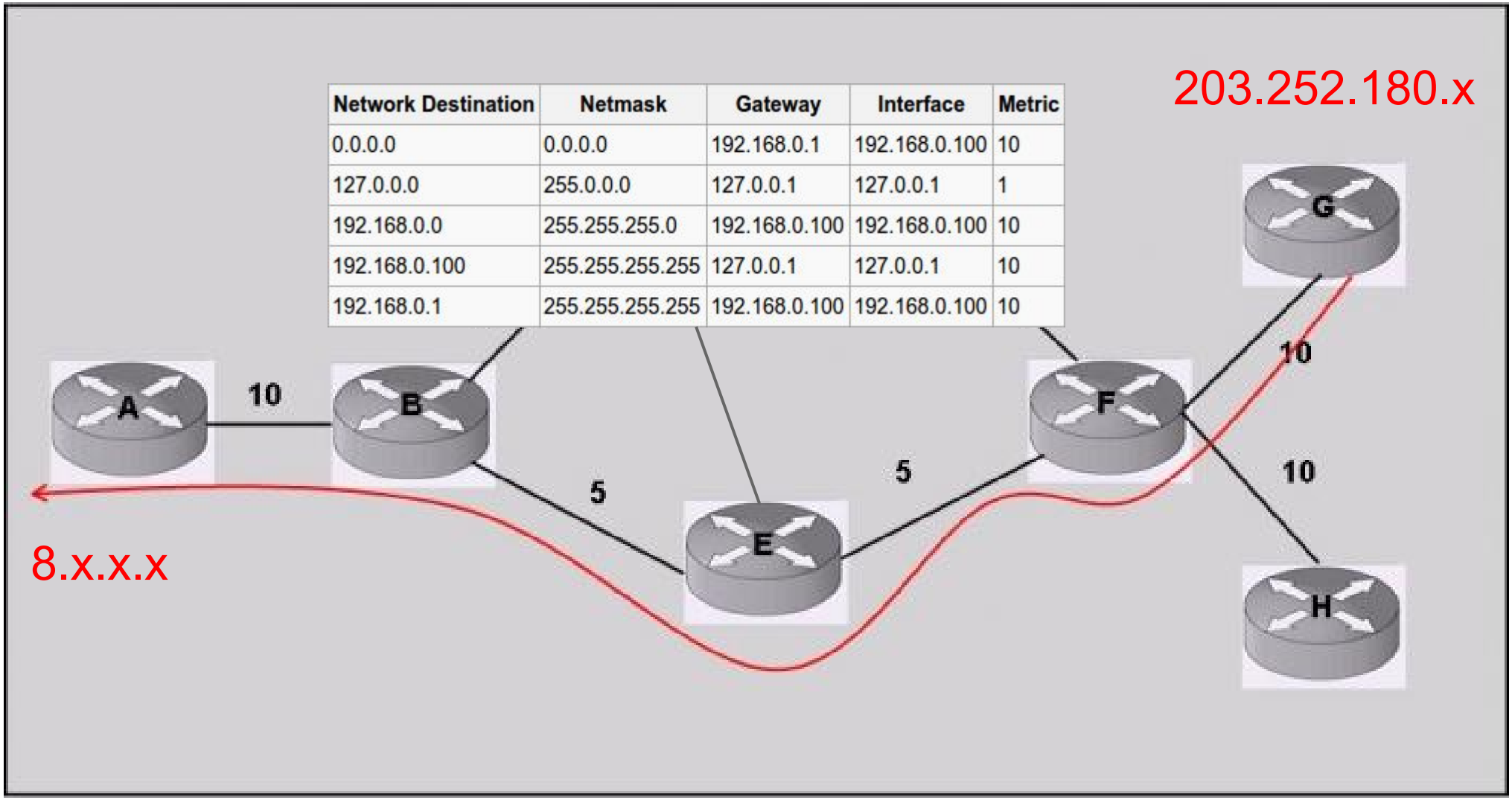
```
.657981 ARP, Ethernet (len 6), IPv4 (len 4), Requ
0x0000:  ffff ffff ffff bc5f f4bf 0155 0806 0001
0x0010:  0800 0604 0001 bc5f f4bf 0155 cbfc b4b4
0x0020:  0000 0000 0000 cbfc b4fe
.658348 ARP, Ethernet (len 6), IPv4 (len 4), Repl
0x0000:  bc5f f4bf 0155 0000 0c07 acb4 0806 0001
0x0010:  0800 0604 0002 0000 0c07 acb4 cbfc b4fe
0x0020:  bc5f f4bf 0155 cbfc b4b4 0000 0000 0000
0x0030:  0000 0000 0000 0000 0000 0000
```

```
static uint32_t address = 0xc0a8640a;    // 192.168.100.10

void process(NetworkInterface* ni) {
        Packet* packet = ni_input(ni);
        if(!packet)
                return;

        Ether* ether = (Ether*)(packet->buffer + packet->start);

        if(endian16(ether->type) == ETHER_TYPE_ARP) {
                // ARP response
                ARP* arp = (ARP*)ether->payload;
                if(endian16(arp->operation) == 1 && endian32(arp->tpa) == address) {
                        ether->dmac = ether->smac;
                        ether->smac = endian48(ni->mac);
                        arp->operation = endian16(2);
                        arp->tha = arp->sha;
                        arp->tpa = arp->spa;
                        arp->sha = ether->smac;
                        arp->spa = endian32(address);

                        ni_output(ni, packet);
                        packet = NULL;
                }
        }

        if(packet)
                ni_free(packet);
}
```

PacketNgin Example - ARP

```
ubuntu@ubuntu-VirtualBox:~/sdk$ arping 192.168.100.10 -Itap0
ARPING 192.168.100.10 from 192.168.100.200 tap0
Unicast reply from 192.168.100.10 [07:49:CA:5D:F3:EA]  405.598ms
Unicast reply from 192.168.100.10 [07:49:CA:5D:F3:EA]  365.855ms
Unicast reply from 192.168.100.10 [07:49:CA:5D:F3:EA]  282.799ms
Unicast reply from 192.168.100.10 [07:49:CA:5D:F3:EA]  203.837ms
^CSent 5 probes (1 broadcast(s))
Received 4 response(s)
ubuntu@ubuntu-VirtualBox:~/sdk$
```

PacketNgin Example - IP



Source: http://en.wikipedia.org/wiki/Routing_table

PacketNgin Example - IP

IPv4 Header Format

Offsets Octet	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Version				IHL				DSCP						ECN		Total Length															
4	32	Identification																Flags			Fragment Offset											
8	64	Time To Live								Protocol								Header Checksum															
12	96	Source IP Address																															
16	128	Destination IP Address																															
20	160	Options (if IHL > 5)																															

PacketNgin Example - IP

Bits	0–7	8–15	16–23	24–31
0	Type	Code	Checksum	
32	Rest of Header			

PacketNgin Example - IP

```
if(endian16(ether->type) == ETHER_TYPE_IPv4) {
 IP* ip = (IP*)ether->payload;

 if(ip->protocol == IP_PROTOCOL_ICMP && endian32(ip->destination) == address) {
        ICMP* icmp = (ICMP*)ip->body;
        // Echo reply
        if(icmp->type == 8) {
                icmp->type = 0;
                icmp->checksum = 0;
                icmp->checksum = endian16(checksum(icmp, packet->end - packet->start - ETHER_LEN - IP_LEN));

                ip->destination = ip->source;
                ip->source = endian32(address);
                ip->ttl = endian8(64);
                ip->checksum = 0;
                ip->checksum = endian16(checksum(ip, ip->ihl * 4));

                ether->dmac = ether->smac;
                ether->smac = endian48(ni->mac);

                ni_output(ni, packet);
                packet = NULL;
        } else if(icmp->type == 0) {
                clock_t time = clock() - *(clock_t*)(icmp->body);
                printf("Echo reply received. time : %d.%d ms\n", time/1000, time%1000);
                packet = NULL;
        }
 }
```

PacketNgin Example - IP

```
Starting VM...
true
***** vmid=1 thread=0 standard Output *****
Thread 0 bootting
Echo request sended
Echo reply received. time : 25.708 ms
Echo request sended
Echo reply received. time : 0.101 ms
Echo request sended
Echo reply received. time : 0.88 ms
Echo request sended
Echo reply received. time : 20.313 ms
Echo request sended
Echo reply received. time : 23.993 ms
Echo request sended
Echo reply received. time : 20.345 ms
Echo request sended
Echo reply received. time : 16.600 ms
Echo request sended
Echo reply received. time : 0.55 ms
Echo request sended
Echo reply received. time : 20.169 ms
Echo request sended
Echo reply received. time : 7.629 ms
Echo request sended
Echo reply received. time : 0.53 ms
```

PacketNgin Example - TCP

Offsets	Octet	0								1								2								3							
Octet	Bit	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
0	0	Source port																Destination port														
4	32	Sequence number																														
8	64	Acknowledgment number (if ACK set)																														
12	96	Data offset				Reserved 0 0 0				NS	CWR	ECE	URG	ACK	PSH	RST	SYN	FIN	Window Size													
16	128	Checksum																	Urgent pointer (if URG set)													
20 ...	160 ...	Options (if *data offset* > 5. Padded at the end with "0" bytes if necessary.) ...																														

- Data offset - Header length / 4
- NS, CWR, ECE - Explicit Congestion Notification
- URG - Urgent, ACK - Acknowledgement, PSH - Push to application right now
- RST - Reset the connection, SYN - Synchronize the seq number, FIN - Close
- Window Size - Sender’s buffer size (generally)
- Urgent pointer -

Source: http://en.wikipedia.org/wiki/Transmission_Control_Protocol

PacketNgin Example - TCP

```
if(ip->protocol == IP_PROTOCOL_TCP) {
 TCP* tcp = (TCP*)ip->body;

 if(endian16(tcp->destination) == 80) {
        printf("Inbound  : source=%u, destination=%u, seq=%u, ack=%d, syn=%d, fin=%d\n",
                endian16(tcp->source), endian16(tcp->destination), endian32(tcp->sequence), tcp->ack, tcp->syn,


        ether->dmac = endian48(linux_mac);
        ether->smac = endian48(ni->mac);

        ip->source = endian32(address);
        ip->destination = endian32(linux_address);
        ip->checksum = 0;
        ip->checksum = endian16(checksum(ip, ip->ihl * 4));

        tcp_source_temp = tcp->source;
        tcp->destination = endian16(8080);
        tcp->checksum = 0;
        tcp_pack(packet, endian16(ip->length) - ip->ihl * 4 - TCP_LEN);

        ni_output(ni, packet);
        packet = NULL;
 } else if(endian16(tcp->source) == 8080) {
```

PacketNgin Example - TCP

```
              onkeyup="searchBox.OnSearchFieldChange(event)"/>
          </span><span class="right">
            <a id="MSearchClose" href="javascript:searchBox.CloseResultsWindow()"><img id="MSearchClos
eImg" border="0" src="search/close.png" alt=""/></a>
          </span>
        </div>
      </li>
    </ul>
  </div>
</div><!-- top -->
<!-- window showing the filter options -->
<div id="MSearchSelectWindow"
     onmouseover="return searchBox.OnSearchSelectShow()"
     onmouseout="return searchBox.OnSearchSelectHide()"
     onkeydown="return searchBox.OnSearchSelectKey(event)">
<a class="SelectItem" href="javascript:void(0)" onclick="searchBox.OnSelectItem(0)"><span class="Selec
tionMark"> </span>All</a><a class="SelectItem" href="javascript:void(0)" onclick="searchBox.OnSel
ectItem(1)"><span class="SelectionMark"> </span>Classes</a><a class="SelectItem" href="javascript
:void(0)" onclick="searchBox.OnSelectItem(2)"><span class="SelectionMark"> </span>Namespaces</a><
a class="SelectItem" href="javascript:void(0)" onclick="searchBox.OnSelectItem(3)"><span class="Select
ionMark"> </span>Files</a><a class="SelectItem" href="javascript:void(0)" onclick="searchBox.OnSe
lectItem(4)"><span class="SelectionMark"> </span>Functions</a><a class="SelectItem" href="javascr
ipt:void(0)" onclick="searchBox.OnSelectItem(5)"><span class="SelectionMark"> </span>Variables</a
><a class="SelectItem" href="javascript:void(0)" onclick="searchBox.OnSelectItem(6)"><span class="Sele
ctionMark"> </span>Typedefs</a><a class="SelectItem" href="javascript:void(0)" onclick="searchBox
.OnSelectItem(7)"><span class="SelectionMark"> </span>Enumerations</a><a class="SelectItem" href=
"javascript:void(0)" onclick="searchBox.OnSelectItem(8)"><span class="SelectionMark"> </span>Enum
erator</a><a class="SelectItem" href="javascript:void(0)" onclick="searchBox.OnSelectItem(9)"><span cl
ass="SelectionMark"> </span>Macros</a><a class="SelectItem" href="javascript:void(0)" onclick="se
archBox.OnSelectItem(10)"><span class="SelectionMark"> </span>Pages</a></div>
```

전망

전망 1 - OpenFlow의 한계

- H/W OpenFlow Switch - 1.1 ~ 1.3
- S/W OpenFlow Switch - 1.1 ~
- OpenFlow 2.0?

전망 2 - NFV의 한계

- NVF Reference Implementation - OPNFV
- KVM
- Network Throughput and Response time

전망 3 - SDN과 NFV의 결합

- OpenFlow는 Flow Control을 위한 표준 API로 (not for performance)
- NFV는 SDN의 확장 기능으로

전망 4 - 말랑말랑한 네트워크

- H/W 보다는 S/W 중심의 네트워크
- Flow Control은 OpenFlow로
- Packet Processing은 NFV(또는 유사 기술)로
- VM Migration
- Cluster Migraion

Thank you for ...

Thank you!

(주)구름네트웍스
김성민 semih@gurum.cc