Project

Implementation of Virtual Customer Premise Equipment

Deadline: 2022/04/25 (MON) 23:59

Outline

- Objective
- Environment
- Customer Premise Equipment
- Virtual Customer Premise Equipment
- Open vSwitch
- Project
- Appendix

Outline

- Objective
- Environment
- Customer Premise Equipment
- Virtual Customer Premise Equipment
- Open vSwitch
- Project
- Appendix

Objective

- Learn and Practice the concepts of
  - NAT Traversal
  - GRE Tunnel
  - Remote Gateway and DHCP Server
  - Docker Container
  - Virtual Networking
  - Open vSwitch
    - Group table
    - Meter table

Outline

- Objective
- Environment
- Customer Premise Equipment
- Virtual Customer Premise Equipment
- Open vSwitch
- Project
- Appendix

Environment

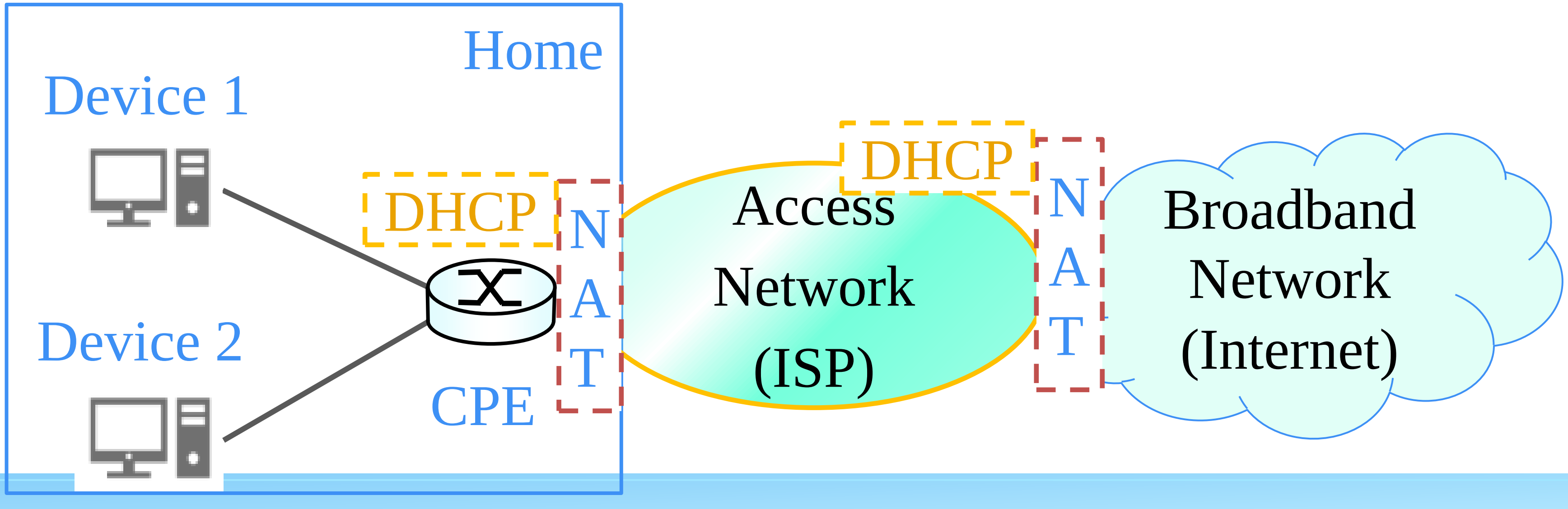
- Environment: same as Lab4
  - Ubuntu 16.04
  - Docker
- Auto Tunnel Creation Program from lab4

Outline

- Objective
- Environment
- Customer Premise Equipment
- Virtual Customer Premise Equipment
- Open vSwitch
- Project
- Appendix

Custom Premise Equipment

- An internet Access Gateway for devices
  - Acquires an IP address from an access network(ISP)
    - Likely a private IP
  - Runs DHCP Server
    - Assigns private IP addresses to home devices

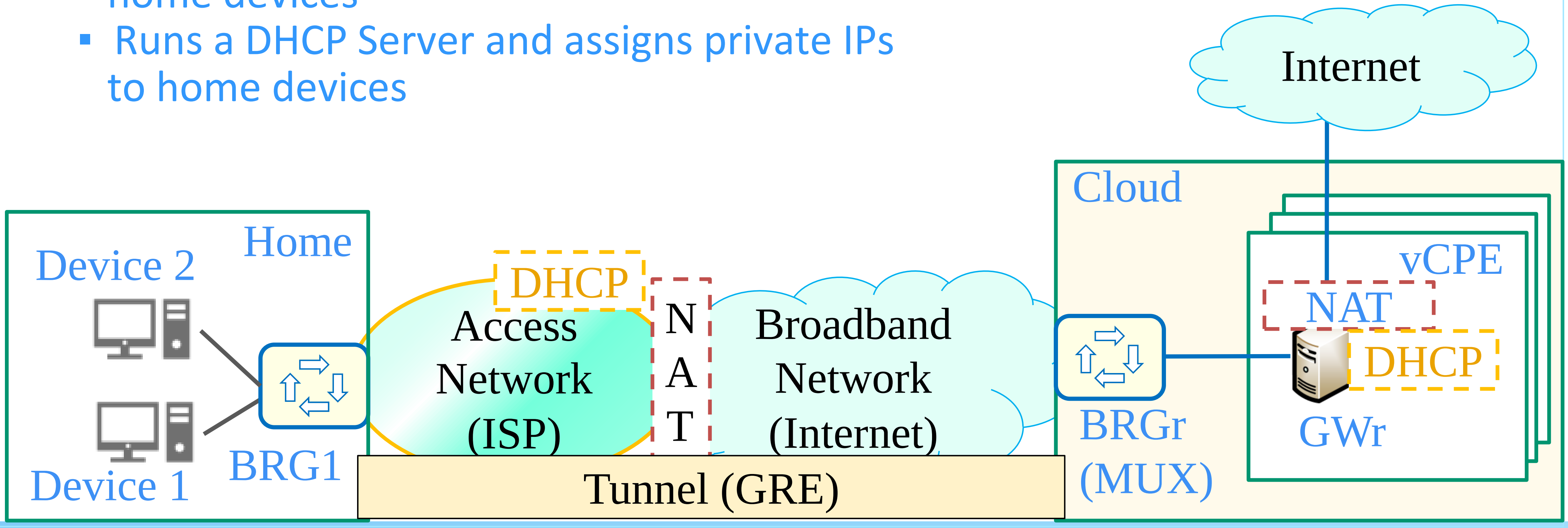


Outline

- Objective
- Environment
- Customer Premise Equipment
- Virtual Customer Premise Equipment
  - Architecture
  - Overview
- Open vSwitch
- Project
- appendix

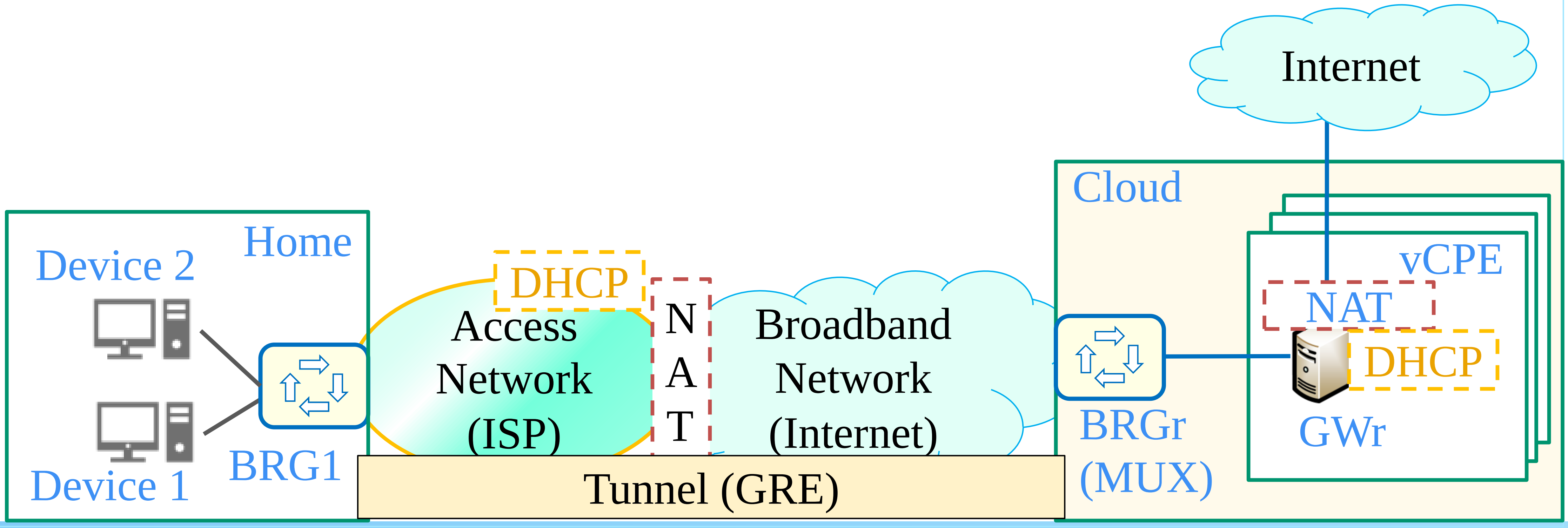
vCPE Architecture (1/2)

- Physical CPE replaced by
  - A home bridge(BRG1) at home
  - A remote virtual (vCPE)
    - Acts as a remote default gateway (GWr) of home devices
    - Runs a DHCP Server and assigns private IPs to home devices



vCPE Architecture (2/2)

- BRG1 WAN port acquire an IP, either private or public, from ISP
- BRGr WAN port has a public IP
- BRG1 establishes a tunnel (logical link) with BRGr

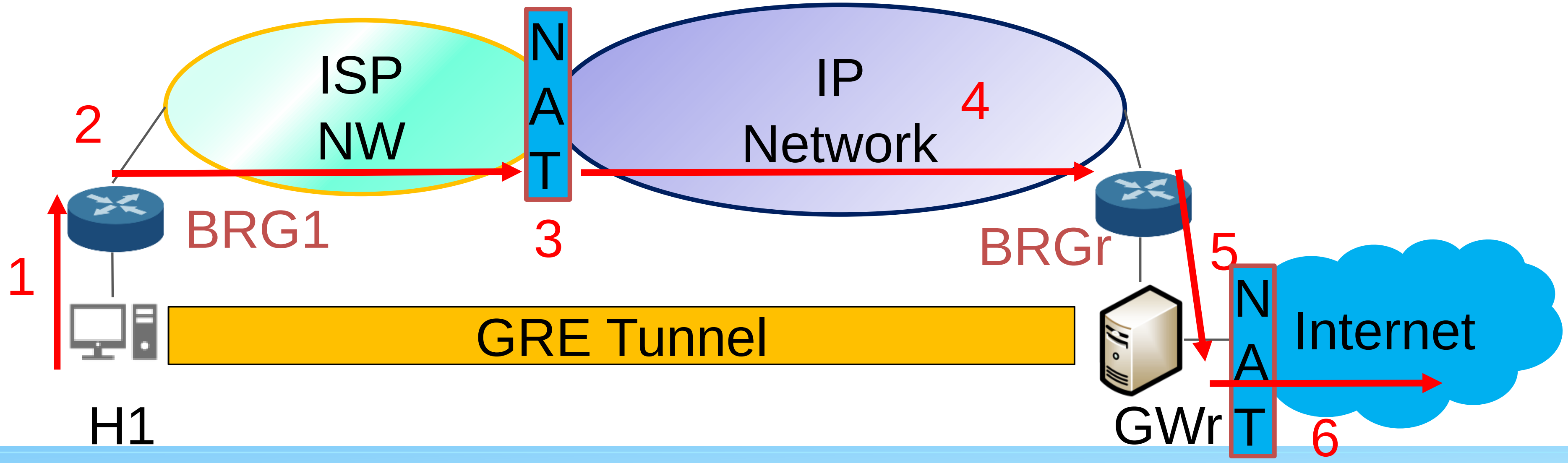


vCPE overview

- vCPE with GRE/VXLAN supporting
  - Multiple LANs can be regarded as a logically large LAN
  - No need to deploy physical CPE to each home subscriber
  - A single virtual CPE platform is able to serve multiple home subscribers

Packet Workflow

1. H1 sends a request to Internet (e.g. request google DNS server)
2. BRG1 encapsulates packets with GRE over UDP (srcIP: private IP)
3. ISP NAT translates Outer src transport address from private to public (srcPort may be changed), then forward packets to BRGr
4. Program on BRGr parses incoming GRE packets and creates corresponding GRE interface
5. BRGr decapulates GRE packets and forwards inner packets to GWr
6. GWr NAT translates src transport address to public transport address then forwards packets to Internet

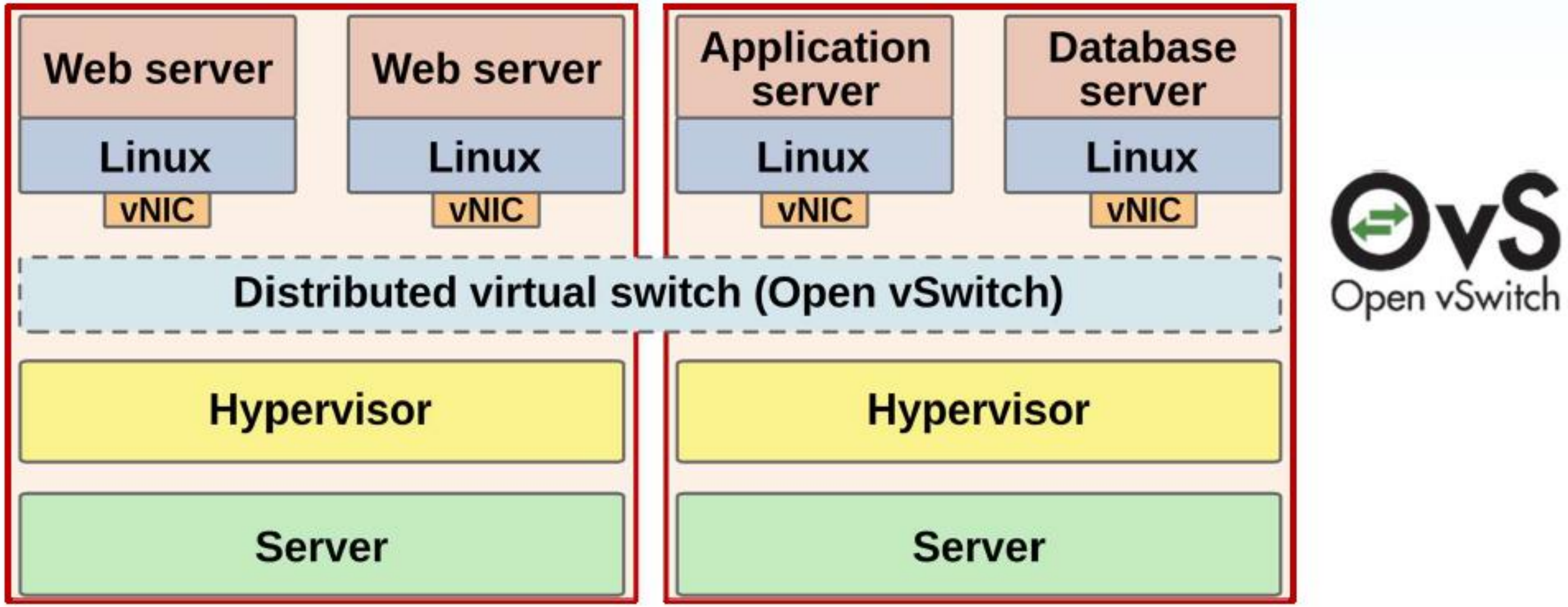


Outline

- Objective
- Environment
- Customer Premise Equipment
- Virtual Customer Premise Equipment
- Open vSwitch
  - Overview
  - Meter
  - Group
- Project
- Appendix

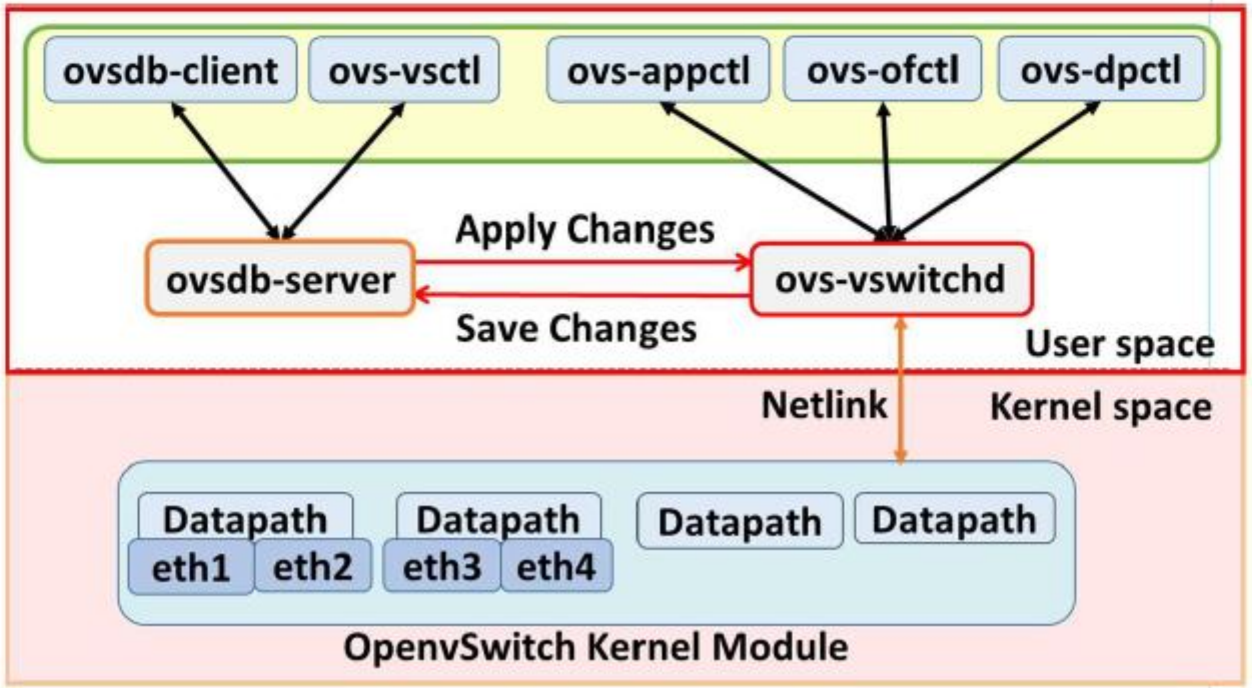
Open vSwitch (1/3)

- An open-source implementation of a distributed virtual multilayer (L2/L3) switch
  - Deployed as a cross-server virtual network switch



Open vSwitch (2/3)

- ovs-appctl: utility for configuring running Open vSwitch daemons
- ovs-ofctl: administer OpenFlow switches
- ovs-vsctl: utility for querying and configuring ovs-vswitchd (via ovsdb database)



Open vSwitch (3/3)

- OVS is a widely-used software virtual switch
  - Supports the latest OpenFlow protocol (ver 1.5)
- Runs OVS on a Linux server with multiple network interfaces
  - Can turn the server into an OpenFlow switch

Open vSwitch Installation

- Add OVS installation in your Dockerfile

```
RUN apt-get install -y --no-install-recommends gcc make && \
        rm –rf /avr/lib/apt/lists/*
RUN pip install six   # install python package
RUN wget https://www.openvswitch.org/releases/openvswitch-
2.11.4.tar.gz
RUN tar zxf openvswitch-2.11.4.tar.gz
RUN cd openvswitch-2.11.4 && \
    ./configure --prefix=/usr --localstatedir=/var --sysconfdir=/etc && \
    make && \
    make install
```

Start Open vSwitch

- Command

```
> /usr/share/openvswitch/scripts/ovs-ctl start
```

- Command Result

```
* Creating empty database /etc/openvswitch/conf.db
* Starting ovsdb-server
* system ID not configured, please use --system-id
* Configuring Open vSwitch system IDs
* Starting ovs-vswitchd
* Enabling remote OVSDB managers
* Starting ovsdb-server
```

Meter table

- Meter is a useful tool for flow control
- A meter table consists of meter entries
- Meter entries defining per-flow meters
  - Enable Rate-limiting, QOS based on the rate
- Meter types
  - DROP: Drop the packet
  - DSCP Remark: Set “change-the-DSCP-value” of traffic entering/leaving the device

Meter Table

Meter ID	Band Type	Rate	Counter
…	…	…	…
100	Drop (remark DSCP)	1000 kbps	…
…	…	…	…

Group table

- Group types
  - All: Execute all buckets in the group
  - INDIRECT: Execute the one defined bucket in the group
  - SELECT: Execute one bucket in the group
  - FAILOVER: Exectue first live bucket
- Make sure to add group entry before adding a flow entry that redirects a flow to the group entry

Group Table:

Group ID	Group type	Counter	Action Buckets
100	All		Port1: output Port3: output Port5: output
...	...		...

Flow Table:

Match Field	Counter	~~Actions~~
...	...	...
Dst IP= 224.2.3.9	...	Group 100

FAILOVER

- Each action bucket associated with a port and/or a group (for group chaining)
- First bucket associated with a live port/group is selected
  - Enables switch to change forwarding without requiring a round trip to controller
- If no buckets are live, packets are dropped

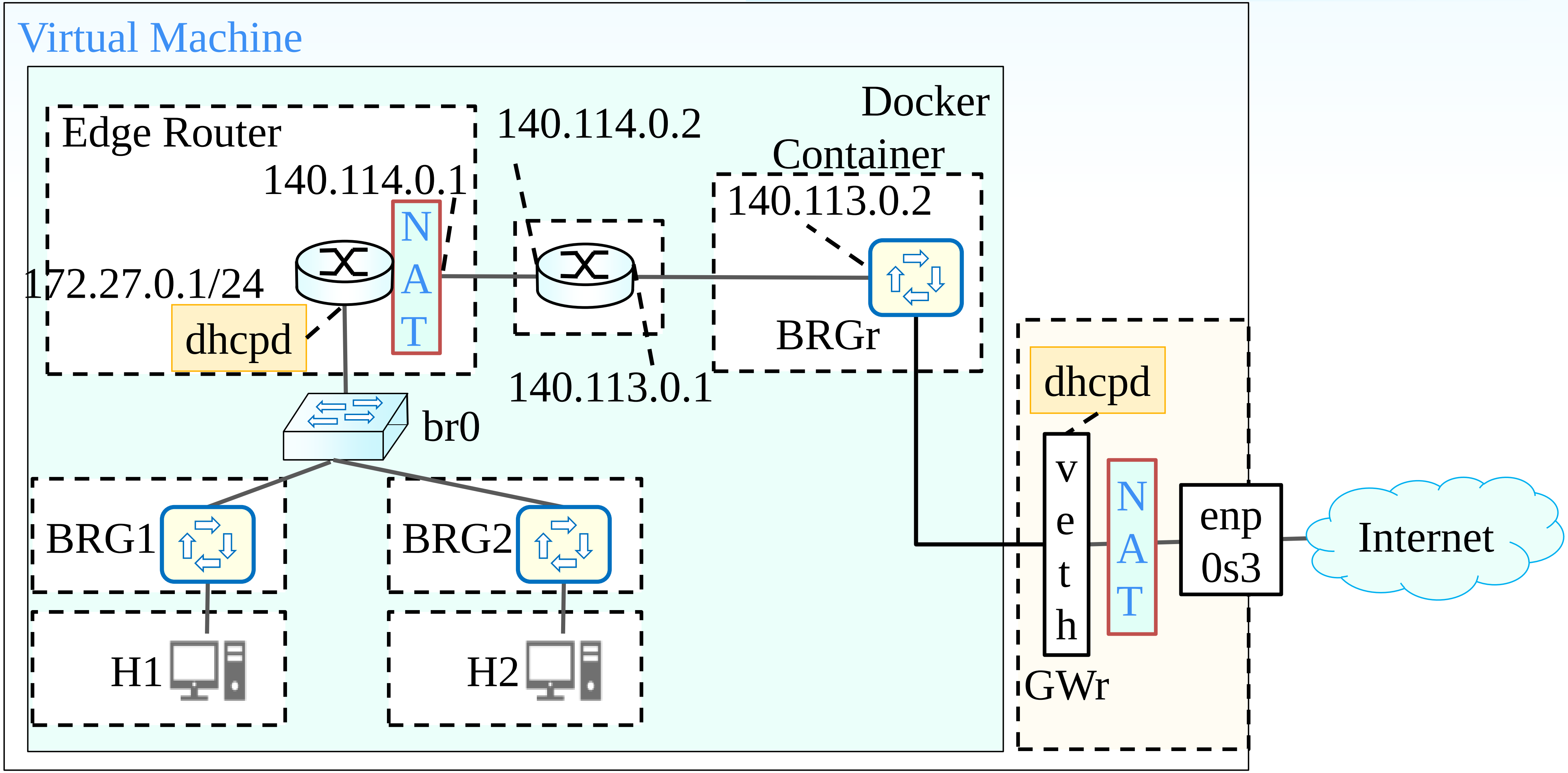
Group Table

Group ID	Group Type	Counter	Action Buckets
100	Fast-failover	777	Port4
			Port5
			Port6

Outline

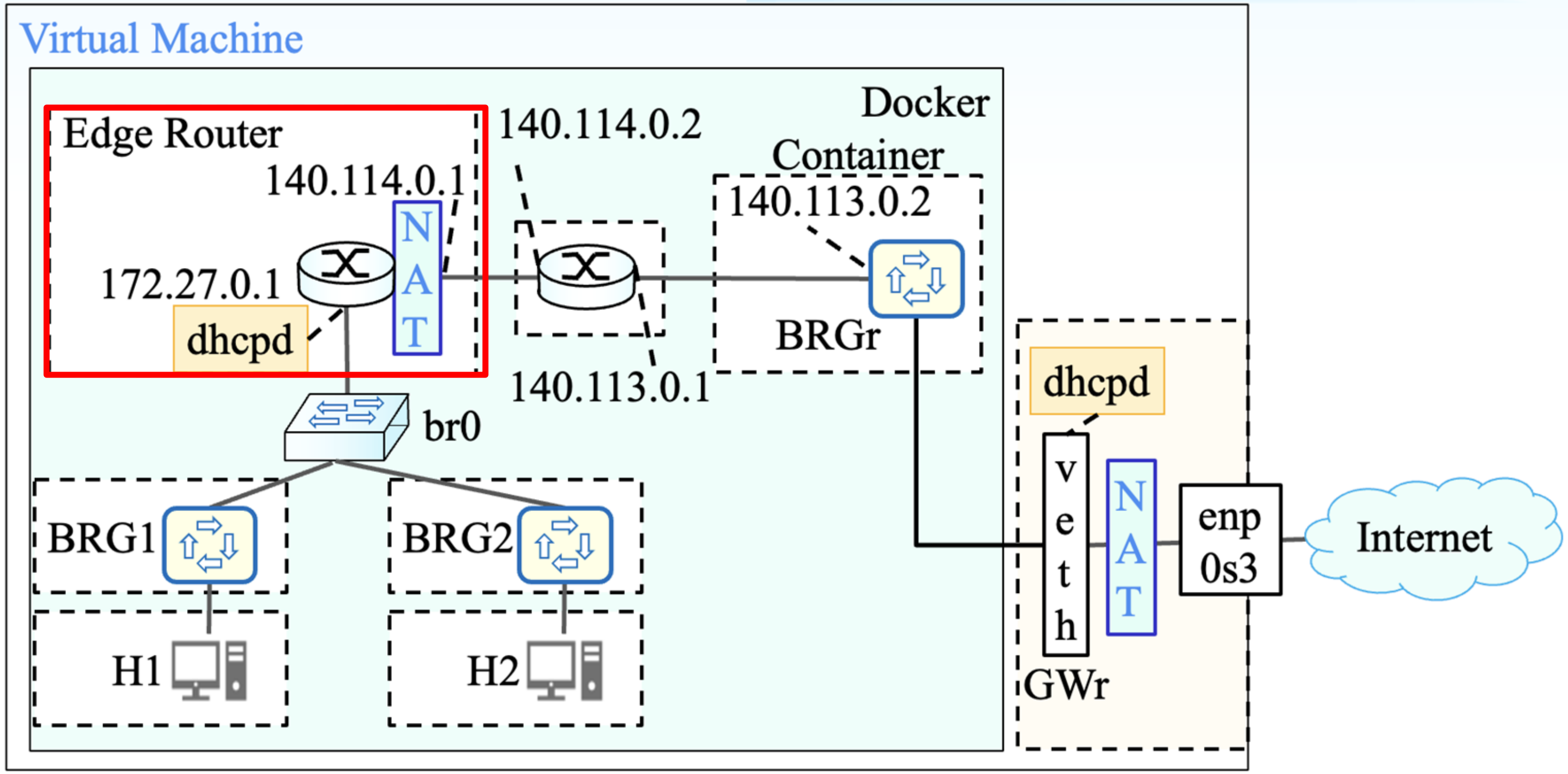
- Objective
- Environment
- Customer Premise Equipment
- Virtual Customer Premise Equipment
- Project
  - Part1: Topology
  - Part1: Node Functions
  - Part1: Requirement
  - Part2: Requirement
  - Part3: Requirement
- Appendix

Lab Topology



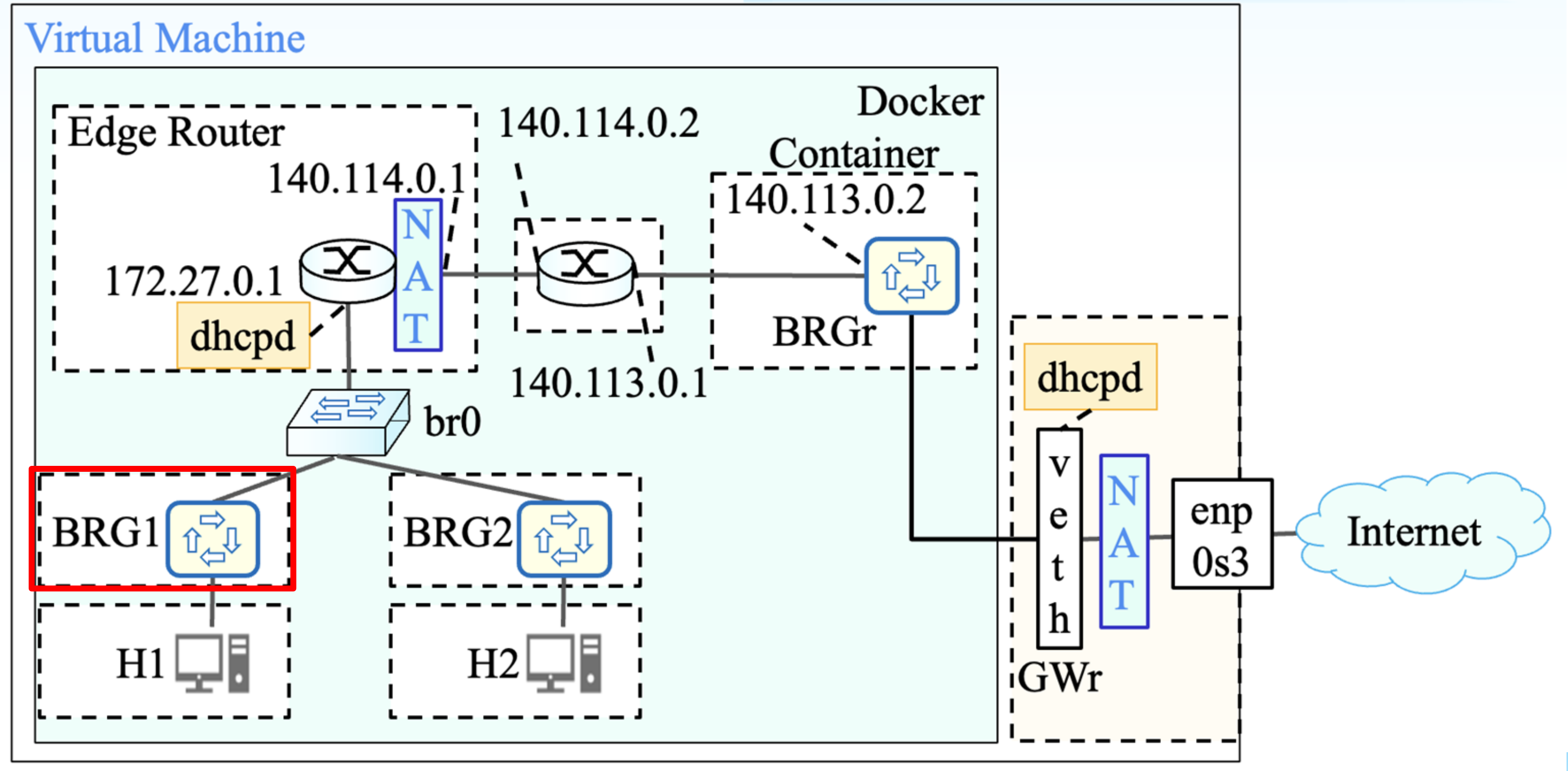
Functionality of Edge Router Node

- A static Public IP for WAN port
- Provides basic routing function
- Runs a DHCP Server on internal interface (LAN port)
  - Assigns IP addresses to WAN ports of BRG1, BGR2
- Perform NAT for BRG1, BRG2
  - Implemented with iptables rules



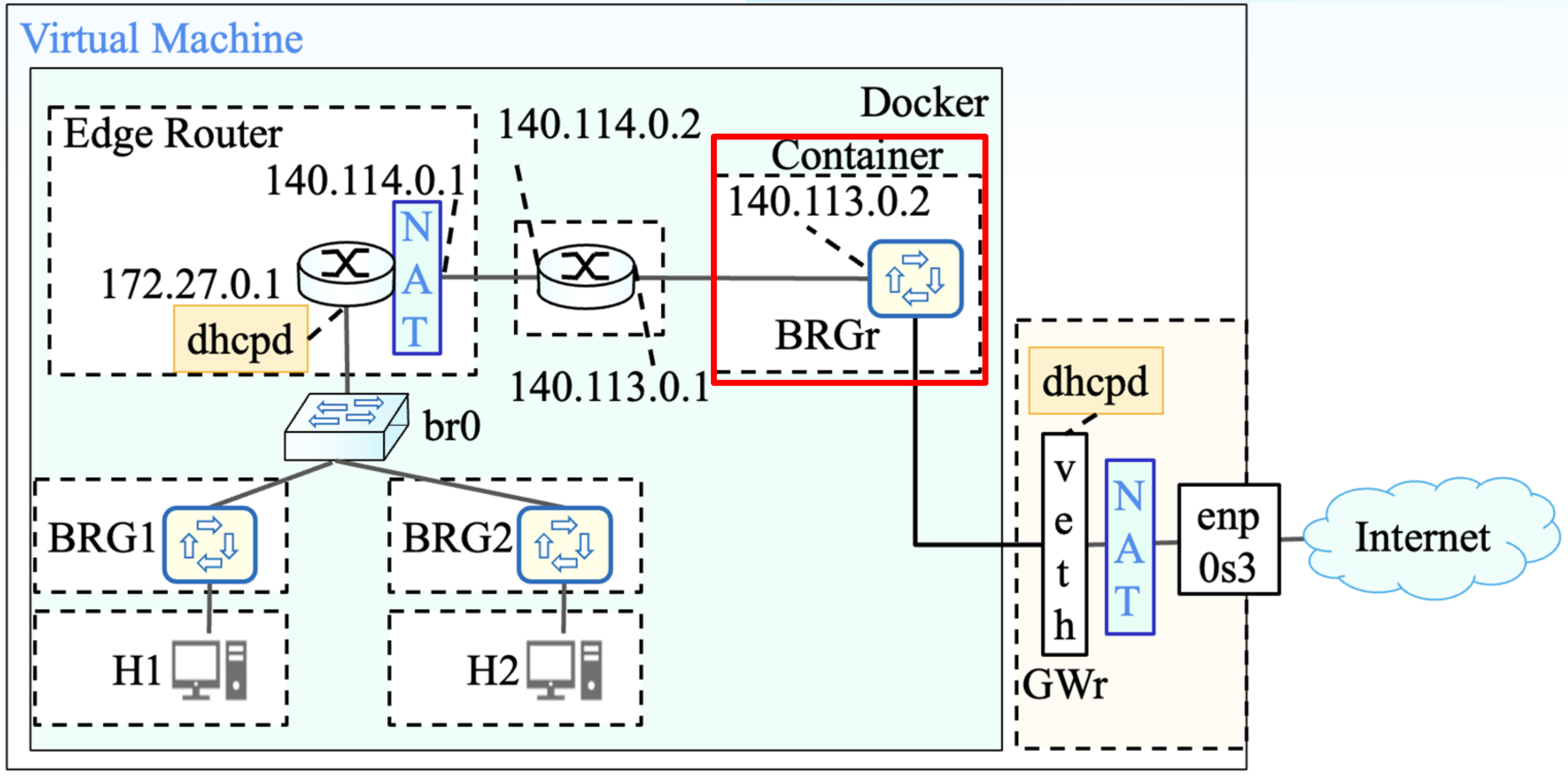
Functionality of BRG1

- BRG1
  - Acquires a dynamic IP address from Edge Router for WAN port
  - Creates a GRE over UDP tunnel on WAN port
  - Sets routing rules for GRE over UDP tunnel to BRGr



Functionality of BRGr

- BRGr
  - Run Auto Tunnel Creation to create GRE over UDP tunnel
    - Corresponding to IP/Port after NAT translation
  - Set routing rules for tunneling to BRG1, BRG2

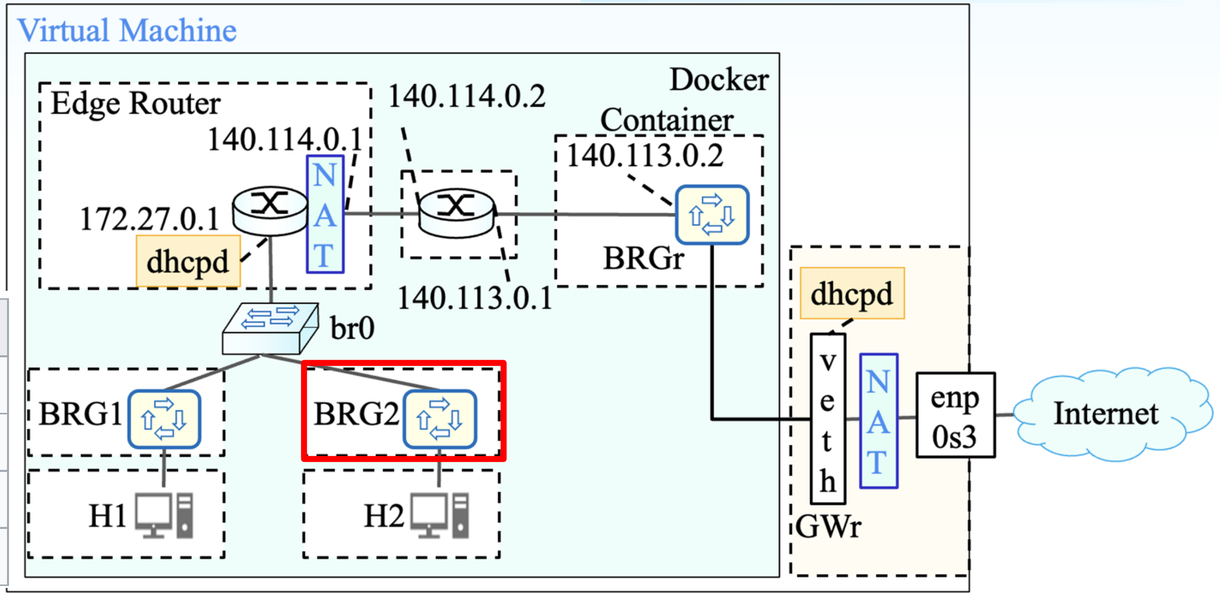


Functionality of BRG2

- Same as BRG1
- In BRGr, need to identify two data stream between BRG1 and BRG2
  - Using Key in GRE header
    - Key value can identify a particular GRE data stream

GRE header

Bits 0–3				4–12	13–15	16–31
C		K	S	Reserved1	Version	Protocol Type
Checksum *(optional)*						Reserved2
Key *(optional)*						
Sequence Number *(optional)*						



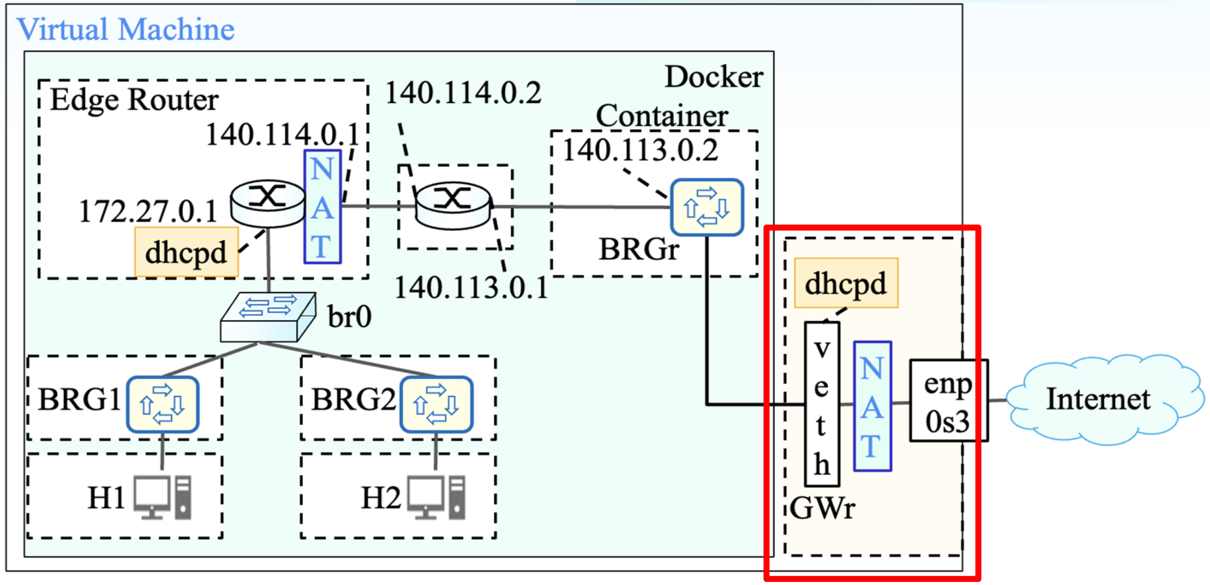
GRETAP Command

- GRETAP comman syntax

```
ip link add DEVICE type { gre | gretap } remote ADDR
local ADDR [ [no][i|o]seq ] [ [i|o]key KEY | no[i|o]key ]
[ [no][i|o]csum ] [ ttl TTL ] [ tos TOS ] [ [no]pmtudisc ]
[ [no]ignore-df ] [ dev PHYS_DEV ] [ encap { fou | gue |
none } ] [ encap-sport { PORT | auto } ] [ encap-dport
PORT ] [ [no]encap-csum ] [ [no]encap-remcsum ] [ external
]
```

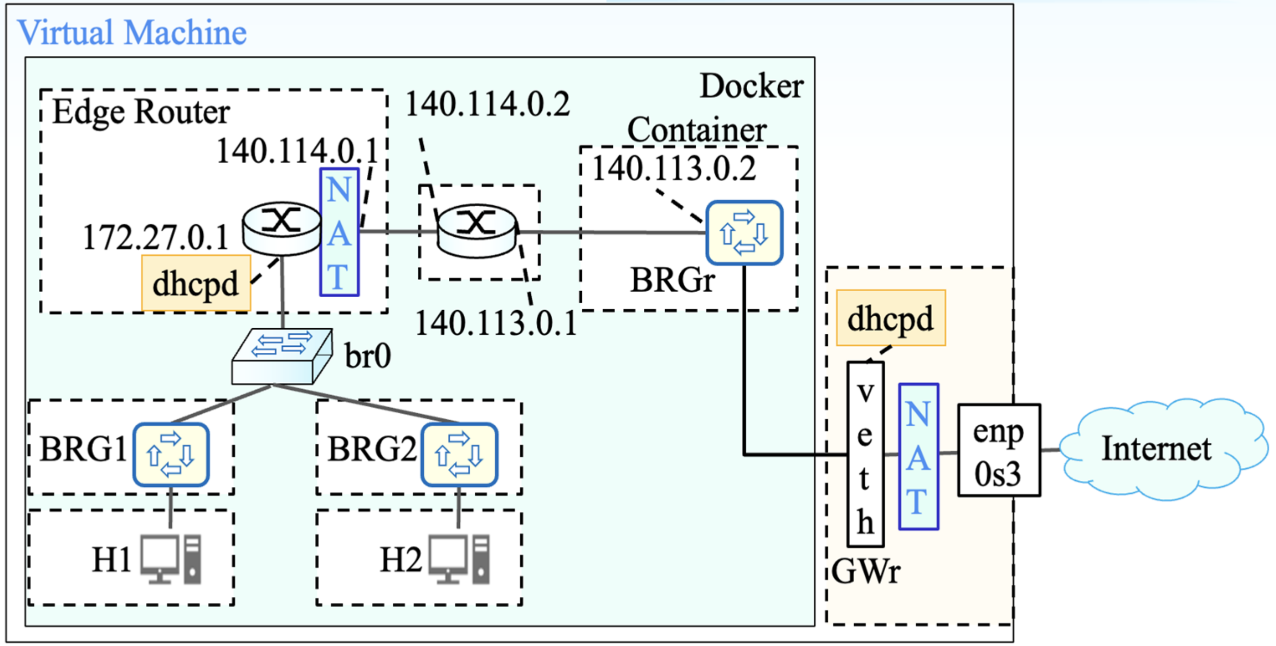
Functionality of GWr node

- Runs DHCPd Server on veth interface
  - Assigning IP addresses and DNS Server to hosts (in BRG1 and BRG2)
    - Subnet 20.0.1.0/24
    - DNS server 8.8.8.8
    - Default gateway (veth)
- Acts as default gateway for H1 and H2
- Set routing and NAT rules for packets
  - From hosts to Internet
  - From Internet to hosts
- Net configuration is default



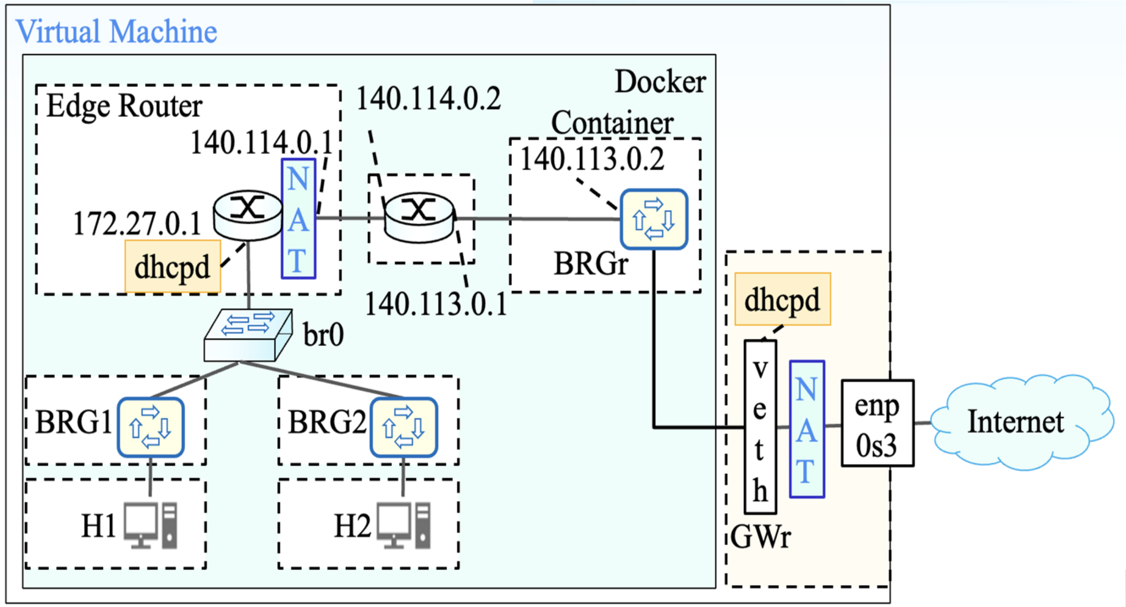
Part1: Requirement (1/2)

- Edge Router functionality
  - DHCP server
    - BRG1 and BRG2 can acquire an IP address from Edge Router
  - NAT rules
    - Perform NAT translation for BRG1 and BRG2
  - Route packets
- BRG1, BRG2
  - Can acquire an IP address from Edge Router
  - Build a GRE tunnel to BRGr
  - Route packets



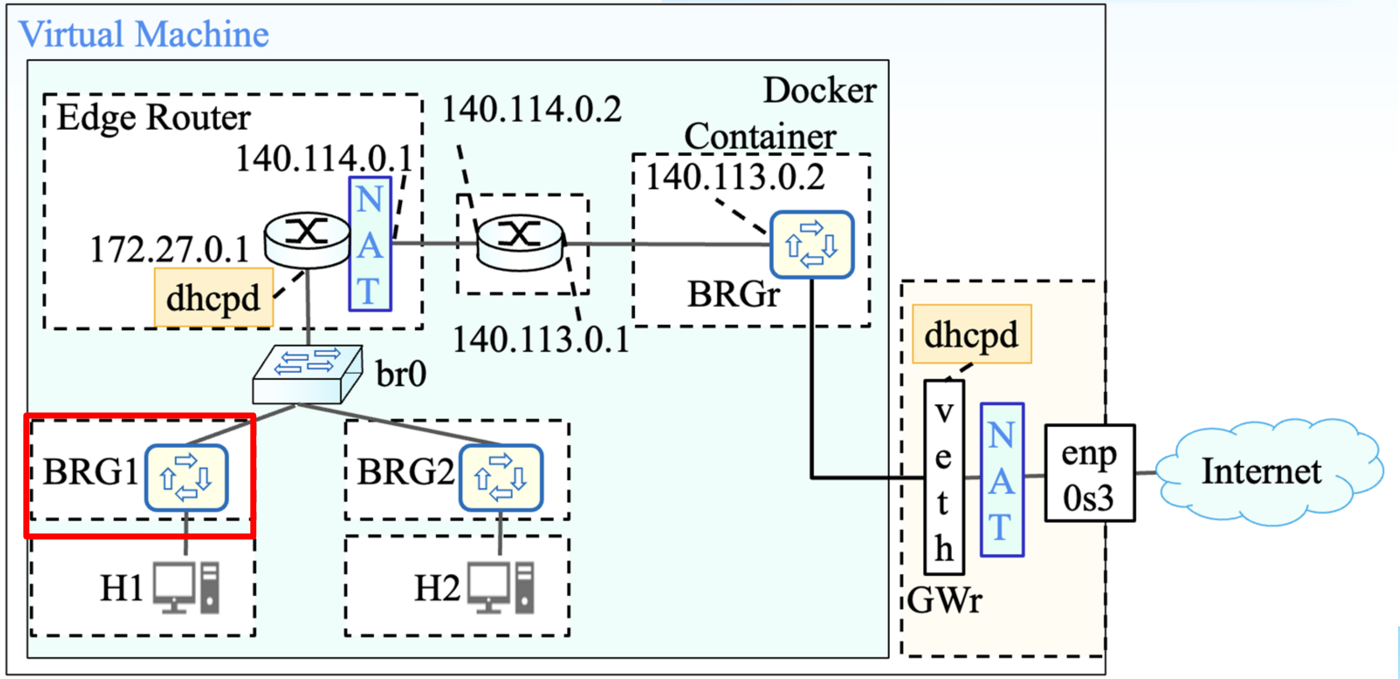
Part1: Requirement (2/2)

- BRGr
  - Run Auto Tunnel Creation to create GRE over UDP tunnels, respectively, to BRG1 and BRG2
  - Route packets
- GWr functionality
  - Dynamic IP assignment with default gateway and DNS configuration
  - NAT translation
  - Route packets
- Hosts H1 and H2
  - Can acquire an IP address from GWr
  - Can ping hosts on Internet



Part2: BRG1 Node with OVS Rate Limiting

- Replace Linux bridge with OVS on BRG1
- Create a GRETAP tunnel between BRG1 and BRGr
- Attach GRETAP interface to OVS on BRG1
- Add a meter entry on OVS for rate limiting of 1Mbps on GRETAP
- Add a flow entry on OVS to redirect flow to a meter entry



iperf3

- A real-time network throughput measurement tool
- Installation in Dockerfile
- Iperf3 client

```
root@99c4a5960337:/# iperf3 -u -b 100M -c 20.0.1.254 --length 1200
Connecting to host 20.0.1.254, port 5201
[  4] local 20.0.1.2 port 49077 connected to 20.0.1.254 port 5201
[ ID] Interval           Transfer     Bandwidth       Total Datagrams
[  4]   0.00-1.00   sec  10.8 MBytes  90.9 Mbits/sec  9471
[  4]   1.00-2.00   sec  12.0 MBytes   100 Mbits/sec  10454
```

- Iperf3 server

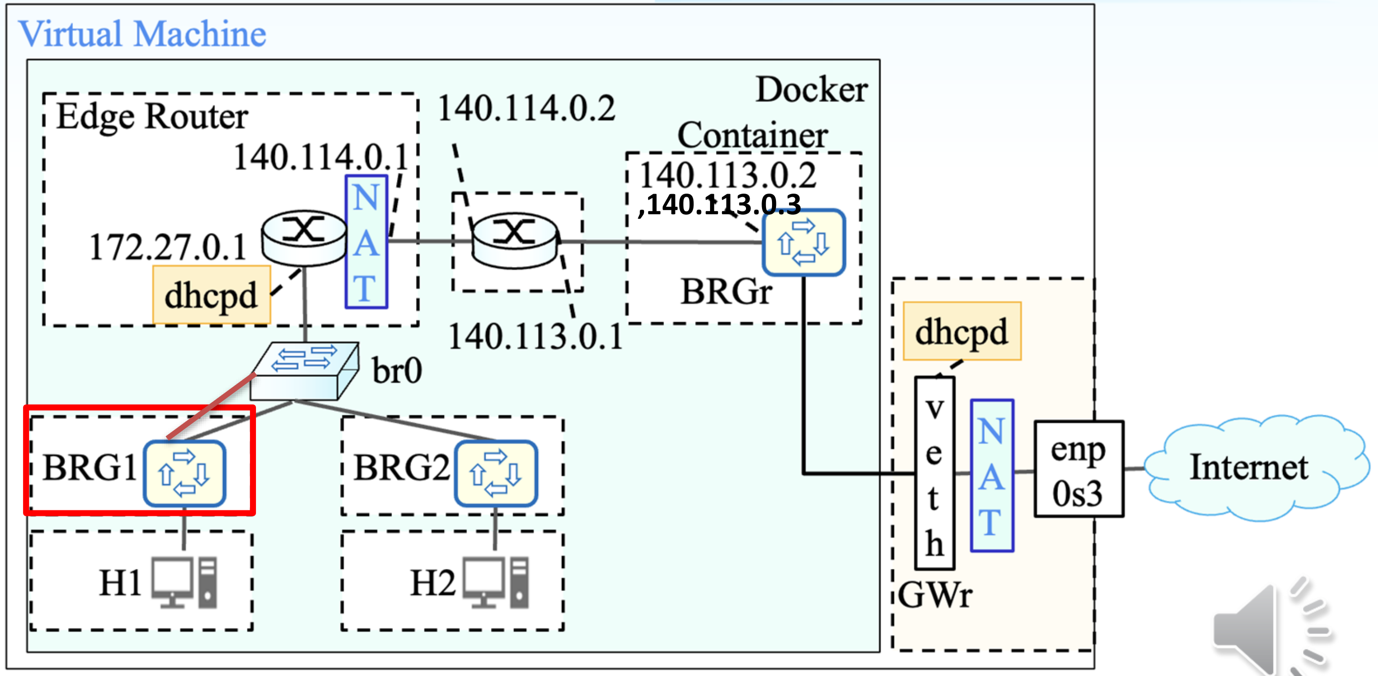
```
andy78644@andy78644:~/ta/project$ iperf3 -s -B 20.0.1.254
-----------------------------------------------------------
Server listening on 5201
-----------------------------------------------------------
Accepted connection from 20.0.1.2, port 46162
[  5] local 20.0.1.254 port 5201 connected to 20.0.1.2 port 54411
[ ID] Interval           Transfer     Bandwidth       Jitter    Lost/Total Datagrams
[  5]   0.00-1.00   sec  10.4 MBytes  87.3 Mbits/sec  0.009 ms  403/9503 (4.2%)
[  5]   1.00-2.00   sec  11.9 MBytes  99.6 Mbits/sec  0.010 ms  11/10409 (0.11%)
```

Part2 : Requirement

- Meter table
  - Use H1 as iperf3 client and GWr as iperf3 server
    - Client sends a UDP traffic larger than 1Mbps to server
      - Because TCP has a congestion and flow control mechanism
    - Highlight the result of rate limiting
  - Show that the traffic speed is limited down to 1Mbps

Part3: BRG1 Node with OVS Fast Failover

- Add secondary IP on the wan port of BRGr
- Add another veth link to connect BRG1 and br0 as the secondary path
- Run DHCP on BRG1 to acquire IP for the new veth link
- Use the new veth link and secondary IP of BRGr to create another GRETAP tunnel for the secondary path
- Set a group entry with two buckets
  - Type: fast failover (ff)
  - Buckets:
    - Output to primary path
    - Output to secondary path



Part3 : Requirement

- Group table
  - Bring up both paths of BRG1 to br0
  - Let H1 ping 8.8.8.8 and observe traffic
  - Shut down primary path and corresponding GRETAP tunnel
  - Observe traffic before and after primary path failure
    - Take screenshot and explain the fast failover result

Part1 Report (1/3)

1. Node configurations (20%)

- Show the configuration commands you made on each node to provide Internet connectivity for hosts and briefly explain the purpose of the commands (take screenshots to justify your answers)(10%)
  - a) DHCP on edge router (2%)
    - BRG1 can acquire an IP address from Edge Router, respectively
  - b) NAT on Edge Router (5%)
    - Use iptables command to show NAT rules
  - c) GRE over UDP (5%)
    - Setup BRG1, but not BRG2. Show that BRG1 can ping BRGr.
    - Show interfaces list on node BRG1 and BRGr

Part1 Report (2/3)

d) DHCP on GWr (2%)
  - H1 acquires IP and gateway dynamically from DHCP on GWr

e) NAT on GWr

Internet connectivity (6%)

  - Use iptables command to show NAT rules

f) Ping to internet
  - Show that H1 can ping google DNS Server 8.8.8.8

```
H1 container> ping 8.8.8.8 –c 5
```

2. Route Trace (5%)

- Let H1 ping 8.8.8.8. Capture packets and take screenshots on nodes and briefly describe the change in packet headers
  - BRG1, BRGr, GWr, Edge Router input/output
  - Briefly explain the purposes of the header changes

Part1 Report (3/3)

3. Setup both BRG1 and BRG2. (5%)
   a) Show that both BRG1 and BRG2 can create GRETAP tunnels with BRGr. Explain how BRGr distinguishes the two GRETAP tunnels.
   b) Briefly explain how BRGr forwards packets back to the correct BRG via the corresponding GRETAP.
   c) Show that H2 can ping google 8.8.8.8

Part2 Report (1/2)

1. GRETAP on OVS bridge (5%)
   a) Show the configuration commands
      - Create OVS bridge
      - Create GRETAP tunnel
      - Attach the GRETAP interface to OVS bridge
   b) Show the interfaces on OVS bridge

Part2 Report (2/2)

2. Meter table (5%)
   a) Show the configuration commands
      - Add meter entry
      - Add flow rule to redirect flow to meter entry
   b) Show meter entries and flow entries
   c) Observe and record the traffic speeds
      - Use iperf3 to send UDP packets with the bandwidth larger than 1Mbps
      - Justify the effect of meter entries, by showing
        - The packets rate sent by iperf3 client
        - The packet rate received iperf3 server

Part3 Report (BONUS)

3. Group table (5%)
   - Show the configuration commands you made
     - Add secondary IP on the wan port of BRGr
     - Add another veth link to connect BRG1 and br0 as the secondary path
     - Run DHCP on BRG1 to acquire the IP for the new veth link
     - Use the new veth link and secondary IP of BRGr to create another GRETAP tunnel for the secondary path
     - Add group table
     - Add a flow entry to redirect flow to group table
   - Show the interfaces on OVS bridge, flow entries and group entries
   - Justify effect of fast failover by following steps
     - Let H1 ping 8.8.8.8
     - Use tcpdump to show the traffic and interfaces in use before and after you shut down the primary link (path) on BRG1

Submission

- Files
  - Code (60%, with DEMO)
    - <studentID>.c/cpp/go
  - Report (40%)
    - project_<studentID>.pdf
- Submission
  - Zip all file into a zip file
    - Name: project_<studentID>.zip
- Wrong filename or format subjects to 10 points deduction

Outline

- Objective
- Environment
- Customer Premise Equipment
- Virtual Customer Premise Equipment
- Project
- Appendix

Appendix

- ovs-vsctl man page
  - https://man7.org/linux/man-pages/man8/ovs-vsctl.8.html
- ovs-ofctl man page
  - https://man7.org/linux/man-pages/man8/ovs-ofctl.8.html
- OVS command
  - https://docs.pica8.com/display/PicOS211sp/ovs-ofctl+Common+Commands
- Gre key extension
  - https://datatracker.ietf.org/doc/html/rfc6245
- Iperf3 document
  - https://iperf.fr/iperf-doc.php

Q & A