

Azblink NFV Platform

Abstract: This document is to introduce Azblink Network Functions Virtualization (NFV) Platform. It provides virtual machines to be installed with operation systems like Windows or Linux with given firewall and routing functions on the network interfaces. It can be used for firewall virtualization, and SBC virtualization; and virtual machines can be deployed into different zones within clicks.

Table of Contents

Chapter 1 Overview.....	8
Bridging and Routing.....	8
Virtual Bridge and Virtual Host.....	10
Other Possible Network Operations(But we do not provide).....	14
Firewall Operations on Bridges.....	18
Firewall Virtualization or SBC Virtualization.....	24
Chapter 2 Virtual Host.....	27
Upload CD Image.....	30
Create an instance of virtual host.....	31
Bridge Assignment.....	33
Change the Setting of Memory and Tablet.....	34
CPU and Chipset.....	35
Storage Device Setting.....	36
Host Management.....	39
Chapter 3 Border Control.....	42
Port Forwarding.....	50
Connection Tracking.....	57
Actions after Receiving Network Packets.....	59
Add Rule.....	61
Allowing Exceptions for TCP Connections from dmz to loc.....	63
Reject or Drop Connections.....	66
Redirect Traffic to Another Port of the Base Platform.....	70
List or Delete Rule.....	72
Using DNAT for Port Forwarding.....	73
IP Load Balance.....	76
Use Web Proxy.....	84
Web Caching.....	85

URL Screening.....	86
Access Block Time.....	87
Traffic Bandwidth Control.....	88
Setting Network Interface Bandwidth.....	89
Define Priority Classes.....	93
Packet Marking for Traffic Control.....	95
The Components of a Bridge.....	98
Zone Definition.....	102
Port Association for NAT Setting.....	103
IP Policy Routing.....	104
Http Reverse Proxy for Request Filtering.....	126
Chapter 4 VPN.....	129
Client-to-Site VPN Connection.....	136
Site-to-site VPN Connection(Routing Mode).....	142
Site-to-site VPN in Bridging Mode.....	153
Chapter 5 Dynamic Routing.....	161
RIPv2 (Router Information Protocol, version 2).....	165
OSPF(Open Shortest Path First).....	171
PIM(Protocol Independent Multicast).....	180
Routing Across VPN.....	184
Chapter 6 Deployment Scenarios.....	187
Example 1: Isolate the Machines with/without Internet Access	187
Example 2: Use VPN to Access Virtual Desktop.....	191
Example 3: SBC or Firewall Virtualization.....	192
Example 4: Place Storage System in Another Subnet.....	194
Example 5: Multicast Router with Multicast Sender.....	195

Illustration Index

Illustration 1: Bridging Operation.....	9
Illustration 2: Routing Operation.....	10
Illustration 3: Traffic Inside the Bridge.....	12
Illustration 4: Traffic Going Across the Bridge to another Subnet.....	13
Illustration 5: NAT on per physical port basis.....	15
Illustration 6: Allowing Sending Traffic between Virtual Hosts on the same bridge.....	18
Illustration 7: Traffic not Allowed between Virtual Hosts on the same bridge....	20
Illustration 8: Traffic Locked inside a Bridge.....	21
Illustration 9: Virtual Bridges inside Base Platform.....	22
Illustration 10: NAT while going across the boundary of br0.....	23
Illustration 11: No NAT Operation inside Bridge br1.....	24
Illustration 12: No NAT Operation while going across the bridge boundaries br1 and br3.....	25
Illustration 13: Virtual Host connecting to the Bridge br0.....	26
Illustration 14: Virtual Host connecting to multiple bridges.....	27
Illustration 15: Virtual Host connecting to Bridges br0 and br1.....	28
Illustration 16: Virtual Host Connection for Firewall Virtualization.....	29
Illustration 17: Base Platform and Other Routers.....	30
Illustration 18: Uploading CD/DVD Image for Guest OS.....	34
Illustration 19: screen snapshot for creating an instance of virtual host.....	35
Illustration 20: Bridge Assignment and Ethernet Type of a Virtual Host.....	37
Illustration 21: Changing Memory Size and Enabling USB Tablet of VM.....	38
Illustration 22: Changing Number of CPUs and Chipset.....	39
Illustration 23: Extra Storage Device Setting.....	40
Illustration 24: Host Management of Virtual Machines.....	43
Illustration 25: Screen Snapshot for Using VNC.....	44
Illustration 26: Base Platform with Virtual Bridges.....	46
Illustration 27: The Equivalent Model by using physical Ethernet Switches.....	47
Illustration 28: Zone partitioning on the base platform.....	48
Illustration 29: Traffic with destination in “net”.....	49
Illustration 30: Traffic originated from “net” is forbidden to access “loc” or “dmz”	50
Illustration 31: Traffic originated from the zone “loc”.....	51
Illustration 32: Traffic originated from zone “dmz” is forbidden to access “loc”.52	
Illustration 33: Screen Snapshot for Port Forwarding.....	54
Illustration 34: Port Forwarding Example.....	55
Illustration 35: Example for HTTP Port Forwarding.....	56
Illustration 36: Screen Snapshot for Setting HTTP Port Forwarding.....	57
Illustration 37: Example of SMTP for Port Forwarding.....	58
Illustration 38: Screen Snapshot after Adding SMTP Port Forwarding.....	59

Illustration 39: Screen Snapshot for Connection Tracking.....	61
Illustration 40: Display for Connection Status.....	62
Illustration 41: Screen Snapshot for Adding Rule.....	65
Illustration 42: Source and Destination Associated with Rule.....	66
Illustration 43: Adding Exception Rule from Zone "dmz" to "loc".....	68
Illustration 44: Screen Snapshot for "dmz" to a Host in "loc".....	68
Illustration 45: Display the Rule from "dmz" to a Host in "loc".....	69
Illustration 46: Screen Snapshot for Dropping Http Traffic from "loc" to "net"...	71
Illustration 47: Screen Snapshot form Dropping HTTPS traffic from "loc" to "net"	72
Illustration 48: Example for Dropping HTTP and HTTPS (Ports 80 and 443)....	73
Illustration 49: Redirect Traffic to a Different Port.....	74
Illustration 50: REDIRECT rule in display list.....	75
Illustration 51: Delete the rule from List.....	76
Illustration 52: Use DNAT for Port Forwarding.....	77
Illustration 53: Port Forwarding to a Host with different port.....	78
Illustration 54: List Rules for Port Forwarding.....	79
Illustration 55: Screen Snapshot for IP Load Balance.....	80
Illustration 56: Distribute HTTP Traffic to 2 Hosts.....	82
Illustration 57: Create a Service Item for Load Balance.....	83
Illustration 58: Service Item for Load Balance in List.....	84
Illustration 59: Add one Host into Service Item for Load Balance.....	85
Illustration 60: Add another Host into Service Item for Load Balance.....	86
Illustration 61: the Hosts associated with Service Item for Load Balance in List	87
Illustration 62: Screen Snapshot for Web Proxy Caching and Access.....	89
Illustration 63: URL Screening in Web Proxy.....	90
Illustration 64: Time Slot Setting to Block HTTP Access on Web Proxy.....	91
Illustration 65: Setting Network Interface Bandwidth.....	93
Illustration 66: Setting Inbound and Outbound Bandwidth.....	94
Illustration 67: Screen Snapshot after Setting Interface Bandwidth.....	95
Illustration 68: Priority Classes after Setting Interface Bandwidth Limit.....	96
Illustration 69: Screen Snapshot for Defining Priority Classes.....	97
Illustration 70: Screen Snapshot for Traffic Prioritizing.....	99
Illustration 71: Setting Mark for Traffic Priority.....	100
Illustration 72: Marking Rule Listing.....	101
Illustration 73: Relationship among Bridge, Base Platform, Physical Ethernet Interface, and Virtual Hosts.....	102
Illustration 74: Place Multiple Ethernet interfaces with Physical Entities into One Bridge.....	105
Illustration 75: Snapshot for Zone Setting.....	106
Illustration 76: NAT Setting.....	107
Illustration 77: Information for IP Policy Routing.....	108
Illustration 78: Example of two WAN ports.....	110
Illustration 79: Change Bridge's IP Address.....	111

Illustration 80: Remove "br4" from zone "loc".....	112
Illustration 81: Listing after Removing "br4" from zone "loc".....	113
Illustration 82: Add "br4" to zone "net".....	114
Illustration 83: List for "br4" in zone "net".....	115
Illustration 84: Remove Original Subnet of "br4" for Using NAT under "br0"...	116
Illustration 85: Remove Subnet of "br4" Using NAT under "br0".....	116
Illustration 86: Add Subnet of "br5" Using NAT under "br4".....	117
Illustration 87: List for Subnet of "br5" Using NAT under "br5".....	118
Illustration 88: Catch Traffic Destined to the Subnet "10.0.0.0/8".....	119
Illustration 89: Catch the Traffic from the Subnet "10.0.0.0/8".....	120
Illustration 90: Catch the Traffic from "172.16.14.0/24".....	121
Illustration 91: Catch the Traffic Destined to "172.16.14.0/24".....	122
Illustration 92: List Rules for Lookup Routing Tables.....	123
Illustration 93: Add Default Gateway in the "moon" Routing Table.....	124
Illustration 94: Default Gateway in the "moon" Routing Table.....	126
Illustration 95: Add routing entry for the subnet connecting to "br4" in "moon"	127
Illustration 96: Add routing entry for the subnet connecting to "br5" in "moon"	128
Illustration 97: Content of "moon" Routing Table.....	129
Illustration 98: Screen Snapshot for HTTP Reverse Proxy.....	130
Illustration 99: Http Reverse Proxy for Two Different Host Names.....	131
Illustration 100: Http Reverse Proxy for Two Hosts.....	132
Illustration 101: VPN Operation Principle.....	133
Illustration 102: Client-to-Site VPN.....	135
Illustration 103: Site-to-Site VPN.....	136
Illustration 104: Bridging.....	137
Illustration 105: Routing.....	137
Illustration 106: VPN Key Generation Process.....	138
Illustration 107: Client-to-Site VPN Address Pool.....	140
Illustration 108: Selection of Data Cipher.....	141
Illustration 109: VPN Client to Access a Subnet.....	142
Illustration 110: Pushed Setting in Client-to-Site VPN.....	143
Illustration 111: Client-to-Site VPN Certificate and Key Generation.....	144
Illustration 112: Client Certificate Download for Client-to-Site VPN.....	145
Illustration 113: Site-to-Site VPN Key Generation.....	147
Illustration 114: Site-to-Site VPN CA Generation.....	148
Illustration 115: Site-to-Site VPN: Server Key and Certificate Generation.....	149
Illustration 116: Site-to-Site VPN: Client Key and Certificate Generation.....	150
Illustration 117: Site-to-site VPN: Sample for Key Generation.....	151
Illustration 118: VPN Gateway as TLS server.....	152
Illustration 119: VPN Gateway on the Remote Site.....	153
Illustration 120: Site-to-Site VPN Sample Setting.....	154
Illustration 121: Screen Snapshot as Site-to-Site Multiplexer.....	156

Illustration 122: Example for Site-to-Site VPN in Bridge Mode.....	158
Illustration 123: Detailed Operation for Site-to-Site VPN in Bridge Mode.....	159
Illustration 124: CA, Keys and Certificates for Site-to-Site VPN in Bridging Mode	160
Illustration 125: Sample Setting for VPN Bridge Server.....	161
Illustration 126: Example for VPN Network Device Joining a Bridge (Server Part)	161
Illustration 127: Sample Setting on the Client of Site-to-Site VPN in Bridging Mode.....	162
Illustration 128: Certificate Display on Client Side.....	163
Illustration 129: Sample for VPN Network Device to Join a Bridge (Client Part)	163
Illustration 130: Border Control and Dynamic Routing.....	166
Illustration 131: Turn Off Border Control.....	166
Illustration 132: Cascade two Machines together.....	167
Illustration 133: Scenario to use PIM.....	168
Illustration 134: OSPF ABR (Area Border Router).....	168
Illustration 135: Example for RIPv2.....	169
Illustration 136: Subnet to Send Out Multicast Update for RIPv2.....	170
Illustration 137: List of the Subnet(s) to Send Multicast Update.....	171
Illustration 138: Set up Authentication Key for RIPv2.....	172
Illustration 139: List of Authentication Key(s).....	172
Illustration 140: Subnet for Multicast Update on another machine.....	173
Illustration 141: List of the Authentication Key.....	174
Illustration 142: Content of Routing Table after Starting RIPv2.....	174
Illustration 143: OSPF Setup Example.....	175
Illustration 144: Subnet and Area ID Setup (Router A).....	176
Illustration 145: List of Subnets (Router A).....	177
Illustration 146: Authentication Setting of OSPF (Router A).....	178
Illustration 147: Subnet Setting (Router B).....	179
Illustration 148: Authentication Setting (Router B).....	180
Illustration 149: Routing Table on Router A so far (After turning on OSPF)....	181
Illustration 150: Subnet Setting (Router C).....	181
Illustration 151: Authentication Setting (Router C).....	182
Illustration 152: Router Table (Router C).....	183
Illustration 153: Scenario to use IGMP and PIM.....	184
Illustration 154: Setting for Multicast Routing.....	185
Illustration 155: After PIM is turned on.....	186
Illustration 156: Multicast Control.....	187
Illustration 157: Two Routers Across Internet.....	188
Illustration 158: Two Private Routers Try to Connect Across Internet.....	190
Illustration 159: OSPF or RIPv2 across VPN (in Bridging Mode).....	190
Illustration 160: Zone "dmz" Rules.....	192
Illustration 161: Block the Access to Internet from Zone "loc".....	193

Illustration 162: Use Virtual Machines for Internet Access.....	194
Illustration 163: Access Virtual Machine via VPN.....	195
Illustration 164: SBC or Firewall Virtualization.....	197
Illustration 165: Use Dedicated Subnet as Storage Area Network.....	198
Illustration 166: Multicast Router with Multicast Sender.....	200
Illustration 167: Example of TTL setting in VLC as Multicast Sender.....	202

Chapter 1 Overview

Have you ever thought to install several hosts into virtual machines by giving them different privileges of network access? For example, install two hosts into virtual machines – one host is used to access Internet only and forbidden to access internal network, the other is used to access internal network only without the access of Internet. There are many ways to achieve this goal via setting up the associated rules in external firewall or router. Our solution to this is to provide virtual machines in a controlled network environment without too much relying on extra network equipments.

The Azblink Network Functions Virtualization Platform is a platform to create virtual hosts in a predefined network environment. The network zones are governed by the rules that can be added or deleted via web interface. The platform by itself is a firewall and router so that virtual hosts can be created in a well-planned environment from the very beginning. VPN is also provided by the platform for managing each virtual host from its console remotely.

In the following sections, we will have the terminology used in this document defined and explain how those network operations work.

Bridging and Routing

We start with the introduction on bridging and routing. In terms of OSI (Open System Interconnection) model, if the data exchange is performed at **data link layer**, it is called “**bridging**”; if it is performed at **network layer**, it is called “**routing**”. In terms of TCP/IP/Ethernet, if where the data should go is judged by MAC addresses of the Ethernet frames, it is called “bridging”; if it is judged by the IP header of IP packets, it is called routing. While discussing data transmitted and received in data link layer, people usually use the term “frame” whereas they use the term “packet” in network layer. However, we are not following that convention so strictly in this document. We just refer them as “network traffic” either in data link layer or network layer.

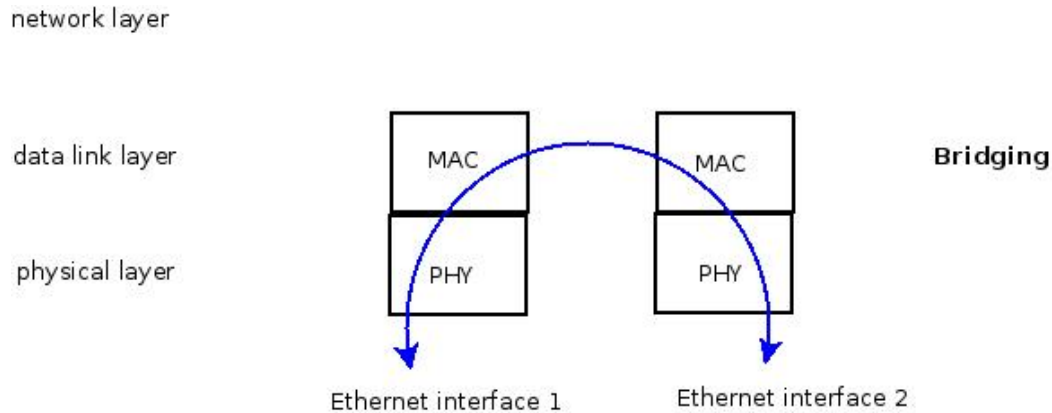


Illustration 1: Bridging Operation

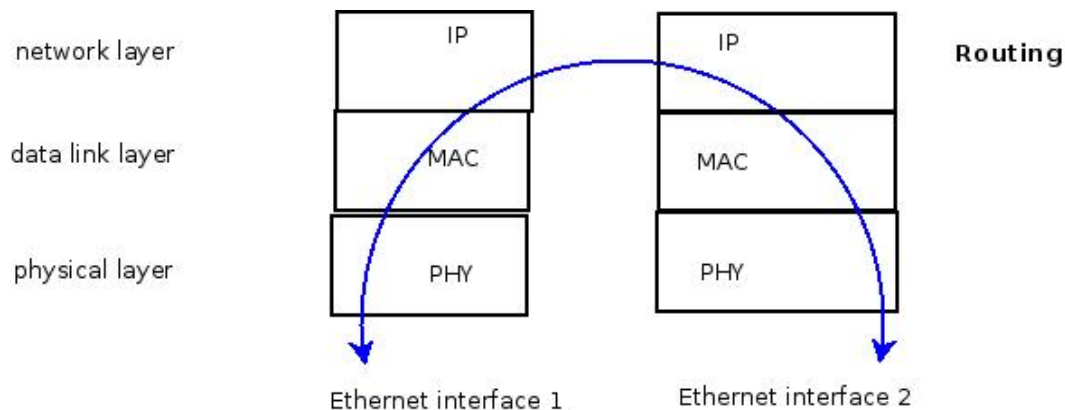


Illustration 2: Routing Operation

The “Ethernet” we are referring to is consisted of physical layer and data link layer (usually known as PHY and MAC). And IP (Internet Protocol) belongs to network layer. Thus, if only Ethernet MAC address is examined and determined where to go, we say that it is bridging. Similarly, if IP address is examined, it belongs to the category of routing. For a network application to send out traffic to the other end, it is a complicated process with many details.

When a host A is sending IP packets to another host B with known IP address, it still has to find out the MAC address of the host B. ARP (Address Resolution Protocol) will be used to find out the MAC address corresponding to that IP address of host B. The host A will first find out if the MAC address with respect to that IP address in the lookup table. If there exists MAC address matched, that MAC address is used. Otherwise, ARP query is broadcast over the network. The host B (or the gateway to the host B if it is in different subnet) will respond its MAC address if the request is matched with its own IP address. This document will not spell out those details; we will just outline some principles for the ease of planning and deployment.

In general, if “VLAN” is not used, you can think that network equipments connecting to a bridge (Ethernet switch) should belong to the same IP subnet. For network traffic going across different subnets, there shall be a router sitting in between – in other words, router is with one leg sitting in one subnet and other legs sitting in different subnets.

Virtual Bridge and Virtual Host

The Azblink NFV platform provides “virtual bridges” and “virtual hosts” so that the other operation systems (like Windows or Ubuntu) can be installed into virtual hosts. In the following sections, we will refer Azblink NFV platform or the system able to create virtual host as “**base platform**”, “**base OS**” or “**host OS**”, and the operation system in the virtual host as “**guest OS**”.

On “base platform” , the network operation policies will be using “virtual bridges” as a unit for network operation. Once a virtual host is created, the associated virtual network interface(s) should be created along with the virtual host too. We will force you to choose the virtual bridge that each of the virtual network interface of the virtual host should connect to.

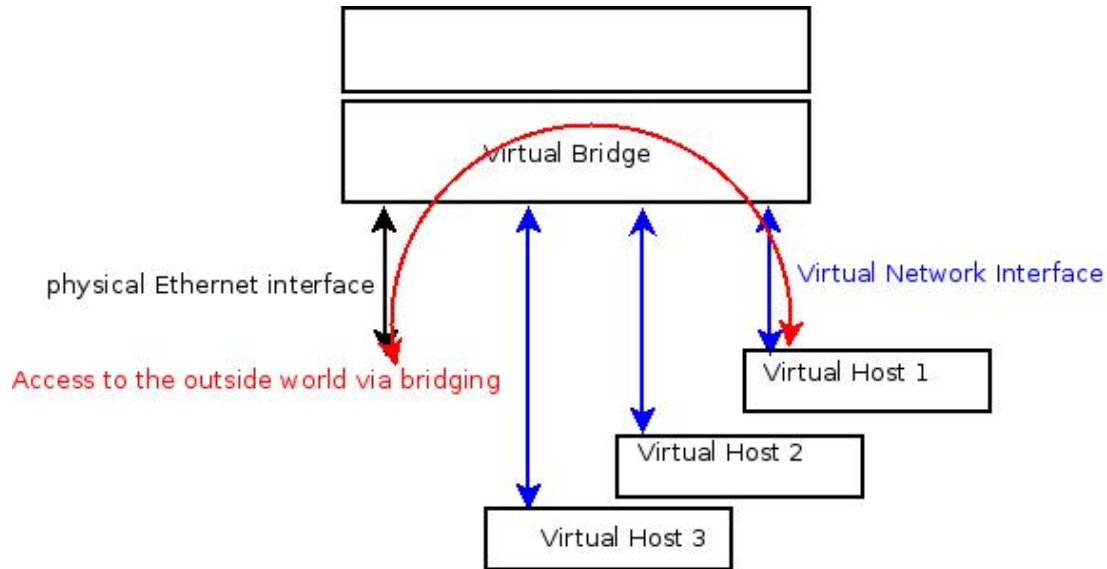


Illustration 3: Traffic Inside the Bridge

Each of the “virtual network interfaces” of virtual hosts will be placed into a virtual bridge. If a physical Ethernet interface is also added into the bridge, a virtual host can access the network outside the base platform via that interface by “bridging process”. The “virtual bridge” also acts as a super Ethernet Interface in the base platform so that it can have its own IP address with proper netmask to indicate the IP subnet it belongs to. You might view that the IP address of the bridge is the interface to reach the local network processes on the base platform. From the side of base platform, the network interfaces inside the bridge (no matter it is physical interface or virtual interface) do not have IP address set on those network interfaces on the base platform. On the other side viewed from the virtual hosts, the guest OS(s) can set their own IP addresses with virtual hosts.

To be specific, assume the virtual bridge is named “**br0**” and it has IP address set on it. Inside the virtual bridge “**br0**”, you might find “**eth0**” and other virtual Ethernet interfaces – from the side of base platform, those Ethernet interfaces within bridge do not have IP addresses set on them.

In practice, this “**virtual bridge**” can be viewed as an Ethernet switch. The hosts connected via the same switch should belong to the same IP subnet unless VLAN is used. The reason to partition IP network into subnets is to isolate network traffic in a relatively local environment without global impact. For example, if an ARP request is sent, every host receiving the request should check if its own IP address is matched with the one in the request. If such requests happen quite often, the resource spent on this kind of activities might not be efficient given the fact that only one host is the real target of each ARP request. Therefore, network broadcast in IP network is only restricted in the same subnet; it will not cross to another subnet. ARP request is done via IP network broadcast and there exist other IP-based protocols also dependent on broadcast. For those hosts relying on those network protocols to communicate with each other, they should be placed into the same IP subnet.

There are chances that the network packets of the virtual hosts will go across the subnet. In the following diagram, if the gateway to the subnet of the target host can be reached via that physical Ethernet interface in the virtual bridge, the packets will go to that gateway via that physical Ethernet interface. However, the base platform is also with IP addresses on the bridges. If the base platform is with a network interface attached to the subnet that the network packets from virtual host are destined for, they could go via base platform through routing process to that subnet.

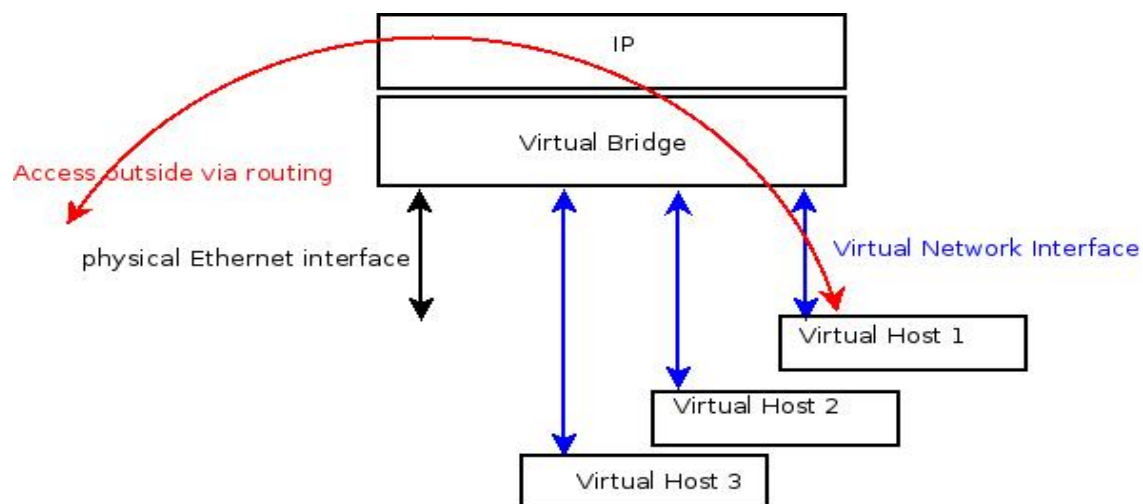


Illustration 4: Traffic Going Across the Bridge to another Subnet

To sum up, although the virtual bridges and virtual hosts are created by the base platform inside it, the base platform can be treated as a host that is parallel to the virtual hosts at the level of IP network. Whenever a network application in a virtual host wants to contact the base platform, it just sends network requests by using the IP address(es) of the base platform.

Other Possible Network Operations(But we do not provide)

In theory, there exist other operations between a virtual host and the physical Ethernet interface it is attached to on the base platform. We list some of the network operations that **they are not provided** on the product. The reason why they are not provided is that Azblink NFV platform is for “server-type” applications. The operation schemes indicated below might not be appropriate for “server-type” applications. For the sake of the completeness, we still discuss them here. In the next section, we will illustrate what Azblink NFV platform provides.

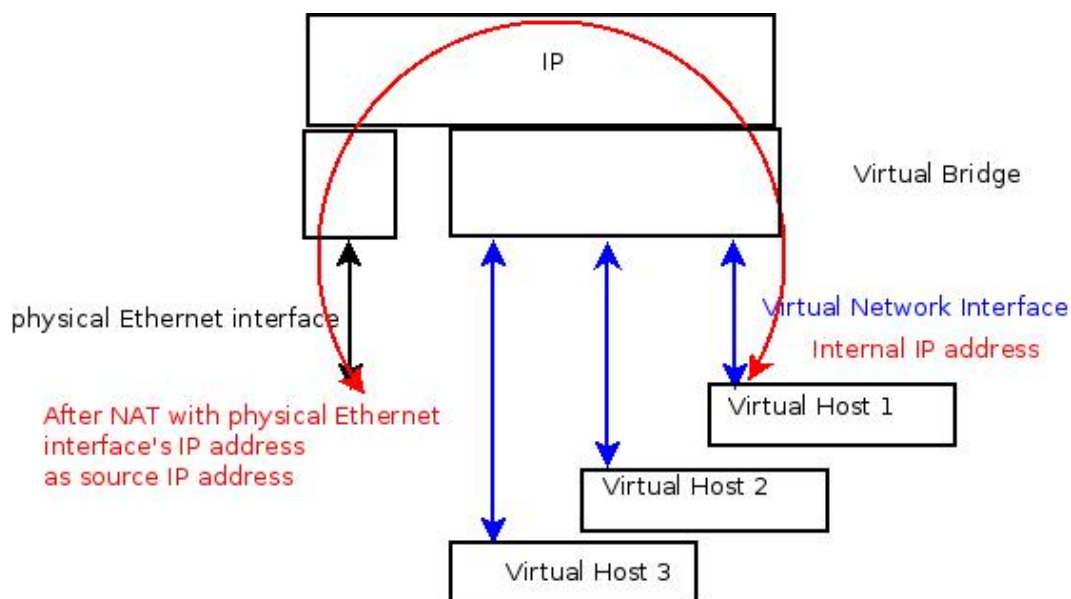


Illustration 5: NAT on per physical port basis

The diagram above indicates that a virtual host is hidden after NAT (Network Address Translation). The scenario usually happens when a virtual host is created on a “base OS” with IP address specified on the physical Ethernet interface. Once the virtual host is created, a corresponding bridge is also created by nullifying the IP address previously set on that physical Ethernet interface; and then, use that IP address on the bridge device, put that Ethernet device into the bridge, and give a private IP address to that virtual host. If the network packets from the virtual host wants to access to the hosts outside the “base OS”, the packets will be replaced with the IP address of the bridge device (the original IP address on the physical Ethernet interface).

To access that host from the world outside the “base OS”, port forwarding is needed.

The following diagram is a scenario that virtual hosts that are behind the same IP address under NAT, but they can communicate with each other with their own private IP addresses without going through NAT; this is implemented by connecting them on the same bridge.

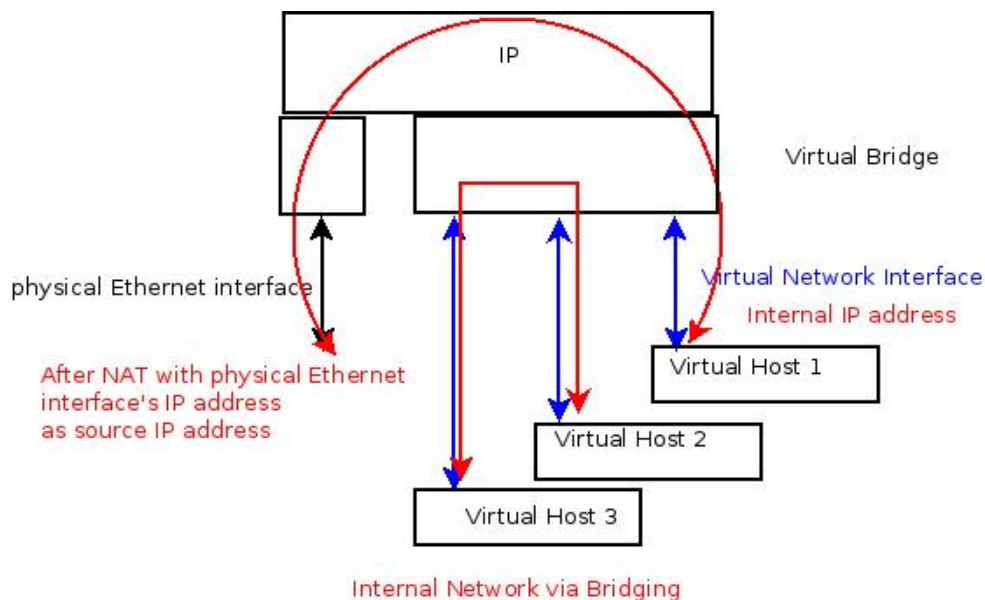


Illustration 6: Allowing Sending Traffic between Virtual Hosts on the same bridge

There exists other scenario that the virtual hosts can not communicate to each other although they are under the same IP address of NAT; they are only allowed to access the outside world via NAT, but they are not allowed to reach each other by using their own private IP addresses. This scenario sounds strange, but it might have the reason to do so when the two virtual hosts are owned by different companies in a cloud-based environment.

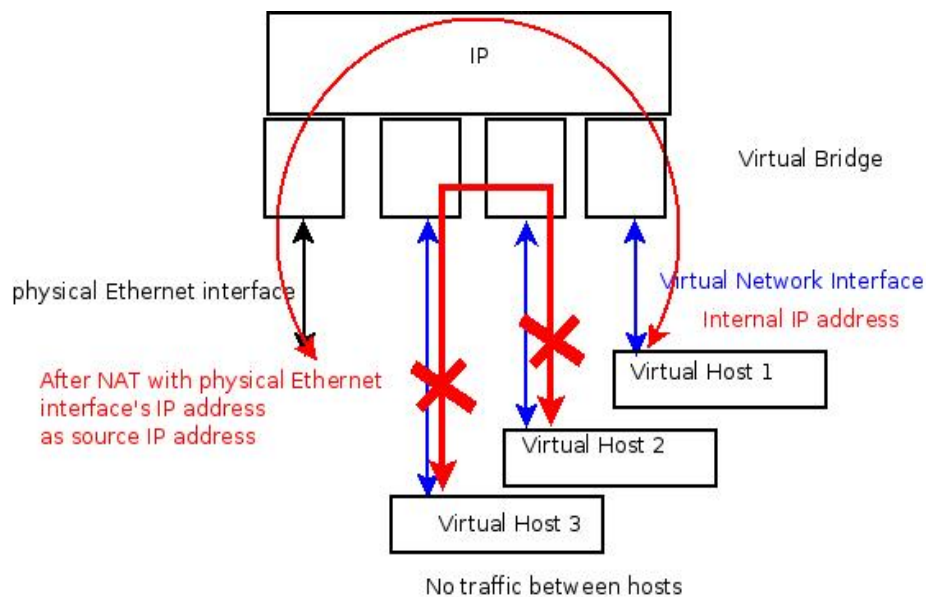


Illustration 7: Traffic not Allowed between Virtual Hosts on the same bridge

The following diagram is to indicate that NAT function is removed so that the virtual hosts can not access outside world; they only can communicate to each other with their local private IP addresses via the same bridge.

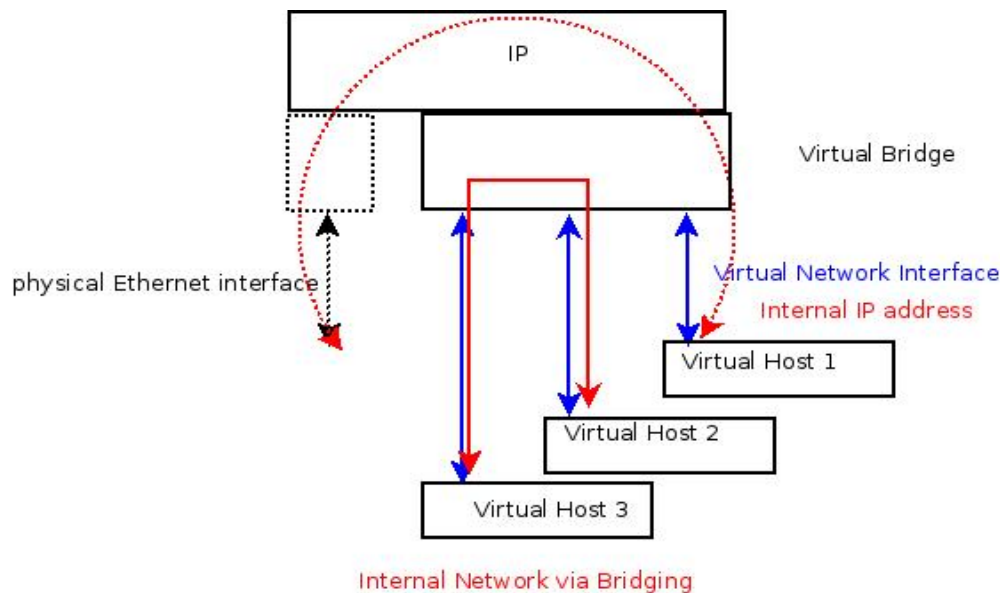


Illustration 8: Traffic Locked inside a Bridge

The operations mentioned above are not provided on our base platform. We only ask you to decide the virtual bridge that your virtual network interface should connect to.

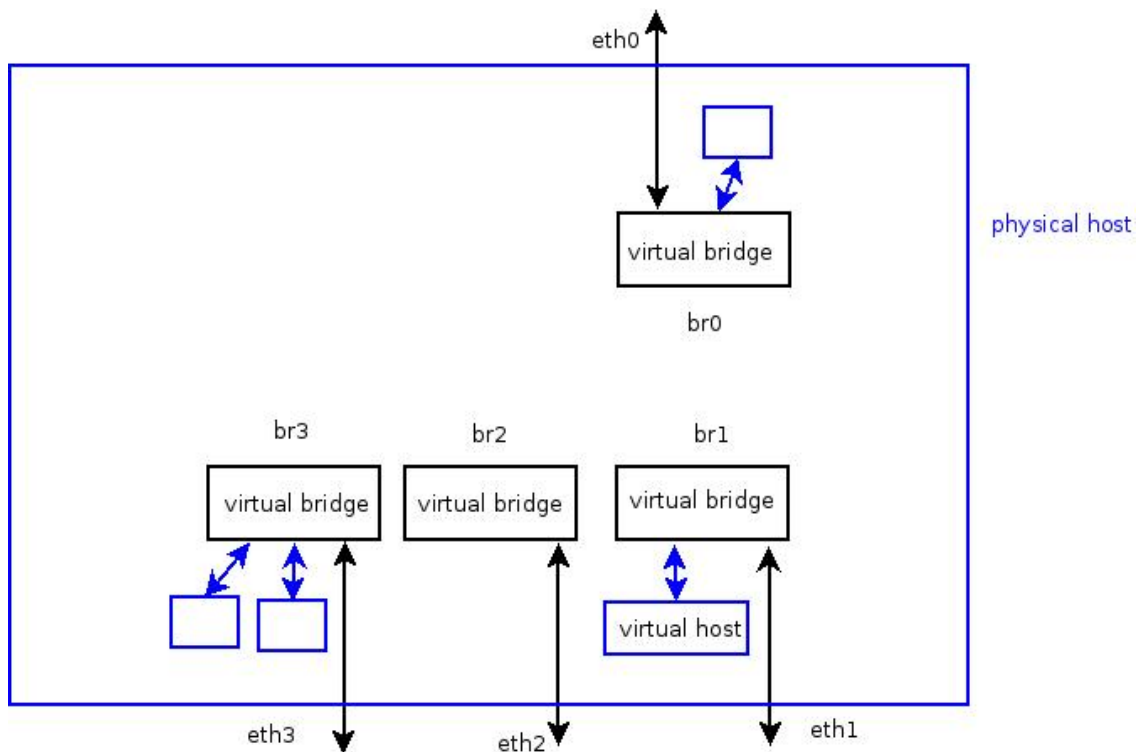
What if some of the functionalities (e.g. NAT) are needed during system deployment? We manage the system with firewall operations on the global arrangement of virtual bridges; each bridge is with its own functions defined in advance. For example, network packets from one subnet A (connecting via a bridge) to another subnet B (connecting via another bridge) might have NAT (Network Address Translation) operation if it is defined in base OS – if it is defined, the source IP address will be replaced by the IP address of the bridge connecting to the destined network. Those will be introduced in the following section.

Firewall Operations on Bridges

Azblink NFV platform provides multiple virtual bridges with predefined rules between those virtual bridges. For example, network traffic from bridges “br1”, “br2”, and “br3” to WAN via “br0” will go through NAT (Network Address Translation) by replacing the source IP address in the IP header with the IP address of “br0”; network traffic can go from “br3” to “br1” and vice versa. It is not allowed to go from “br1” to any other subnets, but traffic from the other subnets to “br1” is allowed. We will explore more details later in this document. The network traffic initiated from the hosts under “br0” is not allowed to go into the other bridges.

The benefit of this configuration is: it is straightforward to place virtual hosts into the different zones according to the permission levels. And the firewall can control the access level in a global way with careful planning in advance. For each virtual host, if it is necessary to be deployed across different zones, it can have multiple virtual network interfaces placed into different bridges.

Illustration 9: Virtual Bridges inside Base Platform



The following diagram shows the network traffic from “br1” or “br2” via “br0” will be processed by NAT (Network Address Translation) with the IP address of “br0” as source IP address – no matter it is from a virtual host or physical network interfaces under the bridge “br1” or “br2”.

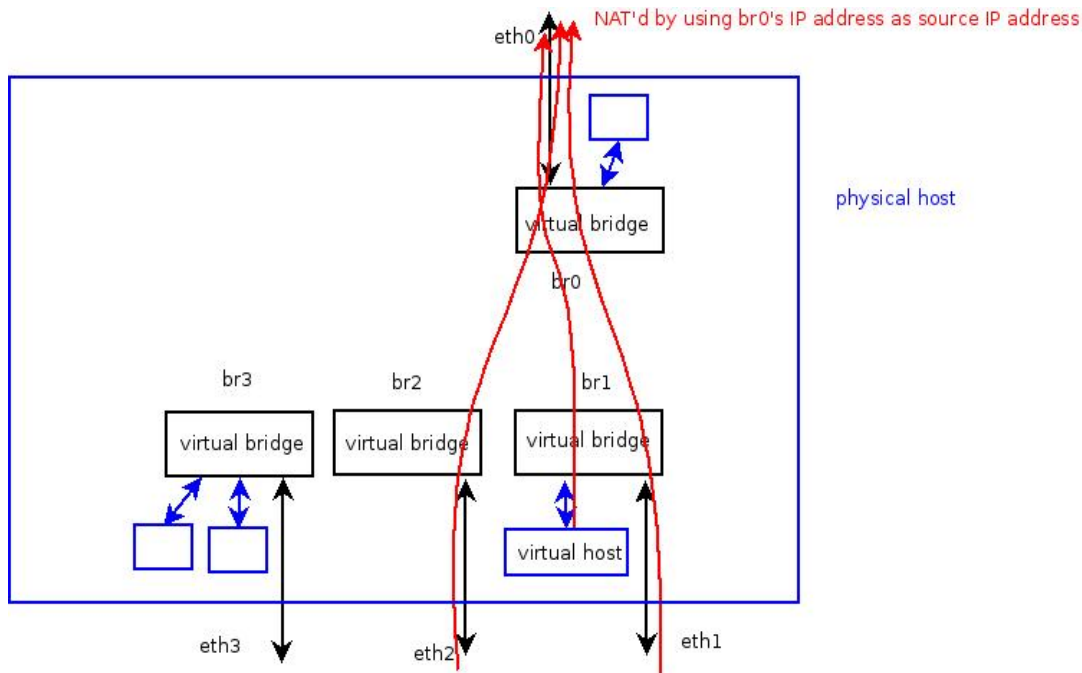


Illustration 10: NAT while going across the boundary of br0

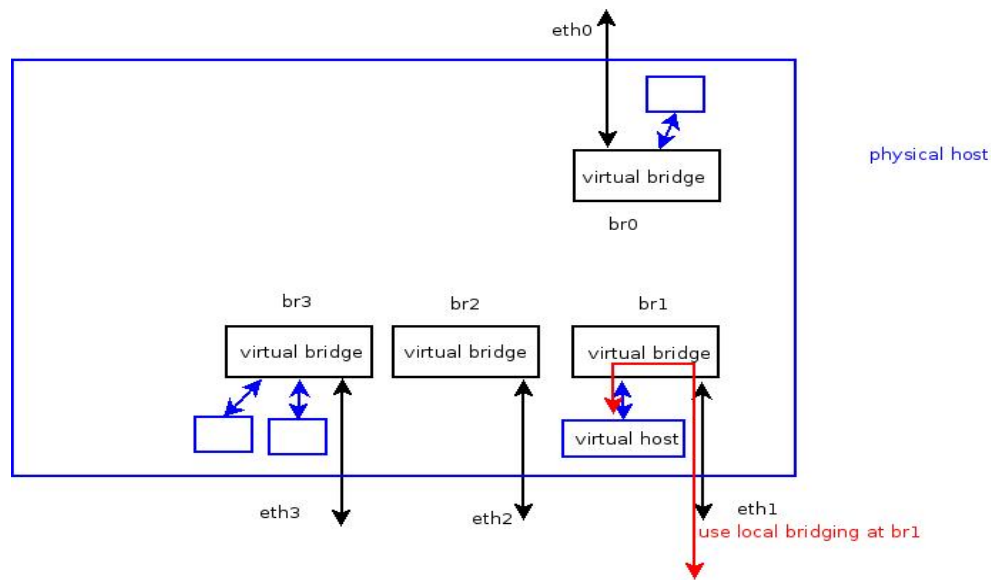


Illustration 11: No NAT Operation inside Bridge br1

And the virtual host placed under “br1” can be accessed via “eth1” without restriction from “base platform”. It depends on the virtual host itself to implement its own access control or not. If the virtual host does not have the control scheme, the firewall of the “base platform” can enforce some extra rules to have more finer control.

It is forbidden to access the hosts in “br1” or “br3” from the hosts in “br0” unless port forwarding is used.

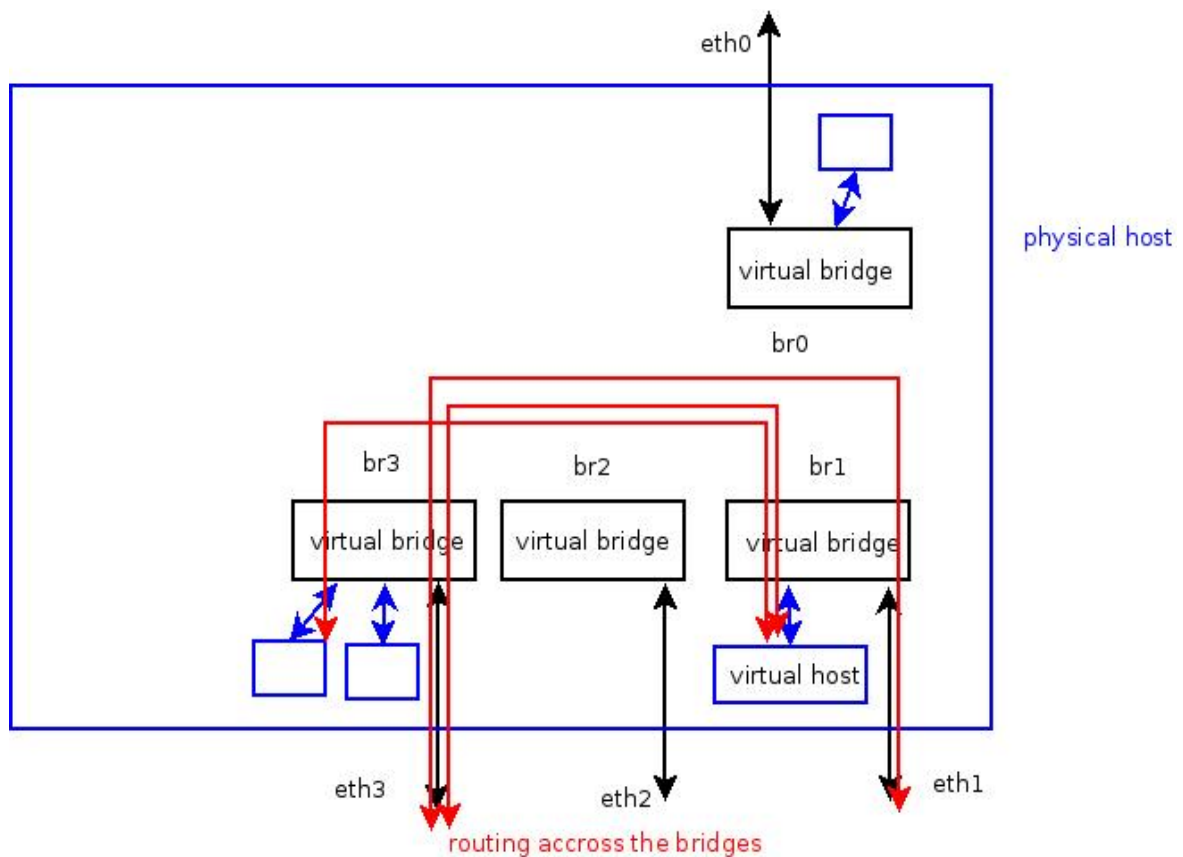


Illustration 12: No NAT Operation while going across the bridge boundaries br1 and br3

The hosts under “br1” and “br3” can exchange network traffic via the routing process provided by the base platform. Although those virtual hosts under the bridges are enclosed within the same physical entity, logically they are independent from the base platform. They have their own “network identities”; they are exactly the same as the other hosts outside the enclosed physical entity. From network point of view, they only can be accessed by their own IP addresses other than console. Thus, logically, they should be viewed as the hosts outside the physical host – just like the other hosts.

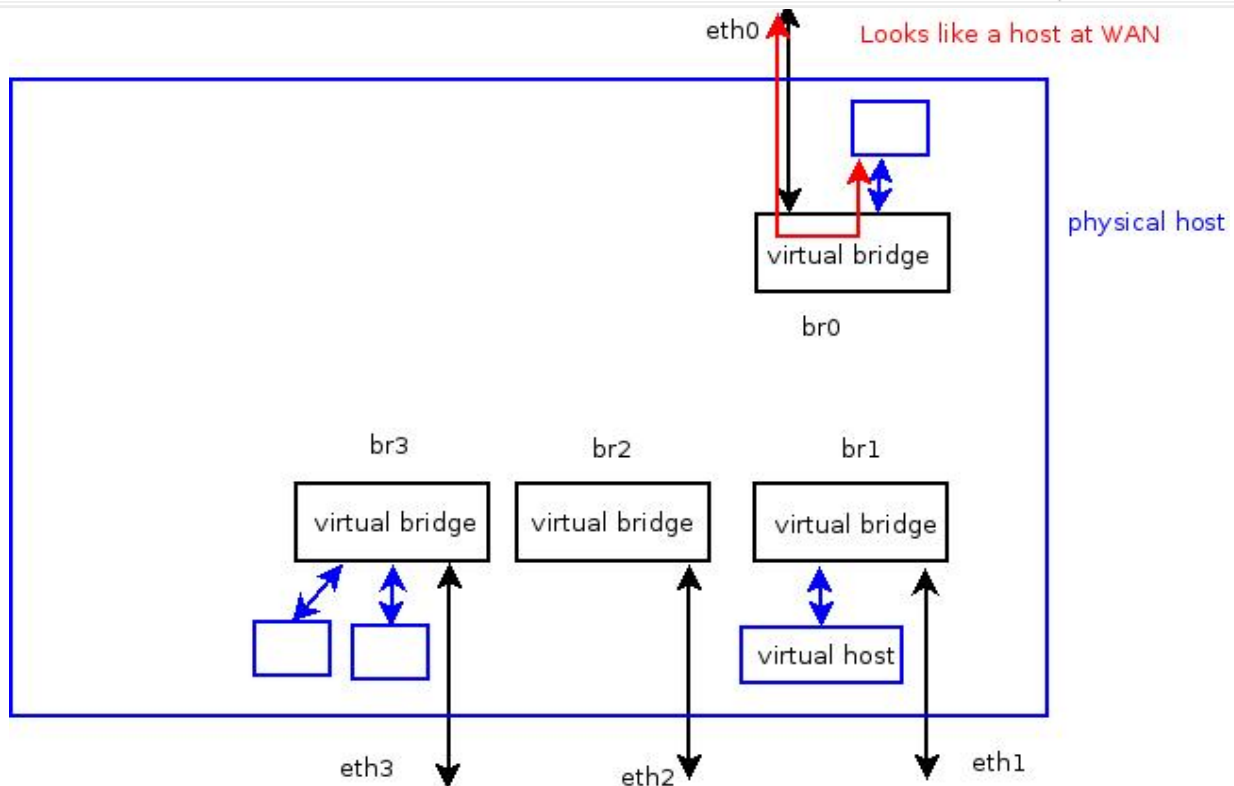


Illustration 13: Virtual Host connecting to the Bridge br0

We use a virtual host under “br0” as an example. A virtual host is created under “br0”. The bridge “br0” is with an IP address and that is the identity of the base platform; the virtual host is also with its own IP address that is on the same subnet where “br0” resides. Although the outbound traffic of this virtual host to the Internet should go through “br0”, the traffic will not be touched by the operation of NAT; it goes out via “bridging” with its own IP address as source address. Thus, this virtual host is **not protected by the firewall of the base platform**. The other hosts under “br1” or “br2” just view the host in “br0” as a host “outside the firewall”.

Similarly, a virtual host can have multiple “legs” into multiple virtual bridges. The following diagram is a virtual host with virtual network interfaces to the bridges “br3” and “br4”. Thus, for network traffic originated from this virtual host to the two subnets where “br3” and “br4” reside, the virtual host is with direct link to the two subnets; they do not have to go through routing process of the base platform. There exist some network protocols that can not go beyond a subnet. With this configuration, this virtual host can apply those network protocols to the two subnets respectively.

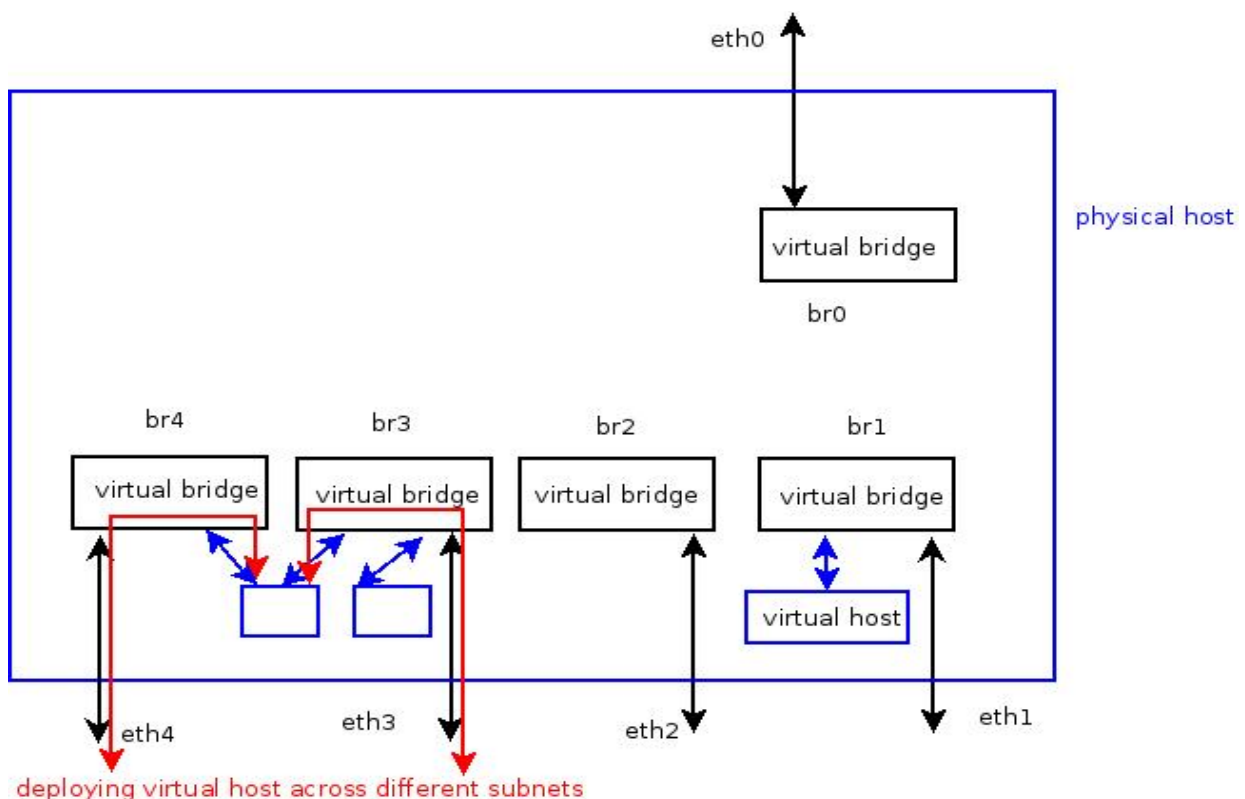


Illustration 14: Virtual Host connecting to multiple bridges

Firewall Virtualization or SBC Virtualization

If a virtual host is with two virtual network interfaces: one is under “br0” and the other is under “br1”, then this virtual host can be implemented as “virtual firewall” or “virtual SBC”.

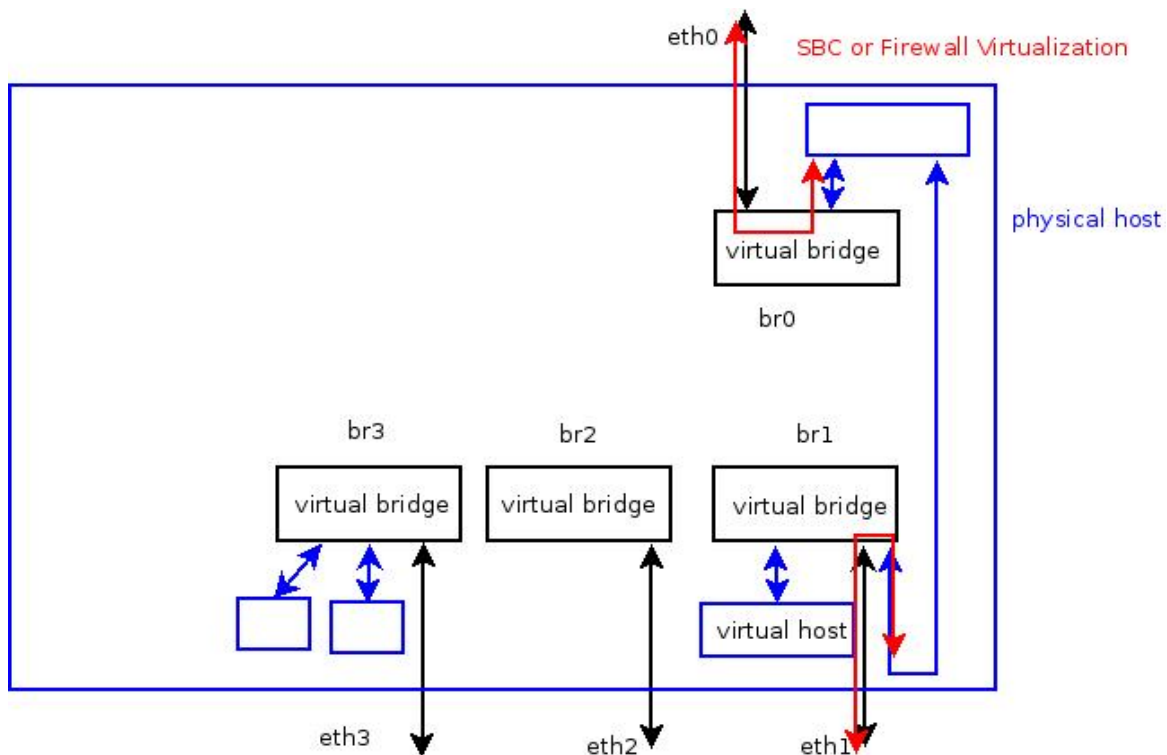


Illustration 15: Virtual Host connecting to Bridges br0 and br1

This virtual host is with one interface in WAN and another in LAN so that it has the potential to become “virtual firewall” or “virtual SBC”. Since the base platform itself is a firewall as well, how to choose the firewall to use from the subnet that “br1” connects to? If the machines in that subnet use the IP address of that virtual firewall as default gateway, the traffic will route through virtual firewall to the Internet.

To force people to use virtual firewall, it only needs to turn off the DHCP server and provision the rule to forbid the traffic from “br1” to “br0” on the base platform. Therefore, people have no choice but use the virtual firewall to go to Internet from that subnet.

Similarly, “virtual SBC” can be deployed in the same way.

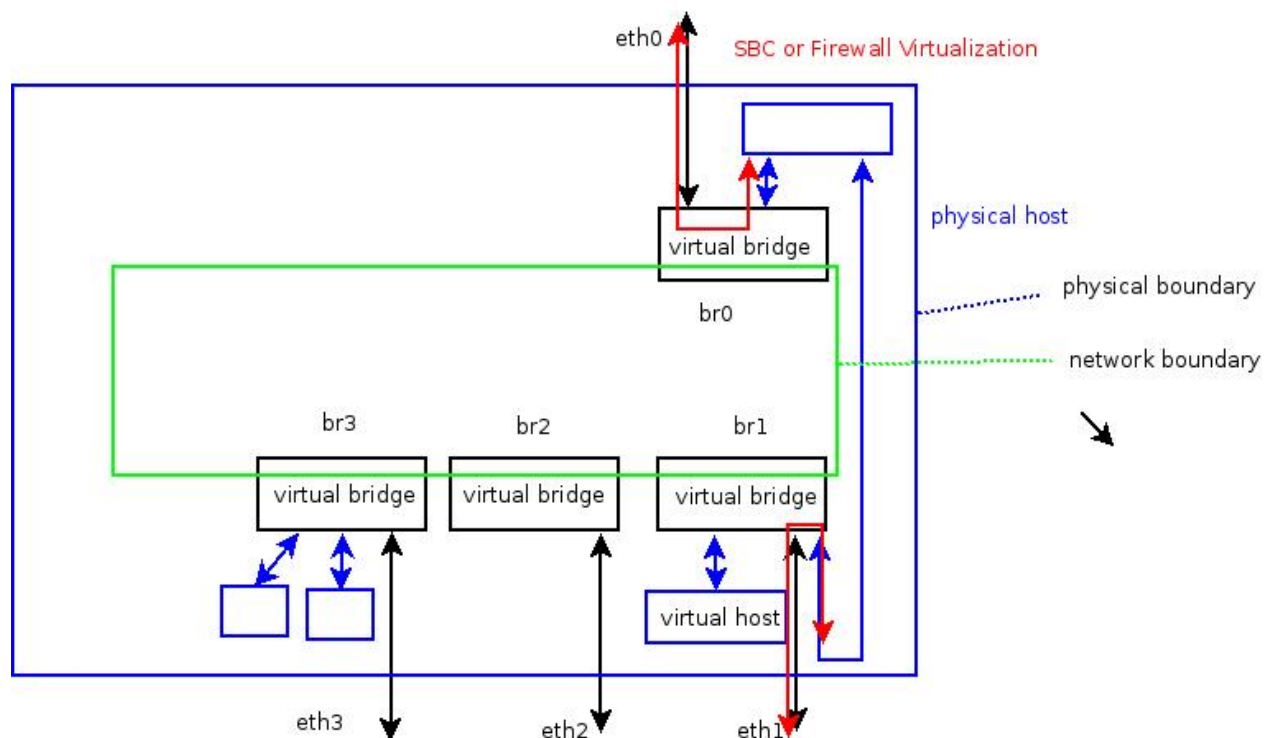


Illustration 16: Virtual Host Connection for Firewall Virtualization

To avoid the confusion of the relation between “base platform” and “virtual host”, we might as well consider those virtual bridges as multiple “Ethernet switches”. The base platform has multiple legs plugged into those “Ethernet switches” with IP address defined on each leg (IP address defined on its bridge interface) whereas the virtual hosts also have legs into “Ethernet switches”. Each “Ethernet switch” is with hosts in the same IP subnet.

The benefit of virtualization is that adding a “network interface” into a “bridge” can be done via software reconfiguration without actually putting a network interface card and wiring cables. And the base platform provides network method to access the consoles of the virtual hosts so that the console of a virtual host can be accessed remotely; the guest OS does not need to install any other software packages for this function.

The base platform is also with OSPF and RIPv2 so that it can interface with the other routers to exchange routing tables automatically.

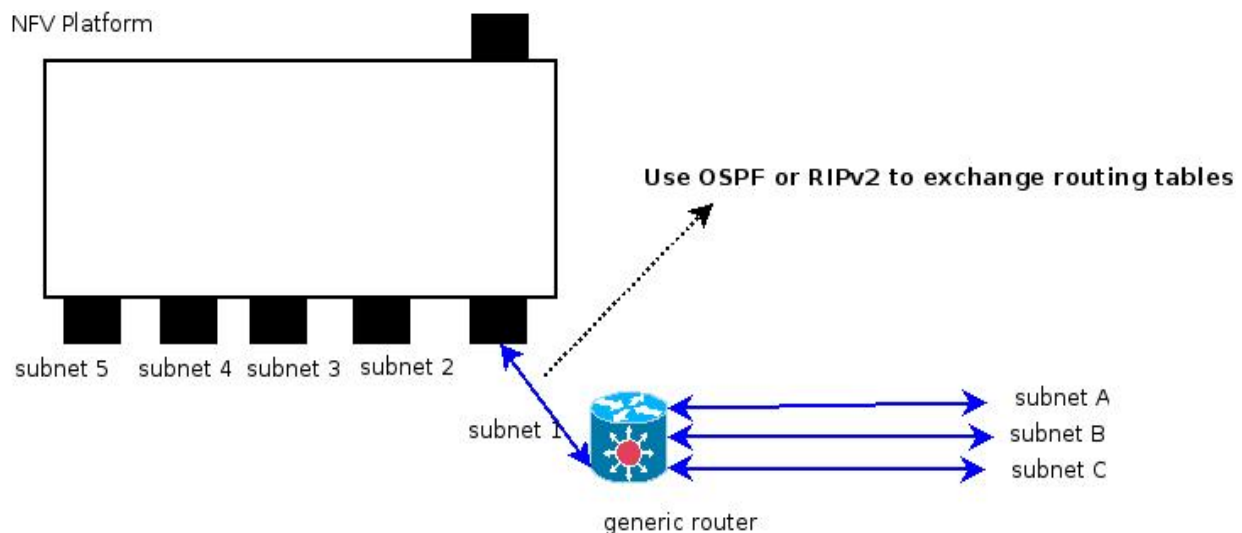


Illustration 17: Base Platform and Other Routers

Furthermore, the base platform also provides firewall functionalities; it is known as “Border Control” that will be introduced later in this document. It is possible that you might implement your own firewall in the virtual host by connecting the virtual bridges indicated earlier in this section. In the following chapter, we will introduce how to create a virtual host on the base platform.

Chapter 2 Virtual Host

The purpose to use virtual host is in the hope that it can function like a host with physical entity. A lot of applications are relying on the physical interface of hardware(for example, RS232 serial port link, or Bluetooth) to function. However, our product here is only focusing on network application. It means the major application transactions are going through network. You might need to pay attention if your application is of this category.

Conceptually, the “virtual host” we mean here is like a container for OS (Operation System) Software to function. From the OS point of view, the virtual host provides CPU, memory, storage space, network interface(s), and other types of peripherals for it to function well – just like a physical machine does. To distinguish the OS providing “virtual hosts” from the OS running on “virtual hosts”, we call the OS providing “virtual host” as “**host OS**” or “**host system**” or “**base platform**” or “**base OS**”; and the OS running on “virtual host” as “**guest OS**” or “**guest system**”. The term “virtual host” sometimes includes guest OS. But in this section, it only stands for the collection of those emulated hardware components.

For a physical machine to install OS, it is usually loaded from the media like CD or DVD or from the network via TFTP. The virtual machine does not have those physical entities, but it needs to simulate the environment similar to that. To install OS into the virtual machine, you need to prepare the CD/DVD image of this OS (usually in ISO format). Once the CD/DVD image is placed on the proper place, the host OS can load it into the virtual machine.

Before we use the virtual host, it is necessary to decide the emulated hardware spec; for example, the size of memory and storage space, how many CPUs can be used by the guest OS, or how many Ethernet interfaces that the virtual host should have. Some of the attributes must be decided correctly at the very first beginning; some of them can be changed later. For example, the size of the storage space can be changed at any time. However, once the OS is installed, resizing the storage space needs to re-partition and format the hard drive from the guest OS. It is not like resizing the memory that the guest OS can utilize it immediately. Those issues need to be considered at first.

The base platform can have multiple instances of virtual hosts running on it. If the total amount of memory allocated to the virtual hosts in operation is larger than the actual memory that the base platform can provide, then some of the virtual hosts might not be able to boot up or some of them might encounter memory-exhausted problems.

The “hardware” viewed from the guest OS in virtual host is all emulated by the base platform. Thus, if the complexity of emulated hardware is reduced, the burden of the base platform is reduced. In order to have better performance, the selected peripherals of the virtual host should just meet the requirements of the guest OS without overbooking.

As mentioned earlier, the base platform is to facilitate the network arrangement of the virtual hosts. Each virtual host can have multiple Ethernet interfaces placed into different bridges, and each bridge belongs to different subnet. The role of each bridge is governed by the base platform. Each guest OS might have different usage on the Ethernet interface(s). For example, one uses DHCP client to fetch IP address whereas the other uses DHCP server to distribute IP addresses for its clients to use. In this case, it is necessary to note if there exist two DHCP servers on the same subnet; you should only allow one to avoid collision. If a bridge A is NAT'd by the other bridge B, the network traffic from the hosts connecting to bridge A to the hosts in bridge B will be NAT'd by using the IP address of bridge B as the source IP address in the IP header. Those constraints should be studied before deploying software packages in virtual hosts.

Along with choosing the bridge to join, setting a network interface also needs to specify the emulated Ethernet card in virtual machine. And the guest OS should have the proper driver in order to use Ethernet.

Providing high-precision clock sources in virtual hosts is stressful for base platform. However, clock timing source is related to how frequent the system should react to the system events. If you find the sluggish response from the guest OS, it might be something to do with the system clock timing source. Windows System usually use low-frequency clock so that it will not have clock source issue. On Linux-based system, if clock source issues are encountered, please switch to use **KVM clock**.

For each virtual host, a TCP port is assigned so that you can connect the console of the virtual host by using “**VNC Viewer**” (or “**SPICE**” client). This network connection is also governed by the access policies of the base platform. You have to connect from the network where it allows to access the host.

To use the web management interface, access via

`http://ip_address:8082/apps/`

The account “**admin**” is with default password “**admin123**”. You might change the password later via the web management interface.

Upload CD Image

Upload CD Image for Installation

System >> Host >> Upload CD Image

Select Image file to Upload

No file selected.

Select the Image File to Delete

- ☐ WinXP_SP4.iso
- ☐ Windows10_64.iso
- ☐ Windows7Professionalx64SP1.iso
- ☐ debian-9.4.0-amd64-xfce-CD-1.iso
- ☐ ubuntu-11.10-desktop-i386.iso
- ☐ ubuntu-17.10.1-desktop-amd64.iso
- ☐ ubuntu-18.04.1-desktop-amd64.iso
- ☐ ued102.iso
- ☐ ued103.iso
- ☐ ued106.iso
- ☐ ued107.iso
- ☐ ued108.iso
- ☐ ued109.iso
- ☐ ued91.iso
- ☐ ved798.iso
- ☐ ved798w.iso
- ☐ ved799.iso
- ☐ ved799ww.iso
- ☐ ved800v.iso
- ☐ ved805.iso
- ☐ ved807.iso

Illustration 18: Uploading CD/DVD Image for Guest OS

In order to boot from CD/DVD drive to install OS into virtual host, you need to provide the file of the OS in **ISO format** and upload it via "**System >> Host >> Upload CD Image**". The size of each image should not be larger than half of the physical memory of the base platform. Otherwise, the loading of the image to the system will fail. In that case, you need to use "**scp**" or the other methods to deliver the image to the destined folder (**"/home/qemu/iso"**). Please note that "**sshd**" shall be running if "**scp**" is used from the remote host. And TCP port 22 shall be open to this remote host if it is accessing from the WAN side (the area we labeled as "net" in our firewall)).

Create an instance of virtual host

Add Virtual Host

System >> Host >> Add/Delete Host

Add Host

Host ID

Disk Space GB ☐ Emulate NVMe

Memory Allocated MB

Bridge Number(s) to Join (0 1 2 3 4 5 6 7 8 9 10 11)

Ethernet Model Intel e1000-82540em

Sound Device Intel AC97 Audio

☐ USB Tablet (especially for Windows)

VNC Port for Console (5900 as basis; 5 for 5905, 6 for 5906)

SPICE Port for Console (for example, 5801 or 5802)

Select the Host to Delete

☐ duda (VNC 5905, SPICE 5805)

☐ ped (VNC 5902)

☐ ubuntu (VNC 5903)

☐ win10 (VNC 5901, SPICE 5801)

☐ winXPa (VNC 5907, SPICE 5804)

Illustration 19: screen snapshot for creating an instance of virtual host


To create an instance of virtual host, it is necessary to specify “Host ID” other than Disk Space, Memory, Bridge(s) to join, the Ethernet interface model, and the port for VNC connection. The “**Host ID**” is for you to identify the virtual host on the base platform. It is nothing to do with the host name or any identifier used by the guest OS.

The check box “**USB Tablet**” is used to resolve the mouse pointer synchronization issue between VNC and the host you are using for connecting virtual host. If the guest OS is Windows system, you need to click this check box. By default, PS2 keyboard and mouse are provided.

Some of the SPICE clients can provide the voice from **“Guest OS”** to **“Client OS”** (the OS where you are running SPICE client). If you need to use this scenario, you should properly choose Audio device. On old operations systems (like Windows XP), it only can detect AC97 for Audio and RealTek 8139 for Ethernet. However, on newer operation systems (like Windows 7, or Windows 10), AC97 can not be used; it is necessary to use Intel HD audio along with the chipset ICH9. We can not list all the possible conditions here and provide all the solutions; you should evaluate the proper emulated hardware before you install the desired OS you want.

Once you finish it by clicking “Add”, the “Host ID” will appear on the right side. If there is nothing to modify or fine-tune, you can just head to **“System >> Host >> Host Management”** to install CD/DVD image to the virtual host.

Bridge Assignment

 **Add Host Networking Interface and Place into Bridge**

System >> Host >> Bridge Assignment

Add or Delete Network Interface

Host ID

Bridge Number

Ethernet Model

Intel e1000-82540em

Delete


Add

Host ID	Bridge Number
duda	00:90:FB:0E:D3:10--->1
ped	00:90:FB:99:B6:69--->0
	00:90:FB:3D:14:0A--->1
	00:90:FB:15:8E:5B--->2
ubuntu	00:90:FB:53:C6:DA--->0
win10	00:90:FB:9D:98:EA--->0
winXPa	00:90:FB:29:70:87--->1

Illustration 20: Bridge Assignment and Ethernet Type of a Virtual Host

In this web page for admin, it lists the Ethernet interface (by its MAC address) and the associated bridge number. If the setting needs to be changed, type "Host ID" and "Bridge Number" above, and press "Delete". And add by entering "Host ID", "Bridge Number" that the Ethernet interface tries to join, and Ethernet model; press "Add" button.

Change the Setting of Memory and Tablet

 **Change the Setting of Memory and Tablet**

System >> Host >> Memory and Tablet

Reset the Size of Memory and Tablet

Host ID

Memory Size(MB)

☐ USB Tablet ☐ USB Mouse


☐ USB Keyboard

Host ID	Setting
duda	2048 MB
ped	2048 MB
ubuntu	2048 MB
win10	2048 MB usb-tablet
winXPa	2048 MB usb-tablet

Illustration 21: Changing Memory Size and Enabling USB Tablet of VM

If the size of the memory needs to be changed, it can be reset here. If the USB tablet can be used if you find out that your mouse pointer is with synchronization problem between “Guest OS” and “Client OS” when using VNC viewer. As we mention earlier, PS2 mouse and keyboard are provided by default. If your “Guest OS” support them, you should not choose others. However, some operation systems (for example, Mac OS X “Mojave”) does not support PS mouse and keyboard and USB tablet. In this case, we would suggest you select USB mouse and keyboard and use “SPICE” client (like “**virt-viewer**” or “**remote-viewer**”).

CPU and Chipset

 **Change the Setting of CPU and Chipset**

System >> Host >> CPU and Chipset

Change the Number of CPU(s) and Chipset

Host ID
Processor Emulated

Add CPU Flag(s)
☐ pae
☐ sse3
☐ sse4.2
☐ aes
☐ xsave
☐ avx
☐ xsaveopt
☐ xsavec
☐ xgetbv1
☐ avx2
☐ bmi2
☐ smep
☐ bmi1
☐ fma
☐ movbe
☐ invtsc

Number of CPU(s)
Sound Device Intel AC97 Audio

☐ Emulate Intel ICH9 Chipset (Otherwise PIIX)

Submit

Host ID	Setting
duda	Penryn,kvm=on,+sse4.2,+aes,+xsave,+avx,+xsaveopt,+xsavec,+xgetbv1 4 CPU(s) ICH9 HDA
ped	1 CPU(s) PIIX HDA
ubuntu	host 2 CPU(s) ICH9 HDA
win10	1 CPU(s) ICH9 HDA
winXPa	1 CPU(s) PIIX AC97

Illustration 22: Changing Number of CPUs and Chipset

The chipset Intel PIIX is used by default while creating an instance of virtual host. The chipset Intel PIIX is mainly for PCI-to-ISA, PCI IDE functions, and audio device like AC97. And Intel ICH9 chipset is mainly for SATA (Serial ATA). Thus, if you plan to emulate IDE hard drive, use PIIX; for SATA, use ICH9. Some of the old systems do not support SATA drive. In this case, you need to use PIIX.

Usually we will use the same CPU model as the “Base OS” has for the emulation on “Guest OS”. In case some of the Operations Systems need to use specific CPU model and CPU flag, you might select them from this screen as well.

Storage Device Setting

Storage Device Setting
System >> Host >> Storage I/O

Specify Extra Storage Device(s)

Host ID

☐ Use Program Flash (to Replace BIOS)

☐ Extra Program Flash

☐ Emulate Intel ICH9 AHCI

☐ Use Extra Hard Drive 0

☐ Use Extra Hard Drive 1

☐ Allow USB2.0 Redirection from Client OS

☐ Allow USB3.0 Redirection from Client OS

Host ID	Setting
duda	-hda /home/qemu/vdisks/duda.img
ped	-hda /home/qemu/vdisks/ped.img
ubuntu	-hda /home/qemu/vdisks/ubuntu.img -device ich9-usb-ehci1,id=usb2 -device ich9-usb-uhci1,masterbus=usb2.0,firstport=0,multifunction=on -device ich9-usb-uhci2,masterbus=usb2.0,firstport=2 -device ich9-usb-uhci3,masterbus=usb2.0,firstport=4 -chardev spicevmc,name=usbredir,id=usbredirchardev1 -device usb-redir,chardev=usbredirchardev1,id=usbredirdev1 -chardev spicevmc,name=usbredir,id=usbredirchardev2 -device usb-redir,chardev=usbredirchardev2,id=usbredirdev2 -chardev spicevmc,name=usbredir,id=usbredirchardev3 -device usb-redir,chardev=usbredirchardev3,id=usbredirdev3

Illustration 23: Extra Storage Device Setting

The screen here may not appear in the standard distribution; it is used to provide “Program Flash” for loading customized BIOS and extra storage devices for specific application. The provided images shall be placed under **“/home/qemu/extra”** in order for showing up in the selection list.

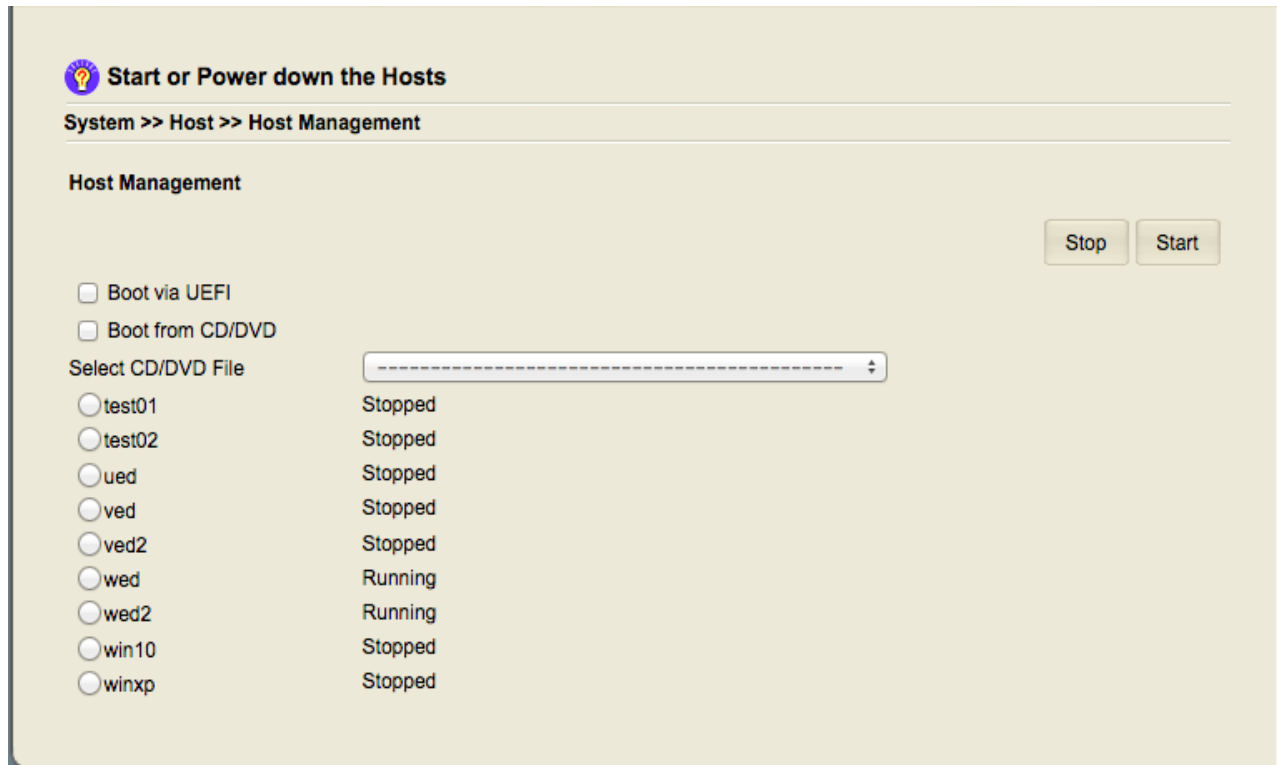
The “base OS” usually provides only one disk as storage device to “guest OS”. If we would like to provision extra storage device(s), those extra hard drives will be placed at “hdb” and “hdd” when using IDE bus. If ICH9 ACHI (SATA) is used, the two extra hard drives will be placed with bus IDs in front of the disk that the “virtual host” originally has. The disk image shall be provided in **“qcow2”** format. You might use the tool **“qemu-img”** on other Linux system to

convert the disk image from the other formats.

USB Redirection happens between “guest OS” and “client OS” while running SPICE client on “client OS”. It does not allow you to use USB redirection at will. On “guest OS”, it needs to detect USB2.0 devices provided by ICH9 USB controller (EHCI and UHCI) or USB3.0 devices provided via NEC chipset. On “client OS” side, the USB device can not be used by the “client OS” during USB redirection. Some SPICE clients would ask you to install extra software package for USB redirection (for example, “**remote-viewer**” needs to have “**usbDK**” for USB redirection to work on **Windows** environment).

If the customized BIOS (or UEFI) image is provided on this screen, you can just boot it from “**Host Management**” screen without specifying any boot options.

Host Management



Start or Power down the Hosts

System >> Host >> Host Management

Host Management

Stop Start

☐ Boot via UEFI

☐ Boot from CD/DVD

Select CD/DVD File

<input type="radio"/> test01	Stopped
<input type="radio"/> test02	Stopped
<input type="radio"/> ued	Stopped
<input type="radio"/> ved	Stopped
<input type="radio"/> ved2	Stopped
<input type="radio"/> wed	Running
<input type="radio"/> wed2	Running
<input type="radio"/> win10	Stopped
<input type="radio"/> winxp	Stopped

Illustration 24: Host Management of Virtual Machines

For the first time to bring up the virtual host, it needs to boot from CD/DVD image in order to install OS to virtual host. You can click the box “**Boot from CD/DVD**”, **select CD/DVD file**, and choose the corresponding virtual host. Then, press “Start”. If the virtual host can be brought up successfully, it will have “Running” next to its “Host ID”. In some cases, the CD/DVD image is not provided with the boot function but with software packages loaded with it for software installation after OS installation. You might select CD/DVD image without asking to boot from CD/DVD in these cases.

VNC viewer can be used to access the console during the installation process. It depends on the guest OS to arrange the installation procedures. Once the process to install from CD/DVD is finished, you might stop the virtual host and bring it up again without booting from CD/DVD. Please note that you use the **IP address(es) and TCP port of the base platform on VNC viewer** (or SPICE client) to access the console of a virtual machine, not the IP address of the virtual host.

The following is the screen snapshot of using “**VNC viewer**” to access the console of a virtual host for Windows installation:

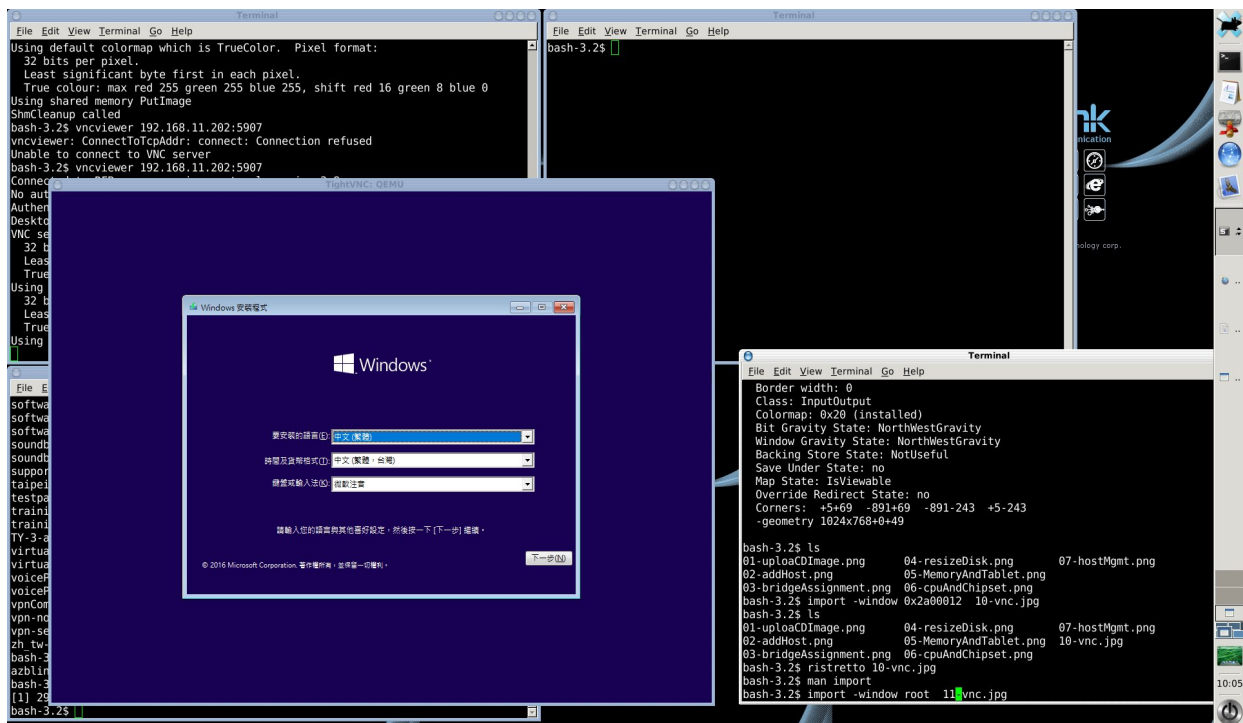


Illustration 25: Screen Snapshot for Using VNC

If using “**UEFI**” to boot system is needed, you have to check the box “Boot via UEFI” as well. Please note that the UEFI firmware provided is for 64-bit system. Thus, the UEFI executables on the corresponding media should be of 64-bit binaries as well.

You can also use SPICE client to access the console of the virtual machine. SPICE client does not work as nimble as VNC client, but it allows the audio to be transmitted to the client side. You can use any base platform's IP address from SPICE or VNC client to access a virtual host. If the virtual host is loaded with **Microsoft Windows**, you can use its **Remote Desktop Service**. To use Microsoft's Remote Desktop, you need to use the **IP address of that virtual host** directly, not the base platform's IP addresses.

Chapter 3 Border Control

The operations of the border control provided by the base platform are **“on bridge basis”**. The term “on bridge basis” means the zone partition is according to the boundaries of bridge devices instead of physical Ethernet interfaces on the base platform. The physical Ethernet interfaces on the base platform are only used for the bridges to connect the networks outside the base platform.

In the previous chapter, what you can do with the Ethernet interface(s) of a virtual host is only to decide the virtual bridges which those Ethernet interface(s) of the virtual host can connect to.

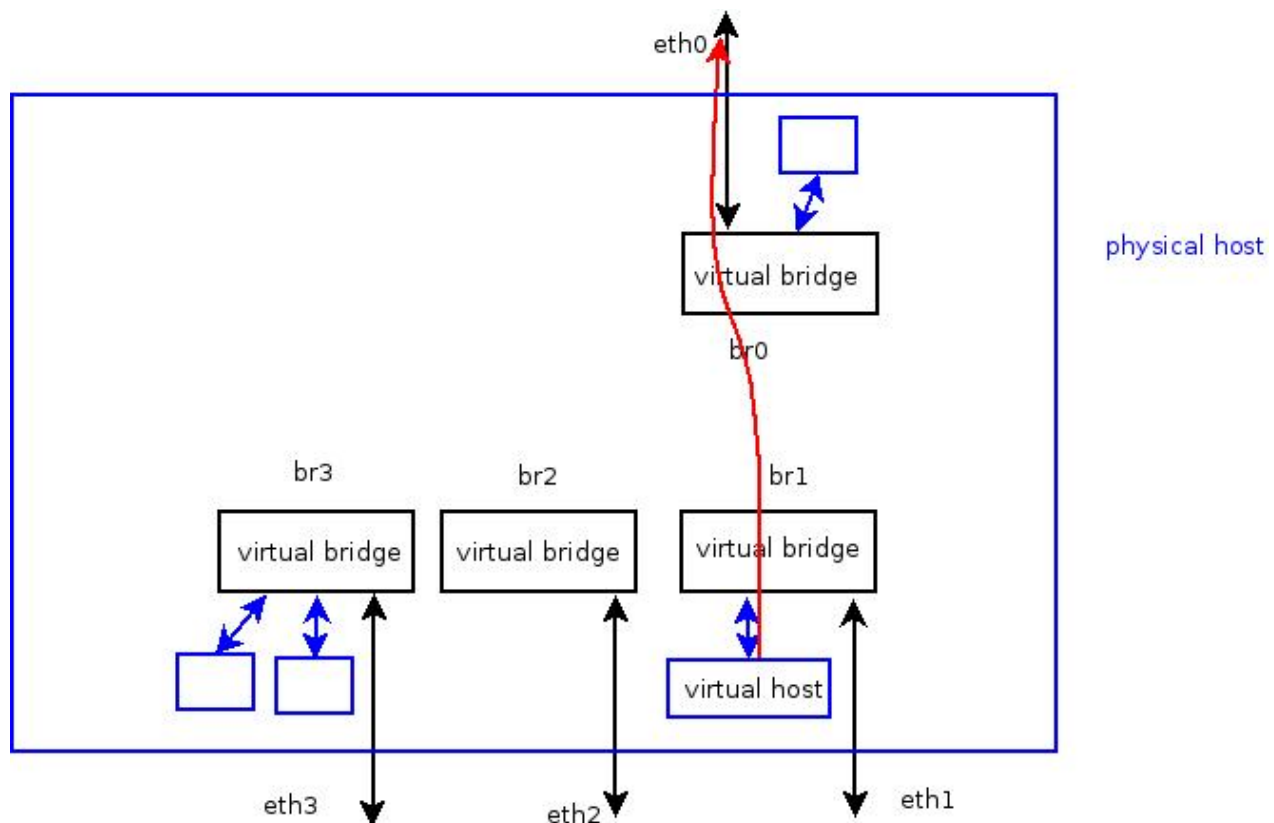


Illustration 26: Base Platform with Virtual Bridges

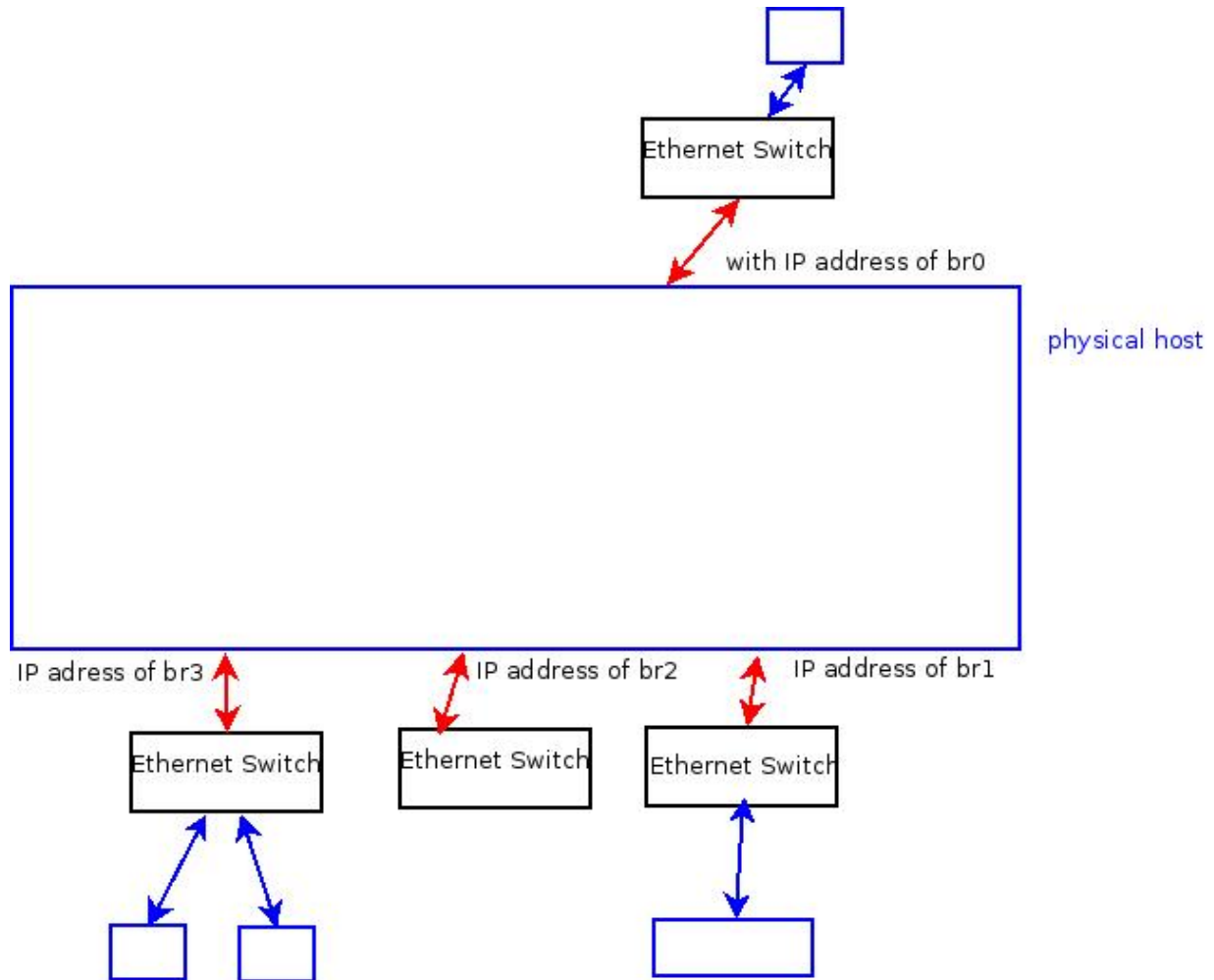


Illustration 27: The Equivalent Model by using physical Ethernet Switches

The two illustrations above are used to explain how to view those virtual bridges of base platform correctly. The virtual bridges can be considered as physical Ethernet switches outside the base platform and the base platform is using extra Ethernet interfaces connecting to the physical Ethernet switches;

and those extra Ethernet interfaces of base platform are with the IP addresses originally set on the virtual bridges.

The base platform is currently configured as follows by default: the network connecting to the boundary of “br0” is labeled as “**net**”, the network connecting to the boundary of “br2” is labeled as “**dmz**”, and the regions connecting to “br1”, “br3”, “br4”.., “br10”, and “br11” are labeled as “**loc**”.

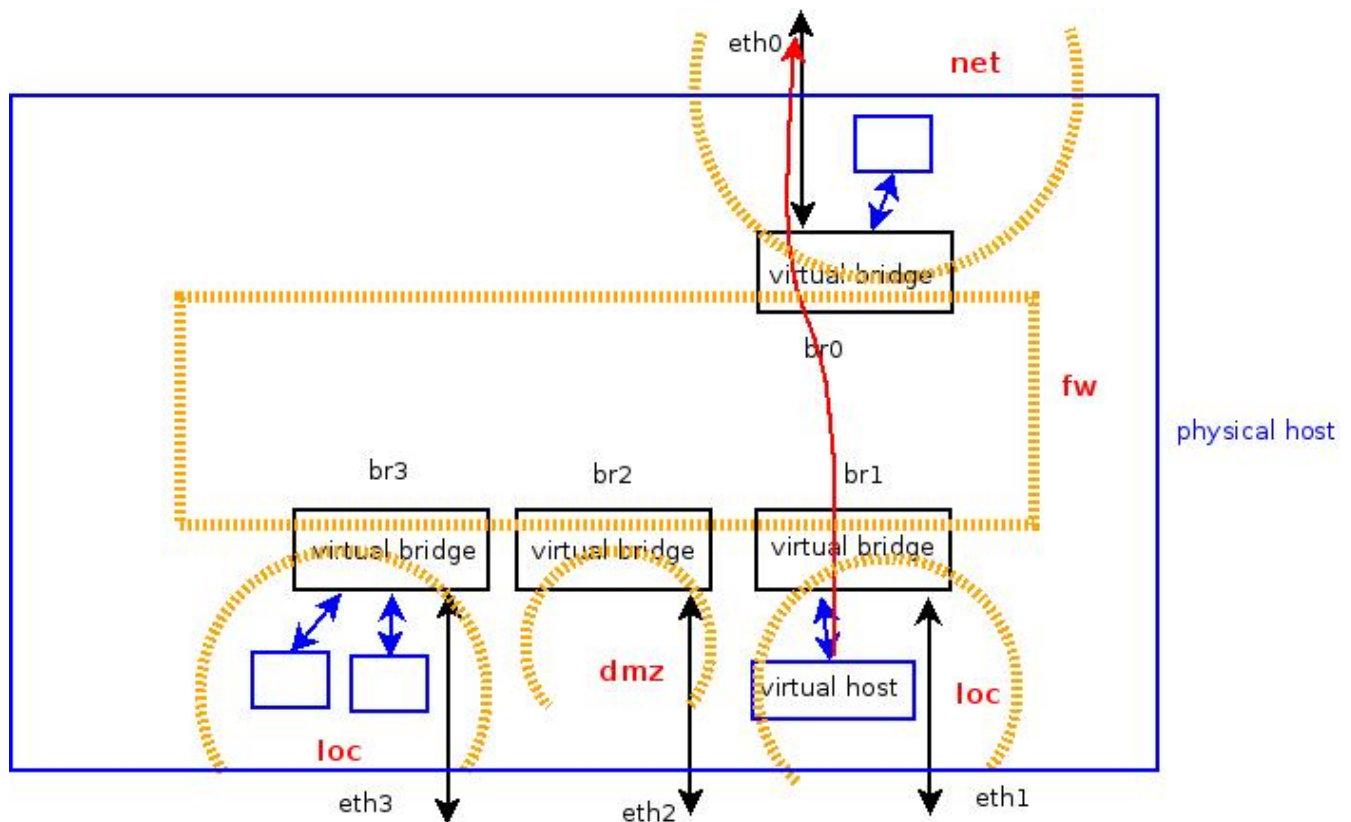


Illustration 28: Zone partitioning on the base platform

And when we create a virtual host by placing one of its Ethernet interface into a virtual bridge, that Ethernet interface is governed by the rules associated with that virtual bridge where the zone it resides. Therefore, it is necessary to be familiar with the definition of those zones and functions of the operations.

The network traffic initiated from “**loc**” or “**dmz**” is allowed to access the hosts in “**net**”.

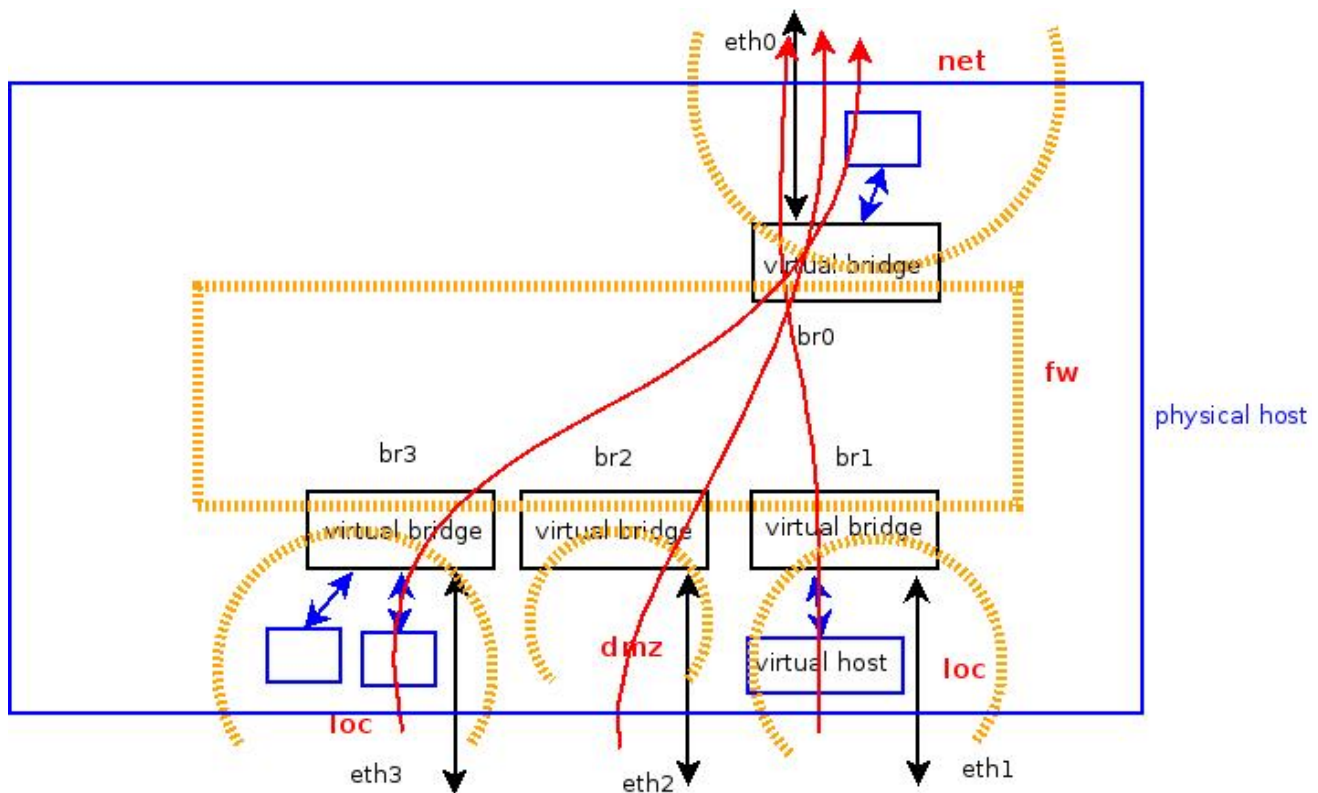


Illustration 29: Traffic with destination in “net”

And network traffic from “**loc**” or “**dmz**” to “**net**” will be under **NAT** (Network Address Translation) operation by replacing its source IP address in the original IP header by the **IP address carried by “br0”**, and the connection will be tracked by using a UDP or TCP port so that the other side of the party will reply the network traffic according to the replaced IP address and source port to the base platform, and then to the corresponding host in “**loc**” or “**dmz**”.

The traffic from zone “**net**” is forbidden to access “**loc**” or “**dmz**” by default.

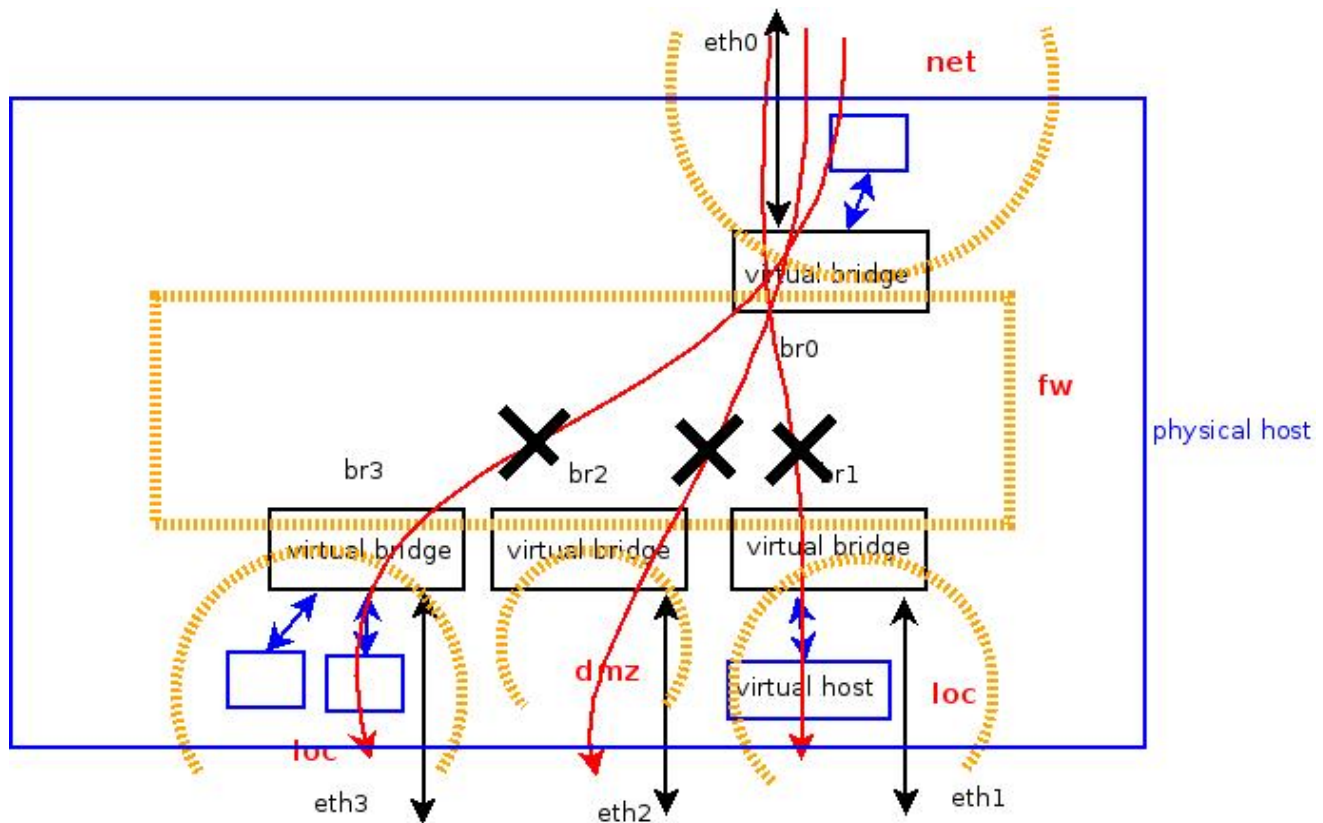


Illustration 30: Traffic originated from “net” is forbidden to access “loc” or “dmz”

Again, this setting can be changed by adding exception rules. Hosts in zone “**net**” can not access hosts in the zone “**loc**” or zone “**dmz**”. Thus, we usually let the zone “**net**” connecting to the Internet so that it is sometimes labeled as “**WAN**” (wide-area network). For network traffic originated from hosts in zone “**net**” to reach hosts in “**dmz**” or “**loc**”, the exception rule of “**port forwarding**” is needed. The operation “**port forwarding**” is to forward the network traffic arriving at base platform by selecting the destination UDP or TCP port and forward to a host in “**dmz**” or “**loc**”. We usually let “**dmz**” to take care of the connections from “**net**” and keep “**loc**” for hosts that only can be accessed locally.

Network traffic originated from zone **“loc”** is allowed to access the hosts in **“loc”** and **“dmz”**.

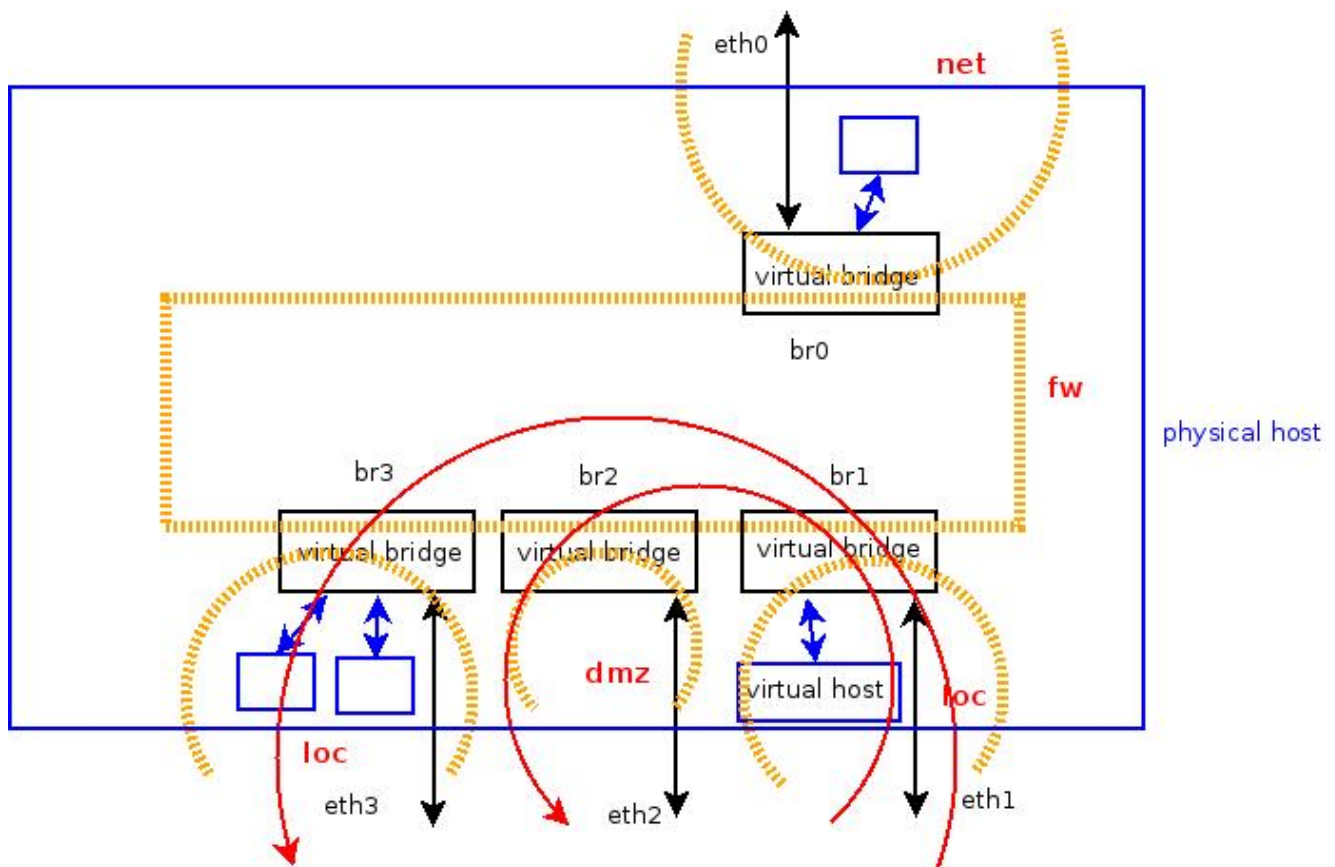


Illustration 31: Traffic originated from the zone “loc”

The hosts in **“loc”** can access the hosts in any zones; network traffic from **“loc”** to zone **“net”** will be under the operation NAT, but there is no NAT operation performed while going from **“loc”** to **“dmz”** or other subnets of **“loc”**. Network traffic going across different IP subnet need to do “routing” on the base platform. The operations will be handled by the base platform automatically once the network packets reach one of its bridges. But for the hosts sending network packets destined to the other subnets, it is necessary to set the gateway to those subnets properly, or just use the IP address of the base platform on the local subnet as default gateway.

Traffic originated from the zone “**dmz**” is forbidden to access the hosts in “**loc**”.

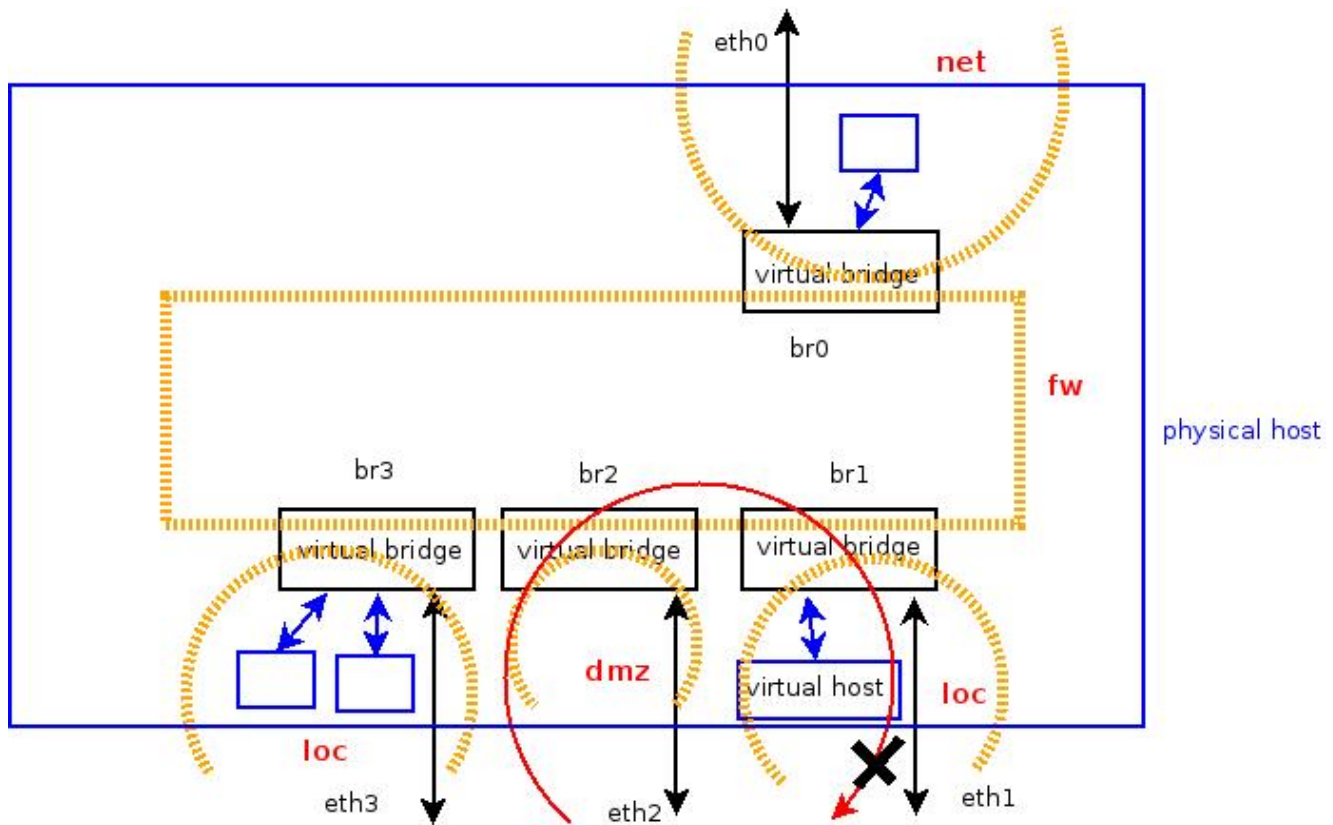


Illustration 32: Traffic originated from zone “dmz” is forbidden to access “loc”.

The hosts in “**dmz**” are forbidden to access the hosts in “**loc**”, but they can access the hosts in “**net**”. Due to this nature, we usually put Wifi AP for the guests visiting the office in this zone, or placing hosts to handle the connections from the Internet by adding exception rules of port forwarding.

We summarize those predefined rules as follows:

```
loc → net (OK)
loc → dmz (OK)
loc → loc (OK )
dmz → net (OK)
dmz → loc ( Forbidden )
net → loc (Forbidden)
net → dmz (Forbidden)
```

The base platform itself except those virtual hosts is labeled as zone “**fw**”. The predefined rules associated with “**fw**” are

```
fw → net (OK)
fw → loc (OK)
fw → dmz (OK)
net → fw (Forbidden)
loc → fw (OK)
dmz → fw (Forbidden)
```

Those predefined rules will not be shown on the web management interface. However, you can **add rules for exceptions** and modify the components of the zone. There is another zone “**road**” that is associated with VPN and other internal interfaces used by the base platform. Those items will be introduced once we mention those functionalities in the following sections.

Port Forwarding

The term “**Port Forwarding**” comes with NAT (Network Address Translation). As we mentioned earlier the network traffic initiated from the zone “**dmz**” or “**loc**” to the zone “**net**” will be replaced with the IP address of “br0” as the source IP address along with a new TCP or UDP port to track the original IP address in zone “**dmz**” or “**loc**”; the response traffic will be using this source IP address and port as destination and the base platform forward the response to the original host in “**dmz**” or “**loc**”. This is for the cases that network traffic is originated from zone “**dmz**” or “**loc**” to “**net**”.

However, for the traffic originated from zone “**net**” arriving at the base platform, the traffic will be simply dropped unless the base platform has a TCP/UDP port open with specific daemon to handle the traffic or it has rules matched this type of traffic and forward to the other host. “Port forwarding” is the process to specify a TCP or UDP port and forward the traffic destined for the base platform with the associated port to the other host in zone “**dmz**” or “**loc**”. The port forwarding rule can be specified at “**Border >> Connection >> Port Forwarding**”:

Port Forwarding Setting ☐ Stop Border Engine

Border >> Connection >> Port Forwarding

Port Forwarding

☒ Target IP Address to forward Https or Http Traffic:

☐ Others

Port Number: Protocol:

Forwarding Target IP Address :

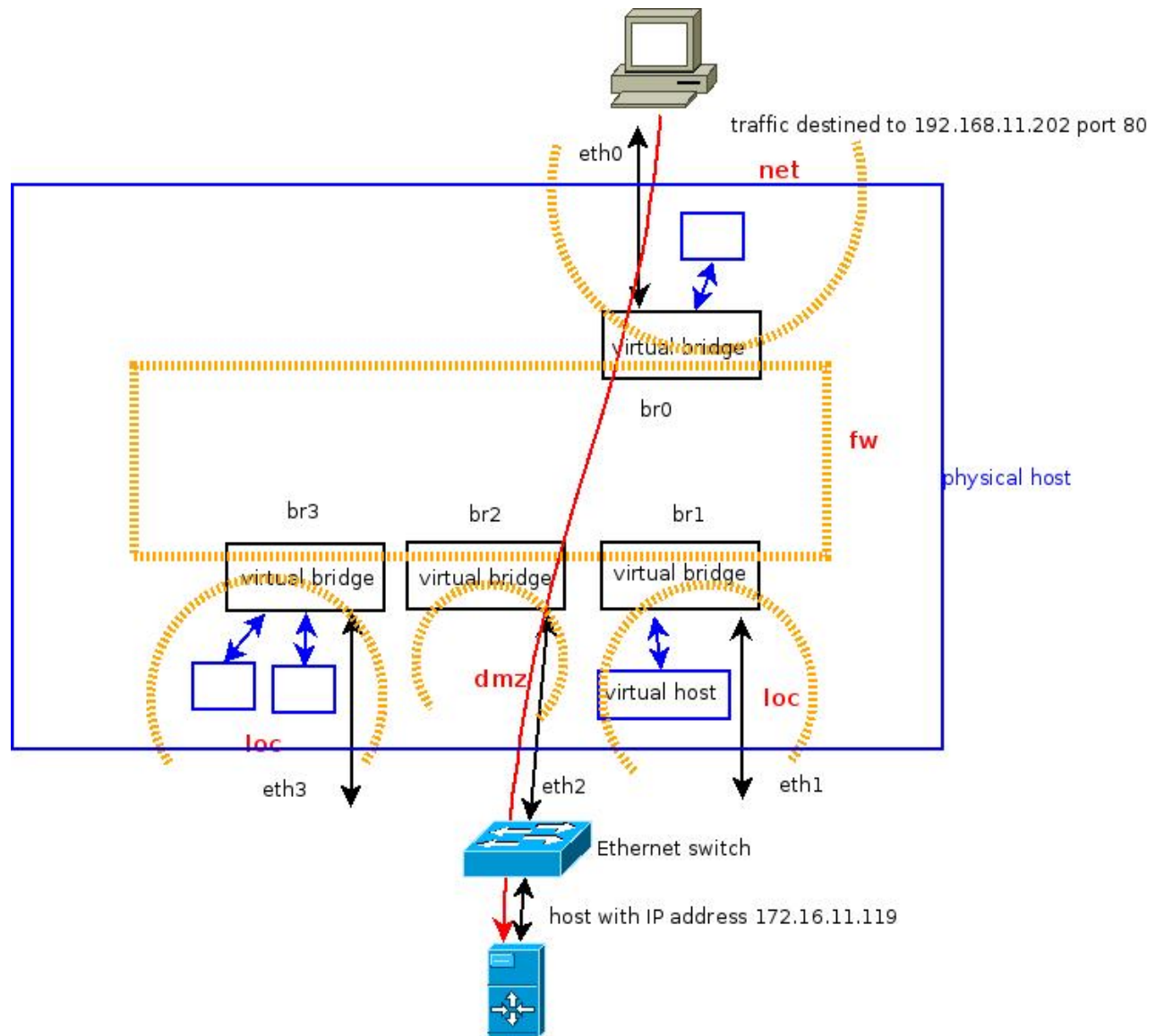
Remove

Servers Behind the Border:

----- None in the list -----

Illustration 33: Screen Snapshot for Port Forwarding

Illustration 34: Port Forwarding Example



To enter the rule to forward http and https traffic, just select “**dmz**” and enter IP address “172.16.11.119” as indicated in the following screen snapshot:

Port Forwarding Setting ☐ **Stop Border Engine** **Set**

Border >> Connection >> Port Forwarding

Port Forwarding

☒ Target IP Address to forward Https or Http Traffic:
dmz 172.16.11.119

☐ Others

Port Number: Protocol : TCP

Forwarding Target IP Address :
loc

Submit

Remove
Servers Behind the Border:
----- None in the list -----

Remove

Illustration 35: Example for HTTP Port Forwarding

And press “Submit” button, the box on the right will display that the traffic associated with TCP port 80 and port 443 will be forwarded to 172.16.11.119 in zone “dmz”.

Port Forwarding Setting ☐ **Stop Border Engine** **Set**

Border >> Connection >> Port Forwarding

Port Forwarding

☒ Target IP Address to forward Https or Http Traffic:

☐ Others

Port Number: Protocol:

Forwarding Target IP Address :

Submit

Remove

Servers Behind the Border:

- >dmz:172.16.11.119:tcp:80
- >dmz:172.16.11.119:tcp:443

Remove

Illustration 36: Screen Snapshot for Setting HTTP Port Forwarding

To make the rules in effect, you have to stop the border engine and restart it again. It can be done by **checking** the box on the top and press “Set”; and then **un-check** the box and press “Set” again. After making any rule changes, it is necessary to restart the border engine to make them in effect.

Port Forwarding Setting ☐ **Stop Border Engine** **Set**

Border >> Connection >> Port Forwarding

Port Forwarding

☐ Target IP Address to forward Https or Http Traffic:
loc

☒ Others

Port Number: Protocol:

Forwarding Target IP Address :
dmz

Submit

Remove
Servers Behind the Border:

```
-->dmz:172.16.11.119:tcp:80
-->dmz:172.16.11.119:tcp:443
```

Remove

Illustration 37: Example of SMTP for Port Forwarding

Similarly, TCP port 25 is used by SMTP. To forward SMTP traffic to the host "172.16.11.119", just enter as above and press "Submit". The display box on the right will be shown that traffic associated TCP port 25 will be forwarded to the host "172.16.11.119" in zone "**dmz**". While using port forwarding, the default gateway of host receiving forwarded traffic should be set to the interface of base platform (in this case, the IP address of "br1"). Otherwise, the reply packets can not go back to the sender.

Port Forwarding Setting ☐ Stop Border Engine Set

Border >> Connection >> Port Forwarding

Port Forwarding

☒ Target IP Address to forward Https or Http Traffic:

loc

☐ Others

Port Number: Protocol: TCP

Forwarding Target IP Address:

loc Submit

Remove

Servers Behind the Border:

```
-->dmz:172.16.11.119:tcp:80
-->dmz:172.16.11.119:tcp:443
-->dmz:172.16.11.119:tcp:25
```

Remove

Illustration 38: Screen Snapshot after Adding SMTP Port Forwarding

As a matter of fact, “Port Forwarding” is using the action “DNAT” to implement on the base platform. “DNAT” is an action by changing the destination IP address on the packets arriving at the base platform and deliver them according to the new destination IP address. Thus, by adding rules here, you will find out the corresponding rules using “DNAT” are shown in **“Border >> Rule >> List / Remove Rule”**.

In general, we encourage to use “Port forwarding” to the hosts in zone **“dmz”** instead of zone **“loc”**. Using “Port forwarding” to the hosts in zone **“loc”** might cause the spam from the Internet to be populated into local area network. Along with “Port forwarding”, sometimes people might ask for looping back the traffic initiated from the same zone of the targeted host of Port Forwarding.

For example, if you do port forwarding to a host in zone “loc”, people use the public IP address (the IP address of “br0” of the base platform) from zone “net” can reach that host. However, people in the zone “loc” might also hope to use that public IP address to reach that host. In this case, it needs “Loopback”. **We do not provide “Loopback” on the base platform.** If you are interested on that feature, please refer to our “**ved**” build.

Connection Tracking

Checking the connection activities might be helpful to do diagnosis on the network issues. It can be done via “**Border >> Connection >> Connection Tracking**” and click “Display” button.

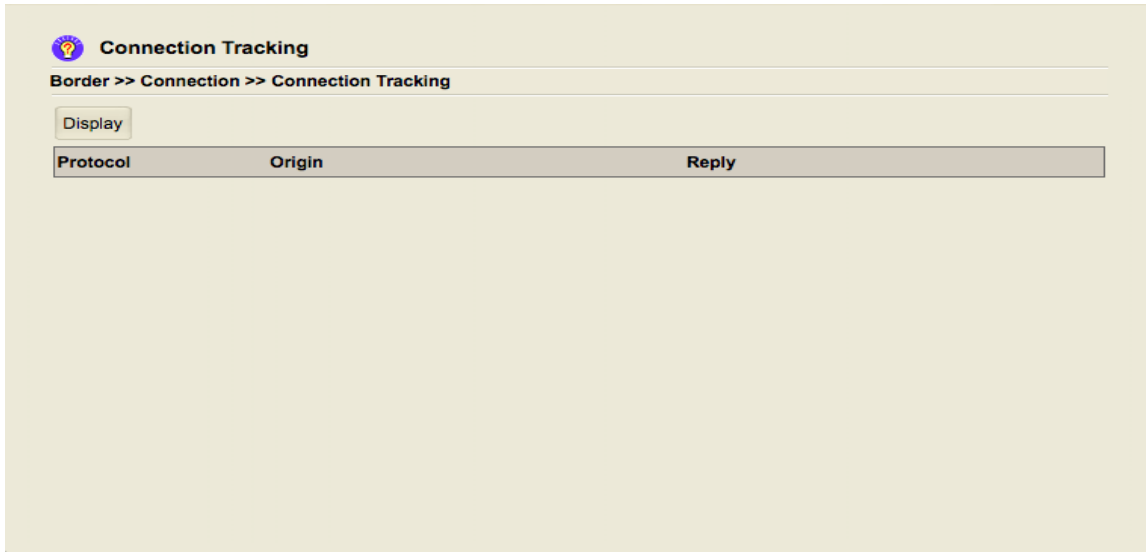



Illustration 39: Screen Snapshot for Connection Tracking

 **Connection Tracking**

Border >> Connection >> Connection Tracking

Display

Protocol	Origin	Reply
tcp 6 26 TIME_WAIT	src=192.168.11.197 dst=192.168.11.202 sport=51889 dport=8082	src=192.168.11.202 dst=192.168.11.197 sport=8082 dport=51889 [ASSURED] mark=0 secctx=null use=1
tcp 6 431999 ESTABLISHED	src=192.168.11.197 dst=192.168.11.202 sport=51893 dport=8082	src=192.168.11.202 dst=192.168.11.197 sport=8082 dport=51893 [ASSURED] mark=0 secctx=null use=1
tcp 6 95 TIME_WAIT	src=192.168.11.197 dst=192.168.11.202 sport=51892 dport=8082	src=192.168.11.202 dst=192.168.11.197 sport=8082 dport=51892 [ASSURED] mark=0 secctx=null use=1
tcp 6 25 TIME_WAIT	src=192.168.11.197 dst=192.168.11.202 sport=51890 dport=8082	src=192.168.11.202 dst=192.168.11.197 sport=8082 dport=51890 [ASSURED] mark=0 secctx=null use=1

Illustration 40: Display for Connection Status

If you find certain personal computer is with tremendous amounts of connections, it might imply that computer is with virus or using some peer-to-peer software. You might go ahead to identify the root cause by the clues provided here.

Actions after Receiving Network Packets

After receiving a network packet, the base platform has the following options to act on this packet: **ACCEPT**, **DROP**, **REJECT**, **DNAT**, and **REDIRECT**. The attributes associated with each rule operating on the IP traffic shall have: where the packets come from (**Source**), where the packets should go (**Destination**), whether the protocol is TCP or UDP, **destination port**, **source port**, and **original destination IP address**. The actions are performed according to some of those attributes by the base platform. We roughly introduce those actions here, and the usage of those actions will be shown in the following sections.

The action “**ACCEPT**” is to accept the network traffic matched all the attributes of the rule. For example, to allow all the “telnet connections”(TCP port 23) from zone “**net**” to the base platform “**fw**”, the action “**ACCEPT**” can be used by specifying

Source: net
Destination: fw
Protocol: TCP
Destination Port: 23

The action “**DROP**” is simply ignoring the arriving packets of the specified attributes. The difference between “**DROP**” and “**REJECT**” is that “**REJECT**” would send some messages back to the sender that the connection is not reachable in some cases. Please note that we have those predefined rules between zones by default. Thus, we only need to specify “**DROP**” and “**REJECT**” if necessary. For example, traffic originated from zone “**loc**” to zone “**net**” is allowed by default. If we would like to block all the http access from “loc” to “net”, then we can do “**DROP**” by specifying

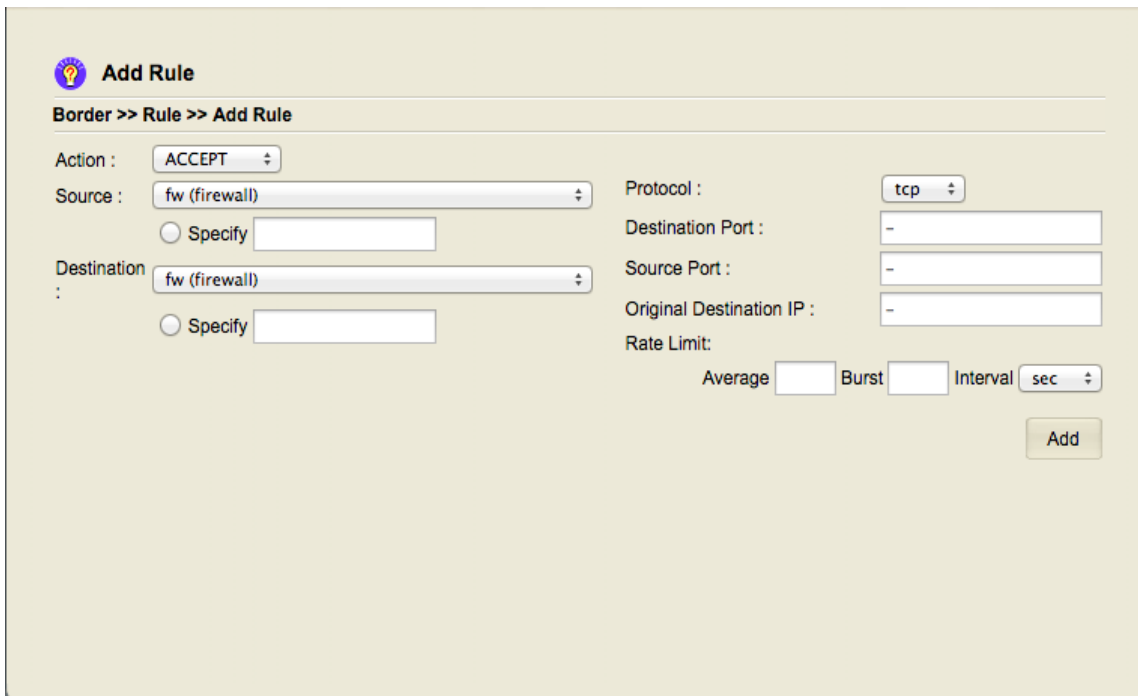
Source: loc
Destination: net
Protocol: TCP
Destination Port: 80

The action “**REDIRECT**” is to redirect the traffic arriving at the base platform to **another port of the base platform**. We usually do this by trying not to reconfigure the application daemon but we would like to have that daemon listen on more port. For example, “telnet traffic” is usually using “TCP port 23”. But we would like to have “TCP port 28” also handled by the “telnet daemon”. In this case, we can use “**REDIRECT**”.

And as we mentioned earlier, the action “**DNAT**” is to do “Port forwarding” by changing the destination IP address of the arrival packets from zone “**net**” to a host in zone “**dmz**” or zone “**loc**”. This host does not need to have a physical entity; it can be a virtual host as long as it can handle the forwarded traffic.

Add Rule

Adding Exception Rules can be done via “**Border >> Rule >> Add Rule**”.



The screenshot shows the 'Add Rule' configuration page. At the top, there is a breadcrumb trail: 'Border >> Rule >> Add Rule'. The page contains several configuration fields:

- Action:** A dropdown menu set to 'ACCEPT'.
- Source:** A dropdown menu set to 'fw (firewall)'. Below it is a radio button labeled 'Specify' followed by an empty text input field.
- Destination:** A dropdown menu set to 'fw (firewall)'. Below it is a radio button labeled 'Specify' followed by an empty text input field.
- Protocol:** A dropdown menu set to 'tcp'.
- Destination Port:** A text input field containing a hyphen (-).
- Source Port:** A text input field containing a hyphen (-).
- Original Destination IP:** A text input field containing a hyphen (-).
- Rate Limit:** A section with three input fields: 'Average', 'Burst', and 'Interval'. The 'Interval' field has a unit dropdown set to 'sec'.

An 'Add' button is located at the bottom right of the form.

Illustration 41: Screen Snapshot for Adding Rule

Before adding a rule, please make sure where your host (no matter if it is virtual host or host with physical entity) is situated. If it is a virtual host, you need to make sure it has a network interface joining that bridge; for a host with physical entity, check if it is properly connected. For hosts situated in zone “dmz”, using DHCP to get IP address from base platform is not working. The reason is that we have predefined rule to block access from zone “dmz” to zone “fw”. Thus, DHCP packets will be rejected by the base platform. You have to **set IP address manually for the hosts in zone “DMZ”**.

Add Rule

Border >> Rule >> Add Rule

Action :

Source : Protocol :

☐ Specify

Destination : ☒ fw (firewall)
☐ loc (br1 br3 br4 br5 br6 br7 br8 br9 br10 br11)
☐ net (br0)
☐ dmz (br2)

Destination Port :

Source Port :

Original Destination IP :

Rate Limit: Average Burst Interval

Illustration 42: Source and Destination Associated with Rule

Those predefined rules will not be displayed, but you should be familiar with them while adding exception rules. It is of no use to add new rule by allowing connections from zone “**loc**” to “**dmz**” because the base platform already admitted the rule by default. However, it makes sense to add exception rules by allowing connections from zone “**dmz**” to zone “**loc**” because the connections are blocked by default.

Similarly, it makes sense to add exceptions from zone “**net**” to zone “**fw**” because the connections are blocked by default as well. For example,

Source: net
 Destination: fw
 Protocol: tcp
 Destination Port: 23

And press “Add” button. The rule will be in effect after restarting the border control engine. It allows “telnet connections” (TCP port 23) from zone “net” to zone “fw”.

In the following sections, we are going to provide some examples for reference.

Allowing Exceptions for TCP Connections from dmz to loc

The connections from zone “**dmz**” to zone “**loc**” are blocked by default. However, we would like to add rule for allowing TCP connections from **dmz** to a host with IP address “172.16.9.12” in zone “**loc**”.

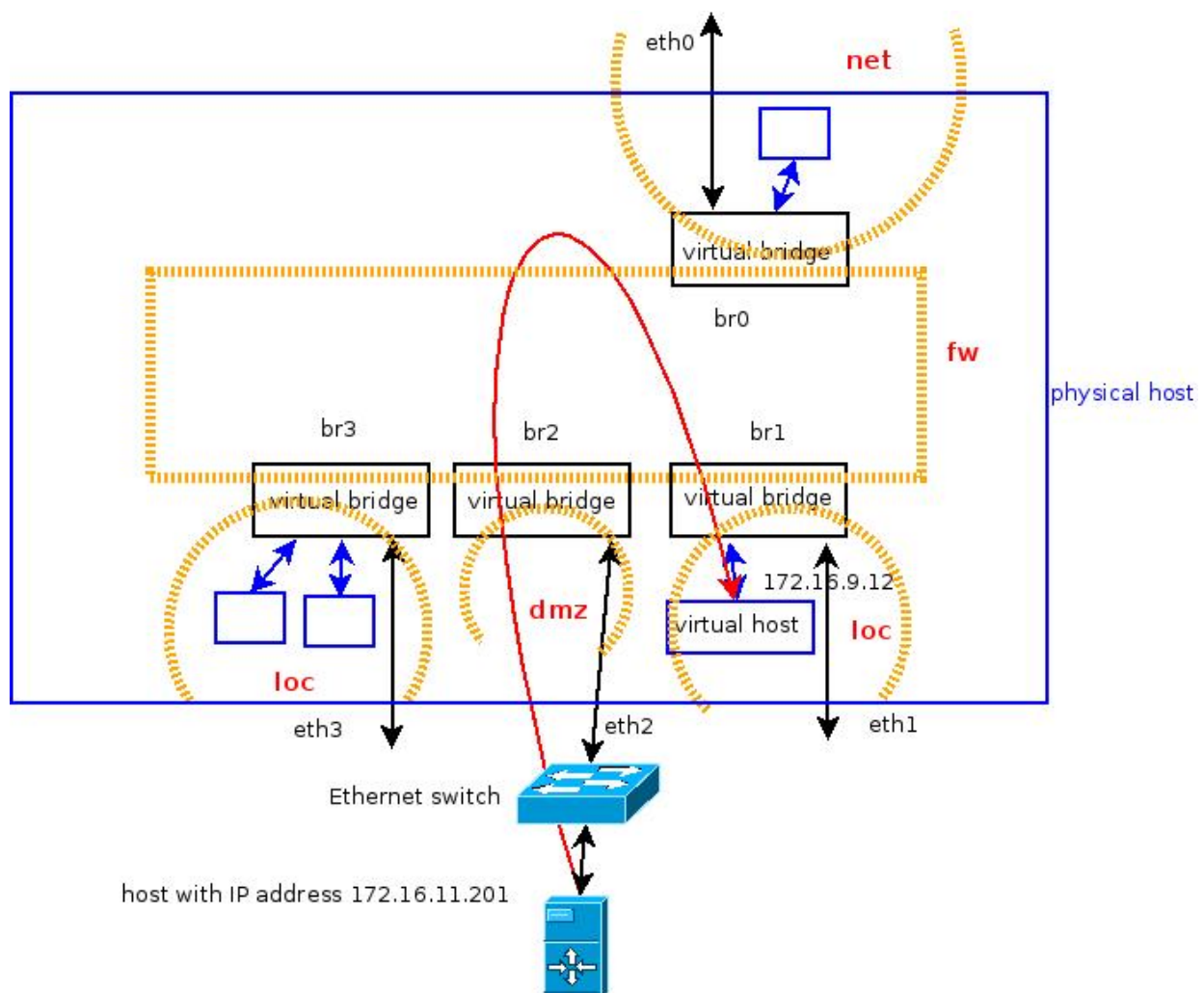


Illustration 43: Adding Exception Rule from Zone "dmz" to "loc"

Please check the following screen snapshot while specifying specific IP address "172.16.9.12" of a host in zone "loc":

Add Rule

Border >> Rule >> Add Rule

Action :

Source :

☐ Specify

Destination :

☒ Specify

Protocol :

Destination Port :

Source Port :

Original Destination IP :

Rate Limit:

Average Burst Interval

Illustration 44: Screen Snapshot for "dmz" to a Host in "loc"

To specify a specific host, it is necessary to use the following format in the box next to "Specify":

"zone:IP_ADDRESS"

or

"zone:SUBNET"

For example,

"loc:172.16.9.12"

or

"loc: 172.16.12.0/24"

After pressing “Add” button, the associated rule can be viewed via **“Border >> Rule >> List / Remove Rule”**:

























Border >> Rule >> List / Remove Rule								
Current Rules								
Action	Source	Destination	Protocol	Destination Port	Source Port	Original Destination IP	Rate Limit	Remove
?SECTION	NEW							
ACCEPT	fw	net	udp	53				
ACCEPT	loc	fw	tcp	22				
ACCEPT	fw	loc	udp	137:139				
ACCEPT	fw	loc	tcp	137,139				
ACCEPT	fw	loc	udp	1024:	137			
ACCEPT	loc	fw	udp	137:139				
ACCEPT	loc	fw	tcp	137,139				
ACCEPT	loc	fw	udp	1024:	137			
ACCEPT	loc	fw	tcp	ssh				
ACCEPT	net	fw	tcp	http				
ACCEPT	net	fw	tcp	443				
ACCEPT	net	fw	tcp	8080				
ACCEPT	net	fw	tcp	8081				
ACCEPT	net	fw	tcp	8082				
ACCEPT	net	fw	udp	1194				
ACCEPT	net	fw	udp	1195				
ACCEPT	net	fw	udp	7777				
ACCEPT	net	fw	udp	4569				
ACCEPT	net	fw	tcp	25				
ACCEPT	net	fw	tcp	23	-	-		
ACCEPT	net	fw	tcp	22	-	-		
ACCEPT	net	fw	tcp	5901-5909	-	-		
ACCEPT	dmz	loc:172.16.9.12	tcp	-	-	-		

Illustration 45: Display the Rule from "dmz" to a Host in "loc"


Reject or Drop Connections

The connections from zone “**net**” to zone “**loc**” are forbidden by default. Thus, it is not necessary to add redundant rules by reject or drop connections from “**net**” to “**loc**”. It is the similar case for the connections from zone “**dmz**” to “**loc**”. The connections from zone “**loc**” to zone “net” are allowed by default. If we want to add exception rule to block http traffic from “**loc**” to “net”, we can do

Source: loc
Destination: net
Protocol: TCP
Destination Port: 80

Source: loc
Destination: net
Protocol: TCP
Destination Port: 443

HTTP and HTTPS are using TCP port 80 and TCP port 443. Just enter them as follows by pressing “Add” button. The rules will take effect after restarting the “Border Engine”:

 **Add Rule**

Border >> Rule >> Add Rule

Action : DROP

Source : loc (br1 br3 br4 br5 br6 br7 br8 br9 br10 br11)
☐ Specify

Destination : net (br0)
☐ Specify

Protocol : tcp

Destination Port : 80

Source Port : -

Original Destination IP : -

Rate Limit:
Average Burst Interval sec

Add

Illustration 46: Screen Snapshot for Dropping Http Traffic from "loc" to "net"

Add Rule

Border >> Rule >> Add Rule

Action : DROP

Source : loc (br1 br3 br4 br5 br6 br7 br8 br9 br10 br11)

☐ Specify

Destination : net (br0)

☐ Specify

Protocol : tcp

Destination Port : 443

Source Port : -

Original Destination IP : -

Rate Limit:

Average Burst Interval sec

Add

Illustration 47: Screen Snapshot form Dropping HTTPS traffic from "loc" to "net"










ACCEPT	net	fw	udp	7777				
ACCEPT	net	fw	udp	4569				
ACCEPT	net	fw	tcp	25				
ACCEPT	net	fw	tcp	23	-	-		
ACCEPT	net	fw	tcp	22	-	-		
ACCEPT	net	fw	tcp	5901-5909	-	-		
ACCEPT	dmz	loc:172.16.9.12	tcp	-	-	-		
DROP	loc	net	tcp	80	-	-		
DROP	loc	net	tcp	443	-	-		

Illustration 48: Example for Dropping HTTP and HTTPS (Ports 80 and 443)

Redirect Traffic to Another Port of the Base Platform

To redirect the network traffic from one TCP/UDP port to another, it usually involves some network environment issues or legacy systems. For example, you are under the environment of blocking TELNET connections (listening on TCP port 23) but allowing some other connections. So, you would like to use another port for TELNET. In this case, you might set like

Action: REDIRECT
Source: net
Destination: 23
Protocol: TCP
Destination Port: 29

This redirects the traffic of TCP port 29 to TCP port 23:

Add Rule

Border >> Rule >> Add Rule

Action :	REDIRECT	Protocol :	tcp
Source :	net (br0)	Destination Port :	29
	<input type="radio"/> Specify	Source Port :	-
Destination :	fw (firewall)	Original Destination IP :	-
	<input checked="" type="radio"/> Specify 23	Rate Limit:	
		Average	Burst Interval sec

Add

Illustration 49: Redirect Traffic to a Different Port























Action	Source	Destination	Protocol	Destination Port	Source Port	Original Destination IP	Rate Limit	Remove
?SECTION	NEW							
ACCEPT	fw	net	udp	53				
ACCEPT	loc	fw	tcp	22				
ACCEPT	fw	loc	udp	137:139				
ACCEPT	fw	loc	tcp	137,139				
ACCEPT	fw	loc	udp	1024:	137			
ACCEPT	loc	fw	udp	137:139				
ACCEPT	loc	fw	tcp	137,139				
ACCEPT	loc	fw	udp	1024:	137			
ACCEPT	loc	fw	tcp	ssh				
ACCEPT	net	fw	tcp	http				
ACCEPT	net	fw	tcp	443				
ACCEPT	net	fw	tcp	8080				
ACCEPT	net	fw	tcp	8081				
ACCEPT	net	fw	tcp	8082				
ACCEPT	net	fw	udp	1194				
ACCEPT	net	fw	udp	1195				
ACCEPT	net	fw	udp	7777				
ACCEPT	net	fw	udp	4569				
ACCEPT	net	fw	tcp	25				
ACCEPT	net	fw	tcp	5901-5909	-	-		
REDIRECT	net	23	tcp	29	-	-		

Illustration 50: REDIRECT rule in display list

And it takes effect after restarting border engine. Although the network packets can be redirected from one TCP port to another, the network protocol in higher layer might be broken by this action. For example, if you use HTTP to access an HTML page from non-standard port, the result might fail to display on your browser completely because the HTML might contain some hardcoded local URL so that they can not be displayed completely while using the other ports.

List or Delete Rule

























 List / Remove Rules								
Border >> Rule >> List / Remove Rule								
Current Rules								
Action	Source	Destination	Protocol	Destination Port	Source Port	Original Destination IP	Rate Limit	Remove
?SECTION	NEW							
ACCEPT	fw	net	udp	53				
ACCEPT	loc	fw	tcp	22				
ACCEPT	fw	loc	udp	137:139				
ACCEPT	fw	loc	tcp	137,139				
ACCEPT	fw	loc	udp	1024:	137			
ACCEPT	loc	fw	udp	137:139				
ACCEPT	loc	fw	tcp	137,139				
ACCEPT	loc	fw	udp	1024:	137			
ACCEPT	loc	fw	tcp	ssh				
ACCEPT	net	fw	tcp	http				
ACCEPT	net	fw	tcp	443				
ACCEPT	net	fw	tcp	8080				
ACCEPT	net	fw	tcp	8081				
ACCEPT	net	fw	tcp	8082				
ACCEPT	net	fw	udp	1194				
ACCEPT	net	fw	udp	1195				
ACCEPT	net	fw	udp	7777				
ACCEPT	net	fw	udp	4569				
ACCEPT	net	fw	tcp	25				
ACCEPT	net	fw	tcp	23	-	-		
ACCEPT	net	fw	tcp	22	-	-		
ACCEPT	net	fw	tcp	5901-5909	-	-		

Illustration 51: Delete the rule from List

We have seen this in the earlier sections. To delete the rule, just press the trash can icon on the right of the rule. And the system will ask you to confirm while removing the rule. Please remember to restart the border engine to take effective.

Using DNAT for Port Forwarding

As we mentioned earlier, port forwarding is done by using “**DNAT**”. The action “**DNAT**” is to change the destination IP address of the received packets on the base platform and forward the packets according to the modified destination IP address.

For example, to forward HTTP traffic (TCP port 80) to a host with IP address “172.16.11.201” in zone “**dmz**”, we can do

Action: DNAT
Source: net
Destination: dmz:172.16.11.201
Protocol: tcp
Destination Port: 80

Add Rule

Border >> Rule >> Add Rule

Action :

Source : ☐ Specify

Destination : ☒ Specify

Protocol :

Destination Port :

Source Port :

Original Destination IP :

Rate Limit: Average Burst Interval

Illustration 52: Use DNAT for Port Forwarding

This will forward the arrival traffic of “TCP port 80” on the base platform to “TCP port 80” of the host receiving forwarded traffic. But there is a chance that we would like to forward the traffic to a different port of that host to process data.

For example, the traffic of “TCP Port 2929” will be forwarded to “TCP port 80” of the host “172.16.11.201” in zone “**dmz**”. It can be done by setting

Action: DNAT
Source: net
Destination: **dmz:172.16.11.201:80**
Protocol: tcp
Destination Port: 2929

Add Rule

Border >> Rule >> Add Rule

Action :

Source :

☐ Specify

Destination :

☒ Specify

Protocol :

Destination Port :

Source Port :

Original Destination IP :

Rate Limit:

Average Burst Interval

Illustration 53: Port Forwarding to a Host with different port

After pressing “Add” button, the rule will be displayed as follows:










ACCEPT	net	fw	udp	1195				
ACCEPT	net	fw	udp	7777				
ACCEPT	net	fw	udp	4569				
ACCEPT	net	fw	tcp	25				
ACCEPT	net	fw	tcp	5901-5909	-	-		
ACCEPT	net	fw	tcp	22	-	-		
ACCEPT	net	fw	tcp	23	-	-		
DNAT	net	dmz:172.16.11.2 01	tcp	80	-	-		
DNAT	net	dmz:172.16.11.2 01:80	tcp	2929	-	-		

Illustration 54: List Rules for Port Forwarding

Assume the IP address of “br0” of base platform is “192.168.11.202”. Thus, you can use “<http://192.168.11.202:2929/>” to reach “TCP port 80” of the host “172.16.11.202” in zone “dmz”.

IP Load Balance

For each tuple (**IP_ADDRESS, TCP_OR_UDP, PORT_NUMBER**), doing “Port forwarding” can only send the traffic to one host. Given an IP address, if you want to forward the traffic of a specific port to multiple hosts, it is called “**Load Balance**”. It can be done via the setting in “**Border >> Rule >> IP Load Balance**”.

The screenshot shows the "Distribute IP Service Requests to other host(s)" configuration window. The breadcrumb path is "Border >> Rule >> IP Load Balance". The interface is divided into two main sections: "Create/Delete Service Item" and "Add/Delete Server(s) within Service Item".

Create/Delete Service Item

- Service IP Address: [Text Input]
- Protocol: [Dropdown Menu, currently showing TCP]
- Port Number: [Text Input]
- Buttons: [Add] [Delete]

Add/Delete Server(s) within Service Item

- Service IP Address: [Text Input]
- Protocol: [Dropdown Menu, currently showing TCP]
- Port Number: [Text Input]
- Real Server IP Address: [Text Input]
- Buttons: [Add] [Delete]

Below these sections is a large list box containing a single entry: "-----none-----".

Illustration 55: Screen Snapshot for IP Load Balance

For each tuple (**IP_ADDRESS, TCP_OR_UDP, PORT_NUMBER**), we call it a “**service item**”. For each service item, there can be multiple hosts associated with it and network requests are distributed to those hosts in a **round-robin fashion**. Once a network request is sent to one of the hosts, the requests from the same client will be sent to the same host within 300 seconds.

The function provided by the base platform here is only to distribute the load to multiple servers. For some application-specific data, the designer has to keep data in sync among those hosts. The base platform can not deal with the data inside those hosts.

The following is an example for load balance associated with HTTP traffic. The IP address of "br0" of the base platform is "192.168.11.202". We would like to distribute the HTTP traffic arriving at the base platform to the hosts "172.16.11.201" and "172.16.11.202" in zone "dmz". Initially, you have to make sure that HTTP traffic (TCP Port 80) can be accept by the base platform by adding exception rule from zone "net" to zone "fw" for TCP port 80. Then, we can deal with the load balance here.

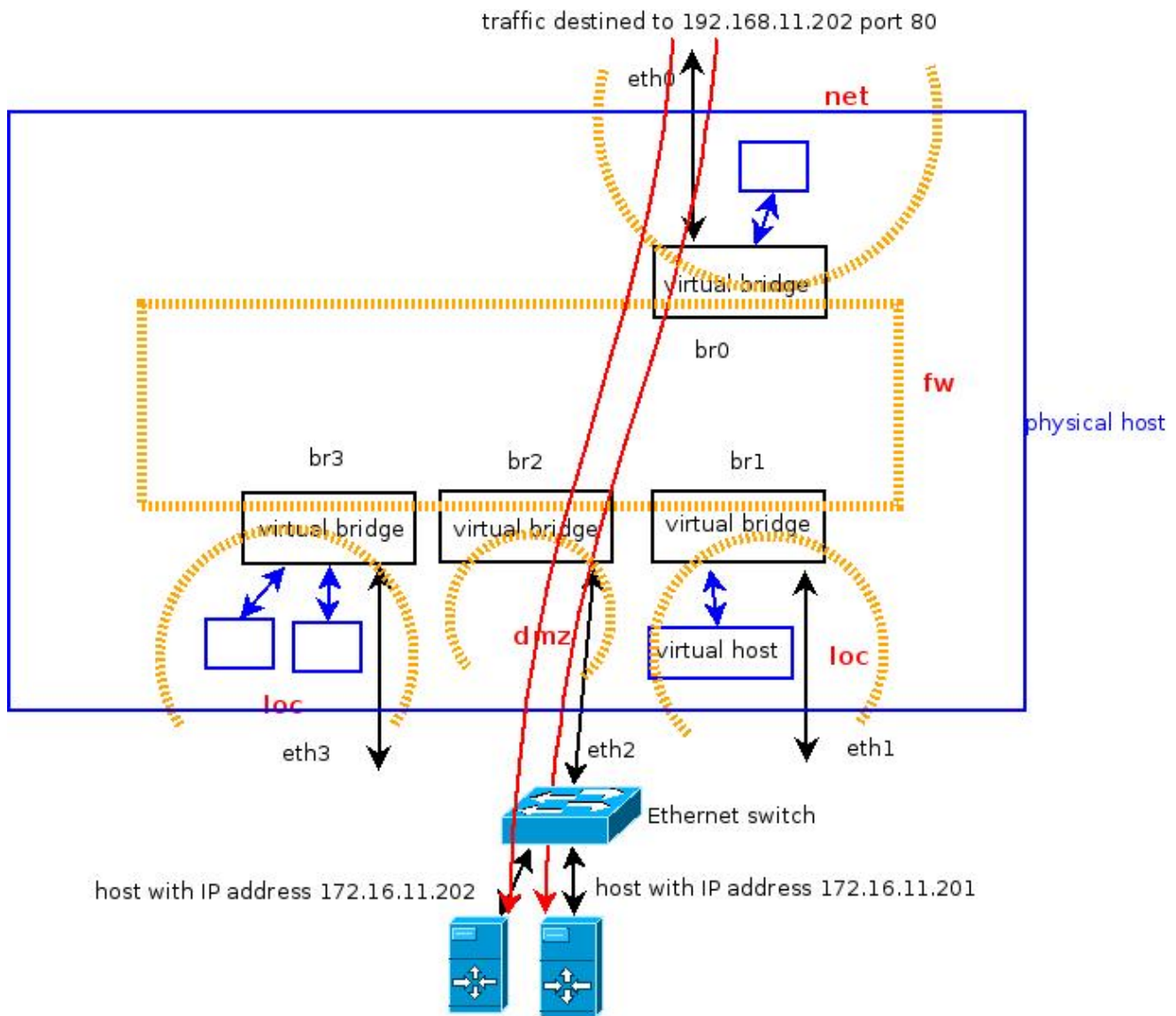



Illustration 56: Distribute HTTP Traffic to 2 Hosts

The setup procedures is as follows: using “192.168.11.202” along with TCP port 80 to create an service item.

 **Distribute IP Service Requests to other host(s)**

Border >> Rule >> IP Load Balance

Create/Delete Service Item

Service IP Address:

Protocol:

Port Number:

Add/Delete Server(s) within Service Item

Service IP Address:

Protocol: Port Number:

Real Server IP Address:

-----none-----

Illustration 57: Create a Service Item for Load Balance

And then, press “ADD” to add this service item.

Distribute IP Service Requests to other host(s)

Border >> Rule >> IP Load Balance

Create/Delete Service Item

Service IP Address:

Protocol:

Port Number:

Add/Delete Server(s) within Service Item

Service IP Address:

Protocol: Port Number:

Real Server IP Address:

TCP 192.168.11.202:http rr persistent 300

Illustration 58: Service Item for Load Balance in List

And then, add the host “172.16.11.201” into this service item.

Distribute IP Service Requests to other host(s)

Border >> Rule >> IP Load Balance

Create/Delete Service Item

Service IP Address:

Protocol: ▾

Port Number:

Add/Delete Server(s) within Service Item

Service IP Address:

Protocol: ▾


Port Number:

Real Server IP Address:

TCP 192.168.11.202:http rr persistent 300

Illustration 59: Add one Host into Service Item for Load Balance

Add the host '172.16.11.202" into this service item.

 **Distribute IP Service Requests to other host(s)**

Border >> Rule >> IP Load Balance

Create/Delete Service Item	Add/Delete Server(s) within Service Item
Service IP Address: <input type="text"/>	Service IP Address: <input type="text" value="192.168.11.202"/>
Protocol: <input type="text" value="TCP"/>	Protocol: <input type="text" value="TCP"/> Port Number: <input type="text" value="80"/>
Port Number: <input type="text"/>	Real Server IP Address: <input type="text" value="172.16.11.202"/>
<input type="button" value="Add"/> <input type="button" value="Delete"/>	<input type="button" value="Add"/> <input type="button" value="Delete"/>

```
TCP 192.168.11.202:http rr persistent 300
-> 172.16.11.201:http Masq 1 0 0
```

Illustration 60: Add another Host into Service Item for Load Balance

Distribute IP Service Requests to other host(s)

Border >> Rule >> IP Load Balance

Create/Delete Service Item

Service IP Address:

Protocol:

Port Number:

Add/Delete Server(s) within Service Item

Service IP Address:

Protocol: Port Number:

Real Server IP Address:

TCP 192.168.11.202:http rr persistent 300
-> 172.16.11.201:http Masq 1 0 0
-> 172.16.11.202:http Masq 1 0 0

Illustration 61: the Hosts associated with Service Item for Load Balance in List

Hence, we are able to see the hosts "172.16.11.201" and "172.16.11.202" taking care of the HTTP requests arriving at "192.168.11.202".

Use Web Proxy

The actions “DNAT”, “ACCEPT”, “DROP”, “REJECT” and “REDIRECT” are manipulating network traffic according to the IP address, or TCP/UDP port number. For web proxy, the control is specific to HTTP itself.

A proxy receives the requests from the client programs and forwards the requests to the actual server; it also gets the responses from the server and sends those responses to the original client programs. A web proxy can be used to cache the previously-loaded data, screen the web links, or control the time slot for web access. Those are the aspects in consideration by using web proxy from the zone “**loc**” to zone “**net**” (the Internet). However, proxy is with the usage to resolve other problems.

Consider the scenario that there are several web hosts for internal use in zone “**loc**”. For people in zone “**net**”, how to access those hosts in zone “**loc**”? The access from zone “**net**” to zone “**loc**” is blocked by default. However, if people have a way to access the proxy in zone “**fw**”, they would the chance to access the web hosts in zone “**loc**” because the connections from zone “**fw**” to zone “**loc**” are allowed. And the problem of accessing internal web hosts turns out to be the problem of accessing web proxy in zone “**fw**”. We usually use this scheme along with VPN. Using VPN alone might not resolve in the case that some of the web hosts do not have the setting of gateway to the subnet of VPN. The web proxy provided on the base platform is listening the requests on all its network interfaces so that accessing via proxy to the web hosts is just like to access the web host from base platform from the view of the web hosts. It can circumvent some routing table setting issues.

There might be some other reasons of using web proxy. For example, some legacy web applications are only accepting the connections from a specific IP address. To open the access of those legacy web applications to larger group of people without modifying the applications, using the web proxy to access the legacy web applications might be a good approach. In the following sections, we introduce how to change some setting of the web proxy provided by the base platform.

Web Caching

The web proxy is listening on TCP port 3128 by default on the base platform. And it can be changed via “**Border >> Proxy >> Web Caching**”.

Setting for Web caching

Border >> Proxy >> Web Caching

☐ Turn Off Proxy Functionalities

HTTP Port for Using Proxy:

Cache Size for Storing Web Pages: MB

☐ Turn on transparent proxy so that users do not need to set http Proxy in the Web browser. (It also needs to use REDIRECT in the Advanced Border Setting to redirect to the proxy port.)

Network allowed to access this proxy

Add

10.0.0.0/8
172.16.0.0/12
192.168.0.0/16
fc00::/7
fe80::/10

Submit

Remove

Illustration 62: Screen Snapshot for Web Proxy Caching and Access

To use web proxy, you need to change the setting on your web browser. If using “**REDIRECT**” action from the base platform to redirect Port 80 to this proxy port, this is called “Transparent Proxy”. And it does need to ask users to change the proxy setting on the web browser.

And the networks allowed to access this proxy are set on the right box. Those might be modified to restrict the access.

URL Screening

If web proxy is used, the list of blocking URLs can be added via “**Border >> Proxy >> URL Screening**”.

URL Screening via Proxy

Border >> Proxy >> URL Screening

Add to Blocked URL List

URL Domain : (e.g: .google.com)

☐ Block File Uploading in html form (multipart/form-data)

Note: data in https can not be deciphered and modified in midway; you need to block traffic with destination TCP port 443 directly if a file is submitted via https.

Remove from Blocked List

----- None in the list -----

Illustration 63: URL Screening in Web Proxy

And file uploading via HTML form can also be blocked. However, data in HTTPS can not be deciphered without the keys associated with certificates so that file uploading can not be blocked in HTTPS. You need to block the whole HTTPS traffic.

Access Block Time

The web access via proxy can be blocked during specific time frame.

The screenshot shows a web interface titled "Proxy Access Block Time" with a light beige background. At the top left is a lightbulb icon. Below the title is a breadcrumb trail: "Border >> Proxy >> Access Block Time". The interface is divided into two main sections. The left section, "Add Time Frame to Block Web Access", contains a "Weekday" dropdown menu set to "S (Sunday)", a "Time (HH:MM-HH:MM)" text input field, and a "Submit" button. Below these is an unchecked checkbox labeled "Perform Web Access Time Frame Checking" with another "Submit" button. A note at the bottom left states: "Note: in the time period setting h1:m1-h2:m2, the start time h1:m1 must be less than end time h2:m2 .". The right section, "Remove from Blocked List", features a large empty rectangular box with the text "----- None in the list -----" at the top. A "Remove" button is located at the bottom right of the interface.

Illustration 64: Time Slot Setting to Block HTTP Access on Web Proxy

For example, on the time slot from 08:00 to 12:00 on every Monday, it is forbidden to use HTTP. The field "Time" can be filled as "08:00-12:00" and select "M(Monday)" on the field "Weekday".

Traffic Bandwidth Control

To regulate the bandwidth of network traffic, we perform the setting according to the following procedure: set the limit for the physical network interface, classify the traffic of that interface into several groups and set the limit and priority for each group, and then determine the group of the packets according to source, destination, and TCP/UDP port numbers.

For example, we limit the inbound and outbound bandwidth of an interface “eth0” to 100 Mbits/sec and classify the traffic into 4 classes:

- Class 1: with highest priority (priority 1),
the minimum rate is 100 kbits/sec,
and the maximum allowed bandwidth is 180 kbits/sec;
- Class 2: with 2nd priority,
the minimum rate is $\frac{1}{4}$ of the total bandwidth of the interface,
and the maximum allowed bandwidth is the full bandwidth of the interface;
- Class 3: with 3rd priority,
the minimum rate is $\frac{1}{4}$ of the total bandwidth,
and the maximum allowed bandwidth is the full bandwidth;
- Class 4: with 4th priority,
the minimum rate is $\frac{1}{8}$ of the total bandwidth,
and the maximum allowed bandwidth is 80% of the total.

But what kind of traffic should be marked as “class 1”, “class 2”, “class 3”, or “class 4”? The mark should be made according to some attributes of the network packets, for example, source, destination, and TCP/UDP port number. And if the network packets do not match anything we specify, they will be marked as “class 3”. This is roughly done for the bandwidth setting for the network traffic.

In the following sections, the detailed steps will be introduced at “**Border >> Bandwidth >> Interface Limiting**”, “**Border >> Bandwidth >> Priority Classes**”, and “**Border >> Bandwidth >> Traffic Prioritizing**”.

Setting Network Interface Bandwidth

Setting Network Interface Bandwidth

Border >> Bandwidth >> Interface Limiting

Ethernet Interface(eth0,ppp0,...)

Incoming Bandwidth(kbits/sec)

Outgoing Bandwidth(kbits/sec)

Add

Listing of the setting limits

----- None in the list -----

Remove

Illustration 65: Setting Network Interface Bandwidth

The inbound bandwidth and outbound bandwidth of an Ethernet interface can be regulated via the setting at “**Border >> Bandwidth >> Interface Limiting**”. In the hardware specification of an Ethernet interface, the bandwidth of the interface is usually provided as “1000Mb/s”, “100Mb/s”, or “10Mb/s”; it is a value selected from some given numbers via auto-negotiation. Our setting here is not to touch that value. Instead, the setting here is a control process to determine how many frames we should send or receive from Ethernet interface.

For the setting of inbound bandwidth, the network frames have already been received by the Ethernet interface before the base platform processes them. Thus, the only way to regulate the inbound bandwidth is to drop some received packets if it over the limit we specify. Throwing out the received traffic sometimes means the other end needs to send them again. The effective bandwidth for the upper layers of applications might be much lower.

The following is an example to set inbound and outbound bandwidth as 100 Mbits/sec. Please note that the number needs to be converted in the unit kbits/sec:

Setting Network Interface Bandwidth

Border >> Bandwidth >> Interface Limiting

Ethernet Interface(eth0,ppp0,...)

Incoming Bandwidth(kbits/sec)


Outgoing Bandwidth(kbits/sec)

Listing of the setting limits

----- None in the list -----

Illustration 66: Setting Inbound and Outbound Bandwidth

After pressing “Add” button, the box on the right will display as

 **Setting Network Interface Bandwidth**

Border >> Bandwidth >> Interface Limiting

Ethernet Interface(eth0,ppp0,...)

Incoming Bandwidth(kbits/sec)

Outgoing Bandwidth(kbits/sec)

Add

Listing of the setting limits

eth0 === 100000kbit-->100000kbit

Remove

Illustration 67: Screen Snapshot after Setting Interface Bandwidth

In the meanwhile, 4 priority classes are created automatically. They can be seen via “**Border >> Bandwidth >> Priority Classes**”:

Define Priority Classes

Border >> Bandwidth >> Priority Classes

Interface

Mark

Minimum Rate

Max Allowed Bandwidth

Priority

Option


Priority Class List

```
eth0 == 1 100kbit 180kbit 1 tos=0x68/0xfc,tos=0xb8/0xfc
eth0 == 2 full/4 full 2 tcp-ack,tos-minimize-delay
eth0 == 3 full/4 full 3 default
eth0 == 4 full/8 full*8/10 4
```

Illustration 68: Priority Classes after Setting Interface Bandwidth Limit

You might modify the setting according to your application. In the screen snapshot above, Mark “3” is the default class. No matter how you modify, you should have a default class for the network traffic. Otherwise, the border engine will fail to start up successfully.

Define Priority Classes

 **Define Priority Classes**

Border >> Bandwidth >> Priority Classes

Interface

Mark

Minimum Rate

Max Allowed Bandwidth

Priority

Option

Submit

Priority Class List

----- None in the list -----

Remove

Illustration 69: Screen Snapshot for Defining Priority Classes

As indicated in the previous section, 4 priority classes will be created after the bandwidth of an Ethernet interface is set. And the setting can be modified thereafter. The field “Mark” can be an integer between 1-255; the field “Priority” can be an integer in the range 1-65535.

The field “Option” can be one of the following:

default:

to indicate the class of the traffic not marked explicitly

tos=0x**value**/0x**mask**:

this is to define a class by selecting IP packet's
TOS/Precedence/DiffServ Octet.

tos-tosname:

The following “tosnames” are used to represent some
of the TOS values of associated mask:

tos-minimize-delay	0x10/0x10
tos-maximize-throughput	0x08/0x08
tos-maximize-reliability	0x04/0x04
tos-minimize-cost	0x02/0x02
tos-normal-service	0x00/0x1e

tcp-ack:

This is for all the TCP-ack packets.

Packet Marking for Traffic Control

With the priority classes defined earlier, we can specify the detailed content for each priority class.

The screenshot shows a web interface for configuring packet marking. The title is "Packet Marking for Traffic Control" with a lightbulb icon. Below the title is a breadcrumb trail: "Border >> Bandwidth >> Traffic Prioritizing". On the left, there are five input fields: "Mark", "Packet Source", "Packet Destination", "Protocol" (with a dropdown menu showing "TCP"), and "Destination Port". Below these fields is an "Add" button. On the right, there is a section titled "Listing of the marking rules" which contains a large empty box with the text "None in the list" at the top. At the bottom right of this section is a "Remove" button.

Illustration 70: Screen Snapshot for Traffic Prioritizing

For example, if we would like to prioritize site-to-site VPN traffic (assume it is using UDP port 7777), then we can set as follows:

Mark: 1
Packet Source: 0.0.0.0/0
Packet Destination: 0.0.0.0/0
Protocol: UDP
Destination Port : 7777

This means: anywhere (0.0.0.0/0) to anywhere (0.0.0.0/0), the traffic of UDP port 7777 will be marked as “1”. And in the previous sections that there are 4 priority classes created right after setting the bandwidth of an Ethernet interface. Network packets with mark “1” are with the highest priority (the priority number is “1”) with maximal allowed bandwidth 180 kbits/sec.

Packet Marking for Traffic Control

Border >> Bandwidth >> Traffic Prioritizing

Mark: 1

Packet Source: 0.0.0.0/0

Packet Destination: 0.0.0.0/0

Protocol: UDP

Destination Port: 7777

Add

Listing of the marking rules

----- None in the list -----

Remove

Illustration 71: Setting Mark for Traffic Priority

Packet Marking for Traffic Control

Border >> Bandwidth >> Traffic Prioritizing

Mark

Packet Source

Packet Destination

Protocol

Destination Port

Listing of the marking rules

1 0.0.0.0/0 0.0.0.0/0 udp 7777

Illustration 72: Marking Rule Listing

Traffic prioritizing is used while bandwidth is very limited so that we would like to prioritize the network traffic in a very congested channel. However, it is not straightforward to define them well. In the example above, the minimum rate for top priority traffic is with 100 kbits/sec to keep minimum connections. And the maximal bandwidth is 180 kbits/sec that would not be enough for the applications running on top of this site-to-site VPN. However, if the maximum bandwidth is too large, it might occupy too many resources so that the other applications might be choked. Thus, the setting is very much dependent on the usage pattern on your environment.

Limiting the usage of the bandwidth usually happens when the system is under abuse so that it needs to limit the usage from specific area.

The Components of a Bridge

In the beginning sections of this document, we mention “bridge” of the basic unit under the network operations provided by the base platform. For a network interface of a virtual host to access the other hosts, it attaches to a bridge; for the zone definition of “net”, “loc”, “fw”, or “dmz”, we use the boundaries of the bridges to partition the whole world into these areas. And the IP addresses of the base platform are set on top of the bridge devices so that the other hosts can use these IP addresses to access the base platform.

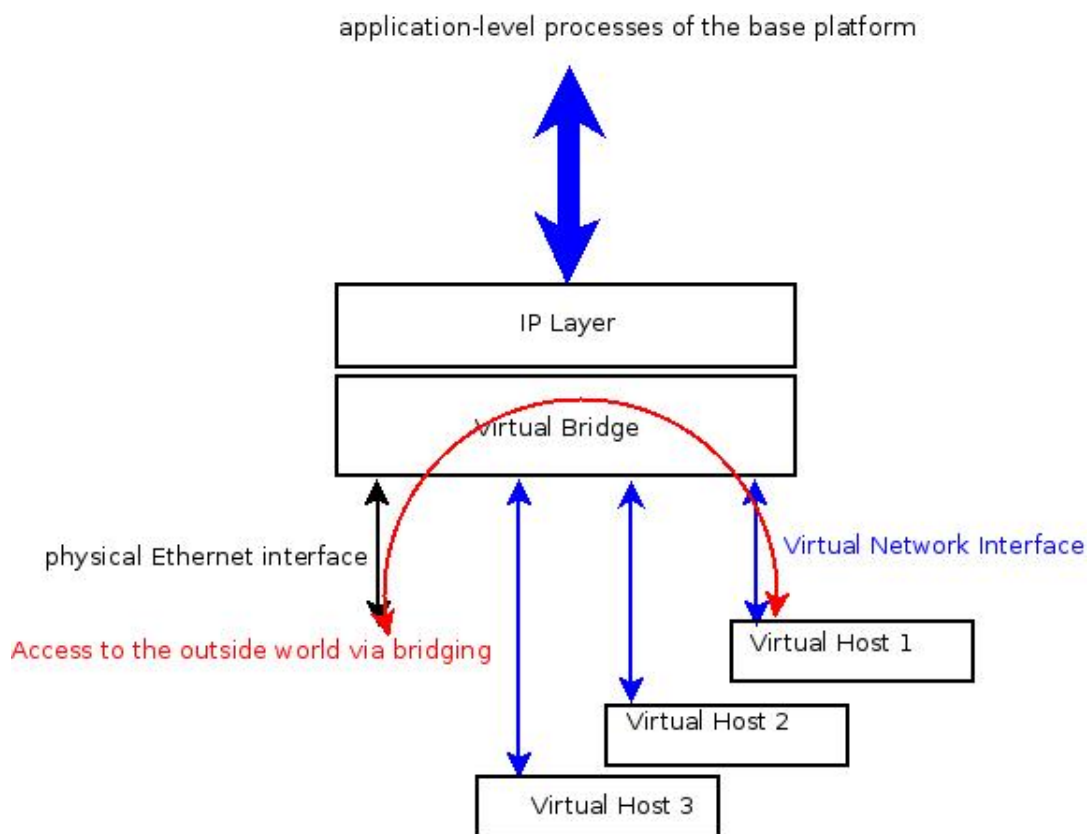



Illustration 73: Relationship among Bridge, Base Platform, Physical Ethernet Interface, and Virtual Hosts

There are 12 bridges on the base platform provided as “br0”, “br1”, “br2”, ..., up to “br11”. Ethernet Interfaces with physical entities are known as “eth0”, “eth1”, ... and so on. And the network devices “tap0”, “tap1”, ... are used by the VPN in bridge mode and the virtual hosts.

Is it possible to have multiple Ethernet interfaces with physical entities residing in one bridge? Yes, it is, but it is not economical. Nowadays, the hardware only for bridging use of Ethernet switch is cheaper than a router. Thus. Whenever there is a chance, you should make those physical interfaces to function as a router instead of a bridge unless there exist other specific uses.

The configuration can be changed via “**System >> Network >> Ethernet / DHCP**”. The network device “eth0” shall be in “br0” – this is should not be changed. And “br0” is used for WAN device (in zone “net”) with public IP address. Other than that, the others can be adjusted as following screen snapshot indicated:

 **Ethernet / DHCP**

System >> Network >> Ethernet / DHCP

Ethernet Bridge (br1)
IP Address:
Start IP:

☒ **Turn on DHCP Server**
Netmask:
End IP:

☒ **Enable Bridge br1**
Ethernet Ports in Bridge br1:

Ethernet Bridge (br2)
IP Address:
Start IP:

☒ **Turn on DHCP Server**
Netmask:
End IP:

☒ **Enable Bridge br2**
Ethernet Ports in Bridge br2:

Ethernet Bridge (br3)
IP Address:
Start IP:

☒ **Turn on DHCP Server**
Netmask:
End IP:

☒ **Enable Bridge br3**
Ethernet Ports in Bridge br3:

And DHCP server can be enabled/disabled along with its address pool.

At the moment of writing this document, the number of the bridges on the base platform is provided as 12, no matter how many physical Ethernet interfaces come to existence. If you have hardware with the number of physical Ethernet interfaces larger than 12, you might as well put them into one bridge:

Ethernet Bridge (br10)		<input checked="" type="checkbox"/> Turn on DHCP Server	
IP Address:	<input type="text" value="172.16.19.253"/>	Netmask:	<input type="text" value="255.255.255.0"/>
Start IP:	<input type="text" value="172.16.19.100"/>	End IP:	<input type="text" value="172.16.19.200"/>
			<input type="button" value="Submit"/>
<input checked="" type="checkbox"/> Enable Bridge br10			
Ethernet Ports in Bridge br10:			
<input type="text" value="eth10"/>		<input type="button" value="Submit"/>	
Ethernet Bridge (br11)		<input checked="" type="checkbox"/> Turn on DHCP Server	
IP Address:	<input type="text" value="172.16.20.253"/>	Netmask:	<input type="text" value="255.255.255.0"/>
Start IP:	<input type="text" value="172.16.20.100"/>	End IP:	<input type="text" value="172.16.20.200"/>
			<input type="button" value="Submit"/>
<input checked="" type="checkbox"/> Enable Bridge br11			
Ethernet Ports in Bridge br11:			
<input type="text" value="eth11 eth12 eth13 eth14 eth15 eth16 eth17"/>		<input type="button" value="Submit"/>	

Illustration 74: Place Multiple Ethernet interfaces with Physical Entities into One Bridge

The zone definition associated with “net”, “loc”, and “dmz” is described in the following sections.

Zone Definition

The following screen snapshot indicates the zones we use for network operations. They are “net”, “loc”, “dmz”, and “road”. The zone “fw” is not listed there because “fw” stands for the base platform itself network-wise.

Zone Definition of Each Ethernet Interface

Border >> Reshuffle >> Zone Setting

☒ Restore to the default setting while rebooting the system

Submit

Zone

Interface

Modify

List of Zone Setting

- net br0
- road ppp+
- loc br1
- dmz br2
- loc br3
- loc br4
- loc br5
- loc br6
- loc br7
- loc br8
- loc br9
- loc br10
- loc br11
- road tun+
- road pimreg

Remove

Illustration 75: Snapshot for Zone Setting

However, if the box on the top is still “checked”, the setting will be restored to the default setting after reboot. If you are very sure about what you have done, please remember to un-check that box.

Port Association for NAT Setting

Whether NAT(Network Address Translation) should be performed network packets passing the boundary of a bridge to the outside world of the base platform is determined by the setting in “**Border >> Reshuffle >> Port Association**”.

Port Association for NAT Setting

Border >> Reshuffle >> Port Association

WAN (net) Interface

LAN (dmz/loc) Subnet

Add

List of Binding Setting


- br0 172.16.9.0/24
- br0 172.16.11.0/24
- br0 172.16.12.0/24
- br0 172.16.13.0/24
- br0 172.16.14.0/24
- br0 172.16.15.0/24
- br0 172.16.16.0/24
- br0 172.16.17.0/24
- br0 172.16.18.0/24
- br0 172.16.19.0/24
- br0 172.16.20.0/24

Remove

Illustration 76: NAT Setting

The subnet of “br1” connecting to is “172.16.9.0/24”. Thus, traffic coming from that subnet to zone “net” will be replaced their source IP address by using the IP address of “br0”. Similarly, “172.16.11.0/24” is the subnet that “br2” connects to; “br3” connects to “172.16.12.0/24”.... “br11” connects to the subnet “172.16.20.0/24”. For the traffic from those subnets, they will be “NAT'd”.

IP Policy Routing

 **IP Policy Routing besides Main Routing Table**

Border >> Reshuffle >> Confined Routing

Traffic Rule Association with Routing Table
To/From
Subnet
Routing Table

Listing of Rules and Routing Tables
0: from all lookup local
32766: from all lookup main
32767: from all lookup default

Add Default Gateway on Non-main Routing Table
Default Gateway
WAN (net) Interface
Routing Table

Listing of Routing Table: moon
----- None in the list -----

Add Interface into non-main Routing Table
Subnet
IP Address
Ethernet Interface
Routing Table

Listing of Routing Table: star
----- None in the list -----

Add Routing Entry on Non-main Routing Table
Network
Gateway
Ethernet Interface
Routing Table

Illustration 77: Information for IP Policy Routing

Routing can be simply described as: looking at the content of IP header and checking the routing table to see where the packet should go. IP traffic usually should be forwarded according to the main routing table. On the base platform we have two other extra routing tables named “moon” and “star”. IP Policy Routing stands for the action by selecting a specific type of network traffic and forwarding them according to those extra routing tables.

Those processes can be done by the setting at **"Border >> Reshuffle >> Confined Routing"**. There are 3 tables shown on the right. The first one on top of the right hand is to display traffic to/from a specific subnet should be handled according to routing table "main", "moon", or "star". The 2nd and 3rd tables on the right are the content of the routing tables "moon" and "star". The content of the main routing table can be found at **"System >> Network >> Static Routing"**. The following example is demonstrate how to use the functions here.

We use IP policy routing to configure the base platform as follows: there are two bridges "br0" and "br4" in zone "net"; "br0" is connecting to the Internet whereas "br4" is connecting to the subnet "10.0.0.0/8" that belongs to an ISP (Internet Service Provider). And "br5" is connecting to a subnet "172.16.14.0/24" with IP address "172.16.14.253". In order to use the subnet "10.0.0.0/8", the ISP asks for using IP address "10.1.1.23" with default gateway "10.1.1.1" and netmask "255.0.0.0". We would like to place some equipments on the subnet "172.16.14.0/24" so that those equipments will be connecting to a server in ISP's private network "10.0.0.0/8". However, "172.16.14.0/24" is our own subnet so that ISP will not route the traffic with source IP addresses from "172.16.14.0/24". Thus, traffic from the subnet "172.16.14.0/24" to "10.0.0.0/8" will be performed NAT by using the IP address of "br4" while passing the boundary of the bridge "br4".

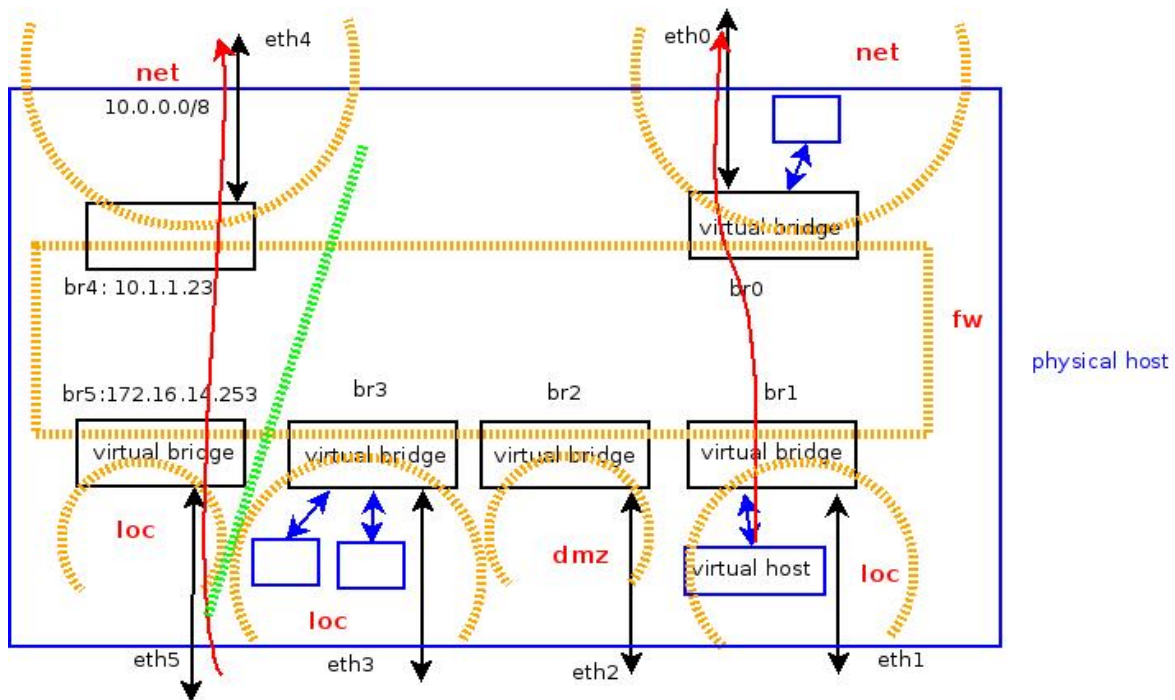


Illustration 78: Example of two WAN ports

We ask one question here: why we would like to put “br4” into zone “net”? The reason is: although it is a private subnet of ISP, there are some other customers on it. Thus, we would like to block the traffic from this subnet by default. The other part of the system should function as usual, and the subnets of “br4” and “br5” should be isolated from the rest of the system.

We proceed with the following setting below.

The screenshot displays the configuration interface for two Ethernet bridges, br4 and br5. For br4, the IP Address is 10.1.1.23, Netmask is 255.0.0.0, Start IP is 172.16.13.100, and End IP is 172.16.13.200. The 'Turn on DHCP Server' checkbox is unchecked. Below this, the 'Enable Bridge br4' checkbox is checked, and the 'Ethernet Ports in Bridge br4' field contains 'eth4'. For br5, the IP Address is 172.16.14.253, Netmask is 255.255.255.0, Start IP is 172.16.14.100, and End IP is 172.16.14.200. The 'Turn on DHCP Server' checkbox is checked. Below this, the 'Enable Bridge br5' checkbox is checked, and the 'Ethernet Ports in Bridge br5' field contains 'eth5'. Each section has a 'Submit' button.

Bridge	IP Address	Netmask	Start IP	End IP	Turn on DHCP Server	Enable Bridge	Ethernet Ports
br4	10.1.1.23	255.0.0.0	172.16.13.100	172.16.13.200	<input type="checkbox"/>	<input checked="" type="checkbox"/>	eth4
br5	172.16.14.253	255.255.255.0	172.16.14.100	172.16.14.200	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	eth5

Illustration 79: Change Bridge's IP Address

At first, the IP address of “br4” should be changed according to the assignment from ISP. It can be done via “**System >> Network >> Ethernet / DHCP**”. It should be in effect after rebooting the system.

And this “br4” should belong to zone “net”. We change its zone definition at “**Border >> Reshuffle >> Zone Setting**” by removing it from zone “loc” and adding it to “net”. To avoid it going back to default setting after reboot, please un-check the box on top.

The following screen snapshots to indicate removal of “br4” from “loc”:

Zone Definition of Each Ethernet Interface

Border >> Reshuffle >> Zone Setting

☒ Restore to the default setting while rebooting the system

Zone

Interface

Submit


Modify

List of Zone Setting

- net br0
- road ppp+
- loc br1
- dmz br2
- loc br3
- loc br4
- loc br5
- loc br6
- loc br7
- loc br8
- loc br9
- loc br10
- loc br11
- road tun+
- road pimreg

Remove

Illustration 80: Remove "br4" from zone "loc"

 **Zone Definition of Each Ethernet Interface**

Border >> Reshuffle >> Zone Setting

☐ Restore to the default setting while rebooting the system

Zone

Interface

Submit

Modify

List of Zone Setting

net br0
road ppp+
loc br1
dmz br2
loc br3
loc br5
loc br6
loc br7
loc br8
loc br9
loc br10
loc br11
road tun+
road pimreg

Remove

Illustration 81: Listing after Removing "br4" from zone "loc"

The following screen snapshots are for adding “br4” to zone “net”:

Zone Definition of Each Ethernet Interface

Border >> Reshuffle >> Zone Setting

☐ Restore to the default setting while rebooting the system

Zone:

Interface:

List of Zone Setting

- net br0
- road ppp+
- loc br1
- dmz br2
- loc br3
- loc br5
- loc br6
- loc br7
- loc br8
- loc br9
- loc br10
- loc br11
- road tun+
- road pimreg

Illustration 82: Add "br4" to zone "net"

Zone Definition of Each Ethernet Interface

Border >> Reshuffle >> Zone Setting

☐ Restore to the default setting while rebooting the system

Zone

Interface

Submit

Modify

List of Zone Setting

- net br0
- road ppp+
- loc br1
- dmz br2
- loc br3
- loc br5
- loc br6
- loc br7
- loc br8
- loc br9
- loc br10
- loc br11
- road tun+
- road pimreg
- net br4

Remove

Illustration 83: List for "br4" in zone "net"

Then we change the setting to let the traffic from the subnet of "br4" to the subnet of "br4" be under NAT; in other word, the source IP address will be replaced by using the IP address of "br4". This can be done at "**Border >> Reshuffle >> Port Association**". At first, remove the original setting of "br4" and "br5".

Port Association for NAT Setting

Border >> Reshuffle >> Port Association

WAN (net) Interface:

LAN (dmz/loc) Subnet:

Add

List of Binding Setting

- br0 172.16.9.0/24
- br0 172.16.11.0/24
- br0 172.16.12.0/24
- br0 172.16.13.0/24**
- br0 172.16.14.0/24
- br0 172.16.15.0/24
- br0 172.16.16.0/24
- br0 172.16.17.0/24
- br0 172.16.18.0/24
- br0 172.16.19.0/24
- br0 172.16.20.0/24

Remove

Illustration 84: Remove Original Subnet of "br4" for Using NAT under "br0"

Port Association for NAT Setting

Border >> Reshuffle >> Port Association

WAN (net) Interface:

LAN (dmz/loc) Subnet:


Add

List of Binding Setting

- br0 172.16.9.0/24
- br0 172.16.11.0/24
- br0 172.16.12.0/24
- br0 172.16.14.0/24**
- br0 172.16.15.0/24
- br0 172.16.16.0/24
- br0 172.16.17.0/24
- br0 172.16.18.0/24
- br0 172.16.19.0/24
- br0 172.16.20.0/24

Remove

Illustration 85: Remove Subnet of "br4" Using NAT under "br0"

 **Port Association for NAT Setting**

Border >> Reshuffle >> Port Association


WAN (net) Interface

LAN (dmz/loc) Subnet

List of Binding Setting

- br0 172.16.9.0/24
- br0 172.16.11.0/24
- br0 172.16.12.0/24
- br0 172.16.15.0/24
- br0 172.16.16.0/24
- br0 172.16.17.0/24
- br0 172.16.18.0/24
- br0 172.16.19.0/24
- br0 172.16.20.0/24

Illustration 86: Add Subnet of "br5" Using NAT under "br4"

 **Port Association for NAT Setting**

Border >> Reshuffle >> Port Association

WAN (net) Interface

LAN (dmz/loc) Subnet

Add

List of Binding Setting

br0 172.16.9.0/24

br0 172.16.11.0/24

br0 172.16.12.0/24

br0 172.16.15.0/24

br0 172.16.16.0/24

br0 172.16.17.0/24

br0 172.16.18.0/24

br0 172.16.19.0/24

br0 172.16.20.0/24

br4 172.16.14.0/24

Remove

Illustration 87: List for Subnet of "br5" Using NAT under "br5"

After we adjust the IP address of “br4”, the zones of “br4” and “br5”, and the relationship between “br5” and “br4” (for NAT), we focus on setting the policies for routing. As mentioned earlier, the 1st box on the right is to display the corresponding routing table to use for specific type of traffic. And it is performed from top to bottom for every network packet passing through the base platform; if one rule is not matched, it goes to the next rule to check. Thus, the bottom two lines are using “main” routing table and “default”.

IP Policy Routing besides Main Routing Table

Border >> Reshuffle >> Confined Routing

Traffic Rule Association with Routing Table

To/From: To

Subnet: 10.0.0.0/8

Routing Table: moon

Add

Add Default Gateway on Non-main Routing Table

Default Gateway:

WAN (net) Interface:

Routing Table: moon

Add

Add Interface into non-main Routing Table

Subnet:

IP Address:

Ethernet Interface:

Routing Table: moon

Add

Add Routing Entry on Non-main Routing Table

Network:

Gateway:

Ethernet Interface:

Routing Table: moon

Add

Listing of Rules and Routing Tables

0: from all lookup local
32766: from all lookup main
32767: from all lookup default

Remove

Listing of Routing Table: moon

----- None in the list -----

Remove

Listing of Routing Table: star

----- None in the list -----

Remove

Illustration 88: Catch Traffic Destined to the Subnet "10.0.0.0/8"

The idea is simple: we would like to catch to/from the subnets of “br4” and “br5”, and use the “moon” routing table to determine where those network packets should go. The following screen snapshot indicates that we would like to catch the traffic from the subnet “10.0.0.0/8” and force them to look up the routing table “moon”:

IP Policy Routing besides Main Routing Table

Border >> Reshuffle >> Confined Routing

Traffic Rule Association with Routing Table

To/From: From ▾

Subnet: 10.0.0.0/8

Routing Table: moon ▾

Add

Add Default Gateway on Non-main Routing Table

Default Gateway:

WAN (net) Interface:

Routing Table: moon ▾

Add

Add Interface into non-main Routing Table

Subnet:

IP Address:

Ethernet Interface:

Routing Table: moon ▾

Add

Add Routing Entry on Non-main Routing Table

Network:

Gateway:

Ethernet Interface:

Routing Table: moon ▾

Add

Listing of Rules and Routing Tables

0: from all lookup local
32765: from all to 10.0.0.0/8 lookup moon
32766: from all lookup main
32767: from all lookup default

Remove

Listing of Routing Table: moon

----- None in the list -----

Remove

Listing of Routing Table: star

----- None in the list -----

Remove

Illustration 89: Catch the Traffic from the Subnet "10.0.0.0/8"

Please remember that “10.0.0.0/8” is the subnet that “br4” connects to. There are chances that some other subnets are sitting behind this subnet and they can be reached via the gateway provided by the ISP. Whether or not we should catch those packets to those subnets as well? A better approach is that we focus on catching the “source” subnets instead of listing all the destination subnets. There might be numerous subnets that all the packets are destined, but the number of source subnets are quite limited.

Therefore, for those subnets for the field “To” in the configuration screen, we only specify adjacent subnets. For those “remote subnets”, we let the default gateway in the routing table to handle them.

IP Policy Routing besides Main Routing Table

Border >> Reshuffle >> Confined Routing

Traffic Rule Association with Routing Table

To/From: From ▾

Subnet: 172.16.14.0/24

Routing Table: moon ▾

Add

Add Default Gateway on Non-main Routing Table

Default Gateway:

WAN (net) Interface:

Routing Table: moon ▾

Add

Add Interface into non-main Routing Table

Subnet:

IP Address:

Ethernet Interface:

Routing Table: moon ▾

Add

Add Routing Entry on Non-main Routing Table

Network:

Gateway:

Ethernet Interface:

Routing Table: moon ▾

Add

Listing of Rules and Routing Tables

0: from all lookup local
 32763: from all to 172.16.14.0/24 lookup moon
 32764: from 10.0.0.0/8 lookup moon
 32765: from all to 10.0.0.0/8 lookup moon
 32766: from all lookup main
 32767: from all lookup default

Remove

Listing of Routing Table: moon

----- None in the list -----

Remove

Listing of Routing Table: star

----- None in the list -----

Remove

Illustration 90: Catch the Traffic from "172.16.14.0/24"

IP Policy Routing besides Main Routing Table

Border >> Reshuffle >> Confined Routing

Traffic Rule Association with Routing Table

To/From: To

Subnet: 172.16.14.0/24

Routing Table: moon

Add

Add Default Gateway on Non-main Routing Table

Default Gateway:

WAN (net) Interface:

Routing Table: moon

Add

Add Interface into non-main Routing Table

Subnet:

IP Address:

Ethernet Interface:

Routing Table: moon

Add

Add Routing Entry on Non-main Routing Table

Network:

Gateway:

Ethernet Interface:

Routing Table: moon

Add

Listing of Rules and Routing Tables

0: from all lookup local
32764: from 10.0.0.0/8 lookup moon
32765: from all to 10.0.0.0/8 lookup moon
32766: from all lookup main
32767: from all lookup default

Remove

Listing of Routing Table: moon

----- None in the list -----

Remove

Listing of Routing Table: star

----- None in the list -----

Remove

Illustration 91: Catch the Traffic Destined to "172.16.14.0/24"

And please remember "172.16.14.0/24" is the subnet that "br5" connects to.

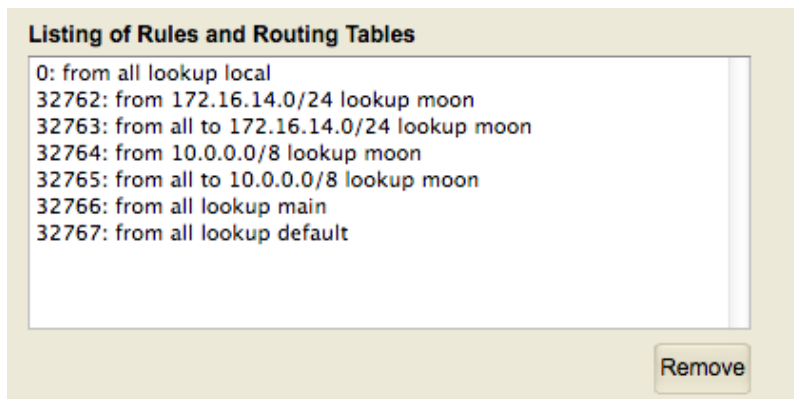


Illustration 92: List Rules for Lookup Routing Tables

The box above shows traffic to/from the subnet “172.16.14.0/24” and “10.0.0.0/8” should look up the routing table “moon”; the rest of them should check the “main” routing table.

The rest of the work is to create routing entries in the “moon” routing table. Imagine the situation that a host only with the interfaces of “br4” and “br5”. We would like to create the content of the routing table similar to that case. Usually the routing entries for the subnets that the Ethernet interfaces are directly connected to are created at the time while setting the IP addresses of those Ethernet interfaces in the “main” routing table. But these steps will not be done automatically in the other routing tables. We have to add them manually.

Other than the subnets associated with Ethernet interfaces, a default gateway is needed. If there is no routing entry matched the destination of a network packet, the packet is sent to the default gateway. There might be some chances to add some gateways to the other local subnets. These are roughly all the scenarios for the entries in the routing table.

We add default gateway to the “moon” routing table as follows:

IP Policy Routing besides Main Routing Table

Border >> Reshuffle >> Confined Routing

Traffic Rule Association with Routing Table

To/From: To
Subnet:
Routing Table: moon
Add

Add Default Gateway on Non-main Routing Table

Default Gateway: 10.1.1.1
WAN (net) Interface: br4
Routing Table: moon
Add

Add Interface into non-main Routing Table

Subnet:
IP Address:
Ethernet Interface:
Routing Table: moon
Add

Add Routing Entry on Non-main Routing Table

Network:
Gateway:
Ethernet Interface:
Routing Table: moon
Add

Listing of Rules and Routing Tables

- 0: from all lookup local
- 32762: from 172.16.14.0/24 lookup moon
- 32763: from all to 172.16.14.0/24 lookup moon
- 32764: from 10.0.0.0/8 lookup moon
- 32765: from all to 10.0.0.0/8 lookup moon
- 32766: from all lookup main
- 32767: from all lookup default

Listing of Routing Table: moon


----- None in the list -----

Listing of Routing Table: star

----- None in the list -----

Illustration 93: Add Default Gateway in the "moon" Routing Table

After pressing “Add” button, the corresponding routing entry will be displayed on the right under “moon” routing table. If nothing is shown after adding default gateway, it means the intended setting can not be correct. The possible reason is: the IP address of “br4” is not in effect so that you need to go back to check IP address is set properly and reboot the system to put it in effect. It is an obvious mistake that the gateway and the IP address of the interface are not in the same subnet.


IP Policy Routing besides Main Routing Table

Border >> Reshuffle >> Confined Routing

Traffic Rule Association with Routing Table

To/From

Subnet

Routing Table

Add

Listing of Rules and Routing Tables

0: from all lookup local
32762: from 172.16.14.0/24 lookup moon
32763: from all to 172.16.14.0/24 lookup moon
32764: from 10.0.0.0/8 lookup moon
32765: from all to 10.0.0.0/8 lookup moon
32766: from all lookup main
32767: from all lookup default

Remove

Add Default Gateway on Non-main Routing Table

Default Gateway

WAN (net) Interface

Routing Table

Add

Listing of Routing Table: moon

default via 10.1.1.1 dev br4 linkdown

Remove

Add Interface into non-main Routing Table

Subnet

IP Address

Ethernet Interface

Routing Table

Add

Listing of Routing Table: star

----- None in the list -----

Remove

Add Routing Entry on Non-main Routing Table

Network

Gateway

Ethernet Interface

Routing Table

Add

Remove

Illustration 94: Default Gateway in the "moon" Routing Table

The default gateway should be displayed as in the diagram above. To add the routing entries for those directly-connected subnets, the process is very similar to setting the IP address and netmask for an Ethernet interface.

The following screen snapshot is for adding the routing entry for the subnet that "br4" is directly-connected:

IP Policy Routing besides Main Routing Table

Border >> Reshuffle >> Confined Routing

Traffic Rule Association with Routing Table

To/From: To
Subnet:
Routing Table: moon
Add

Add Default Gateway on Non-main Routing Table

Default Gateway:
WAN (net) Interface:
Routing Table: moon
Add

Add Interface into non-main Routing Table

Subnet: 10.0.0.0/8
IP Address: 10.1.1.23
Ethernet Interface: br4
Routing Table: moon
Add

Add Routing Entry on Non-main Routing Table

Network:
Gateway:
Ethernet Interface:
Routing Table: moon
Add

Listing of Rules and Routing Tables

- 0: from all lookup local
- 32762: from 172.16.14.0/24 lookup moon
- 32763: from all to 172.16.14.0/24 lookup moon
- 32764: from 10.0.0.0/8 lookup moon
- 32765: from all to 10.0.0.0/8 lookup moon
- 32766: from all lookup main
- 32767: from all lookup default


Listing of Routing Table: moon

- default via 10.1.1.1 dev br4 linkdown




Listing of Routing Table: star



----- None in the list -----



Illustration 95: Add routing entry for the subnet connecting to "br4" in "moon"



 **IP Policy Routing besides Main Routing Table**

Border >> Reshuffle >> Confined Routing

Traffic Rule Association with Routing Table
To/From 
Subnet
Routing Table 



Add Default Gateway on Non-main Routing Table
Default Gateway
WAN (net) Interface
Routing Table 


Add Interface into non-main Routing Table
Subnet
IP Address
Ethernet Interface
Routing Table 


Add Routing Entry on Non-main Routing Table
Network
Gateway
Ethernet Interface
Routing Table 



Listing of Rules and Routing Tables

0: from all lookup local
32762: from 172.16.14.0/24 lookup moon
32763: from all to 172.16.14.0/24 lookup moon
32764: from 10.0.0.0/8 lookup moon
32765: from all to 10.0.0.0/8 lookup moon
32766: from all lookup main
32767: from all lookup default



Listing of Routing Table: moon

default via 10.1.1.1 dev br4 linkdown
10.0.0.0/8 dev br4 scope link src 10.1.1.23 linkdown



Listing of Routing Table: star

----- None in the list -----




Illustration 96: Add routing entry for the subnet connecting to "br5" in "moon"

The screen snapshot above is for adding the routing entry for the subnet that "br5" is connecting to.

The final result of the content of “moon” routing table is as follows:

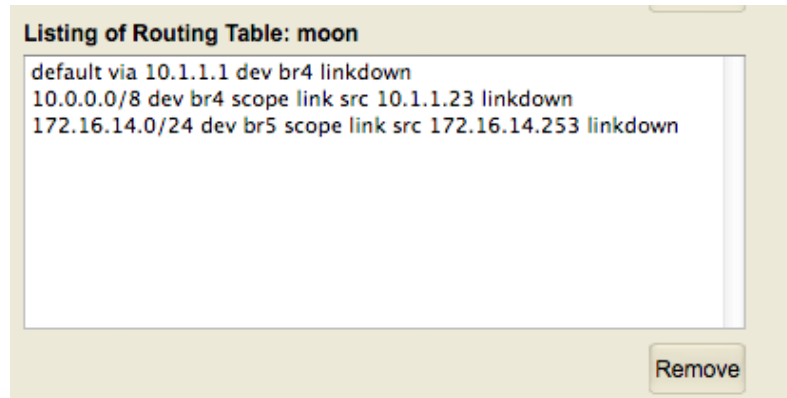


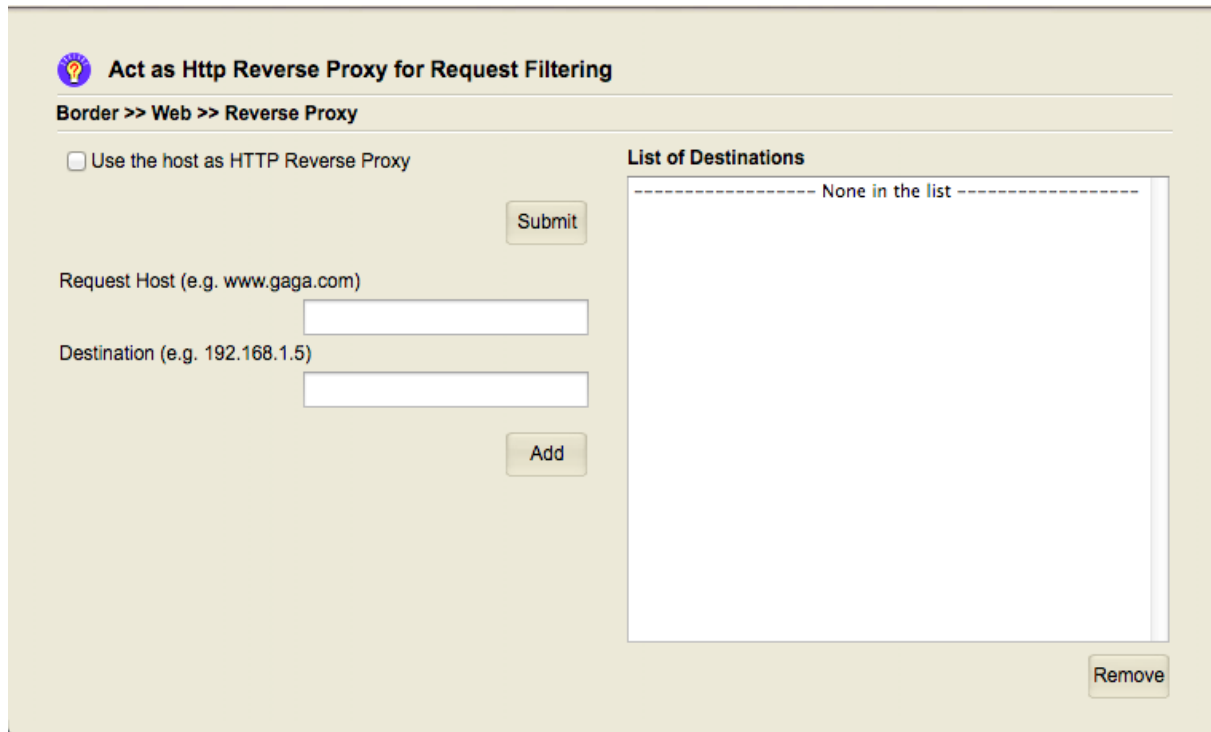
Illustration 97: Content of "moon" Routing Table

There exists another routing table named “star”. If you have another pair of bridges for similar use, the routing entries for those specific traffic can be placed into that routing table.

IP policy routing also can be used to set up some rules to send traffic to another machine for some other purposes. The steps are not as complicated as we introduce here. However, the network routing here is only limited to “static routing”. Dynamic routing only happens in “main” routing table. Those dynamic routing topics will be introduced later.

Http Reverse Proxy for Request Filtering


In the previous sections, we cover the topics for “port forwarding” and “IP load balance”. The two approaches can forward HTTP requests (TCP Port 80) to a server in the private network.



The screenshot shows a web interface titled "Act as Http Reverse Proxy for Request Filtering". Below the title is a breadcrumb trail: "Border >> Web >> Reverse Proxy". There is a checkbox labeled "Use the host as HTTP Reverse Proxy" which is currently unchecked. To the right of this checkbox is a "Submit" button. Below the checkbox are two input fields: "Request Host (e.g. www.gaga.com)" and "Destination (e.g. 192.168.1.5)". Below the "Destination" field is an "Add" button. To the right of these fields is a large rectangular area titled "List of Destinations" which currently displays "None in the list". At the bottom right of this area is a "Remove" button.

Illustration 98: Screen Snapshot for HTTP Reverse Proxy

However, “port forwarding” and “IP load balance” are handling the requests based on the content of “IP header”. If there are two host names “www.gaga.z” and “www.dada.z” pointing to the same IP address, it can not tell the difference from HTTP requests with the two host names by looking at IP header. The reverse proxy approach allows you to send the requests to different internal servers according to the host names in URLs.

 **Act as Http Reverse Proxy for Request Filtering**

Border >> Web >> Reverse Proxy

☒ Use the host as HTTP Reverse Proxy

Submit

Request Host (e.g. www.gaga.com)

Destination (e.g. 192.168.1.5)

Add

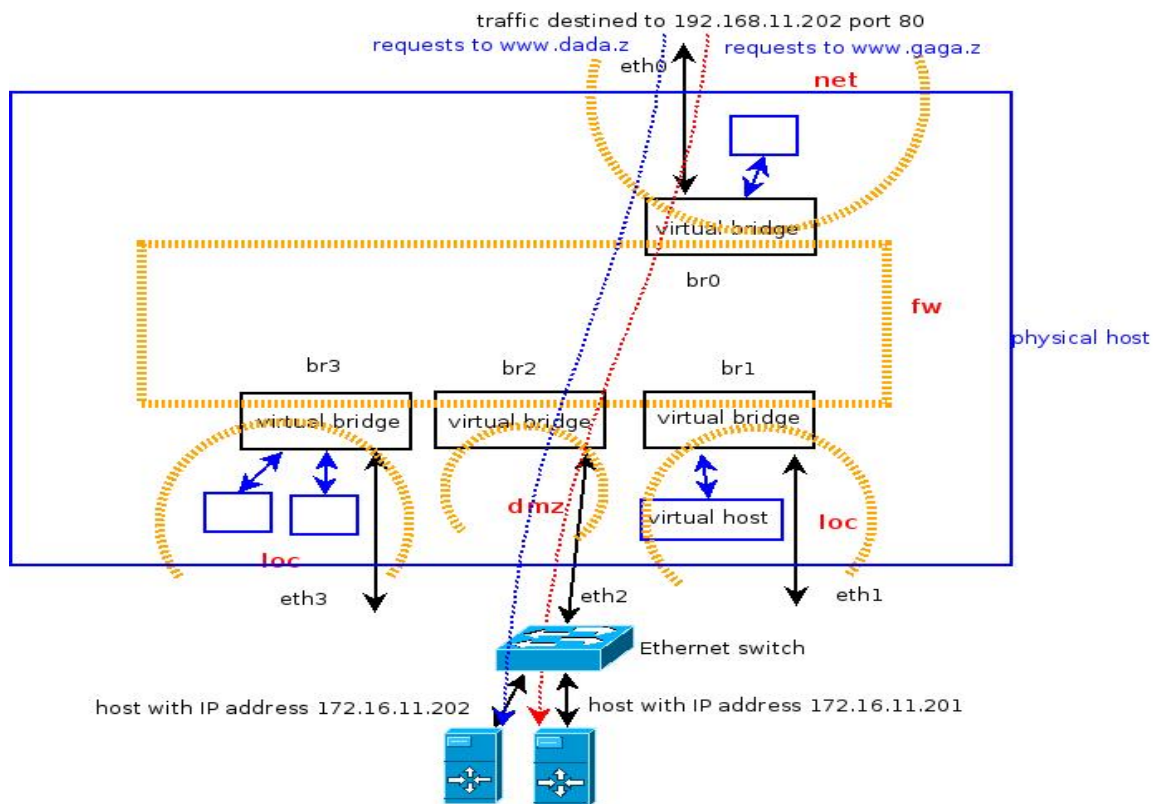
List of Destinations

www.gaga.z--> 172.16.11.201
www.dada.z--> 172.16.11.202

Remove

Illustration 99: Http Reverse Proxy for Two Different Host Names

Illustration 100: Http Reverse Proxy for Two Hosts



Chapter 4 VPN

VPN (Virtual Private Network) in this document means: we create virtual network interfaces both on VPN server and VPN client(s), and these virtual network interfaces pack the network traffic sent to the virtual network interfaces and wrap them as the payloads of physical network interfaces to reach the other end. On the other end, the VPN server/client unwrap the payload and let application-level software components get network traffic from the virtual network interfaces. From application point of view, VPN is just another network interface that can be used to send or receive network traffic.

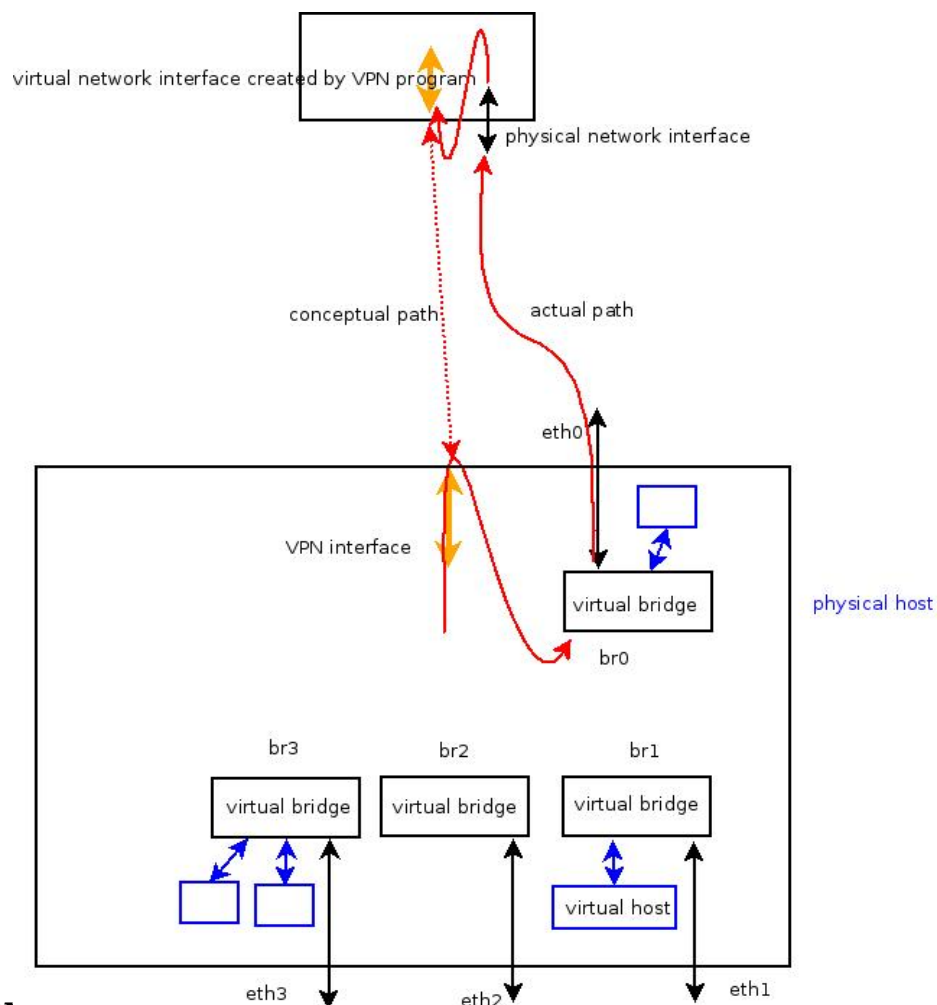


Illustration 4.1.1: VPN Operation Principle

There might exist other types of VPN in operation. But in this document the VPN we are referring to is using virtual network interface to interact with the other system components.

In terms of network topology, there exist client-to-site VPN and site-to-site VPN; in terms of the level of network traffic it encapsulates, it can run in bridging mode or routing mode. While VPN is running in routing mode, it has its own IP subnet and virtual network interfaces created by VPN processes are with IP addresses in that subnet. To use VPN in this mode, just route the traffic to/via that subnet.

For VPN running in bridging mode, Ethernet frames are encapsulated and carried from one end to the other. In the common practice of network planning, we use “bridging” within an IP subnet. Thus, VPN running in bridging mode tends to be in a small scale. If it is running in a big scale, it means that IP subnet will be very big so that some network traffic can not be easily isolated to a small portion of the network.

Client-to-site VPN is an VPN client connecting to a VPN server that the server is with an IP subnet behind so that network applications can use the virtual network devices created by VPN client to access the hosts located in the subnet behind VPN server.

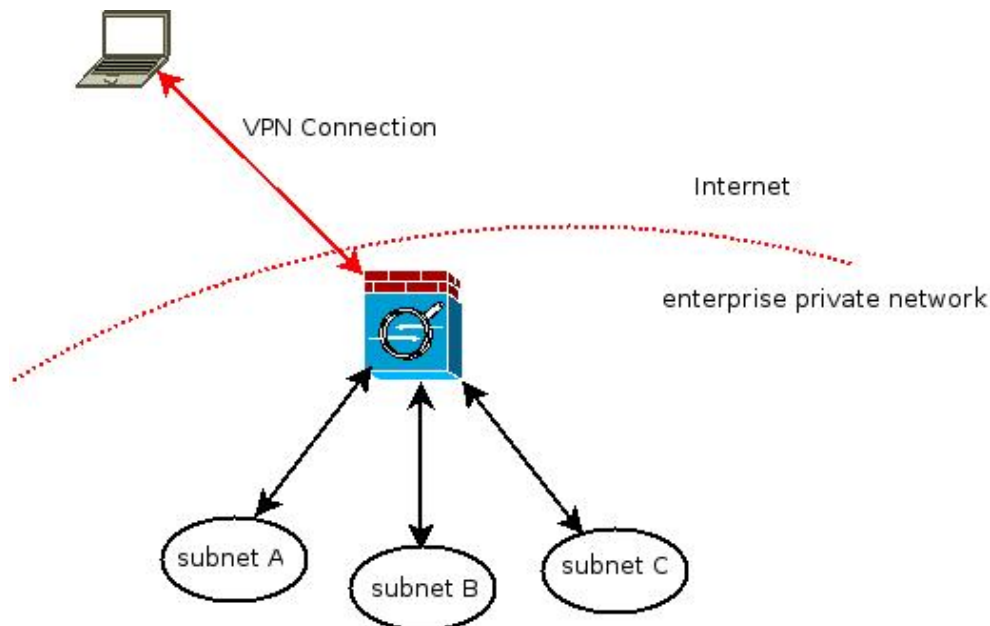
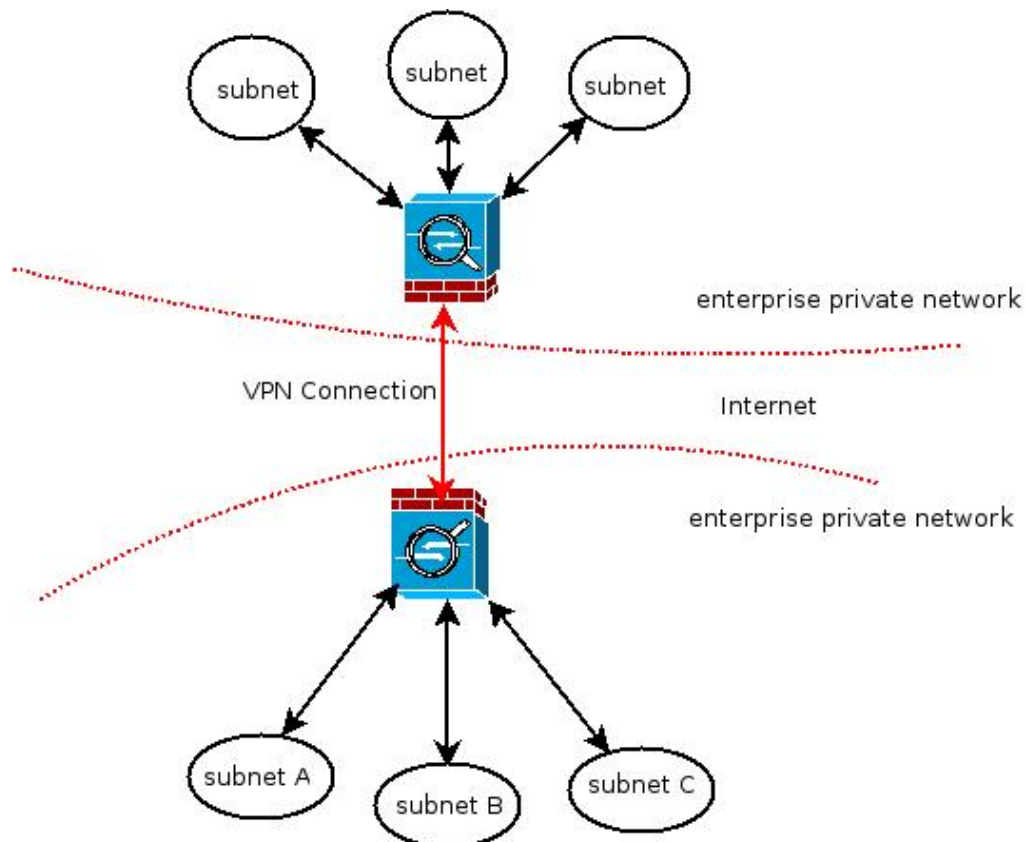


Illustration 102: Client-to-Site VPN

Site-to-site VPN is the scenario that two machines establish VPN connection so that the subnet behind each of them can access the network of the remote side.

**Illustration 103: Site-to-Site VPN**

If client-to-site is running at “bridging mode”, the client's IP address in VPN shall belong to one of the subnets in enterprise private network. Similarly, site-to-site VPN running in “bridging mode” implies one IP subnet is across two sites. We do not support client-to-site VPN in bridging mode; we only support in routing mode.

Please recall what we emphasize on the beginning of this document. Bridging is only done in data link layer; the system looks at the MAC address of an Ethernet frame to determine which interface it should go. As for routing, the system is checking the IP address in the IP header.

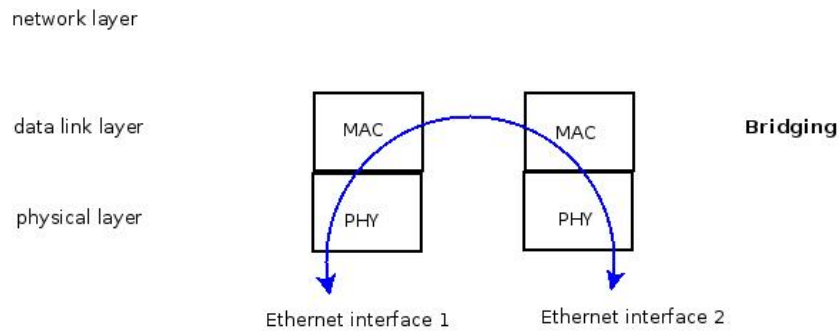


Illustration 104: Bridging

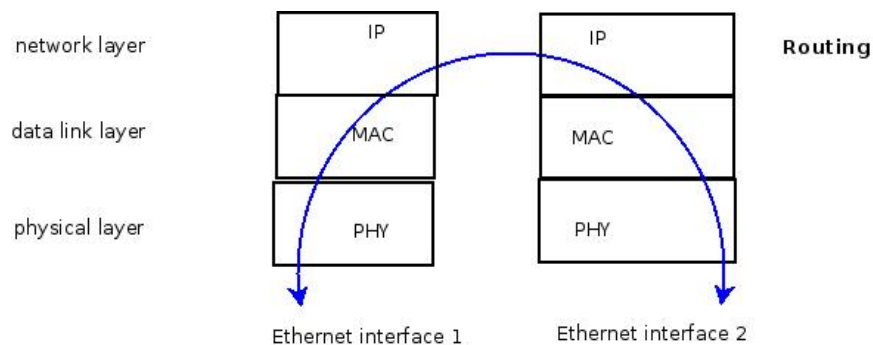


Illustration 105: Routing

For VPN running in bridging mode, that virtual network interface created by VPN process does not have an IP address; it should join an bridge to exchange traffic with the other network interfaces in the bridge. In routing mode, the virtual network interface created by VPN process is with IP address, and the routing table on the system determines where to go for network packets.

The VPN on the base platform is with its own CA (Certificate Authority) so that it can issue digital certificates by itself. The certificates are used for user authentication and encrypting information on the control channel in the VPN provided by the base platform. This “key generation process” will be used in all scenarios. Thus, we have brief introduction without too many technical details.

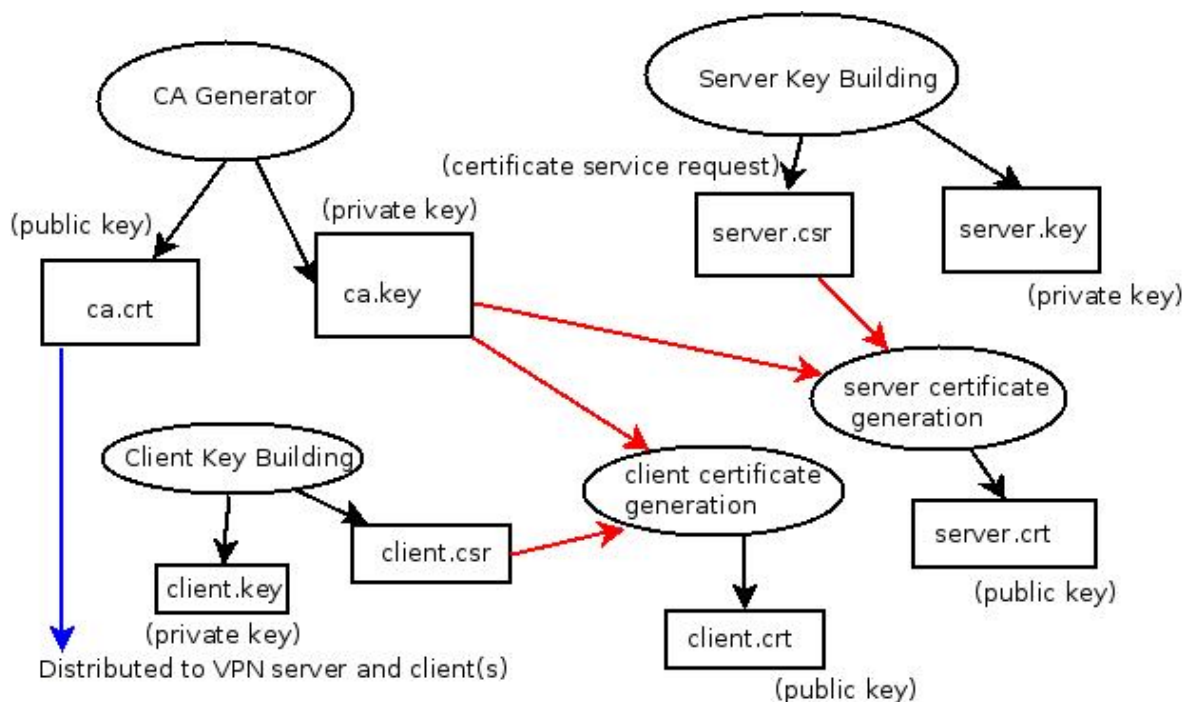


Illustration 106: VPN Key Generation Process

It starts with CA generation. It produces a pair of keys (“private key” and “public key”). The message encrypted by using “private key” can only be deciphered by “public key”. This CA “public key” will be provided to VPN server and clients to verify if a certificate is issued from this CA. For VPN server suite, the “public key” and “certificate service request” will be produced; this “certificate service request” will be submitted to CA and CA uses its “private key” to produce “server certificate”. This “server certificate” contains not only the “public key” of the server, but also the signature of CA by using CA “private key”. The VPN client suite also goes through similar process to produce “private key” and “client certificate”.

Once VPN client connects to VPN server, they will send the certificate file to the other end, and each party uses CA public key to verify if the certificate is issued from this CA. There are some other algorithms involved to select the cipher for data encryption and the negotiation process for the key used by the cipher.

Client-to-Site VPN Connection

As we mentioned earlier, the base platform does not provide client-to-site VPN on “bridging mode”; it is working on routing mode. Thus, it is necessary to arrange an IP subnet for VPN to use. The VPN is with control channel and data channel. It is not on TLS/SSL, but it is using the algorithms and ciphers provided by TLS/SSL. And it is using UDP port 1194.

The control channel of VPN is used to negotiate the key for the cipher in the data channel. Usually we use AES in CBC mode or GCM mode.

If the VPN is using the subnet “172.16.38.0/24”, then the IP address of VPN server in this subnet is “172.16.38.1”. You can use that IP address to access the VPN server. For example, if you install a virtual host in the base platform with TCP port 5904 for accessing the console via VNC, you can access the console of the virtual host by VPN with the following setting in VNC viewer:

172.16.38.1 TCP port 5904

The screenshot shows a web-based configuration interface titled "Subnet Allocated for VPN" with a "Setup Wizard: Steps 1/4 - Next" indicator. The breadcrumb trail is "Vpn >> Connection >> Address Pool". The form contains the following fields and options:

- Network Address:** Text input field containing "172.16.38.0".
- Netmask:** Text input field containing "255.255.255.0".
- Maximum Number Of Concurrent Clients:** Text input field containing "91".
- ☐ Turn Off VPN Server Process
- ☐ Allow Client to Client
- ☒ Force to use TLS1.2

There are two "Submit" buttons: one at the top right and one at the bottom center. A note at the bottom states: "The IP address of VPN server will be the first one in the range you specify on above. Changing Data Cipher requires all the clients fetching new configuration."

Illustration 107: Client-to-Site VPN Address Pool

Subnet Allocated for VPN Setup Wizard: Steps 1/4 - Next

Vpn >> Connection >> Address Pool

Network Address: 172.16.38.0 ☐ Turn Off VPN Server Process

Netmask: 255.255.255.0

Maximum Number Of Concurrent Clients: 91

☐ Allow Client to Client

☒ Force to use TLS1.2

Data Cipher: AES-128-CBC

- ✓ AES-128-CBC
- AES-192-CBC
- AES-256-CBC
- AES-128-GCM
- AES-192-GCM
- AES-256-GCM
- CAMELLIA-128-CBC
- CAMELLIA-192-CBC
- CAMELLIA-256-CBC
- SEED-CBC
- CAST5-CBC
- BF-CBC

The IP address of VPN server will you specify on above. Changing D clients fetching new configuration.

Illustration 108: Selection of Data Cipher

If you are provided with the versions that allow you to select ciphers, you might select them according to your hardware computation power. Here is a note for the ciphers SEED, CASTS, and BF(Blowfish) – they are 64-bit ciphers. It is reported that they are be cracked after your user is collecting around 700GB data. By avoiding this scenario to happen, the re-negotiation for the key of the cipher will happen after sending 64MB data. Thus, those ciphers are still safe to use if you have less powerful hardware.

The VPN server will assign the IP addresses to VPN clients from the designated subnet. By changing the routing table on the VPN client side, it can control the access from VPN clients to the subnets behind VPN server.

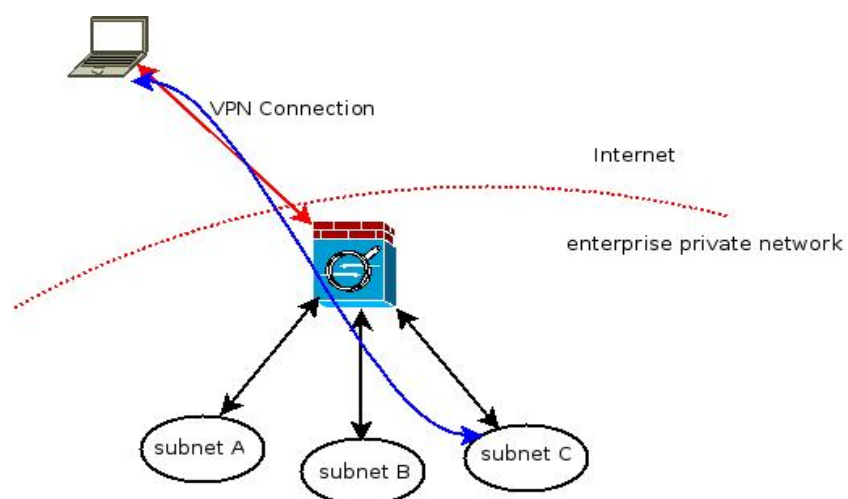


Illustration 109: VPN Client to Access a Subnet

From the diagram above, if a VPN client wants to access a host in subnet C, it is necessary to let the VPN server push the setting of the subnet C at “**Vpn >> Connection >> Pushed Setting**”:

Setting to be pushed to VPN Client(s) Setup Wizard: Previous - Steps 2/4 - Next

Vpn >> Connection >> Pushed Setting

Traffic Routing Server at the VPN Destination:

Destination Network:

Netmask:

Add

172.16.38.0/255.255.255.0

Remove

☐ Redirect Default Gateway

Submit

Setting Published via DHCP in VPN:

WINS

Add

----- None in the list -----

Remove

Illustration 110: Pushed Setting in Client-to-Site VPN

By doing that, the VPN clients will use the VPN server's IP in VPN as gateway for that subnet. And it also can ask the VPN clients to use specified WINS server or DNS server on the right of the screen shown above.

The CA, server key and certificate, and client keys and certificates can be established via “**Vpn >> Connection >> Key Generation**”.

Certificate and Key Generation Setup Wizard: Previous - Steps 3/4 - Next

Vpn >> Connection >> Key Generation

Country Code State Code
Locality Org. Name
Org. Unit Email

CA Generation :
Common Name
CA Certificate Expiration : Aug 2 03:19:41 2029 GMT

Cert. & Key for Server:
Common Name

Cert. & Key for Client(s):
Common Name
Valid days

Client Configuration Set List

client1:earth Aug 2 03:20:00 2029 GMT

Illustration 111: Client-to-Site VPN Certificate and Key Generation

To have a unique CA (Certificate Authority) for your VPN server, you might press “Purge” to remove all the pre-installed keys and certificates and generate new ones.

CA can be generated by filling the field “Common Name” and press the button “Generate”. After generating CA, you start generating the certificate and key for server. The server key generation will take a little long time. Once you have key and certificate, the client keys and certificates can be generated by entering “Common Name” and “Valid days” of certificate for each client. The field “Common Name” should be unique among all clients, server, and CA.

Once the server key and certificate are generated, you should restart the VPN server to use the new CA, key, and certificate. The client key and certificate can be created at any time after CA and server keys are generated. In the screen snapshot above, a client certificate with Common Name “earth” is created. The key and certificate for that client can be downloaded via “**Vpn >> Connection >> Client File Download**”:

Client Cert. and Key Download Setup Wizard: Previous - Steps 4/4

Vpn >> Connection >> Client File Download

Client Configuration Set List Client Program Download

client1:earth

Please press reload button if the New Cert. and Key are the very first one you have generated, or the Cert. and Key have been modified. Reload

☐ Unified Form Download (for OpenVPN Connect)

Per User Key Download

Illustration 112: Client Certificate Download for Client-to-Site VPN

By selecting the file set in the box and pressing the button below, the client key and certificate can be downloaded in this way. The admin people can send the file to the user(s) and ask them to install them on the VPN client program.

Site-to-site VPN Connection(Routing Mode)

Site-to-site VPN can be used when some hosts shall be open to the remote site across the Internet. It does not need every user on the remote site to install VPN client program on his/her computer or mobile devices while site-to-site VPN is deployed. Only the two hosts acting as VPN gateways shall have the VPN setting in the site-to-site VPN scenario. In this section, we introduce site-to-site VPN running in routing mode.

Site-to-site VPN is with its own CA; it is different from the CA used in client-to-site VPN. It needs to be created independently. Site-to-site VPN provided by the base platform will be using UDP port 7777, and the two VPN gateways will lock the other party's public IP address closely.


As we mentioned several times, the VPN provided here is to create a virtual network interface (also known as “local tunnel device”). If the VPN is running in routing mode, this virtual network interface is with an IP address. In client-to-site VPN, the IP addresses used by the clients and server in VPN are all from the address pool. For site-to-site, we only need two IP addresses for the “local tunnel devices” on both sides. Please note that the two IP addresses should not conflict with any IP addresses in the enterprise subnets.

The screenshot shows a web interface titled "Certificate and Key Generation" with a sub-header "Vpn >> Site-to-Site >> Keys". The interface is divided into several sections:

- Metadata Fields:** Country Code (NB), State Code (NA), Locality (here3), Org. Name (thisPlace), Org. Unit (IT), and Email (me@myhost.mydom). A "Submit" button is located below these fields.
- CA Generation:** A section with a "Common Name" input field and a "Generate" button.
- Cert. & Key to be used at Local:** A section with a "Common Name" input field and a "Generate" button.
- Cert. & Key to be used at Remote:** A section with a "Common Name" input field and a "Generate" button.
- Client Configuration Set List:** A large text area displaying "----- None in the list -----".
- Bottom Controls:** "Save", "Remove", and "Purge" buttons.

Illustration 113: Site-to-Site VPN Key Generation

The following screen snapshots from key generation process.
Site-to-site VPN's CA is generated as follows:

 **Certificate and Key Generation**

Vpn >> Site-to-Site >> Keys

Country Code

NB

State Code

NA

Locality

here3

Org. Name

thisPlace

Org. Unit

IT

Email

me@myhost.mydom

Submit

CA Generation :

Common Name

quark

Generate

Cert. & Key to be used at Local:

Common Name

Generate

Cert. & Key to be used at Remote:

Common Name

Generate

Client Configuration Set List

----- None in the list -----

Save

Remove

Purge

Illustration 114: Site-to-Site VPN CA Generation

The screenshot shows a web interface titled "Certificate and Key Generation" with a sub-header "Vpn >> Site-to-Site >> Keys". The interface is divided into several sections:

- Country and State Information:** Fields for "Country Code" (NB) and "State Code" (NA).
- Local Information:** Fields for "Locality" (here3), "Org. Name" (thisPlace), "Org. Unit" (IT), and "Email" (me@myhost.mydom).
- CA Generation:** A section for generating a Certificate Authority with a "Common Name" field (quark) and a "Generate" button.
- Local Certificate and Key:** A section for generating a local certificate and key with a "Common Name" field (neutrino) and a "Generate" button.
- Remote Certificate and Key:** A section for generating a remote certificate and key with a "Common Name" field and a "Generate" button.
- Client Configuration Set List:** A large text area displaying "----- None in the list -----".
- Buttons:** A "Submit" button is located below the local information fields. At the bottom right, there are "Save", "Remove", and "Purge" buttons.

Illustration 115: Site-to-Site VPN: Server Key and Certificate Generation

The server key and certificate are generated by giving "Common Name" and pressing "Generate" button.

To generate client key and certificate, fill in the field “Common Name” and press “Generate” button:

Certificate and Key Generation

Vpn >> Site-to-Site >> Keys

Country Code State Code
Locality Org. Name
Org. Unit Email

CA Generation :
Common Name


Cert. & Key to be used at Local:
Common Name

Cert. & Key to be used at Remote:
Common Name

Client Configuration Set List
----- None in the list -----

Illustration 116: Site-to-Site VPN: Client Key and Certificate Generation

Illustration 117: Site-to-site VPN: Sample for Key Generation

 **Certificate and Key Generation**

Vpn >> Site-to-Site >> Keys

Country Code

NB

State Code

NA

Locality

here3

Org. Name

thisPlace

Org. Unit

IT

Email

me@myhost.mydom

Submit

Cert. & Key to be used at Remote:

Common Name

Generate

CA Generation :

Common Name

quark

Generate

Cert. & Key to be used at Local:

Common Name

neutrino

Generate

Client Configuration Set List

client1:electron

Save

Remove

Purge

The client key and certificate generated on this side of the gateway should be submitted to the other side. It can be done by selecting the file and press “Save” button to download. And upload it to the remote machine on its corresponding menu at right-hand site of “**Vpn >> Site-to-Site >> Gateway Network Setting**”.

The following two screen snapshots are an example for the setting on two VPN gateways:

The screenshot shows a web interface titled "VPN Gateway Network Setting". Below the title is a breadcrumb trail: "Vpn >> Site-to-Site >> Gateway Network Setting". The interface is divided into two main sections. The left section contains several input fields for network configuration: "UDP Port for site-to-site VPN" (7777), "Local Public IP" (192.168.11.202), "Remote Public IP" (192.168.11.138), "Local tunnel device IP" (192.168.99.3), "Remote tunnel device IP" (192.168.99.1), "Remote LAN Address" (10.2.1.0), and "Remote LAN Netmask" (255.255.255.0). Below these fields is a checkbox labeled "Acting as TLS Server by using locally-generated CA and keys (otherwise, file upload will be needed)" which is checked. Underneath this checkbox is a label "Data Cipher:" followed by a dropdown menu showing "AES-128-CBC". The right section is titled "CA and Key File Set Upload (will replace current CA)" and contains a "Browse..." button, the text "No file selected.", an "Upload" button, and a "Submit" button. There is also a checkbox labeled "Start Site-to-Site VPN process" which is checked, with its own "Submit" button.

Illustration 118: VPN Gateway as TLS server

The host is with IP address “192.168.11.202” and the counterpart is with IP address “192.168.11.138”. The local tunnel device is with IP address “192.168.99.3” and the remote tunnel device is with IP address “192.168.99.1”. On the remote side, there is a subnet “10.2.1.0/255.255.255.0”. For the network packets to the subnet “10.2.1.0/24”, they should be routed via this VPN gateway.

The following is the setting on the remote host:

The screenshot shows a web interface titled "VPN Gateway Network Setting". Below the title is a breadcrumb trail: "Vpn >> Site-to-Site >> Gateway Network Setting". The interface contains several input fields and buttons. On the left, there are fields for "UDP Port for site-to-site VPN" (7777), "Local Public IP" (192.168.11.138), "Remote Public IP" (192.168.11.202), "Local tunnel device IP" (192.168.99.1), "Remote tunnel device IP" (192.168.99.3), "Remote LAN Address" (172.16.9.0), and "Remote LAN Netmask" (255.255.255.0). On the right, there is a section for "CA and Key File Set Upload (will replace current CA)" with a "Browse..." button and the text "No file selected.". Below this is an "Upload" button. There is a checkbox labeled "Start Site-to-Site VPN process" which is checked. At the bottom, there is a checkbox labeled "Acting as TLS Server by using locally-generated CA and keys (otherwise, file upload will be needed)" which is unchecked. There are two "Submit" buttons: one at the bottom right and one at the bottom center.

Field	Value
UDP Port for site-to-site VPN	7777
Local Public IP	192.168.11.138
Remote Public IP	192.168.11.202
Local tunnel device IP	192.168.99.1
Remote tunnel device IP	192.168.99.3
Remote LAN Address	172.16.9.0
Remote LAN Netmask	255.255.255.0

CA and Key File Set Upload (will replace current CA)

Browse... No file selected.

Upload

☒ Start Site-to-Site VPN process

☐ Acting as TLS Server by using locally-generated CA and keys (otherwise, file upload will be needed)

Submit

Illustration 119: VPN Gateway on the Remote Site

Similarly, this host is with IP address "192.168.11.138" and the site-to-site VPN connection is to "192.168.11.202". The local tunnel device is using IP address "192.168.99.1" whereas the remote tunnel device is with IP address "192.168.99.3". On the other end, it is with a subnet "172.16.9.0/24". For the traffic going to that subnet, it should be routed via this gateway.

Of course, this is the setup we use in the lab. For using them across the Internet, you should replace the IP addresses "192.168.11.202" and "192.168.11.138" with public IP addresses.

For this setting to function as expected that the hosts between the two subnet “172.16.9.0/24” and “10.2.1.0/24” can reach each other, their default gateway or the gateway to the remote subnet needs to be set to the IP address of the VPN gateway on the corresponding subnet.

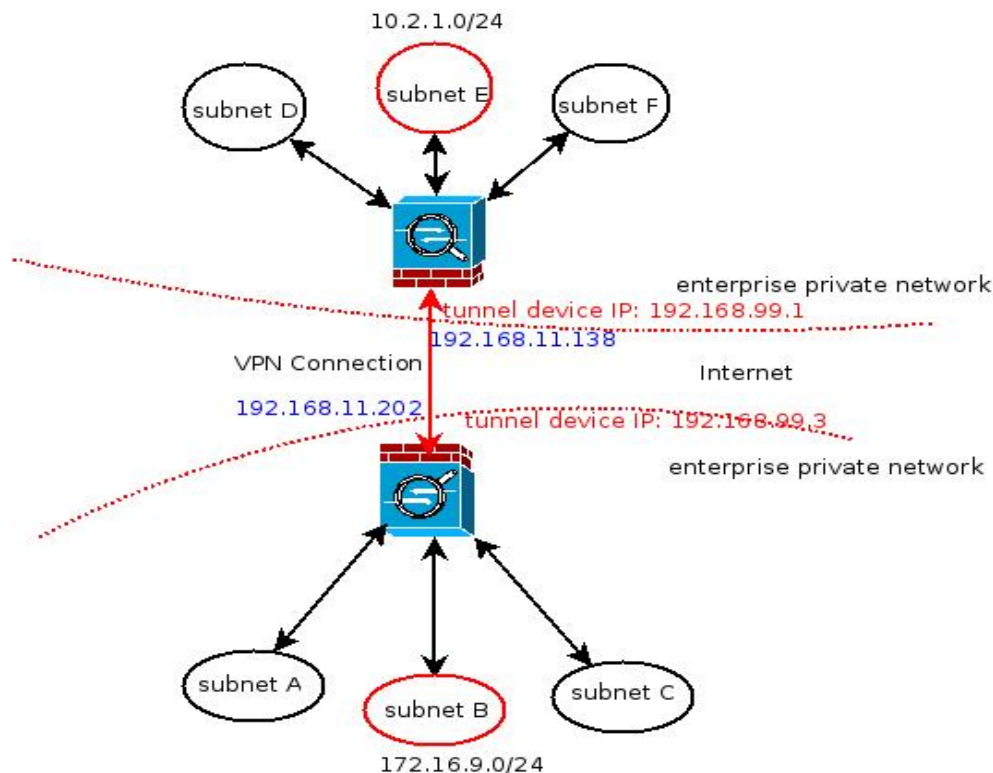



Illustration 120: Site-to-Site VPN Sample Setting

For example, the VPN gateway on the bottom side in the diagram is with IP address “172.16.9.1” connecting to the subnet “172.16.9.0/24”. For the hosts in that subnet, they should use the IP address “172.16.9.1” as a gateway to reach “10.2.1.0/24”.

As for the other subnets to reach the other side, the main routing table on VPN gateway should add the route to each remote subnet.

There might be chances to use multiple instances of site-to-site VPN on one machine. Other than the setting of UDP port, it is quite similar to the setting introduced above. For each instance, this machine has to be set up as “TLS server” and listen the requests on different UDP port.

 **Running Multiple Instances of Site-to-Site TLS Servers as a Multiplexer**

Vpn >> Site-to-Site >> Multiplexer

UDP Port

Local Public IP

Remote Public IP

Local tunnel device IP

Remote tunnel device IP

Remote LAN Address

Remote LAN Netmask

Data Cipher

AES-128-CBC

Add

<input type="checkbox"/>	UDP Port	Local Public IP	Remote Public IP	Local tunnel device IP	Remote tunnel device IP	Remote LAN Address	Remote LAN Netmask	Data Cipher
---- none ----								

Delete

Populate

Illustration 121: Screen Snapshot as Site-to-Site Multiplexer

The fields are the same as the ones for establishing site-to-site VPN. Please note that those multiple instances are sharing the same server key and certificate under “**Vpn >> Site-to-Site >> Keys**”. Thus, they have to be TLS server. While using multiple instances of site-to-site VPN, you can not have TLS server and client on the same machine. By doing so, the CA will be corrupted.

Site-to-site VPN in Bridging Mode

Site-to-site VPN in bridging mode is with drawback that many network traffic originally intended for local use will go across the Internet to reach the other side. For example, the messages like the traffic of DHCP clients looking for DHCP server, and the messages for looking for uPnP devices, shall only be populated within the IP subnet. But after using site-to-site VPN in bridging mode, the traffic will reach to the other side. It is not the fault of the network protocols themselves; the two sides of the network do work as one IP subnet by site-to-site VPN in bridging mode. Thus, once you decide to use it, you might need to rearrange the network equipments: for example, you should have only one DHCP server in an IP subnet; and investigate if there exists IP conflict when two parts of the network are joined together.

However, site-to-site VPN in bridging mode is also with its benefit: the IP multicast packets can go across VPN connection to the side. The IP multicast packets can not go across the Internet because most of the ISPs (Internet Service Providers) will refuse to route those packets. And the routing protocol OSPF also uses IP multicast for its Hello or Discovery messages. In some scenarios, site-to-site VPN in bridging mode can simplify those deployment issues.

We use the following diagram to explain the setting of site-to-site VPN in bridging mode. To emulate in the lab, two machines with IP addresses “192.168.11.202” and “192.168.11.138” are used. For the use on the Internet, they should be public IP addresses. What we are trying to do is: establish site-to-site VPN connection in bridging mode such that the subnet “172.16.9.0/24” will show up behind “192.168.11.202” and “192.168.11.138”.

As we mentioned earlier, the virtual network device created by VPN process in bridging mode does not have IP address on it. It just deals with Ethernet frames. To exchange Ethernet frames with other network interfaces, this network device created by VPN process has to join a bridge with other network interfaces.

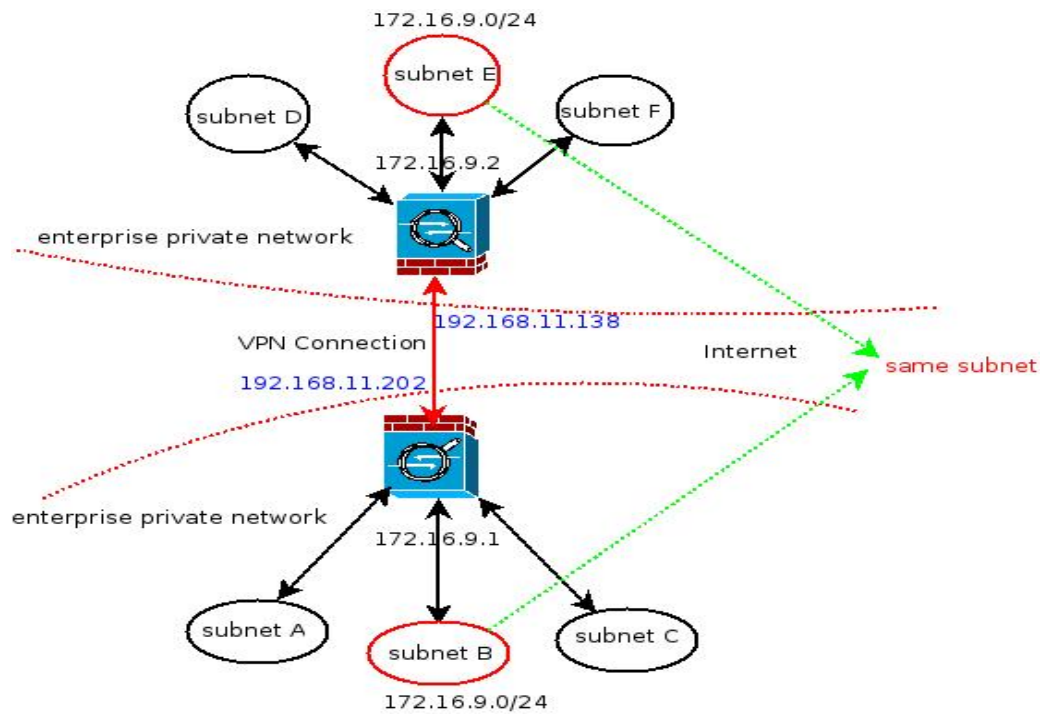


Illustration 122: Example for Site-to-Site VPN in Bridge Mode

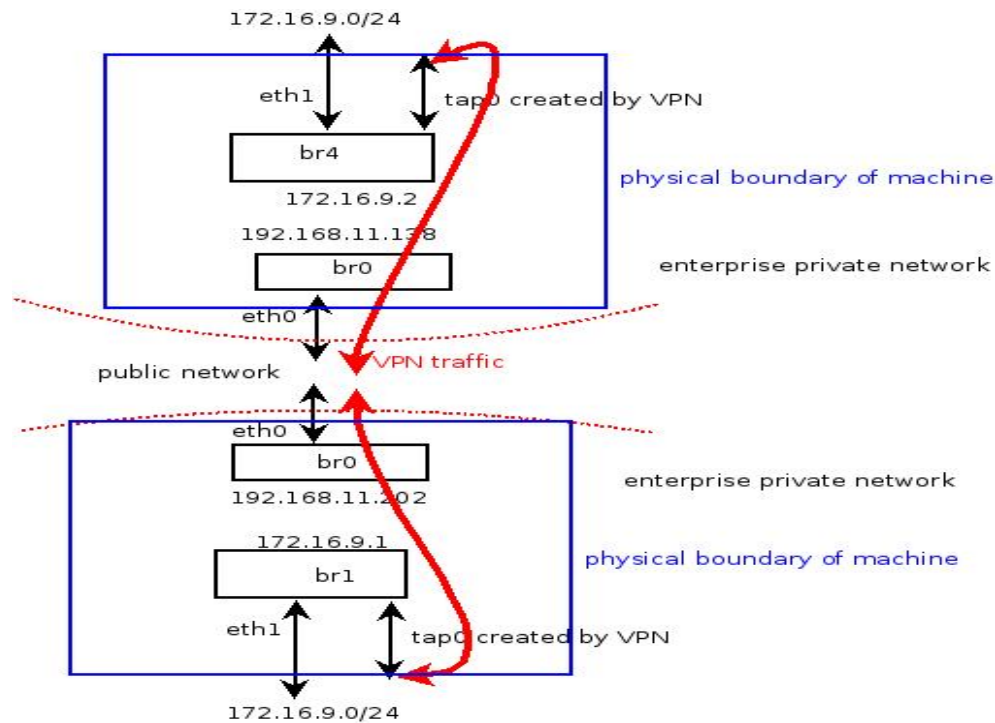


Illustration 123: Detailed Operation for Site-to-Site VPN in Bridge Mode

On the base platform, we have virtual bridges created in advance along with physical network interfaces. It does not need to create extra bridge for this operation. We use “tap0” for the network device created by VPN process on bridging mode. Thus, the idea is just to let “tap0” join the bridge on each site, and the VPN processes on two sides will bring the two bridges together under the same IP subnet. Since we give each bridge an IP address, these two bridges on the same subnet should be with different IP addresses set on them.

That is roughly the idea about the setting for site-to-site VPN in bridging mode. We start with creating CA, server key and certificate, and client key and certificate. Similarly, those are set independently from the client-to-site VPN and site-to-site VPN in routing mode. They can be created via **“Vpn >> Bridge >> Server/Client”**:

Certificate and Key Generation for VPN Bridging

Vpn >> Bridge >> Key Management

Country Code State Code
Locality Org. Name
Org. Unit Email

CA Generation :
Common Name

Cert. & Key to be used at Local:
Common Name


Cert. & Key to be used at Remote:
Common Name

Client Configuration Set List

client1:Big

Illustration 124: CA, Keys and Certificates for Site-to-Site VPN in Bridging Mode

We are doing this on the machine "192.168.11.202". Thus, the client key and certificate should be downloaded and submitted to the other machine. The CA and server key along with server certificate are on this machine so that we use this machine as "VPN Bridge Server".

 **VPN Bridge Server/Client**

Vpn >> Bridge >> Server/Client

☒ Acting as VPN Bridge Server (otherwise, fill the following items and upload certificate.)

CA and Key File Set Upload (will replace current CA)

Browse... No file selected.

Server UDP port: 1195

Server IP Address:

Data Cipher: AES-128-CBC

☒ Start VPN Bridging process(es)

Upload

Submit

AES-128-CBC

Submit


☐ Use 2nd VPN Bridge Server (tap1)

Server UDP port:

Data Cipher: AES-128-CBC

Submit

Illustration 125: Sample Setting for VPN Bridge Server

 **Ethernet / DHCP**

System >> Network >> Ethernet / DHCP

Ethernet Bridge (br1)

☒ Turn on DHCP Server

IP Address: 172.16.9.1

Netmask: 255.255.255.0

Start IP: 172.16.9.100

End IP: 172.16.9.200

Submit

☒ Enable Bridge br1

Ethernet Ports in Bridge br1:

eth1 tap0

Submit

Illustration 126: Example for VPN Network Device Joining a Bridge (Server Part)

And do not forget to let “tap0” join the virtual bridge “br1” with “eth1”.

On the other side (the machine “192.168.11.132”), we upload the client key and certificate on the menu of “**Vpn >> Bridge >> Server/Client**”:

The screenshot shows a web-based configuration interface for a VPN Bridge. The title is "VPN Bridge Server/Client" with a lightbulb icon. Below the title is a breadcrumb trail: "Vpn >> Bridge >> Server/Client". The interface is divided into two main sections. The left section contains a checkbox "Acting as VPN Bridge Server (otherwise, fill the following items and upload certificate.)". Below this are input fields for "Server UDP port" (value: 1195) and "Server IP Address" (value: 192.168.11.202), followed by a "Submit" button. There is also a checkbox "Use 2nd VPN Bridge Server (tap1)" with an empty "Server UDP port" input field and another "Submit" button. The right section is titled "CA and Key File Set Upload (will replace current CA)". It includes a "Browse..." button, the text "No file selected.", an "Upload" button, and a checked checkbox "Start VPN Bridging process(es)" with a "Submit" button.

Illustration 127: Sample Setting on the Client of Site-to-Site VPN in Bridging Mode

After submitted the client key and certificate, the screen on “**Vpn >> Bridge >> Key Management**” will be shown as follows:

Certificate and Key Generation for VPN Bridging

Vpn >> Bridge >> Key Management

Country Code State Code

Locality Org. Name

Org. Unit Email

Cert. & Key to be used at Remote:

Common Name

Client Configuration Set List

client0:Big

CA Generation :

Common Name

Cert. & Key to be used at Local:

Common Name

Illustration 128: Certificate Display on Client Side

And do not forget to let “tap0” join the bridge:

Ethernet Bridge (br4)

☐ Turn on DHCP Server

IP Address: Netmask:

Start IP: End IP:

☒ Enable Bridge br4

Ethernet Ports in Bridge br4:

Illustration 129: Sample for VPN Network Device to Join a Bridge (Client Part)

The VPN connection in the example above is using UDP port 1195. Thus, do not forget to open the access on border control. The VPN connection will take effect after rebooting the two machines.

Please note that the network devices created for virtual hosts are also label as “tapN”. Since they are created dynamically, sometimes “tap0” will be owned by a virtual host when “site-to-site VPN in bridging mode” is not in use. When we put the site-to-site VPN in bridging mode into use, please always reboot the system to bring up VPN. And then, bring up virtual hosts one by one.

Chapter 5 Dynamic Routing

“Routing” here in this document refers to “IP Routing”. Theoretically, selecting a path in network level is called “routing” (for example, telephone network or IP network). But we focus on “IP network” here. We start with the introduction for “Static Routing” and “Dynamic Routing” on the base platform.

“Static Routing” implies we configure the routing table by adding entries manually. For the subnet that is directly connected by the network interface with IP address, the corresponding routing entries should be added into routing table while setting the IP address. Usually when we refer to “adding routing entry”, it is the work of specifying the gateway to a specific IP subnet. Setting the “default gateway” is kind of doing “static routing”. If there is no matched subnet for the destination of a network packet, it goes to the default gateway.

“Static Routing” is useful for small network, and it is easy to debug. But for large network, it is default to add routing entries one by one manually. “Dynamic Routing” is trying to resolve this issue by constructing the routing table automatically. It is usually done by exchanging information with the other routers and build the routing table accordingly.

For the base platform, it is usually with 12 IP subnets. If it is enough to have these 12 subnets without any other subnets, it suffices to use “static routing”. However, if there exists other routers, it might be troublesome to add routing entries on each router.

There exist other scenarios that can not be resolved by simple “static routing”. For example, on the environment of using IP multicast, each host is with its own IP address for “unicast”. For some of the machines to receive IP multicast packets, they have to join the group address for IP multicast. In other words, those machines are expecting the packets with destinations of their own IP addresses for “unicast” and group address for multicast. And the group addresses of IP multicast (224.0.0.0~239.255.255.255) can appear in any subnet. Unless you would like to confine those hosts within one subnet to use IP multicast, it can not be done by using “static routing”.

On the base platform, we provide RIPv2 (Routing Information Protocol , version 2) and OSPF for IP unicast (IPv4). PIM (Protocol Independent Multicast) will be use for multicast routing across the subnets.

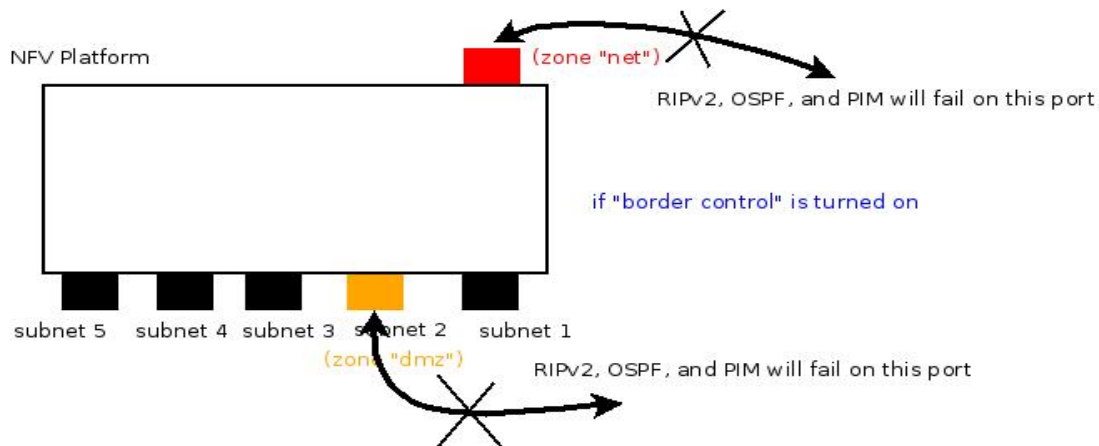


Illustration 130: Border Control and Dynamic Routing

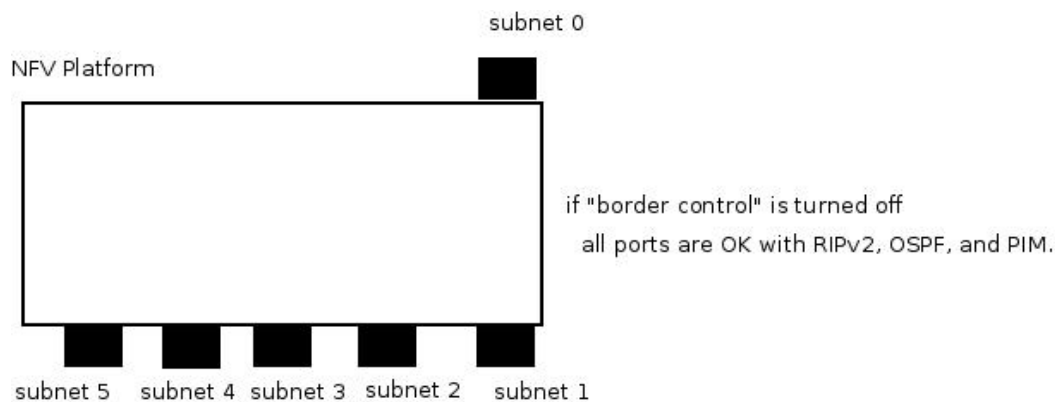


Illustration 131: Turn Off Border Control

The diagrams above generic network operations on the base platform associated with border control. It can be **turned on** or **turn off** via the setting at the top of "**Border >> Connection >> Port Forwarding**". Please recall the relationship between the zones: usually, the traffic from zone "**net**" to zone "**fw**" is dropped, and the traffic from zone "**dmz**" to zone "**fw**" is also dropped. Thus, RIPv2 and OSPF will fail on the two zones while border control is turned on. PIM will also fail because the group address membership can not be received properly. Please keep this in mind and use dynamic routing properly.

For RIPv2 and OSPF (OSPF v2 is for IPv4; OSPF v3 is for IPv6) to work properly, both of them are using IP multicast to find the other routers whereas RIP version 1 is using broadcast to do that. For IP multicast to function within a subnet, the network equipments on that subnet shall support IGMP.

OSPF and RIP are used between routers; PIM is used for multicast packets to cross an IP subnet.

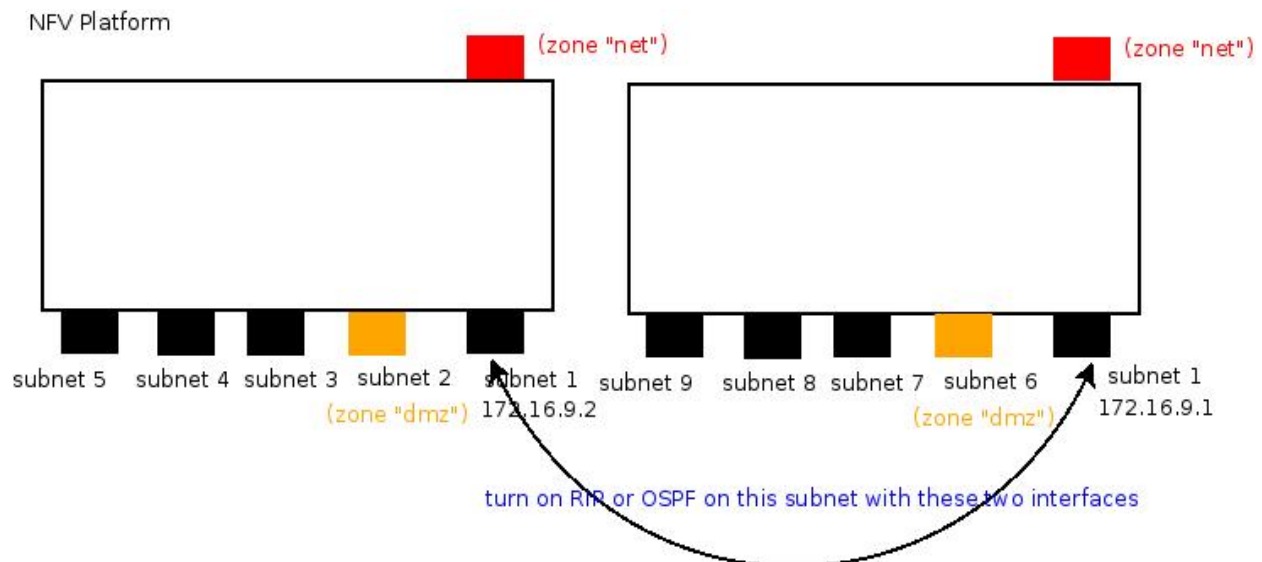


Illustration 132: Cascade two Machines together

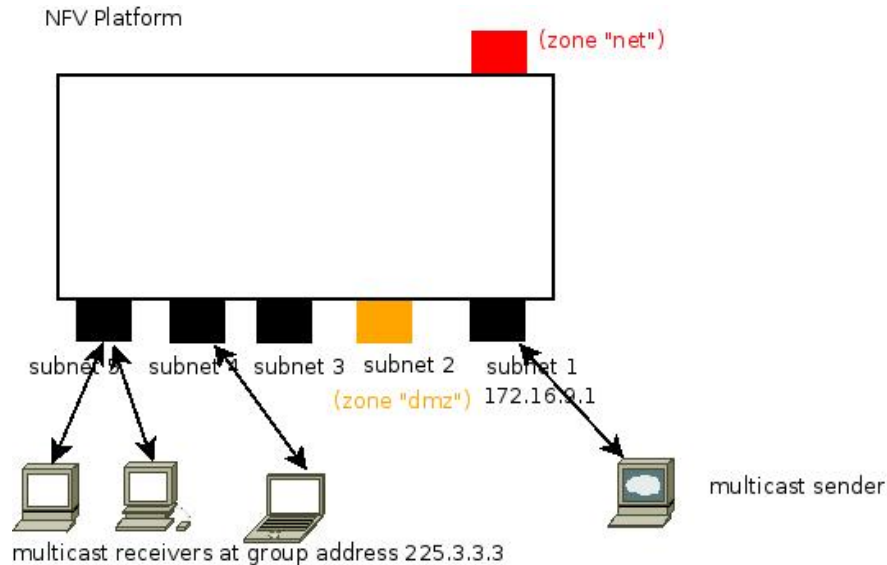


Illustration 133: Scenario to use PIM

OSPF is with the concept to set “**area ID**”. The “Area 0” is used for core network; the values can range from 0 to $2^{32}-1$ or use the form a.b.c.d (for example, 1.1.1.1). In OSPF, the router sitting between two areas with one interface in area 0 is known as ABR (Area Border Router).

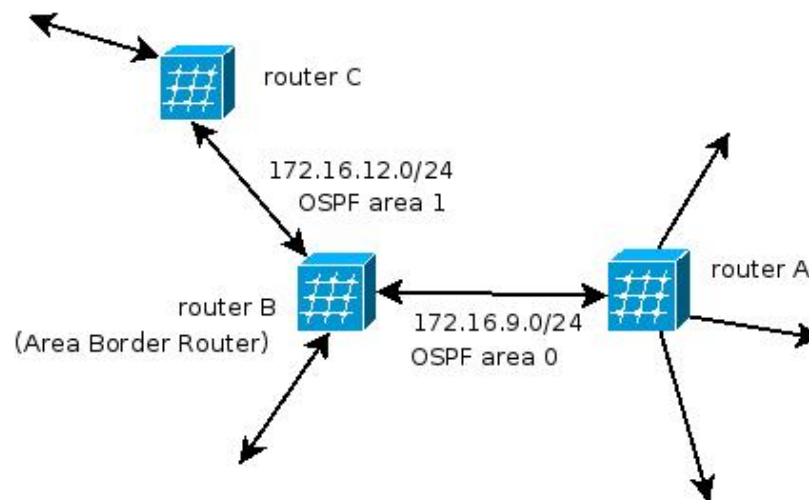


Illustration 134: OSPF ABR (Area Border Router)

RIPv2 (Router Information Protocol, version 2)

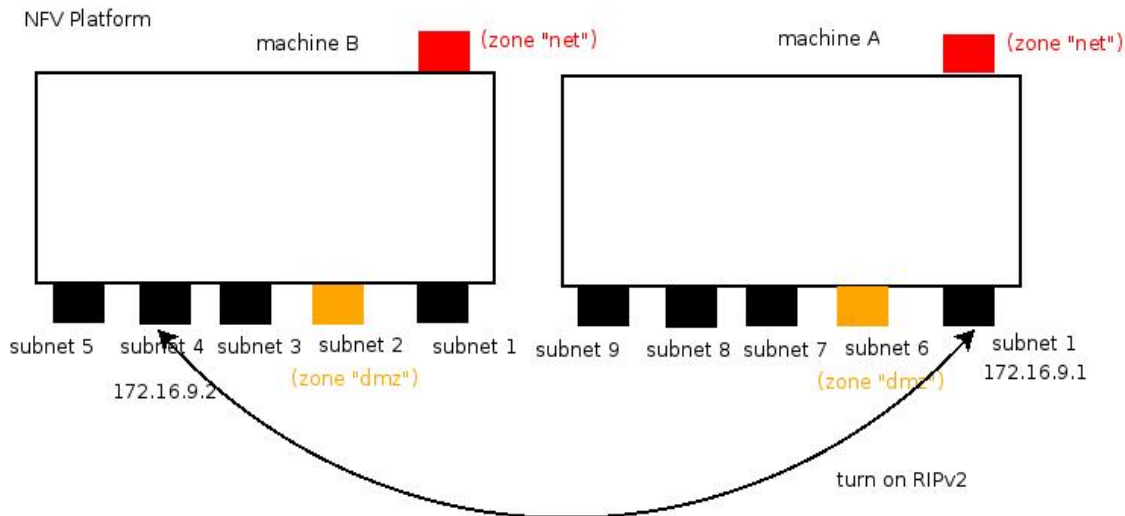


Illustration 135: Example for RIPv2

We would like to connect two machines together by using RIPv2 to exchange the routing tables so that we do not need to add route entry one by one. One of the physical ports of machine A will be connected to the port of machine B as shown above: on machine A, that physical port is placed under “br0” whereas it is placed under “br4” on machine B. The two ports from the two machines are joining to the same subnet with the IP addresses “172.16.9.1” and “172.16.9.2” respectively.

Once the two ports are connected, we set up as follows: at machine A, browse via “**System >> Network >> RIP**” to specify the subnet for sending updates.

RIP v2 (Routing Information Protocol v2)

System >> Network >> RIP

Add Network for Multicasting Route Update

Network: 172.16.9.0

Netmask Length: 24

☐ Start RIP

Network to send multicast update

-----none-----

Illustration 136: Subnet to Send Out Multicast Update for RIPv2

RIP v2 (Routing Information Protocol v2)

System >> Network >> RIP

Add Network for Multicasting Route Update

Network:

Netmask Length:

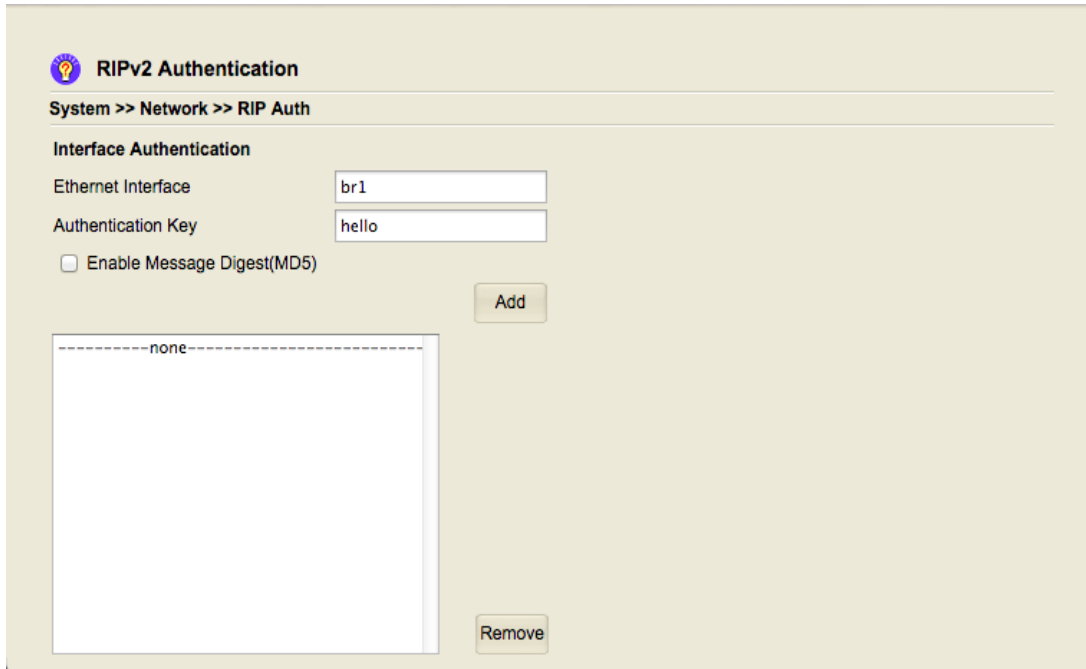
☐ Start RIP

Network to send multicast update

172.16.9.0/24

Illustration 137: List of the Subnet(s) to Send Multicast Update

To allow exchanging information with each other, an authentication key needs to be set up for the interface we are trying to send out update. It can be done via **“System >> Network >> RIP Auth”**:



RIPv2 Authentication

System >> Network >> RIP Auth

Interface Authentication

Ethernet Interface

Authentication Key

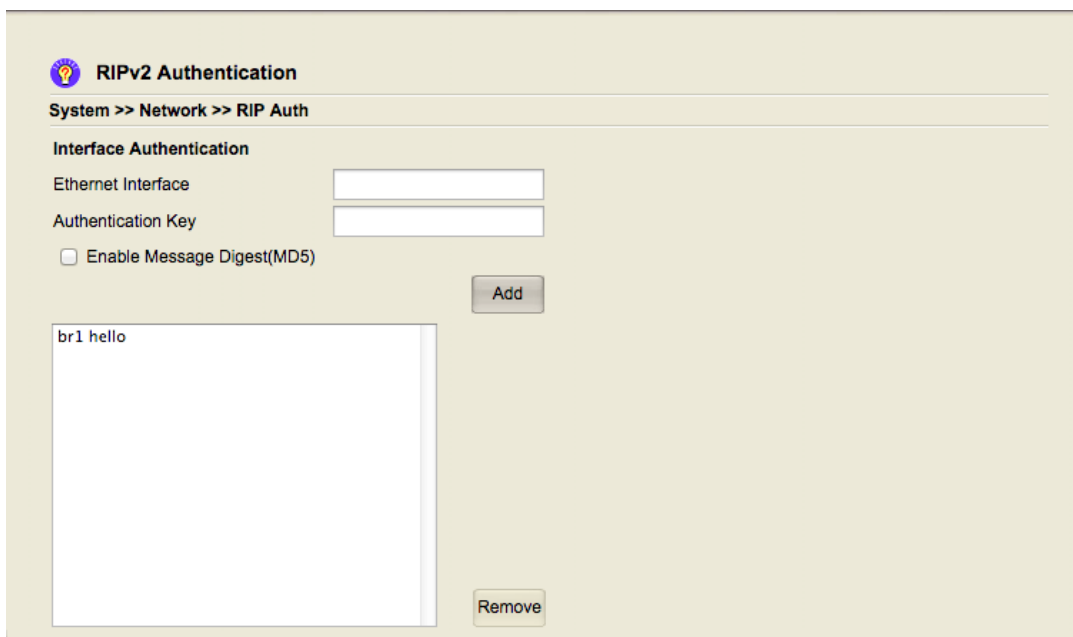
☐ Enable Message Digest(MD5)

Add

-----none-----

Remove

Illustration 138: Set up Authentication Key for RIPv2



RIPv2 Authentication

System >> Network >> RIP Auth

Interface Authentication

Ethernet Interface

Authentication Key

☐ Enable Message Digest(MD5)

Add

br1 hello

Remove

Illustration 139: List of Authentication Key(s)

On the other side (machine B), we also have the similar configuration. We set up the subnet and authentication key as follows:

The screenshot displays the 'RIP v2 (Routing Information Protocol v2)' configuration window. The breadcrumb path is 'System >> Network >> RIP'. Under the heading 'Add Network for Multicasting Route Update', there are input fields for 'Network' and 'Netmask Length', followed by an 'Add' button. A 'Start RIP' checkbox is also present, along with a 'Submit' button. Below this, the 'Network to send multicast update' section contains a list box with the entry '172.16.9.0/24' and a 'Remove' button at the bottom right.

Illustration 140: Subnet for Multicast Update on another machine

RIPv2 Authentication

System >> Network >> RIP Auth

Interface Authentication

Ethernet Interface

Authentication Key

☐ Enable Message Digest(MD5)

Add

br4 hello

Remove

Illustration 141: List of the Authentication Key

The authentication key is set up on “br4”. Then we start up the routing process by selecting the check box on “**System >> Network >> RIP**”. By looking at the routing table at “**System >> Network >> Static Routing**”, we will find out the subnets of the other side are in the list of the routing table. The following is what is known as “machine A”:

Active Route List:

10.2.15.0/24 via 172.16.9.2 dev br1 proto zebra metric 20
10.2.16.0/24 via 172.16.9.2 dev br1 proto zebra metric 20
10.2.17.0/24 via 172.16.9.2 dev br1 proto zebra metric 20
10.2.18.0/24 via 172.16.9.2 dev br1 proto zebra metric 20
10.2.19.0/24 via 172.16.9.2 dev br1 proto zebra metric 20
10.2.20.0/24 via 172.16.9.2 dev br1 proto zebra metric 20
172.16.9.0/24 dev br1 proto kernel scope link src 172.16.9.1
172.16.11.0/24 dev br2 proto kernel scope link src 172.16.11.1
172.16.12.0/24 dev br3 proto kernel scope link src 172.16.12.253 linkdown

Illustration 142: Content of Routing Table after Starting RIPv2

OSPF(Open Shortest Path First)

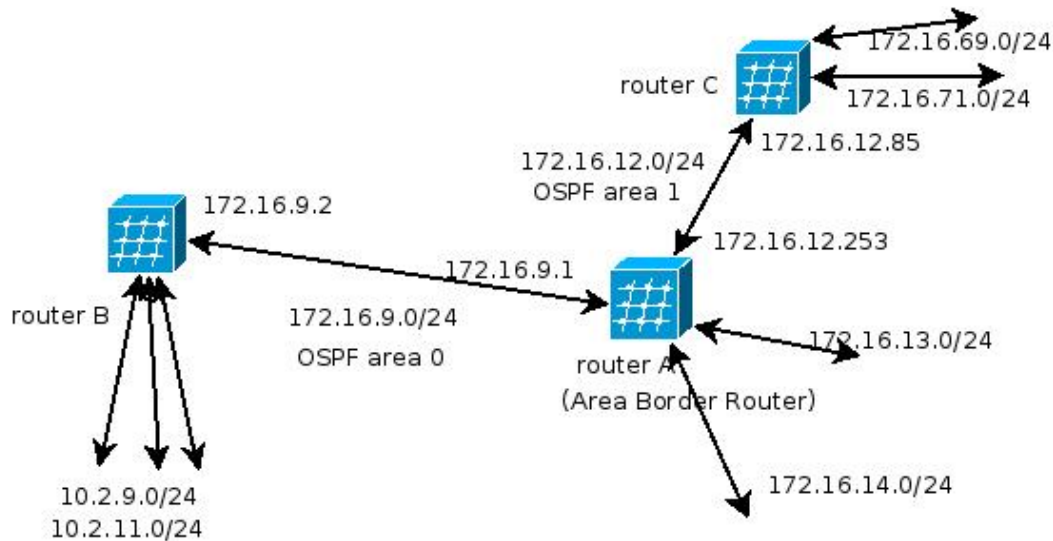


Illustration 143: OSPF Setup Example

We use the diagram above as an example to set up OSPF routing. There are three instances of our base platforms to be used as OSPF routers, designated as “router A”, “router B”, and “router C”. The IP subnets under each router are listed as follows:

router A: 172.16.9.0/24, 172.16.11.0/24, 172.16.12.0/24, 172.16.13.0/24, 172.16.14.0/24, 172.16.15.0/24, 172.16.16.0/24, 172.16.17.0/24, 172.16.18.0/24, 172.16.19.0/24, 172.16.20.0/24

router B: 10.2.9.0/24, 10.2.11.0/24, 10.2.12.0/24, 172.16.9.0/24, 10.2.14.0/24, 10.2.15.0/24, 10.2.16.0/24, 10.2.17.0/24, 10.2.18.0/24, 10.2.19.0/24, 10.2.20.0/24

router C: 172.16.12.0/24, 172.16.69.0/24, 172.16.17.0/24, 172.16.72.0/24, 172.16.73.0/24, 172.16.74.0/24, 172.16.75.0/24, 172.16.76.0/24, 172.16.77.0/24, 172.16.78.0/24, 172.16.79.0/24, 172.16.80.0/24

And router A and router B are adjacent to “172.16.9.0/24” that it will be designated as “area 0”; router A and router C are adjacent to “172.16.12.0/24” that it will be designated as “area 1”. The area ID can be set from $0 \sim 2^{32}-1$ in OSPF. The number “0” stands for core network. Furthermore, as we indicate previously, please do not use the subnet of zone “net” and the subnet of “dmz” to do OSPF updates; OSPF update will not work on the two zones.

The screenshot shows a web-based configuration interface for OSPF (Open Shortest Path First Protocol). The breadcrumb trail at the top reads "System >> Network >> OSPF". The main section is titled "Classify Network with Area ID" and includes a "Start OSPF" checkbox, which is currently unchecked. Below this, there are three input fields: "Network" with the value "172.16.9.0", "Netmask Length" with the value "24", and "Area ID (number or a.b.c.d)" with the value "0". A "Submit" button is located to the right of the "Start OSPF" checkbox, and an "Add" button is positioned below the "Area ID" field. At the bottom of the form, there is a section titled "Listing Area(s) and the associated Networks" which contains a large, empty text area. A "Remove" button is located at the bottom right of the interface.

Illustration 144: Subnet and Area ID Setup (Router A)

On “router A”, it needs to do OSPF updates on the two subnets “172.16.9.0/24” and “172.16.12.0/24”. Thus, we specify them in “**System >> Network >> OSPF**” and put the “Area ID” respectively.

OSPF (Open Shortest Path First Protocol)

System >> Network >> OSPF

Classify Network with Area ID

Network

Netmask Length

Area ID (number or a.b.c.d)

☐ Start OSPF

Listing Area(s) and the associated Networks

```
network 172.16.9.0/24 area 0
network 172.16.12.0/24 area 1
```

Illustration 145: List of Subnets (Router A)

The OSPF routers will authenticate with each other to check if the routing information can be populated or accepted from the remote machines. On each interface, we need an “Authentication key” and it has to match the setting on the other end. If “Area Authentication” is required, just enter the associated “Area ID” and press “Add”.

OSPF Authentication

System >> Network >> OSPF Auth

Interface Authentication

Ethernet Interface

Authentication Key

☐ Enable Message Digest(MD5)

Enable Area Authentication

Area ID

☐ Enable Message Digest(MD5)

Listing Interface Auth Keys

br3 dafa
br1 hello

Listing Area Auth Keys


area 0 authentication
area 1 authentication

Illustration 146: Authentication Setting of OSPF (Router A)

Please remember that IP addresses are set on the “bridge devices” (“br0”, “br1”, ... “br11”) on the base platform. Those interfaces should be used as “Ethernet interface” here. It goes without saying that the IP addresses of those interfaces should be “unique” among those routers. Some of them will be used as the “router ID” of each router; “router ID” will be used to compute the routing path. Even you think some of the subnets are not in use, you should keep each network interface with unique IP address.

Once those settings are done, we can go back to “**System >> Network >> OSPF**” to start OSPF process.

The following diagrams are the setting on “router B”:

 **OSPF (Open Shortest Path First Protocol)**

System >> Network >> OSPF

Classify Network with Area ID

Network

Netmask Length

Area ID (number or a.b.c.d)

☐ Start OSPF

Submit

Add

Listing Area(s) and the associated Networks

network 172.16.9.0/24 area 0

Remove

Illustration 147: Subnet Setting (Router B)

Please note that the authentication key should match the one set on “router A”.

The screenshot shows the 'OSPF Authentication' configuration page. At the top, there is a breadcrumb trail: 'System >> Network >> OSPF Auth'. The page is divided into two main sections: 'Interface Authentication' and 'Enable Area Authentication'. In the 'Interface Authentication' section, there are input fields for 'Ethernet Interface' and 'Authentication Key', and a checkbox for 'Enable Message Digest(MD5)'. In the 'Enable Area Authentication' section, there is an input field for 'Area ID' and a checkbox for 'Enable Message Digest(MD5)'. Below these sections are two lists. The 'Listing Interface Auth Keys' list contains one entry: 'br4 hello'. The 'Listing Area Auth Keys' list contains one entry: 'area 0 authentication'. Each list has an 'Add' button above it and a 'Remove' button below it.

OSPF Authentication

System >> Network >> OSPF Auth

Interface Authentication

Ethernet Interface

Authentication Key

☐ Enable Message Digest(MD5)

Enable Area Authentication

Area ID

☐ Enable Message Digest(MD5)

Listing Interface Auth Keys

br4 hello

Listing Area Auth Keys

area 0 authentication

Illustration 148: Authentication Setting (Router B)

After turning on OSPF and “router A” and “router B”, we check the routing table on “router A”. The subnets on “router B” are populated to “router A”:

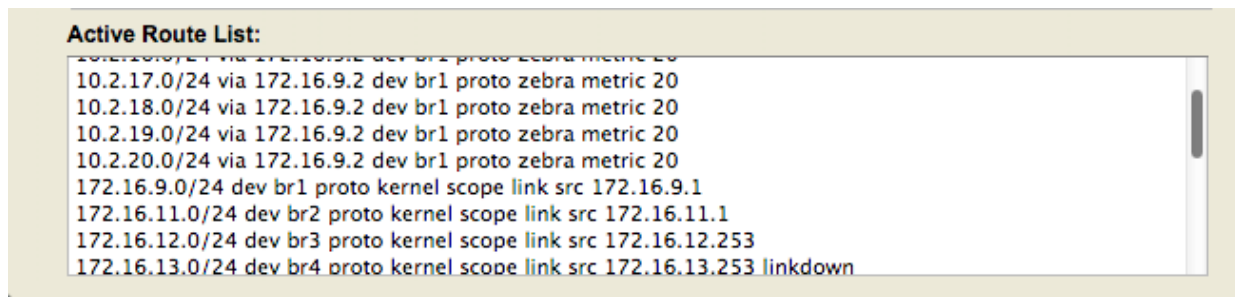


Illustration 149: Routing Table on Router A so far (After turning on OSPF)

Then we go on to set up “router C”:

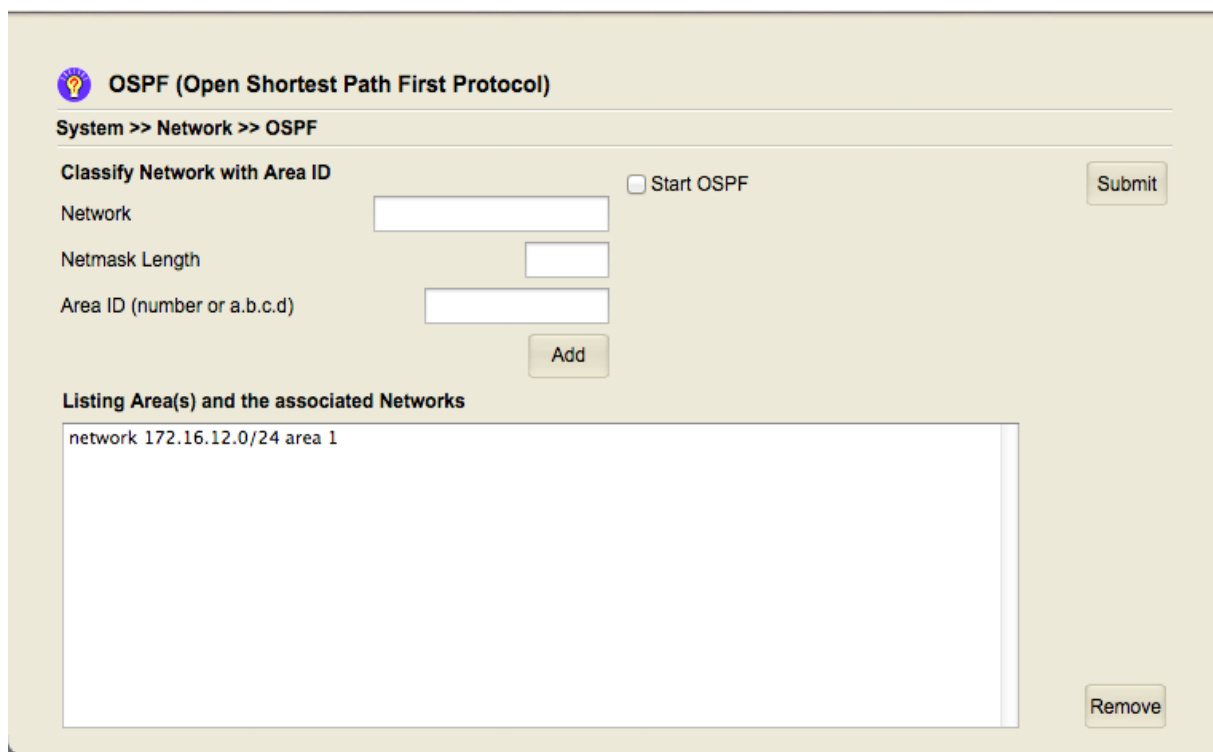



Illustration 150: Subnet Setting (Router C)

 **OSPF Authentication**

System >> Network >> OSPF Auth

Interface Authentication
Ethernet Interface
Authentication Key
☐ Enable Message Digest(MD5)

Enable Area Authentication
Area ID
☐ Enable Message Digest(MD5)

Add

Add

Listing Interface Auth Keys
br0 dafa

area 1 authentication

Remove

Remove

Illustration 151: Authentication Setting (Router C)

Similarly, the authentication key here should match the one set on “router A”. After starting OSPF on “router C”, the routing table on “router C” is shown below:

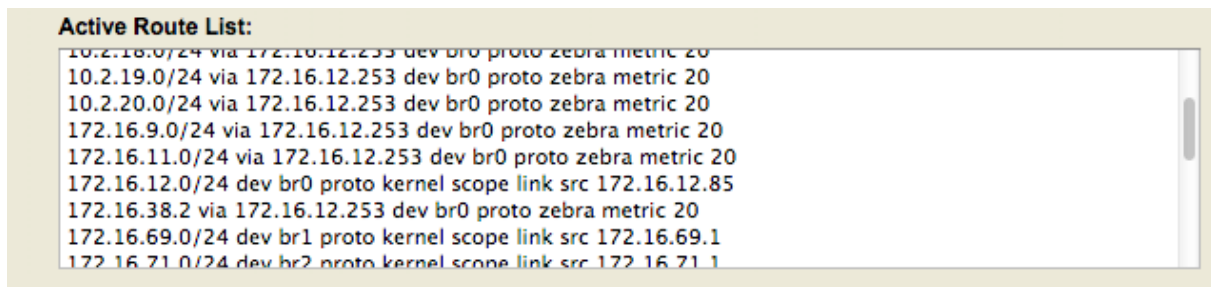


Illustration 152: Router Table (Router C)

If the link to a subnet is down, the corresponding routing entries in the routing table will not be populated to other routers. In some cases, you might check why the associated link is down.

PIM(Protocol Independent Multicast)

To send IP multicast packets and receive them in the same subnet, it only needs all the network equipments in that subnet supports IGMP (Internet Group Management Protocol). For the multicast packets to reach the other subnets, the routers need to support DVMRP(Distance Vector Multicast Routing Protocol), MOSPF (Multicast Open Shortest Path First), or PIM. In our base platform to support multicast routing, we use PIM. For details about multicast, please refer to IETF RFC 1112.

PIM is used to achieve the scenario that one sender sends out the IP multicast packets and expect the receivers on the other subnets can receive them. If the receivers are in the same subnet, PIM is not needed; it only needs IGMP for the case that sender and receivers are in the same subnet.

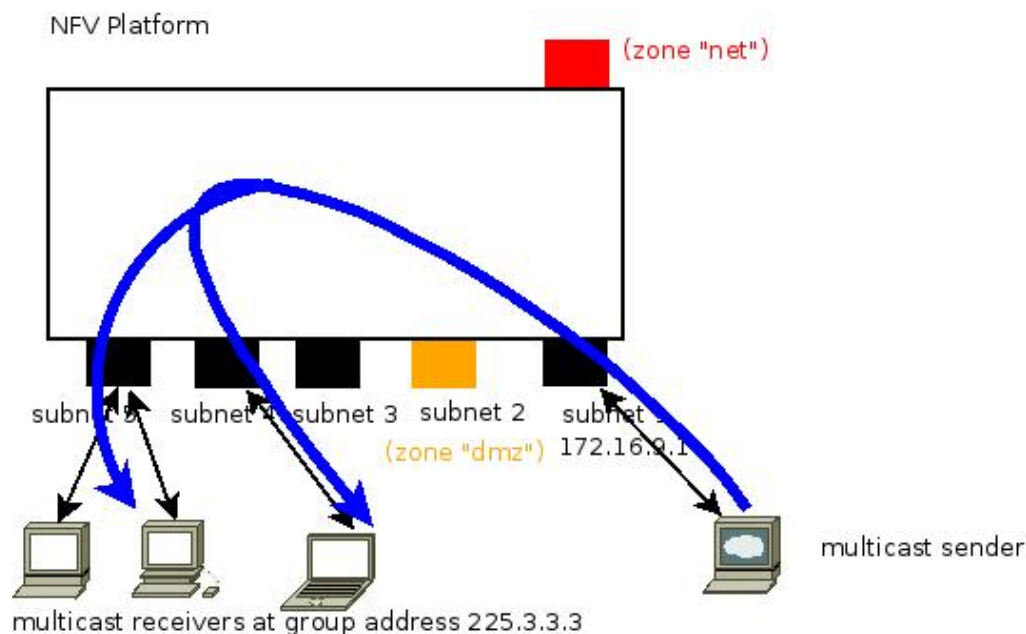



Illustration 153: Scenario to use IGMP and PIM

And in our base platform, you also need to consider the impact of the policies between the zones. Please remember that the packets from zone "net" and zone "dmz" will be dropped by default.

For the scenario shown above to happen, it only needs to turn on PIM at **"System >> Network >> Multicast Routing"**:

The screenshot shows a web interface for configuring Dynamic Multicast Routing (PIM v1 and v2). The page has a light beige background. At the top left, there is a lightbulb icon and the title "Dynamic Multicast Routing (PIM v1 and v2)". Below the title is a breadcrumb trail: "System >> Network >> Multicast Routing". The main configuration area contains several input fields and buttons. On the left, there are three rows of input fields: "Additional Network:" followed by a text box, "Netmask Length:" followed by a text box, and "Interface:" followed by a text box. To the right of the "Additional Network:" field is a checkbox labeled "Start PIM". To the right of the "Interface:" field is an "Add" button. At the bottom right of the configuration area is a "Submit" button. Below the input fields is a section titled "Alternate network List :". Inside this section is a large white rectangular area with the text "No alternate route added". To the right of this area is a "Remove" button. At the bottom of the page is a section titled "Information Listing:" followed by a large empty white rectangular area.

Illustration 154: Setting for Multicast Routing

 **Dynamic Multicast Routing (PIM v1 and v2)**

System >> Network >> Multicast Routing

Additional Network:

☒ Start PIM

Submit

Netmask Length:

Interface:

Add

Alternate network List :

No alternate route added

Remove

Information Listing:

Installing br0 (192.168.11.202 on subnet 192.168.11) as vif #0-22 - rate 0

Installing br1 (172.16.9.1 on subnet 172.16.9/24) as vif #1-23 - rate 0

Installing br2 (172.16.11.1 on subnet 172.16.11/24) as vif #2-24 - rate 0

Installing br3 (172.16.12.253 on subnet 172.16.12/24) as vif #3-25 - rate 0

Installing br4 (172.16.13.253 on subnet 172.16.13/24) as vif #4-26 - rate 0

Installing br5 (172.16.14.253 on subnet 172.16.14/24) as vif #5-27 - rate 0

Installing br6 (172.16.15.253 on subnet 172.16.15/24) as vif #6-28 - rate 0

Installing br7 (172.16.16.253 on subnet 172.16.16/24) as vif #7-29 - rate 0

Illustration 155: After PIM is turned on

Only when the sender is located in the subnet that is not directly-linked to the router should you specify the subnet and the associated interface that multicast packets arrive. Otherwise, you do not need the other setting; just turn on PIM. It automatically sets up the group-specific Rendez-vous point (RP).

IP multicast is based on UDP so that it is not like TCP to have reliable transmission by knowing the arrival of the packets at the other end. Thus, to confirm if the PIM functions, you might check if the UDP packets are dropped due to congested network when the multicast packets do not arrive at the expected subnet.

IP multicast is with the benefit that the sender only sends the message once and multiple receivers can receive the message. However, to prevent IP multicast traveling too far away, you might ask PIM not send out multicast packets to a specific network interface other than restrict the subnet of the sender. It can be done via “**System >> Network >> Multicast Control**”:

The screenshot shows a web-based configuration interface titled "Multicast Interface Control". Below the title is a breadcrumb trail: "System >> Network >> Multicast Control". The interface is divided into two main sections. The left section, titled "Ethernet Interface to avoid Direct Multicast from PIM", contains an "Interface:" label, a text input field, and an "Add" button. Below this is a list box labeled "Interface Listing :" which currently shows "-----none-----". At the bottom of this section is a "Remove" button. The right section, titled "Add Static Rendez-vous Point", contains an "IP Address:" label, a text input field, and an "Add" button. Below this is a list box labeled "Static Rendez-vous Point Listing :" which also shows "-----none-----". At the bottom of this section is a "Remove" button.

Illustration 156: Multicast Control

The Rendez-vous Point for a specific group address can also be set up statically. But this is not needed when we only have one machine (base platform).

Routing Across VPN

In this section, we consider to populate the routing entries across VPN. If possible, we consider to use RIPv2 or OSPF to do dynamic routing. If two routers can exchange routing entries either dynamically or statically, at least they have to have the access to the same IP subnet. Based on this idea, we look at a diagram we have and modify it a little bit:

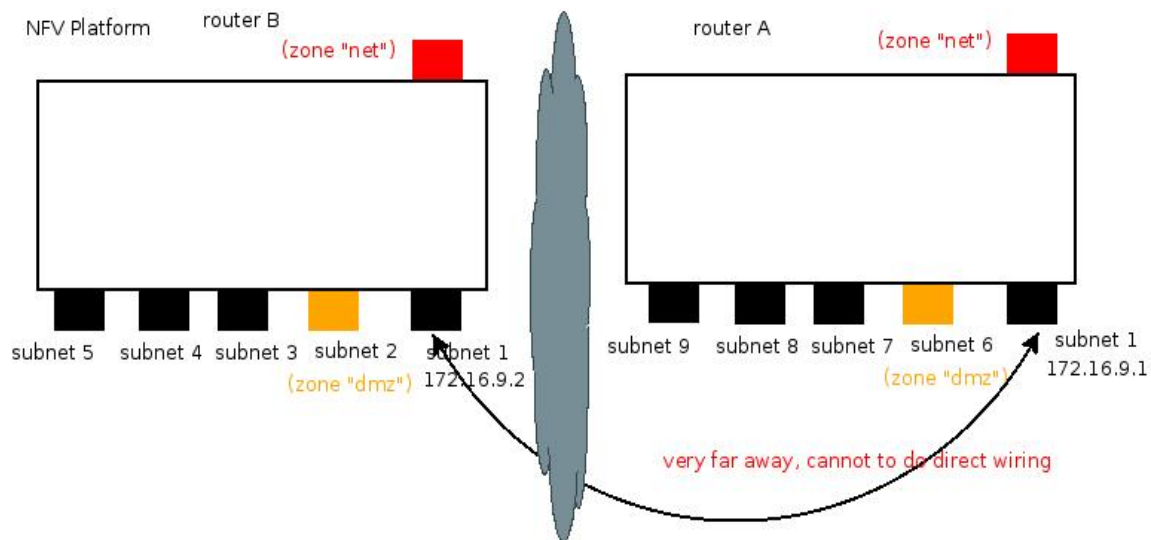


Illustration 157: Two Routers Across Internet

In the previous example, we can establish the direct wiring (via Ethernet cables or switches) between the two physical ports of the two routers. However, the two routers are at different sites across the Internet. It is impossible to establish the direct link on the two ports. Therefore, OSPF or RIPv2 is impossible too.

If the physical link is impossible, can we establish virtual link to bridge the two ports? We rearrange the diagram again. OSPF or RIPv2 can run on the same subnet if the two ports of the routers can be bridged.

Hence, the problem turns out to be: establishing site-to-site VPN in bridge mode. Of course, the VPN connection has to go through zone "net" to reach the other side.

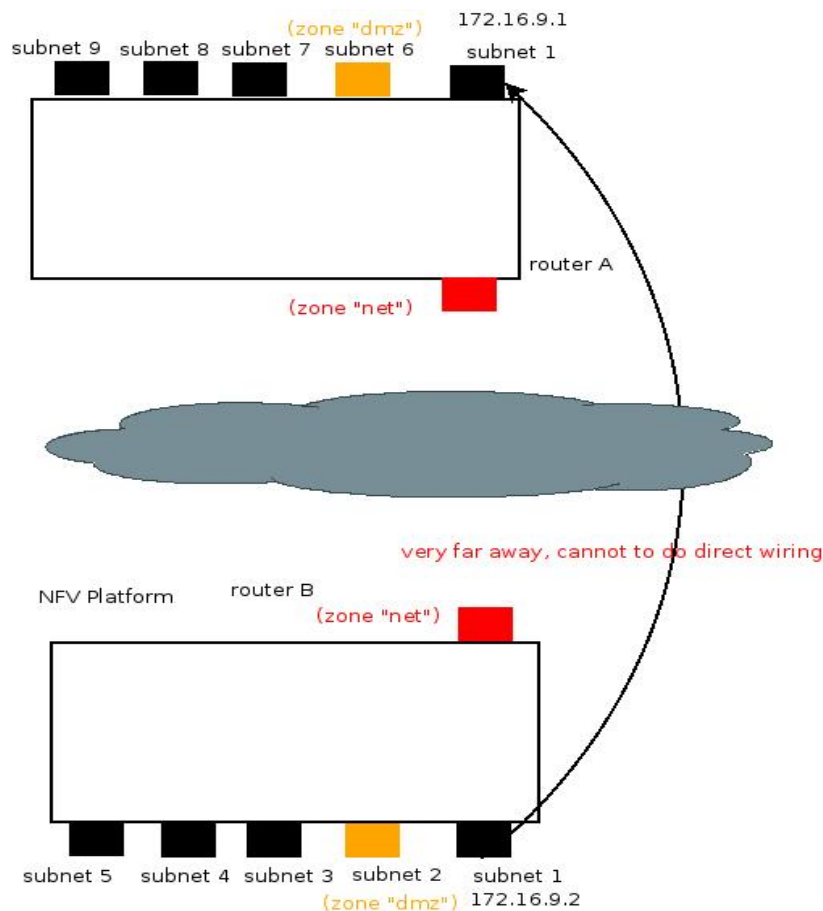
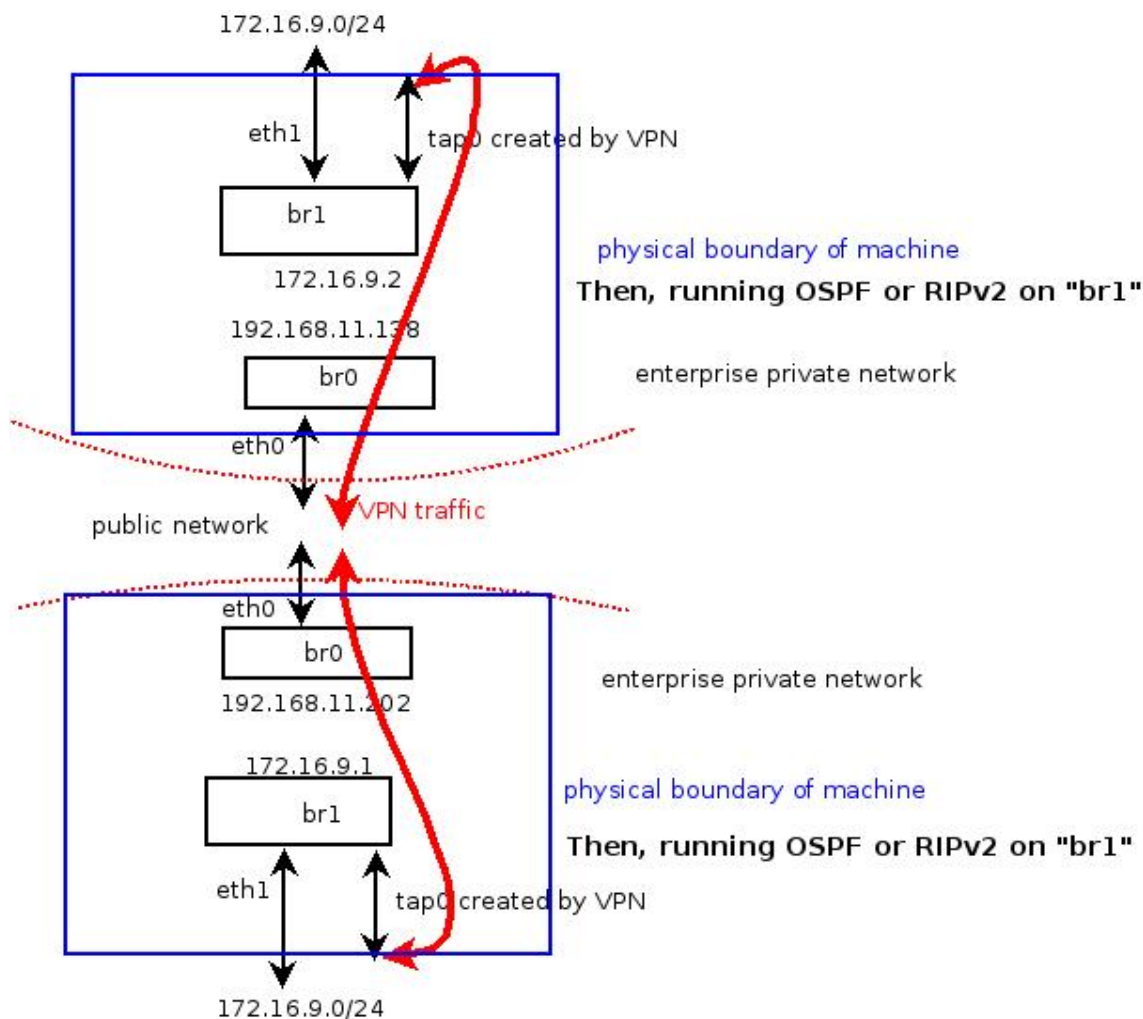


Illustration 158: Two Private Routers Try to Connect Across Internet

For site-to-site VPN in bridging mode, we have a similar diagram to the setting above. However, this time we bring up OSPF or RIPv2 to exchange routing information across VPN.

**Illustration 159: OSPF or RIPv2 across VPN (in Bridging Mode)**

Chapter 6 Deployment Scenarios

In this chapter we provide some examples along with their corresponding requirements and briefly introduce how to use the functions shown in the previous chapters to fulfill those requirements.

Example 1: Isolate the Machines with/without Internet Access

In this example, we would like to place the machines with/without Internet Access into different zones. For security reasons, the machines for internal business operations are not allowed to have Internet connections; the machines with Internet access can not initiate network connections to those machines for business operations.

Let's recall the section for zone “**dmz**” in the chapter of Border Control: it is forbidden for those hosts in zone “**dmz**” to initiate network connections to those machines in zone “**loc**” by default.

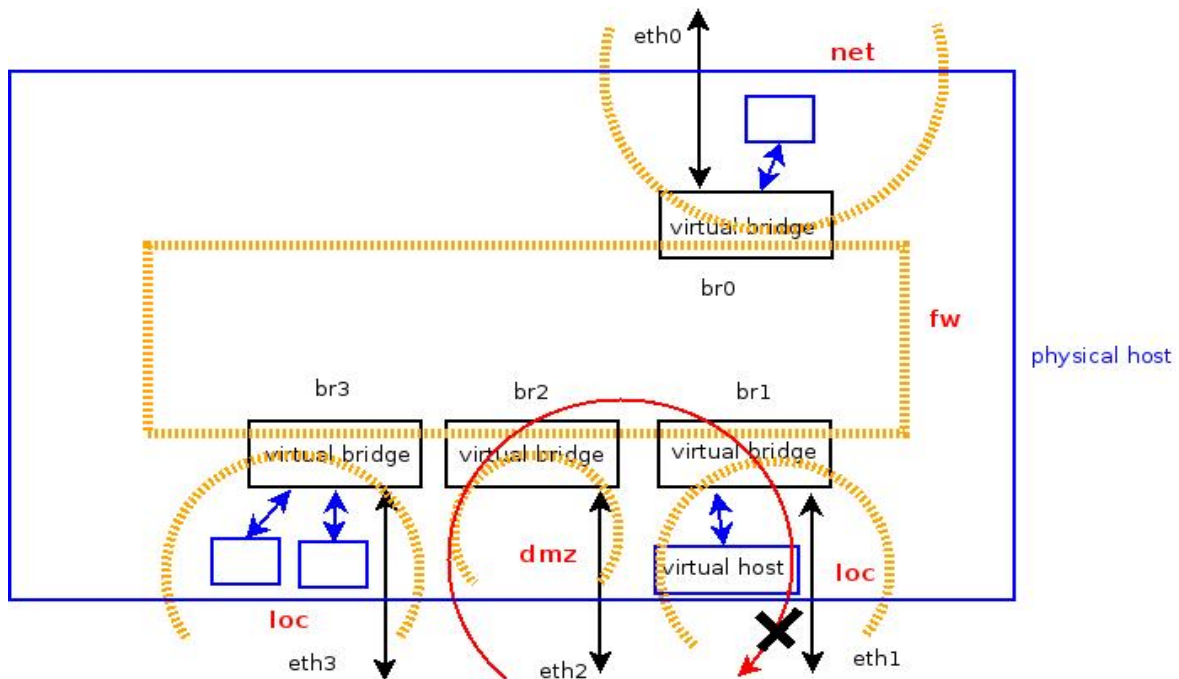


Illustration 160: Zone "dmz" Rules

Thus, we can place those machines with Internet access into the zone "dmz". And the machines for internal business operations into zone "loc". But zone "loc" are allowed to access zone "net" by default. Therefore, we need to add extra rule to block access from zone "loc" to "net". This can be done via "**Border >> Rule >> Add Rule**" as follows:

Add Rule

Border >> Rule >> Add Rule

Action : DROP

Source : loc (br1 br3 br4 br5 br6 br7 br8 br9 br10 br11)

☐ Specify

Destination : net (br0)

☐ Specify

Protocol : tcp

Destination Port : -

Source Port : -

Original Destination IP : -

Rate Limit:

Average Burst Interval sec

Add

Illustration 161: Block the Access to Internet from Zone "loc"

A lot of office environments are based on this configuration by placing Internet servers into zone "**dmz**" by proper port forwarding setup and place the other hosts into zone "**loc**" by restricting the Internet access.

But this configuration is with more applications. Assume there is a requirement to ask that each person in the office is with two desktop machines: one is for internal business use, the other is for Internet access; the machine for internal business use is forbidden to access Internet directly. To fulfill the requirement, those machines for the use of accessing Internet can exist in the form of "virtual machines" in zone "**dmz**"; the person can browse the Internet by using VNC client, SPICE client, or Microsoft Remote Terminal (if the OS in virtual machine is Microsoft Windows) from his/her machine for internal business use to those virtual machines in zone "DMZ" and use the browser from the virtual machines to access Internet. Therefore, it reduces the chances that the machines for internal business uses are compromised.

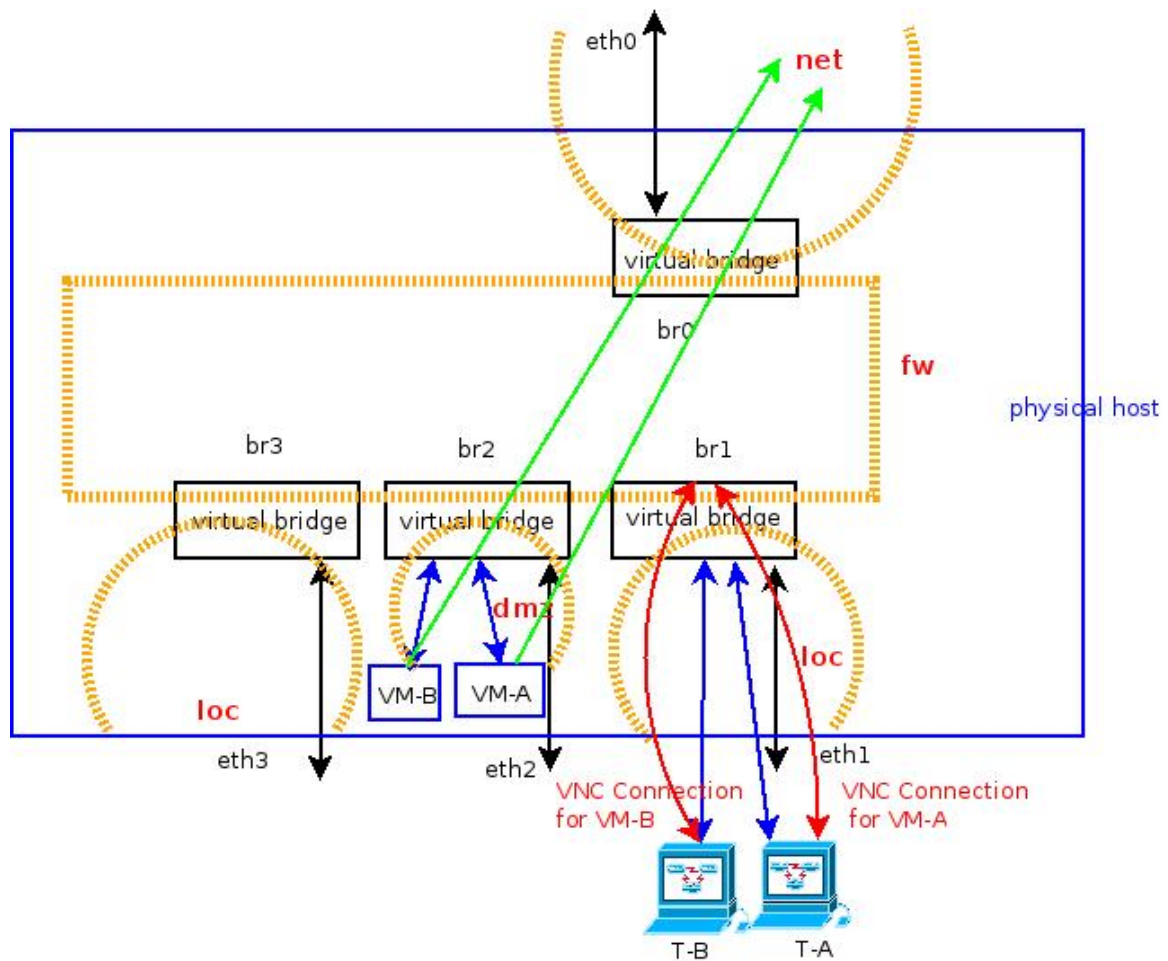


Illustration 162: Use Virtual Machines for Internet Access

Please note that you are connecting to base platform's IP address (for example, the IP address of "br0") and TCP port from VNC client. In the diagram above, the machine T-A is with VM-A's console displayed on its screen via VNC. From that VM-A's console, he/she can access the Internet from the programs installed on VM-A.

Example 2: Use VPN to Access Virtual Desktop

While using VNC client (or SPICE client) to access virtual machines within base platform, please note that the VNC client should connect to the IP address(es) of the base platform, not the IP address of the virtual host. Therefore, while using VNC over VPN, you should use the IP address of the base platform in VPN(the first IP address of the address pool for VPN use, 172.16.38.1 by default). Once you have the console of virtual machine passed back from VNC, you can use that virtual machine to access the other machines in the office governed by the network policies.

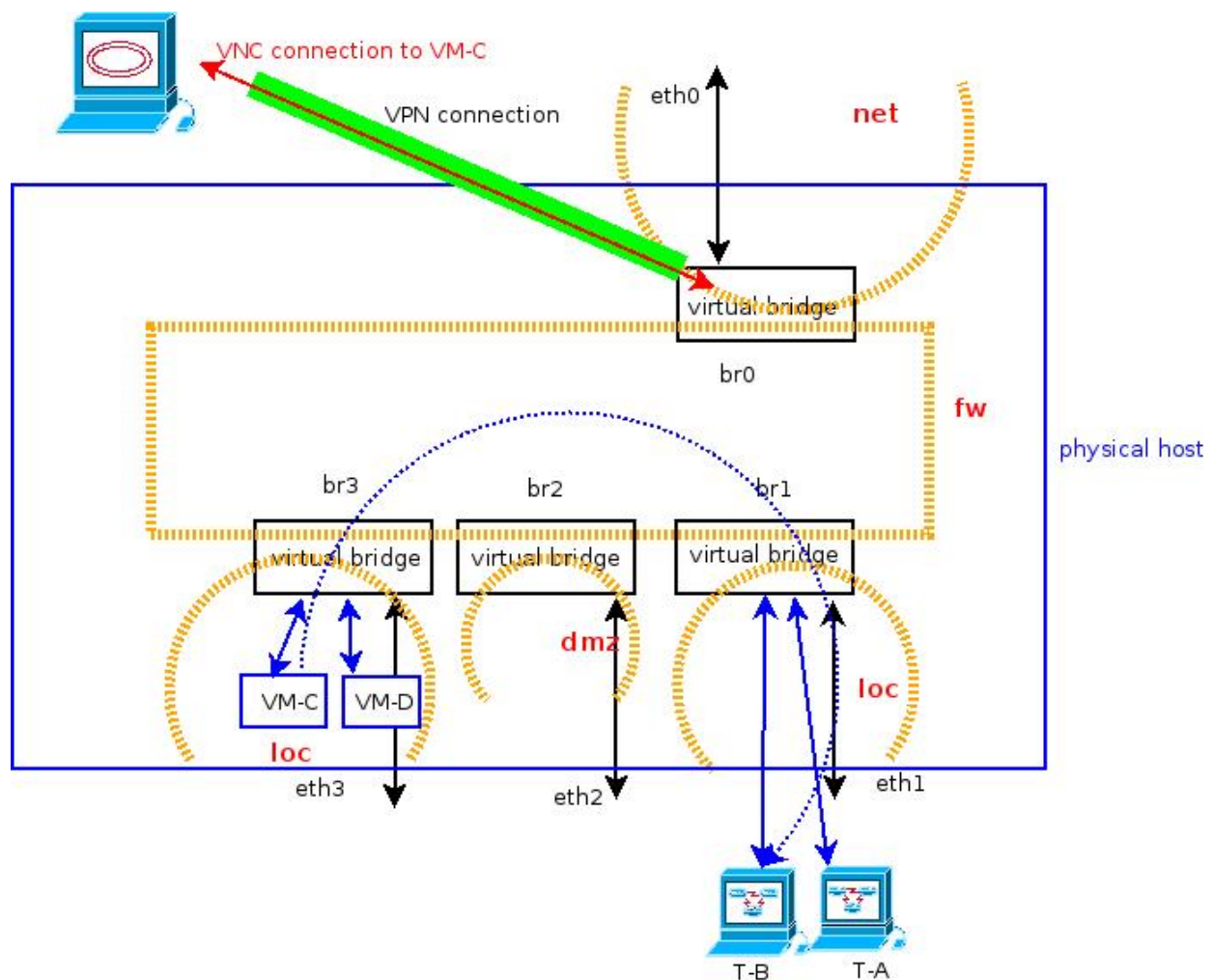


Illustration 163: Access Virtual Machine via VPN

In the diagram above, the console of VM-C can be accessed via VNC over VPN from the place outside the office. With the console of VM-C at hand, it is easier to access the other machines in office. In this way, it is not necessary to grant the other subnet's access to VPN client directly; and some other authentication scheme can be enforced in the virtual host VM-C.

Example 3: SBC or Firewall Virtualization

SBC and Firewall are known with at least two network interfaces: one is connecting to the Internet, and the other is connecting to the private network. Thus, to create a virtual for the use of SBC or firewall, just provide the network interfaces for this virtual machine into "br0" (for WAN connection) and "br1" (for LAN connection). To use the SBC or Firewall, just use the LAN IP address of this virtual host as the gateway to redirect the traffic.

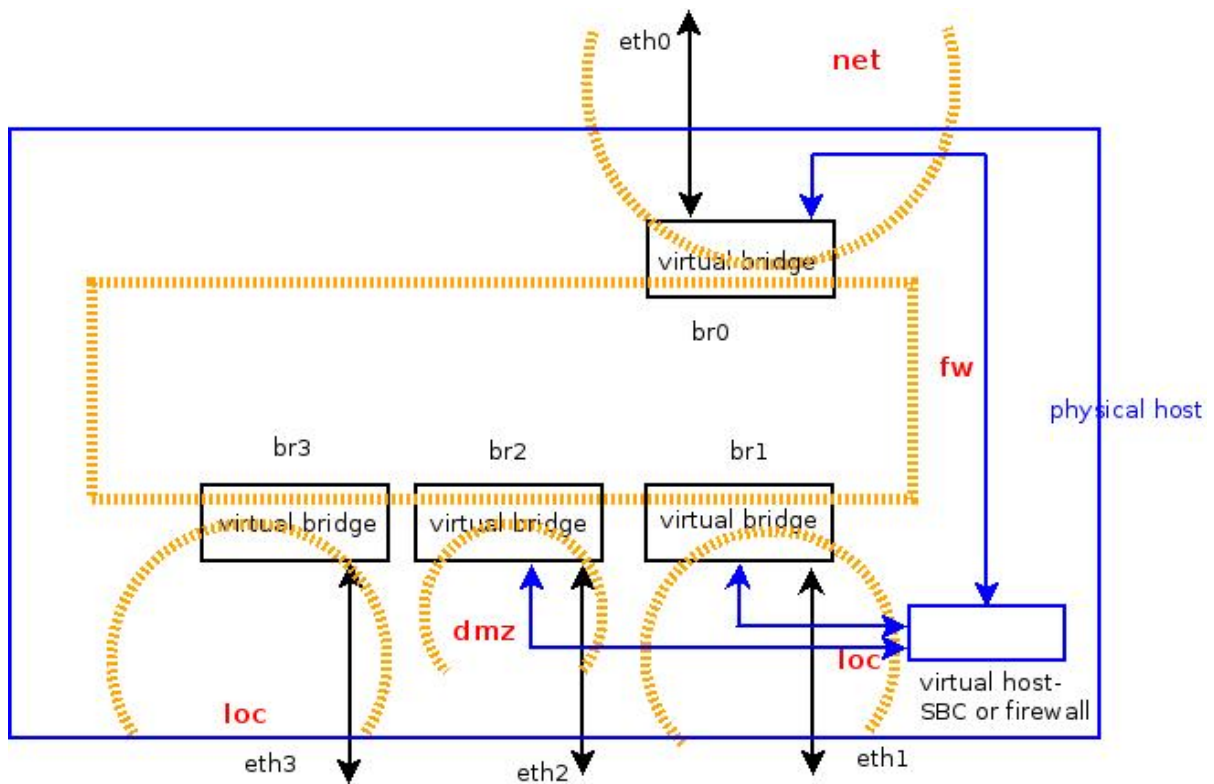


Illustration 164: SBC or Firewall Virtualization

Example 4: Place Storage System in Another Subnet

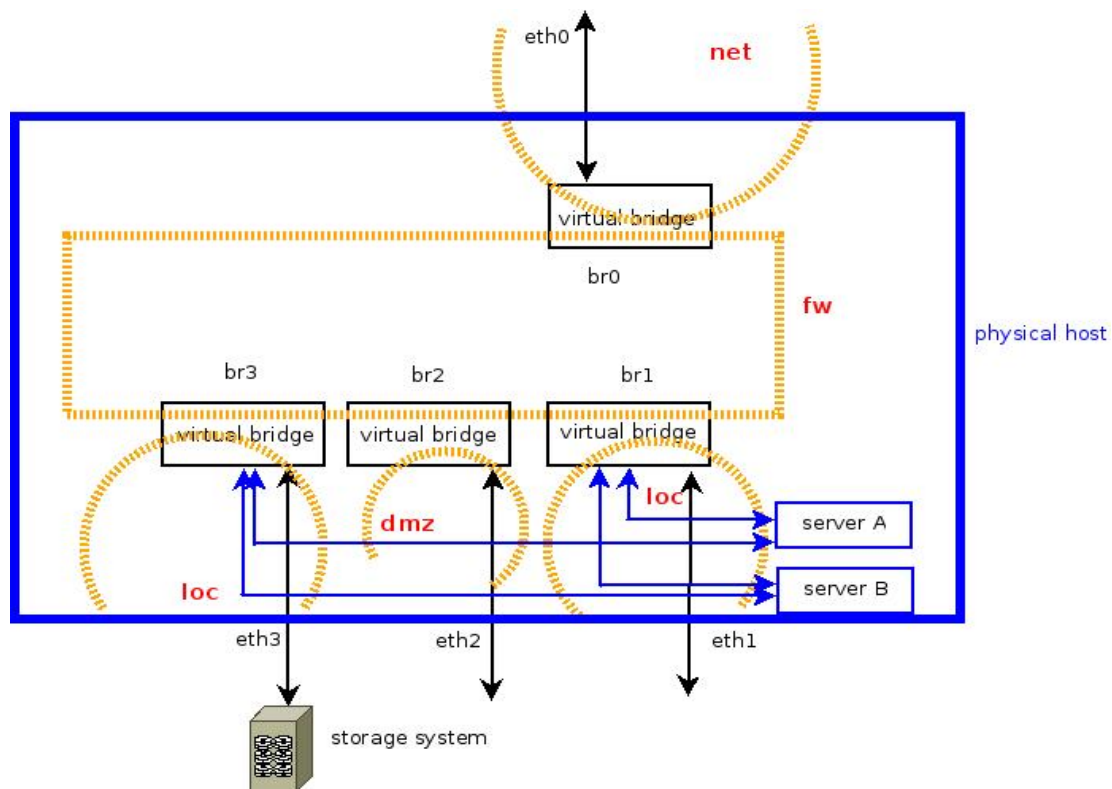


Illustration 165: Use Dedicated Subnet as Storage Area Network

The base system might not be able to provide sufficient storage space for the virtual machines. The diagram above can be used as reference for this type of applications: “server A” and “server B” are used as the servers for the subnet connecting to “br1”, but they are with extra network interfaces in “br3” to connect to an external storage system by using iSCSI. In other words, the subnet where “br3” resides is used as storage area network.

Example 5: Multicast Router with Multicast Sender

The following diagram is an example to use the base system as a multicast router and the multicast sender resides in the virtual machine. The “Border Control” (Firewall functions) has been turned off so that we do not have the labels for “**net**”, “**dmz**”, and “**loc**” on the diagram; each subnet is of the same role. If “Border Control” is turned on, please note “**net**” and “**dmz**” are governed by the rules we mentioned earlier so that multicast routing does not work for these two zones. To make multicast routing function on the the base platform, just turn on PIM.

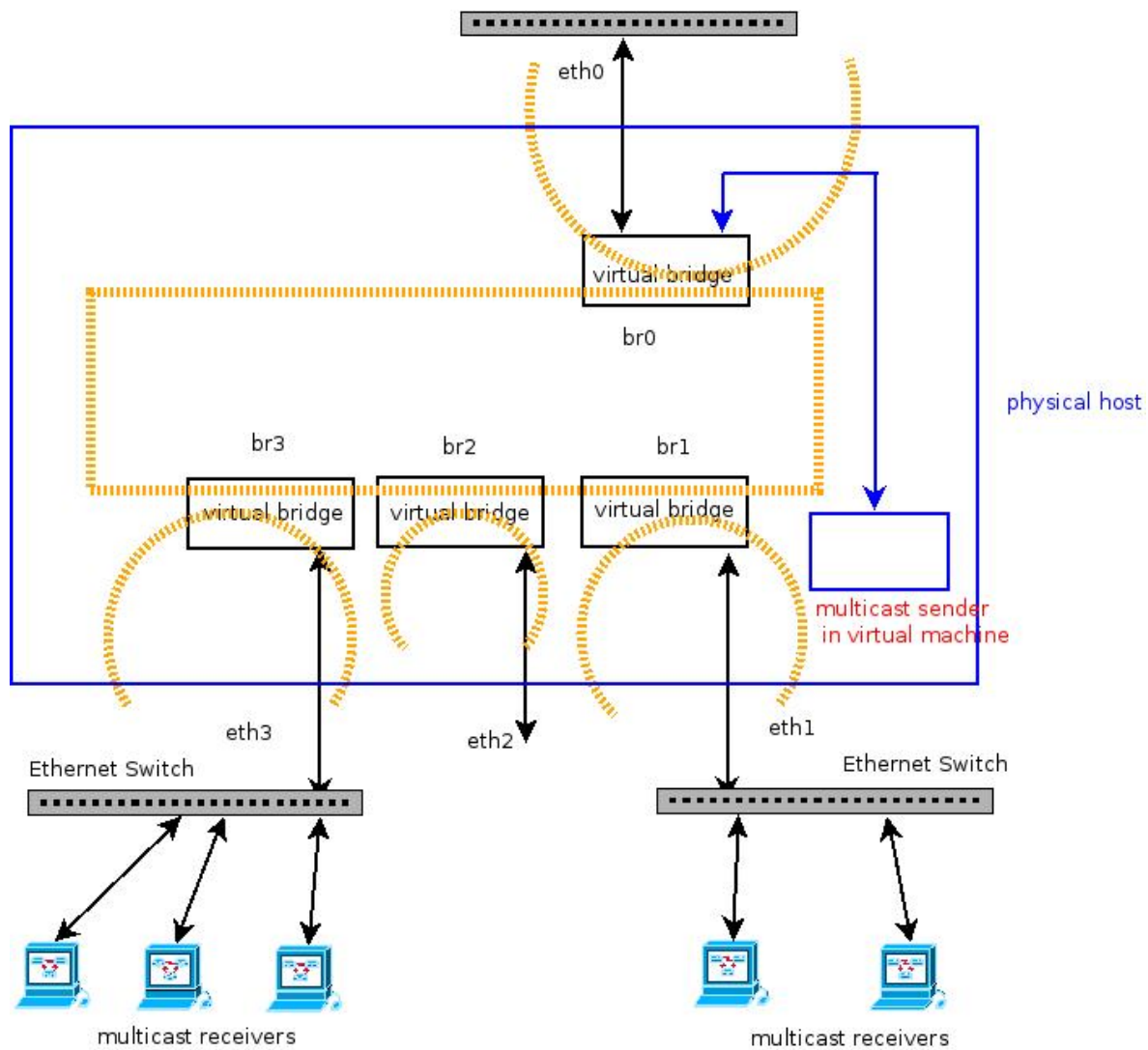


Illustration 166: Multicast Router with Multicast Sender

In the diagram above, a multicast sender is installed in a virtual machine attached to "br0".

As far as the base platform is concerned, it only needs to turn on PIM. The rest of the work will be dependent on the multicast sender and receivers. The "TTL" (time-to-live) setting in the multicast packets at least should be larger than 1 to go across the router to reach the other subnet. Each time when a multicast packet go across the router, the value of TTL will be decreased by 1. Once the value of TTL is less or equal to zero, the router will stop forwarding the packet.

Some of the people might use the open source package "**VLC**" to test multicast sending and receiving. At the moment of writing this document, the default setting of TTL in **VLC** is "-1" so that its multicast packets can not go across its own subnet by default. You need to change that setting while using **VLC** as multicast sender. Otherwise, the multicast sender and receivers can only reside on the same subnet.

We do not want this document as VLC tutorial; we just provide some hints for **VLC** to function as we expect. The following screen snapshot is about where to modify the TTL value in **VLC**.

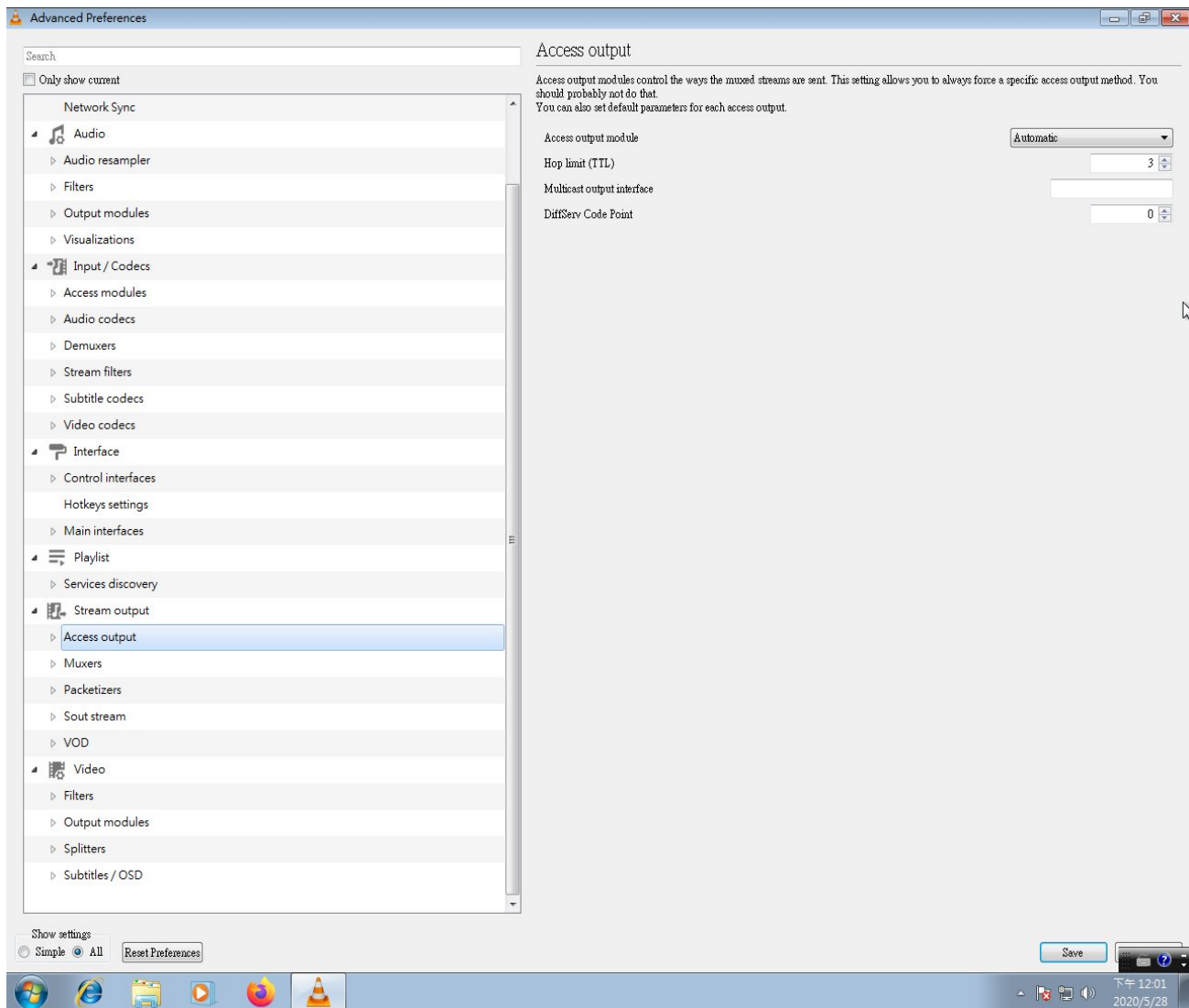


Illustration 167: Example of TTL setting in VLC as Multicast Sender