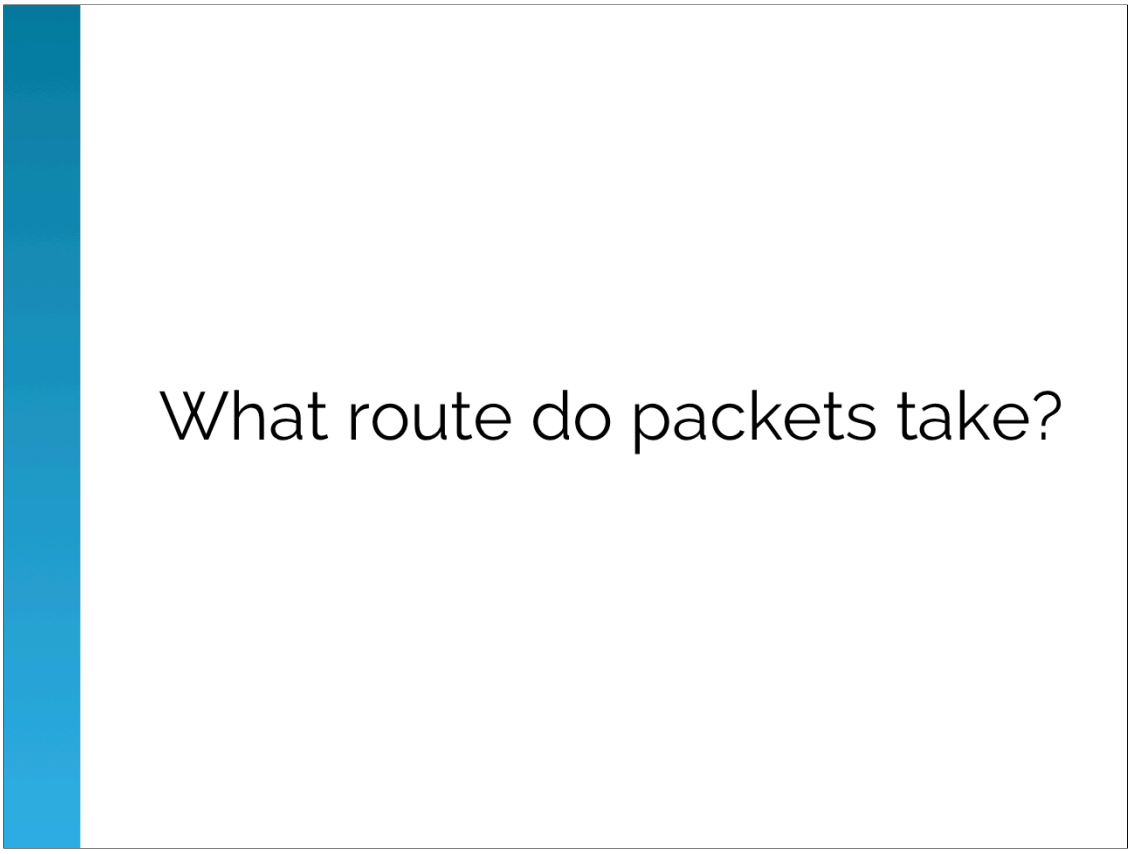
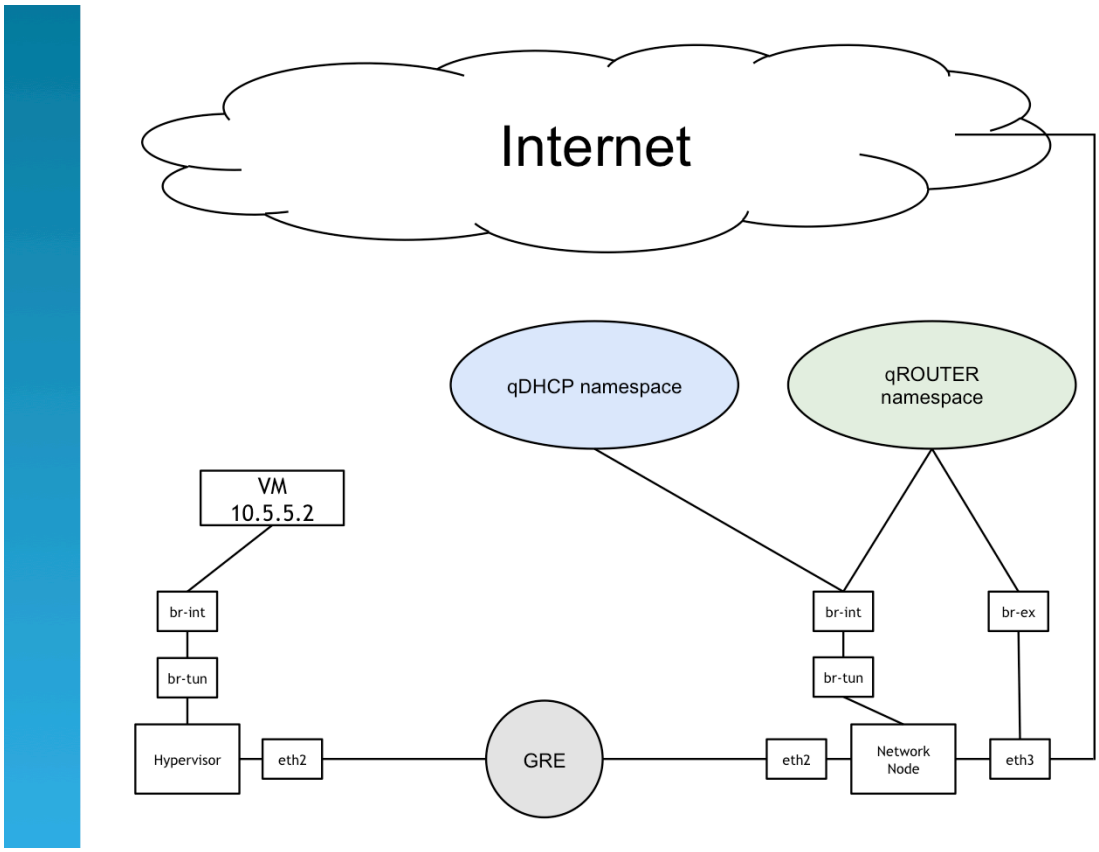
Neutron Packet Flows

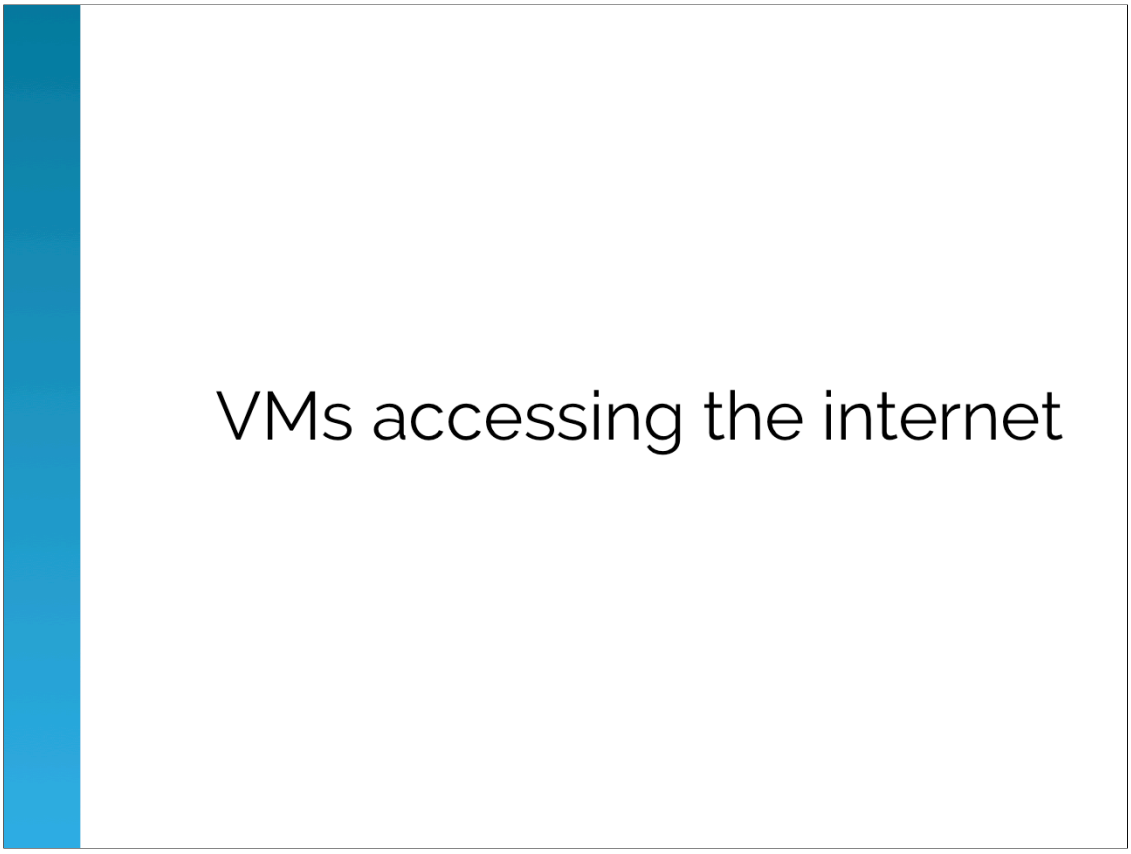
hastexo!



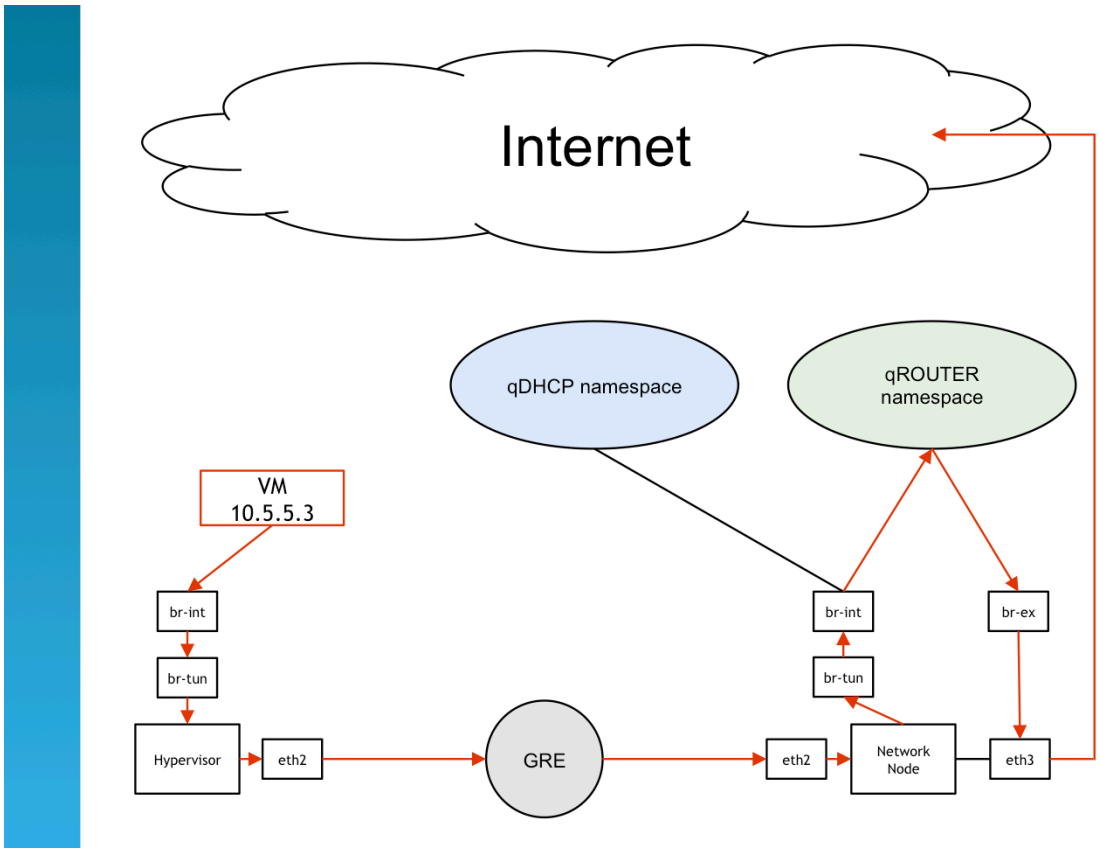
As you have seen by now, we are talking about numerous interfaces here, br-int, br-tun, br-x, eth1/2/3, and for Neutron beginners, it's hard to understand what route within all these devices and hosts packets actually take. So let's take a closer look.



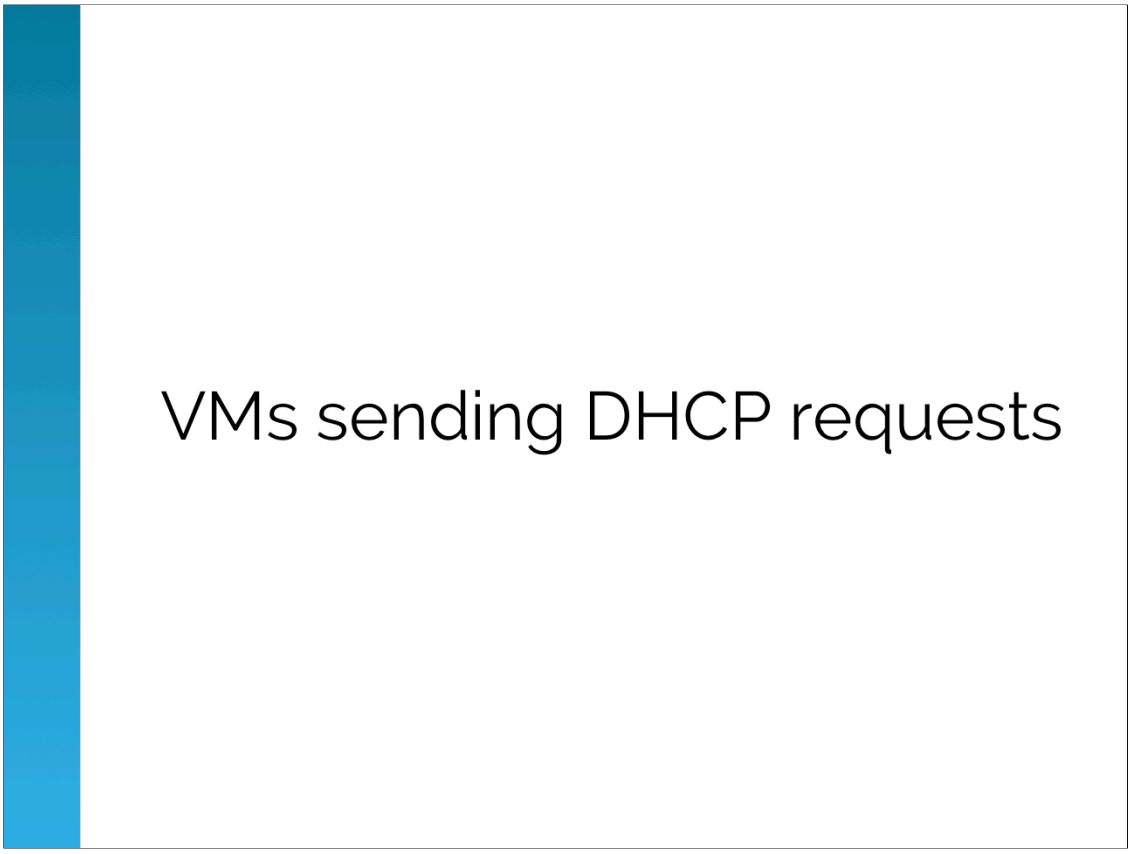
- This is the standard situation. We have
- A hypervisor node with
  - br-int: This is what the VMs connect to; every VM running on a hypervisor will have its virtual network interface attached to br-int
  - br-tun: This is the tunnel interface that, in GRE mode, adds GRE tunnel headers to the packages
  - eth2: The actual internal physical interface where packages send out via eth2 go to
- The GRE tunnel, which, of course, is virtual only and would have all hypervisors and the network node in it
- The networking node with
  - br-tun/br-int, which serve the same purpose they serve on the hypervisor node
  - br-ex physically attached on top of eth3, which is used for communication to the outside
  - Network namespaces:
    - qDHCP as interface within the GRE tunnels (one for every tenant network)
    - qROUTER for connections to the outside (one for every external network)
- Now, let's take a look at what route packets take. Let's start with an easy example



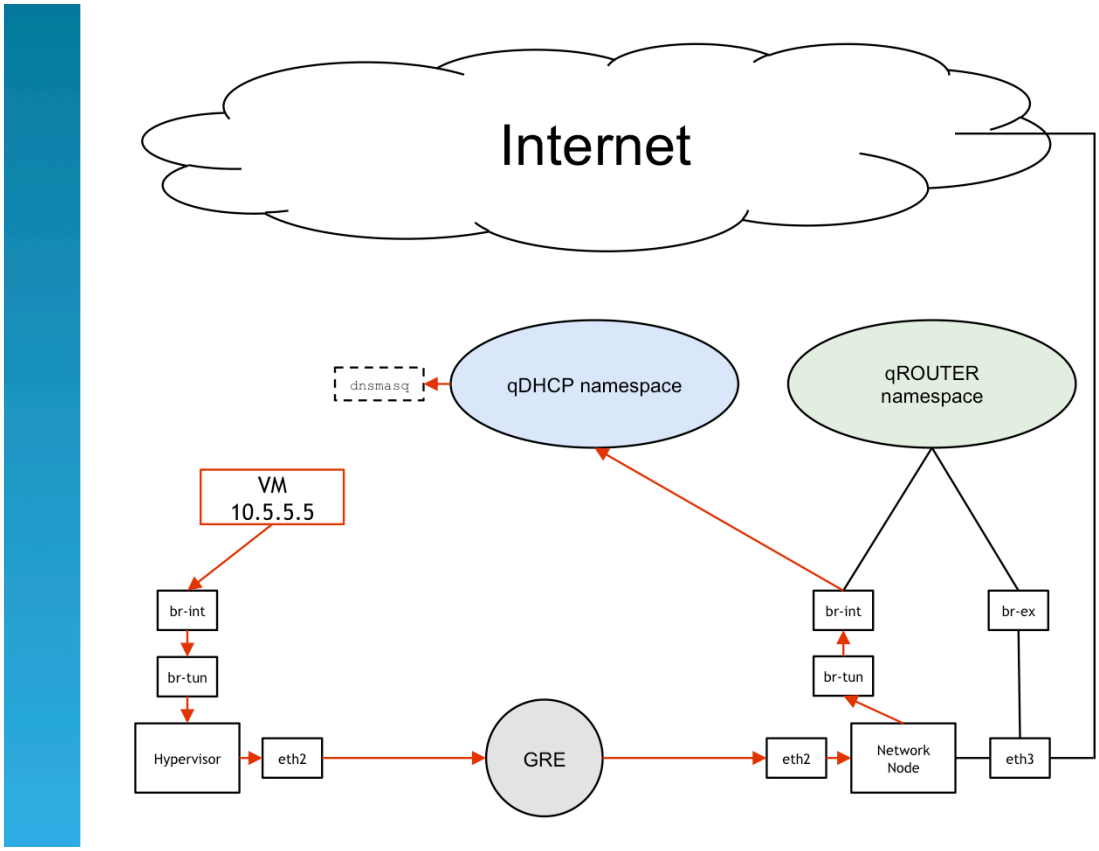
How exactly does a packet make it to the internet when it leaves the VM?



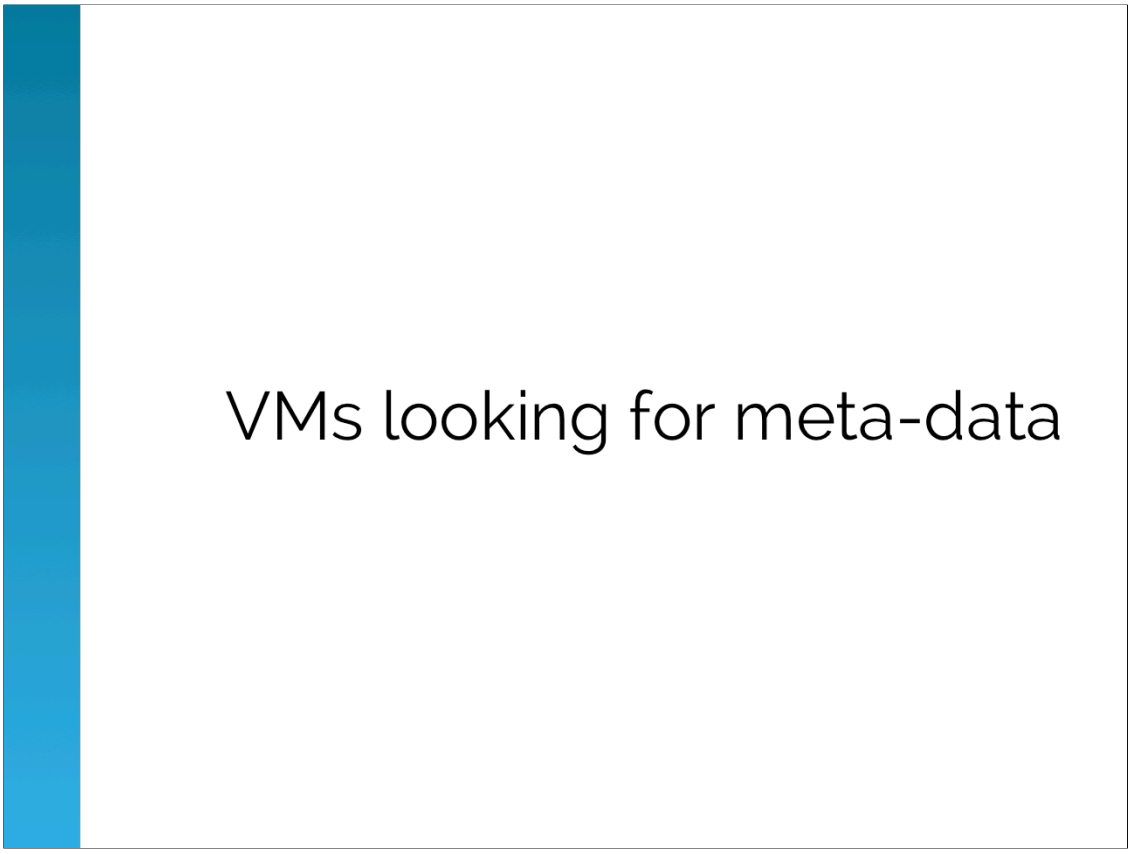
- The packet leaves the VM and hits the hypervisor node on br-int
- From there, it is forwarded to br-tun, which adds the GRE encapsulation to it
- Finally, it leaves the hypervisor on eth2 into the GRE network
- Finally, it reaches eth2 on the network node
- It gets send through br-tun, which removes the GRE Headers
- It is then forwarded to br-int
- br-int on the network node is connected to the qROUTER namespace
    - That is why we defined our external network as router for the admin tenant network earlier!
- From br-int, the packet hits the external bridge (br-ex) and finally leaves the network node into the Internet



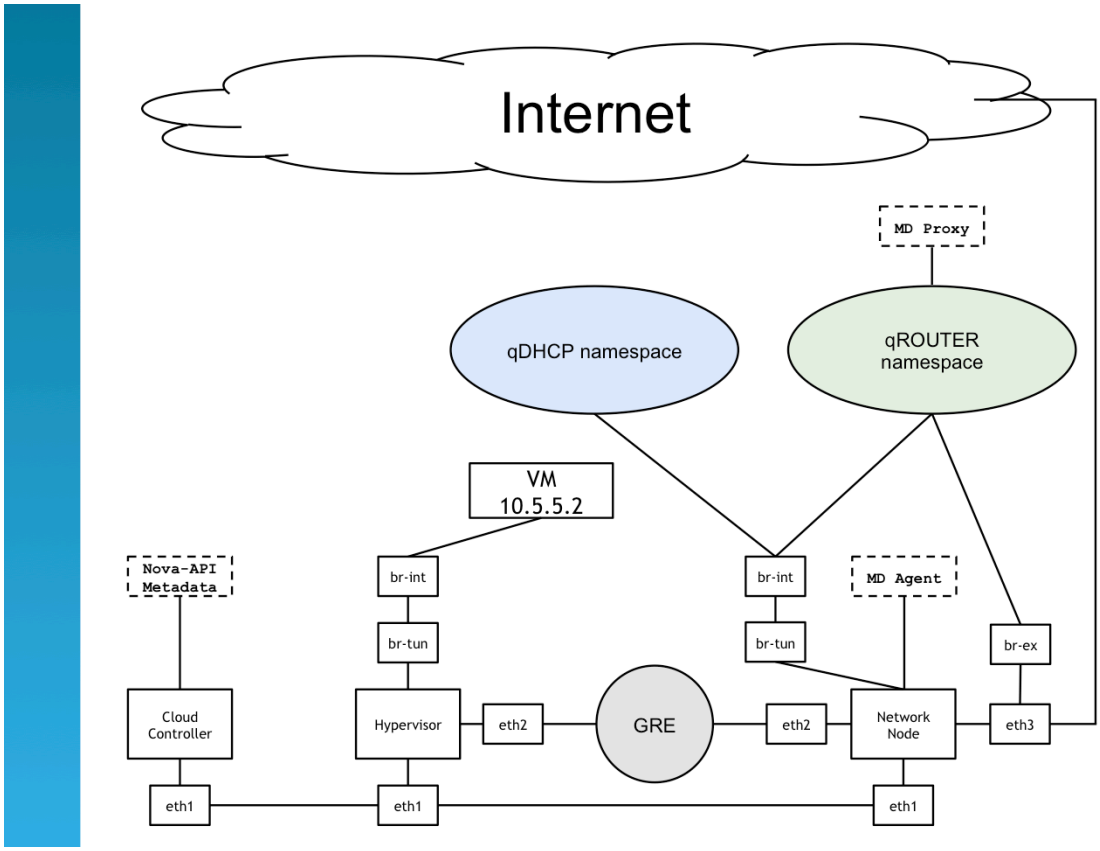
How exactly does a packet make it to the internet when it leaves the VM?



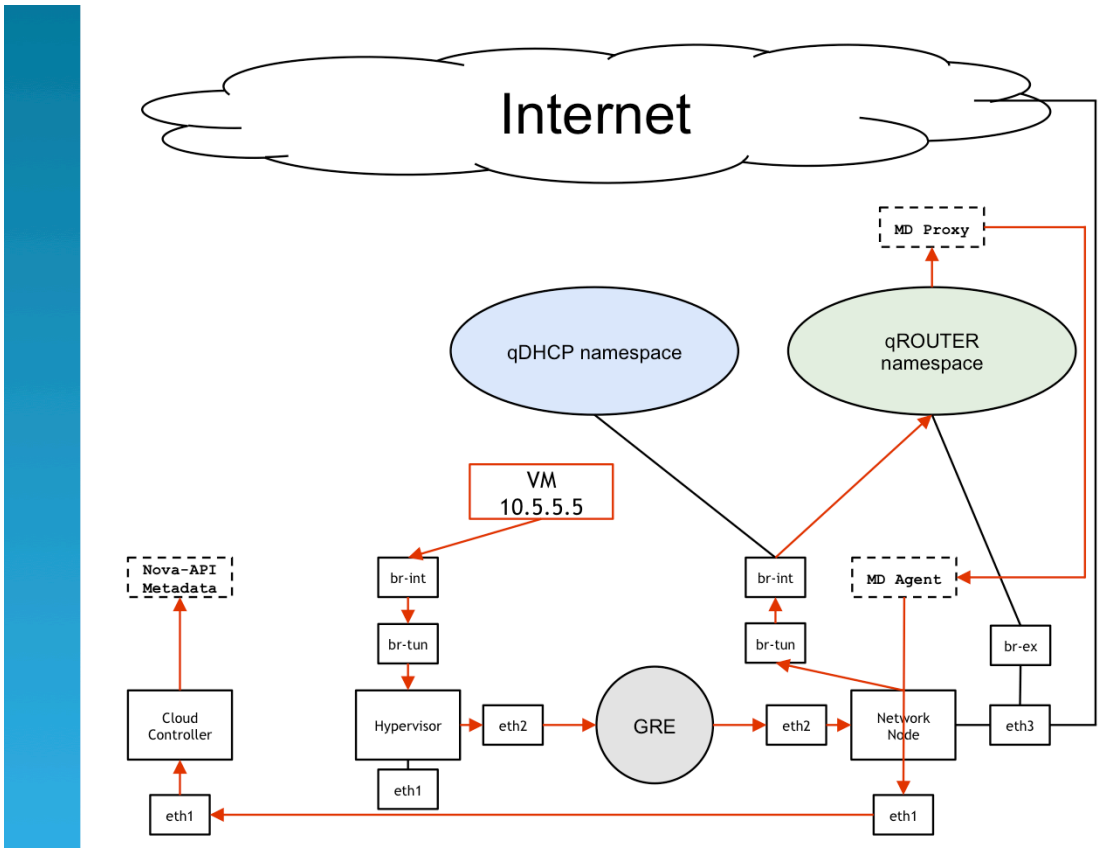
- The packet leaves the VM and hits the hypervisor node on br-int
- From there, it is forwarded to br-tun, which adds the GRE encapsulation to it
- Finally, it leaves the hypervisor on eth2 into the GRE network
- Finally, it reaches eth2 on the network node
- It gets send through br-tun, which removes the GRE Headers
- It is then forwarded to br-int
- br-int on the network node is connected to the qDHCP namespace
    - The dnsmasq processes actually are running within the qDHCP namespace context
- The qDHCP request is then answered accordingly



The third example demonstrates what happens in the background if a VM asks for its meta-data by running the cloud-init script at boot time. Thing's are slightly more complicated here, but let's see.



- To get a better understanding of the process, first, we need to add the cloud controller to our scheme here, and also, we need to add the so-called management network, which all nodes are physically connected to, with no tunneling, via the eth1 physical interface
- Also, we have the Nova-API metadata service running on our cloud controller. This is the source of metadata, this is the service that VMs eventually need to connect to to get meaningful metadata information
- And then, we have the metadata-agent and the metadata proxy running on the network node
- So how to packets flow?



- Let's start with the actual request coming from the VM. It takes its usual way and will eventually make it to the qROUTER namespace
  - Remember: Targes to 169.254.169.254, thus leaves the VM to its default route, which is an IP inside the qROUTER namespace
- Within the qROUTER namespace, the package will be forwarded via DNAT from port 80 to the qrouter's port 9697
  - ... which is where the Metadata Proxy is running
- The Metadata Proxy will relay the packets to the Metadata agent running on the Network node host
  - This happens outside of any TCP/IP network connectivity; in fact, the MD agent has a UNIX socket open in `/var/lib/quantum/metadata_proxy` which the Metadata proxy sends the packages into