Aula 08 - Dispositivos de rede

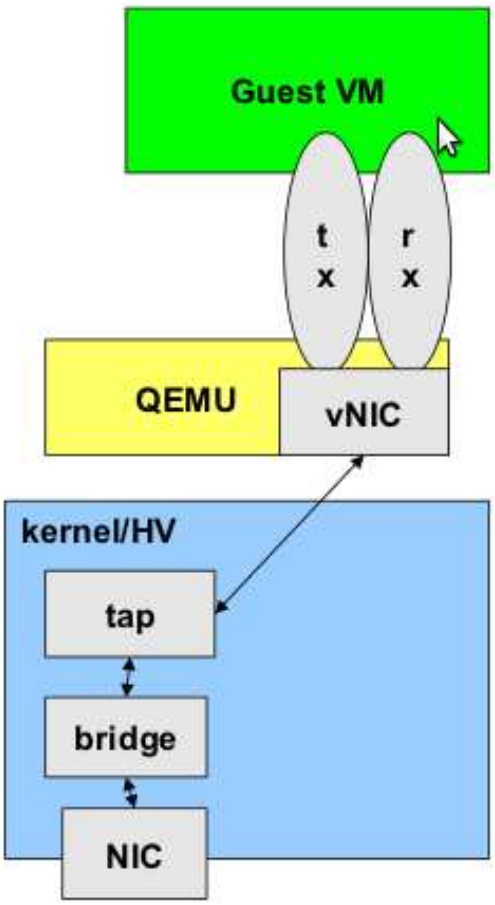
Sobre

- Objetivos:
  - Implementar mecanismos de segurança e isolamento de redes
  - Ajustes de performance para melhor desempenho em redes Gigabit
  - Introdução a redes virtuais

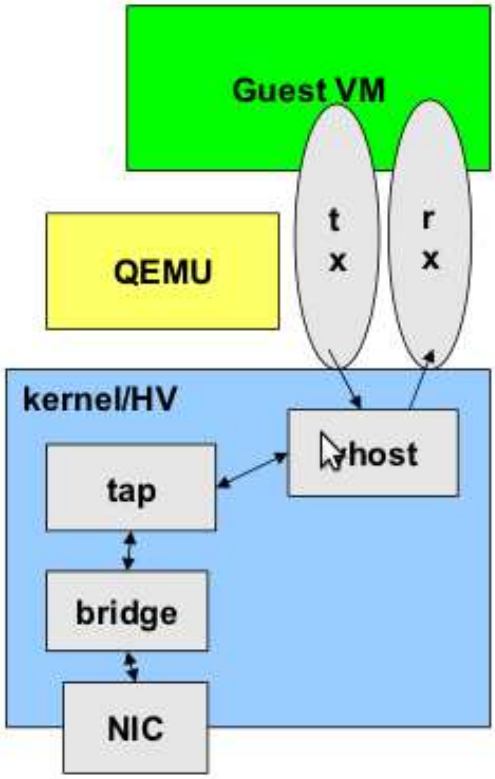
Ajustes de performance

- Comparativo de performance com iperf
- iperf host->guest (com e sem vhost)
- iperf guest->guest (com e sem vhost)

Sem vhost-net



Com vhost-net

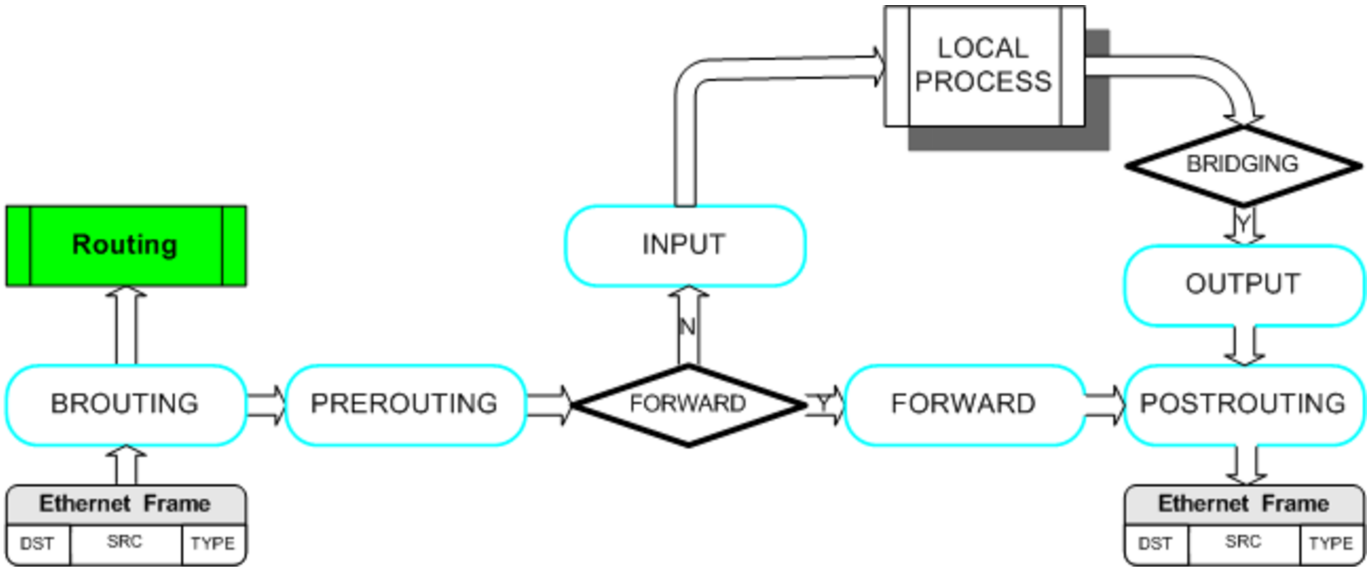


Segurança em bridges

- Camada de enlace
- Riscos conhecidos:
  - ARP spoofing
  - MAC flooding

Para proteger, usamos o ebtables:

```
# ebtables -P FORWARD DROP
# ebtables -A FORWARD -s 00:00:00:00:00:01 -i tap1 -j ACCEPT
# ebtables -A FORWARD -s 00:00:00:00:00:02 -i tap2 -j ACCEPT
# ebtables -A FORWARD --log
```



Deixando o host incomunicável com a bridge:

```
# ebtables -P OUTPUT DROP
# ebtables -A OUTPUT --log
```

- IDS: http://www.linuxsecure.de/index.php?action=90
- http://ebtables.sourceforge.net/br_fw_ia/br_fw_ia.html
- http://ebtables.sourceforge.net/misc/ebtables-faq.html

Cuidados com iptables

- Processamento de iptables dentro das bridges
- O módulo conntrack funciona globalmente

```
/etc/sysctl.conf:
 net.bridge.bridge-nf-call-ip6tables = 0
 net.bridge.bridge-nf-call-iptables = 0
 net.bridge.bridge-nf-call-arptables = 0
```

PCI passthrough

- Passar um dispositivo diretamente para o guest.
- Apenas um guest pode ter acesso.
- Pode ser útil em casos de muita sensibilidade à latência como VoIP, ou hardware especializado.

```
# Parâmetro intel_iommu=on no boot
# lspci (identificar endereço no barramento)
```

```
# lspci -n (pegar o ID do dispositivo)
# echo "10ec 8136" > /sys/bus/pci/drivers/pci-stub/new_id
# echo "0000:04:00.0" >
/sys/bus/pci/devices/0000\:04\:00.0/driver/unbind
# echo "0000:04:00.0" > /sys/bus/pci/drivers/pci-stub/bind
# hvm -device pci-assign,host=04:00.0
```

VMDq (Virtual Machine Device Queues

- Ordenador de camada 2 que envia o quadro para filas específicas associadas com uma interrupção específica para um guest.
- Intel VMDq Demonstration - Live

SR-IOV (Single Root I/O Virtualization)

- Permite que um dispositivo PCIe pareça ser múltiplos dispositivos físicos separados.
- Placas de rede com múltiplas portas
- Cada dispositivo pode ser passado para um guest
- Para saber mais:
  - http://blog.scottlowe.org/2009/12/02/what-is-sr-iov/
- Intel SR-IOV Explanation

VLAN 802.1q (VLAN taggeada)

- Configuração usando **vconfig** ou **ip** no guest.
- Importante: manter uma bridge por VLAN ID.
- Evitar misturar VLANs na mesma bridge.

Inter-conexão de bridges

- Tunelando bridges gretap: http://benoit.papillault.free.fr/blog/?p=54
- GRE: http://en.wikipedia.org/wiki/Generic_Routing_Encapsulation

Performance e QoS

- MTU (jumbo frames)
- Tuning 10Gb network cards on Linux: http://www.kernel.org/doc/ols/2009/ols2009-pages-169-184.pdf
- txqueuelen

- ethtool -K eth0
- Isolamento de banda utilizando Xen+tc (serve para KVM também) http://horms.net/projects/xen-bw-isolation/2010-08/bw.en.pdf
- cGroups

Switches e redes virtuais

- http://openvswitch.org/
- http://www.openflow.org/
- http://vde.sourceforge.net/

Referências

- Cenários: http://blog.klauskiwi.com/wp-content/uploads/2010/08/KVM-Security_en.pdf
- Artigo muito bom sobre detalhes internos de interfaces TAP: http://www.linuxfi.com.br/artigos/tun_tap.pdf
- Disable net.bridge.bridge-nf-call-*tables by default: https://bugzilla.redhat.com/show_bug.cgi?id=512206
- VLAN 802.1q, seguir thread: http://www.mail-archive.com/bridge@lists.linux-foundation.org/msg01269.html

Tópicos para estudo

- JLS2009: Generic receive offload: http://lwn.net/Articles/358910/
- Receive packet steering: http://lwn.net/Articles/362339/
- Rps: Receive packet steering: http://lwn.net/Articles/361440/
- Multiqueue networking: http://lwn.net/Articles/289137/
- http://virt.kernelnewbies.org/MacVTap
- Provide a zero-copy method on KVM virtio-net. http://lwn.net/Articles/407939/
- Linux Containes and Networking: http://blog.flameeyes.eu/2010/09/04/linux-containes-and-networking
- Comandos `ip` equivalentes ao ifconfig: http://jengelh.medozas.de/2008/0219-ifconfig-sucks.php