- Log in

Documentation/Networking - QEMU

- Page
- Discussion
- View source
- History

Search Go Search

- About
  - Home
- Get
  - Download
- Contribute
  - Start Here
  - Report a Bug
  - Report a security issue
  - Submit a Patch
  - Mailing Lists
  - Testing QEMU
- Virtualize
  - KVM
- Learn
  - Documentation
  - Links
  - License
- Toolbox
  - What links here
  - Related changes
  - Special pages

Documentation/Networking

From QEMU

There are two parts to networking within QEMU:

- the virtual network device that is provided to the guest (e.g. a PCI network card).
- the network backend that interacts with the emulated NIC (e.g. puts packets onto the host's network).

There are a range of options for each part. By default QEMU will create a SLiRP user network backend and an appropriate virtual network device for the guest (eg an E1000 PCI card for most x86 PC guests).

Note - if you specify any networking options on the command line (via `-net` or `-netdev`) then QEMU will require you to provide options sufficient to define and connect up both parts. (Forgetting to specify the backend or the network device will give a warning message such as "Warning: netdev mynet0 has no peer" or "Warning: vlan 0 is not connected to host network"; the VM will then boot but will not have functioning networking.)

Note - if you are using the (default) SLiRP user networking, then ping (ICMP) will not work, though TCP and UDP will. Don't try to use ping to test your QEMU network configuration!

Note - As this page is probably very brief or even incomplete you might find these pages rather useful:

- QEMU Networking on wikibooks.org, mainly dealing with Linux hosts
- QEMU Networking on bsdwiki, showing used networking principles and dealing with BSD hosts

Contents

Creating a network backend

There are a number of network backends to choose from depending on your environment. Create a network backend like this:

```
-netdev TYPE,id=NAME,...
```

The `id` option gives the name by which the virtual network device and the network backend are associated with each other. If you want multiple virtual network devices inside the guest they each need their own network backend. The name is used to distinguish backends from each other and must be used even when only one backend is specified.

Network backend types

In most cases, if you don't have any specific networking requirements other than to be able to access to a web page from your guest, user networking (slirp) is a good choice. However, if you are looking to run any kind of network service or have your guest participate in a network in any meaningful way, tap is usually the best choice.

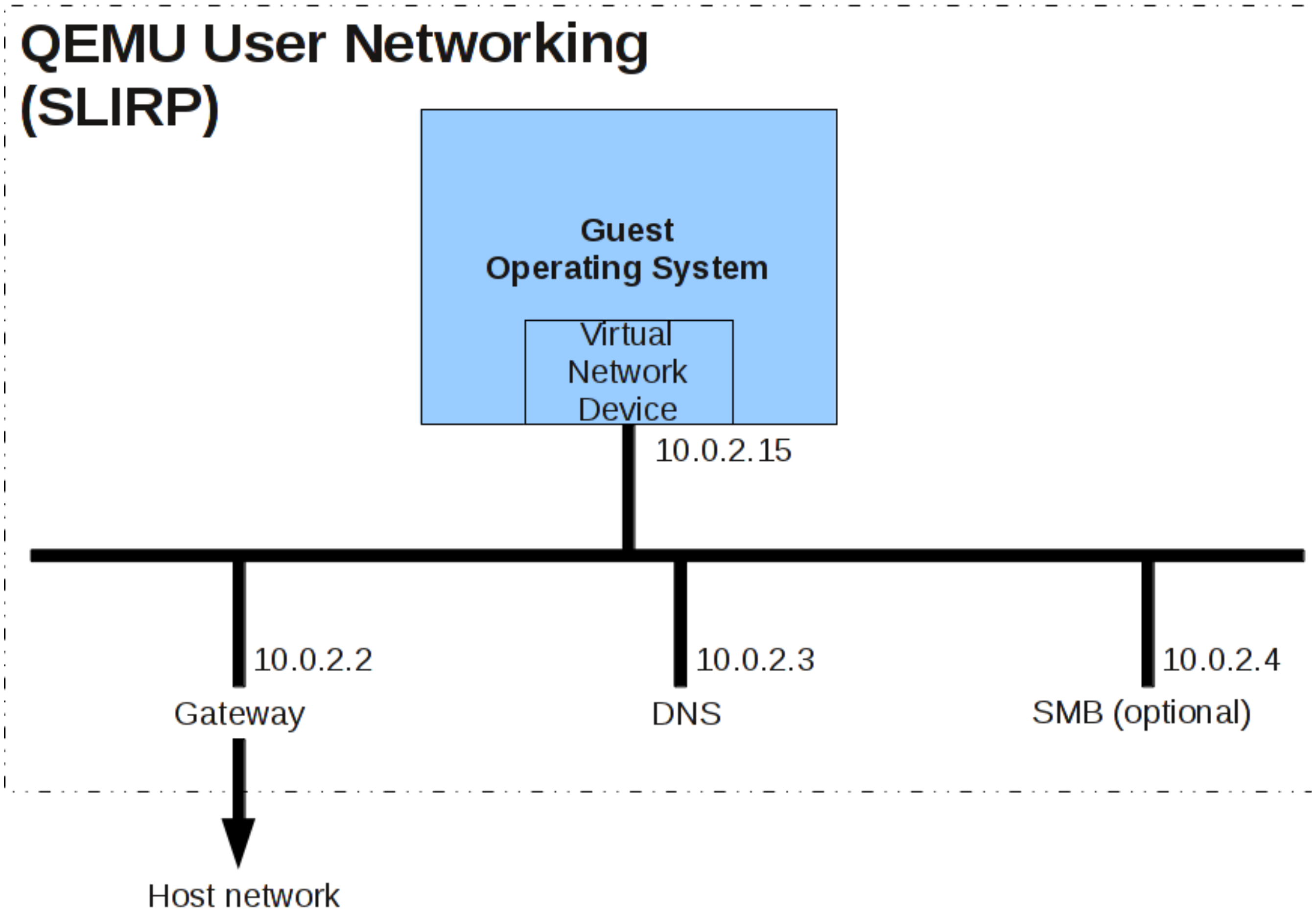
User Networking (SLIRP)

This is the default networking backend and generally is the easiest to use. It does not require root / Administrator privileges. It has the following limitations:

- there is a lot of overhead so the performance is poor
- ICMP traffic does not work (so you cannot use ping within a guest)
- the guest is not directly accessible from the host or the external network

User Networking is implemented using "slirp", which provides a full TCP/IP stack within QEMU and uses that stack to implement a virtual NAT'd network.

A typical (default) network is shown below.



You can configure User Networking using the `-netdev user` command line option.

Adding the following to the qemu command line will change the network configuration to use 192.168.76.0/24 instead of the default (10.0.2.0/24) and will start guest DHCP allocation from 9 (instead of 15):

```
-netdev user,id=mynet0,net=192.168.76.0/24,dhcpstart=192.168.76.9
```

You can isolate the guest from the host (and broader network) using the `restrict` option. For example `-netdev user,id=mynet0,restrict=y` or `-netdev type=user,id=mynet0,restrict=yes` will restrict networking to just the guest and any virtual devices. This can be used to prevent software running inside the guest from phoning home while still providing a network inside the guest. You can selectively override this using `hostfwd` and `guestfwd` options.

TODO:

```
-netdev user,id=mynet0,dns=xxx

-netdev user,id=mynet0,tftp=xxx,bootfile=yyy

-netdev user,id=mynet0,smb=xxx,smbserver=yyy

-netdev user,id=mynet0,hostfwd=hostip:hostport-guestip:guestport

-netdev user,id=mynet0,guestfwd=

-netdev user,id=mynet0,host=xxx,hostname=yyy
```

Tap

The tap networking backend makes use of a tap networking device in the host. It offers very good performance and can be configured to create virtually any type of network topology. Unfortunately, it requires configuration of that network topology in the host which tends to be different depending on the operating system you are using. Generally speaking, it also requires that you have root privileges.

```
-netdev tap,id=mynet0
```

VDE

The VDE networking backend uses the Virtual Distributed Ethernet infrastructure to network guests. Unless you specifically know that you want to use VDE, it is probably not the right backend to use.

Socket

The socket networking backend, together with QEMU VLANs, allow you to create a network of guests that can see each other. It's primarily useful in extending the network created by Documentation/Networking/Slirp to multiple virtual machines. In general, if you want to have multiple guests communicate, tap is a better choice unless you do not have root access to the host environment.

```
-netdev socket,id=mynet0
```

Creating a virtual network device

The virtual network device that you choose depends on your needs and the guest environment (i.e. the hardware that you are emulating). For example, if you are emulating a particular embedded board, then you should use the virtual network device that matches that embedded board's configuration.

On machines that have PCI bus, there are a wider range of options. The e1000 is the default network adapter in qemu. The rtl8139 is the default network adapter in qemu-kvm. In both projects, the virtio-net (para-virtualised) network adapter has the best performance, but requires special guest driver support.

Use the `-device` option to add a particular virtual network device to your virtual machine:

```
-device TYPE,netdev=NAME
```

The `netdev` is the name of a previously defined `-netdev`. The virtual network device will be associated with this network backend.

Note that there are other device options to select alternative devices, or to change some aspect of the device. For example, you want something like: `-device DEVNAME,netdev=NET-ID,macaddr=MACADDR,DEV-OPTS`, where `DEVNAME` is the device (e.g. `i82559c` for an Intel i82559C Ethernet device), `NET_ID` is the network identifier to attach the device to (see discussion of `-netdev` below), `MACADDR` is the MAC address for the device, and `DEV-OPTS` are any additional device options that you may wish to pass (e.g. `bus=PCI-BUS,addr=DEVFN` to control the PCI device address), if supported by the device.

Use `-device ?` to get a list of the devices (including network devices) you can add using the `-device` option for a particular guest. Remember that `?` is a shell metacharacter, so you may need to use `-device \?` on the command-line.

Monitoring Networking

You can monitor the network configuration using `info network` and `info usernet` commands.

You can capture network traffic from within qemu using the `-net dump` command line option. See Stefan Hajnoczi's blog post on this feature.

The legacy -net option

QEMU previously used the `-net nic` option instead of `-device DEVNAME` and `-net TYPE` instead of `-netdev TYPE`. This is considered obsolete since QEMU 0.12, although it continues to work.

The legacy syntax to create virtual network devices is:

```
-net nic,model=MODEL
```

You can use `-net nic,model=?` to get a list of valid network devices that you can pass to the `-net nic` option. Note that these model names are different from the `-device ?` names and are therefore only useful if you are using the `-net nic,model=MODEL` syntax. [If you'd like to know all of the virtual network devices that are currently provided in QEMU, a search for "NetClientInfo" in the source code may be useful.]

QEMU "VLANs"

The obsolete `-net` syntax automatically created an emulated hub (called a QEMU "VLAN", for virtual LAN) that forwards traffic from any device connected to it to every other device on the "VLAN". It is **not** an 802.1q VLAN, just an isolated network segment. When creating multiple network devices using the `-net` syntax, you generally want to specify different vlan ids. The exception is when dealing with the socket backend. For example:

```
-net user,vlan=0 -net nic,vlan=0 -net user,vlan=1 -net nic,vlan=1
```

How do I...

- Use tap to let guests be visible on the host network
- Use tap with a wireless adapter on the host
- Forward ports with slirp
- Pass QEMU a physical card rather than emulation/simulation.

Retrieved from "http://wiki.qemu.org/Documentation/Networking"