

Netkit 安装与实验介绍

Netkit 介绍【参见 1】

功能介绍:

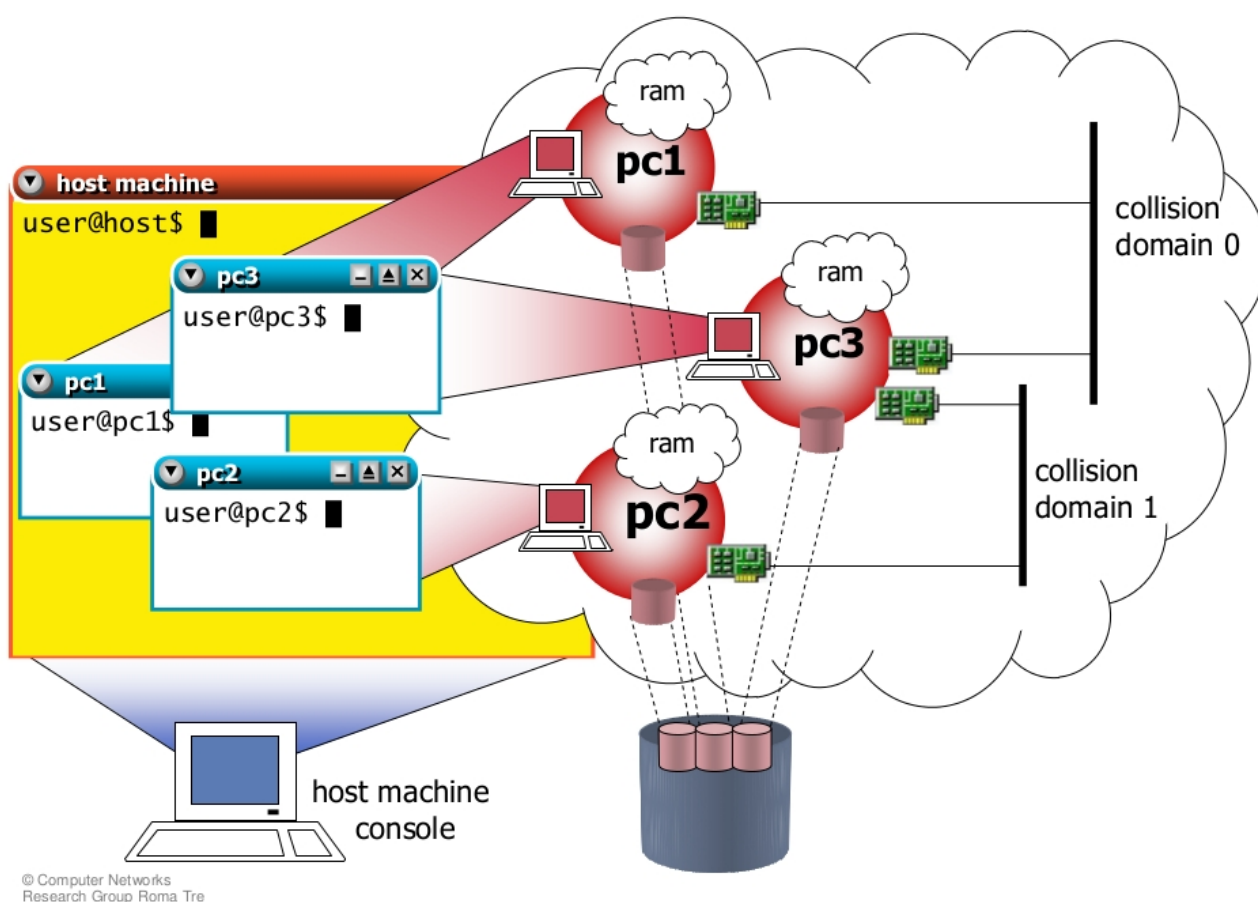
Netkit is a system for emulating computer networks. 以往网络实验往往需要昂贵的设备，因此每次实验都要进行器材和设备的准备工作，代价很高，Netkit 正是一种虚拟化的解决方案，它可以用一台 PC，模拟出实验所需的大部分设备以及网络拓扑，从而把实验搬到了虚拟环境进行。

原理介绍:

Netkit 可以模拟出 PC 机、路由等设备，虚拟的每个设备都是一个虚拟机。Netkit 的虚拟机制采用的 User Mode Linux，通过一个改装版的最小内核，在宿主 Linux 机上虚拟出多个虚拟的 Linux 机，由于虚拟机的硬件和文件系统都可配置，因此通过配置虚拟机多网卡并且在虚拟机上安装路由模拟软件就可以达到模拟路由的目的。

基本实验思路:

虚拟 PC，虚拟 Router，虚拟碰撞域，搭建拓扑：



Netkit 安装【参见 1】

Netkit 软件分三部分构成：

1. Netkit 软件包：这里包含了建立虚拟机、网络设备、拓扑并且进行实验的一系列辅助工具。
2. UML Kernel：每个 Linux 虚拟机都把这个 Kernel 作为自己的内核来建立。
3. 已构建好的一个 file system：Linux 虚拟机只有内核没有文件系统还不能正常工作，这个 filesystem 内包含了许多已编译好的常用软件，包括后文提到的 quagga，并且放置到了合适的位置。当虚拟机使用这个 filesystem 时，可以直接使用这些软件，就好像自己已经安装好了一样。

下载:

名称	下载地址
netkit-2.8.tar.bz2	http://wiki.netkit.org/download/netkit/netkit-2.8.tar.bz2
netkit-filesystem-i386-F5.2.tar.bz2	http://wiki.netkit.org/download/netkit-filesystem/netkit-filesystem-i386-F5.2.tar.bz2
netkit-kernel-i386-K2.8.tar.bz2	http://wiki.netkit.org/download/netkit-kernel/netkit-kernel-i386-K2.8.tar.bz2

安装【参见 2】：

(1) 解压

将三个压缩包放在统一位置解压。

```
tar -xjSf netkit-x.y.tar.bz2
```

```
tar -xjSf netkit-filesystem-Fx.y.tar.bz2
```

```
tar -xjSf netkit-kernel-Kx.y.tar.bz2
```

解压后所有内容都在 netkit 文件夹内。注意 filesystem 是 sparse 文件，解压一定要有 -S 选项，不然 netkit 不能正常运行。

(2) 配置环境变量

filesystem 和 kernel 都是预先编译好的，netkit 工具包都是 script，因此整体不再需要编译安装，只需要配置一些环境变量供在 bash 里使用。

如果要所有 linux 用户都可使用，要修改/etc/profile，否则只更改本地配置文件 ~/.bashrc。在末尾添加如下内容：

```
export NETKIT_HOME=/home/mtawaken/opt/netkit/netkit
export MANPATH=${NETKIT_HOME}/man:$MANPATH
export PATH=${NETKIT_HOME}/bin:$PATH
. $NETKIT_HOME/bin/netkit_bash_completion
```

注意 NETKIT_HOME 就是解压后 netkit 的根目录。

配置完成后运行 netkit 根目录的 check_configuration.sh 检测是否配置正确。

如果结果显示 filesystem 有问题，是因为它检测到了非 linux 系统的文件系统格式，非 ext 格式会导致不支持 sparse 或性能下降。这可能是由于你挂载了 U 盘或 windows 的磁盘造成的，可以忽略。

Netkit 运行示例【参见 3】

启动一个虚拟机：

```
vstart pc1 --eth0=0 --eth1=1 --exec=run_this_script
```

vstart 表示启动一个虚拟机。

Pc1 是虚拟机名。

eth0=0 表示配置 eth0 网卡，并且用网线连接到名称为“0”的碰撞域。碰撞域不需要额外配置，只要在实验中保持一致的拓扑逻辑就可以了。

exec 表示在虚拟机启动之后运行的脚本，这可以预先完成一些任务比如配置网卡。

除了这些指令，还可以配置虚拟机使用的内存等等，甚至可以把宿主机的文件拷贝到虚拟机中。执行 vstart 后，在当前目录会出现 pc1.disk 文件，这就是虚拟机的文件系统，里面保存了对 pc1 的所有改动。如果一切顺利，虚拟机 pc1 的控制台会弹出：

```
pc1
Cleaning up ifupdown....
Mounting kernel modules directory (/home/mtawaken/opt/netkit/netkit/kernel/modules/lib/modules) on /lib/modules/ ...
Loading kernel modules...done.
Setting kernel variables (/etc/sysctl.conf)...done.
Setting up networking....
Configuring network interfaces...done.
Starting portmap daemon....
INIT: Entering runlevel: 2

— Starting Netkit phase 1 init script —
Mounting /home/mtawaken on /hoshome...
Configuring host name...
— Netkit phase 1 initialization terminated —

Starting system log daemon....
Starting kernel log daemon....

— Starting Netkit phase 2 init script —
— Netkit phase 2 initialization terminated —

pc1 login: root (automatic login)
pc1:~#
```

接下来你可以把 pc1 当作一个真实的 linux 系统操作了，比如配置刚才设置的两块网卡。

```
Ifconfig eth0 10.0.0.1 netmask 255.255.255.0 broadcast 10.0.0.255 up
```

```
Ifconfig eth1 11.0.0.1 netmask 255.255.255.0 broadcast 11.0.0.255 up
```

linux 自带了很多网络相关的命令，可以配置路由表，运行路由协议等，这样就可以用虚拟 linux 机模拟一个路由了。

在需要搭建复杂一些的拓扑时，我们有更自动化的方式，见下文。

netkit 的 vlist 命令可以察看当前有哪些虚拟机在运作，使用 vhalt pc1 可以关闭 pc1 虚拟机。

Quagga 介绍

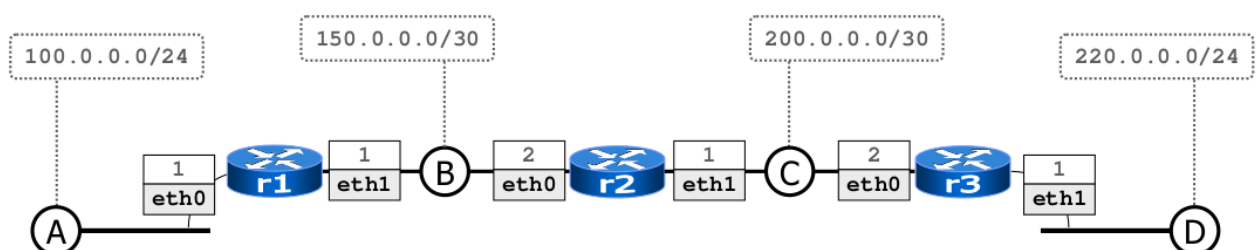
Quagga 是一套路由模拟软件，它可以把一台 linux 机变成一台路由器。主要由两部分组成，一是各种路由协议如(RIP,BGP,OSPF)，一是 zebra 管理软件。路由协议与 zebra 是松耦合关系，启动 zebra 管理进程后，任意启动一个路由协议进程即可。

最后，可以利用 telnet localhost zebra 或者 telnet localhost ripd 等来管理 zebra 或路由协议，实现与操作路由器类似的功能。

Quagga 的具体安装使用不在本文范围内，这里需要了解的是，Quagga 可以通过预先设置配置文件，在启动时就完成模拟路由器的配置工作。我们可以写好配置文件，在启动虚拟机时将对应的配置文件压入相应的虚拟机文件系统，来实现快速的把虚拟机变成虚拟路由。

进行一个 Netkit+Quagga 实验【参见 4】

我们的任务是配置如下拓扑：



思路如下：

启动三个虚拟机，每个虚拟机配置两个网络端口。

在虚拟机启动之后，配置每个网络端口。
启动 Quagga,运行一个路由协议，配置路由表。

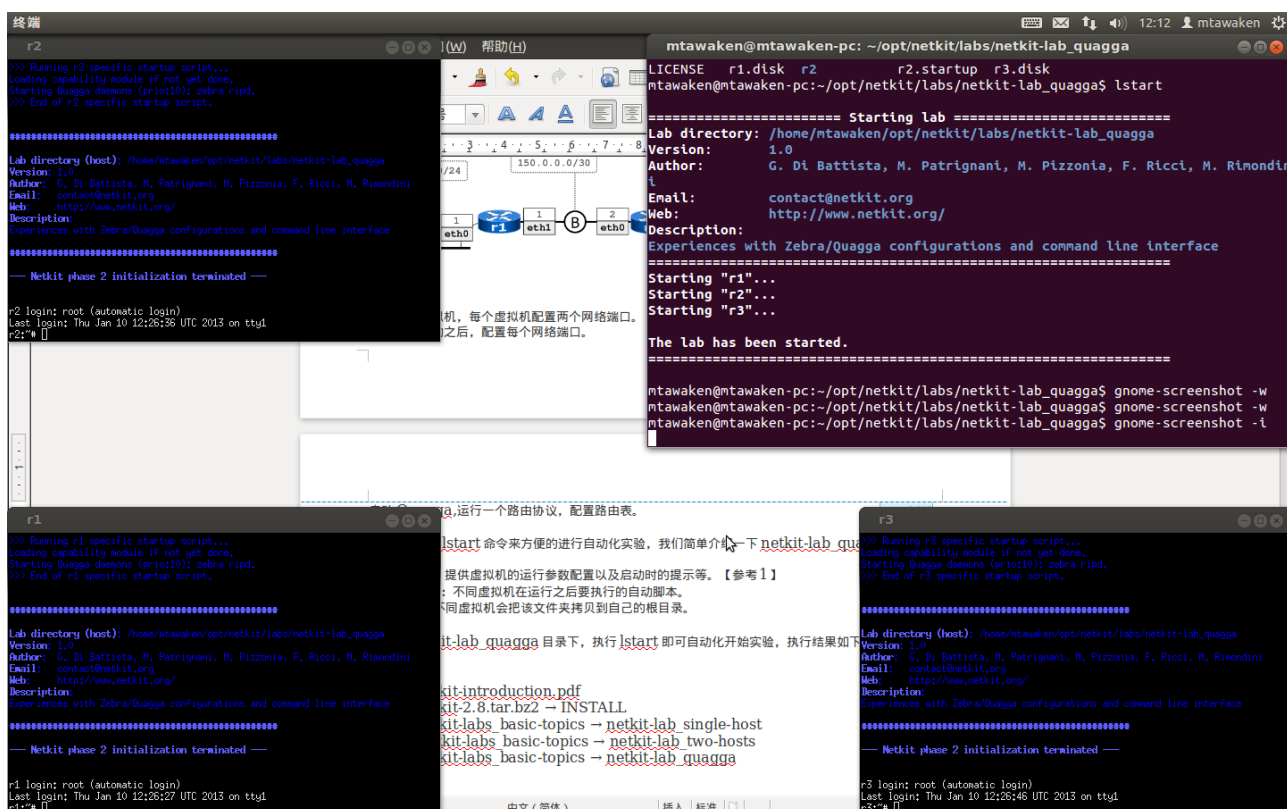
netkit 提供 lstart 命令来方便的进行自动化实验，我们简单介绍一下 netkit-lab_quagga 实验中的文件意义。

Lab.conf: 提供虚拟机的运行参数配置以及启动时的提示等。【参考 1】

r*.startup: 不同虚拟机在运行之后要执行的自动脚本。

r*文件夹: 不同虚拟机会把该文件夹拷贝到自己的根目录。

进入到 netkit-lab_quagga 目录下，执行 lstart 即可自动化开始实验，执行结果如下：



```
mtawaken@mtawaken-pc: ~/opt/netkit/labs/netkit-lab_quagga
LICENSE r1.disk r2 r2.startup r3.disk
mtawaken@mtawaken-pc:~/opt/netkit/labs/netkit-lab_quagga$ lstart

===== Starting lab =====
Lab directory: /home/mtawaken/opt/netkit/labs/netkit-lab_quagga
Version: 1.0
Author: G. Di Battista, M. Patrignani, M. Pizzonia, F. Ricci, M. Rimondini
Email: contact@netkit.org
Web: http://www.netkit.org/
Description: Experiences with Zebra/Quagga configurations and command line interface

Starting "r1"...
Starting "r2"...
Starting "r3"...

The lab has been started.

mtawaken@mtawaken-pc:~/opt/netkit/labs/netkit-lab_quagga$ gnome-screenshot -w
mtawaken@mtawaken-pc:~/opt/netkit/labs/netkit-lab_quagga$ gnome-screenshot -i
```

对于每个虚拟机，telnet localhost zebra 进入 quagga，使用 show interface eth0，show ip route 等命令察看拓扑是否已经搭建完成。

察看每个控制台的启动信息，可以发现 lstart 之后每个虚拟机的开机过程：

加载 UML，filesystem，加载 kernel 配置信息，加载 labs.conf 里对虚拟机的配置信息，例如配置网卡，执行各自的 script，包含配置网卡 IP 和启动 zebra rip 工作。

参考：

- 【1】：netkit-introduction.pdf
- 【2】：netkit-2.8.tar.bz2 → INSTALL
- 【3】：netkit-labs_basic-topics → netkit-lab_single-host
netkit-labs_basic-topics → netkit-lab_two-hosts
- 【4】：netkit-labs_basic-topics → netkit-lab_quagga