Università degli Studi Roma Tre
Dipartimento di Informatica e Automazione
Computer Networks Research Group

netkit

Sistema simple para experimentar en redes de ordenadores

Version	2.2
Author(s)	G. Di Battista, M. Patrignani, M. Pizzonia, M. Rimondini
E-mail	contact@netkit.org
Web	http://www.netkit.org/
Description	Una introducción a la arquitectura, puesta en marcha y uso de Netkit

copyright notice

- All the pages/slides in this presentation, including but not limited to, images, photos, animations, videos, sounds, music, and text (hereby referred to as "material") are protected by copyright.
- This material, with the exception of some multimedia elements licensed by other organizations, is property of the authors and/or organizations appearing in the first slide.
- This material, or its parts, can be reproduced and used for didactical purposes within universities and schools, provided that this happens for non-profit purposes.
- Information contained in this material cannot be used within network design projects or other products of any kind.
- Any other use is prohibited, unless explicitly authorized by the authors on the basis of an explicit agreement.
- The authors assume no responsibility about this material and provide this material "as is", with no implicit or explicit warranty about the correctness and completeness of its contents, which may be subject to changes.
- This copyright notice must always be redistributed together with the material, or its portions.

Sobre redes de ordenadores

- Las redes de ordenadores son (habitualmente) bastante complejas
  - Muchos dispositivos (ordenadores, routers, ...)
  - Muchos interfaces
  - Muchos protocolos en ejecución
  - Las conexiones interconexiones originan topologías complejas

¿Cómo realizar experimentos?

- Realizar experimentos puede ser inviable
- La red de producción no puede ser usada para experimentar
  - Los servicios en los sistemas pueden ser críticos para la compañía
  - Puede ser necesario coordinar los diferentes departamentos de la compañía
- Los equipos de red son caros
  - A veces, para realizar experimentos simples, muchos equipos deben estar disponibles en el mismo laboratorio

Simulación frente a emulación

- Emulación y simulación sistemas ponen a disposición de los usuarios un entorno virtual para pruebas, experimentos y medidas
- Los sistemas de simulación apuntan a reproducir el rendimiento de uns sistema real (latencia, pérdida de paquetes, etc.)
  - P.e.: ns, real, …
- Los sistemas de emulación apuntan a reproducir con precisión las funcionalidades de un sistema real (configuraciones, arquitecturas, protocolos), con una atención limitada al rendimiento

Netkit: un sistema para emular redes de ordenadores

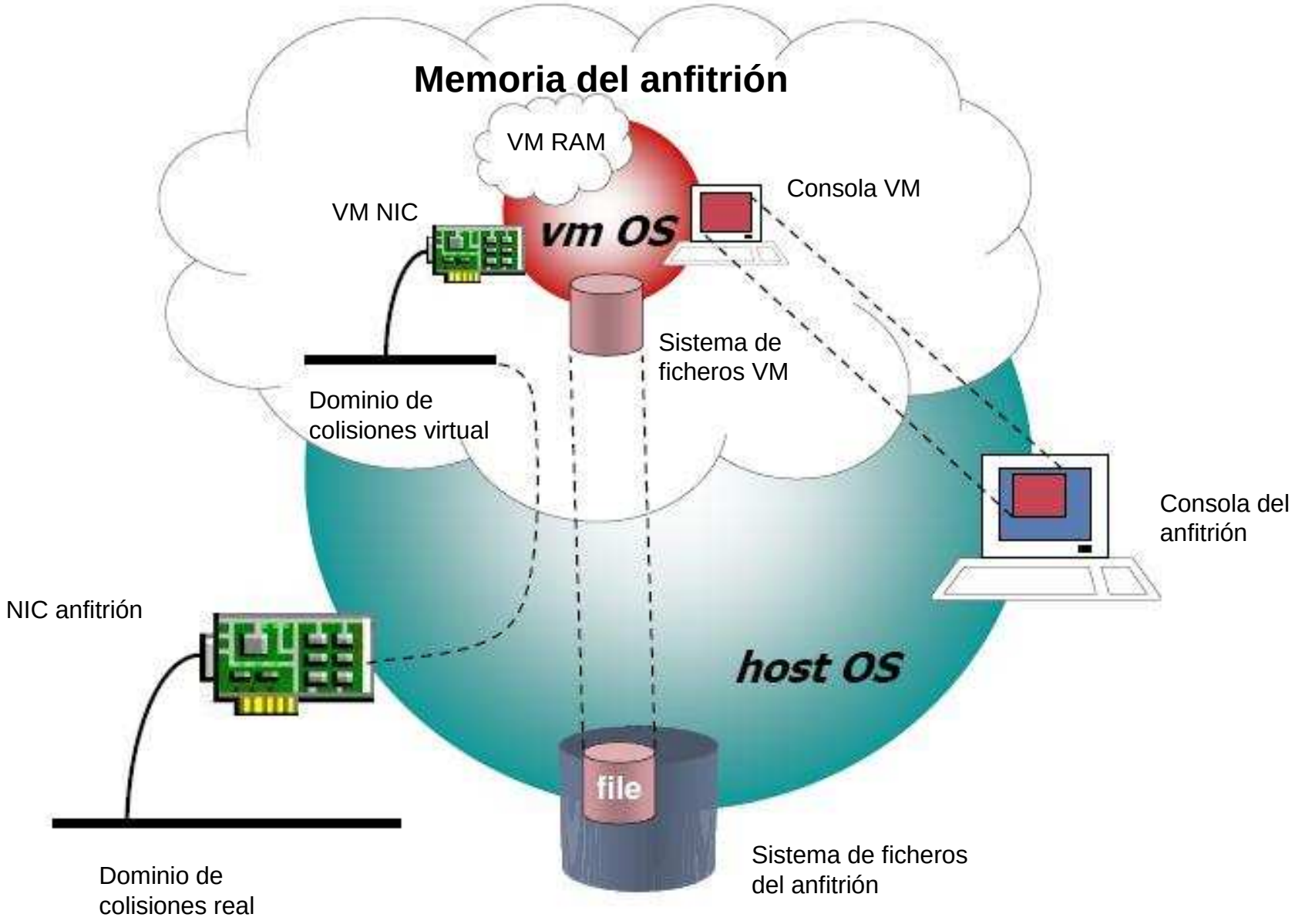
- Basado en UML (user mode linux)
  - http://user-mode-linux.sourceforge.net/
- Cada dispositivo de red emulado es una caja virtual linux
  - Una caja virtual linux está basada en el kernel UML
- El sistema operativo Linux dispone de soporte para la mayor parte de los protocolos de red
  - Un sistema Linux puede ser configurado como enrutador o puente/conmutador

User-mode Linux

- User-mode linux es un kernel de Linux (el núcleo del sistema operativo Linux) que puede ser ejecutada como un proceso de usuario en un caja estándar Linux
- Un proceso UML es llamado también máquina virtual (VM), mientras que la caja Linux anfitrión de la VM se llama máquina anfitrión (host)
- Muchas máquinas virtuales pueden ser ejecutadas al mismo tiempo en el mismo host

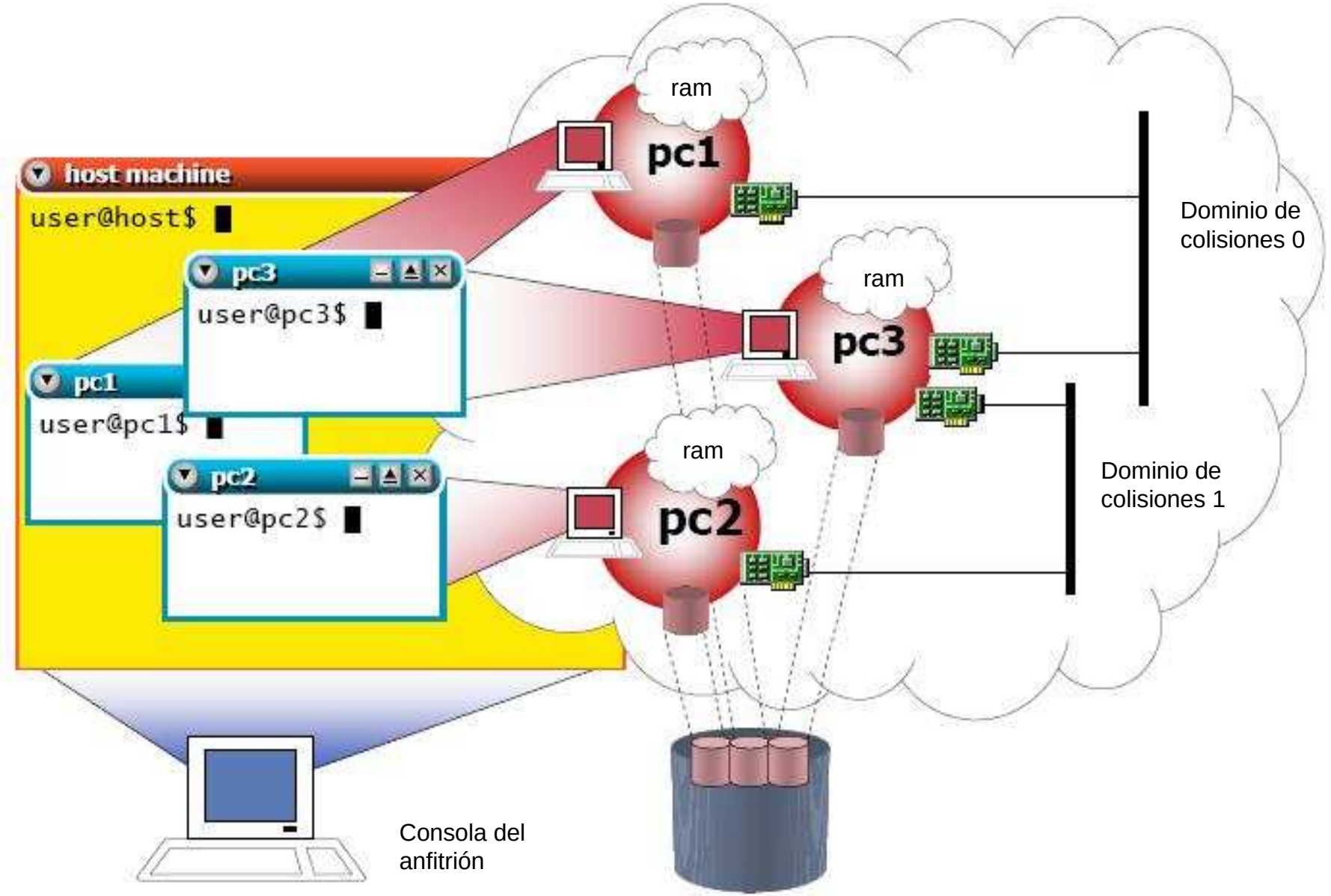
Máquinas virtuales UML

- Cada máquina virtual tiene
  - Una consola (una ventana terminal)
  - Una memoria (incrustada en la memoria del anfitrión)
  - Un sistema de ficheros (almacenada en un solo fichero en el sistema de ficheros del anfitrión)
  - Uno ó más interfaces de red
- Cada interfaz de red puede estar conectado a un (virtual) dominio de colisión
- En cada dominio de colisión puede estar conectado a muchos interfaces



Emulando una red de ordenadores con UML

- Idea básica:
  - Muchas máquinas virtuales se crean dentro de la máquina del anfitrión
  - Las máquinas virtuales se conectan a dominios de colisión virtuales y así se comunican entre ellas
- Cada máquina virtual puede ser configurada para desempeñar el papel de un sistema, de un enrutador, o de un conmutador



¿Qué es Netkit?

- Un conjunto de herramientas y comandos que puede ser usados para poner en marcha una red virtual de ordenadores
  - La mayor parte de los comandos se implementan como scripts
- Un sistema de ficheros preparado como patrón para crear el sistema de ficheros para cada VM
  - Las herramientas más usadas en redes están ya instaladas en este sistema de ficheros
- Un kernel UML que se usa como kernel las máquinas virtuales

Poniendo en marcha Netkit

Poniendo en marcha Netkit

- Descargar en http://www.netkit.org/
- Requisitos de Hardware
  - I386 32 bit arquitectura
  - ≥600 MHz CPU
  - ~10 MB de memoria por cada VM (dependiendo de la configuración de la VM)
  - ~600 MB de espacio de disco + ~1-20 MB por cada VM (depende del uso de la VM)
- Requisitos de Software
  - Una caja Linux
  - Funciona bien en muchas distribuciones, ver http://www.netkit.org/status.html
  - Estándar, y habituales herramientas de sistema (awk, lsof, etc.)

Poniendo en marcha Netkit

- Descargar los siguientes tres ficheros que componen la distribución
  - netkit-X.Y.tar.bz2
  - netkit-filesystem-FX.Y.tar.bz2 (atención: >100MB)
  - netkit-kernel-KX.Y.tar.bz2
- Desempaquetarlos en el mismo directorio
  - tar xjf netkit-X.Y.tar.bz2
  - tar xjf netkit-filesystem-FX.Y.tar.bz2 (puede llevar un tiempo xconsiderable; atención: descomprimido ocupa mas de 600MB)
  - tar xjf netkit-kernel-KX.Y.tar.bz2

Poniendo en marcha Netkit

- Configurar la shell para poner las siguientes variables de entorno
  - NETKIT_HOME debe apuntar al directorio que contiene la versión descomprimida de Netkit "$NETKIT_HOME/bin" debe añadirse a la variable PATH
  - ":$NETKIT_HOME/man" debe añadirse a la variable MANPATH
- por ejemplo (asumiendo que se usa bash)
  - export NETKIT_HOME=~/netkit2
  - export PATH=$PATH:$NETKIT_HOME/bin
  - export MANPATH=:$NETKIT_HOME/man

Poniendo en marcha Netkit

- Se puede comprobar la configuración entrando al directorio de Netkit …
  - cd $NETKIT_HOME
- … y ejecutando check_configuration.sh
  - ./check_configuration.sh
- Si todas las comprobaciones tienen éxito, está preparado para usar Netkit

Usando Netkit

Comandos de Netkit

- Netkit proporciona dos conjuntos de comandos
  - Comandos con prefijos en v (comandos-v)
  - Comandos con prefijos en l (comandos-l)
- Los comandos-v actúan como herramientas de baja nivel para configurar y arrancar máquinas virtuales
- Los comandos-l proporcionan un entorno fácil de usar para poner en marcha laboratorios complejos constituidos por muchas máquinas virtuales

Comandos-V Netkit

- Permite arrancar máquinas virtuales con configuraciones arbitrarias (memoria, interfaces de red, etc.).
  - vstart: arranca una nueva máquina virtual
  - vlist: ofrece una lista de las máquinas virtuales activas
  - vconfig: enlaza interfaces de red a máquinas virtuales
  - vhalt: para correctamente una máquina virtual
  - vcrash: para abruptamente una máquina virtual
  - vclean: comando tipo botón del pánico limpiando los procesos del entorno de Netkit (incluyendo las máquinas virtuales) y los valores de la configuración en el anfitrión

Comandos-L Netkit

- Fácil puesta en marcha de laboratorios complejos formados por muchas máquinas virtuales
  - lstart: arranca un laboratorio Netkit
  - lhalt: parada corracta de todaslas máquinas virtuales de un laboratorio
  - lcrash: parada abrupta de todas las máquinas virtuales de un laboratorio
  - lclean: borra los ficheros temporales del directorio de un laboratorio
  - linfo: proporciona información sobre un laboratorio sin arrancarlo
  - ltest: permite ejecutar comprobaciones para verificar que el laboratorio está funcionando apropiadamente

Accediendo al "mundo exterior" desde una máquina virtual

- Dos modos de hacerlo
  - El directorio **/hosthome** en la máquina virtual apunta directamente la directorio home del usuario en el anfitrión
    - El acceso en lectura/escritura está permitido
  - vstart puede configurar automáticamente túneles ("tap interfaces") a través de los cuales una máquina virtual puede acceder a una red externa
    - ver **man vstart** para más información

Preparando un laboratorio Netkit

Preparando un laboratorio

- Un laboratorio de Netkit es un conjunto de máquinas virtuales preconfiguradas que pueden ser arrancadas y paradas conjuntamente
- Puede ser implementadas de (al menos) dos maneras
  - Escribiendo un script **lab-script** que llama un comando **vstart** para cada máquina virtual a arrancar
  - Poniendo en marcha un laboratorio estándar que puede ser lanzado usando comandos-l (recomendado)

Un laboratorio Netkit como un script único

- Un script (p.e., **lab-script**) llama a **vstart** con algunas opciones para arrancar cada máquina virtual
- Usando la opción **--exec** de **vstart**, elmismo script puede ser llamado desde dentro de una máquina virtual (p.e., para configurar automáticamente los interfaces de red)
- Una comprobación en **lab-script** puede ser usado para comprobar si estamos en el anfitrión o dentro de una máquina virtual

Un laboratorio Netkit como un script único

- ejemplo

```
vstart pc1 --eth0=0 --eth1=1 --exec=this_script
vstart pc2 --eth0=0 --exec=this_script
vstart pc3 --eth0=1 --exec=this_script
if [ `id –u` == “0” ]; then
        case “$HOSTNAME” in
                pc1)
                        ifconfig eth0 10.0.0.1 up
                        ifconfig eth1 10.0.0.2 up;;
                pc2)
                        ifconfig eth0 10.0.0.3 up;;
                pc3)
                        ifconfig eth0 10.0.0.4 up;;
        esac
fi
```

Laboratorios Netkit usando comandos-l

- Un laboratorio Netkit estándar es un árbol de directorio conteniendo:
  - Un fichero lab.conf describiendo la topología de red
  - Un conjunto de subdirectorios que contienen los valores de configuración para cada máquina virtual
  - Los ficheros .startup y .shutdown que describen las acciones realizadas por las máquinas virtuales cuando son arrancadas o paradas
  - [opcional] un fichero lab.dep describiendo las relaciones del orden de arranque de las máquinas virtuales
  - [opcional] un directorio _test conteniendo scripts para comprobar que el laboratorio está trabajando correctamente

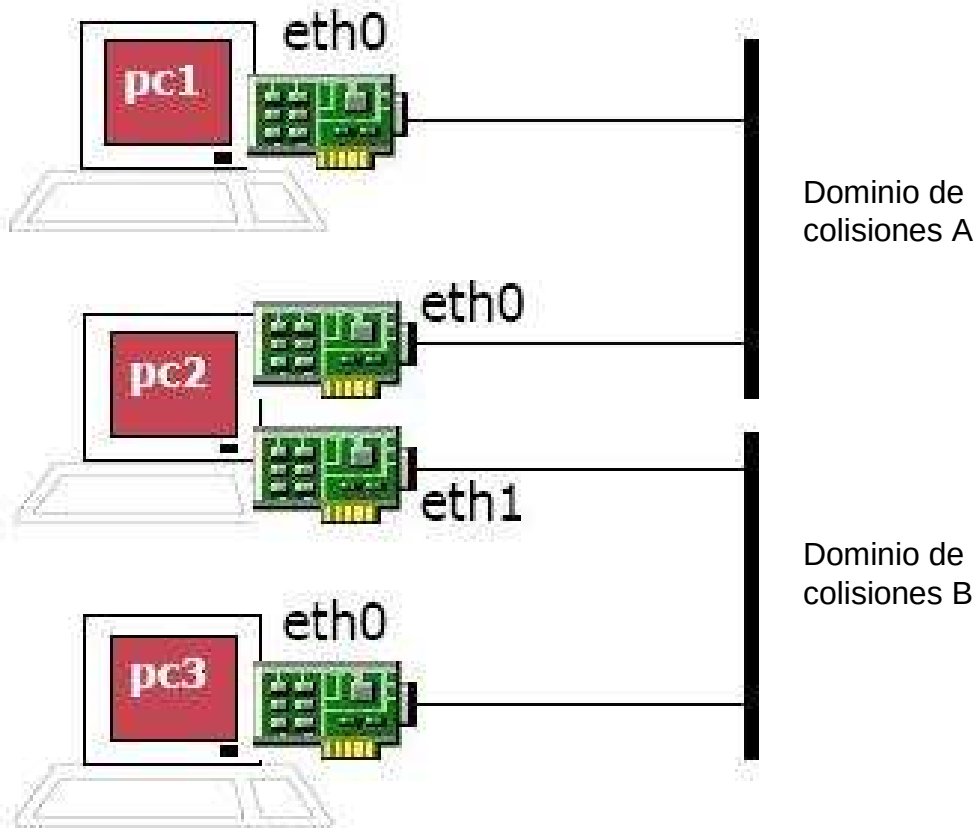
Lab.conf

- Este fichero describe
  - Los valores de las máquinas virtuales que forman un laboratorio
  - La topología de la red que interconecta las máquinas virtuales del laboratorio
- Lista de las asignaciones **machine[arg]=value**
  - **machine** es el nombre de la máquina virtual (p.e., **pc1**)
  - Si **arg** es número entero (como **i**), entonces **value** es el nombre del dominio de colisión al que el interfaz **eth i** debe ser conectado
  - Si **arg** es una cadena, entonces debe ser el nombre de una opción de **vstart** y **value** es el argumento (si tiene) de la opción

Lab.conf

- Ejemplo

```
pc1[0]=A

pc2[0]=A
pc2[1]=B
pc2[mem]=256

pc3[0]=B
```



pc2 tiene 256 MB de memoria virtual

Lab.conf

- Otras asignaciones opcionales
  - **machines="pc1 pc2 pc3..."**: declara explícitamente las máquinas virtuales que conforman el laboratorio
    - Por defecto, la existencia de un subdirectorio **vm_name** en el directorio del laboratorio implica que una máquina virtual **vm_name** está arrancada
  - **LAB_DESCRIPTION**
  - **LAB_VERSION**
  - **LAB_AUTHOR**
  - **LAB_EMAIL**
  - **LAB_WEB**

Información descriptiva mostrada cuando el laboratorio se arranca

Subdirectorios del laboratorio

- Netkit arranca una máquina virtual por cada subdirectorio, con el mismo nombre del subdirectorio
  - A menos que **lab.conf** contenga una sentencia **machines=**
- El contenido del subirectorio **vm** se mapea (=copia) en el raíz (**/**) del sistema de ficheros de la máquina virtual
  - Por ejemplo, **vm/foo/file.txt** es copiado a **/foo/file.txt** dentro de la máquina virtual **vm**
  - Esto sólo pasa la primera vez que **vm** es arrancada; para forzar el mapeo es necesario borrar el sistema de ficheros de la máquina virtual (fichero **.disk**)

Ficheros de arranque y parada

- Scripts de shell que dicen a las máquinas virtuales que hacer cuando arrancan o paran
- Son ejecutados dentro de las máquinas virtuales
- **shared.startup** y **shared.shutdown** afectan a todas las máquinas virtuales
- En el arranque, una máquina virtual llamada **vm_name** ejecuta
  - **shared.startup**
  - **vm_name.startup**
- En la parada, una máquina virtual llamada **vm_name** ejecuta
  - **vm_name.shutdown**
  - **shared.shutdown**

Ficheros de arranque y parada

- Un uso habitual del fichero .startup es configurar los interfaces de red y/o arrancar servicios de red
- Ejemplo de **vm_name.startup**

```
ifconfig eth0 10.0.0.1 up
/etc/init.d/zebra start
```

Lab.dep

- Múltiples máquinas virtuales pueden arrancar a la vez (arranque paralelo)
  - **-p** opción de lstart
- El orden de arranque de las máquinas virtuales pueden ser influenciadas estableciendo dependencias
  - P.e., "pc3 pueden arrancar sólo después de que pc2 y pc1 estén arrancadas"
- Un fichero lab.dep dentro del directorio del laboratorio describe las dependencias y permite el arranque paralelo
  - El formato del fichero es similar al de Makefile
  - ejemplo

```
pc3: pc1 pc2
```

Lanzando/parando un laboratorio

- ***lcommand* -d <lab_directory> [machine...]**
- o
  - Entrar en el directorio del laboratorio (**cd lab_directory**)
  - ***lcommand***
- Donde ***lcommand*** puede ser uno de los siguientes:
  - **lstart**, para arrancar el laboratorio
  - **lhalt**, para parar ordenadamente las máquinas virtuales de un laboratorio
  - **lcrash**, para parar abruptamente las máquinas virtuales de un laboratorio
- Opcionalmente, una lista de nombres de máquinas pueden ser dados en la línea de comandos, en este caso sólo estas máquinas estarán afectadas por el comando

Borrando los ficheros temporales

- Un laboratorio en ejecución crea varios ficheros temporales dentro del directorio actual, y del directorio del laboratorio
- Para deshacerse de todos ellos, usar **lclean** después de que el laboratorio haya sido parado
  - advertencia: **lclean** también borra los sistemas de ficheros de las máquinas virtuales (ficheros **.disk**); no debe usarse si se va a lanzar del laboratorio de nuevo usando los mismos sistemas de ficheros

ltest

- Hace fácil verificar que los laboratorios distribuidos trabajan correctamente
- Ltest arranca un laboratorio y vuelca información sobre cada máquina virtual **vm**
  - La salida se guarda en **_test/results/vm.default**
- [opción] un script **_test/vm.test** puede contener comandos adicionales a ser ejecutados dentro de **vm** para guardar otra información
  - La salida se guarda en **_test/results/vm.user**

ltest

- Ejemplo de un fichero **vm.default**

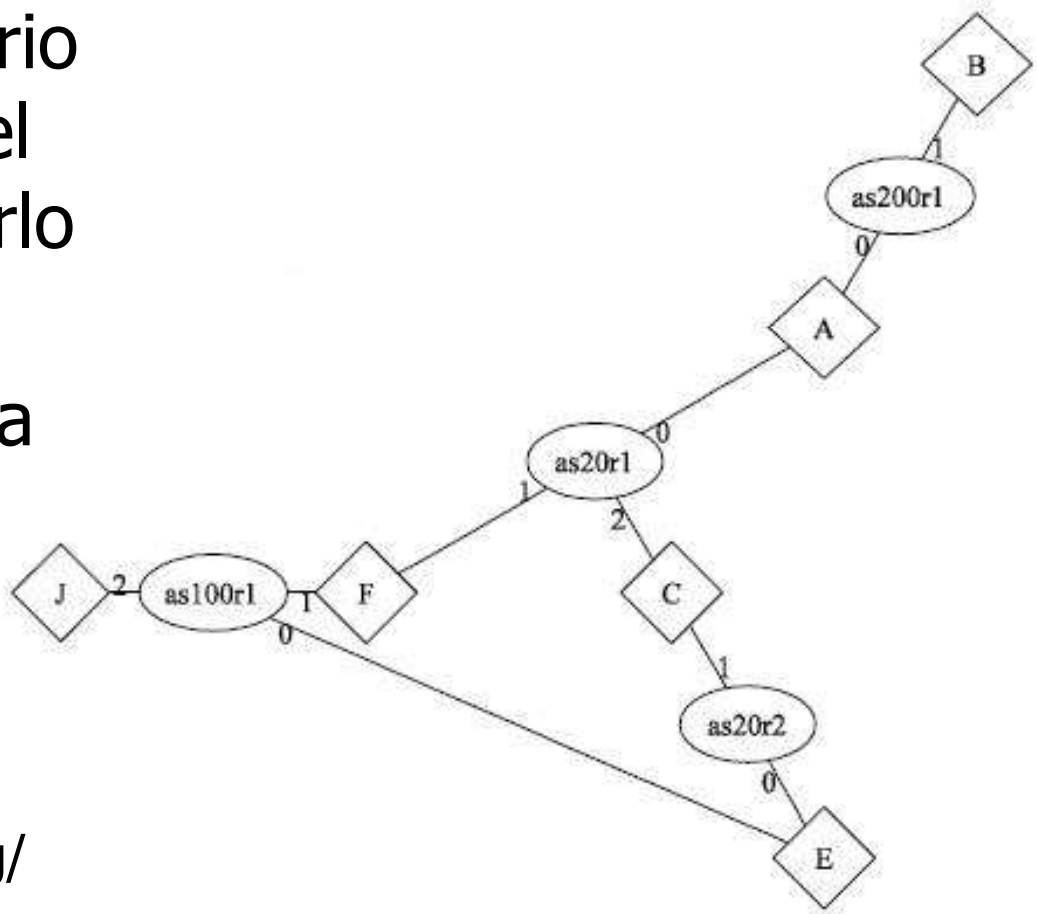
```
[INTERFACES]

Lo              Link encap:Local Loopback
                inet addr:127.0.0.1 Mask:255.0.0.0
                inet6 addr: ::1/128 Scope:Host
                UP LOOPBACK RUNNING MTU:16436 Metric:1

[ROUTE]

Kernel IP routing table
Destination Gateway Genmask Flags MSS Window irtt
Iface

[LISTENING PORTS]

Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address Foreign Address State

[PROCESSES]

UID COMMAND
                0 init [2]
                0 [ksoftirqd/0]
                0 [events/0]
.........
```

ltest

- Cuando se prepara un laboratorio
  - Lanzan ltest para guardar información del laboratorio
  - Mover los ficheros **_test/results/*** a un subdirectorio **_test/results/good**
- Cuando se prueba un laboratorio
  - Lanzar ltest para volcar información del laboratorio
  - Comparar los ficheros (p.e., usando diff) **_test/results/*** con **_test/results/good/***
  - Comprobar si todo coincide

Adquiriendo información sobre el laboratorio

- linfo imprime un sumario de información sobre el laboratorio sin ejecutarlo
- La opción **-m** permite crear un esquema de la topología a nivel de enlace del laboratorio
  - Requiere la librería GraphViz
    - http://www.graphviz.org/



Más información

- Más información puede ser encontrada …
- … en las páginas man (se puede arrancar con **man netkit**)
- … en el sitio web http://www.netkit.org/