

## Virtualización en la Educación: Laboratorio Portátil de Redes

Barrionuevo Mercedes<sup>1</sup>, Gil Cristian, Giribaldi Matias, Suarez Christopher,  
Taffermaberry Carlos<sup>12</sup>

<sup>1</sup> Universidad Nacional de San Luis,

Ejército de los Andes 950 - 5700 - San Luis - República Argentina

<sup>2</sup> Universidad Tecnológica Nacional, Facultad Regional Mendoza

Rodriguez 273, Ciudad Mendoza -M5502AJE - República Argentina

{mbarrio, ctaffer}@unsl.edu.ar

**Abstract.** El principal objetivo de la virtualización es la reproducción completa de una red de computadoras físicas en software, pudiendo trabajar con dispositivos y servicios de red lógicos. Ayudar a la metodología de enseñanza tradicional mediante la experimentación usando un laboratorio virtual y portátil permitirá realizar pruebas en cualquier momento y lugar. LPR (Laboratorio Portátil de Redes) es una herramienta que permite realizar un ambiente virtualizado de redes de computadoras permitiendo conectar, configurar y comunicar máquinas clientes y servidores. Su objetivo es permitirle al alumno un ambiente apto para verificar el correcto funcionamiento de los servicios de red que se necesiten. En este trabajo se presentan las consideraciones principales tanto del diseño como de la implementación de LPR, además de los resultados obtenidos.

**Palabras claves:** Virtualización. Servicios de Red. Software Libre. Redes de Computadoras. Educación.

### 1 Introducción

En la actualidad, nuestra sociedad experimenta Internet a través del uso del protocolo World Wide Web, los correos electrónicos, programas que permiten compartir información, etc., sin la necesidad de conocer cómo es que estas aplicaciones funcionan realmente. Sin embargo, a medida que avanzamos en el estudio de las redes de datos se vuelve cada vez más importante conocer cómo un servicio de red es capaz de transmitir e interpretar los mensajes enviados y recibidos a través de la red.

Poder configurar y probar el correcto funcionamiento de los servicios más comunes utilizados en Internet es fundamental para todo estudiante de Tecnologías de la Información y la Comunicación (TIC). Realizar diversas pruebas en un ambiente seguro, donde se pueda volver a una configuración inicial y estable, es un objetivo de toda cátedra dedicada a la puesta a punto de los servicios de red.

Permitir a diversas aplicaciones ejecutarse en una red virtual exactamente igual que en una red física es un logro de la virtualización de redes. Por otra parte, realizar diversas pruebas de configuración y ejecución de comandos o de aplicaciones

pudiendo guardar el estado actual y recuperar el trabajo realizado luego de un tiempo es una de las ventajas del uso de diversos entornos de trabajo virtualizados.

En base a lo mencionado anteriormente, se propone la creación de un Laboratorio Portátil de Redes, una herramienta fundamental para el desarrollo de las prácticas en las materias Servicios en SO de Redes y toda aquella materia de la Tecnicatura Universitaria en Redes de Computadoras (TUR) de la Universidad Nacional de San Luis que requiera su utilización, buscando ayudar a los profesores y estudiantes en el proceso de enseñanza-aprendizaje. LPR surgió como trabajo final de aplicación de los alumnos autores de este documento.

El resto de este documento está organizado de la siguiente manera: la próxima sección describe los conceptos teóricos involucrados en el desarrollo del prototipo. La sección 3 especifica aspectos básicos del diseño de la herramienta propuesta, detallando el contexto donde será utilizada y las problemáticas abordadas. En la sección 4 se muestran los ensayos y pruebas experimentales. A continuación se especifican los criterios utilizados en la selección de la tecnología a usar y la evaluación del prototipo. Finalmente, se detallan las conclusiones y trabajos futuros.

## 2 Marco Teórico

Esta sección introduce diferentes conceptos utilizados en el desarrollo de nuestro trabajo, destacándose la virtualización, tipos e implementaciones. A continuación, se enumeran los principales de ellos.

### 2.1 Virtualización

Normalmente, cuando se utiliza una computadora, el Sistema Operativo (SO) es instalado y ejecutado directamente sobre el hardware y de esta forma se aprovecha de forma completa su potencial. Sin embargo, existe un tipo de programa llamado software de virtualización, o máquina virtual, encargado de emular un hardware determinado, aprovechando los recursos reales de la computadora y, sobre este software, es posible instalar un SO como si se tratase de una computadora real. El SO huésped funciona, a grandes rasgos, como si fuera un SO principal, por lo que todas sus funciones y características están disponibles, convirtiéndolo en una herramienta ideal para realizar pruebas [1].

Cuando hablamos de virtualización hacemos referencia al proceso de reemplazar dispositivos físicos por dispositivos virtuales, disponibles mediante el uso de un software. Se pueden virtualizar servidores, estaciones de trabajo, redes y aplicaciones. Para ello, el software de virtualización administra los recursos físicos de esa máquina: memoria, CPU, almacenamiento y ancho de banda de la red, entre los aspectos más relevantes.

Junto con la disminución del uso de equipos físicos, la virtualización trae como beneficios la reducción de costos de mantenimiento y consumo energético. Esto deriva una consolidación de servidores, optimizando el uso del espacio físico.

## 2.2 Tipos de Virtualizaciones

Cuando se trata de la virtualización, existen varias maneras de lograr el mismo resultado a través de diferentes niveles de abstracción, siendo los más utilizados [2, 3]:

- *Paravirtualización*: Este mecanismo agrega un conjunto especial de instrucciones (llamado Hypercalls) reemplazando las instrucciones del set de instrucciones referidas a la arquitectura de la máquina real. En la arquitectura x86, por ejemplo, el hipervisor se ejecuta justo por encima del hardware físico (Ring 0) para que los SO huéspedes se ejecuten en niveles superiores. El núcleo de los SO huéspedes necesita ser modificado para poder utilizar los Hypercalls. Por otro lado, tiene como ventaja una sobrecarga de virtualización baja.
- *Nivel de SO*: Esta técnica virtualiza el servidor físico en el nivel del SO. Aquí, el SO anfitrión es un kernel modificado que permite la ejecución de múltiples contenedores aislados también conocidos como Virtual Private Servers (VPS) o servidores virtualizados. Cada contenedor es una instancia que comparte el mismo kernel del SO anfitrión. Tiene como ventaja una baja sobrecarga, y sus implementaciones son ampliamente utilizadas. El principal inconveniente es que no admite virtualizar otros tipos de SO.
- *Virtualización completa*: Esta técnica se utiliza para la emulación de una arquitectura de procesador sobre la arquitectura de otro. Permite ejecutar SO huéspedes no modificados, emulando cada instrucción del primero por medio de la traducción de código. La ventaja de la emulación completa del procesador es su portabilidad multiplataforma de aplicaciones y de SO. La desventaja es la sobrecarga causada por la emulación en software del conjunto completo de instrucciones.
- *Asistida por hardware*: Este tipo de virtualización implementa un nuevo anillo (Ring -1) con un modo de privilegio más alto en la arquitectura del procesador normal de cuatro anillos. Las extensiones de CPU para el soporte de virtualización permiten ejecutar los SO huéspedes sin modificación para que trabajen en Ring 0, mientras que el Hipervisor se ejecuta en Ring -1. Esto permite soportar la virtualización sin necesidad del uso de *Virtualización Completa* o *Paravirtualización*. La ventaja de esta técnica es la reducción de sobrecarga causada por la emulación de software.

## 2.3 Implementaciones de virtualización

Para poder implementar ambientes de virtualización existen diversas tecnologías. En esta subsección se hace una revisión de algunas tecnologías existentes:

- *KVM (Kernel based Virtual Machine)* [4]. Es una herramienta de software libre la cual permite la virtualización asistida por hardware, para plataformas Intel y AMD. Está incluida por defecto a partir del Kernel 2.6.20 de Linux, permitiendo así una rápida implementación. Esta es soportada por la comunidad y tiene una gran cantidad de herramientas de administración que pueden ser usadas en conjunto.
- *Xen* [5]. Es un hipervisor desarrollado con software libre para virtualización. La virtualización puede llevarse a cabo usando Paravirtualización o Asistencia por hardware. Tiene un gran soporte en la comunidad de desarrolladores y su implementación está madura, pues lleva más de 10 años en uso.
- *LXC (Linux Container)* [6]. Es una interface en espacio usuario para virtualización a nivel SO, y lo logra por medio de una potente API (Application Program Interface) y herramientas simples, permitiendo a los usuarios de Linux crear y manejar contenedores. El desarrollo de esta interfaz fue realizado utilizando software libre y el soporte está incluido actualmente en la mayoría de las distribuciones de Linux. Su desarrollo data del año 2008.
- *Docker* [7]. Es una herramienta de software libre que virtualiza en Nivel SO, permitiendo empaquetar entornos y aplicaciones que posteriormente se pueden desplegar en otro SO con esta tecnología. Docker accede a la virtualización del kernel de Linux a través de la biblioteca *libcontainer* [8], o indirectamente a través de LXC. El proyecto inició en el año 2013.
- *Proxmox* [9]. Es una plataforma de código abierto que no permite virtualizar por sí misma. Utiliza el hipervisor KVM para realizar virtualización asistida por hardware y LXC para hacer virtualización en nivel de SO. Dispone de una página web de administración muy completa, permitiendo realizar toda la administración necesaria, y está basada en la distribución Debian, disponible desde el año 2009.
- *VirtualBox* [10]. Es un software de virtualización que utiliza técnicas de asistencia de hardware y virtualización completa. El software es ofrecido bajo una licencia de software privativo, pero en el año 2007, después de años de desarrollo, surgió la versión Open Source Edition, VirtualBox OSE bajo la licencia GPL 2.
- *QEMU* [11]. Es una herramienta que permite virtualización y emulación de la arquitectura de hardware. Cuando es usado para virtualizar, QEMU utiliza la técnica de Virtualización Completa, mientras que cuando es usado como emulador de hardware, puede ejecutar programas para otra arquitectura. Puede utilizarse en conjunto con el Hipervisor Xen o KVM. La primera versión estable del software data del año 2006, por lo cual su desarrollo está maduro.

Si bien existen otros software de virtualización, solo se consideraron los anteriores por cuestiones relacionadas a licencias libres.

### 3 Diseño de un Laboratorio Portátil de Redes - LPR

Este trabajo tiene como objetivo general crear un ambiente donde se pueda construir una red de computadoras usando un método de virtualización. A continuación se

detalla el contexto donde va a ser utilizado, los problemas y motivaciones, los objetivos perseguidos y las tecnologías utilizadas para la creación del prototipo LPR.

### 3.1 Contexto

Un servicio de red es una aplicación que provee información a distintos clientes que la soliciten. Generalmente los servicios de red son instalados en uno o más servidores para compartir información y recursos a computadoras clientes.

El trabajo desarrollado tiene como objetivo ser utilizado en la materia Servicios de Sistemas Operativos correspondiente al 3° año de la Tecnicatura Universitaria en Redes. Esta materia tiene un alto porcentaje práctico dado que se basa principalmente en la configuración, puesta en marcha y prueba de los servicios.

Los servicios de red de interés para la materia, son: Protocolo de Configuración Dinámica de Host (DHCP), Correo electrónico (SMTP), Servicio de Nombre de Dominio (DNS), Protocolo de Transferencia de Hipertexto (HTTP), Proxy, Firewall y Redes Privadas Virtuales (VPN) y Server Message Block (SMB).

### 3.2 Problemática y motivación

Dada la naturaleza práctica de la materia, en su primer dictado durante el año 2009, se presentaron varios inconvenientes para llevar a cabo de manera eficiente dicha práctica en los laboratorios de la Facultad de Ciencias Físico Matemáticas y Naturales (FCFMyN) de la UNSL. Algunos de ellos se detallan a continuación:

- Limitación en cantidad de redes para realizar los prácticos de encaminamiento en capa de Red. Se necesitaban al menos dos redes físicas distintas o un dispositivo activo con soporte para crear distintas Vlan's pero no se contaba con ellos.
- Limitación en hardware con respecto a placas de red para computadoras multi-homed. Las máquinas que tenían el rol de encaminadores, necesitaban al menos dos placas de red, y no se contaban con ellas.
- Limitación en cantidad de computadoras, las cuales permitieran consumir servicios desde muchos clientes en forma simultánea.
- Inexistencia de software libre para instalar algunos servicios en el SO propietario instalado en las computadoras.
- Falta de permisos de administración en los SO de las computadoras del laboratorio para instalar y/o modificar los servicios a ensayar.
- Propagación de servicios de prueba a toda la red de la Facultad, como DHCP y DNS, entregando direcciones de red, puertas de enlace y resolución de nombres ficticias.
- Finalmente, si los alumnos intentaban recrear los mismos prácticos en sus hogares, se encontraban con las mismas o mayores limitaciones, debido a que generalmente contaban con una sola computadora.

Con el afán de mejorar las prácticas de laboratorio, y en particular el proceso de enseñanza-aprendizaje, surge la motivación para la realización de LPR.

### 3.3 Trabajos previos: Distribución ADIOS

Una alternativa a algunos de los problemas presentados en el punto 3.2 fue buscar entre diversos softwares la posibilidad de virtualizar escenarios de pruebas.

Se evaluaron distintas alternativas de simulación, como Cisco Packet Tracer [12], CLOONIX [13] o herramientas de virtualización como VNUML (Virtual Network User Mode Linux) [14] de la Universidad Politécnica de Madrid, o ADIOS [15] de la CQ University de Australia. Por funcionalidad y uso de escasos recursos de hardware se seleccionó ADIOS, la cual es una distribución basada en Fedora Release 8 live CD que utiliza User Mode Linux (UML) [16] para crear máquinas virtuales, tiene algunos servicios de los necesitados en la cátedra ya instalados, y una topología de red muy flexible [17], como se puede observar en la figura 1.

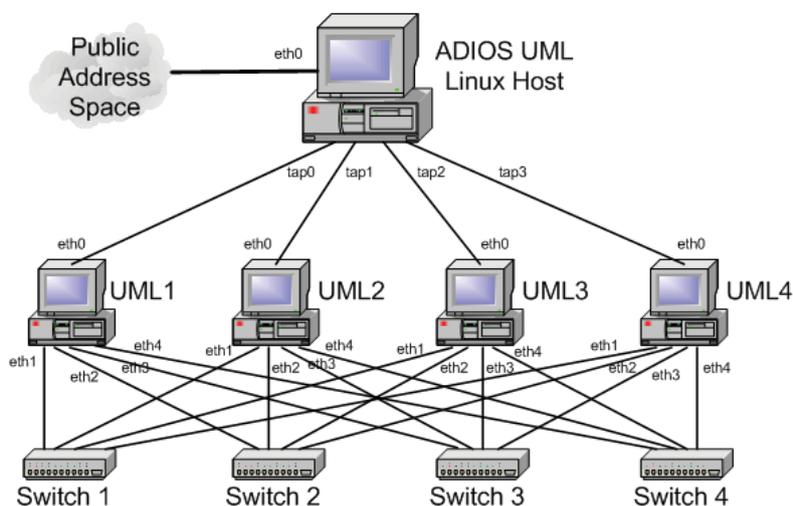


Figura 1. Detalle de la interconexión de las placas de red de las VM de ADIOS

Como la distribución es libre, fue posible hacer varias modificaciones a lo largo de los años, adaptándola a lo necesitado por la cátedra. Ellas son:

- Desasignación de direcciones de red preconfiguradas, e interfaces activas, para que los alumnos activen y configuren las necesarias en cada práctico.
- Instalación del servicio OpenLDAP, y clientes telnet, elinks, mutt y thunderbird.
- Solución de problemas con direcciones de IPv6 asignadas mediante EUI-64, debido a direcciones mac mal generadas.
- Creación de scripts para almacenar los cambios en los archivos de configuración e historial de comandos ejecutados en cada máquina virtual.
- Modificación de accesibilidad, como: tamaños de letra en las terminales y creación de una GUI para iniciar o detener las máquinas virtuales.

En la actualidad, la distribución (con un tamaño de 650 Mb) es totalmente funcional para realizar todas las prácticas de la cátedra, permitiendo utilizarla desde un liveCD. Por lo tanto, sin instalar software alguno, los alumnos pueden realizar los ensayos en

el laboratorio de la Facultad, o en su casa, si disponen de una computadora con lectora de CD y al menos 512 Mb de memoria RAM.

Sin embargo, surgieron nuevos problemas originados por la actualización del hardware en los laboratorios de la FCFMyN hace dos años. Se observó que el kernel de ADIOS (2.6.24) no reconoce algunos nuevos procesadores multi-núcleo, por lo que no funciona en ellos.

Junto con esta nueva limitación, la versión de los servicios sobre los que está desarrollando ADIOS tiene diez años, muchos de los servicios actualmente presentan nuevas características y son deseables de enseñar en la cátedra. Se intentó actualizar en ADIOS la versión de los distintos servicios utilizados, pero por problemas de dependencia de otros paquetes y del kernel mismo, fue imposible realizarlo.

### 3.4 Objetivos Propuestos:

Los objetivos y alcances específicos de LPR se detallan a continuación:

- Desarrollar una herramienta de software libre.
- Desplegar una red de máquinas virtualizada en una única computadora física, sin tener que instalar software alguno en la misma.
- Trabajar con al menos 6 nodos con distintos tipos de SO.
- Interconectar de diversas maneras los nodos creados para lograr distintas topologías de red en capa de enlace.
- Almacenar toda la configuración realizada en usos futuros.
- Crear una imagen de arranque en dispositivos tales como DVD o USB.
- Optimización de uso de recursos para que pueda ser utilizado en computadoras con limitaciones en el hardware.

### 3.5 Tecnologías propuestas para el desarrollo:

Para la implementación de los distintos nodos se propone la utilización de tecnologías de virtualización, lo que permite abstraer y aislar los distintos nodos entre sí. Entre ellas se destacan KVM, XEN, LXC, VirtualBox, Qemu, Uml.

La conectividad de capa de enlace entre nodos está dada por el uso de interfaces virtuales (vnics) y switch virtuales, como Open Vswitch. Un método alternativo a considerar es el uso de Vlans.

En cuanto a la implementación de servicios, se plantea la utilización de software GPL, el cual permite mayor flexibilidad a la hora de realizar configuraciones.

El almacenamiento en las máquinas virtuales se propone a través de la utilización de tecnologías de Snapshots, aplicadas en cada máquina virtual (dependiendo del hipervisor a usar), o mediante el uso de tecnologías de almacenamiento como LVM.

Con respecto a la optimización de recursos en las máquinas virtuales se propone el uso de una interface de tipo CLI (Character Line Interface), como el Bourne Shell.

## 4 Ensayos y pruebas

En esta sección se presentan las pruebas realizadas en cada una de las virtualizaciones. Para ello se utilizaron dos computadoras, con el siguiente hardware:

- Computadora 1: procesador AMD Athlon (tm) 64 X2 Dual Core con 2 Gb de Memoria Ram, almacenamiento en disco SAMSUNG HD080HJ de 8g Gb. Motherboard ASUS M2N-MX SE Plus.
- Computadora 2: procesador AMD Athlon 64 X2 Dual core 5000+ con 2Gb. de Memoria Ram, almacenamiento en disco de 160Gb. Motherboard ASUS M2N-MX SE Plus.

Por otra parte, los SO utilizados fueron Ubuntu 12.04 LTS, Debian 8 y Debian 9.

### 4.1 Tareas realizadas en cada ambiente de virtualización:

- Estudio de las tecnologías LXC, VNX, CLOONIX y Snapshots.
- Virtualización de redes LAN y WAN, probando la conexión entre los distintos hosts.
- Interconexión entre dos, cuatro y seis host virtuales.
- Verificación del consumo de recursos de las máquinas virtuales.
- Instalación, verificación y prueba de los servicios utilizados en la materia.
- Acceso e interacción a un escritorio remoto de una máquina virtual usando la herramienta de software libre VNCserver.

Siguiendo el esquema de red presentado por ADIOS, fue utilizado el soporte de Linux para Bridging y, de acuerdo a la documentación de Docker al respecto [18,19], se logró configurar exitosamente un ambiente con iguales características.

### 4.2 Comparativa de tecnologías:

Luego de la instalación de todos los requisitos para el correcto funcionamiento de cada una de las virtualizaciones se realizó una comparativa entre cada una de ellas. La Tabla 1 muestra en detalle la comparación realizada.

Características	KVM	VIRTUAL BOX	QEMU	LXC
<i>Espacio en Disco</i>	6 Gb por máquina virtual	8 Gb por máquina virtual	2 Gb por máquina virtual	488 Mb por contenedor
<i>Espacio en memoria RAM</i>	300 Mb por máquina virtual	512 Mb por máquina virtual	512 Mb por máquina virtual	14 Mb por contenedor
<i>Interconexión de host</i>	SI	SI	SI	SI

**Tabla 1.** Comparativa entre las virtualizaciones realizadas

## 5 Criterios de selección de tecnología y Evaluación del prototipo

Como uno de los objetivos planteados es que la práctica de laboratorio pueda ser reproducida por los alumnos en sus hogares, y como no todas las computadoras cuentan con soporte de hardware para virtualización, las técnicas que utilizan asistencia por hardware deben ser descartadas y, en consecuencia, KVM no sería utilizado.

Además, no es necesario emular procesadores con otra arquitectura en las prácticas planteadas, por lo cual el uso de QEMU para virtualizar no aporta ninguna mejora.

Finalmente, a partir de la comparación realizada en la Tabla 1, y teniendo en mente que debe poder ser ejecutado en hardware con pocos recursos, es que se decidió realizar el prototipo a evaluar utilizando la herramienta DOCKER.

El prototipo desarrollado está en proceso de evaluación. Si bien ha tenido una evaluación preliminar por parte de los integrantes de la cátedra, la evaluación real y exhaustiva se llevará a cabo cuando se presente el próximo año a los alumnos de la materia Servicios en SO de Redes. De ahí se podrán analizar las debilidades y fortalezas presentes en LPR.

Sin embargo, se pudo observar el correcto armado de diversas topologías de red, la correcta comunicación entre los distintos host y se pudieron probar todos los servicios dados en la materia. El prototipo se encuentra disponible para ser descargado y utilizado de forma libre y gratuita [20].

## 7 Conclusiones y futuros trabajos

LPR logró cumplir con la mayoría de los objetivos propuestos. Se considera que el trabajo realizado es de un alto grado de importancia en el ámbito de desarrollo las prácticas de la asignatura Servicios en SO de Redes, siendo una herramienta fundamental en el proceso de aprendizaje de los alumnos. Al ser un desarrollo de software libre, se permite el uso por otra cátedra que lo requiera o necesite, pudiéndolo modificar en base a otros requisitos específicos.

Por otro lado, se está analizando la posibilidad de poder adicionar como SO huésped Microsoft Windows, para poder desarrollar el práctico de SMB y ensayarlos sobre diferentes SO.

Se desea, además, que LPR pueda ser ejecutado desde un medio removible como un USB, para poder almacenar las nuevas configuraciones dado que fue el único objetivo todavía no alcanzado.

## References

1. Chowdhury, N. M. K., & Boutaba, R. (2010). A survey of network virtualization. *Computer Networks*, 54(5), 862-876.
2. Fernando Rodríguez-Haro, Felix Freitag, Leandro Navarro, A summary of virtualization techniques, *Procedia Technology*, Volume 3, 2012, Pages 267-272, ISSN 2212-0173

3. A. Souvik Pal , B. Prasant Kumar Pattnaik, Classification of Virtualization Environment for Cloud Computing, Indian Journal of Science and Technology Vol: 6 Issue: 1 January 2013 ISSN:0974-6846
4. Kernel Virtual Machine. web site: <https://www.linux-kvm.org/>
5. XEN Project. web site: <https://www.xenproject.org/>
6. LXC. Linux Containers. web site: <https://linuxcontainers.org/>
7. Docker Enterprise Edition. web site: <https://www.docker.com/>
8. Introducing Execution Drivers and Libcontainer. <http://blog.docker.io/2014/03/docker-0-9-introducing-execution-drivers-and-libcontainer/>
9. Proxmox. <https://www.proxmox.com/>
10. Virtualbox. <https://www.virtualbox.org/>
11. Qemu Process emulator. <http://www.qemu.org/>
12. Packet tracer. <http://www.packettracernetwork.com/>
13. Cloonix Routing and Network Simulation. <http://www.brianlinkletter.com/tag/cloonix/>
14. VNUML. Virtual Network UML [https://www.dit.upm.es/vnumlwiki/index.php/Main\\_Page](https://www.dit.upm.es/vnumlwiki/index.php/Main_Page)
15. N.Richter et al., ADIOS web site: <http://os.cqu.edu.au/adios>
16. The User Mode Linux Home-page. web site: <http://user-mode-linux.sourceforge.net/>
17. Adios y networking . Giacchini, M., INFN-LNL, L., & Richtes, I. N. LivEPICS: EPICS Official Distribution Made in LNL.
18. Marmol, V., Jnagal, R., & Hockin, T. (2015). Networking in containers and container clusters. Proceedings of netdev 0.1, February.
19. Docker Networking. <https://docs.docker.com/engine/userguide/networking/>
20. Mediafire. <http://www.mediafire.com/file/52tx4ra8j1pk1ov/livecd.iso>