FreeBSD Netzwerk Virtualisierung

Das Jail in FreeBSD wurde erheblich erweitert. Hier werden die neuen Funktionen und deren Anwendung vorgestellt. Insbesondere die Netzwerk Virtualisierung wird an Bedeutung gewinnen.

Betriebssystem: FreeBSD
Zeitrahmen: 45 min
Veranstaltung: GUUG−Frühjahrsfachgespräch 2010
Ort: Universität zu Köln, 28.05.2010

Author: Diplom Informatiker (FH) Dirk Meyer, Commiter im FreeBSD Projekt.
[dirk.meyer@dinoex.sub.org],[dirk.meyer@guug.de],[dinoex@FreeBSD.org]

Abstrakt (Teil 1)

Was sind Jails und warum sollte man sie einsetzen? Es gibt praktische Beispiele zur Trennung von Diensten und deren Verwaltung. Mit Jails V2 kann man jetzt mehre IP's pro Jail haben, und auch IPv6 innerhalb von Jails verwenden.

Eine gibt eine kurze Übersicht über die Systemwerkzeuge für Jails. Dem folgt das Einrichten und Überwachen von Jails mit SNMP.

Mit der Einrichtung der Netzwerk Virtualisierung geht es in neue Anwendungsfelder. Netzwerkkarten können exklusiv einem Jail zugewiesen werden, dieses Jail hat dann einen eignen Netzwerk−Stack und damit auch eine eigene Routing Tabelle. Damit ist es möglich mehrfache Uplinks, DSL oder Tunnel zu verwenden, ohne dass dieses miteinander oder mit dem Hauptsystem in Konflikt kommen. Auch mehrfache VPNs lassen sich so komplett abbilden.

Abstrakt (Teil 2)

Dann zur Vorstellung des IMUNES Netzwerk Emulators, der es erlaubt ganze Netzwerke unter FreeBSD zu simulieren. Hier wird die Vielseitigkeit dieses Tools an einigen Beispielen demonstriert. Es können Kapazitäten, Latenzen, Paketverluste, Routing und vieles mehr direkt erprobt werden. Im Gegensatz zu anderen Simulationen kann man einen Shell oder auch andere Programme auf den Knoten des Netzwerkes starten und direkt das Netzwerk benutzen.

Zum Abschluß wird ein Ausblick auf die aktuellen Entwicklungen für Jails und Netzwerk Virtualisierung gegeben. Hier sind zu nennen die Hierarchical Jails, FreeVPS, CPU and RAM Limits.

Übersicht

- 1. Jails
- 2. Netzwerk Virtualisierung
- 3. IMUNES
- 4. Ausblick

1. Jails

- 1.1. Was sind Jails
- 1.2. Multi−IPv4 Jails
- 1.3. IPv6 Jails
- 1.4. Weitere Features
- 1.5. Werkzeuge für Jails
- 1.6. SNMP Monitoring von Jails

1.1. Was sind Jails

Ein FreeBSD "jail" ist eine virtueller Bereich, der aus Sicht der Programme ein eigenständiges Betriebssystem ist. Es gibt jedoch nur einen einzigen Kernel, der alle Ressourcen verwaltet. Im Jail ist "root" nicht mehr allmächtig. Hier nur einige Punkte:

- Kein Zugriff auf Festplatten.
- Kein nachträgliches "mounten" von Dateisystemen.
- Es können nur die fest konfigurierten IP−Adressen von den Programmen im Jail verwendet werden.
- Kein Zugriff auf die Dateien und Prozesse außerhalb.
- Volle Performance der Applikationen

1.1.1 Anwendungen von Jails

Jeder Bereich hat seine eigenen Konfigurationsdateien, Programme, Bibliotheken und Daten. Damit sind folgende Möglichkeiten naheliegend:

- Administrative Aufgaben können verteilt werden.
  - Der "root" im Jail kann seinen Bereich verwalten.
- Umgebung zum Testen und Lernen
  - Benutzen von neuen Programmen parallel zum laufenden System.
- Virtualisierung von Diensten
  - Mail−Server, DNS−Server, Web−Server

1.2. Multi−IPv4 Jails

Seit FreeBSD 7.2 kann man einem Jail auch mehrere IP−Adressen zuweisen. Die IP−Adressen werden einfach beim Starten des Jails durch ein Komma getrennt. Ein Jail kann auch alle IP−Adressen des Hostsystems erben oder gar keine IP−Adresse haben.

- Alte Version

```
jail / jail1 127.0.0.1 /bin/csh
```

- Neue Version

```
jail −c path=/ host.hostname=jail1 ip4.addr=127.0.0.1 command=/bin/csh
```

1.3. IPv6 Jails

Seit FreeBSD 7.2 kann man einem Jail auch IPv6−Adressen zuweisen. IPv4 und IPv6 Adressen können natürlich gleichzeitig verwendet werden.

- Alte Version

```
jail / jail1 2001:1440:5001:1::2 /bin/csh
```

- Neue Version

```
jail −c path=/ host.hostname=jail1 ip6.addr=2001:1440:5001:1::2 \
command=/bin/csh
```

1.4.1 Weitere Features

Einige Optionen lassen sich beim Start des Jails festlegen:

- "securelevel=n" Setzt den Securelevel des Jails.

Folgende Optionen lassen sich per "sysctl" festlegen:

- "security.jail.sysvipc_allowed"
  - Erlaube System V IPC, z.B. für Postgres.
- "security.jail.allow_raw_sockets"
  - Erlaube Ping und Traceroute.
- "security.jail.chflags_allowed"
  - Erlaube das Ändern der Inumutebale Flags.
- "security.jail.mount_allowed"
  - Erlaube das Einbinden weiterer Dateisyteme.

1.4.2 Weitere Features

Mit Jails v2 können Optionen für jedes Jail einzeln gesetzt werden:

- allow.noset_hostname
  - Verbiete das ändern des Namens im Jail.
- allow.sysvipc
  - Erlaube System V IPC, z.B. für Postgres.
- allow.raw_sockets
  - Erlaube Ping und Traceroute.
- allow.chflags
  - Erlaube das Ändern der Inumutebale Flags.
- allow.mount
  - Erlaube das Einbinden weiterer Dateisyteme.

1.5.1. Werkzeuge für Jails

- jls

Anzeigen aller Jails und deren wichtigsten Daten.

```
t5# jls
   JID  IP Address      Hostname                      Path
     1  194.45.71.18    build8.dinoex.sub.de          /usr/jail/build
     2  194.45.71.35    geobaldi1.dinoex.sub.de       /
     3  92.79.54.35     geobaldi2.dinoex.sub.de       /
```

1.5.2. Werkzeuge für Jails

- jexec

Starten eines Programms innerhalb eines aktiven Jails. Im einfachsten Fall ist das eine Shell. Man kann aber auch jedes andere, im Jail vorhandene Programm starten.

```
t5# jexec 1 csh
build8# exit
t5#

t5# jexec 1 hostname
build8.dinoex.sub.de

t5# jexec 1 killall httpd
No matching processes were found
```

1.5.3. Werkzeuge für Jails

- /etc/rc.d/jail

Dieses Script erlaubt das Starten von Jails nach dem Start des Hosts. Die Konfiguration der Jails wird hierbei in der Datei /etc/rc.conf eingetragen.

```
jail_enable="YES"
jail_list="myjail"
jail_set_hostname_allow="NO"
jail_sysvipc_allow="NO"

jail_myjail_rootdir="/usr/jail/myjail"
jail_myjail_hostname="myjail.example.com"
jail_myjail_ip="172.16.0.3"
jail_myjail_exec="/bin/sh /etc/rc"
jail_myjail_devfs_enable="YES"
jail_myjail_mount_enable="NO"
```

1.5.4. Werkzeuge für Jails

- ezjail (/usr/ports/sysutils/ezjail)

Ein Framework zum einfachen Erzeugen und Verwalten von virtuellen Servern in Jails. Hier werden auch die benötigten Dateisysteme des Grundsystems nur einmal erzeugt und dann in alle Jails nicht beschreibbar eingeblendet.

Installation mit:

```
# pkg−add −r ezjail
```

1.5.5. Werkzeuge für Jails

Hier ein Beispiel mit 2 virtuellen Servern mit "ezjail" eingerichtet.

```
Filesystem            Size    Mounted on
/dev/mirror/gm0s1a     9.7G    /
devfs                  1.0K    /dev
/dev/mirror/gm0s1e      29G    /tmp
/dev/mirror/gm0s1d     9.7G    /usr
/dev/mirror/gm0s1f     850G    /var
/usr/ports             9.7G    /var/jails/build/usr/ports
/usr/jails/basejail    9.7G    /var/jails/www/basejail
devfs                  1.0K    /var/jails/www/dev
fdescfs                1.0K    /var/jails/www/dev/fd
procfs                 4.0K    /var/jails/www/proc
/usr/jails/basejail    9.7G    /var/jails/test/basejail
/usr/ports             9.7G    /var/jails/test/usr/ports
devfs                  1.0K    /var/jails/test/dev
fdescfs                1.0K    /var/jails/test/dev/fd
procfs                 4.0K    /var/jails/test/proc
```

1.6. SNMP Monitoring von Jails

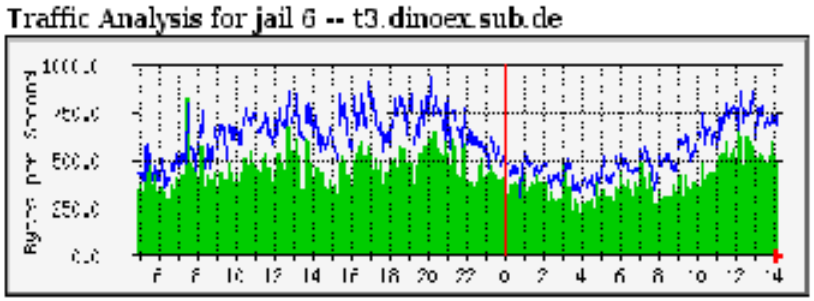
- bsnmp−jails (/usr/ports/net−mgmt/bsnmp−jails)

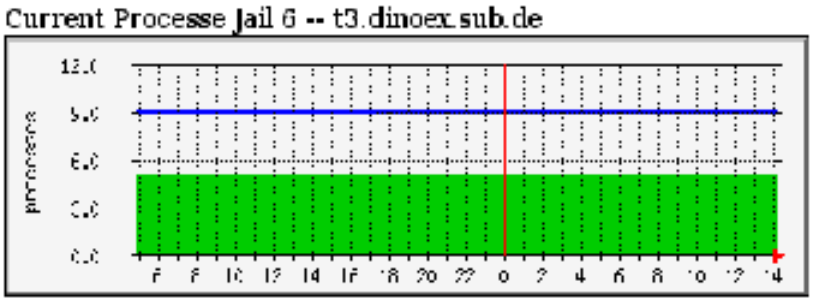
In FreeBSD ist ein SNMP−Server im Basissystem enthalten. In den Ports gibt es dafür ein Modul, mit dem man viele Daten der Jails erfassen kann. Mit MRTG oder anderen Programmen kann man daraus schöne Grafiken für die Auslastung bauen. Oder mit Nagios deren Funktion überwachen.

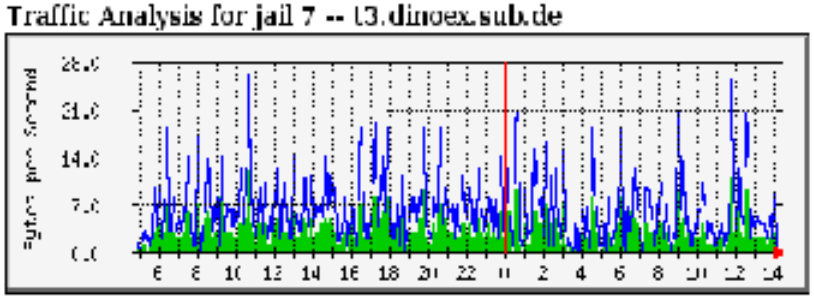
Beispiel der Ergänzung in /etc/snmpd.config

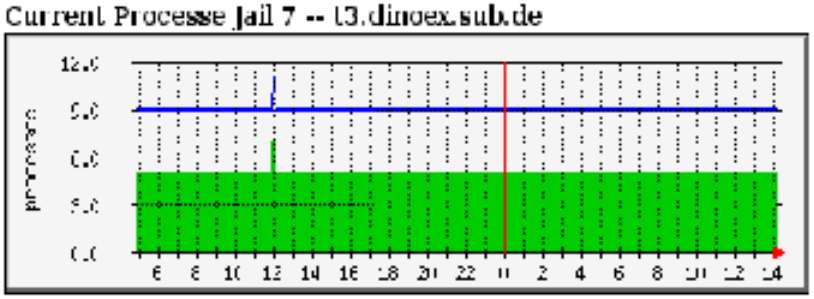
```
begemotSnmpdModulePath."jails" = "/usr/local/lib/snmp_jails.so"
%jails
jailNetworkFilter = "not net 10.0.0.0/8"
jailMeasureInterval = 360000
```

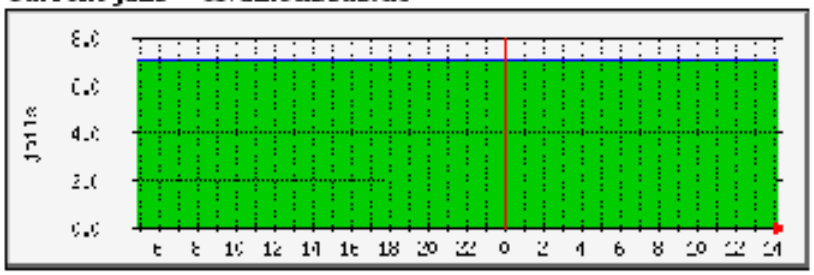
1.6.1 SNMP Monitoring von Jails











MRTG MULTI ROUTER TRAFFIC GRAPHER

version 2.16.2 Tobias Oetiker <oetiker@ee.ethz.ch> and Dave Rand <dlr@bungi.com>

2. Netzwerk Virtualisierung

- 2.1. Netzwerk Virtualisierung
- 2.2. Netzwerkkarten einem Jail zuweisen
- 2.3. PPPoE im Jail
- 2.4. Gleichzeitig mehrere VPNs
- 2.5. Firewalls

2.1. Netzwerk Virtualisierung

Die Netzwerk Virtualisierung in FreeBSD 8 erschließt völlig neue Anwendungsfelder. Für diese muß jedoch ein passender Kernel erzeugt werden.

Beispiel um einen erweiterten Kernel zu erstellen:

```
# cat > /usr/src/sys/amd64/conf/VIMAGE8
include GENERIC
ident VIMAGE8
nooptions SCTP
options VIMAGE
^D
# cd /usr/src && make KERNCONF=VIMAGE8 buildkernel
# cd /usr/src && make KERNCONF=VIMAGE8 installkernel
# shutdown −r now
```

2.1.1. Netzwerk Virtualisierung

Damit stehen jetzt Jails mit eigenem Netzwerk−Stack zur Verfügung. Diese Jails skalieren sehr gut, da sie nur ein wenig zusätzlichen Arbeitsspeicher benötigen.

Mit dem Netzwerk−Stack hat das Jail dann:

- eine eigene Routingtabelle
- ein eigenes Loopback−Interface
- eigene Speicher für die Netzwerkpakete
  - net.inet.tcp.recvbuf_max
- eigene Netzwerkparameter
  - net.inet.ip.ttl
- einen unabhängigen IP−Adressraum

2.1.2. Netzwerk Virtualisierung

Beim Starten des Jail mit der Option "vnet" wird ein virtuelles Netzwerk angelegt. Damit ist das Jail im Netzwerk ein vollständig unabhängiges System.

```
host:~# jail −c jid=9 vnet path=/ host.hostname=jail1 command=/bin/csh
jail1:# ifconfig
lo0: flags=8008<LOOPBACK,MULTICAST> metric 0 mtu 16384
        options=3<RXCSUM,TXCSUM>
jail1:# ifconfig lo0 127.0.0.1/8
jail1:# ifconfig
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> metric 0 mtu 16384
        options=3<RXCSUM,TXCSUM>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=3<PERFORMNUD,ACCEPT_RTADV>
```

2.2. Netzwerkkarten einem Jail zuweisen

Wenn das Jail existiert, so kann man eine oder mehrere Netzwerkkarten vom Host in das Jail verschieben. Bereits gesetzte IP−Adressen gehen dabei verloren, diese müssen dann im Jail konfiguriert werden.

```
host:~# jail −c jid=9 vnet path=/ host.hostname=jail1 command=/bin/csh
jail1:#

host:~# ifconfig epair0 create
host:~# ifconfig epair0a 10.0.0.1/24
host:~# ifconfig epair0b vnet 9
jail1:~# ifconfig epair0b 10.0.0.10/24
jail1:~# ping −t 1 10.0.0.1
jail1:~# route add default 10.0.0.1
```

2.2.1 Netzwerkkarten einem Jail zuweisen

Alternativ kann man auch ein vorhandenes Netzwerk in das Jail weiterreichen. Dabei verwenden wir eine "Bridge".

```
host:~# jail −c jid=9 vnet path=/ host.hostname=jail1 command=/bin/csh
jail1:#

host:~# ifconfig epair1 create
host:~# ifconfig epair1a up
host:~# ifconfig epair1b vnet 9
host:~# ifconfig bridge0 create
host:~# ifconfig bridge0 addm bge0 addm epair1a up
jail1:~# dhclient epair1b
jail1:~# ping −t 1 www.dinoex.de
```

2.3. PPPoE im Jail

PPPoE im Jail ist prinzipiell möglich, leider funktioniert es zur Zeit nur dann, wenn die Kernel−Module für Netgraph nach dem Erstellen des Jails im Host geladen werden. Der Fehler wird vermutlich mit FreeBSD 8.1 behoben sein.

Aber bereits jetzt es ist möglich in jedem Jail einen anderen Uplink oder Tunnel aufzubauen, ohne dass diese miteinander oder mit dem Hauptsystem in Konflikt kommen.

2.4. Gleichzeitig mehrere VPNs

In einem Jail kann man OpenVPN als Client starten. Damit kann man in unterschiedlichen Jails gleichzeitig mehrere VPNs benutzen. Je nachdem in welchem Jail man ein Programm startet, benutzt es das dort vorhandene Netzwerk.

Die VPN’s sind dadurch vom Host und anderen Jails isoliert. Damit kann man effektiv verhindern, dass andere Programme über das VPN Verbindungen aufbauen.

2.5. Firewalls

Mit dem eigenen Netzwerk−Stack hat ein Jail auch eine eigene Firewall. Zur Zeit kann man nur die BSD eigene Firewall ’ipfw’ einsetzen. So kann man gut den Fall demonstrieren, wenn ein Router unsinniger weise alle ICMP Pakete blockiert.

3. IMUNES

- 3.1. IMUNES Netzwerk Emulator
- 3.2. IMUNES Quality of Service
- 3.3. IMUNES Routing
- 3.4. IMUNES Screenshot

3.1.1. IMUNES Netzwerk Emulator

Im Jahr 2007 wurde an der Universität von Zagreb die Netzwerk Virtualisierung für FreeBSD 4.11 fertiggestellt. Auf Basis dieser wurde IMUNES entwickelt. IMUNES steht für Integrated Multiprotocol Network Emulator / Simulator.

Bei einer Simulation erzeugt IMUNES die passende Anzahl von Jails, und verbindet diese passend. Jedes Jail ist dann ein eigener Knoten im Netzwerk, in dem alle Systemfunktionen zur Verfügung stehen.

3.1.2. IMUNES Netzwerk Emulator

Unter IMUNES kann man alle Netzwerkfunktionen benutzen, alle simulierten Knoten existieren und der Netzwerkverkehr findet in Echtzeit statt. Die Diganose mit "tcpdump" oder "wireshark" ist jederzeit möglich.

- ARP, Ping, Ping6, Traceroute, Traceroute6

ARP kann man in der Emulation genauso beobachten, wie im realen Netzwerk. Ping und Traceroute für IPv4 und IPv6 funktionieren wie gewohnt.

3.2.1. IMUNES Quality of Service

- Bandbreite

Auf den Verbindungen kann man passend die Bandbreite begrenzen. So kann man das Verhalten an der Leistungsgrenze emulieren.

- Latenz

Auf den Verbindungen kann man Verzögerungen einstellen. Damit kann das Laufzeitverhalten von Weitverkehrsverbindungen nachgebildet werden.

3.2.2. IMUNES Quality of Service

Paketverluste

Mit der Fehlerrate kann man gestörte Leitungen oder Drahtlos−Netzwerk abbilden. Das Verhalten der Netzwerkprotokolle bie Paketverlusten ist bequem reproduzierbar. Bei einer TCP−Verbindung kann man die erneuten Übertragungen sehen, und die Windowsize gegebenenfalls optimieren.

3.3. IMUNES Routing

Weitere Beispiele:

- DHCP

Das Verhalten von DCHP Server und Client testen.

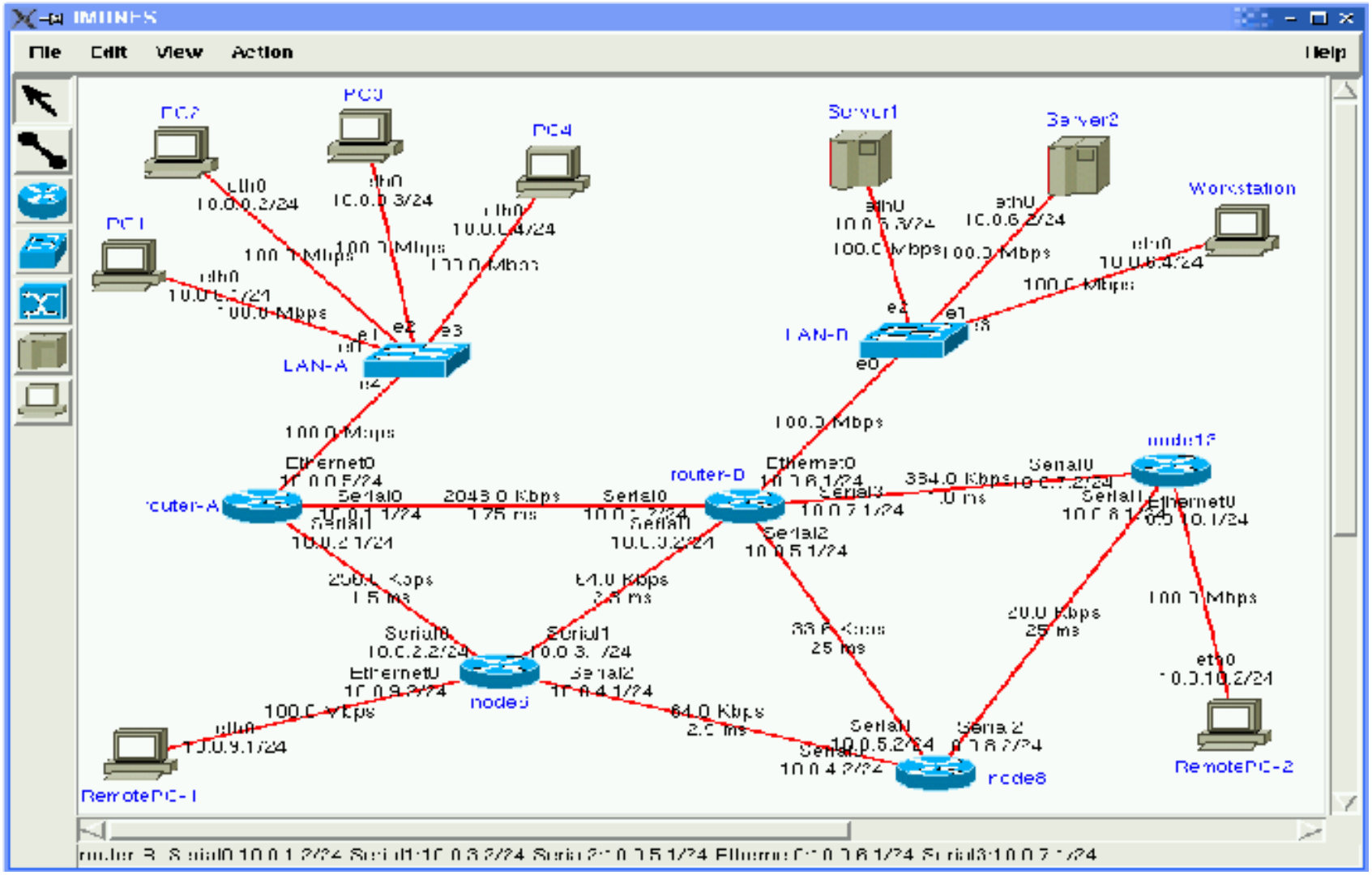
- IPsec

OpenVPN: VPN−Tunnel gefahrlos erproben.

- OSPF, RIP

Routingprotokolle einsetzen, redundante Wege ausprobieren.

3.4. IMUNES Screenshot



4.1. Ausblick

- Hierarchische Jails sind jetzt möglich.
- Verwalten der Quotas innerhalb eines Jails.
- Firewall "pf" im Jail.
- Ähnliche Projekte

http://www.sun.com/bigadmin/content/zones/index.jsp (Solaris)
http://sourceforge.net/projects/fvm−rni/ (XP)

4.2. Links

http://wiki.freebsd.org/NetworkVirtualization
http://imunes.tel.fer.hr/virtnet/
http://wiki.freebsd.org/Jails
http://en.wikipedia.org/wiki/FreeBSD_jail
http://www.freebsd.org/cgi/man.cgi?query=jail

http://www.freebsd.org/doc/de_DE.ISO8859−1/books/handbook/jails.htm
http://wiki.unixboard.de/index.php/FreeBSD_−_Jails

4.3. Quellen

http://phk.freebsd.dk/pubs/sane2000−jail.pdf
http://old.tel.fer.hr/imunes/dl/imunes.pdf
http://wiki.freebsd.org/Image
http://bsdbased.com/2009/12/06/freebsd−8−vimage−epair−howto
http://www.onlamp.com/pub/a/bsd/2003/09/04/jails.html
http://www.freebsddiary.org/jail−6.php Jail on FreeBSD 6
http://www.sufixo.com/articles/jails_barcamp06.pdf
http://www.section6.net/wiki/index.php/Creating_a_FreeBSD_Jail

4.4. Fragen & Feeback

- War es Interessant?
- Fragen?

Feedback: Dirk Meyer
[dirk.meyer@dinoex.sub.org],[dirk.meyer@guug.de],[dinoex@FreeBSD.org]