Marionnet: networking for dummies.

Marco Stronati

2009

Marionnet

a virtual network laboratory for teaching purposes

Marionnet

a virtual network laboratory for teaching purposes

- nice and clean GUI (french only)

Marionnet

a virtual network laboratory for teaching purposes

- nice and clean GUI (french only)
- 100% real experience

Marionnet

a virtual network laboratory for teaching purposes

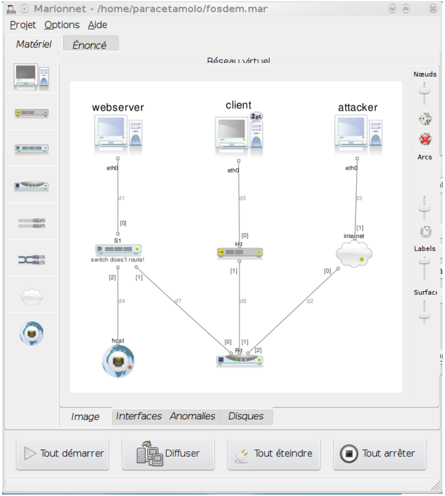
- nice and clean GUI (french only)
- 100% real experience
- OCaml

Marionnet

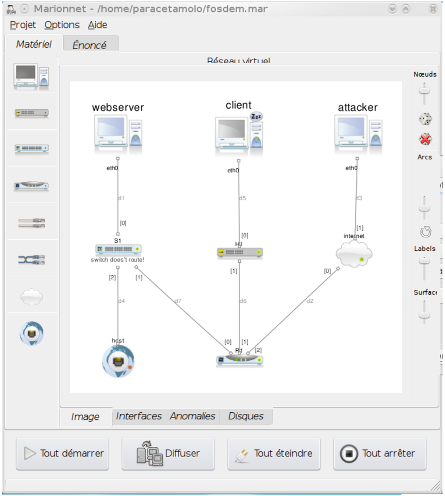
a virtual network laboratory for teaching purposes

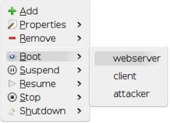
- nice and clean GUI (french only)
- 100% real experience
- OCaml
- GPLv2-or-later

Marionnet

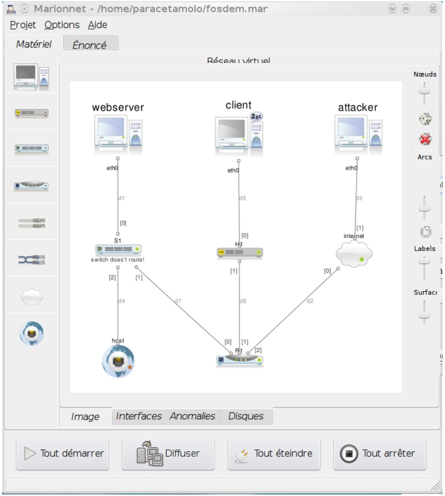

Marionnet - /home/paracetamolo/fosdem.mar
Projet
Options
Aide
Matériel
Énoncé
Réseau virtuel
webserver
client
attacker
eth0
eth0
eth0
d1
d5
d3
[0]
[0]
[1]
S1
H1
internet
switch doesn't route!
[2]
[1]
[1]
[0]
d4
d7
d6
d2
host
[0]
[1]
[2]
R1
Nœuds
Arcs
Labels
Surface
Image
Interfaces
Anomalies
Disques
Tout démarrer
Diffuser
Tout éteindre
Tout arrêter

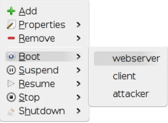
Marionnet

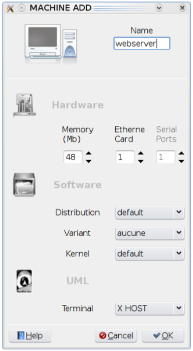

Marionnet - /home/paracetamolo/fosdem.mar
Projet Options Aide
Matériel
Énoncé
Réseau virtuel
webserver
client
attacker
eth0
eth0
eth0
d1
d5
d3
[0]
[0]
[1]
S1
H1
internet
switch does't route!
[2]
[1]
[1]
[0]
d4
d7
d6
d2
host
[0]
[1]
R1
[2]
Nœuds
Arcs
Labels
Surface
Image
Interfaces
Anomalies
Disques
Tout démarrer
Diffuser
Tout éteindre
Tout arrêter


Add
Properties
Remove
Boot
Suspend
Resume
Stop
Shutdown
webserver
client
attacker

Marionnet







Name: webserver

Memory (Mb): 48

Etherne Card: 1

Serial Ports: 1

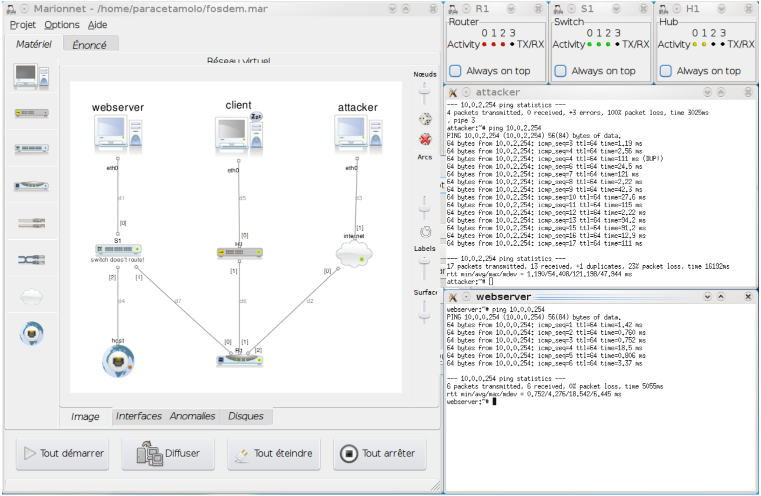
Distribution: default

Variant: aucune

Kernel: default

Terminal: X HOST

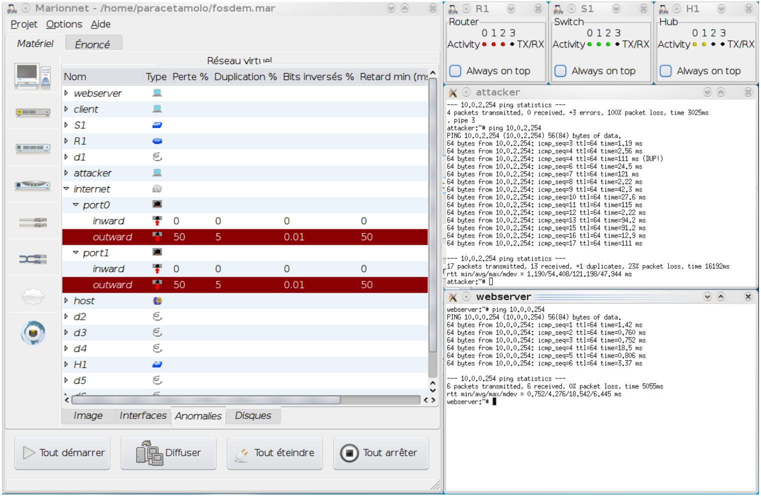
Marionnet



```
--- 10.0.2.254 ping statistics ---
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3025ms
, pipe 3
attacker:~# ping 10.0.2.254
PING 10.0.2.254 (10.0.2.254) 56(84) bytes of data.
64 bytes from 10.0.2.254: icmp_seq=3 ttl=64 time=1.19 ms
64 bytes from 10.0.2.254: icmp_seq=4 ttl=64 time=2.56 ms
64 bytes from 10.0.2.254: icmp_seq=4 ttl=64 time=111 ms (DUP!)
64 bytes from 10.0.2.254: icmp_seq=6 ttl=64 time=24.5 ms
64 bytes from 10.0.2.254: icmp_seq=7 ttl=64 time=121 ms
64 bytes from 10.0.2.254: icmp_seq=8 ttl=64 time=2.22 ms
64 bytes from 10.0.2.254: icmp_seq=9 ttl=64 time=42.3 ms
64 bytes from 10.0.2.254: icmp_seq=10 ttl=64 time=27.6 ms
64 bytes from 10.0.2.254: icmp_seq=11 ttl=64 time=115 ms
64 bytes from 10.0.2.254: icmp_seq=12 ttl=64 time=2.22 ms
64 bytes from 10.0.2.254: icmp_seq=13 ttl=64 time=94.2 ms
64 bytes from 10.0.2.254: icmp_seq=15 ttl=64 time=91.2 ms
64 bytes from 10.0.2.254: icmp_seq=16 ttl=64 time=12.9 ms
64 bytes from 10.0.2.254: icmp_seq=17 ttl=64 time=111 ms

--- 10.0.2.254 ping statistics ---
17 packets transmitted, 13 received, +1 duplicates, 23% packet loss, time 16192ms
rtt min/avg/max/mdev = 1.190/54.408/121.198/47.944 ms
attacker:~# 
```

```
webserver:~# ping 10.0.0.254
PING 10.0.0.254 (10.0.0.254) 56(84) bytes of data.
64 bytes from 10.0.0.254: icmp_seq=1 ttl=64 time=1.42 ms
64 bytes from 10.0.0.254: icmp_seq=2 ttl=64 time=0.760 ms
64 bytes from 10.0.0.254: icmp_seq=3 ttl=64 time=0.752 ms
64 bytes from 10.0.0.254: icmp_seq=4 ttl=64 time=18.5 ms
64 bytes from 10.0.0.254: icmp_seq=5 ttl=64 time=0.806 ms
64 bytes from 10.0.0.254: icmp_seq=6 ttl=64 time=3.37 ms

--- 10.0.0.254 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5055ms
rtt min/avg/max/mdev = 0.752/4.276/18.542/6.445 ms
webserver:~# 
```

Defects

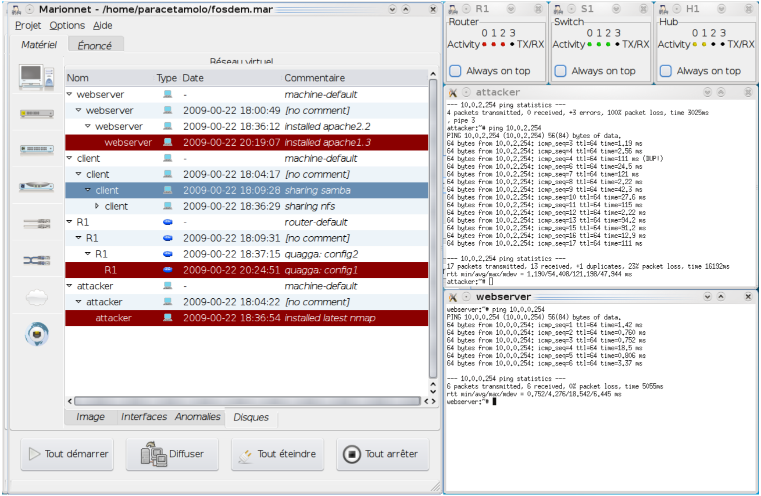


Nom	Type	Perte %	Duplication %	Bits inversés %	Retard min (m
webserver					
client					
S1					
R1					
d1					
attacker					
internet					
port0					
inward		0	0	0	0
outward		50	5	0.01	50
port1					
inward		0	0	0	0
outward		50	5	0.01	50
host					
d2					
d3					
d4					
H1					
d5					

```
--- 10.0.2.254 ping statistics ---
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3025ms
, pipe 3
attacker:~# ping 10.0.2.254
PING 10.0.2.254 (10.0.2.254) 56(84) bytes of data.
64 bytes from 10.0.2.254: icmp_seq=3 ttl=64 time=1.19 ms
64 bytes from 10.0.2.254: icmp_seq=4 ttl=64 time=2.56 ms
64 bytes from 10.0.2.254: icmp_seq=4 ttl=64 time=111 ms (DUP!)
64 bytes from 10.0.2.254: icmp_seq=6 ttl=64 time=24.5 ms
64 bytes from 10.0.2.254: icmp_seq=7 ttl=64 time=121 ms
64 bytes from 10.0.2.254: icmp_seq=8 ttl=64 time=2.22 ms
64 bytes from 10.0.2.254: icmp_seq=9 ttl=64 time=42.3 ms
64 bytes from 10.0.2.254: icmp_seq=10 ttl=64 time=27.6 ms
64 bytes from 10.0.2.254: icmp_seq=11 ttl=64 time=115 ms
64 bytes from 10.0.2.254: icmp_seq=12 ttl=64 time=2.22 ms
64 bytes from 10.0.2.254: icmp_seq=13 ttl=64 time=94.2 ms
64 bytes from 10.0.2.254: icmp_seq=15 ttl=64 time=91.2 ms
64 bytes from 10.0.2.254: icmp_seq=16 ttl=64 time=12.9 ms
64 bytes from 10.0.2.254: icmp_seq=17 ttl=64 time=111 ms

--- 10.0.2.254 ping statistics ---
17 packets transmitted, 13 received, +1 duplicates, 23% packet loss, time 16192ms
rtt min/avg/max/mdev = 1.190/54.408/121.198/47.944 ms
attacker:~# 
```

```
webserver:~# ping 10.0.0.254
PING 10.0.0.254 (10.0.0.254) 56(84) bytes of data.
64 bytes from 10.0.0.254: icmp_seq=1 ttl=64 time=1.42 ms
64 bytes from 10.0.0.254: icmp_seq=2 ttl=64 time=0.760 ms
64 bytes from 10.0.0.254: icmp_seq=3 ttl=64 time=0.752 ms
64 bytes from 10.0.0.254: icmp_seq=4 ttl=64 time=18.5 ms
64 bytes from 10.0.0.254: icmp_seq=5 ttl=64 time=0.806 ms
64 bytes from 10.0.0.254: icmp_seq=6 ttl=64 time=3.37 ms

--- 10.0.0.254 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5055ms
rtt min/avg/max/mdev = 0.752/4.276/18.542/6.445 ms
webserver:~# 
```

Disks



Nom	Type	Date	Commentaire
webserver		-	machine-default
webserver		2009-00-22 18:00:49	[no comment]
webserver		2009-00-22 18:36:12	installed apache2.2
webserver		2009-00-22 20:19:07	installed apache1.3
client		-	machine-default
client		2009-00-22 18:04:17	[no comment]
client		2009-00-22 18:09:28	sharing samba
client		2009-00-22 18:36:29	sharing nfs
R1		-	router-default
R1		2009-00-22 18:09:31	[no comment]
R1		2009-00-22 18:37:15	quagga: config2
R1		2009-00-22 20:24:51	quagga: config1
attacker		-	machine-default
attacker		2009-00-22 18:04:22	[no comment]
attacker		2009-00-22 18:36:54	installed latest nmap

□ Always on top

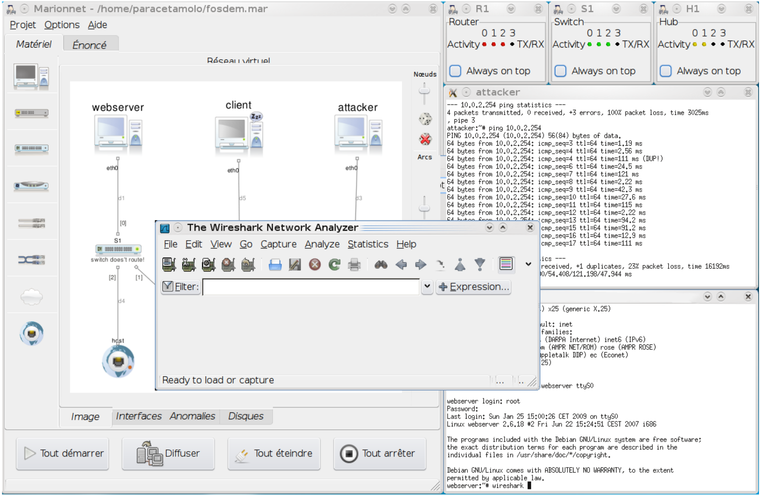
□ Always on top

□ Always on top

```
--- 10.0.2.254 ping statistics ---
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3025ms
, pipe 3
attacker:~# ping 10.0.2.254
PING 10.0.2.254 (10.0.2.254) 56(84) bytes of data.
64 bytes from 10.0.2.254: icmp_seq=3 ttl=64 time=1.19 ms
64 bytes from 10.0.2.254: icmp_seq=4 ttl=64 time=2.56 ms
64 bytes from 10.0.2.254: icmp_seq=4 ttl=64 time=111 ms (DUP!)
64 bytes from 10.0.2.254: icmp_seq=6 ttl=64 time=24.5 ms
64 bytes from 10.0.2.254: icmp_seq=7 ttl=64 time=121 ms
64 bytes from 10.0.2.254: icmp_seq=8 ttl=64 time=2.22 ms
64 bytes from 10.0.2.254: icmp_seq=9 ttl=64 time=42.3 ms
64 bytes from 10.0.2.254: icmp_seq=10 ttl=64 time=27.6 ms
64 bytes from 10.0.2.254: icmp_seq=11 ttl=64 time=115 ms
64 bytes from 10.0.2.254: icmp_seq=12 ttl=64 time=2.22 ms
64 bytes from 10.0.2.254: icmp_seq=13 ttl=64 time=94.2 ms
64 bytes from 10.0.2.254: icmp_seq=15 ttl=64 time=91.2 ms
64 bytes from 10.0.2.254: icmp_seq=16 ttl=64 time=12.9 ms
64 bytes from 10.0.2.254: icmp_seq=17 ttl=64 time=111 ms

--- 10.0.2.254 ping statistics ---
17 packets transmitted, 13 received, +1 duplicates, 23% packet loss, time 16192ms
rtt min/avg/max/mdev = 1.190/54.408/121.198/47.944 ms
attacker:~# 
```

```
webserver:~# ping 10.0.0.254
PING 10.0.0.254 (10.0.0.254) 56(84) bytes of data.
64 bytes from 10.0.0.254: icmp_seq=1 ttl=64 time=1.42 ms
64 bytes from 10.0.0.254: icmp_seq=2 ttl=64 time=0.760 ms
64 bytes from 10.0.0.254: icmp_seq=3 ttl=64 time=0.752 ms
64 bytes from 10.0.0.254: icmp_seq=4 ttl=64 time=18.5 ms
64 bytes from 10.0.0.254: icmp_seq=5 ttl=64 time=0.806 ms
64 bytes from 10.0.0.254: icmp_seq=6 ttl=64 time=3.37 ms

--- 10.0.0.254 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5055ms
rtt min/avg/max/mdev = 0.752/4.276/18.542/6.445 ms
webserver:~# 
```

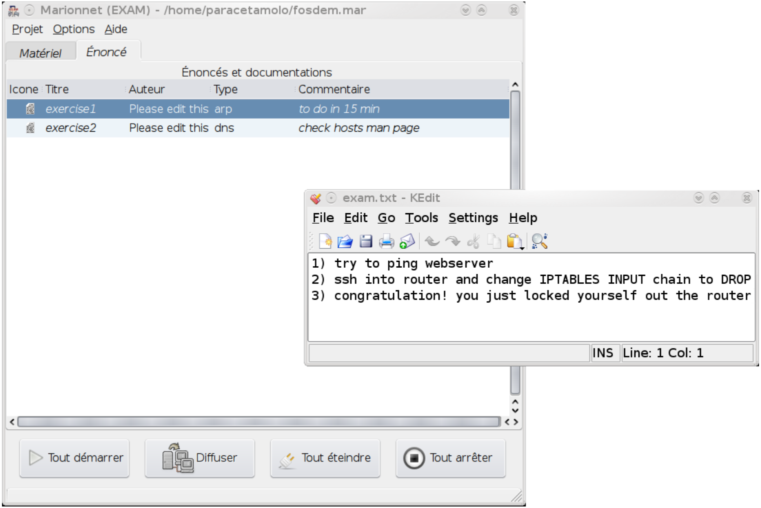
Host X server



```
--- 10.0.2.254 ping statistics ---
4 packets transmitted, 0 received, +3 errors, 100% packet loss, time 3025ms
, pipe 3
attacker:~# ping 10.0.2.254
PING 10.0.2.254 (10.0.2.254) 56(84) bytes of data.
64 bytes from 10.0.2.254: icmp_seq=3 ttl=64 time=1.19 ms
64 bytes from 10.0.2.254: icmp_seq=4 ttl=64 time=2.56 ms
64 bytes from 10.0.2.254: icmp_seq=4 ttl=64 time=111 ms (DUP!)
64 bytes from 10.0.2.254: icmp_seq=6 ttl=64 time=24.5 ms
64 bytes from 10.0.2.254: icmp_seq=7 ttl=64 time=121 ms
64 bytes from 10.0.2.254: icmp_seq=8 ttl=64 time=2.22 ms
64 bytes from 10.0.2.254: icmp_seq=9 ttl=64 time=42.3 ms
64 bytes from 10.0.2.254: icmp_seq=10 ttl=64 time=27.6 ms
64 bytes from 10.0.2.254: icmp_seq=11 ttl=64 time=115 ms
64 bytes from 10.0.2.254: icmp_seq=12 ttl=64 time=2.22 ms
64 bytes from 10.0.2.254: icmp_seq=13 ttl=64 time=94.2 ms
                          cmp_seq=15 ttl=64 time=91.2 ms
                          cmp_seq=16 ttl=64 time=12.9 ms
                          cmp_seq=17 ttl=64 time=111 ms

                          ics ---
                          received, +1 duplicates, 23% packet loss, time 16192ms
                          0/54.408/121.198/47.944 ms
```

```
                          ) x25 (generic X.25)

                          ult: inet
                          families:
                          (DARPA Internet) inet6 (IPv6)
                          m (AMPR NET/ROM) rose (AMPR ROSE)
                          ppletalk DDP) ec (Econet)
                          25)


                          webserver ttyS0

webserver login: root
Password:
Last login: Sun Jan 25 15:00:26 CET 2009 on ttyS0
Linux webserver 2.6.18 #2 Fri Jun 22 15:24:51 CEST 2007 i686

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
webserver:~# wireshark
```

Exam mode



Icone	Titre	Auteur	Type	Commentaire
	exercise1	Please edit this	arp	*to do in 15 min*
	exercise2	Please edit this	dns	*check hosts man page*

```
1) try to ping webserver
2) ssh into router and change IPTABLES INPUT chain to DROP
3) congratulation! you just locked yourself out the router
```

Every good work of software starts to scratch a developer's personal itch.

Jean-Vincent Loddo is teacher of computer networks at Univesité de Paris 13:

Every good work of software starts to scratch a developer's personal itch.

Jean-Vincent Loddo is teacher of computer networks at Univesité de Paris 13:

- hw malfunctioning

Every good work of software starts to scratch a developer's personal itch.

Jean-Vincent Loddo is teacher of computer networks at Univesité de Paris 13:

- hw malfunctioning
- long preparation of network

Every good work of software starts to scratch a developer's personal itch.

Jean-Vincent Loddo is teacher of computer networks at Univesité de Paris 13:

- hw malfunctioning
- long preparation of network
- expensive hw

Every good work of software starts to scratch a developer's personal itch.

Jean-Vincent Loddo is teacher of computer networks at Univesité de Paris 13:

- hw malfunctioning
- long preparation of network
- expensive hw
- no personal work

Every good work of software starts to scratch a developer's personal itch.

Jean-Vincent Loddo is teacher of computer networks at Univesité de Paris 13:

- hw malfunctioning
- long preparation of network
- expensive hw
- no personal work
- exercises: hard to prepare and revise

Every good work of software starts to scratch a developer's personal itch.

Jean-Vincent Loddo is teacher of computer networks at Univesité de Paris 13:

- hw malfunctioning
- long preparation of network
- expensive hw
- no personal work
- exercises: hard to prepare and revise
- difficult exams evaluation

Every good work of software starts to scratch a developer's personal itch.

Jean-Vincent Loddo is teacher of computer networks at Univesité de Paris 13:

- hw malfunctioning
- long preparation of network
- expensive hw
- no personal work
- exercises: hard to prepare and revise
- difficult exams evaluation
- little network

Every good work of software starts to scratch a developer's personal itch.

Jean-Vincent Loddo is teacher of computer networks at Univesité de Paris 13:

- hw malfunctioning
- long preparation of network
- expensive hw
- no personal work
- exercises: hard to prepare and revise
- difficult exams evaluation
- little network
- need to access to the lab

Every good work of software starts to scratch a developer's personal itch.

Jean-Vincent Loddo is teacher of computer networks at Univesité de Paris 13:

- hw malfunctioning
- long preparation of network
- expensive hw
- no personal work
- exercises: hard to prepare and revise
- difficult exams evaluation
- little network
- need to access to the lab
- few student

Every good work of software starts to scratch a developer's personal itch.

Jean-Vincent Loddo is teacher of computer networks at Univesité de Paris 13:

- hw malfunctioning
- long preparation of network
- expensive hw
- no personal work
- exercises: hard to prepare and revise
- difficult exams evaluation
- little network
- need to access to the lab
- few student

Personal note: I spent 2 days to try a point-to-point connection, with Marionnet you can explore the most obscure iptables configurations!!

Université de Paris 13.

IUT de Villetaneuse and L.I.P.N. are supporting the project.

2007

Although Marionnet was started in 2005 by Jean-Vincent Loddo, it saw its major development during six months of 2007 when Luca Saiu joined the project.

2007

Although Marionnet was started in 2005 by Jean-Vincent Loddo, it saw its major development during six months of 2007 when Luca Saiu joined the project.
2008 was the "year of the truth": live dvd released, Marionnet was adopted in several universities in France and other countries.

2007

Although Marionnet was started in 2005 by Jean-Vincent Loddo, it saw its major development during six months of 2007 when Luca Saiu joined the project.
2008 was the "year of the truth": live dvd released, Marionnet was adopted in several universities in France and other countries.
Right now we are cleaning up code, writing documentation and how-tos.

2007

Although Marionnet was started in 2005 by Jean-Vincent Loddo, it saw its major development during six months of 2007 when Luca Saiu joined the project.
2008 was the "year of the truth": live dvd released, Marionnet was adopted in several universities in France and other countries.
Right now we are cleaning up code, writing documentation and how-tos.
I'm working on developer documentation and i18n.

Authors:

Jean-Vincent Loddo is currently assistant professor (Maître de conférence) of computer science at the Univesité of Paris 13 and founder/lead developer of Marionnet.

Authors:

Jean-Vincent Loddo is currently assistant professor (Maître de conférence) of computer science at the Univesité of Paris 13 and founder/lead developer of Marionnet.
Luca Saiu is Ph.D student at the university of Paris 13 where he is developing his Epsilon language for the GNU project (stay tuned).

Authors:

Jean-Vincent Loddo is currently assistant professor (Maître de conférence) of computer science at the Univesité of Paris 13 and founder/lead developer of Marionnet.
Luca Saiu is Ph.D student at the university of Paris 13 where he is developing his Epsilon language for the GNU project (stay tuned).
They both are strong advocates of functional programming and free software.

Under the hood

We love free software:

Under the hood

We love free software:

- UML emulates machines and routers(quagga)

Under the hood

We love free software:

- UML emulates machines and routers(quagga)
- VDE emulates cables(defects), hubs, switches

Under the hood

We love free software:

- UML emulates machines and routers(quagga)
- VDE emulates cables(defects), hubs, switches
- Marionnet Network

Under the hood

We love free software:

- UML emulates machines and routers(quagga)
- VDE emulates cables(defects), hubs, switches
- Marionnet Network
- Marionnet GUI (GTK+, dot)

Cool stuff:

- dynamic reconfiguration

Cool stuff:

- dynamic reconfiguration
- concurrency

Cool stuff:

- dynamic reconfiguration
- concurrency
- fault tollerant

Cool stuff:

- dynamic reconfiguration
- concurrency
- fault tollerant
- hw defects(port+wire)

Cool stuff:

- dynamic reconfiguration
- concurrency
- fault tollerant
- hw defects(port+wire)
- gateway(linux bridge)

Cool stuff:

- dynamic reconfiguration
- concurrency
- fault tollerant
- hw defects(port+wire)
- gateway(linux bridge)
- ditribution choice(future)

Cool stuff:

- dynamic reconfiguration
- concurrency
- fault tollerant
- hw defects(port+wire)
- gateway(linux bridge)
- ditribution choice(future)
- COW+sparse(save disk, usefull to test configurations)

Cool stuff:

- dynamic reconfiguration
- concurrency
- fault tollerant
- hw defects(port+wire)
- gateway(linux bridge)
- ditribution choice(future)
- COW+sparse(save disk, usefull to test configurations)
- full GNU/Linux binary compatibility

Cool stuff:

- dynamic reconfiguration
- concurrency
- fault tollerant
- hw defects(port+wire)
- gateway(linux bridge)
- ditribution choice(future)
- COW+sparse(save disk, usefull to test configurations)
- full GNU/Linux binary compatibility
- graphical applications (wireshark)

Cool stuff:

- dynamic reconfiguration
- concurrency
- fault tollerant
- hw defects(port+wire)
- gateway(linux bridge)
- ditribution choice(future)
- COW+sparse(save disk, usefull to test configurations)
- full GNU/Linux binary compatibility
- graphical applications (wireshark)
- GUI(set up, feedback)

Cool stuff:

- dynamic reconfiguration
- concurrency
- fault tollerant
- hw defects(port+wire)
- gateway(linux bridge)
- ditribution choice(future)
- COW+sparse(save disk, usefull to test configurations)
- full GNU/Linux binary compatibility
- graphical applications (wireshark)
- GUI(set up, feedback)
- exam mode (retrieve student info)

Cool stuff:

- dynamic reconfiguration
- concurrency
- fault tollerant
- hw defects(port+wire)
- gateway(linux bridge)
- ditribution choice(future)
- COW+sparse(save disk, usefull to test configurations)
- full GNU/Linux binary compatibility
- graphical applications (wireshark)
- GUI(set up, feedback)
- exam mode (retrieve student info)
- file format back-forth-compatible

Cool stuff:

- dynamic reconfiguration
- concurrency
- fault tollerant
- hw defects(port+wire)
- gateway(linux bridge)
- ditribution choice(future)
- COW+sparse(save disk, usefull to test configurations)
- full GNU/Linux binary compatibility
- graphical applications (wireshark)
- GUI(set up, feedback)
- exam mode (retrieve student info)
- file format back-forth-compatible
- pinocchio (100mb of what you need)

Future

- debian packaging: many packages, eavy stuff, vde2svn

Future

- debian packaging: many packages, eavy stuff, vde2svn
- machine fs: smallest possible without loosing apps (also graphical)

Future

- debian packaging: many packages, eavy stuff, vde2svn
- machine fs: smallest possible without loosing apps (also graphical)
- i18n: interface is french only, glade and ocaml-gettext are not really friends but coming soon!

Future

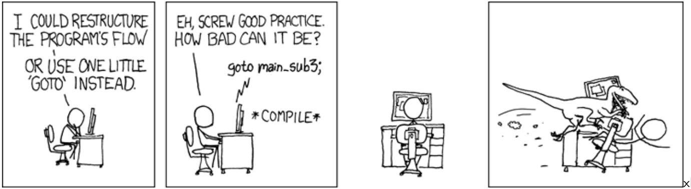
- debian packaging: many packages, eavy stuff, vde2svn
- machine fs: smallest possible without loosing apps (also graphical)
- i18n: interface is french only, glade and ocaml-gettext are not really friends but coming soon!
- advanced features: vlan (switch), firewall+snat+dhcp (router)

Future

- debian packaging: many packages, eavy stuff, vde2svn
- machine fs: smallest possible without loosing apps (also graphical)
- i18n: interface is french only, glade and ocaml-gettext are not really friends but coming soon!
- advanced features: vlan (switch), firewall+snat+dhcp (router)
- new devices: IP phone

Thanks

Problems you don't have in OCaml...


I COULD RESTRUCTURE THE PROGRAM'S FLOW
OR USE ONE LITTLE 'GOTO' INSTEAD.
EH, SCREW GOOD PRACTICE. HOW BAD CAN IT BE?
goto main_sub3;
COMPILE

xkcd.com