

Marionnet: networking for dummies.

Marco Stronati

2009



what

why

where

when

who

how

features

future

Marionnet

Marionnet: networking
for dummies.

M. Stronati

a virtual network laboratory for teaching purposes

what

why

where

when

who

how

features

future

Marionnet

Marionnet: networking
for dummies.

M. Stronati

what

why

where

when

who

how

features

future

a virtual network laboratory for teaching purposes

- ▶ nice and clean GUI (french only)

Marionnet

Marionnet: networking
for dummies.

M. Stronati

what

why

where

when

who

how

features

future

a virtual network laboratory for teaching purposes

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

what

why

where

when

who

how

features

future

a virtual network laboratory for teaching purposes

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

what

why

where

when

who

how

features

future

a virtual network laboratory for teaching purposes

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

Marionnet

Marionnet: networking for dummies.

M. Stronati

what

why

where

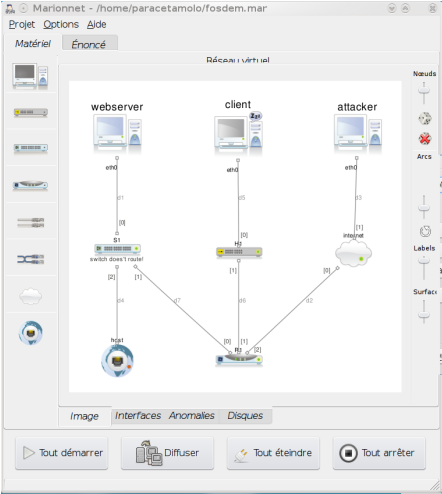
when

who

how

features

future



Marionnet

what

why

where

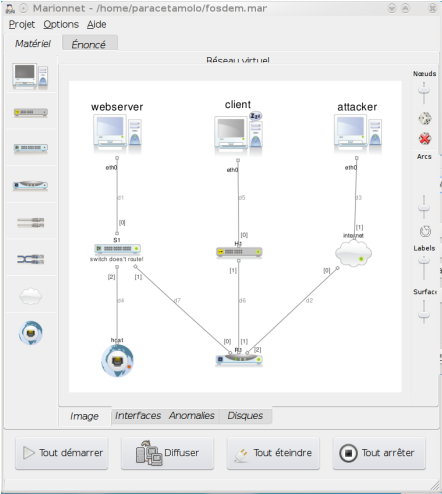
when

who

how

features

future



Marionnet

Marionnet: networking
for dummies.

M. Stronati

The screenshot shows the Marionnet application window titled "Marionnet - /home/paracetamol/fosdem.mar". The interface includes a menu bar with "Projet", "Options", and "Aide". Below the menu bar are tabs for "Matériel" and "Énoncé". The main workspace displays a network diagram with three hosts: "webservers", "client", and "attacker". Each host is connected to a central switch labeled "switch doesn't route!". The "webservers" host is connected to the switch via interface "eth0" (port 01) and "s1" (port 02). The "client" host is connected via interface "eth0" (port 05) and "H2" (port 01). The "attacker" host is connected via interface "eth0" (port 03) and "infante" (port 01). The switch is connected to a cloud labeled "cloud" via interface "s1" (port 04) and "H2" (port 02). The interface also includes a toolbar with buttons for "Tout démarrer", "Diffuser", "Tout éteindre", and "Tout arrêter".

The screenshot shows a context menu for a host in Marionnet. The menu items are: "+ Add", "Properties >", "Remove >", "Boot >", "Suspend >", "Resume >", "Stop >", and "Shutdown >". A sub-menu is open for the "Boot" option, showing three choices: "webservers", "client", and "attacker".

The screenshot shows the "MACHINE ADD" dialog box in Marionnet. The dialog has a "Name" field with the value "webservers". Below the name field are sections for "Hardware" and "Software". The "Hardware" section includes "Memory (Mb)" set to 48, "Ethern Card" set to 1, and "Serial Ports" set to 1. The "Software" section includes "Distribution" set to "default", "Variant" set to "aucune", and "Kernel" set to "default". The "UML" section includes "Terminal" set to "X HOST". At the bottom of the dialog are buttons for "Help", "Cancel", and "OK".

what

why

where

when

who

how

features

future

Marionnet

Marionnet: networking for dummies.

M. Stronati

what

why

where

when

who

how

features

future

The screenshot displays the Marionnet network simulator interface. The main window shows a network diagram with three hosts: 'webserver', 'client', and 'attacker'. The 'webserver' is connected to a switch (S1) via interface 'eth0'. The 'client' is connected to the same switch (S1) via interface 'eth0'. The 'attacker' is connected to the switch (S1) via interface 'eth0'. The switch (S1) is connected to a hub (H1) via interface 'eth0'. The hub (H1) is connected to the 'attacker' via interface 'eth0'. The network diagram also shows a 'router' (R1) and another switch (S1) connected to the hub (H1). The interface 'eth0' of the router (R1) is connected to the interface 'eth0' of the switch (S1). The interface 'eth0' of the switch (S1) is connected to the interface 'eth0' of the hub (H1). The interface 'eth0' of the hub (H1) is connected to the interface 'eth0' of the attacker.

Terminal windows show the following output:

```
attacker
--- 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=2.96 ms
64 bytes from 10.0.2.254: icmp_seq=5 ttl=64 time=11.1 ms (DPI!)
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=47.5 ms
64 bytes from 10.0.2.254: icmp_seq=10 ttl=64 time=27.5 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=34.2 ms
64 bytes from 10.0.2.254: icmp_seq=15 ttl=64 time=31.2 ms
64 bytes from 10.0.2.254: icmp_seq=16 ttl=64 time=12.3 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 16152ms
etc min/avg/max/mdev = 1.190/54.408/121.198/47.944 ms
attacker:~#

webserver
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
etc min/avg/max/mdev = 0.752/4.076/18.542/5.440 ms
webserver:~#
```

Defects

Marionnet: networking for dummies.

M. Stronati

what

why

where

when

who

how

features

future

The screenshot displays the Marionnet interface with a network configuration table and several terminal windows showing ping statistics.

Network Configuration Table:

Nom	Type	Perte %	Duplication %	Bits inversés %	Retard min (ms)
webservers					
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					

attacker terminal:

```
--- 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(64) 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=2.96 ms
64 bytes from 10.0.2.254: icmp_seq=5 ttl=64 time=11.1 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=47.5 ms
64 bytes from 10.0.2.254: icmp_seq=10 ttl=64 time=27.5 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=34.2 ms
64 bytes from 10.0.2.254: icmp_seq=15 ttl=64 time=31.2 ms
64 bytes from 10.0.2.254: icmp_seq=16 ttl=64 time=12.3 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 16152ms
rtt min/avg/max/mdev = 1.190/54.408/121.198/47.944 ms
attacker:~#
```

webservers terminal:

```
webservers:~# ping 10.0.2.254
PING 10.0.2.254 (10.0.2.254): 56(64) bytes of data:
64 bytes from 10.0.2.254: icmp_seq=1 ttl=64 time=1.42 ms
64 bytes from 10.0.2.254: icmp_seq=2 ttl=64 time=0.760 ms
64 bytes from 10.0.2.254: icmp_seq=3 ttl=64 time=0.752 ms
64 bytes from 10.0.2.254: icmp_seq=4 ttl=64 time=18.5 ms
64 bytes from 10.0.2.254: icmp_seq=5 ttl=64 time=0.806 ms
64 bytes from 10.0.2.254: icmp_seq=6 ttl=64 time=3.37 ms

--- 10.0.2.254 ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5055ms
rtt min/avg/max/mdev = 0.752/4.076/18.542/6.440 ms
webservers:~#
```

Disks

Marionnet: networking for dummies.

M. Stronati

what

why

where

when

who

how

features

future

The screenshot shows the Marionnet GUI with a virtual network topology. The main window displays a list of nodes and their configurations. The nodes are:

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

The terminal window shows ping statistics for the attacker node:

```
attacker:~# ping 10.0.2.254
--- 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(64) 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=2.96 ms
64 bytes from 10.0.2.254: icmp_seq=5 ttl=64 time=111 ms (RPI)
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=67.5 ms
64 bytes from 10.0.2.254: icmp_seq=10 ttl=64 time=27.5 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=34.2 ms
64 bytes from 10.0.2.254: icmp_seq=15 ttl=64 time=31.2 ms
64 bytes from 10.0.2.254: icmp_seq=16 ttl=64 time=12.3 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 16152ms
rtt min/avg/max/mdev = 1.190/54.408/121.198/47.944 ms
attacker:~#
```

Host X server

Marionnet: networking for dummies.

M. Stronati

what

why

where

when

who

how

features

future

The screenshot displays the Marionnet graphical user interface. At the top, the window title is "Marionnet - /home/paracetamol/fosdem.mar". Below the title bar, there are menu options: "Projet", "Options", and "Aide". The main workspace is titled "Réseau virtuel" and shows a network diagram with three hosts: "webserver", "client", and "attacker". These hosts are connected to a central "Switch" (S1) and a "Router" (R1). The router is further connected to a "Hub" (H1). A terminal window titled "attacker" shows the following output:

```
--- 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): 64(64) 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=2.96 ms
64 bytes from 10.0.2.254: icmp_seq=5 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=27.5 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=34.2 ms
64 bytes from 10.0.2.254: icmp_seq=14 ttl=64 time=31.2 ms
64 bytes from 10.0.2.254: icmp_seq=15 ttl=64 time=12.3 ms
64 bytes from 10.0.2.254: icmp_seq=17 ttl=64 time=111 ms

ics ---
received: +1 duplicates, 23% packet loss, time 16152ms
0/54,408/121,198/47,944 ms

) x25 (generic X.25)

alt: inet
family:
  (MFR Internet) inet6 (IPv6)
  (MFR NET/ROM) rose (MFR ROSE)
  (MFR talk DDP) ec (Econet)
  (25)

webserver tty0

webserver login: root
Password:
Last login: Sun Jan 25 15:00:26 CET 2009 on tty0
Linux webserver 2.6.18 #2 Fri Jun 22 15:24:51 EEST 2007 #686

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
```

The Wireshark window is titled "The Wireshark Network Analyzer" and shows a menu bar with "File", "Edit", "View", "Go", "Capture", "Analyze", "Statistics", and "Help". The main area is currently empty, with a filter field and a "Ready to load or capture" message.

Exam mode

The screenshot shows the Marionnet (EXAM) interface. At the top, there is a menu bar with 'Projet', 'Options', and 'Aide'. Below it are tabs for 'Matériel' and 'Énoncé'. The main area displays a table titled 'Énoncés et documentations' with the following content:

Icone	Titre	Auteur	Type	Commentaire
	exercice1	Please edit this arp	arp	to do in 15 min
	exercice2	Please edit this dns	dns	check hosts man page

Overlaid on the bottom right is a KEdit window titled 'exam.txt - KEdit'. It has a menu bar with 'File', 'Edit', 'Go', 'Tools', 'Settings', and 'Help'. The text in the window reads:

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

At the bottom of the Marionnet window, there is a control bar with four buttons: 'Tout démarrer', 'Diffuser', 'Tout éteindre', and 'Tout arrêter'.

what

why

where

when

who

how

features

future

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

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

what

why

where

when

who

how

features

future

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

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

- ▶ hw malfunctioning

what

why

where

when

who

how

features

future

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

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

- ▶ hw malfunctioning
- ▶ long preparation of network

what

why

where

when

who

how

features

future

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

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

- ▶ hw malfunctioning
- ▶ long preparation of network
- ▶ expensive hw

what

why

where

when

who

how

features

future

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

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

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

what

why

where

when

who

how

features

future

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

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

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

what

why

where

when

who

how

features

future

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

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

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

what

why

where

when

who

how

features

future

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

Jean-Vincent Loddó is teacher of computer networks at Université 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 Loddó is teacher of computer networks at Université 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 Université 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

what

why

where

when

who

how

features

future

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

Jean-Vincent Loddó is teacher of computer networks at Université 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!!

what

why

where

when

who

how

features

future

what

why

where

when

who

how

features

future

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.

Marionnet: networking
for dummies.

M. Stronati

what

why

where

when

who

how

features

future

2007

Marionnet: networking
for dummies.

M. Stronati

what

why

where

when

who

how

features

future

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

Marionnet: networking
for dummies.

M. Stronati

what

why

where

when

who

how

features

future

Although Marionnet was started in 2005 by Jean-Vincent Loddio, 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

what

why

where

when

who

how

features

future

Although Marionnet was started in 2005 by Jean-Vincent Loddò, 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.

Marionnet: networking
for dummies.

M. Stronati

what

why

where

when

who

how

features

future

Authors:

Jean-Vincent Loddo is currently assistant professor (Maître de conférence) of computer science at the Université 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 Université 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:

Marionnet: networking
for dummies.

M. Stronati

what

why

where

when

who

how

features

future

Under the hood

Marionnet: networking
for dummies.

M. Stronati

what

why

where

when

who

how

features

future

We love free software:

- ▶ UML emulates machines and routers(quagga)

Under the hood

Marionnet: networking
for dummies.

M. Stronati

what

why

where

when

who

how

features

future

We love free software:

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

Under the hood

Marionnet: networking
for dummies.

M. Stronati

what

why

where

when

who

how

features

future

We love free software:

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

Under the hood

what

why

where

when

who

how

features

future

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

what

why

where

when

who

how

features

future

Cool stuff:

- ▶ dynamic reconfiguration
- ▶ concurrency

what

why

where

when

who

how

features

future

Cool stuff:

- ▶ dynamic reconfiguration
- ▶ concurrency
- ▶ fault tollerant

what

why

where

when

who

how

features

future

Cool stuff:

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

what

why

where

when

who

how

features

future

Cool stuff:

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

what

why

where

when

who

how

features

future

Cool stuff:

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

what

why

where

when

who

how

features

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)

what

why

where

when

who

how

features

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)
- ▶ full GNU/Linux binary compatibility

what

why

where

when

who

how

features

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)
- ▶ full GNU/Linux binary compatibility
- ▶ graphical applications (wireshark)

what

why

where

when

who

how

features

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)
- ▶ full GNU/Linux binary compatibility
- ▶ graphical applications (wireshark)
- ▶ GUI(set up, feedback)

what

why

where

when

who

how

features

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)
- ▶ full GNU/Linux binary compatibility
- ▶ graphical applications (wireshark)
- ▶ GUI(set up, feedback)
- ▶ exam mode (retrieve student info)

what

why

where

when

who

how

features

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)
- ▶ full GNU/Linux binary compatibility
- ▶ graphical applications (wireshark)
- ▶ GUI(set up, feedback)
- ▶ exam mode (retrieve student info)
- ▶ file format back-forth-compatible

what

why

where

when

who

how

features

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)
- ▶ 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)

what

why

where

when

who

how

features

future

Future

Marionnet: networking
for dummies.

M. Stronati

what

why

where

when

who

how

features

future

- ▶ debian packaging: many packages, eavy stuff, vde2svn

Future

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

what

why

where

when

who

how

features

future

Future

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

Future

what

why

where

when

who

how

features

future

- ▶ debian packaging: many packages, eavy stuff, vde2svn
- ▶ machine fs: smallest possible without losing 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

what

why

where

when

who

how

features

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

what

why

where

when

who

how

features

future

Problems you don't have in OCaml...

