kathara lab

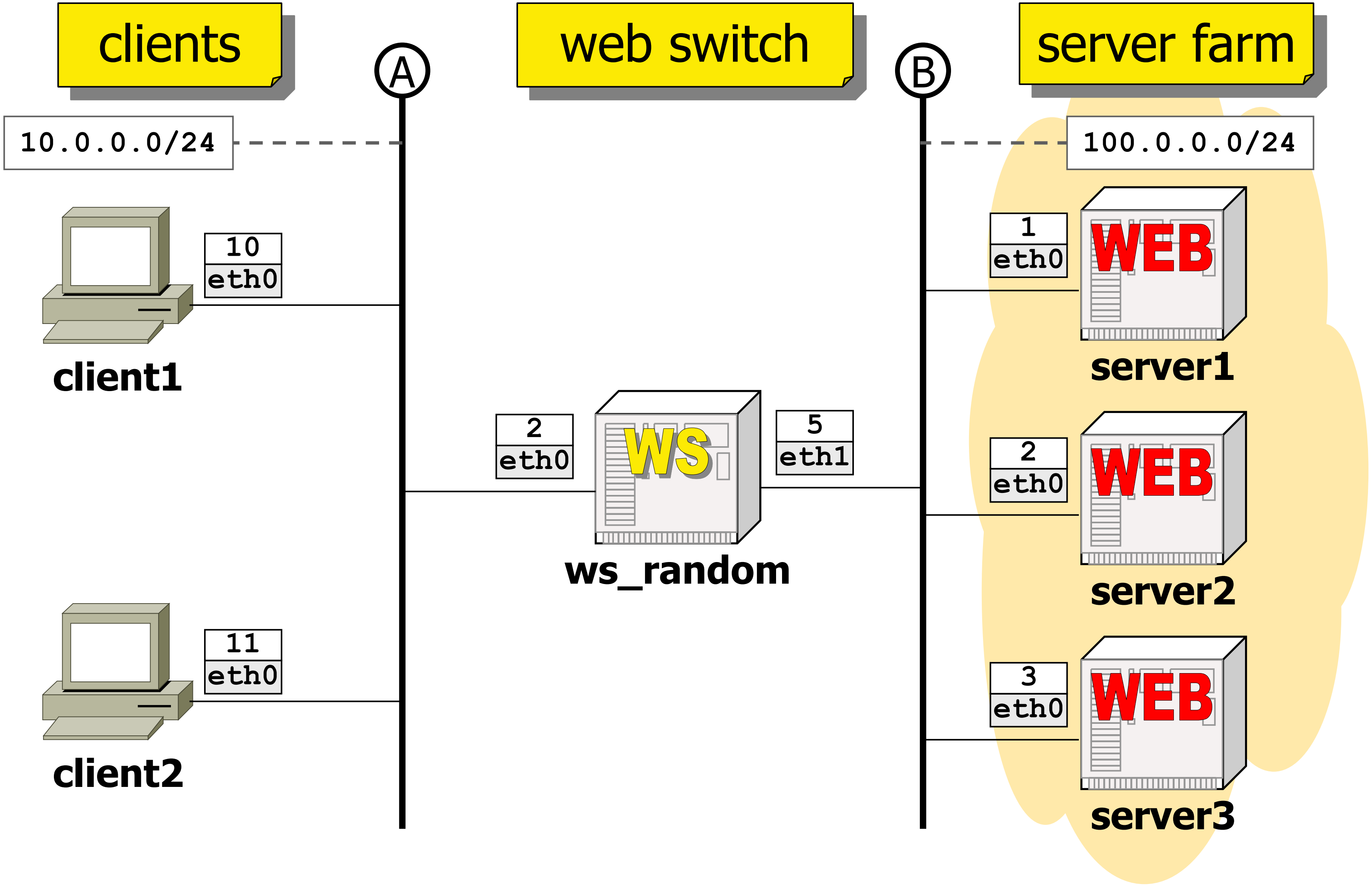
load balancer – web switch – random

Version	1.0
Author(s)	Giuseppe Di Battista, Massimo Rimondini
E-mail	contact@kathara.org
Web	http://www.kathara.org/
Description	A lab showing the operation of a web switch based on iptables – kathara version of a netkit lab

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.

lab topology



lab description

- servers
  - offer a simple HTML default page
  - each physical server hosts a different page, so that they can be easily distinguished
- web switch
  - web switch implements a policy for directing requests to the servers
    - **ws_random**: sends each request to a random server
- clients
  - host a simple web browser (`links`)

lab description – servers

- each server has a different IP address in the subnet **`100.0.0.0/24`**
- no special configuration, just a simple HTML default page in **`/var/www/index.html`**

lab description – web switch

- each web switch has two interfaces
  - one facing the internal network, with an IP address in the same subnet as the servers
  - one facing the external network, exposing a single virtual IP address (VIP) to the clients
- clients only see VIPs of the web switch: they do not know how many servers are in the farm

lab description – web switch

- web switch is implemented using the Linux firewall iptables
  - random

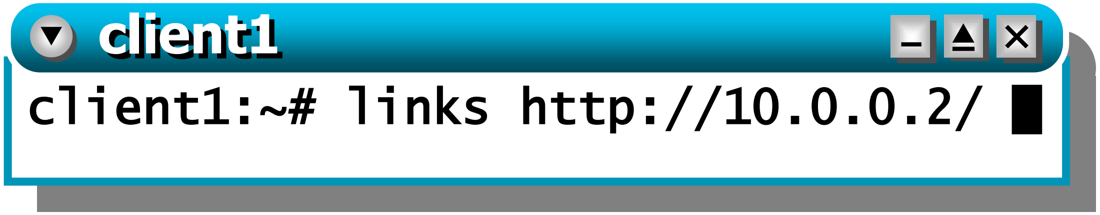
```
iptables --table nat --append PREROUTING --destination 10.0.0.2 --match statistic --mode random --probability 0.33 --
jump DNAT --to-destination 100.0.0.1
iptables --table nat --append PREROUTING --destination 10.0.0.2 --match statistic --mode random --probability 0.5 --
jump DNAT --to-destination 100.0.0.2
iptables --table nat --append PREROUTING --destination 10.0.0.2 --jump DNAT --to-destination 100.0.0.3
iptables --table nat --append POSTROUTING --source 10.0.0.0/24 --destination 100.0.0.0/24 --jump MASQUERADE
```

```
iptables --table nat --append PREROUTING
--destination 10.0.0.2
--match statistic --mode random --probability 0.33
--jump DNAT --to-destination 100.0.0.1
```

the rule applies with a certain probability

experiments

- to experiment load balancing, pick one of the clients, start **`links`**, and direct it to the VIP exposed by web switch:



```
client1:~# links http://10.0.0.2/
```

to experiment random balancing

experiments

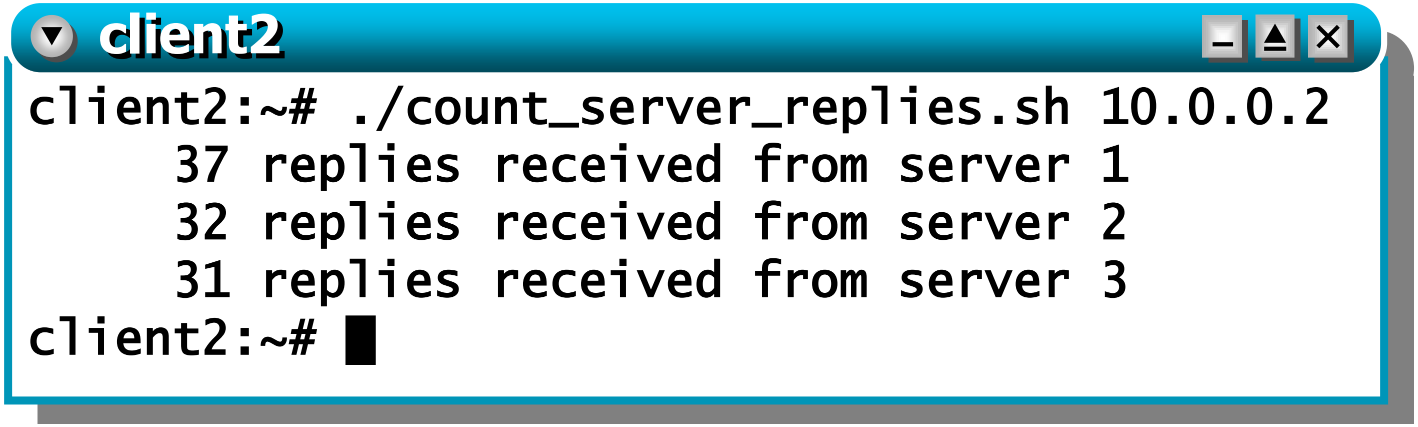
- once you have accessed one of the VIPs, you get a page stating which is the physical server that has served it
- load balancing can be checked by reloading the page (**`ctrl+R`**), but…

experiments

- once you have accessed one of the VIPs, you get a page stating which is the physical server that has served it
- load balancing can be checked by reloading the page (**`ctrl+R`**), but...
  - ...by default all HTTP requests use the same connection (HTTP 1.1)!
  - since iptables tracks TCP connections, all HTTP requests within the same connection are directed to the same physical server
  - to really appreciate load balancing you need to close and re-open links

more experiments

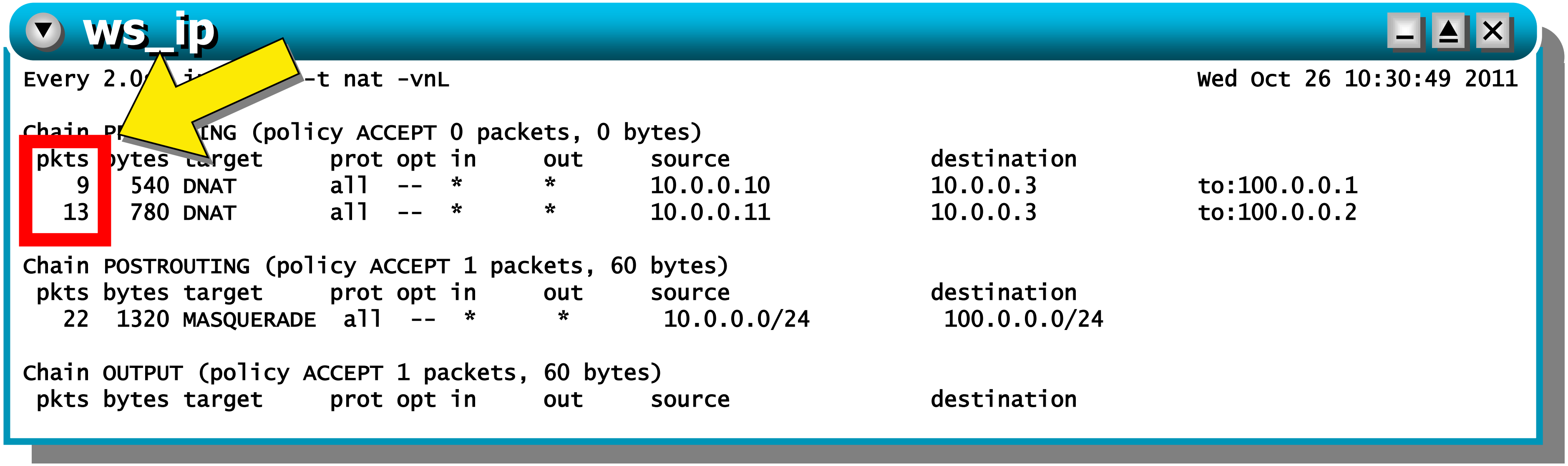
- each client has a handy script that
  - sends 100 HTTP requests (each on a different connection) to a user-specified IP
  - reports the number of pages that have been served by each physical server



```
client2:~# ./count_server_replies.sh 10.0.0.2
     37 replies received from server 1
     32 replies received from server 2
     31 replies received from server 3
client2:~# █
```

more experiments

- after booting, each web switch automatically displays statistics about the number of times that iptables rules have matched
- check the **`pkts`** field in chain **`PREROUTING`**



```
Every 2.0s: iptables -t nat -vnL                                                 Wed Oct 26 10:30:49 2011

Chain PREROUTING (policy ACCEPT 0 packets, 0 bytes)
 pkts bytes target     prot opt in     out     source               destination
    9   540 DNAT       all  --  *      *       10.0.0.10            10.0.0.3             to:100.0.0.1
   13   780 DNAT       all  --  *      *       10.0.0.11            10.0.0.3             to:100.0.0.2

Chain POSTROUTING (policy ACCEPT 1 packets, 60 bytes)
 pkts bytes target     prot opt in     out     source               destination
   22  1320 MASQUERADE  all  --  *      *       10.0.0.0/24          100.0.0.0/24

Chain OUTPUT (policy ACCEPT 1 packets, 60 bytes)
 pkts bytes target     prot opt in     out     source               destination
```