

New Balance

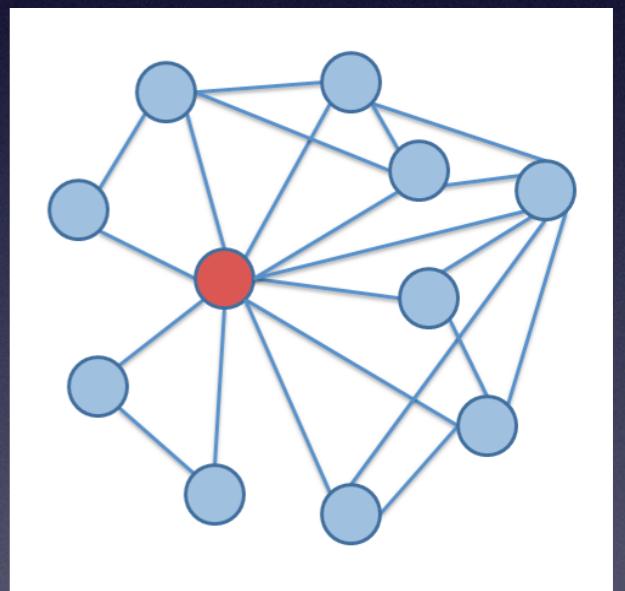
SDN Course, 2017 Fall

Outline

- 1.Background: P4 and Load Balance
- 2.Introduction: What's New Balance?
- 3.Demo Presentation
- 4.Conclusion

Outline

- *1.Background: P4 and Load Balance*
- 2.Introduction: What's New Balance?
- 3.Demo Presentation
- 4.Conclusion



P4 Language



- P4: Programming Protocol-Independent Packet Processors
- A Network Domain-Specific Language:
 - *Allow programming of packet forwarding planes*
- Design Principles:
 - *Target / Protocol independence, Reconfigurability*

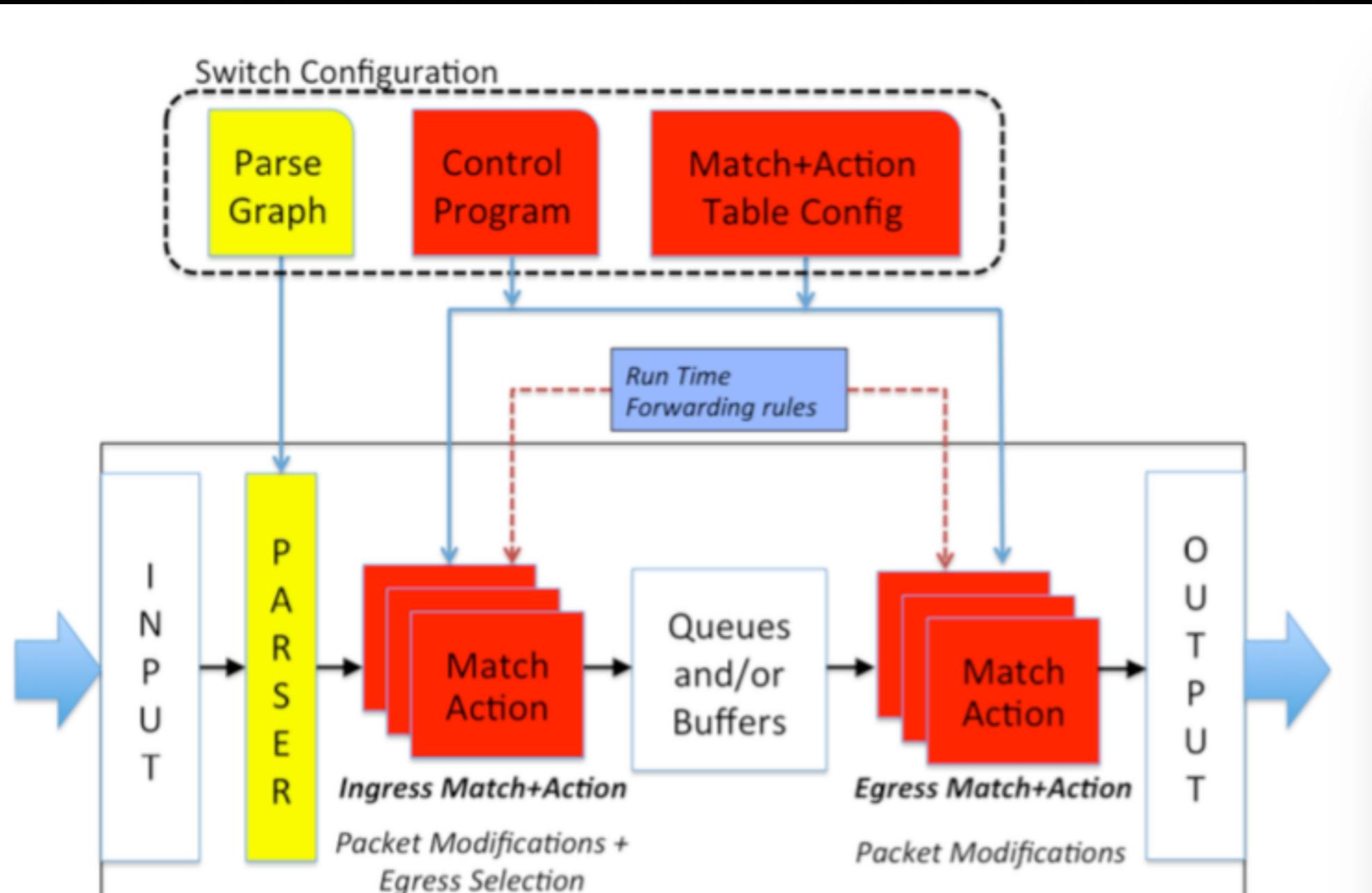


Figure 1: Abstract Forwarding Model

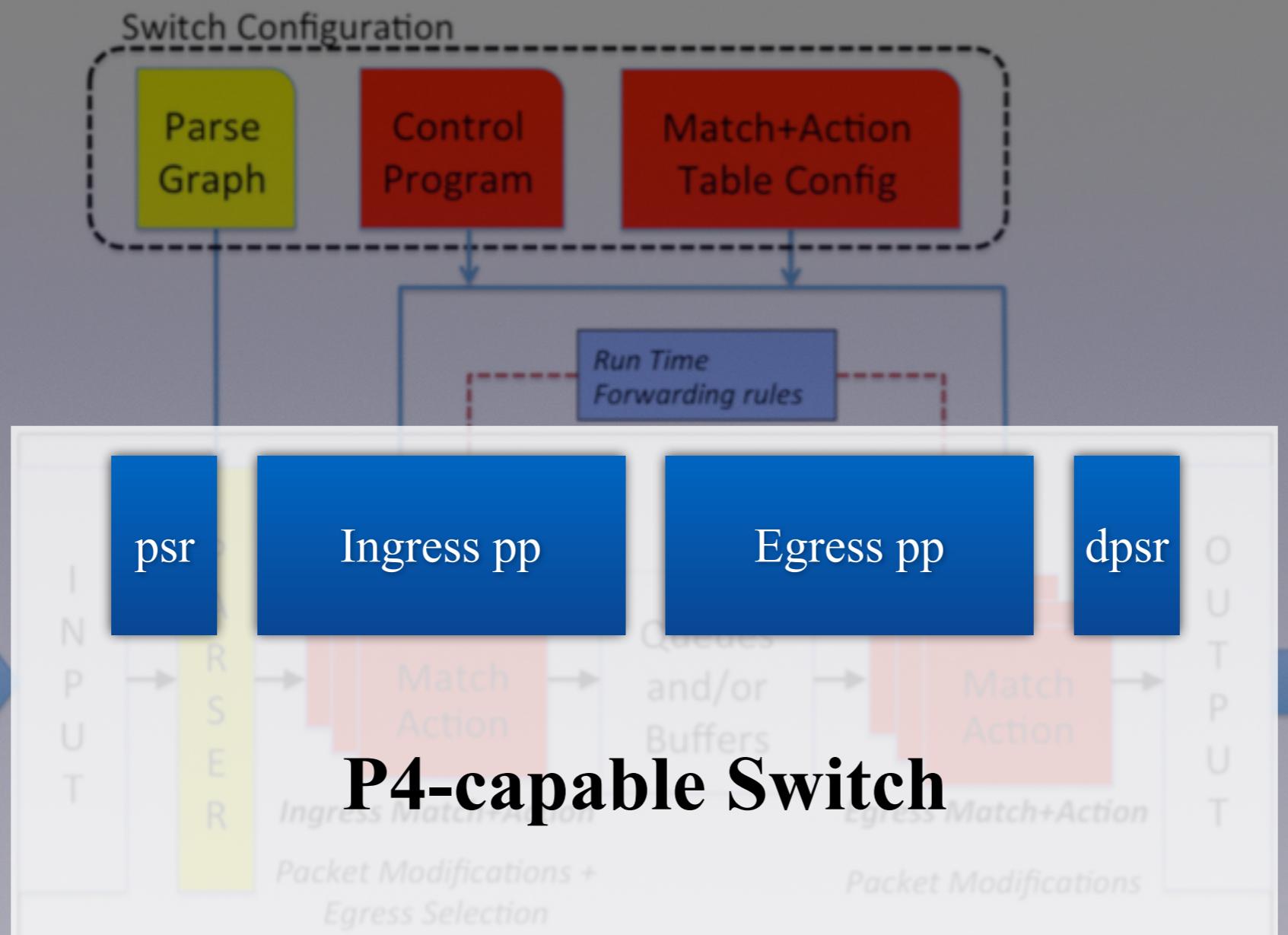


Figure 1: Abstract Forwarding Model

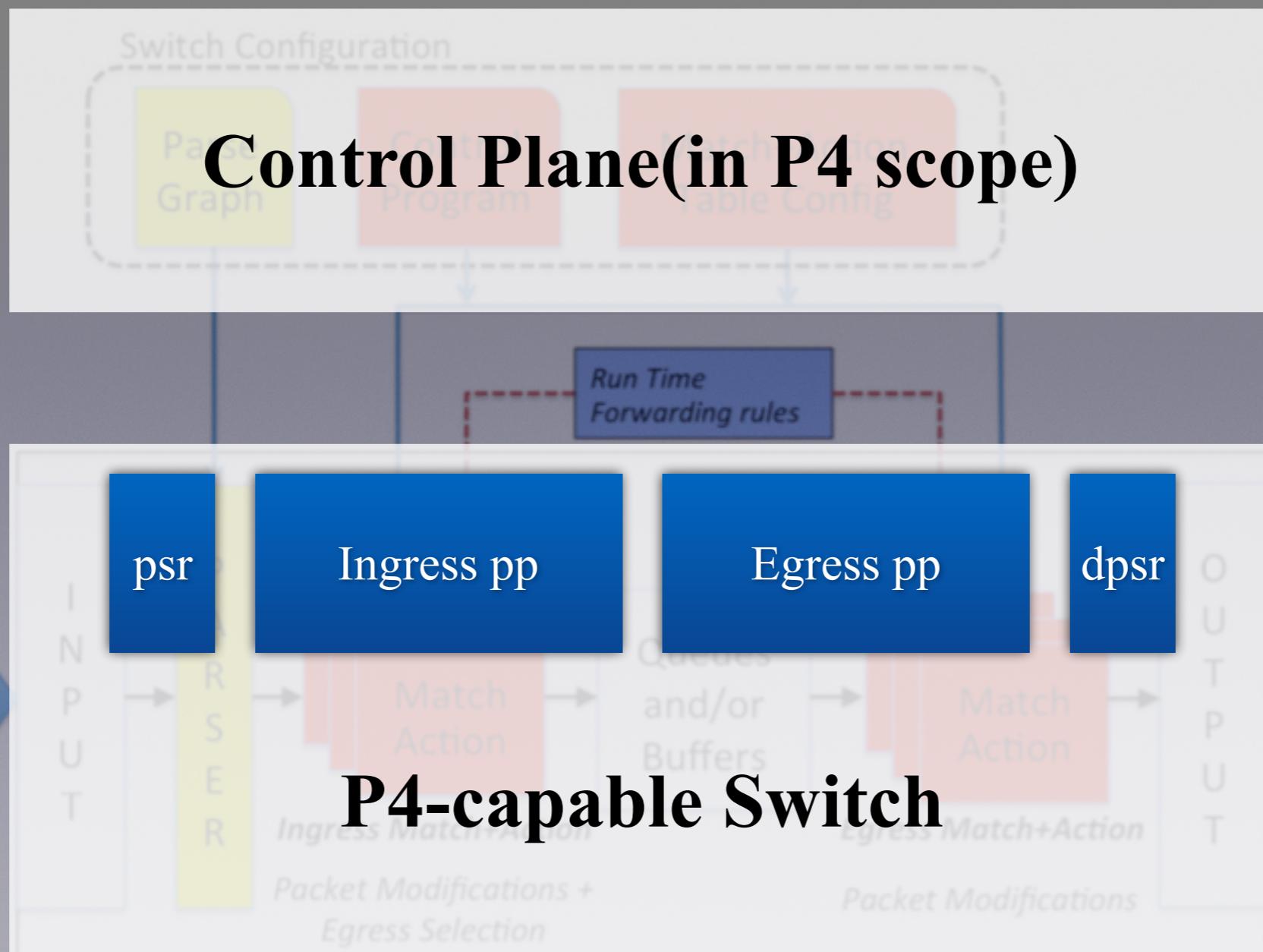


Figure 1: Abstract Forwarding Model

Control Plane(in P4 scope)

psr

Ingress pp

Egress pp

dpsr

P4-capable Switch

```
// Flexible Tables
#define LPM_TABLE_SIZE 16384
#define IPV6_LPM_TABLE_SIZE 4096
#define HOST_TABLE_SIZE 131072
#define IPV6_HOST_TABLE_SIZE 32768
...
header_type routing_metadata_t {
    fields {
        bd : BD_BIT_WIDTH; /* bridge domain */
        vrf : VRF_BIT_WIDTH; /* routing domain */
        v6_vrf : VRF_BIT_WIDTH; /* routing domain */
    ...
}

table bridge_domain {
    reads {
        routing_metadata.bd : exact;
    }
    actions {
        nop; // Not used
        bd_set;
    ...
}
```

Control Plane(in P4 scope)

“The backbone of pipeline is ...”

Uploading P4 program



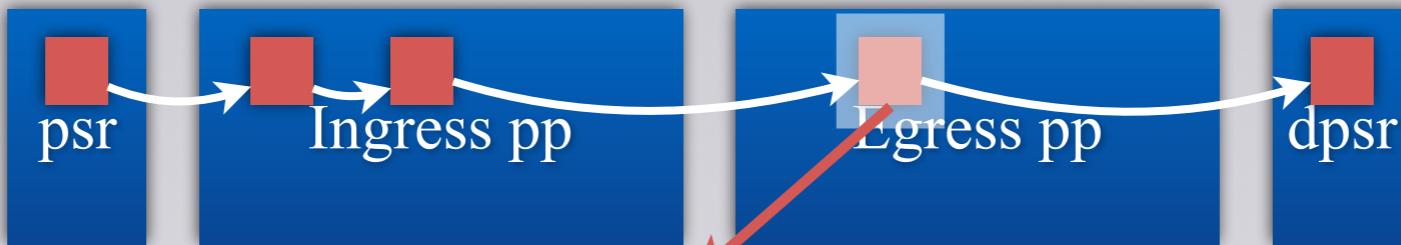
P4-capable Switch
(Offline or Runtime)

Control Plane(in P4 scope)

“table_add l3_forward forward 10.0.0.2/32 => 2”

“That’s how I process the packets”

Populating Control Rules



Forward the packets with dstIP 10.0.0.2/32 to port 2

**P4-capable Switch
(Offline or Runtime)**

Network Load Balancing

- A Classic Network Technology
 - Improving the **utilization rate** of network links
 - Increasing the **total amount of bandwidth available**

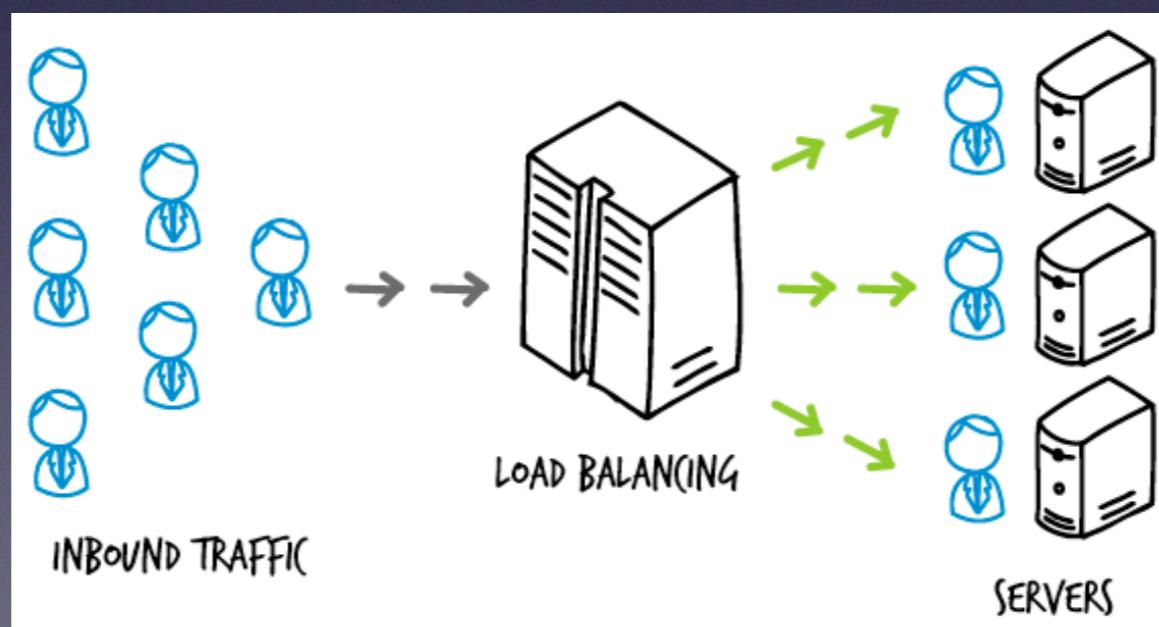


Figure: Network Load Balancing

Outline

- 1.Background: P4 and Load Balance
- *2.Introduction: What's New Balance?*
- 3.Demo Presentation
- 4.Conclusion



New Balance(NewB)

- A P4-based Network Load Balancing Application
- Scheduling traffics to avoid *heavy switch loads*
- Design Principles:
 - Previously record all the possible paths
 - Dynamically choose back-up paths



New Balance(NewB)

- A P4-based Network Load Balancing Application
- The P4 program:
 - *Counters for recording total received packet number*
 - *Match-Action Tables for identifying and forwarding flows*



A blurred screenshot of a computer screen displaying code, likely the P4 program for New Balance. The code is written in Python and includes imports for os, re, and json. It contains variables like tempString, dataCal, ricName, flagCheckRicname, myEvent, vtest, and a dictionary named ccData. The code also includes file operations such as mkdirs, makedirs, and writing to 'Input4RTAVTEST'. The text is mostly illegible due to blurring.

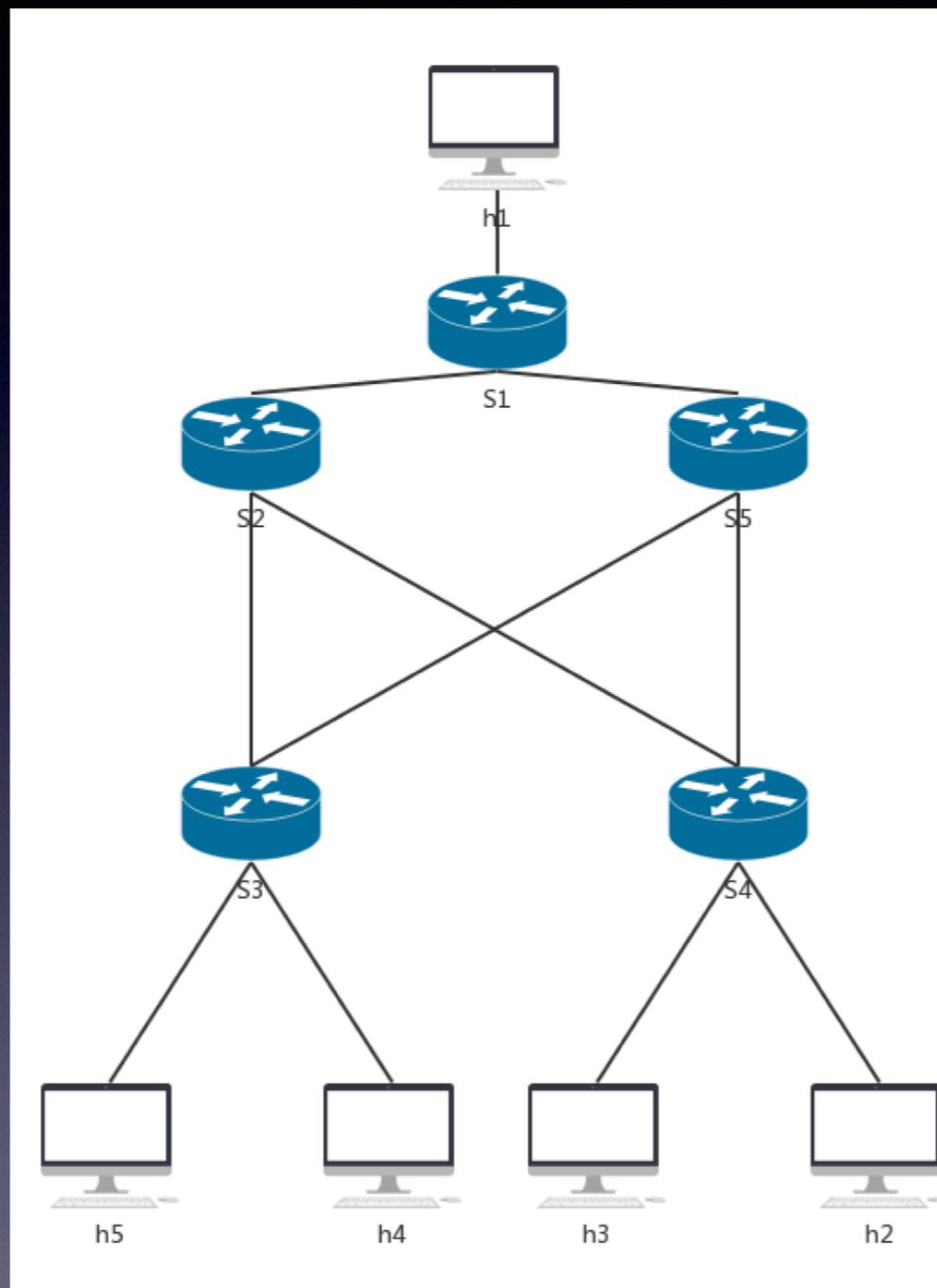
New Balance(NewB)

- A P4-based Network Load Balancing Application
- The NewB Controller:
 - *Acquire the counter value of switches per second*
 - *Calculate the processing rate of switch*
 - *Runtime traffic scheduling*

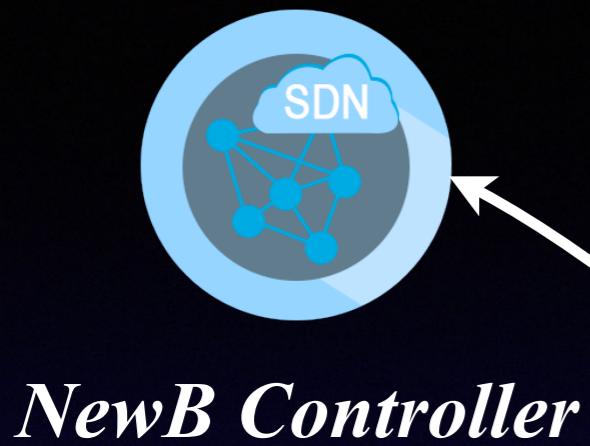


BMv2:
OvS-based P4
Software Switch

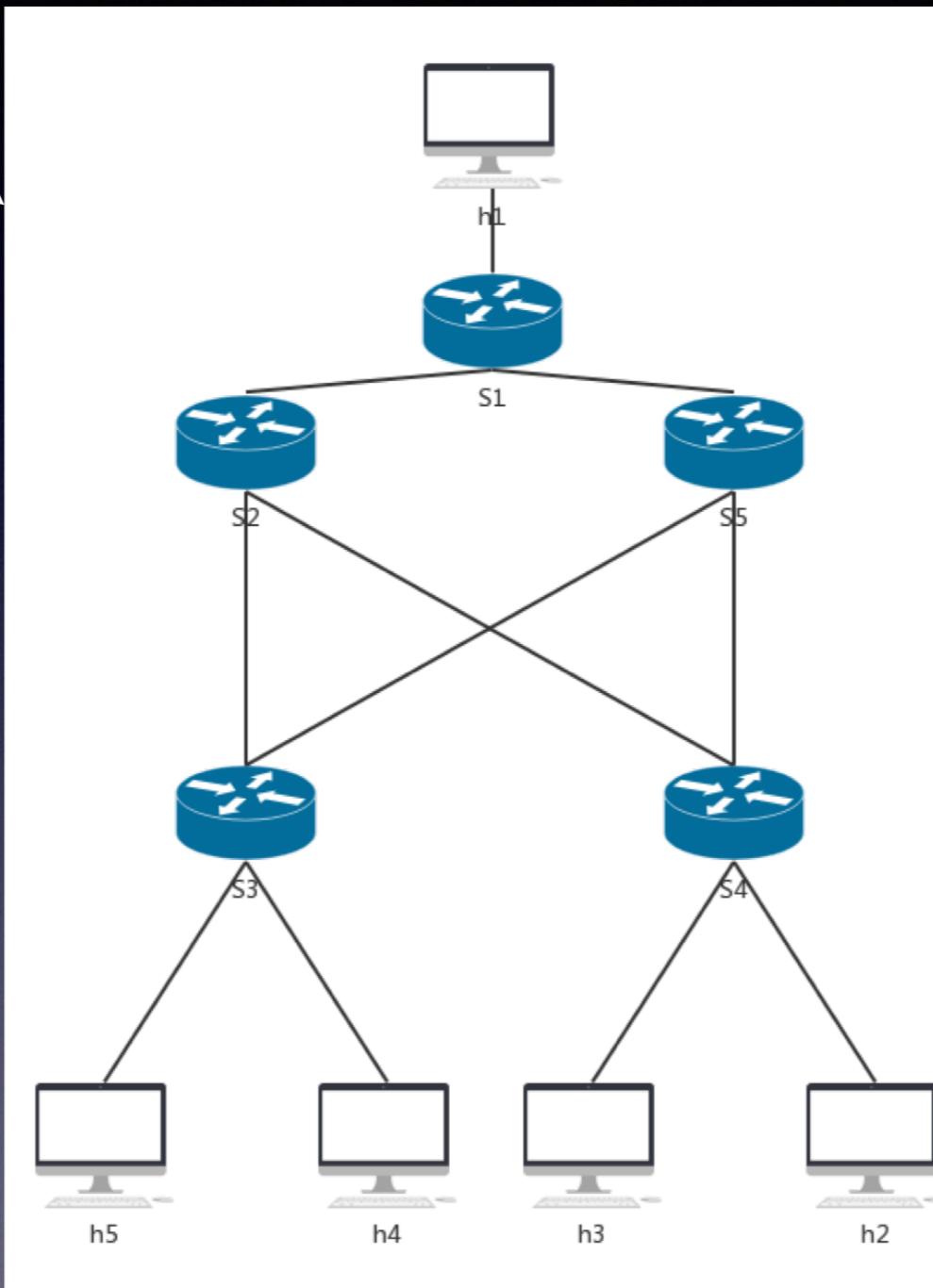
Simulated in Mininet



5x BMv2 Switches and 5x Hosts



NewB Controller



BMv2:
OvS-based P4
Software Switch

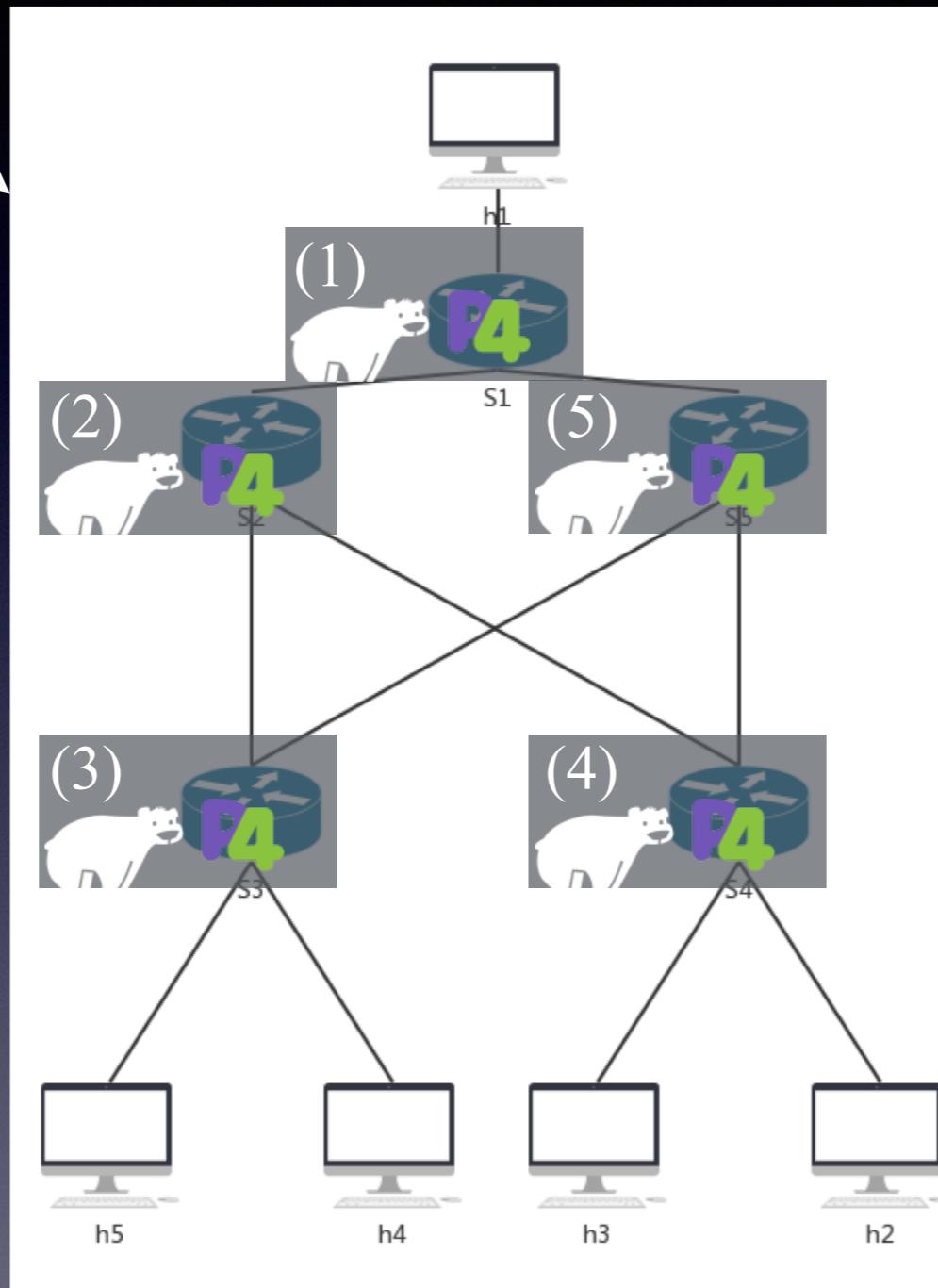
Simulated in Mininet

5x BMv2 Switches and 5x Hosts



NewB Controller

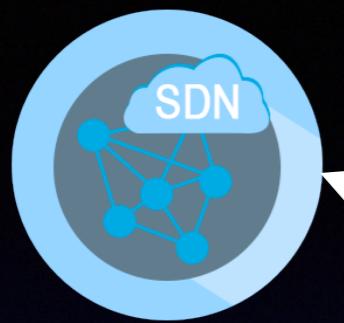
(1) Loading P4 program



BMv2:
OvS-based P4
Software Switch

Simulated in Mininet

5x BMv2 Switches and 5x Hosts



NewB Controller

- (1) Loading P4 program
- (2) Inspect paths

path1: 1-2-3-h5

path2: 1-2-3-h4

path3: 1-2-4-h3

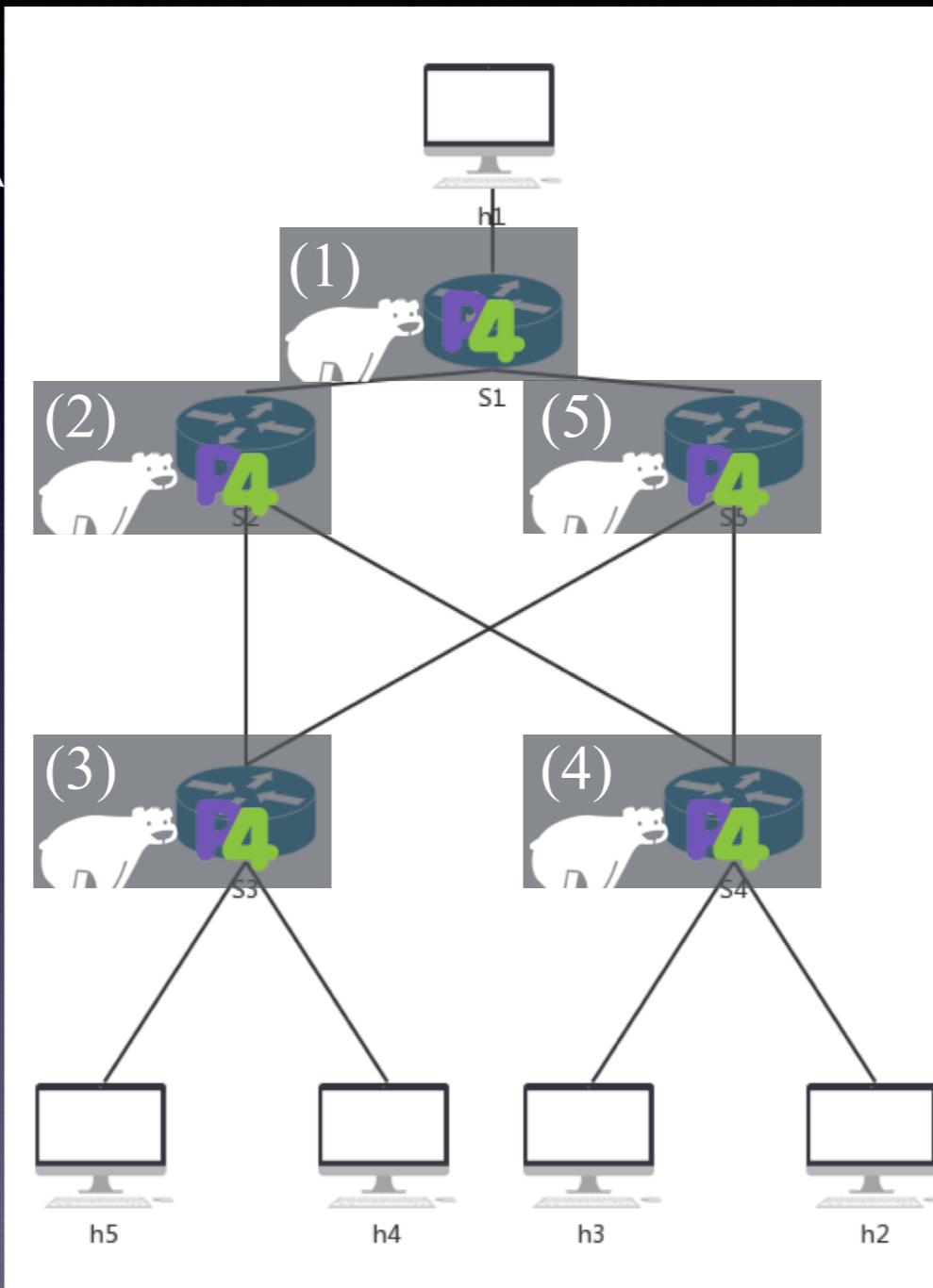
path4: 1-2-4-h2

path5: 1-5-3-h5

path6: 1-5-3-h4

path7: 1-5-4-h3

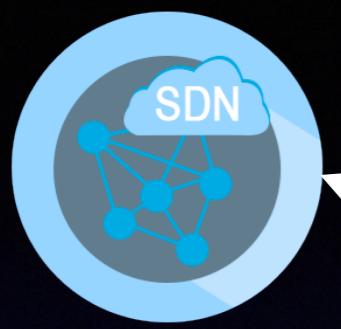
path8: 1-5-4-h2



BMv2:
OvS-based P4
Software Switch

Simulated in Mininet

5x BMv2 Switches and 5x Hosts



NewB Controller

- (1) Loading P4 program
- (2) Inspect paths
- (3) Read Counter per sec

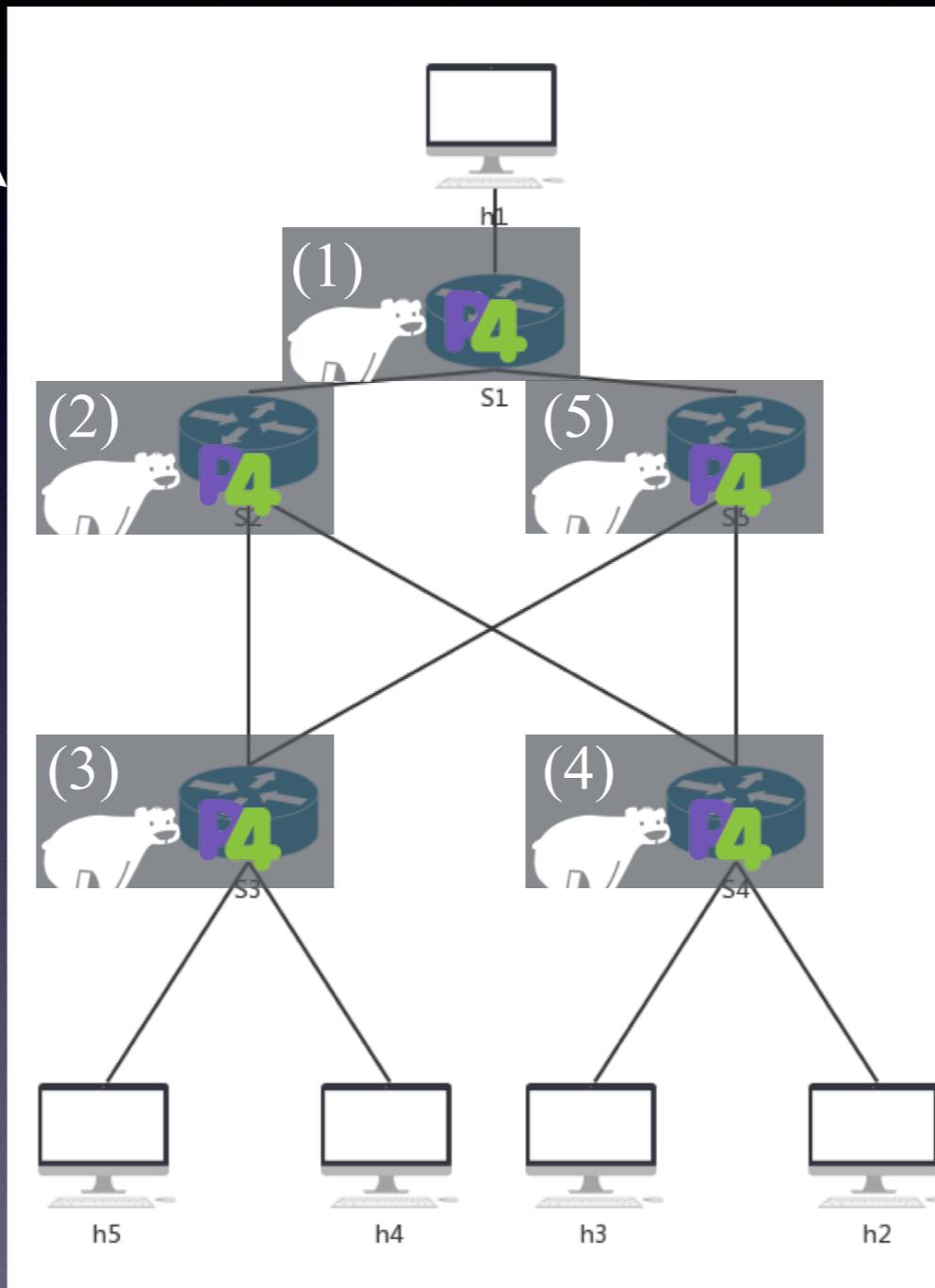
cnt1: 20

cnt2: **50** → warning!

cnt3: 20

cnt4: 0

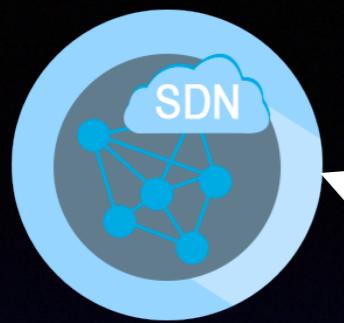
...



BMv2:
OvS-based P4
Software Switch

Simulated in Mininet

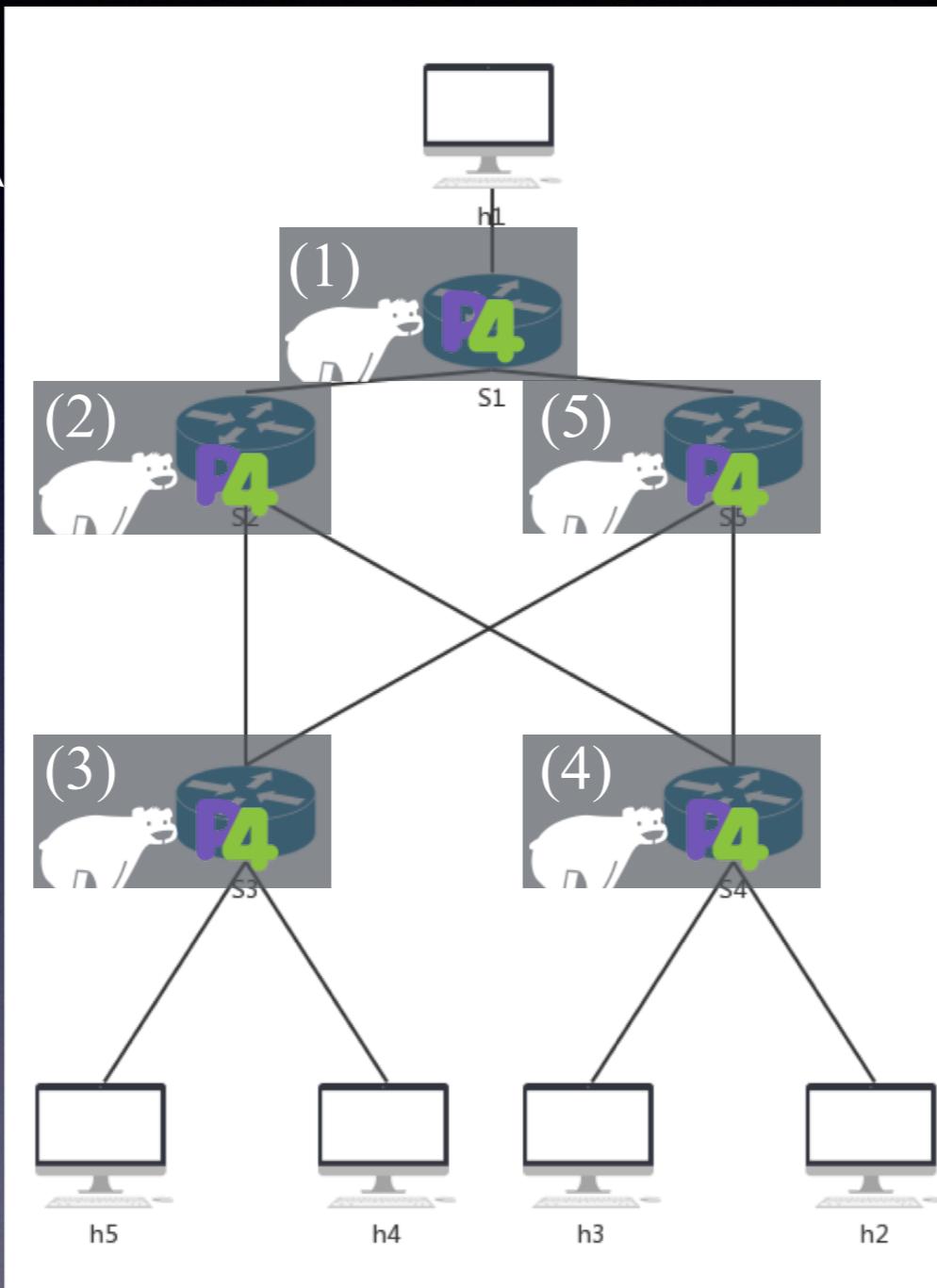
5x BMv2 Switches and 5x Hosts



NewB Controller

- (1) Loading P4 program
- (2) Inspect paths
- (3) Read Counter per sec
- (4) Calculate new path

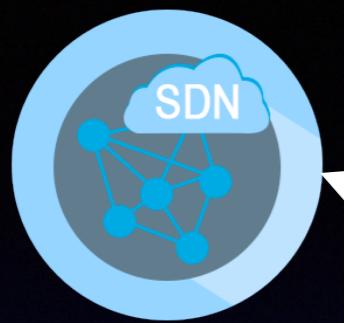
path1: 1-2-3-h5



5x BMv2 Switches and 5x Hosts

BMv2:
OvS-based P4
Software Switch

Simulated in Mininet



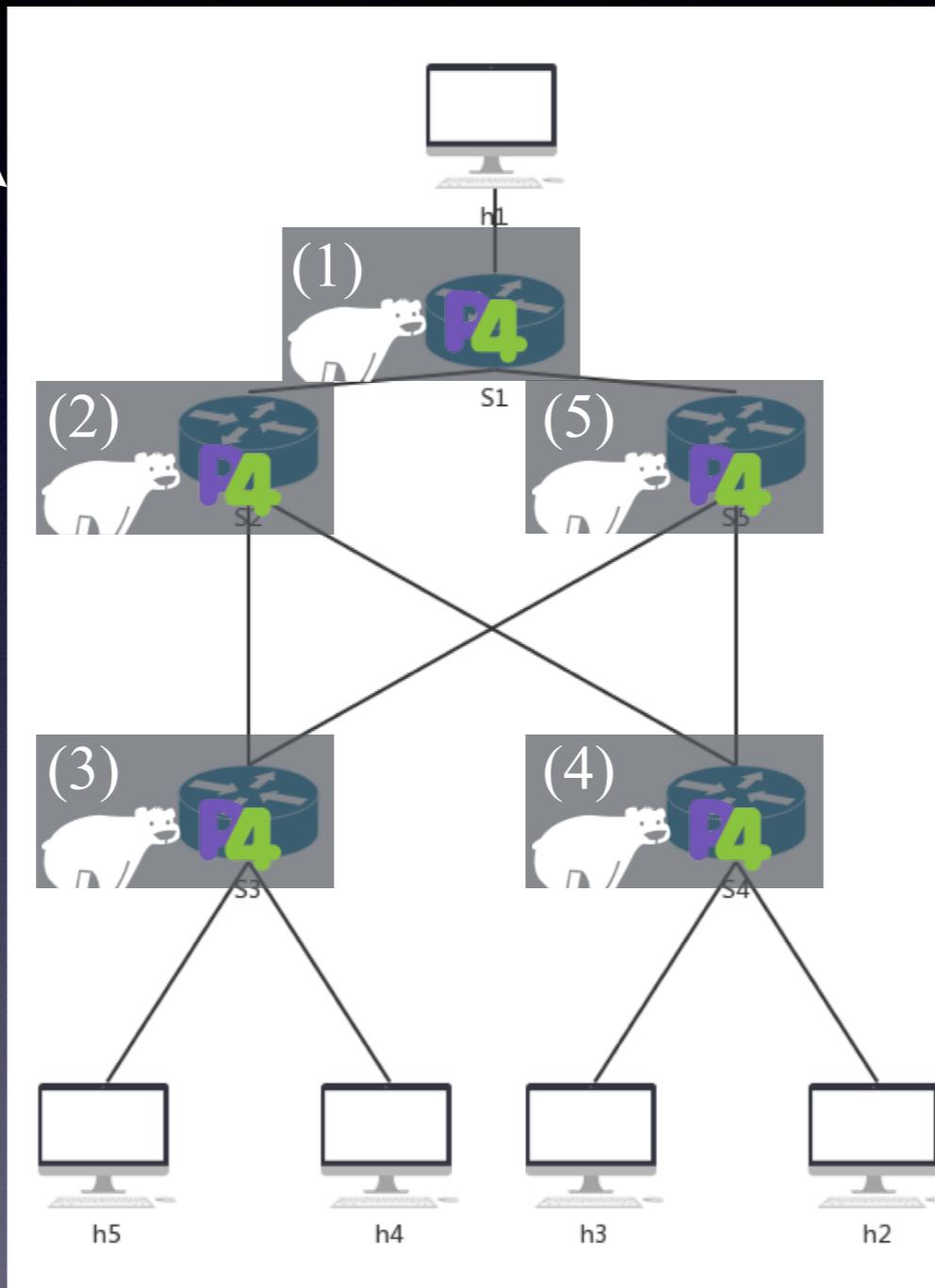
NewB Controller

- (1) Loading P4 program
- (2) Inspect paths
- (3) Read Counter per sec
- (4) Calculate new path

path1: 1-2-3-h5



path5: 1-5-3-h5



BMv2:
OvS-based P4
Software Switch

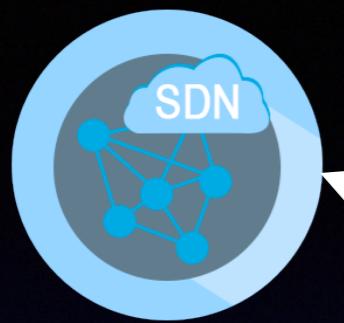
Simulated in Mininet

5x BMv2 Switches and 5x Hosts

Validation

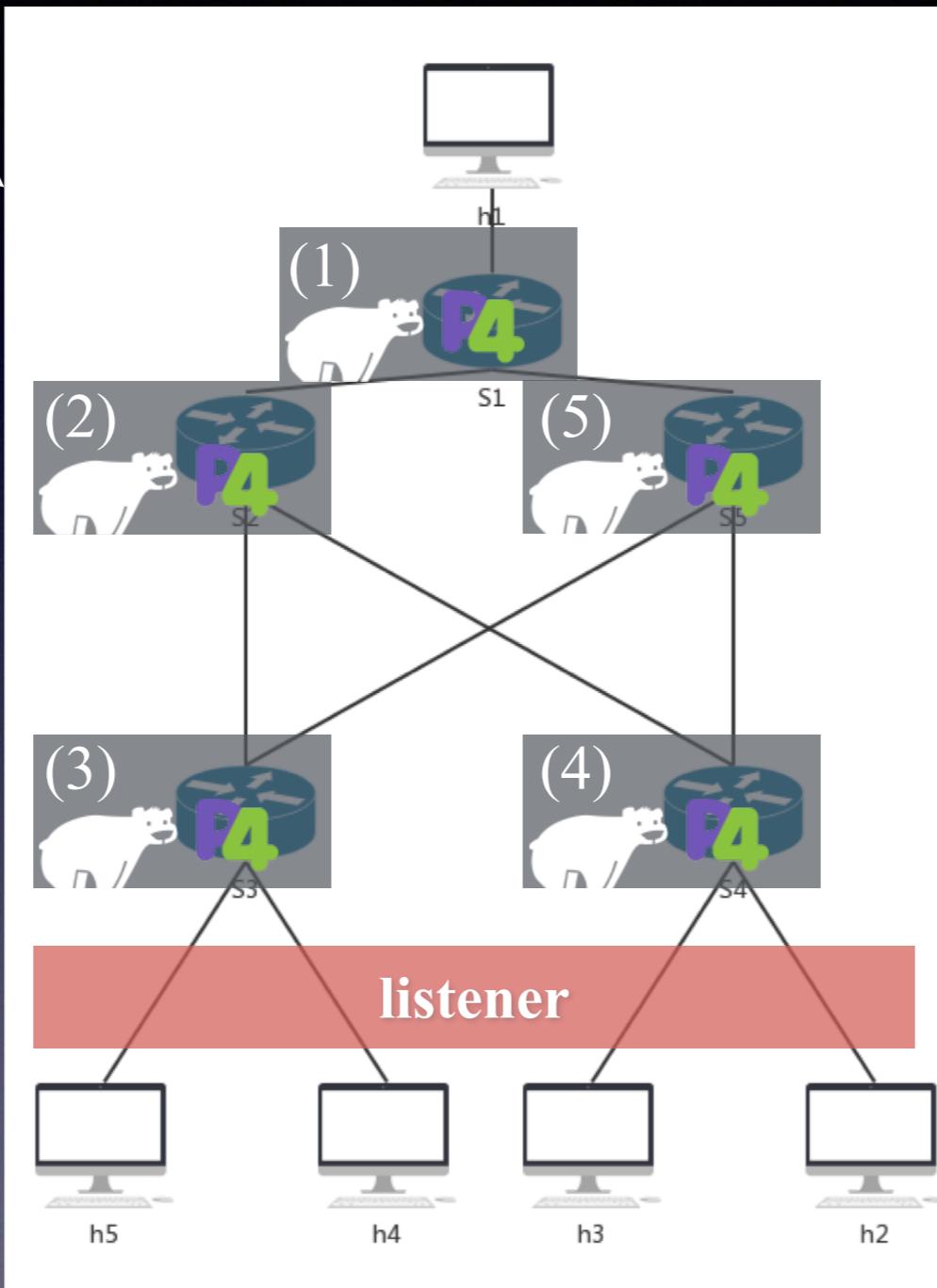
- Listener:
 - *Listen to all the possible server ports*
 - *Report packet number per second, in order to validate the behavior at runtime.*





NewB Controller

- (1) Loading P4 program
- (2) Inspect paths
- (3) Read Counter per sec
- (4) Calculate new path



BMv2:
OvS-based P4
Software Switch

Simulated in Mininet

5x BMv2 Switches and 5x Hosts

Outline

- 1.Background: P4 and Load Balance
- 2.Introduction: What's New Balance?
- 3.*Demo Presentation*
- 4.Conclusion



The image shows a screenshot of an Ubuntu desktop environment. On the left, there is a vertical dock with various application icons. In the center, a terminal window is open, displaying a directory listing. The terminal window has a dark background and light-colored text. The title bar of the terminal window says "root@ubuntu: /home/wasdns/sdn-course/demo". The command entered was "ls", which listed several files and scripts: cleanup.sh, controller.py, init_switches.sh, p4src, read_counter.py, read_counter.sh, README, reset_all_switches.sh, reset_counter.sh, rules, run_14_demo.sh, scripts, set_path.py, set_path.sh, topo.txt, and utils. The file "simple_switch_CLI" is also mentioned in the output. The desktop background is a dark orange gradient.

```
root@ubuntu:/home/wasdns/sdn-course/demo
root@ubuntu:/home/wasdns/sdn-course# cd demo/
root@ubuntu:/home/wasdns/sdn-course/demo# ls
cleanup.sh      read_counter.sh      run_14_demo.sh      topo.txt
controller.py   README             scripts            utils
init_switches.sh reset_all_switches.sh  set_path.py
p4src          reset_counter.sh    set_path.sh
read_counter.py rules              simple_switch_CLI
root@ubuntu:/home/wasdns/sdn-course/demo#
```

Figure: Directory

A screenshot of an Ubuntu desktop environment. The terminal window is open and shows the output of a command. The terminal title is "root@ubuntu: /home/wasdns/sdn-course/demo". The output lists five hosts (h1, h2, h3, h4, h5) with their default interfaces and MAC addresses. It also shows the "Ready!" message and the "Starting CLI" prompt. The desktop background is orange, and the Unity interface is visible on the left.

```
root@ubuntu: /home/wasdns/sdn-course/demo
disable ipv6
*****
h1
default interface: eth0 10.0.0.1      00:00:00:00:00:01
*****
h2
default interface: eth0 10.0.0.2      00:00:00:00:00:02
*****
h3
default interface: eth0 10.0.0.3      00:00:00:00:00:03
*****
h4
default interface: eth0 10.0.0.4      00:00:00:00:00:04
*****
h5
default interface: eth0 10.0.0.5      00:00:00:00:00:05
*****
Ready !
*** Starting CLI:
mininet> 
```

Figure: Start Mininet

The screenshot shows a Linux desktop environment, likely Ubuntu, with a terminal window open. The terminal window title is "root@ubuntu: /home/wasdns/sdn-course/demo". The terminal displays the output of a script named "set_path.py" which is configured to set default paths for network interfaces h1 through h5. The script adds entries to an LPM table named "l3_forward". The entries are defined by match keys (e.g., LPM-0a:00:00:02/32) and actions (e.g., forward, update_sw1, update_sw2). The process involves obtaining JSON from a switch and running control utilities for P4 table manipulation. The terminal also shows the user's prompt "mininet>" at the bottom.

```
match key: LPM-0a:00:00:02/32
action: forward
runtime data: 00:02
Entry has been added with handle 0
RuntimeCmd:
h1 Obtaining JSON from switch...
default interface: eth0Done
*****
Control utility for runtime P4 table manipulation
*****
RuntimeCmd: Adding entry to lpm match table l3_forward
h2 match key: LPM-0a:00:00:02/32
default interface: eth0action: update_sw1
*****
runtime data: 02 00:02
Entry has been added with handle 0
RuntimeCmd:
h3 Obtaining JSON from switch...
*****
Done
Control utility for runtime P4 table manipulation
h4 RuntimeCmd: Adding entry to lpm match table l3_forward
default interface: eth0match key: LPM-0a:00:00:02/32
*****
action: update_sw2
*****
runtime data: 03 00:03
h5 Entry has been added with handle 0
default interface: eth0RuntimeCmd:
*****
root@ubuntu:/home/wasdns/sdn-course/demo# ./set_path.py -p 1
Ready !
*** Starting CLI:
mininet>
```

Figure: Set default path

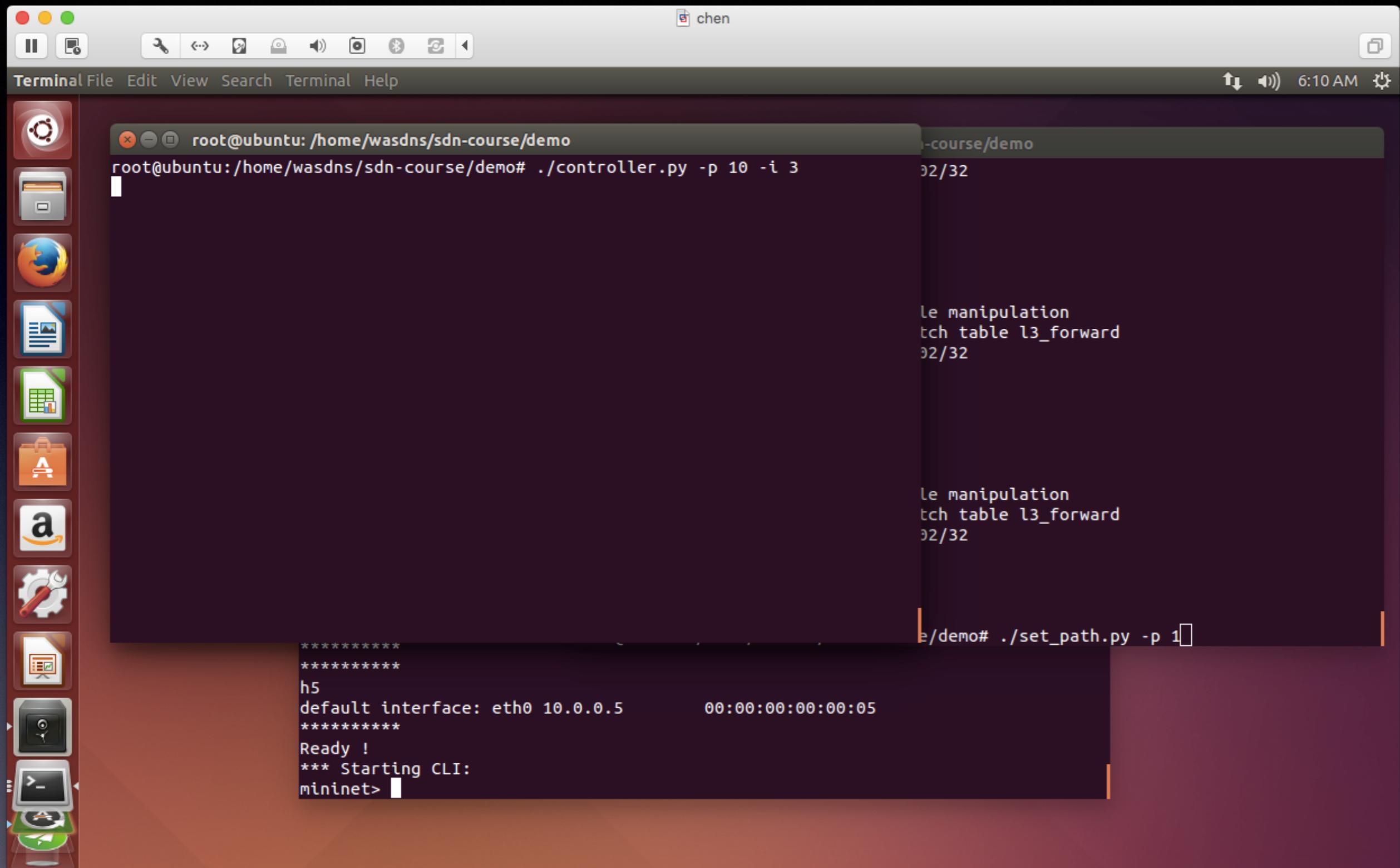


Figure: Start controller

The image shows a screenshot of an Ubuntu desktop environment. On the left, there is a vertical dock with various application icons. In the center, there are two terminal windows open.

The top terminal window is titled "root@ubuntu: /home/wasdns/sdn-course/demo". It displays the following output:

```
root@ubuntu:/home/wasdns/sdn-course/demo# ./controller.py -p 10 -i 3
Warnning: l2 s2 exceeds the threshold!
Warnning: l3 s3 exceeds the threshold!
Find an available path: s1-s5-s4-h3
Find an available path: s1-s5-s4-h2
Set new path successfully! Path: s1-s5-s4-h3

Warnning: l2 s5 exceeds the threshold!
Warnning: l3 s4 exceeds the threshold!
Find an available path: s1-s2-s3-h5
Find an available path: s1-s2-s3-h4
Set new path successfully! Path: s1-s2-s3-h4

Warnning: l2 s2 exceeds the threshold!
Warnning: l3 s3 exceeds the threshold!
Find an available path: s1-s5-s4-h3
Find an available path: s1-s5-s4-h2
Set new path successfully! Path: s1-s5-s4-h3
```

The bottom terminal window is also titled "root@ubuntu: /home/wasdns/sdn-course/demo". It displays the following output:

```
root@ubuntu:/home/wasdns/sdn-course/demo# ./utils/
listener.py      low_rate_sender.py  sender.py
root@ubuntu:/home/wasdns/sdn-course/demo# ./utils/sender.py -n 1000
```

Figure: Send packets

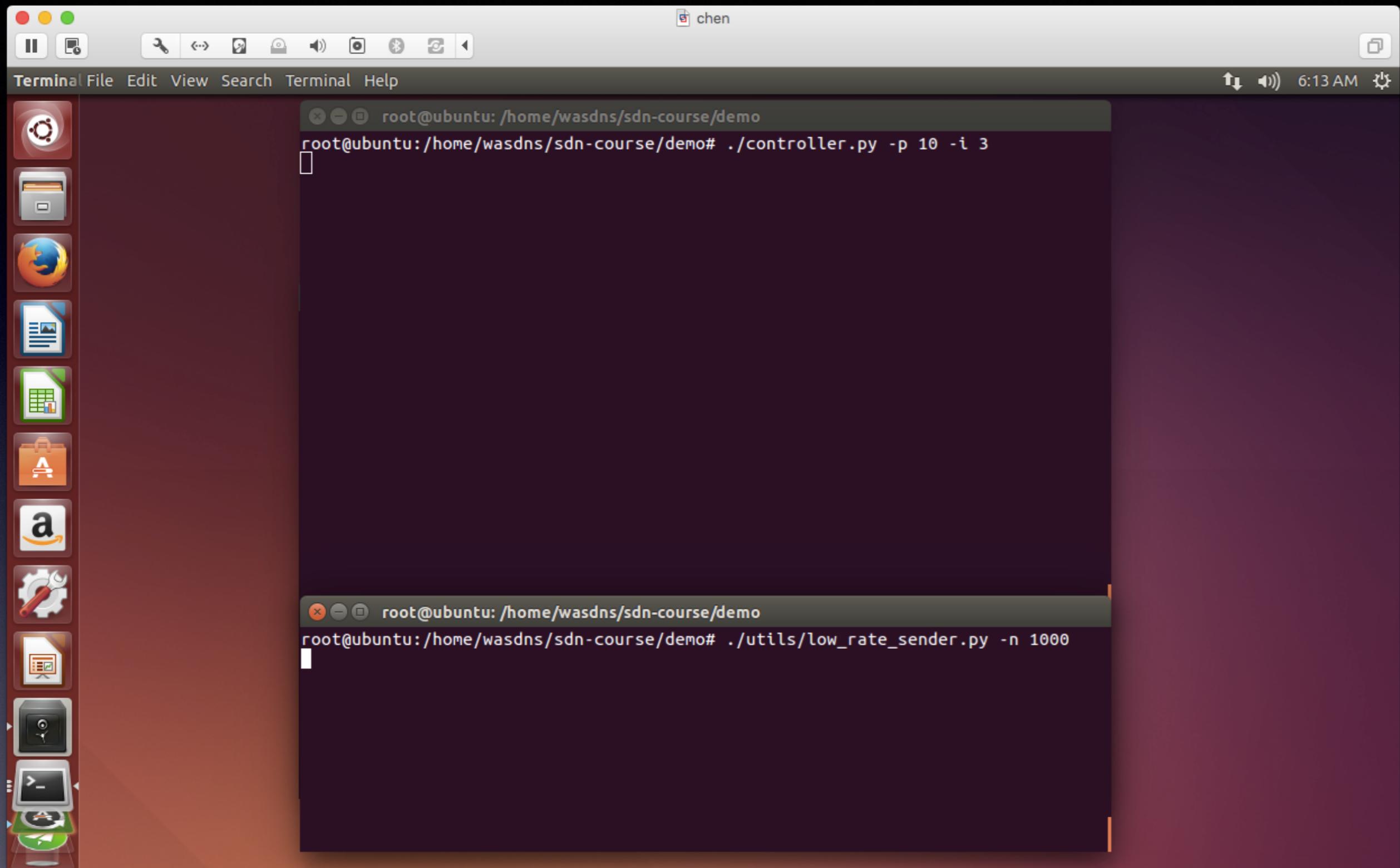


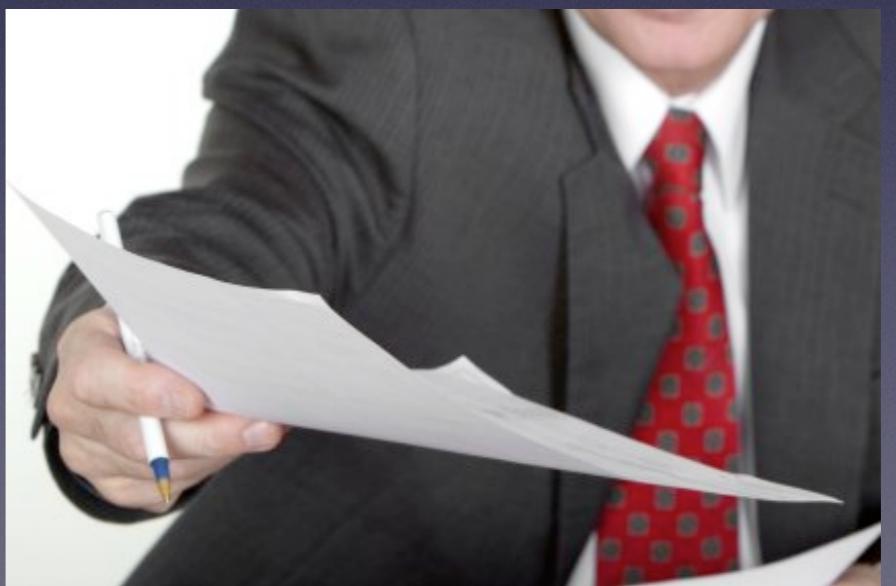
Figure: Send packets with low rate

This demo is available at Github:

https://github.com/Wasdns/new_balance

Outline

- 1.Background: P4 and Load Balance
- 2.Introduction: What's New Balance?
- 3.Demo Presentation
- 4.*Conclusion*



Conclusion

- Introducing P4 and Load Balance
- Presenting the mechanism of implementing NewB
- Illustrating how NewB works through demonstration



Thank you!

Group Members: 陈翔, 张昭锡, 胡武成, 孙浩楷, 钟梦真