



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

# Δίκτυα Καθοριζόμενα από Λογισμικό

Exercise Session:  
Introduction to Pyretic

Τμήμα Επιστήμης Υπολογιστών

# HY436: Software Defined Networks



Exercise Session

## **Introduction to Pyretic**

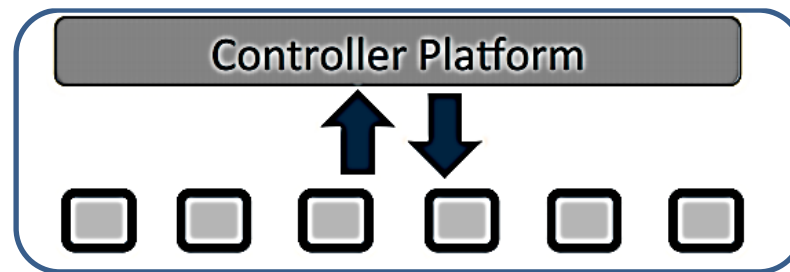
George Nomikos

15/10/2014

# Do we need other low-level controllers?

- Need to perform multiple independent tasks
- Need high-level mechanisms (abstractions) for reusability
- Need to handle packet events when the controller does not know how to figure them out
- Need to properly manipulate switch-level rules
  - Switches have limited space for rules
  - Cannot install all possible patterns

# Common Abstraction

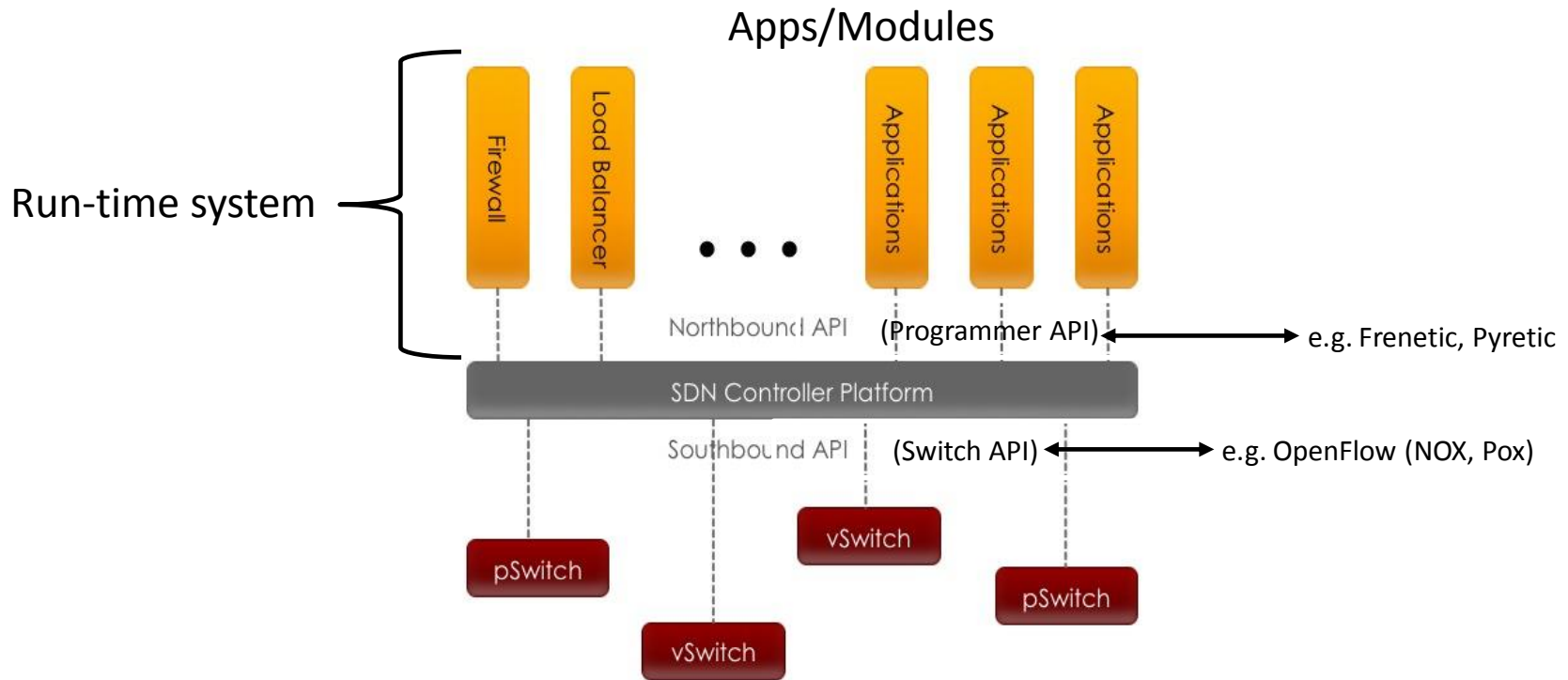


OpenFlow Protocol  
Controller-Switches

- Difficult to develop, test, debug, reuse, maintain
- **Modularity (key concept), a way to slice the traffic NOT the network**
- **Each module (slice) can control different portion of the traffic**

# Solution: “Northbound API”

A shift from low-level protocols (Openflow) to high level applications:



# SDN Languages

- Quite different from ordinary ones
- Program SDN Control (run-time systems)
- A run-time system translates higher level functions/policies to low-level rules on switches
- Each module works on its own abstract view of the network (topology abstraction)
- E.g. Frenetic (implemented in OCaml), Pyretic

# Python + Frenetic = Pyretic

- Frenetic: an SQL-like query language
- SDN Language and Runtime
  - Language: Way of expressing high-level policies
  - Runtime: Way of “compiling” those policies to low-level OpenFlow rules
- Embodies more sophisticated SDN applications
- Supports modular programming

# Primitives (1/3)

- Virtual Packet Headers
  - Packet metadata, a unified way of representing packet information.

For pyretic:

FIELD	TYPE	EXAMPLE
switch	int	4
inport	int	3
outport	int	2
srcmac	EthAddr	EthAddr('00:00:00:00:00:01')
dstmac	EthAddr	EthAddr('00:00:00:00:00:03')
srcip	IPAddr,string	IPAddr('10.0.1.1'), '10.0.0/24'
dstip	IPAddr,string	IPAddr('10.0.1.2'), '10.0.1/24'
tos	int	0
srcport	int	80
dstport	int	8080
ethype	int	1
protocol	int	7
vlan_id	int	0
vlan_pcp	int	0



# Primitives (2/3)

- Predicates
  - Expressions/Functions (static or dynamic) to match traffic policies
  - Each module can control a different portion of the traffic (based on packet-header fields, e.g. source-ip)

For pyretic:

```
match(f=v): filters only those packets whose header field f's value matches v
~A:        negates a predicate
A & B:     logical intersection of predicates A and B
A | B:     logical union of predicates A and B
fwd(a):    forward packet out port a
flood():   send all packets to all ports on a network minimum spanning tree,
           except for the input port
drop:      produces the empty set (no packet is output)
if_(p,A,B): if packet filtered by p, then use A, otherwise use B
```

```
match(srcip=IPAddr("1.1.1.2"))          split = if_(match(dstip=IPAddr('10.0.0.1')),fwd(1),fwd(2))
match(srcmac=EthAddr("00:00:00:00:00:01"))
match(switch=4)
```

# Primitives (3/3)

- Composition
  - **Sequential**
    - Perform action on one packet, followed by another action in sequence
    - E.g.: Load balancing, Routing
  - **Parallel**
    - Perform actions on one packet in parallel
    - E.g.: Monitor and Routing
  - No priorities needed
  - Overcome potential interference among actions

For pyretic:

```
A >> B:      A's output becomes B's input (sequential composition)
A + B:      A's output and B's output are combined in parallel (e.g., run a
            monitoring module in parallel with a routing module, without the need
            for sequential composition)
```

Example:

```
R = (match(dstip='2.2.2.8') >> fwd(1)) +
    (match(dstip='2.2.2.9') >> fwd(2))
```

# How to make a complete application?

We need:

1. Reusable modules that all process the same traffic (Virtual packet headers)
2. A proper slicing of the traffic (Predicates)
3. Module interaction without conflicts (Composition)

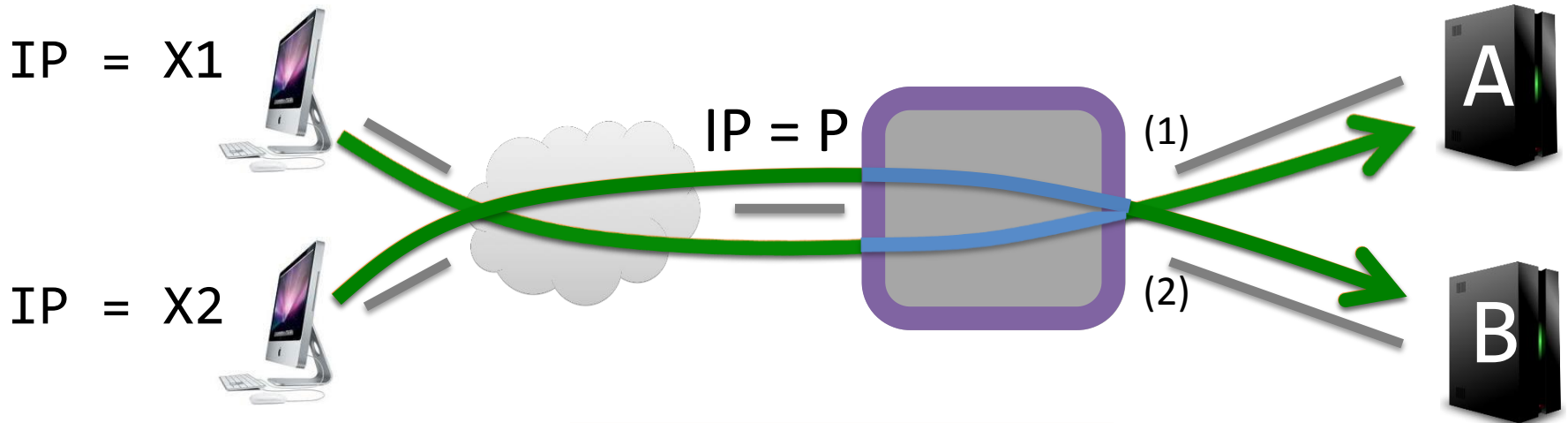
Examples of applications

1. Security applications
2. Resource management and control
3. Large virtual switches

# Let's see an example

What is this?

## Load Balancer



```
(match(srcip=X1,dstip=P)>>mod(dstip=A)>>fwd(1)) +  
(match(srcip=X2,dstip=P)>>mod(dstip=B)>>fwd(2))
```

What else we have here?

## Firewall

# Traffic Monitoring

Syntax	Summary
<code>packets(   limit=n,   group_by=[f1,f2,...])</code>	callback on every packet received for up to <code>n</code> packets identical on fields <code>f1,f2,...</code>
<code>count_packets(   interval=t,   group_by= [f1,f2,...])</code>	count every packet received callback every <code>t</code> seconds providing count for each group
<code>count_bytes(   interval=t,   group_by=[ f1,f2,...])</code>	count every byte received callback every <code>t</code> seconds providing count for each group

## Scenario:

- Monitor web-traffic requests
- Tcp port: 80

Step 1: `Q = packets(limit=1,group_by=['srcip'])`

Step 2: `def printer(pkt):  
 print pkt`

`Q.register_callback(printer)`

Step 3: `match(dstport=80) >> Q`

# Τέλος Ενότητας



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Πανεπιστήμιο Κρήτης**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «**Εκπαίδευση και Δια Βίου Μάθηση**» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



**Σημειώματα**



# Σημείωμα αδειοδότησης

- Το παρόν υλικό διατίθεται με τους όρους της άδειας χρήσης Creative Commons Αναφορά, Μη Εμπορική Χρήση, Όχι Παράγωγο Έργο 4.0 [1] ή μεταγενέστερη, Διεθνής Έκδοση. Εξαιρούνται τα αυτοτελή έργα τρίτων π.χ. φωτογραφίες, διαγράμματα κ.λ.π., τα οποία εμπεριέχονται σε αυτό και τα οποία αναφέρονται μαζί με τους όρους χρήσης τους στο «Σημείωμα Χρήσης Έργων Τρίτων».



[1] <http://creativecommons.org/licenses/by-nc-nd/4.0/>

- Ως **Μη Εμπορική** ορίζεται η χρήση:
  - που δεν περιλαμβάνει άμεσο ή έμμεσο οικονομικό όφελος από την χρήση του έργου, για το διανομέα του έργου και αδειοδόχο
  - που δεν περιλαμβάνει οικονομική συναλλαγή ως προϋπόθεση για τη χρήση ή πρόσβαση στο έργο
  - που δεν προσπορίζει στο διανομέα του έργου και αδειοδόχο έμμεσο οικονομικό όφελος (π.χ. διαφημίσεις) από την προβολή του έργου σε διαδικτυακό τόπο
- Ο δικαιούχος μπορεί να παρέχει στον αδειοδόχο ξεχωριστή άδεια να χρησιμοποιεί το έργο για εμπορική χρήση, εφόσον αυτό του ζητηθεί.



# Σημείωμα Αναφοράς

Copyright Πανεπιστήμιο Κρήτης, Ξενοφώντας Δημητρόπουλος. «Δίκτυα Καθοριζόμενα από Λογισμικό. Exercise Session 4: Introduction to Pyretic». Έκδοση: 1.0. Ηράκλειο/Ρέθυμνο 2015. Διαθέσιμο από τη δικτυακή διεύθυνση: <http://www.csd.uoc.gr/~hy436/>