



ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ  
ΠΑΝΕΠΙΣΤΗΜΙΟ ΚΡΗΤΗΣ

# Δίκτυα Καθοριζόμενα από Λογισμικό

Exercise Session 2:

Introduction to Mininet  
and learning switch tutorial

Ξενοφώντας Δημητρόπουλος  
Τμήμα Επιστήμης Υπολογιστών

# HY436: Software Defined Networks



Exercise Session 2:  
Introduction to Mininet  
and learning switch tutorial

Dimitris Gkounis

01/10/2014

# Introduction to Mininet (I)

- Mininet: Network emulator:
  - End-hosts, switches and links on a single Linux kernel
  - Software network elements with behavior similar to hardware elements
- Why using Mininet?
  - Fast network set-up, easy to use, custom topologies
  - Can run real Linux programs
  - Mininet switches support OpenFlow
  - Open source, under active development

# Introduction to Mininet (II)

- Mininet drawbacks:
  - Resource limitation since all virtual network elements run on a single system.
  - No simulations: no virtual timing

# Introduction to Mininet (III)

- Help message for Mininet's startup options
  - `sudo mn -h`
- Start a minimal topology<sup>1</sup> and enter the Command Line Interface (CLI):
  - `sudo mn`
- Start a network with 1 OF switch, 3 hosts with MAC equal to their IP<sup>2</sup> and an external controller:
  - `sudo mn --topo single,3 --mac --switch ovsk --controller remote`

1. 1 OpenFlow switch, 2 hosts and the OF reference controller  
2. By default, hosts start with randomly assigned MAC addresses. This makes debugging tough

# Introduction to Mininet (IV)

- Naming in Mininet:
  - Usually: hosts h1..hN and switches s1...sN, host h1's default interface h1-eth0, switch s1's first port s1-eth1.
- Mininet CLI commands:
  - mininet> help                   display available commands
  - mininet> nodes                 display nodes
  - mininet> net                   display links
  - mininet> xterm hi hj           open terminals to hosts hi, hj
  - mininet> exit/quit             exit/quit Mininet

# Introduction to Mininet (V)

- Custom topology example:

```
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.cli import CLI

class SingleSwitchTopo(Topo):
    "Single switch connected to n hosts."
    def __init__(self, n=2, **opts):
        Topo.__init__(self, **opts)
        switch = self.addSwitch('s1')
        for h in range(n):
            host = self.addHost('h%s' % (h + 1))
            self.addLink(host, switch)

def simpleTest():
    topo = SingleSwitchTopo(n=4)
    net = Mininet(topo)
    net.start()
    print "Testing network connectivity"
    net.pingAll()
    CLI(net)
    net.stop()

if __name__ == '__main__':
    simpleTest()
```

# Reminder: What is OpenFlow?

- Software Defined Networking (SDN):
  - Separates *Control Plane* (computing routes and updating forwarding tables etc.) and *Data Plane* (forwarding, buffering, marking packets etc.) of networks
  - Network control is logically centralized
  - Open Interface between control and data plane
- OpenFlow:
  - A protocol which enables SDN



# Introduction to POX (I)

- What is POX?
  - Python-based OpenFlow controller framework
  - Supports OpenFlow specification version 1.0
  - Widely used and supported
  - Easy learning curve
  - Slow performance but easy and fast development of SDN controller applications:
    - Preferred in research, experimentation, demonstrations

# Introduction to POX (II)

- How does POX work?
  - Event-driven programming model: controller registers for events and developer implements event handlers
  - Controller functionalities implemented on *Components*
  - When an application is started, the *launch()* function is invoked. This is the function where the application registers event listeners or creates objects of any *Component* class: e.g.
    - <https://github.com/noxrepo/pox/blob/carp/pox/forwarding/hub.py>
    - [https://github.com/noxrepo/pox/blob/carp/pox/forwarding/l2\\_learning.py](https://github.com/noxrepo/pox/blob/carp/pox/forwarding/l2_learning.py)

# Learning Switch in POX (I)

- Remember Monday lecture's pseudocode:

when frame received at switch:

1. record incoming port, MAC address of sending host
2. index switch table using MAC destination address
3. **if** entry found for destination

**then** {

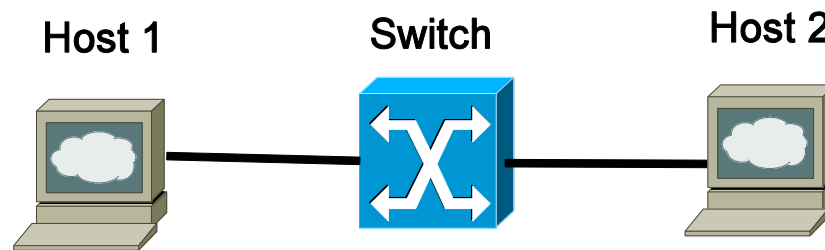
**if** destination on segment from which frame arrived  
    **then** drop frame

**else** forward frame on interface indicated by entry

**}** **else** flood /\*forward on all interfaces except arriving interface\*/

# Learning Switch in POX (II)

- `sudo mn --topo single,2 --mac --switch ovsk --controller remote`



- `~/pox$ ./pox.py log.level --DEBUG forwarding.l2_learning`

[https://github.com/noxrepo/pox/blob/carp/pox/forwarding/l2\\_learning.py](https://github.com/noxrepo/pox/blob/carp/pox/forwarding/l2_learning.py)

# Preparation for Exercise Session 3

- Review Python, Mininet, POX:
  - Check the course web site (Links) for more resources
- Next session: Assignment 1 - Transparent Load-Balancer

# 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 2: Introduction to Mininet and learning switch tutorial». Έκδοση: 1.0. Ηράκλειο/Ρέθυμνο 2015. Διαθέσιμο από τη δικτυακή διεύθυνση: <http://www.csd.uoc.gr/~hy436/>