



SDN hálózatok monitorozása

Kosztka Sándor

Konzulens: Megyesi Péter

sandor.kosztka@gmail.com | megyesi@tmit.bme.hu

A cél

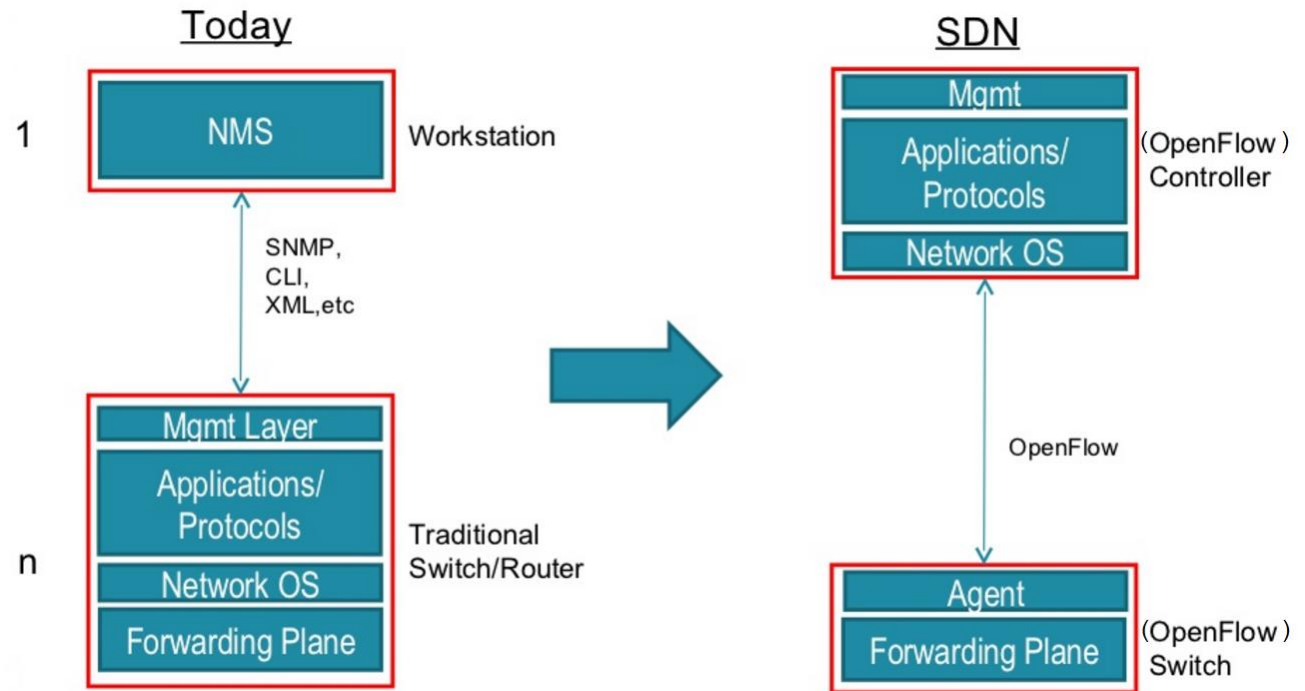
A félév során

- Megismerkedni az SDN hálózatok alapkonceptiójával
- Mérés környezet kiépítése Mininet hálózatszimulátorral
- A környezethez a szimulátor Python API-ját megismerni
- Több különböző kontrollert bekötni a hálózatba (POX, Floodlight, ONOS)
- Ezek milyen monitorozási lehetőségeket, API-kat nyújtanak

Software Defined Networking

Mi is ez?

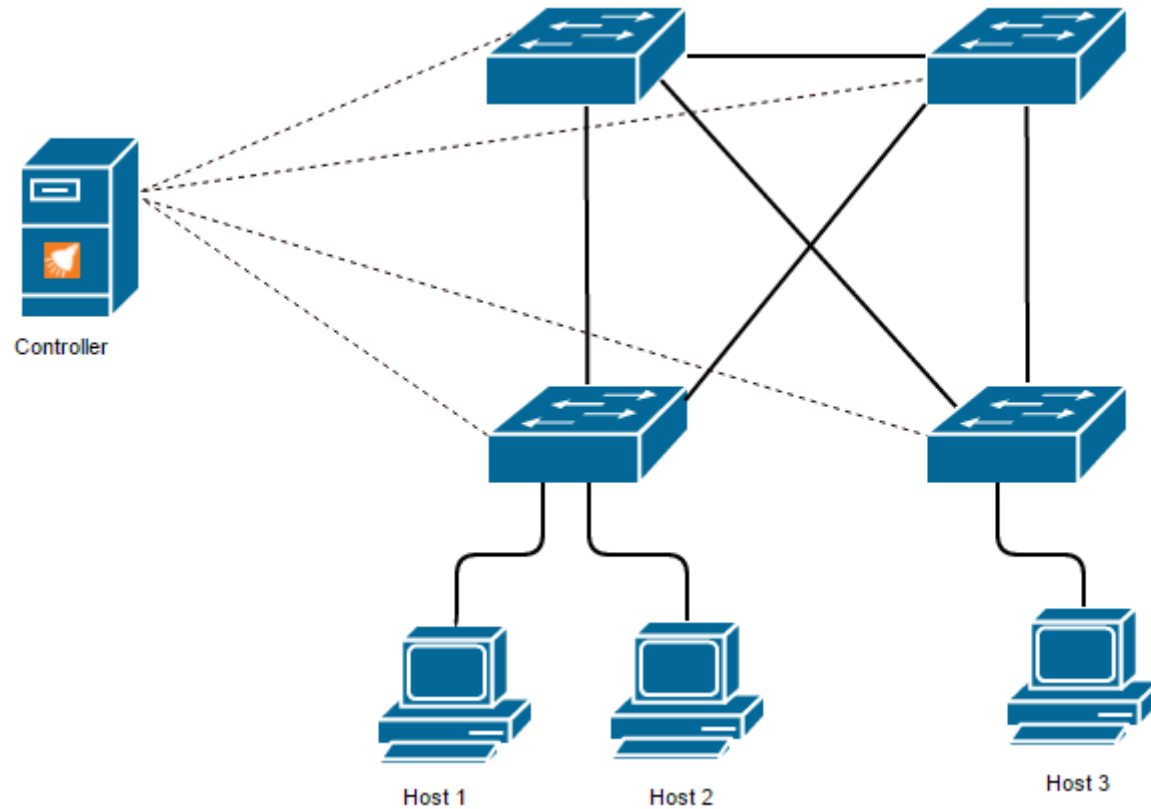
- Alapötlet, a mgmt és kontroll réteg különválasztása az adattól
- Centralizált kontroll
- Kontroller ↔ switch kommunikáció
- OVSDB, SNMP, NETCONF
- Openflow
a *legelterjedtebb* Southbound protokoll
- Northbound API



A mérési környezet

Mininet

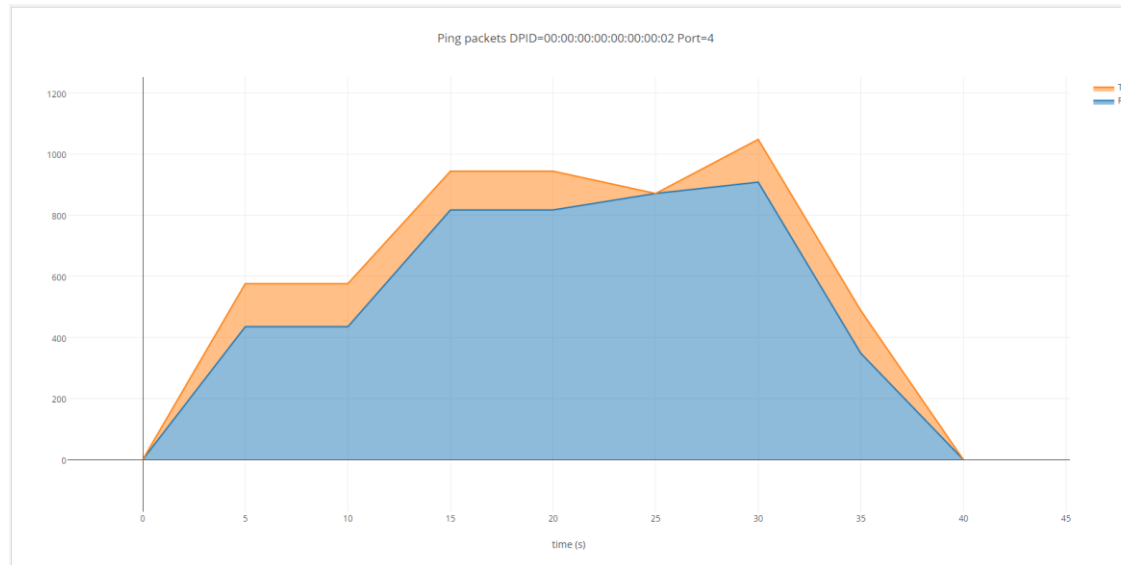
- Mininet
- DigitalOcean Dropleten (NYC)
- OS: Ubuntu 14.04 LTS
- POX & Floodlight itt lokálisan
- ONOS egy másik szerveren dockerben (BP)



Floodlight

REST API







- Van GUI-ja de az elég minimal
- DigitalOcean Droplet-en fut ő is
- A REST API-ján / Northbound API az alkalmazások felé, elég sokminden lekérdezhető
- A meterek még nem
- Egyszerű port statisztika:




```
"00:00:00:00:00:00:00:03":  
{  
  "port_reply": [1]  
    0: {  
      "version": "OF_13"  
      "port": [4] 0: {  
        {  
          "portNumber": "2"  
          "receivePackets": "74"  
          "transmitPackets": "116"  
          "receiveBytes": "5759"  
          "transmitBytes": "8970"  
          "receiveDropped": "0"  
          "transmitDropped": "0"  
          "receiveErrors": "0"  
          "transmitErrors": "0"  
          "receiveFrameErrors": "0"  
          "receiveOverrunErrors": "0"  
          "receiveCRCErrors": "0"  
          "collisions": "0"  
          "durationSec": "362"  
          "durationNsec": "532000000"  
        }  
      }  
    }  
  }  
}
```

Open Network OS WebGUI

- GUI
- Saját docker konténerben futott
- default nem futott az OpenFlow function benne
- Már ipari felhasználásra szánják



 of:000000000000000001

URI : of:000000000000000001

Type : Switch

Master ID : 10.0.10.11

Chassis ID : 1

Vendor : Nicira, Inc.

H/W Version : Open vSwitch



SW Version : 2.3.1

Protocol : OF_13







Serial # : None

Ports

Enabled	ID	Speed	Type	Egress Links	Name
false	Local	0	Copper		s1
true	1	10000	Copper	of:000000000000000002/1	s1-eth1
true	2	10000	Copper	of:000000000000000003/1	s1-eth2
true	3	10000	Copper	of:000000000000000004/1	s1-eth3
false	4	10000	Copper		s1-eth4
false	5	10000	Copper		s1-eth5

 Open Network Operating System [logout](#)

Ports for Device of:000000000000000001 (3 Ports total)



Port ID	Pkts Received	Pkts Sent	Bytes Received	Bytes Sent	Pkts Received Dropped	Pkts Sent Dropped	Duration (sec)
1	4,732	4,730	383,292	383,130	0	0	7320
2	4,732	4,730	383,292	383,130	0	0	7320
3	78,584	95,780	5,346,552	269,280,862	0	0	7320

ONOS

REST API

- Támogatja az 1.3-as OpenFlow újdonságait (meters)
- Csak a CLI-n lehet felvenni egyelőre
- Nem elég a kontrolleren felvenni, ezt a switchnek is támogatni kell
- Fel kell venni a sw konfigurációba, ez a példán még hiányzik

Meters for Device of:000000000000000001 (1 total)

Meter ID	App ID	State	Packets	Bytes
0x1	org.onosproject.cli.meterCmd	PENDING_ADD	0	0

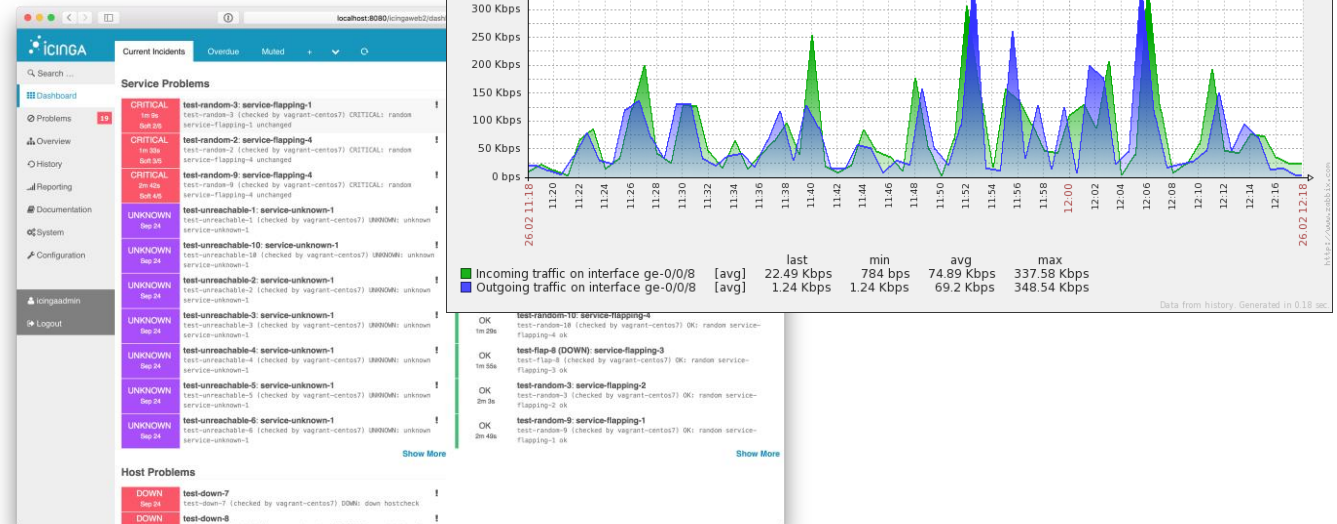
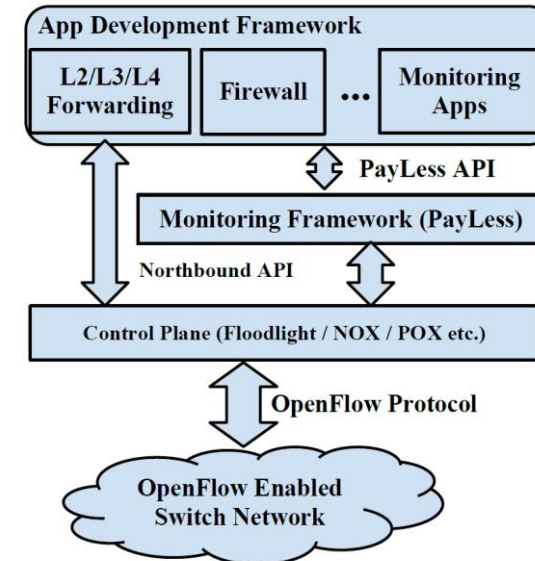
Bytes: 0 Packets: 0 Type: DROP

```
{
  "meters": [
    {
      "id": "of:000000000000000001",
      "life": 0,
      "packets": 0,
      "bytes": 0,
      "referenceCount": 0,
      "unit": "KB_PER_SEC",
      "burst": false,
      "deviceId": "1",
      "appId": "DefaultApplicationId{id=82, name=org.onosproject.cli.meterCmd}",
      "state": "PENDING_ADD",
      "bands": [
        {
          "type": "DROP",
          "rate": 500,
          "packets": 0,
          "bytes": 0,
          "burstSize": null
        }
      ]
    }
  ]
}
```

Monitoring rendszerek

Payless

- Ígéretesek tűnő keretrendszer, de sajnos csak papíron
- Monitoring rendszerek:
 - A JSON objektumok egyszerűen kezelhetők szkriptekkel
 - Monitoring rendszerekbe viszonylag könnyen integrálhatók (Nagios/Icinga, Zabbix, stb)



Összegzés

Hova jutottunk

- Megismerkedtem az SDN hálózatok alapjaival és néhány kontrollerrel
- Hogyan kell bekötni őket a hálózatba, miként kell kezelni az APIjaikat
- Hogyan kell lekérni néhány statisztikai adatot
- **Jövőbeni terv:**
megvalósítani egy konkrét monitoring rendszerbe történő bekötést

Köszönöm a figyelmet!