
Università degli Studi di Perugia

Dipartimento di Ingegneria

ingegneria informatica e robotica

# Uso di Mininet distribuito per creare una rete monitorata con NetFlow

**Autore:**
**Daniele Nanni Cirulli**

# Abstract

Questo progetto si focalizza sull'impiego di Mininet distribuito per la creazione di una rete simulata, integrando il protocollo NetFlow per il monitoraggio del traffico. L'obiettivo è simulare una rete distribuita e utilizzando NetFlow, ci proponiamo di analizzare il comportamento del traffico di rete simulato. Il report descrive il processo di configurazione di Mininet distribuito e l'integrazione di NetFlow.

# 1. Mininet

Mininet è un framework di emulazione di rete open-source progettato per la creazione e la simulazione di reti virtuali su un singolo computer o su un cluster di computer. Basato su Linux, Mininet offre un ambiente flessibile e scalabile per sperimentare e testare le configurazioni di rete senza la necessità di hardware fisico dedicato.

Le caratteristiche principali di Mininet includono:

- **Emulazione di Reti Virtuali**: Mininet consente agli utenti di creare reti virtuali complesse, includendo switch, router, host e collegamenti, tutto emulato in ambiente software. Questo permette agli utenti di sperimentare con diverse topologie di rete e configurazioni senza dover investire in hardware reale.
- **Flessibilità e Scalabilità**: Mininet è altamente flessibile e può essere adattato alle esigenze specifiche del progetto. Può essere eseguito su un singolo computer o distribuito su un cluster di macchine per simulazioni di grandi dimensioni. Questa scalabilità consente agli utenti di simulare reti di varie dimensioni e complessità.
- **Integrazione con OpenFlow**: Mininet supporta l'integrazione con OpenFlow, un protocollo di comunicazione utilizzato nei network definibili tramite software (SDN). Questo consente agli utenti di esplorare e sperimentare con le reti SDN, implementando e testando i propri controller SDN in un ambiente di laboratorio virtuale.
- **Facilità di Utilizzo**: Mininet è progettato per essere facile da utilizzare e offre un'interfaccia intuitiva per la creazione e la gestione delle reti virtuali. Gli utenti possono utilizzare API Python o interagire direttamente dalla riga di comando per configurare e controllare le reti.

Le principali classi e funzioni che sono state utilizzate in python per la definizione di una topologia di rete sono:

- **Topo**: consente la definizione personalizzata delle topologie di rete virtuali
- **build()**: metodo sovrascritto che crea la topologia definita nella classe **Topo**
- **addSwitch()**: aggiunge uno switch alla topologia
- **addHost()**: aggiunge un host alla topologia
- **addLink()**: aggiunge un link bidirezionale alla topologia
- **Mininet**: classe principale utilizzata per creare, avviare e gestire simulazioni di reti virtuali.
- **start()**: funzione della classe Mininet che inizializza la rete
- **pingAll()**: funzione che testa la connettivita tra i vari nodi
- **stop()**: funzione che interrompe l'esecuzione della simulazione di rete.

# 2. NetFlow

NetFlow è un protocollo di monitoraggio del traffico di rete sviluppato da Cisco Systems, ampiamente utilizzato per analizzare e raccogliere informazioni dettagliate sul traffico di rete in tempo reale. Il protocollo consente ai dispositivi di rete, come router e switch, di raccogliere dati sul traffico di rete, tra cui informazioni sulle sorgenti, le destinazioni, i tipi di protocollo e i volumi di traffico.

Le principali caratteristiche e funzionalità di NetFlow includono la raccolta dei dati di traffico, che consente agli amministratori di monitorare l'utilizzo della larghezza di banda, identificare i flussi di dati più significativi e individuare eventuali anomalie o problemi di congestione di rete. Utilizzando i dati raccolti da NetFlow, è possibile condurre analisi approfondite sul comportamento del traffico di rete, identificare i modelli di utilizzo della rete e valutare le prestazioni complessive della rete. Inoltre, NetFlow fornisce agli amministratori di rete una panoramica completa delle attività di rete, consentendo loro di prendere decisioni informate sulla gestione e l'ottimizzazione delle risorse di rete. Una soluzione basata su NetFlow coinvolge tipicamente tre entità principali:

- **Flow Exporter**: dispositivi di rete (router o switch) che generano i record NetFlow per il traffico che attraversa il dispositivo.
- **Flow Collector**: un server o un dispositivo dedicato che riceve, elabora e archivia i record NetFlow per l'analisi e il monitoraggio.
- **Flow Analyzer**: applicazioni o strumenti di gestione e monitoraggio di rete che ricevono i dati NetFlow dal collector e li utilizzano per l'analisi, la generazione di report e il monitoraggio delle prestazioni di rete.

# 3. Implementazione e Analisi della Rete Simulata con Mininet e NetFlow

## 3.1 Setup del sistema

Per condurre il lavoro, ho utilizzato due macchine virtuali basate su Ubuntu 22.04 come ambiente di sviluppo. La prima macchina virtuale **Mininet** è stata utilizzata per la simulazione della rete di host ed è configurata con tre interfacce di rete. Una di queste è collegata alla scheda di rete del PC host tramite NAT, mentre le altre due sono connesse alla rete interna *"monitoring"*, dove sarà successivamente collegata anche la seconda macchina virtuale **Monitoring**.

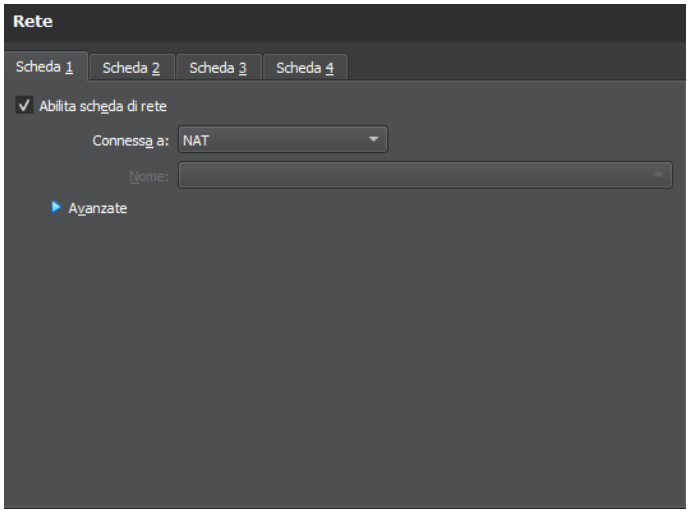



Figura 3.1: Scheda di rete con NAT (enp0s3)

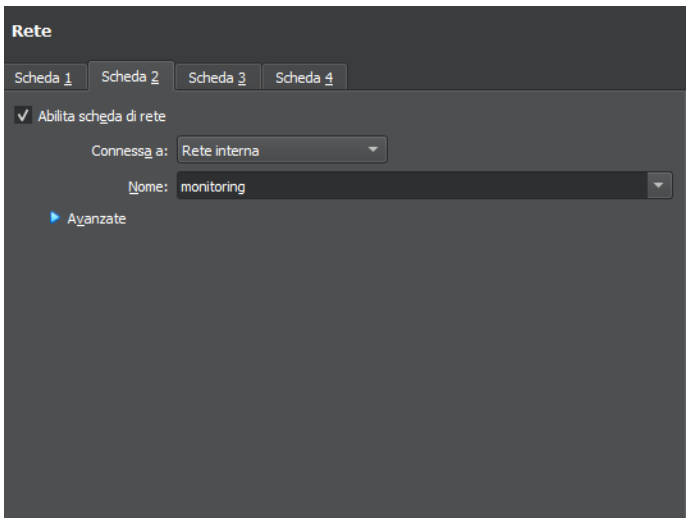



Figura 3.2: Scheda di rete interna (enp0s8)

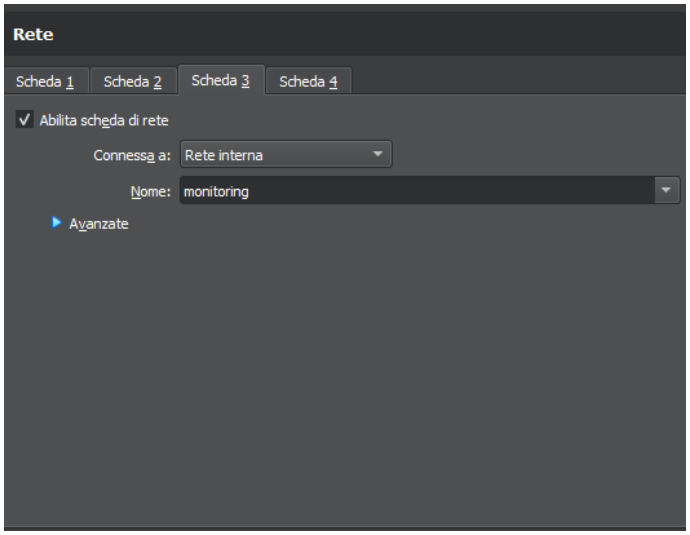



Figura 3.3: Scheda di rete interna (enp0s9)

## 3.2 Creazione della rete con Mininet

All'interno della prima macchina virtuale è stato installato Mininet tramite i seguenti comandi:

```
$ sudo apt-get update
$ git clone https://github.com/mininet/mininet #get source code
$ git checkout -b mininet-2.3.0 2.3.0 #choose version of mininet
$ mininet/util/install.sh -a #install all dependencies
```

Dopo aver installato Mininet, è stato creato uno script Python personalizzato per configurare la topologia di rete desiderata. Questo script è stato utilizzato per definire e istanziare gli elementi della rete, come nodi, switch e collegamenti.

```
from mininet.topo import Topo
from mininet.net import Mininet
from mininet.nodelib import NAT
from mininet.node import Node
from mininet.log import setLogLevel
from mininet.cli import CLI
from mininet.link import Intf

class Router(Node):
   def config(self, **params):
      super(Router, self).config(**params)
      self.cmd("sysctl net.ipv4.ip_forward=1") #comando per abilitare ip
         forward

   def terminate(self):
      self.cmd("sysctl net.ipv4.ip_forward=0") #comando per disabilitare
         ip forward
      super(Router, self).terminate()

#Creazione topologia di rete personalizzata
class MyTopologia(Topo):
   def build(self, **_kwargs):
      r0 = self.addNode("r0", cls=Router) #Aggiunta del router
      switch1 = self.addSwitch("s1")#Aggiunta del primo switch
      #Aggiunta di 3 host e assegnazione ip nella prima subnet con
         indirizzo 10.0.0.0/24
      h1 = self.addHost("h1", ip="10.0.0.10/24", defaultRoute="via
         10.0.0.1")
      h2 = self.addHost("h2", ip="10.0.0.11/24", defaultRoute="via
         10.0.0.1")
      h3 = self.addHost("h3", ip="10.0.0.12/24", defaultRoute="via
         10.0.0.1")
      #Creazione collegamento host-switch
      self.addLink(h1, switch1)
      self.addLink(h2, switch1)
      self.addLink(h3, switch1)

      #Aggiunta di 3 host e assegnazione ip nella seconda subnet con
         indirizzo 192.168.0.0/24
      switch2 = self.addSwitch("s2")#Aggiunta del secondo switch
      h4 = self.addHost("h4", ip="192.168.0.13/24", defaultRoute="via
         192.168.0.1")
      h5 = self.addHost("h5", ip="192.168.0.14/24", defaultRoute="via
```

```
            192.168.0.1")
36      h6 = self.addHost("h6", ip="192.168.0.15/24", defaultRoute="via
            192.168.0.1")
37      #Creazione collegamento host-switch
38      self.addLink(h4, switch2)
39      self.addLink(h5, switch2)
40      self.addLink(h6, switch2)
41
42      #Collegamento tra il router e i due switch attraverso interfaccia
43      self.addLink(r0, switch1, intfName1="r0-eth1", params1={"ip":
            "10.0.0.1/24"})
44      self.addLink(r0, switch2, intfName1="r0-eth2", params1={"ip":
            "192.168.0.1/24"})
45
46  def run():
47      topo = MyTopologia() #istanziamento topologia di rete personalizzata
48      net = Mininet(topo=topo, waitConnected=True)
49      net.start()
50      router = net.getNodeByName("r0")
51      _intf0 = Intf("enp0s3", node=router) #interfaccia enp0s3 configurata
            come router
52      router.cmd('dhclient enp0s3') #assegna tramite dhcp indirizzo ip
53      router.cmd("ifconfig r0-eth1 10.0.0.1 netmask 255.255.255.0")
            #configurazione ip per interfaccia r0-eth1
54      router.cmd("ifconfig r0-eth2 192.168.0.1 netmask 255.255.255.0")
            #configurazione ip per interfaccia r0-eth2
55      router.cmd("iptables -t nat -A POSTROUTING -o enp0s3 -j MASQUERADE")
            #regola per il NAT
56
57      CLI(net)
58      net.stop()
59
60
61  if __name__ == "__main__":
62      setLogLevel("info")
63      run()
```

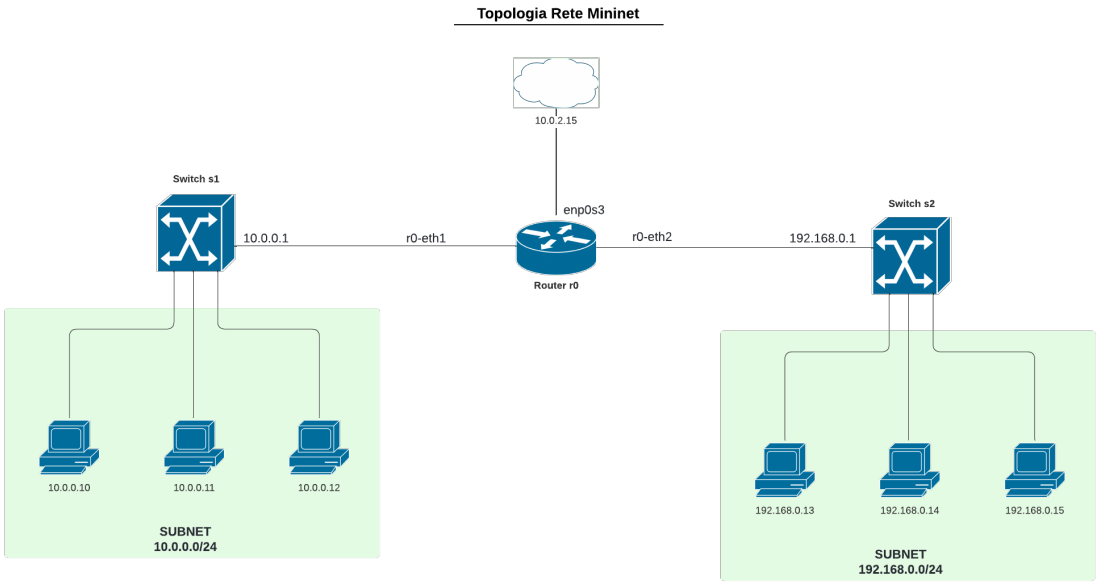



Figura 3.4: Topologia della rete Mininet

## 3.3 Applicativi Software Utilizzati

### 3.3.1 nProbe e ntopng

I software applicativi impiegati per l'analisi e la visualizzazione dei dati NetFlow sono **nProbe** e **ntopng**, due strumenti progettati per offrire una panoramica dettagliata e intuitiva sul traffico di rete.
nProbe è un software NetFlow v5/v9/IPFIX che permette di raccogliere, analizzare ed esportare rapporti sul traffico di rete utilizzando il formato standard Cisco NetFlow v5/v9/IPFIX ed è compatibile con la maggior parte dei sistemi operativi presenti sul mercato.

Invece, ntopng è un'applicazione di monitoraggio del traffico basata sul web con una vasta gamma di funzionalità, tra cui:

- **Monitoraggio passivo del traffico**: ntopng cattura passivamente il traffico di rete per fornire una visione dettagliata delle attività di rete in corso.
- **Raccolta dei flussi di rete**: Supporta la raccolta di flussi di rete utilizzando protocolli come NetFlow, sFlow e IPFIX, consentendo di analizzare e visualizzare i flussi di traffico in entrata e in uscita.
- **Monitoraggio attivo dei dispositivi di rete**: ntopng può monitorare attivamente dispositivi di rete specifici per raccogliere informazioni dettagliate sulle loro prestazioni e sul traffico associato.
- **Monitoraggio dell'infrastruttura di rete tramite SNMP**: Supporta il monitoraggio dell'infrastruttura di rete utilizzando il protocollo SNMP, con-

sentendo di raccogliere e visualizzare informazioni dettagliate su dispositivi di rete come router, switch e firewall.

Con l'integrazione di nProbe e ntopng, è possibile ottenere una visione completa e dettagliata del traffico di rete, fornendo agli amministratori di rete gli strumenti necessari per monitorare e gestire efficacemente le prestazioni e la sicurezza della rete. Queste due potenti soluzioni possono essere utilizzate insieme in diversi scenari per massimizzare l'efficienza e l'efficacia del monitoraggio del traffico di rete.

In particolare, l'utilizzo di nProbe con ntopng offre vantaggi significativi in diverse situazioni:

- **Visualizzazione dei dati NetFlow/sFlow**: nProbe può raccogliere e analizzare il traffico NetFlow/sFlow proveniente da router, switch e altri dispositivi di rete, e inviare i risultati a ntopng per una visualizzazione dettagliata e intuitiva dei flussi di traffico.
- **Monitoraggio delle interfacce di rete remote**: Utilizzando nProbe su sistemi remoti, è possibile catturare il traffico delle interfacce di rete e inviare i flussi risultanti a un'istanza centrale di ntopng per l'analisi e la visualizzazione. Questo approccio consente di monitorare in modo efficiente le prestazioni e la sicurezza delle reti distribuite.

## 3.4 Installazione di nProbe e ntopng

Per quanto riguarda la seconda macchina virtuale **Monitoring** è stata configurata una scheda di rete connessa su rete interna *monitoring* utilizzata per la comunicazione tra gli agenti NetFlow che compongono la rete virtuale creata con Mininet ed eventuali applicazioni di monitoraggio.

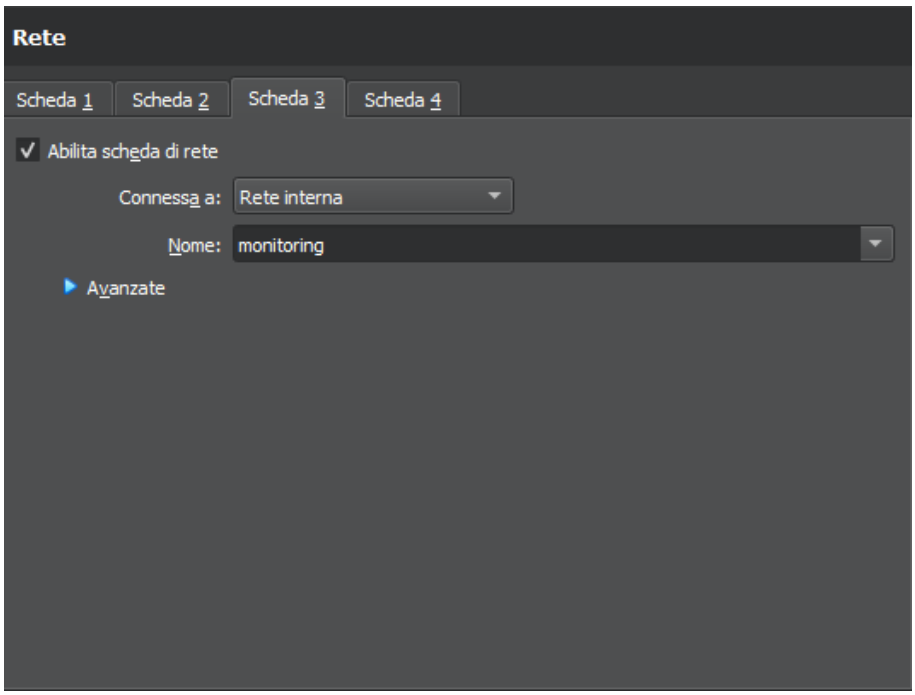



Figura 3.5: Scheda di rete interna Monitoring (enp0s9)

L'installazione e la configurazione dei due applicativi è stata fatta attraverso i seguenti comandi:

```
#Installazione di nProbe e ntopng
$ sudo apt-get install software-properties-common wget
$ sudo add-apt-repository universe
$ wget https://packages.ntop.org/apt-stable/22.04/all/apt-ntop-stable.deb
$ sudo apt install ./apt-ntop-stable.deb
$ sudo apt-get clean all apt-get update
$ sudo apt-get install pfring-dkms nprobe ntopng n2disk cento
```

Successivamente sono stati configurati i due file di configurazione **nprobe.conf** e **ntopng.conf** nel seguente modo:

```
#Configurazione nprobe.conf
 --zmq "tcp://127.0.0.1:5556"
 -i none
 -n none
 --collector-port 2055
 -T "@NTOPNG@"

 #Configurazione ntopng.conf
 -i tcp://127.0.0.1:5556
```

L'opzione -T "@NTOPNG@", indica a nprobe l'insieme minimo di campi da esportare per garantire l'interoperabilità con ntopng. Questa opzione è consigliata quando si usa nProbe con ntopng.
Per quanto riguarda ZMQ, essa è una libreria di messaggistica asincrona ad alta velocità, che fornisce un mezzo per lo scambio efficiente di dati tra applicazioni distribuite e processi utilizzato in nProbe e ntopng per consentire la comunicazione e lo scambio di dati tra i due componenti. Ad esempio, nProbe utilizza ZMQ per inviare i flussi NetFlow raccolti a ntopng per l'elaborazione e la visualizzazione.

Gli agenti NetFlow inviano regolarmente i dati di telemetria raccolti sulla porta 2055. nProbe è responsabile di inviare il traffico raccolto a ntopng, che offre un'interfaccia grafica web per visualizzare i dati di traffico. Per accedere a questa interfaccia web direttamente dalla macchina virtuale è sufficiente accedere via browser all'indirizzo 127.0.0.1:3000. La prima pagina che verrà visualizzata è quella di autenticazione, da cui è possibile accedere alle varie funzionalità di ntopng.

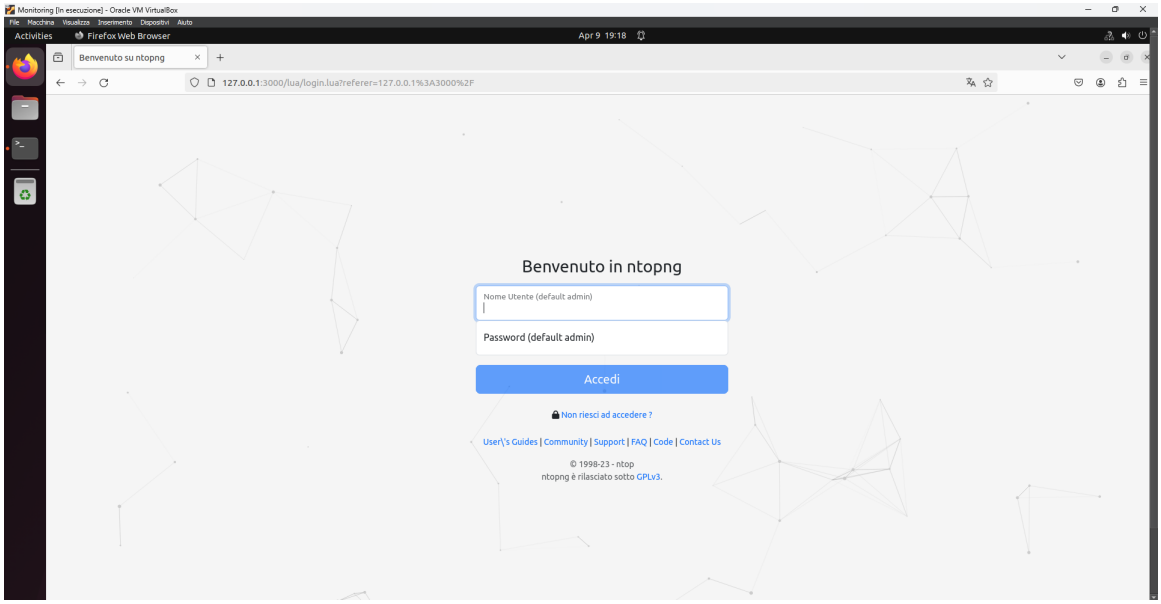



Figura 3.6: login in ntopng

Una volta effettuato l'accesso troviamo un'interfaccia utente per analizzare e monitorare il traffico.

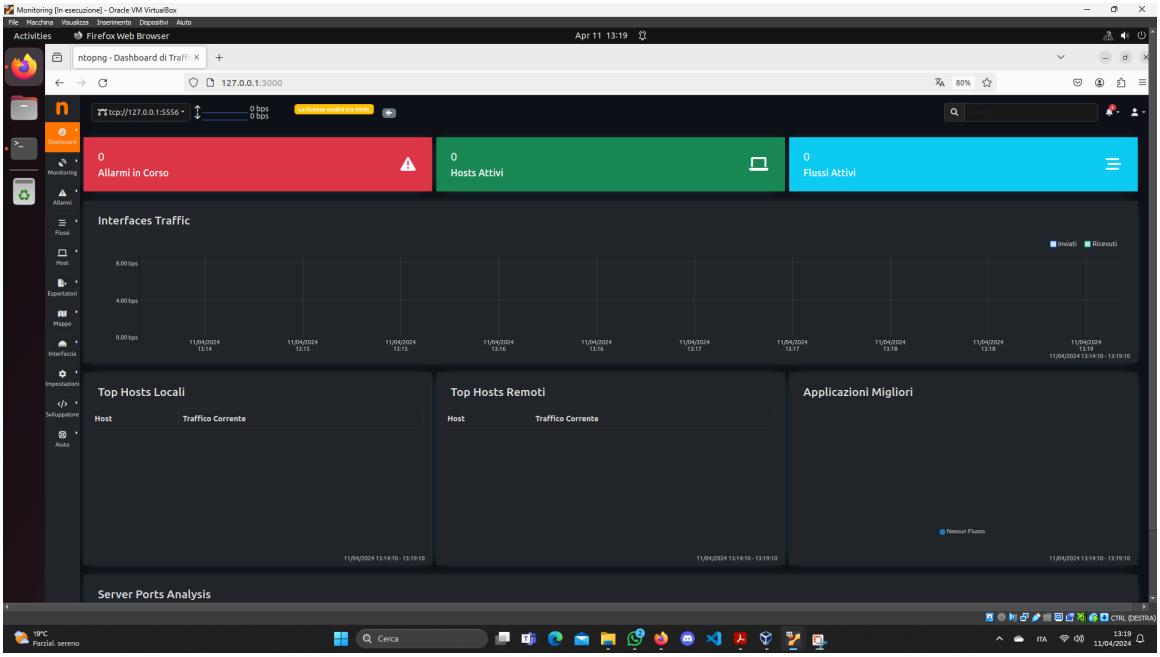



Figura 3.7: Interfaccia utente di ntopng

## 3.5 Monitoraggio della rete Mininet con NetFlow

Per poter monitorare la rete Mininet attraverso nprobe e ntopng è stato abilitato NetFlow nei due switch s1 e s2 che compongono la rete. Per farlo abbiamo utilizzato il comando **xterm** che permette di aprire i terminali dei singoli host all'interno di mininet. Per abilitare NetFlow nei due switch sono stati usati i seguenti comandi:

```
#switch s1
$ sudo ovs-vsctl -- --id=@nf create netflow target="192.168.1.2:2055"
    active-timeout=60 -- -- set Bridge s1 netflow=@nf
#switch s2
$ sudo ovs-vsctl -- --id=@nf create netflow target="192.168.1.2:2055"
    active-timeout=60 -- -- set Bridge s2 netflow=@nf
```

I parametri hanno i seguenti significati:

- **target**: Rappresenta l'indirizzo IP del terminale di monitoraggio e la porta in cui esso è in ascolto.
- **active-timeout**: Questo parametro specifica il tempo prima che un flusso inattivo venga considerato chiuso, impostato su 60 secondi.

A questo punto sono stati fatti partire gli applicativi nprobe e ntopng utilizzando i comandi:

```
$ sudo service nprobe start
$ sudo service ntopng start
```

Sono stati eseguiti una serie di test utilizzando il comando ping. Abbiamo testato la connettività sia all'interno della rete Mininet sia tra gli host della stessa rete, sia con host di reti diverse. Inoltre, abbiamo eseguito il ping verso 8.8.8.8 per valutare la connettività esterna.

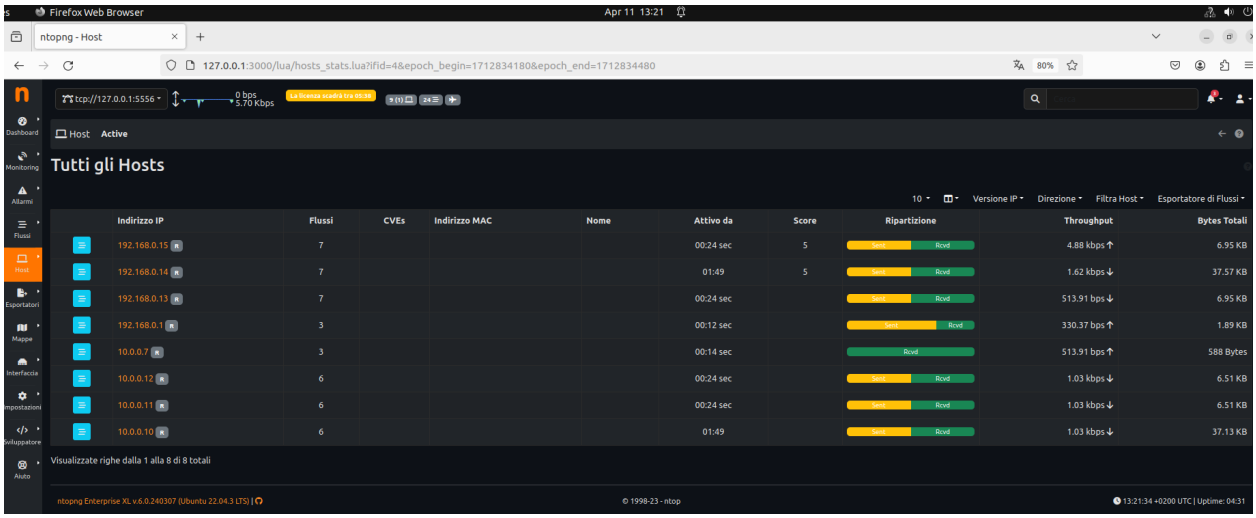



Figura 3.8: Host attivi.

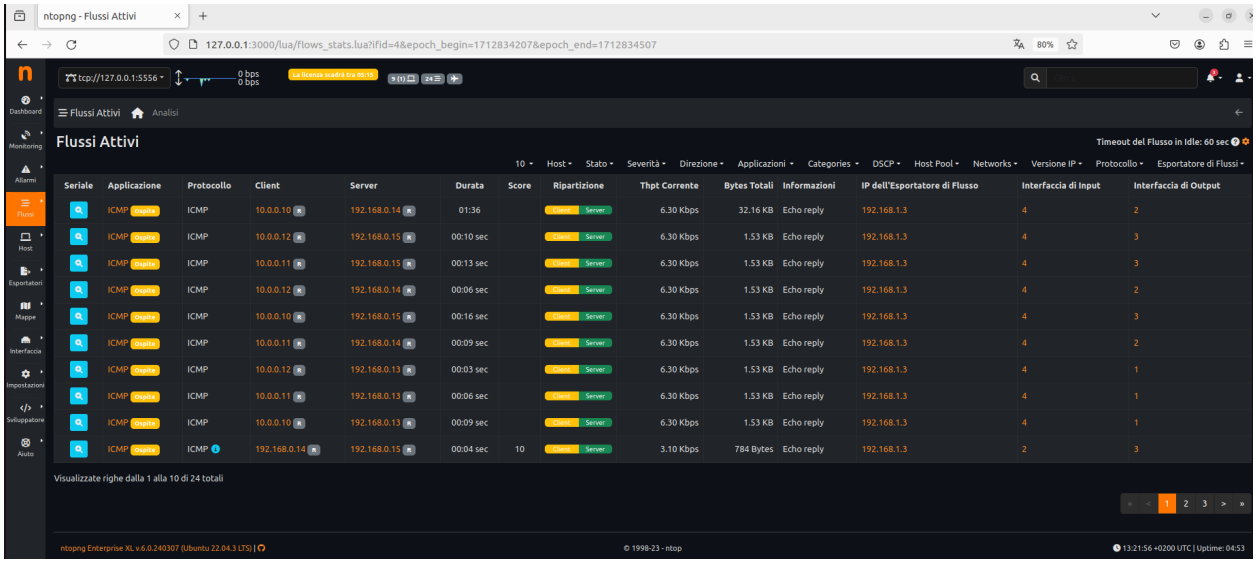
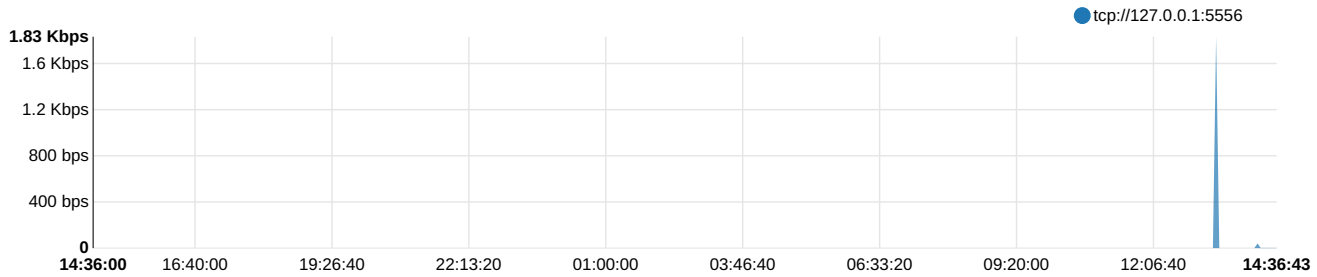



Figura 3.9: Flussi attivi.

Inoltre, ntopng offre la possibilità di generare report dettagliati sul traffico di rete e scaricarli in formato PDF. Questi report consentono di analizzare il traffico di rete nel corso del tempo, potendo selezionare l'arco temporale desiderato per il report.

# Report di Traffico: 1 Day Da Wed Apr 10 14:36:00 2024

## Interfacce di Rete

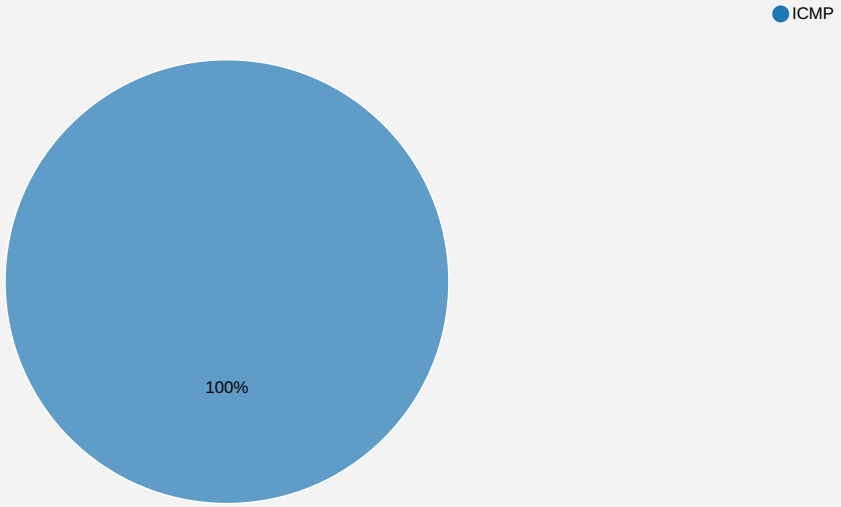



# Interfaccia di Rete tcp://127.0.0.1:5556

## Applicazioni

## Traffico Totale: 49.93 KB


## Breakdown Applicazione




| Top Non-Local OS [Traffico/sec medio] | | | |
|---|---|---|---|
| **Mittenti** | | **Destinatari** | |
| Unknown | 53.9 KB (5.12 bps) | Unknown | 53.9 KB (5.12 bps) |

| Top ASN [Traffico/sec medio] | | | |
|---|---|---|---|
| **Mittenti** | | **Destinatari** | |
| | | Google LLC | 98 Bytes (0.01 bps) |

| Top Paesi [Traffico/sec medio] | | | |
|---|---|---|---|
| **Mittenti** | | **Destinatari** | |
| | | US | 98 Bytes (0.01 bps) |

| Top Hosts Remoti [Traffico/sec medio] | | | |
|---|---|---|---|
| **Mittenti** | | **Destinatari** | |
| 10.0.0.10 | 19.2 KB (1.82 bps) | 10.0.0.10 | 19.1 KB (1.81 bps) |
| 192.168.0.14 | 18.8 KB (1.78 bps) | 192.168.0.14 | 18.8 KB (1.78 bps) |
| 10.0.0.12 | 3.9 KB (0.37 bps) | 10.0.0.12 | 3.9 KB (0.37 bps) |
| 192.168.0.15 | 3.4 KB (0.33 bps) | 192.168.0.15 | 3.5 KB (0.33 bps) |
| 10.0.0.11 | 3.3 KB (0.31 bps) | 10.0.0.11 | 3.3 KB (0.31 bps) |
| Other | 1.3 KB (0.12 bps) | Other | 588 Bytes (0.05 bps) |
| 10.0.0.1 | 588 Bytes (0.05 bps) | 8.8.8.8 | 98 Bytes (0.01 bps) |