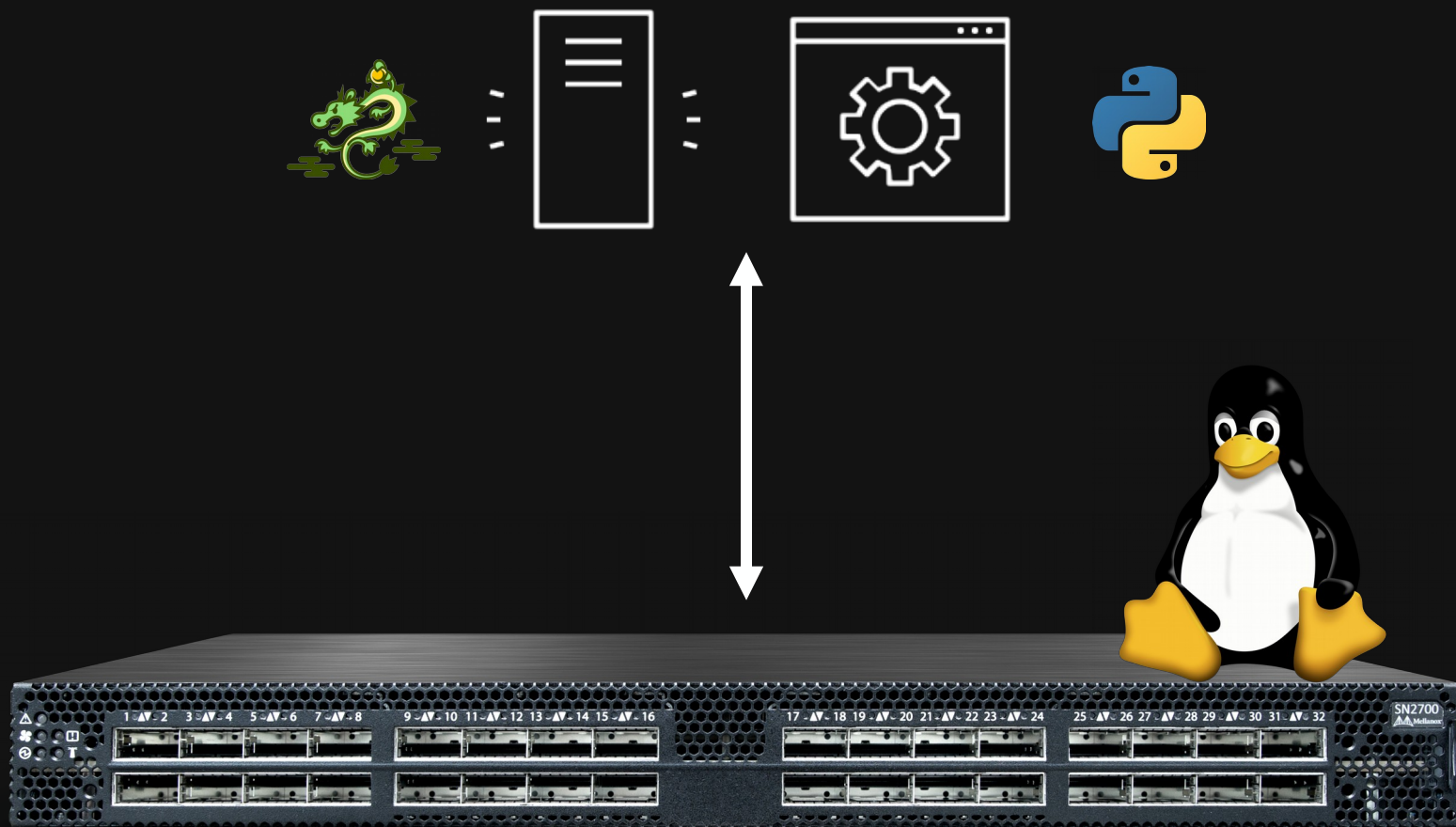


Software Defined Networking

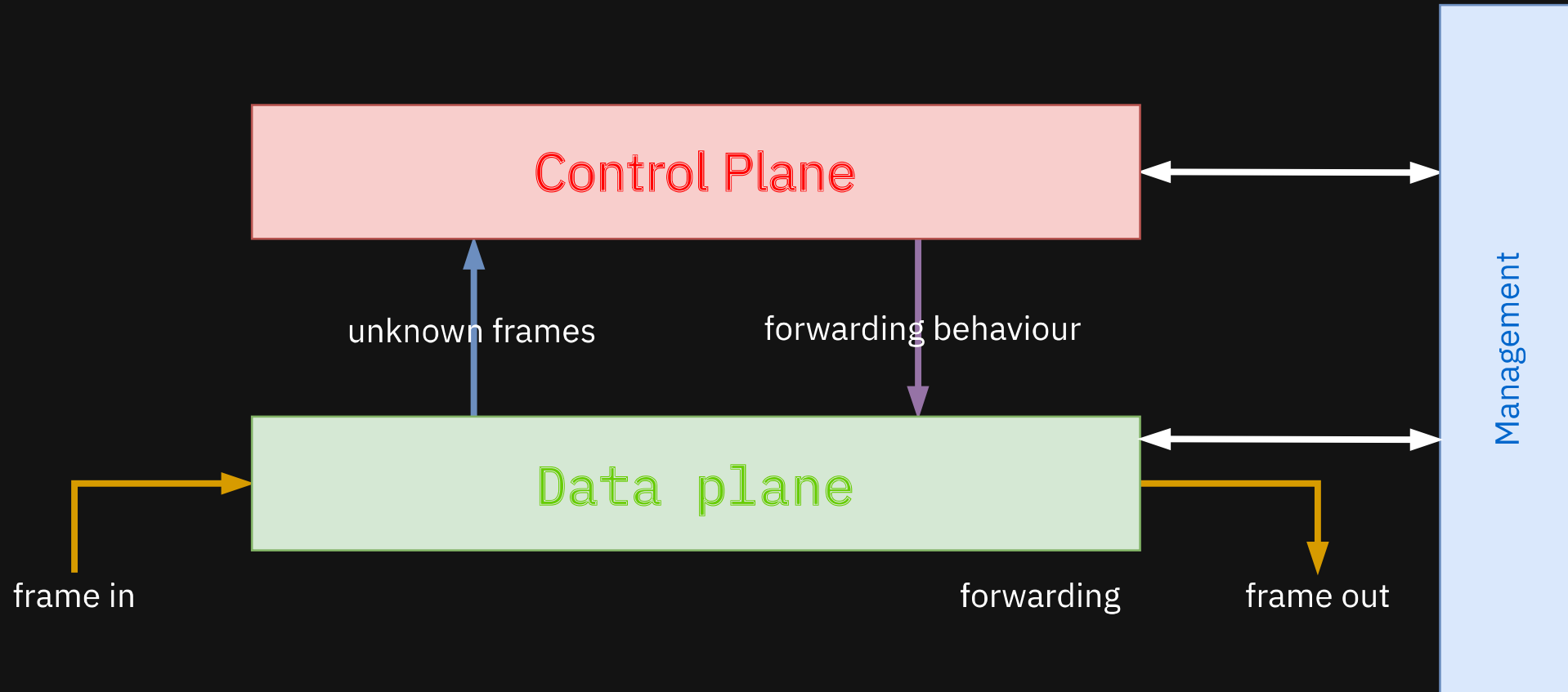
Giulio Corradini, Alessandro Maiani
SSH 2021

Software defined networking

Separazione del piano di forwarding dal piano di controllo



Separazione dei piani



Piano di controllo - protocolli, algoritmi, routing

Data plane - responsabile dell'inoltro fisico dei frame

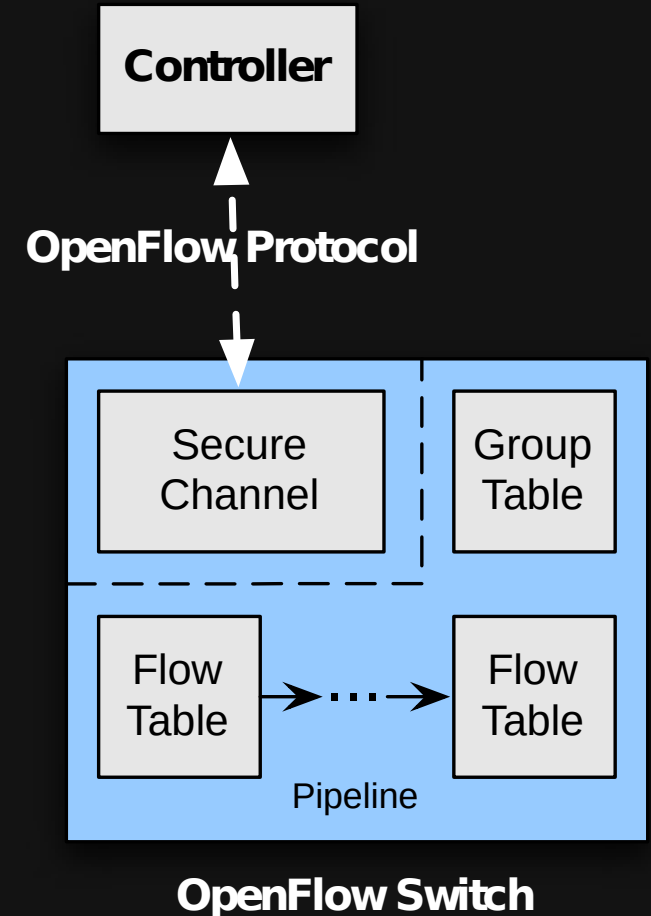
Casi d'uso

- Datacenter
- Metropolitan area networks
- Applicazioni video (YouTube)
- Controllo fine del traffico

Applicazioni ad elevato utilizzo di banda, che i network legacy non riescono a fornire.

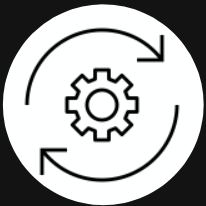
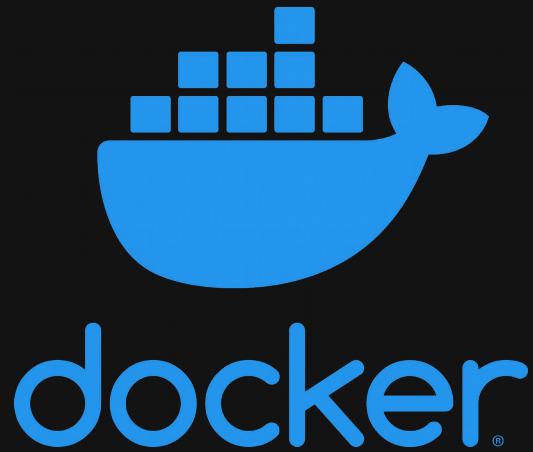
OpenFlow

- Protocollo di comunicazione tra switch e controller
- Accesso al piano di forwarding
- Connessione sicura TCP + TLS su porta 6653
- Standard aperto e interoperabile

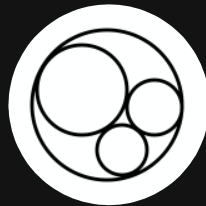


Docker

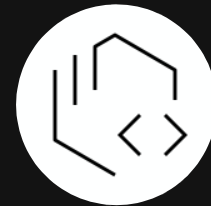
Virtualize everything!



Separazione dei processi

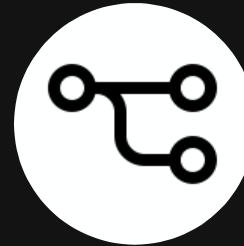
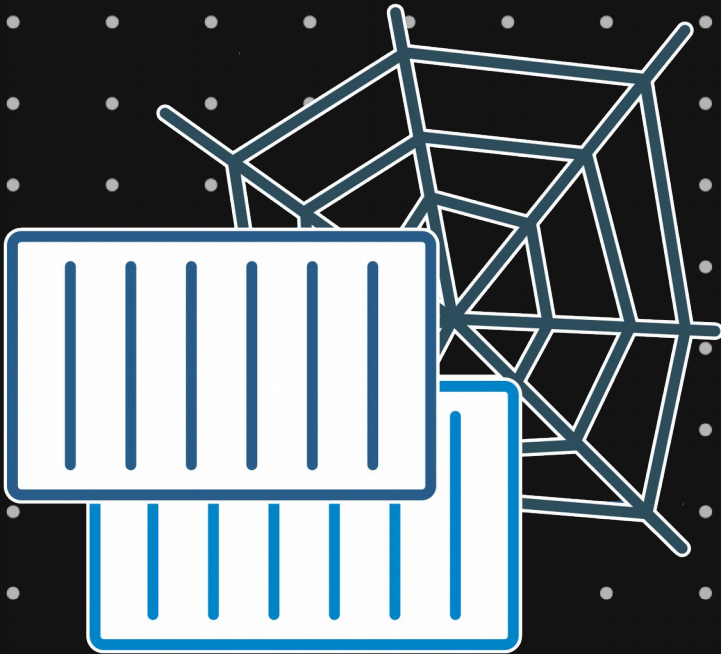


cgroups e namespaces

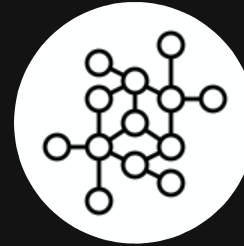


Codice, librerie, utility e
runtime

Containernet



Fork di mininet



Testbed
Simulatore di rete



Docker-based
Simulatore e host come container

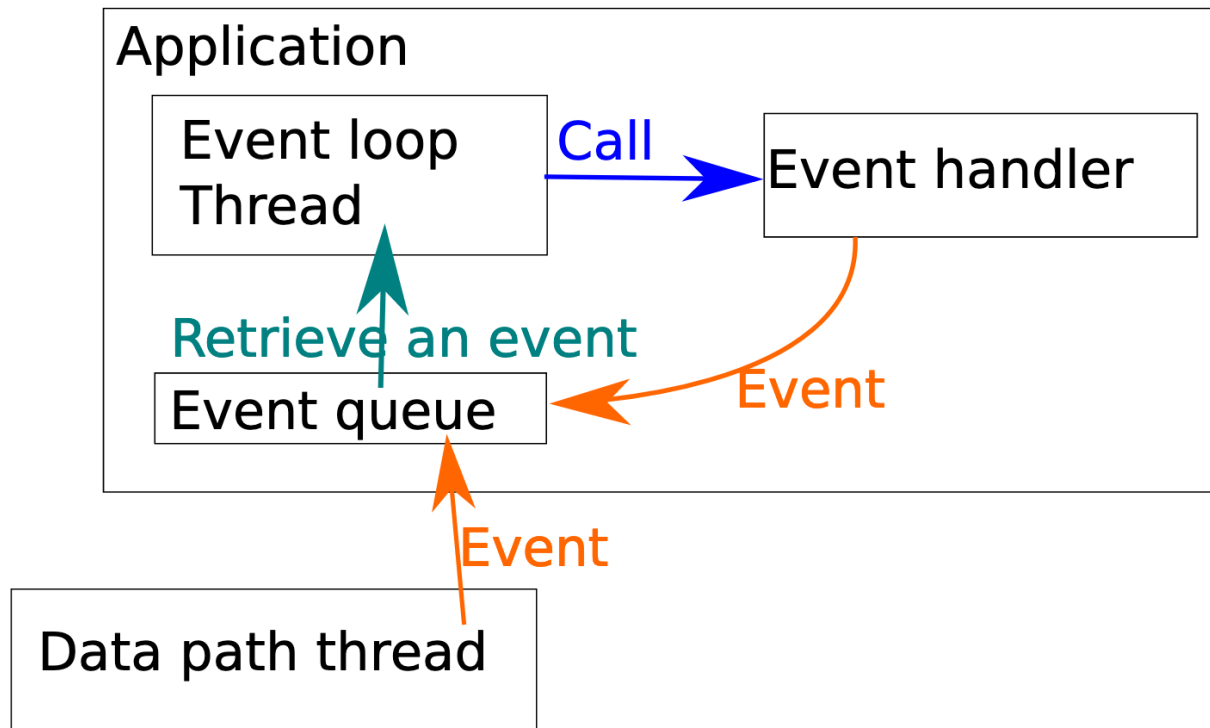
Ryu

- Modulo Python per scrivere applicazioni per SDN
- Incapsula e gestisce le connessioni OpenFlow
- Creato da Nippon Telegraph and Telephone Co.



Architettura di Ryu

ryu-manager process



L'utente definisce:

Sottoclassi di RyuApp

Handler e callback

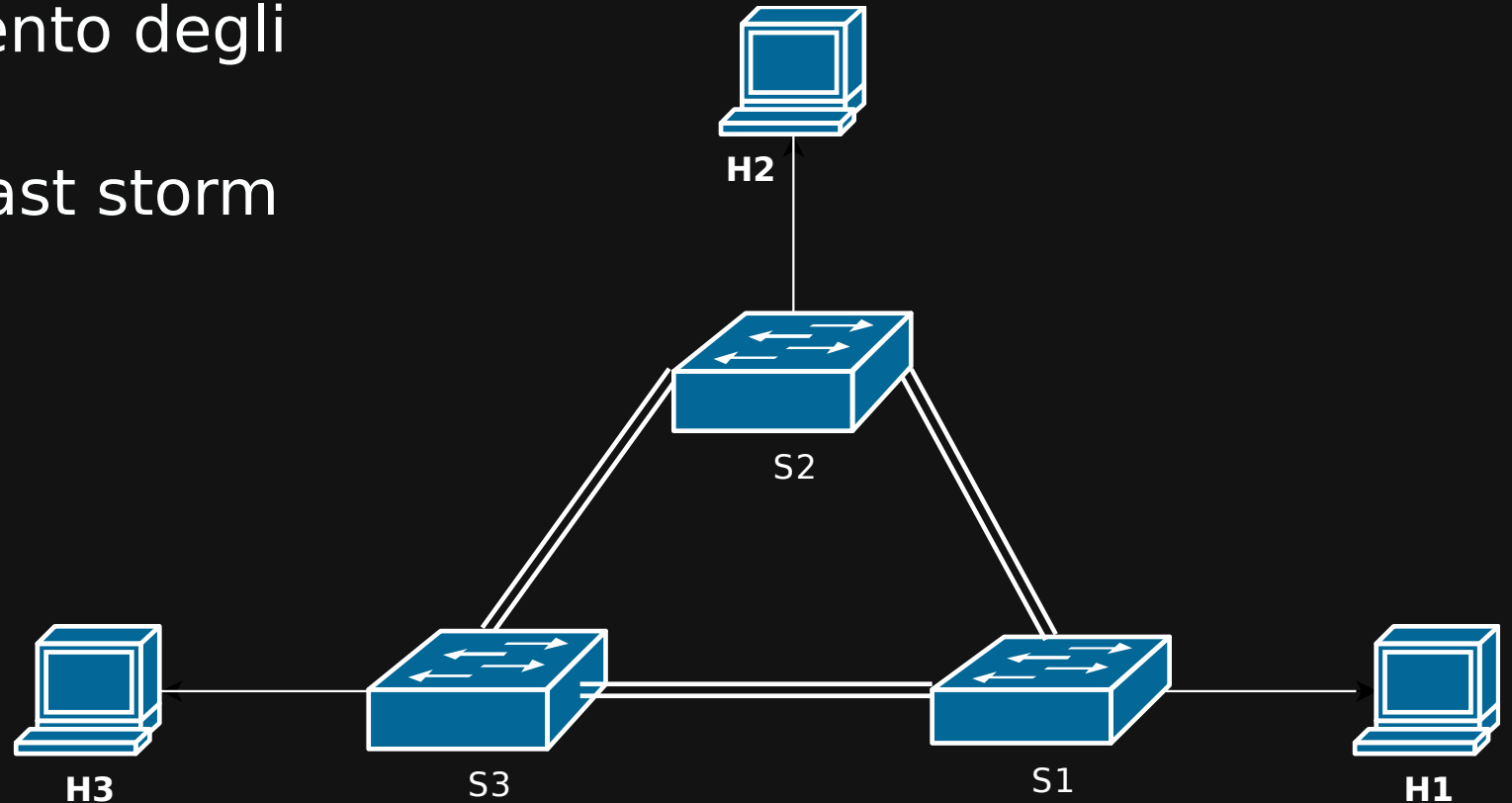
Eventi custom

Triangolare

Non c'è un controller

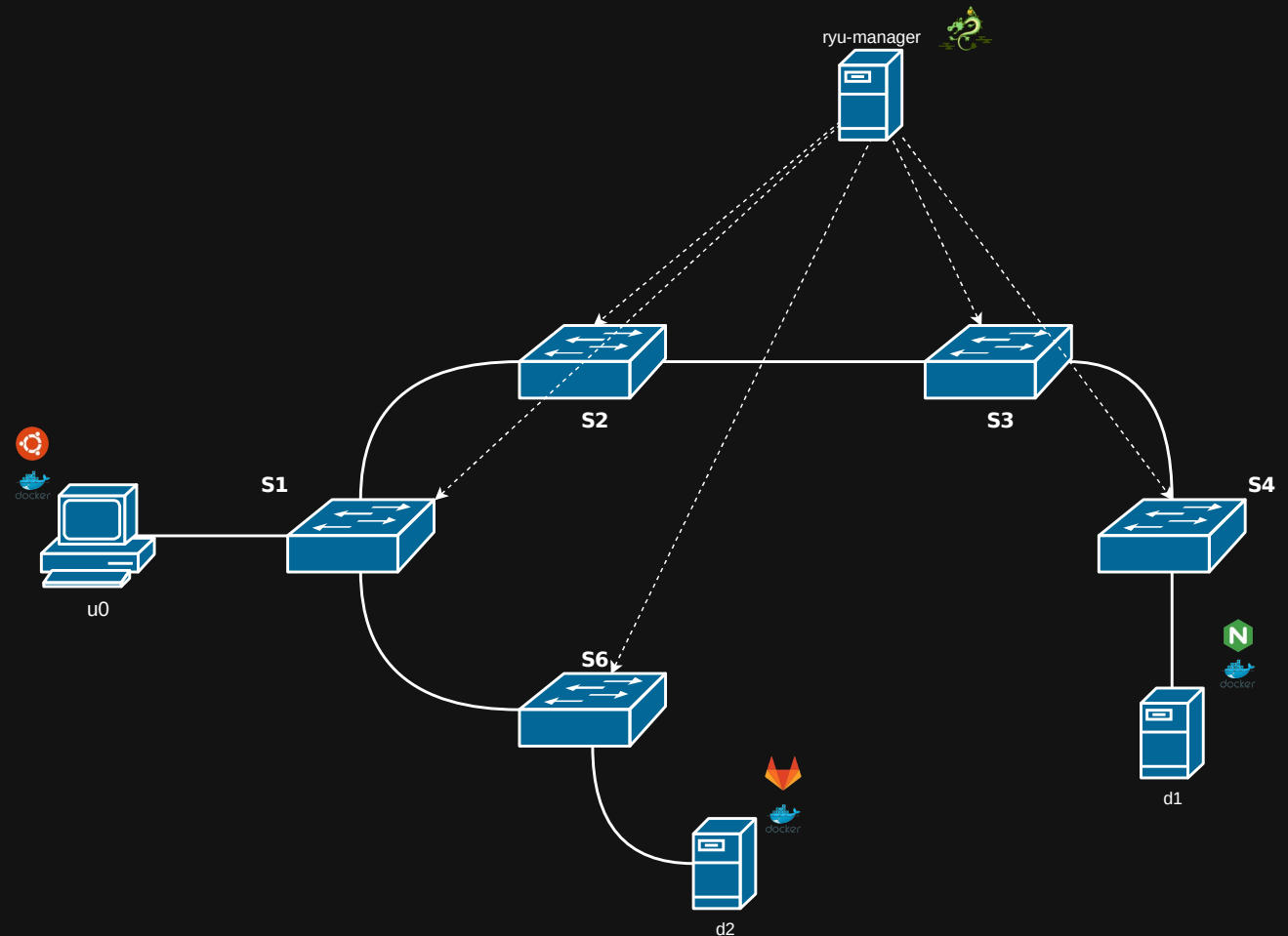
Classico comportamento degli switch

Problema del broadcast storm



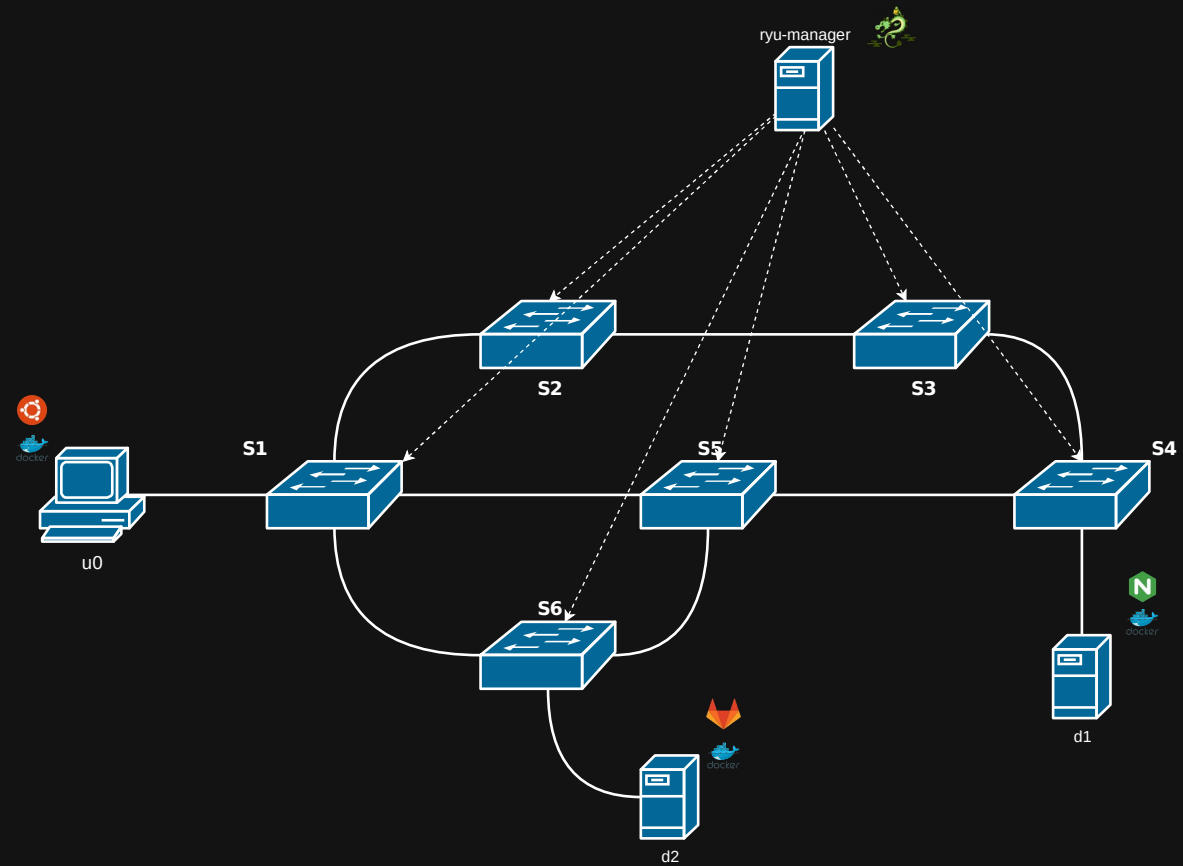
Topology 1 «no_loop»

- 1 host
- 5 switch
- 2 server (d1,d2)
- Controller
- Niente loop



Topologia completa

- 1 host
- 6 switch
- 2 server (d1,d2)
- Controller

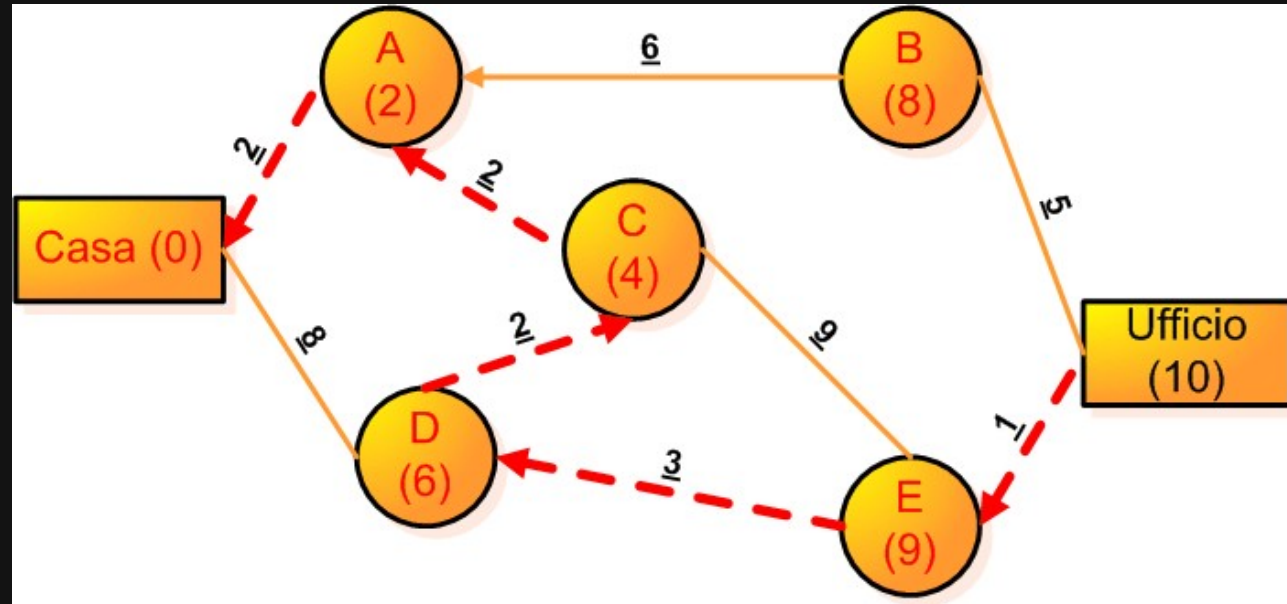


OpenFlow Discovery Protocol + host

- Sfrutta i pacchetti LLDP per trovare gli switch e i link sulla rete.
- Nostra aggiunta: ARP storm alle porte attive non collegate ai link.
- Trovo gli host, la loro porta e gli indirizzi MAC e IP.

Dijkstra

Algoritmo per calcolo dei cammini minimi da un nodo sorgente verso ogni altro nodo.



Per ogni coppia di host trova la rotta più veloce.
L'applicazione analizza il cammino e produce per ogni switch presente, una flow.

I flussi broadcast vengono inviati al controller. Il protocollo ARP viene gestito da ARPResponder, gli altri flussi vengono bloccati

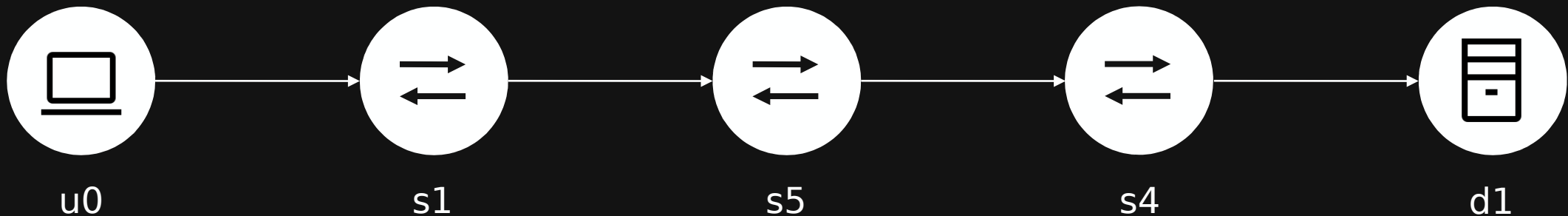
```
>_ sdn.controller.topology_solver
```

Esempio di path

Nodo sorgente: u0

```
{  
  ("82:3c:7a:5b:3f:62", "10.0.0.10"): [("82:3c:7a:5b:3f:62", "10.0.0.10")],  
  ("5e:4b:31:01:0f:bd", "10.0.0.252"): [  
    ("82:3c:7a:5b:3f:62", "10.0.0.10"), 1, 6, ("5e:4b:31:01:0f:bd", "10.0.0.252"),  
  ],  
  ("a8:e3:2e:e2:d0:69", "10.0.0.251"): [  
    ("82:3c:7a:5b:3f:62", "10.0.0.10"), 1, 5, 4, ("a8:e3:2e:e2:d0:69", "10.0.0.251"),  
  ],  
  ... # rotte switch  
}
```

u0 → d1



Metrics

Recupero delle statistiche di utilizzo e stima della congestione

Ogni 10 secondi il controller invia un PortStatsRequest.

Gli switch rispondono con le statistiche delle porte.

Il numero di byte inviati su una porta è proporzionale alla congestione del link.

```
>_ sdn.controller.metrics
```

Bibliografia

Documentazione

- <https://www.docker.com>
- <https://containernet.github.io>
- <https://ryu-sdn.org>
- <https://book.ryu-sdn.org/en/Ryubook.pdf>
- https://ryu.readthedocs.io/en/latest/ofproto_v1_3_ref.html
- <https://opennetworking.org/wp-content/uploads/2014/10/openflow-spec-v1.3.0.pdf>

Paper:

- A POX OpenFlow Loop Solution for Mininet Network Emulations:
https://www.researchgate.net/publication/276415129_A_POX_OpenFlow_Loop_Solution_for_Mininet_Network_Emulations
- Host discovery solution: <https://www.scitepress.org/papers/2016/59670/59670.pdf>