GARR

The Italian Academic & Research Network

www.garr.it

INFN Istituto Nazionale di Fisica Nucleare

OpenFlow GARR virtual test-bed

Presentazione dell'attività svolta

Luca Prete

Borsisti Day, Roma, 06.12.2012

Consortium GARR

Il lavoro di ricerca svolto su OF

- Borsista GARR 2011/2012
  - Scopo della borsa: studiare le potenzialità delle SDN, in particolare del protocollo OpenFlow

- Fasi di lavoro
  - Studio del protocollo e degli strumenti tecnologici
  - Sviluppo di un test-bed di rete virtuale installato localmente su hardware di GARR ospitato a INFN - Milano Bicocca
  - Sviluppo di tre casi d'uso
  - Raccolta dati e analisi dei risultati
  - Partecipazione a workshop e stesura articoli (EWSDN 2012)
  - Stesura documentazione e report trimestrali

AGENDA

- Software Defined Networks
- Il protocollo OpenFlow
  - Architettura
  - Switch e controller
- Il test-bed virtuale: studio delle potenzialità di OpenFlow
  - Struttura del test-bed
  - Funzionalità e casi d'uso implementati
- Conclusioni e possibile utilità in GARR
- Domande

Software Defined Networks (SDN)

- Fino ad ora Internet basato su device configurabili
  - protocolli + ACL via CLI o API
- Modalità di configurazione e funzionalità dipendenti dai vendor e dai modelli
- Con l'evoluzione della tecnologia aumenta anche la complessità delle infrastrutture
  - Cloud computing e virtualizzazione:
    - Vantaggi ma la gestione dei DC diventa sempre più complessa
    - Necessità di maggiore automazione
  - Infrastrutture di rete datate
  - Bisogno di gestire la rete in modo più furbo e dinamico

Software Defined Networks (SDN)

- La rigida distinzione tra strato di rete e strato applicativo, tipica delle tradizionali reti a pacchetto, potrebbe essere superata
- Le applicazioni possono
  - accedere alle risorse hardware di rete
  - “parlare” con gli switch
  - ...e anche programmarli
- È possibile instradare il traffico in base allo stato dell’intera infrastruttura IT programmando adeguatamente il controller centralizzato
  - La logica è definita dal codice (open source) scritto dall’utente
- Uniformità per il piano di controllo (centralizzato!)
  - Apparati di rete di diversi vendor usano lo stesso protocollo

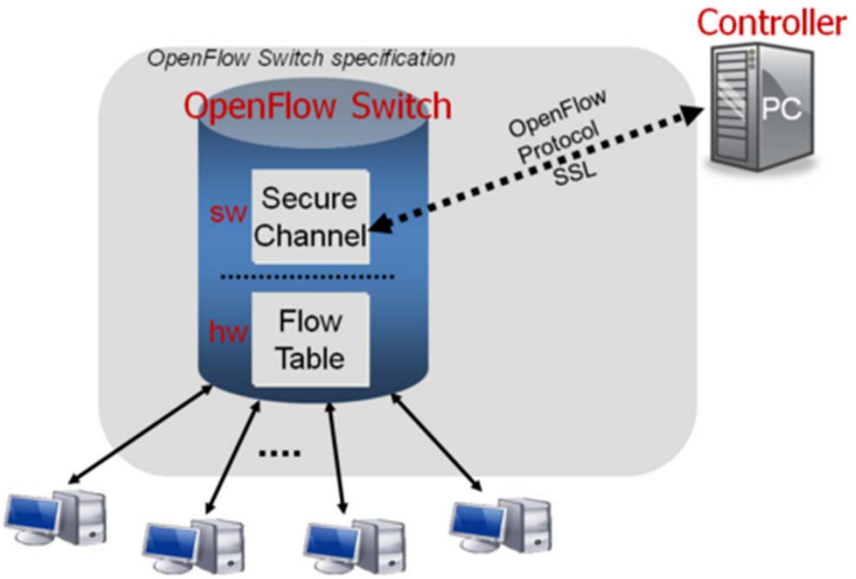
Un'implementazione di SDN - OpenFlow

- OpenFlow è proposto da Stanford nel 2007 come esempio di SDN
- Ad oggi è considerato dalla maggior parte dei produttori hardware e software uno standard de facto, un concorrente *per un singolo dominio* a MPLS

- Trattamento del traffico basato sul concetto di "flusso"
  - Un flusso è una sequenza di pacchetti identificabili da uno od un insieme di 'etichette' comuni (indirizzo IP, MAC address, porta,...)
  - Il nodo di rete fa solo forwarding

- Piano di controllo (esterno ai router/switch switch) disaccoppiato da quello di forwarding.
  - Un computer esterno intelligente + tanti switch/router "semplici"

Un'implementazione di SDN - OpenFlow



- Interfaccia standard verso i device di diversi produttori (HW e SW)
  - Tanti switch scambiano messaggi OpenFlow standard con un "controller" centralizzato attraverso un canale sicuro (TCP/SSL)
- Tabelle dei flussi: regole + azioni + statistiche
- Se un pacchetto non fa parte di un flusso noto, è inviato al controller che decide cosa fare
- I flussi scadono

Device OpenFlow enabled

- Diversi produttori credono e investono in OpenFlow
  - Già presenti apparati sul mercato che implementano nativamente il protocollo
  - Cisco, Juniper, HP, NEC, Dell, IBM, ...
  - Carot: Switch a basso costo OpenFlow 1.3 certified
    - Meno di 1000€ per 8 porte 1Gb
  - OpenWRT: firmware OpenFlow compilant per diversi apparati di categoria inferiore

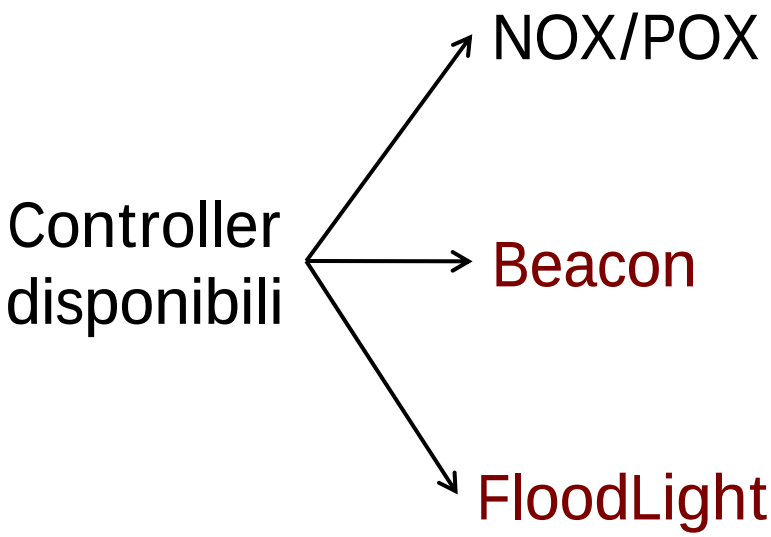
- Un'alternativa software: Open vSwitch
  - Funzionalità di livello enterprise
  - Usato anche in diversi hypervisor, es. Xen
  - Validità oltre la semplice simulazione

OpenFlow: presente e futuro

- Attualmente la versione maggiormente implementata è la 1.0
  - Ethernet
  - IPv4
  - TCP, UDP
  - Oltre a MPLS, VLAN, PBB, VXLAN (plugin)...

- A breve, versione 1.2 e 1.3
  - Controllo a layer fisico per controllare le WAN
    - Controllo lambda, circuiti ottici
  - Configurazione PBB (MAC-in-MAC)
  - IPv6
  - Configurazione L3 tunneling / MPLS / GRE
  - Drastici cambiamenti nel software lato controller per cambio di semantica

I controller OpenFlow



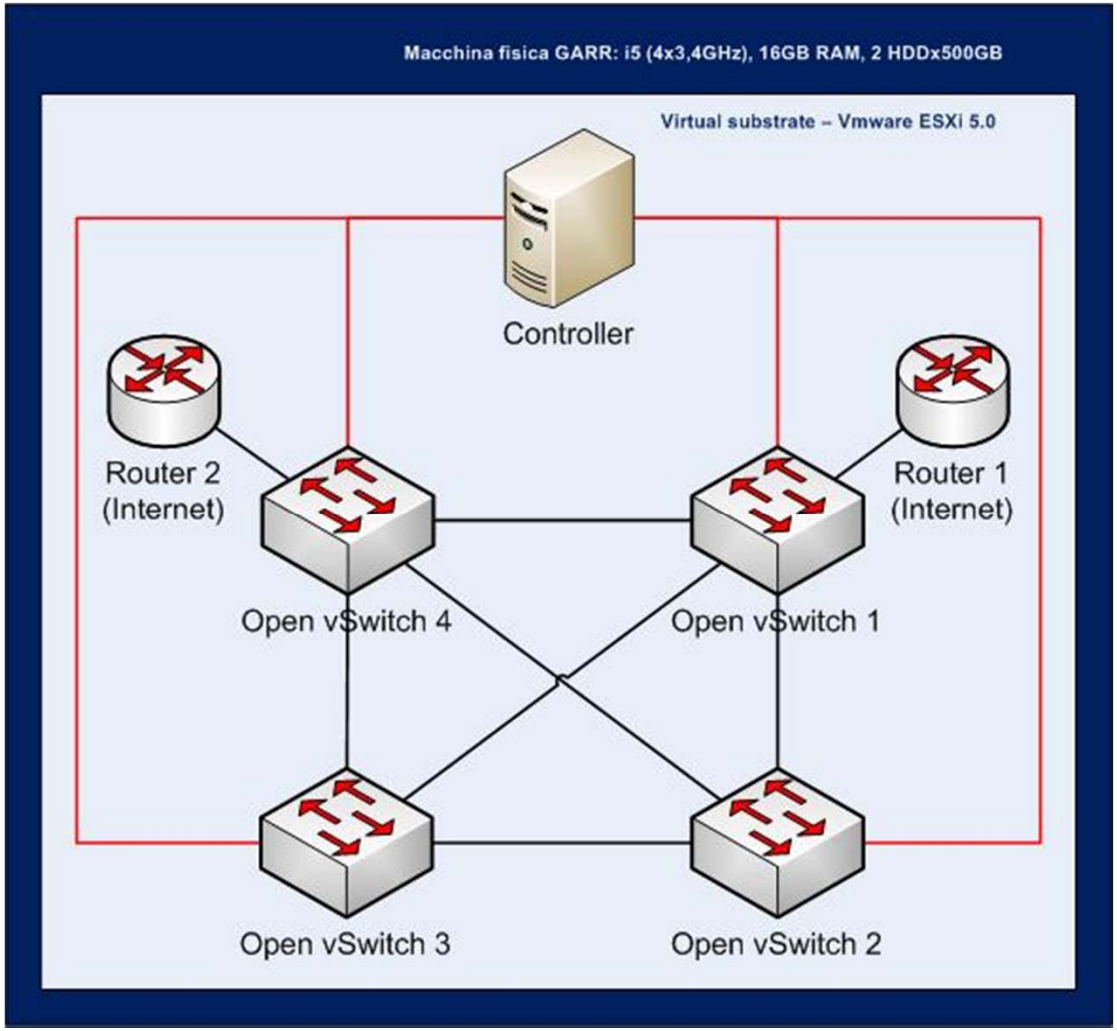
- Perché?
  - Java
    - Multi-piattaforma
    - Thread-oriented
  - Usati in test-bed di grandi dimensioni (+150 switch)
  - Modularità

Focus: Instradamento del traffico

- La logica con la quale sono instradati i pacchetti nella rete è implementata centralmente nel controller e quindi le regole inviate ai nodi rete

- I controller prevedono due metodi per l'inoltro dei pacchetti:
  - LearningSwitch: Emulazione di un switch L2
    - Le tabelle dei MAC sono memorizzate nel controller
    - Problema: non si possono gestire i loop nativamente (se non supportato dagli switch)
  - Routing:
    - Non è il routing che conosciamo, è la vera novità introdotta da OF!
    - Pacchetti instradati (a flussi) grazie ad una conoscenza globale della rete
    - Mappa è mantenuta aggiornata grazie a Device e Topology manager
    - Topology si occupa dei collegamenti switch – switch
    - Device memorizza i punti di connessione device – switch

Il test-bed virtuale



- Obiettivo: sperimentazione delle potenzialità offerte da OpenFlow

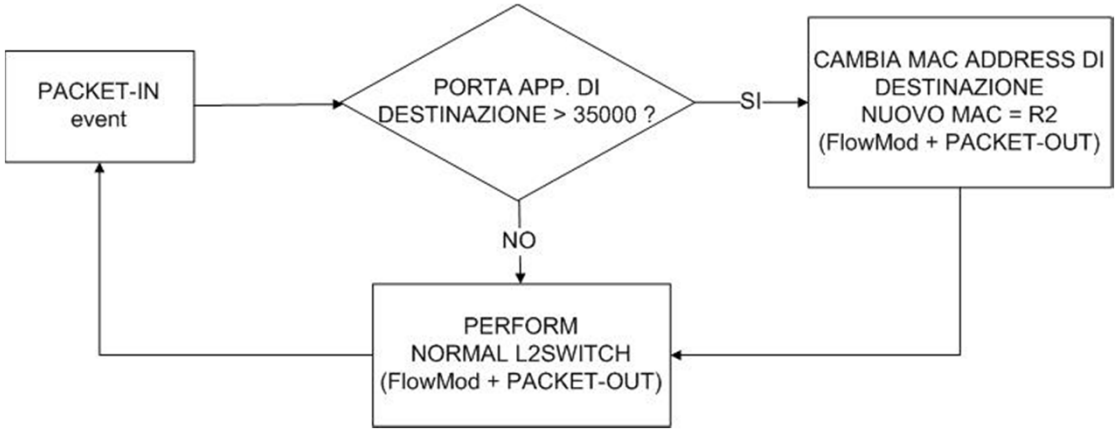
Use Case 1 - Instradamento del traffico in base all'applicazione

- Problema:
  - Fino ad ora costoso/complesso re-direzionare i pacchetti verso gateway diversi in base al loro contenuto.

- Obiettivo use case 1:
  - Re-direzionare il traffico di rete verso gateway differenti in base all'analisi del traffico
  - Se possibile, i pacchetti devono sempre seguire il percorso minimo per arrivare al router di destinazione

- Soluzione:
  - Discriminare il tipo di traffico attraverso un'analisi multi-layer
  - Filtrare i pacchetti in ingresso al controller prima che siano processati da Learning Switch e cambiare il MAC address di destinazione.

Test: traffico di “ricerca”



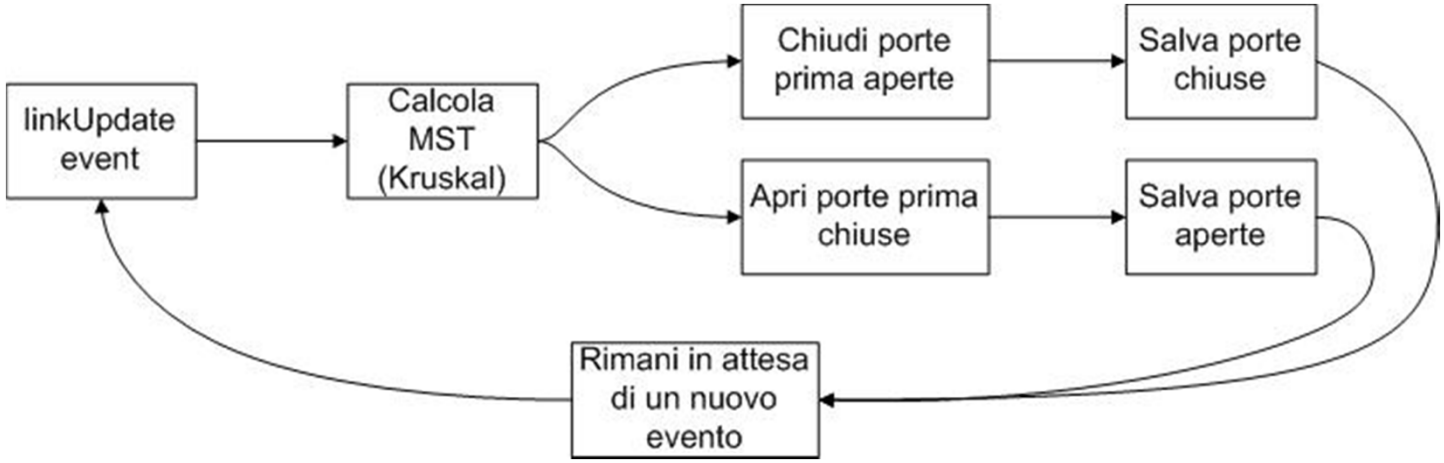
- Due router
  - Router primario per best-effort
  - Router secondario per traffico della ricerca

- Pacchetti diretti normalmente verso il router primario (def. gateway)
- Se porta TCP_dst superiore a 35000 (tipico grid)
  - Cambio indirizzo MAC address di destinazione e instrado il pacchetto con L2Switch

- Niente porte di uscita statiche
  - Funzionalità di L2 switch rimangono! (MAC table)

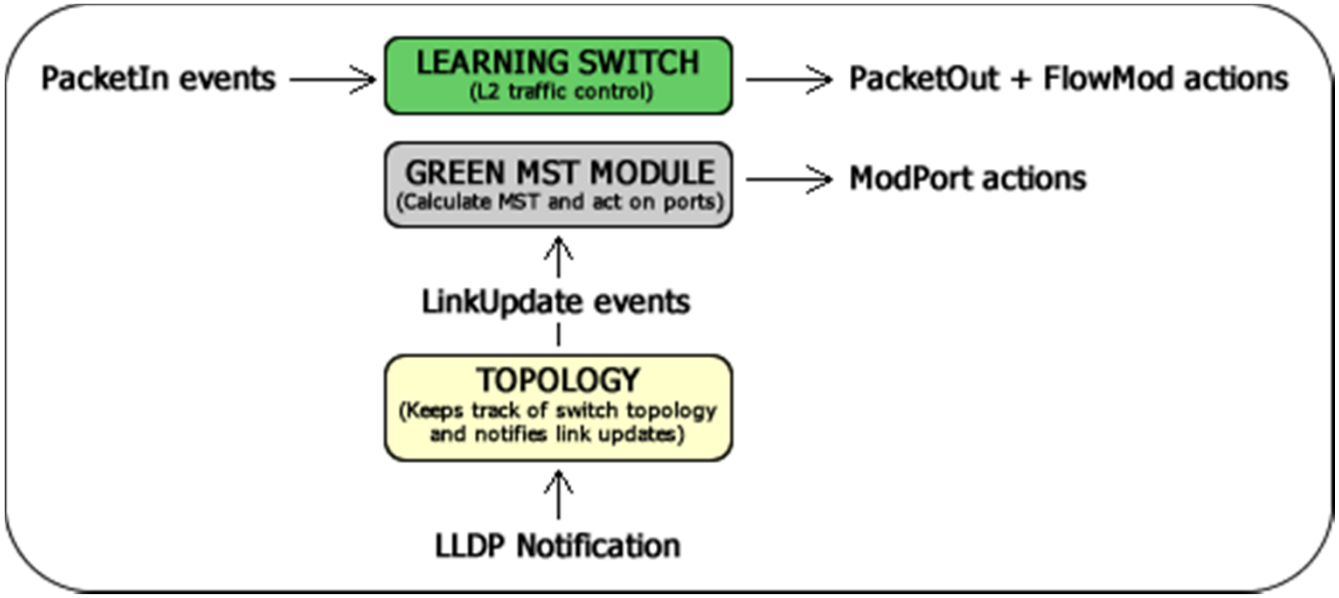
Use case 2 - Loop-free e fast-failover

- Problema: Il modulo di L2Switch dei controller non supporta i loop di rete (a meno che non lo facciano nativamente gli switch)
- Obiettivi:
  - Usare L2Switch con looped topologies
  - Offrire funzionalità di recovery e fail-over
  - Additional requirement: risparmiare energia
- Soluzione:
  - Sfruttare il modulo topology per calcolare il Minimum Spanning Tree della rete e spegnere le porte che non ci servono



Implementazione in Beacon

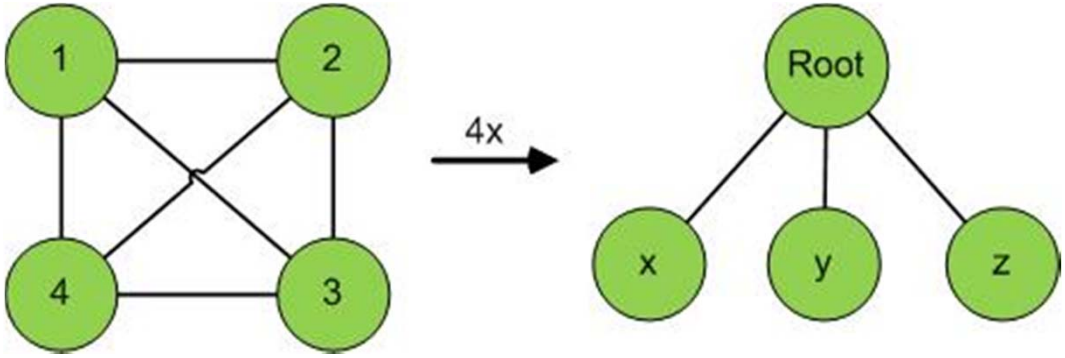
- Creazione di un nuovo modulo: GreenMST
  - Input: eventi di cambio topologia tra gli switch (Topology)
  - Costruzione di MST (Kruskal)
  - Modifica dello stato delle porte attive attraverso comandi OF
  - Minimizzazione comandi inviati agli switch (ricordo lo stato precedente)



- Gli L2 switch fanno il forwarding dei pacchetti solo attraverso i percorsi minimi, grazie alle porte aperte a loro disposizione

Use case 3 - Floodlight e modulo di routing

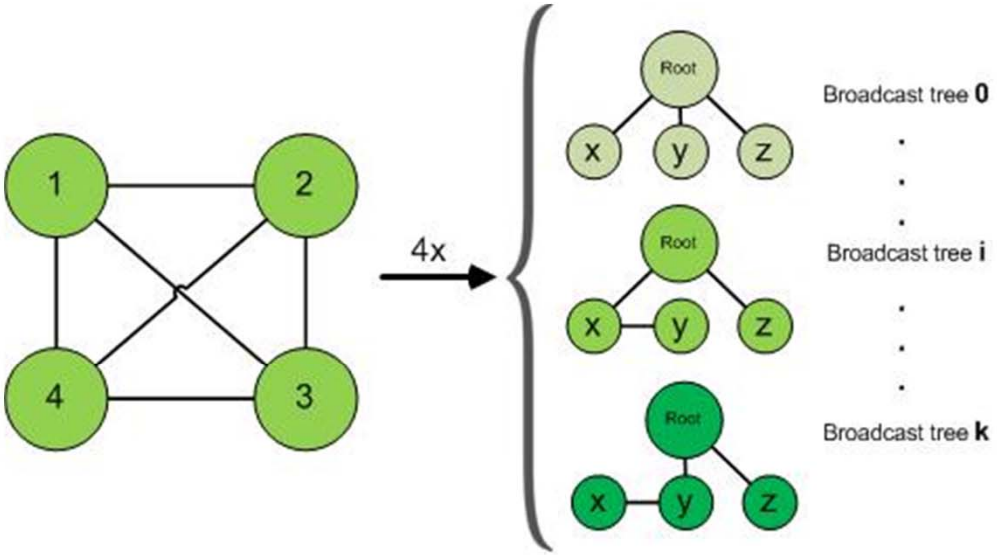
- Modulo di routing: evoluzione del modulo di LearningSwitch
- L'algoritmo originario:



  - Per ogni nodo della rete costruisce l'albero di broadcast
  - Installa adeguatamente i flussi sugli switch
  - Niente loop, flooding solo su albero di broadcast
- Benefici:
  - Superato il concetto di Spanning Tree Protocol
  - Ho sempre la mappa aggiornata della rete grazie ai moduli Device e Topology
- Problema:
  - In caso di recovery devo ricalcolare tutti gli alberi di broadcast

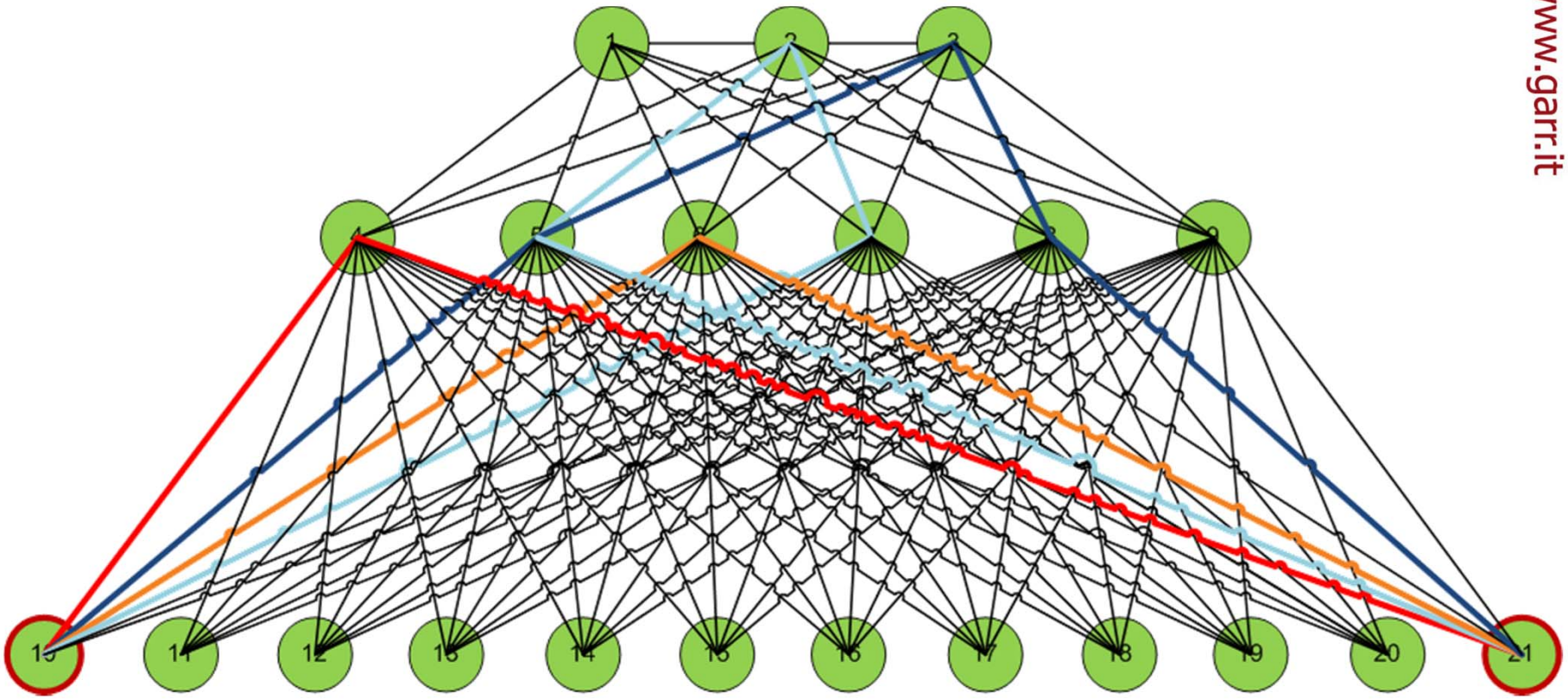
Soluzione: K-Shortest path algorithm

- Obiettivo:
  - Non dover ricalcolare ad ogni cambio di topologia gli alberi di broadcast per ogni nodo di rete
- Soluzione:
  - Adozione dell'algoritmo di K-Shortest-Path (KSP): per ogni coppia di nodi calcola i k cammini minimi in ordine di peso
- Nuovo algoritmo:



- Cache
- Costruzione dinamica broad. tree
- Calcola KSP per ogni nuovo nodo
- Se un percorso si guasta annullalo
- Se tutti i k percorsi sono invalidati per una coppia di nodi ricalcola KSP

Un caso d'uso realistico



- L'algoritmo ha calcolato i primi quattro percorsi tra il nodo 11 e il nodo 21
- In caso di fallimento del percorso minimo (rosso = 2 hop)
  - Selezionato il percorso minimo disponibile (arancione = 2 hop)
  - ...e così via

Conclusioni

- Lo sviluppo del test-bed e dei protocolli ha permesso in tempi brevi di acquisire la conoscenza del protocollo OF e di contribuire al suo sviluppo
  - Feedback positivi da partecipanti conferenza EWSDN, anche commerciali

- Validato ampio spettro applicativo:
  - È possibile re-direzionare i flussi di traffico in base all'analisi dei loro pacchetti
  - Si possono usare tecnologie loop-free con LearningSwitch, anche con switch OpenFlow non muniti di STP
  - Grazie al modulo GreenMST, risparmiare energia all'interno dei DC
  - Sono state migliorate le tempistiche e le performance, in caso di recovery, del modulo di routing

- Il protocollo permette infinite possibilità grazie alla introduzione dello strato SW (SDN)
  - Va adeguatamente progettato e sviluppato

Campi di applicabilità di OF per GARR

- Grazie al multi layer switching è semplice e rapido controllare un grande numero di device L2/L3 definendo e gestendo policy multi-layer
  - Routing
  - Firewall
  - Traffic engeneering
  - Reti virtuali per utente

- Possibile alternativa alle tecnologie per creazione delle tradizionali VPN
  - VLAN, Q-in-Q, PBB, MPLS, ...

- Possibile semplificazione nella gestione del layer2 di GARR-X

Prossimi passi

- Valutazione di un test-bed con risorse anche fisiche
  - Partecipazione a progetti europei
  - Sviluppo di eventuali collaborazioni

- Introduzione di FlowVisor: Ambiente multi-controller distribuiti
  - Sperimentazione
  - Analisi della sicurezza, isolamento del traffico

- Semplificazione dell'esperienza utente
  - Interfaccia web + client desktop
  - Analisi dei principali tool di amministrazione e debug

- Integrazione dei dati provenienti dal monitoring passivo
  - SNMP
  - Database + Netflow

GRAZIE

Domande

Reference

- Software Defined Networking
  - Nick McKeown, “Software Defined Networking”, Infocom April 2009, Brasil
  - Open Networking Foundation (ONF): https://www.opennetworking.org
- Sito originale OpenFlow: http://www.openflow.org
- Controllers:
  - Beacon: https://openflow.stanford.edu/display/Beacon/Home
  - Floodlight: http://floodlight.openflowhub.org
  - Open vSwitch: http://www.openvswitch.org
- EWSDN 2012: http://www.ewsdn.org
- Materiale su OpenFlow prodotto fino ad ora: https://wiki.garr.it/bin/view/EP/OFSDN