Università di Pisa

Gestione degli indirizzi anycast IPv4 con SDN

Progetto di Advanced Network Architectures and Wireless Systems

A.A. 2016/2017

Lorenzo Biagini

Chiara Caiazza

Martina Troscia

Introduzione

Con questa relazione si intende mostrare la progettazione di un controllore SDN che gestisca indirizzi anycast IPv4 all'interno di una rete composta da più switch SDN a cui sono connessi degli host. Il controllore, implementato tramite il framework Floodligth, espone un'interfaccia WEB di tipo RESTful che permette l'assegnamento dinamico di indirizzi anycast a un gruppo di host. A un host possono essere assegnati più indirizzi anycast. Il controllore garantisce che un pacchetto destinato ad un indirizzo anycast venga inoltrato ad un solo host tra quelli registrati a tale indirizzo. La selezione dell'host è decisa tenendo conto della distanza dell'host stesso dal mittente del pacchetto e del carico sugli host candidati come destinatari.

Terminologia

Nel corso di questo documento verranno utilizzati i seguenti termini:

- Un *server* è un host iscritto ad un gruppo anycast. I server non effettuano richieste ad altri server.
- Un *client* è un host che effettua richieste verso un indirizzo anycast.
- L'aggettivo *fisico*, quando è riferito ad un indirizzo IP di un server, serve a discriminarlo dall'indirizzo anycast a cui tale server si è registrato.

Scelte di progetto

Durante l'implementazione, sono state fatte alcune assunzioni sul funzionamento del progetto:

- Il controllore intercetta i messaggi inviati dai client e converte l'indirizzo anycast destinatario nell'indirizzo host fisico del server che deve prendere in carico la richiesta. Anche i messaggi di risposta inviati dai server devono essere modificati dal controllore, che vi inserisce l'indirizzo anycast corrispondente in modo che i client siano convinti di parlare con un solo server.
- Ogni server può gestire al più MAX_NUM richieste ogni TIME_INTERVAL millisecondi. Superata questa soglia, non è più disponibile fino allo scadere dell'intervallo temporale e ulteriori richieste devono essere inoltrate a un altro server o scartate, se tutto il gruppo anycast è occupato. A questo proposito, il controllore, alla ricezione di una richiesta diretta a un certo indirizzo anycast, controlla la lista dei server iscritti a tale gruppo ed inoltra il messaggio ricevuto al server più vicino disponibile.
- Ad un indirizzo anycast possono corrispondere più servizi, purché siano registrati su porte diverse. Questo significa che i server che forniscono servizi diversi possono registrarsi in uno stesso gruppo anycast.

- Un server può iscriversi a più gruppi anycast. Dal momento che un servizio è determinato dalla porta con cui il server si mette in ascolto, questi non può effettuare più di un'iscrizione su una stessa porta. Tale scelta è guidata dal fatto che, per determinare quale sia l'indirizzo anycast che il client si aspetta di vedere come mittente nella risposta, il controllore deve utilizzare la porta per capire quale sia il servizio (e quindi l'indirizzo anycast) da cui il messaggio proviene.
- I ping effettuati dai client verso un indirizzo anycast devono essere gestiti dal controllore e non possono essere inoltrati all'interno della rete. Si supponga, ad esempio, che il messaggio di ping venga inoltrato fino al server più vicino, disponibile e iscritto al gruppo anycast. Tale server genererebbe una risposta per la quale il controllore non ha informazioni sufficienti per scambiare l'indirizzo fisico del server con l'indirizzo anycast che il client si aspetta di ricevere. All'arrivo di un ping, quindi, il controllore produce direttamente il messaggio di risposta, se ci sono server registrati all'indirizzo anycast, e lo invia allo switch per l'inoltro.

Limiti dell'implementazione

La soluzione realizzata presenta dei limiti dovuti ai mezzi usati per simulare le richieste dei client ai server. Gli host generati da Mininet possono eseguire solamente alcuni comandi di shell. Pertanto, è stato usato il software `iperf` per generare pacchetti di tipo TCP/UDP, i quali sono considerati come se fossero delle richieste da parte dei client.

- Con TCP ci sono dei problemi dopo che un server non è più in grado di accettare le richieste e il traffico viene quindi reindirizzato. La comunicazione del client con il secondo server non avviene correttamente perché non viene rieseguito l'handshake per stabilire una nuova connessione con il nuovo server: il client continua a mandare pacchetti che però vengono rifiutati dal server con un messaggio di reset.
- Nell'utilizzo di `iperf` con il protocollo UDP, solo l'ultimo pacchetto inviato richiede una reazione da parte del server. Se all'arrivo di questo pacchetto tutti i server appartenenti al gruppo anycast sono momentaneamente non disponibili, non viene generata alcuna risposta.

Gestione delle richieste

Alcuni tipi di pacchetti (ARP, ICMP, TCP e UDP) sono gestiti esplicitamente per permettere la corretta traduzione di un indirizzo anycast in un indirizzo fisico e viceversa. È stato necessario intervenire sul comportamento standard del modulo *Forwarding* il quale inserisce negli switch le regole per l'inoltro in broadcast di tutti i pacchetti ricevuti. Infatti, inserendo il modulo LoadBalancer *dopo* quello di Forwarding,

- Il modulo Forwarding installa una regola per l'inoltro in broadcast del messaggio che ha come destinazione (sorgente) un indirizzo anycast (fisico).

- Non è stato trovato un modo per annullare tale comportamento una volta impostato.

- Il modulo LoadBalancer genera un pacchetto traducendo l'indirizzo anycast di destinazione (fisico di sorgente) in indirizzo fisico (anycast).

- Il modulo LoadBalancer installa una regola per l'inoltro di tale pacchetto che, con la regola impostata dal precedente modulo, genera l'invio di due pacchetti dal medesimo switch.

Mettendo il modulo LoadBalancer *prima* di quello di Forwarding, non è possibile reperire informazioni sui punti di collegamento degli host con gli switch (non è quindi possibile calcolare le rotte tra sorgente e destinazione).
È stato anche necessario implementare il modo in cui installare le rotte che i pacchetti devono seguire verso la destinazione: la funzione pushRoute offerta dal modulo Forwarding non installa alcuna regola sull'ultimo hop del percorso. Viene a mancare così l'informazione sulla porta che quest'ultimo dovrebbe usare per raggiungere la destinazione.
Quando uno switch riceve una richiesta, invia un messaggio OFPacketIn al controllore che analizza l'header del pacchetto per stabilire se è una richiesta da parte di un client o una risposta da parte di un server.

- *Richiesta verso un servizio*: la sorgente ha un indirizzo unicast e la destinazione ha un indirizzo anycast. Il controllore sceglie l'host fisico a cui inoltrare la richiesta in base alla distanza dal client e al carico di lavoro su di esso. Viene cambiato l'indirizzo IPv4 del destinatario applicando una lista di OFAction e generato un OFPacketOut per lo switch. La regola di traduzione dell'indirizzo non viene installata nella FlowTable dello switch che ha ricevuto la richiesta, perché il controllore deve tener traccia di tutti i messaggi inviati per monitorare il carico di lavoro e aggiornare il contatore delle richieste. Viene invece installata una regola sugli switch successivi al primo per impostare il percorso che i pacchetti dovranno seguire (tramite la funzione pushRoute).

- *Risposta verso un client*: la sorgente ha un indirizzo anycast e la destinazione ha un indirizzo unicast. Il controllore installa una regola sul primo switch della rotta verso il client per modificare il campo contenente l'indirizzo fisico del mittente con l'indirizzo anycast che il client si aspetta. Questa regola viene sia applicata al pacchetto ricevuto che installata nella FlowTable: il contatore delle richieste viene resettato allo scadere dell'intervallo temporale e quindi non è necessario l'intervento del controllore in questo caso. Negli switch successivi al primo è installata una regola, che ha un match con l'indirizzo sorgente anycast appena scambiato, per impostare il percorso che i pacchetti devono seguire.

All'arrivo di un messaggio tra host fisici, il controllore non viene mai contattato perché le regole per l'inoltro di tali pacchetti sono settate all'arrivo dei messaggi di richiesta da parte dei client e di risposta da parte dei server.

Manuale d'uso

Configurazione di Floodlight e della Mininet

Il controllore SDN implementato tramite il framework Floodlight, va eseguito sovrascrivendo i due file

```
src/main/resources/floodlightdefault.properties
```

```
src/main/resources/metainf/services/net/floodlightcontroller
    /core/modules/IFloodlightModule
```

Tali file contengono la lista dei moduli utilizzati e l'ordine in cui questi devono elaborare i messaggi inviati al controllore. Il corretto funzionamento del controllore viene testato collegando ad esso una rete virtuale emulata contenente switch SDN e host che inviano e generano traffico. La rete da usare nei test può essere generata con il comando:

```
sudo mn --topo=linear,4 --mac --switch=ovsk --controller=
    remote,ip=127.0.0.1,port=6653,protocols=OpenFlow13 --
    ipbase=10.0.0.0
```

In questo modo si genera una rete contenente quattro switch, a ognuno dei quali è connesso un host.

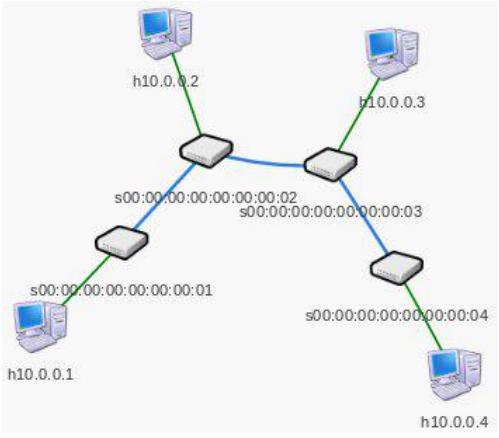

Figura 1: Topologia della rete

La connettività di base tra gli host può essere verificata tramite il comando `pingall`: in questo modo, si assicura anche che gli switch apprendano quali siano gli host collegati e la loro posizione.

Uso dell'interfaccia REST Easy

Tramite l'interfaccia REST Easy messa a disposizione dal browser Firefox è possibile iscrivere più server a un indirizzo anycast. Supponendo che si vogliano registrare due server all'indirizzo anycast `9.9.9.9` per uno stesso servizio, va fatta una richiesta all'indirizzo

```
http://localhost:8080/lb/controller/subscribe/json
```

Nel corpo del messaggio di tipo POST (in *Data*) specificare i seguenti valori:

- Opzione: *Custom*
- MIME type: application/json
- Corpo: {"type":"subscribe", "anycast":"9.9.9.9", "physical":["10.0.0.1:1080", "10.0.0.2:1080"]}

Tramite la stessa interfaccia, interrogando la risorsa

```
http://localhost:8080/lb/controller/unsubscribe/json
```

è invece possibile cancellare l'iscrizione di uno o più server a un certo indirizzo anycast. Il formato del messaggio è del tutto analogo al caso precedente, l'unica differenza è che in quest'ultimo caso si ha "type":"unsubscribe" all'interno del corpo. Infine, REST Easy può anche essere utilizzata per verificare quali siano i server iscritti a un certo indirizzo anycast. In questo caso va fatta una richiesta di tipo GET alla risorsa

```
http://localhost:8080/lb/controller/showlist/json
```

specificando come parametro `anycast=9.9.9.9`, se si intende visualizzare i server iscritti all'indirizzo anycast `9.9.9.9`.

Test di funzionamento

È possibile fare il ping a un indirizzo anycast, assicurandosi che sia stata impostata in precedenza una rotta statica per inoltrare allo switch ogni pacchetto con un indirizzo di destinazione non appartenente alla sottorete (`10.0.0.0` nell'esempio), con il comando

```
route add -net 0.0.0.0/32 dev h3-eth0
```

(se è l'host 3 a voler effettuare il ping). Il controllore verifica che ci sia almeno un server registrato all'indirizzo anycast a cui si sta facendo ping e risponde con un messaggio di *Echo reply* in caso affermativo, o con il messaggio *Destination unreachable* nel caso contrario.
La verifica del bilanciamento del carico di lavoro può essere verificata tramite l'uso del comando `iperf`: supponendo di aver iscritto due server, `10.0.0.1` e `10.0.0.2`, all'indirizzo anycast `9.9.9.9` per il servizio offerto sulla porta `1080`, questi possono essere messi in ascolto su tale porta con il comando

```
iperf -s -p 1080 -u
```

Un client, `10.0.0.3` per esempio, può connettersi all'indirizzo anycast `9.9.9.9` con il comando

```
iperf -c 9.9.9.9 -p 1080 -u
```

Per far ciò, anche in questo caso, è necessario che sia stata impostata in precedenza la rotta statica verso lo switch. Tramite Wireshark è possibile vedere che il server più vicino al client gestisce MAX_COUNT richieste e poi è costretto a fermarsi per il raggiungimento del valore limite; il controllore devia quindi il traffico all'altro server appartenente al gruppo anycast. Quando anche in questo secondo server si raggiunge il massimo numero di richieste, per un po' di tempo il traffico prodotto dal client viene scartato finché il primo server non torna disponibile.