«به نام خدا»

مرضیه علیدادی _ 9631983 _ پروژه 2

2_1_1 .

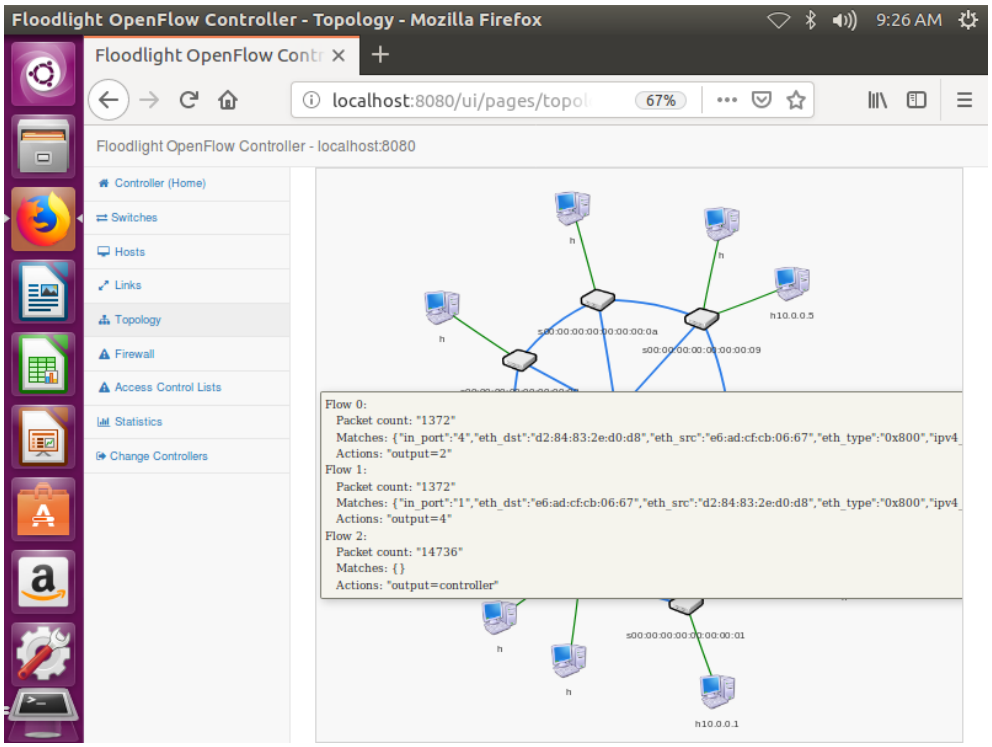
H1 -> S1 -> S3 -> S6 -> S9 -> H5

با توجه به اینکه packet count برای سوییچ های این مسیر در حال افزایش بود، متوجه شدم که از این مسیر بسته ها در حال ارسال هستند.

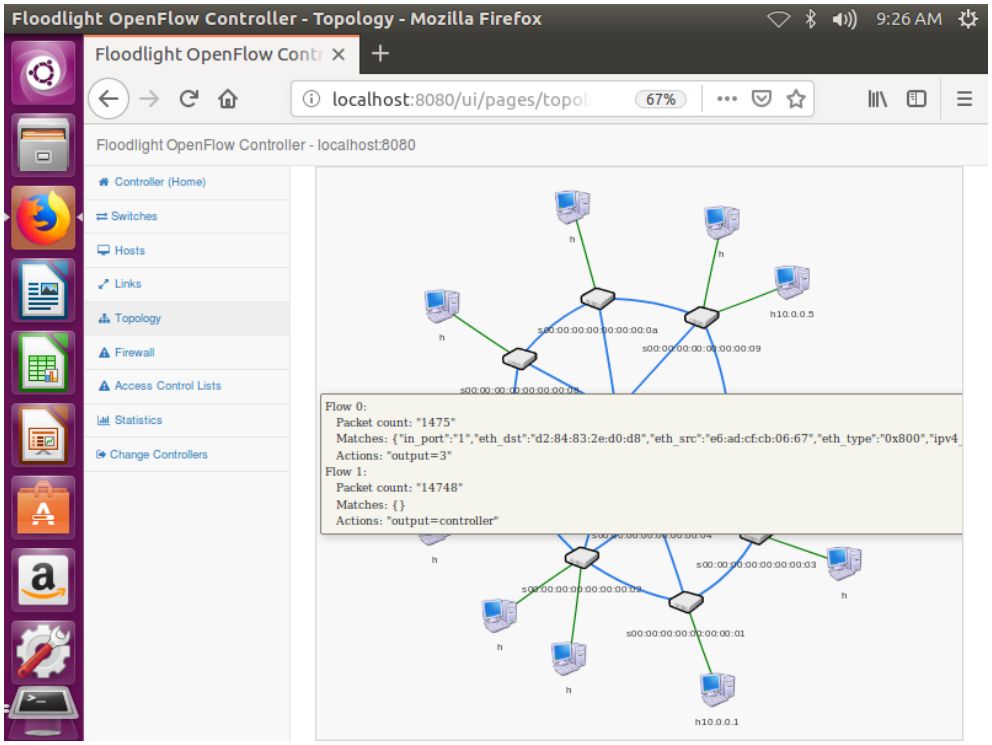
و البته در سوییچ ها، flow های متناظر با این مسیر ذخیره شده است.

{{وقتی که میخواستم یک بار دیگر این کار ها را انجام دهم تا ویدئو بگیرم، بسته ها از مسیر دیگری عبور کردند. ولی زمانی که این فایل را مینوشتم، مسیر گذشتن packet ها به این شکلِ گزارش شده بود. تفاوت این فایل با چیزی که در ویدئو گزارش کردم، به این دلیل است}}

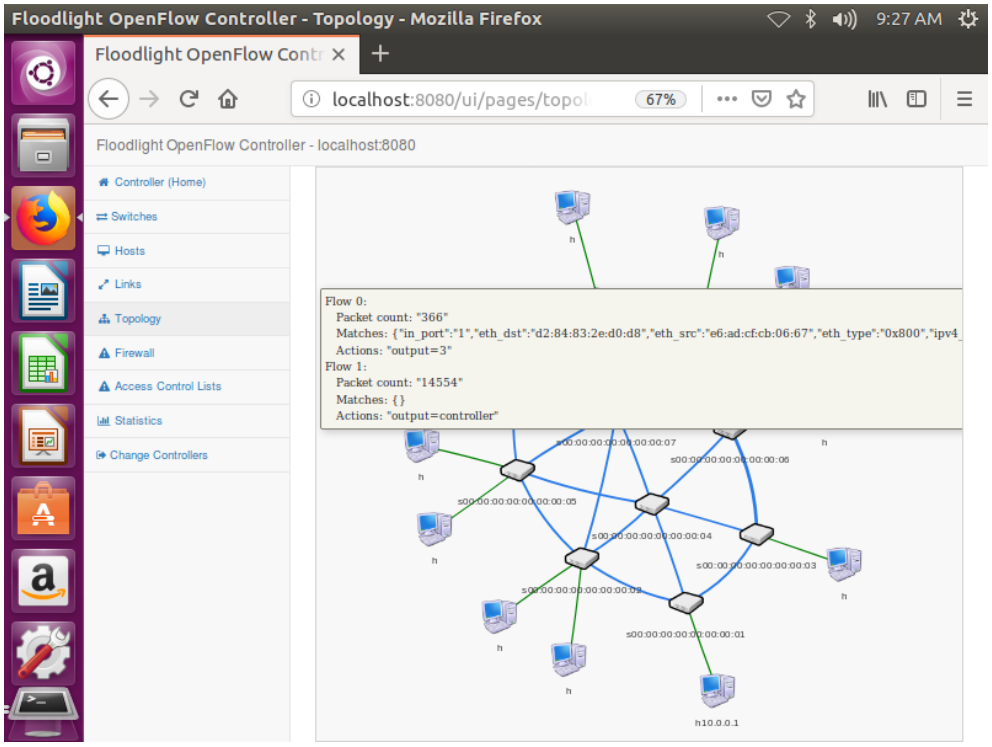
در switch1 :



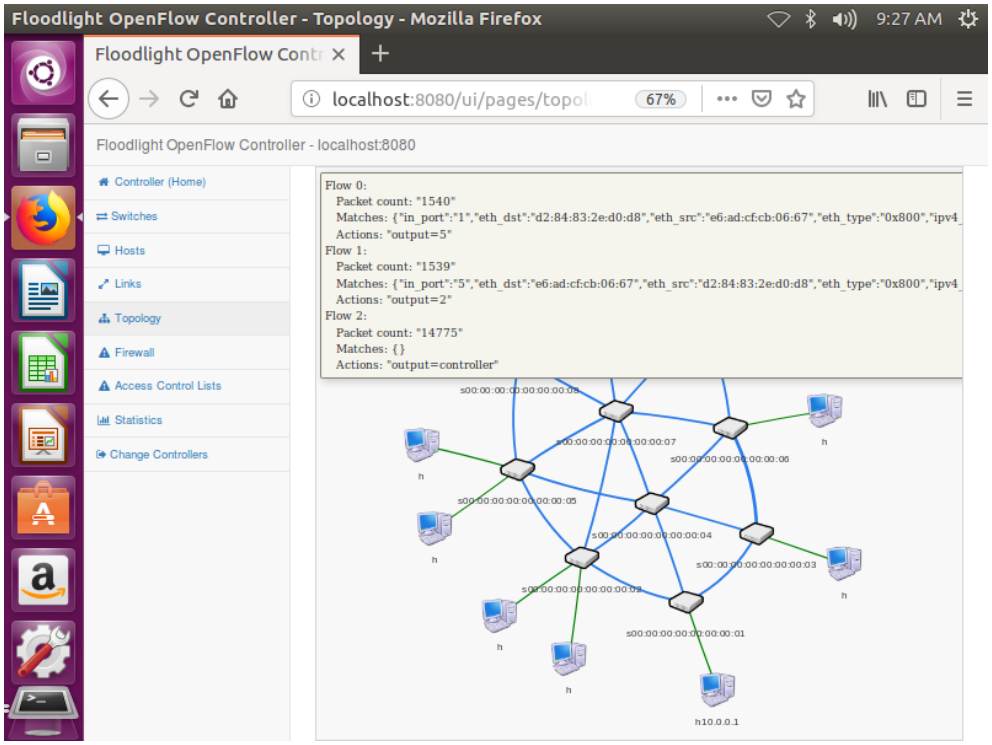
در switch3 :



در switch6 :



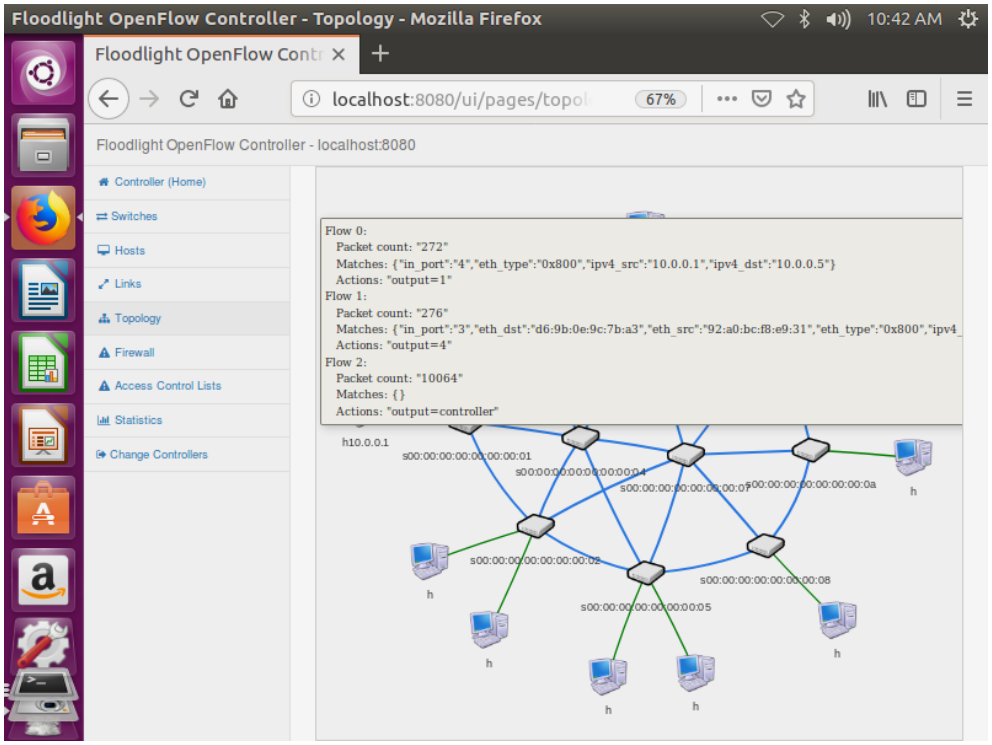
در switch9 :



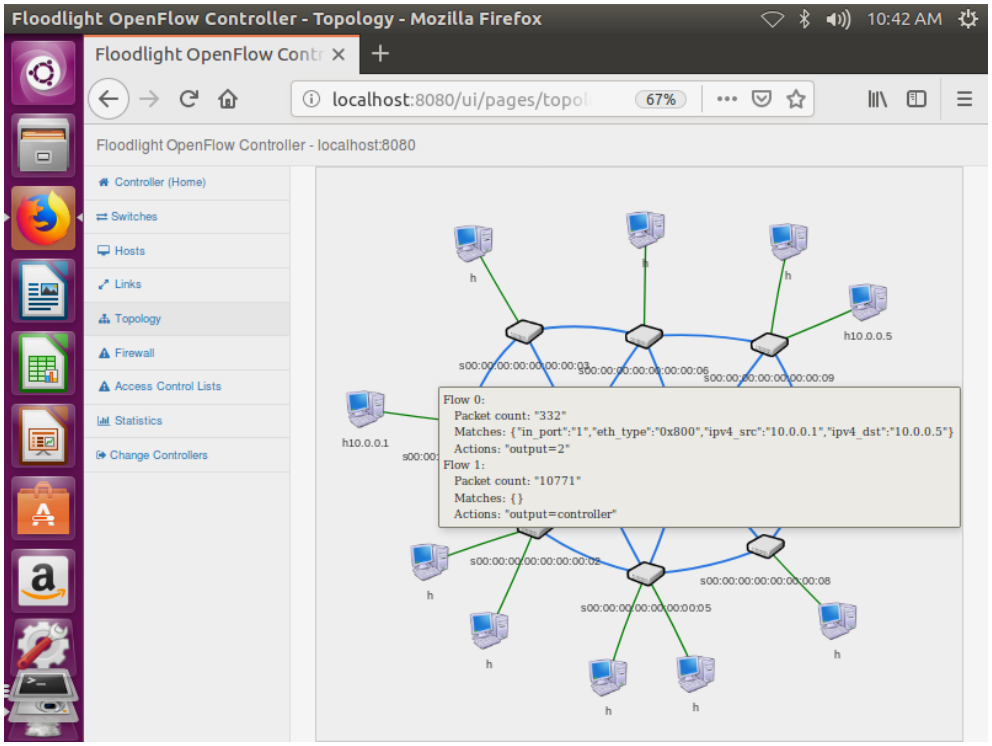
همانطور که مشخص است یک مسیر برای ارسال از 1 به 5 مشخص شده. که در بالا نوشتم.
(Flow ی دیگری که در سوییچ ها تشکیل شده، مربوط به مسیر برگشت بسته ها است؛
همانطور که میدانیم ping یک ارتباط دوطرفه است.)

2_2 : با اضافه کردن entry ها به سوییچ ها، مسیر تخصیص داد شده به flow ها به این صورت تغییر کرد:

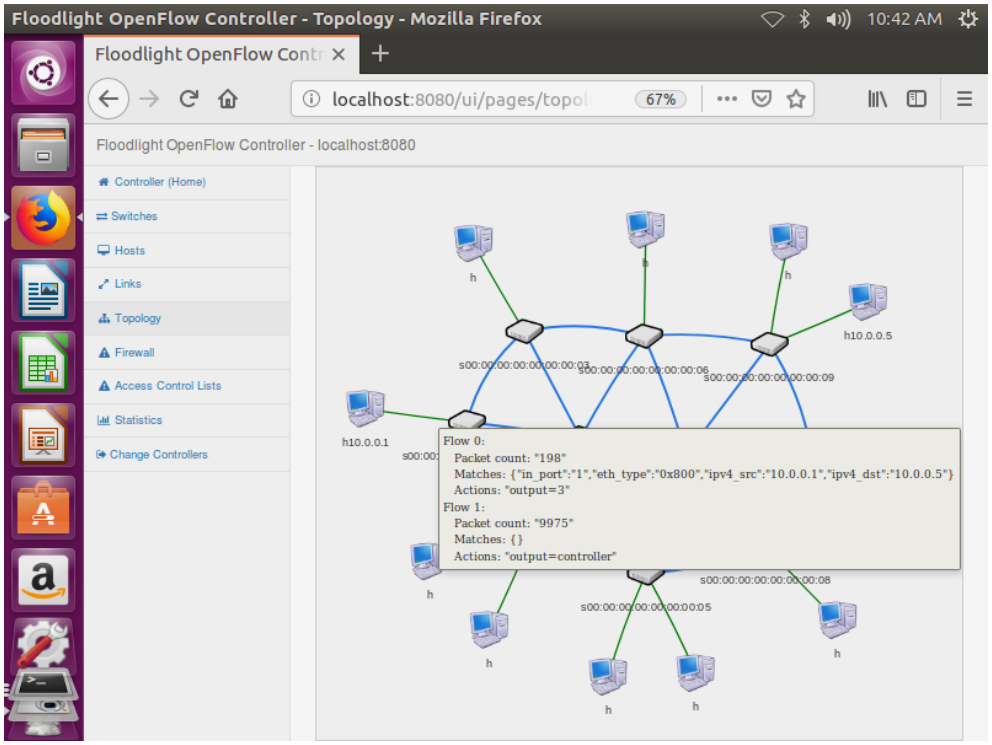
در switch1 :



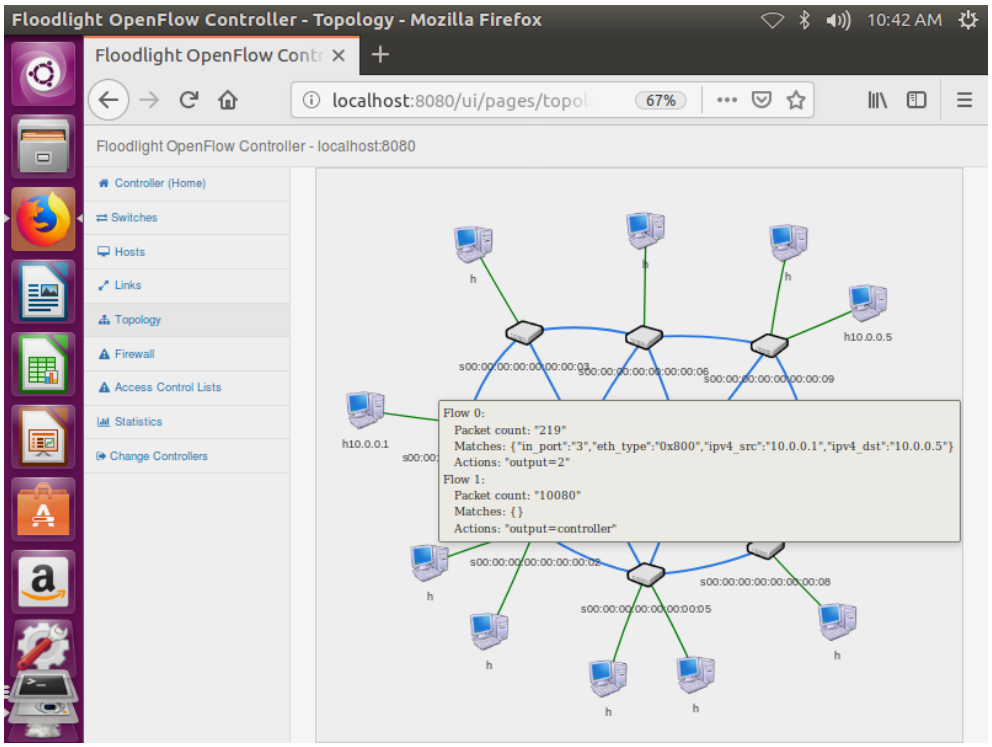
در switch2 :



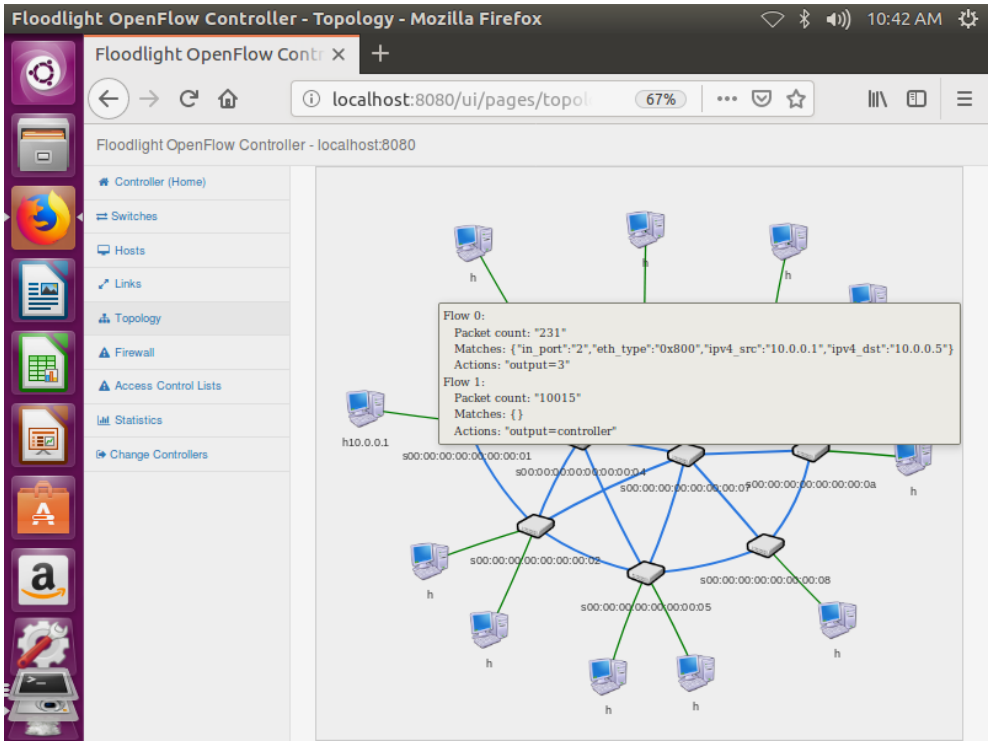
در switch5 :



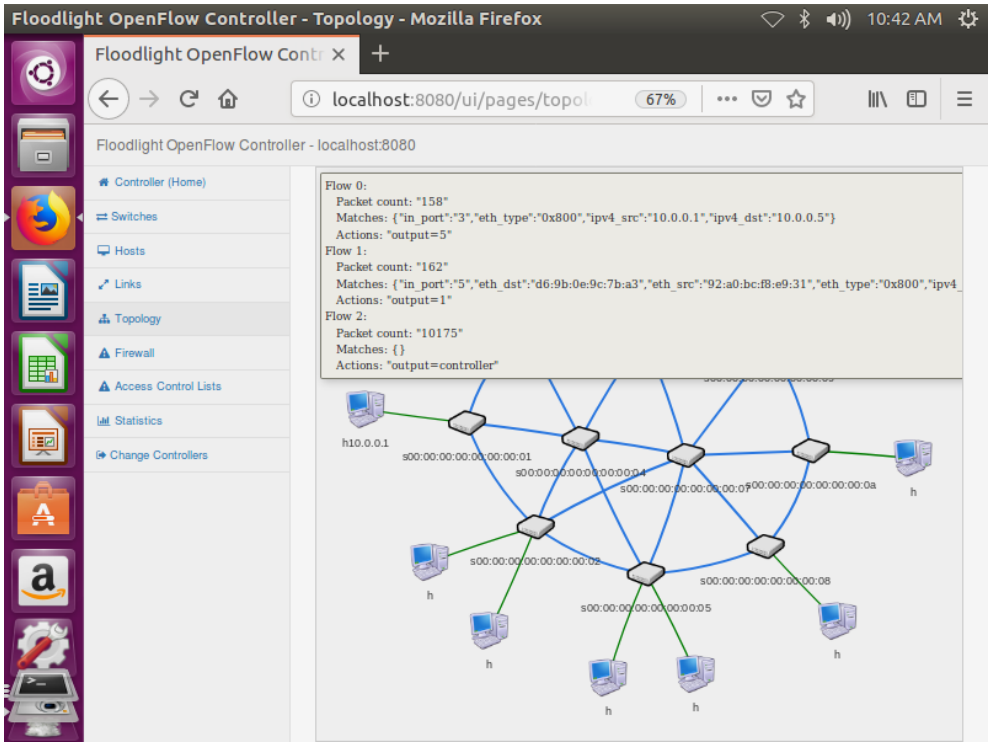
در switch8 :



در switch10 :

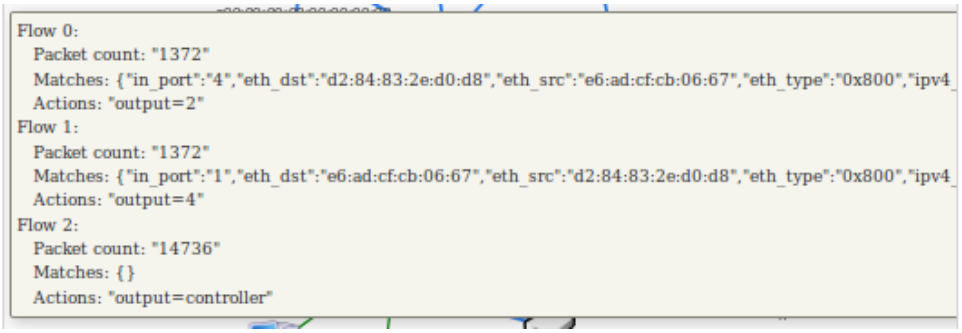


در switch9 :

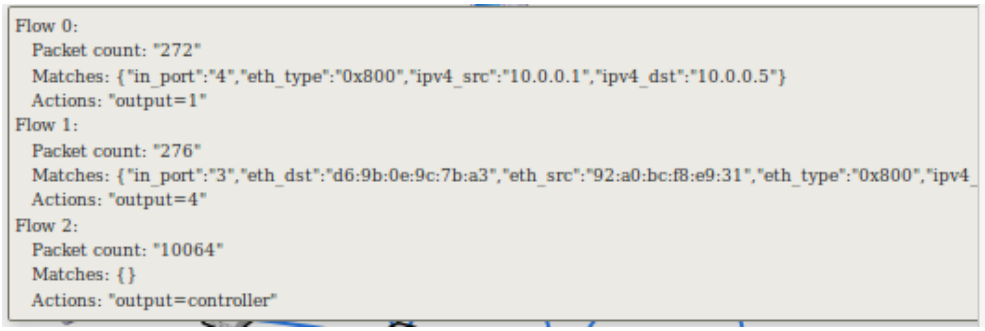


. 1_3_2

جدول Switch1 در مرحله ی اول:



جدول Switch1 در مرحله ی دوم:



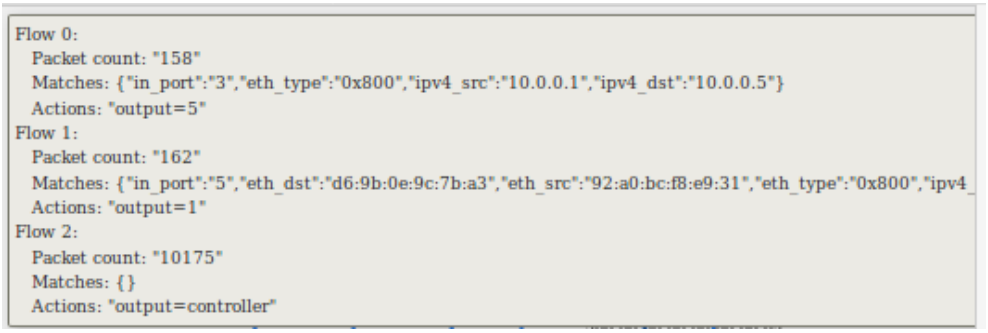
در این سوییچ entry های موجود برای flow ی مورد نظر تغییر کرده. و در مرحله ی اول وقتی بسته ای از پورت 4 به سوییچ وارد میشد، به پورت 2 هدایت میشد. ولی در مرحله ی دوم وقتی بسته ای از پورت 4 به سوییچ وارد بشود، به پورت 1 هدایت میشود.

این هدایت های متفاوت بر اساس مسیرهای متفاوتی است که به این flow اختصاص داده شده است.

البته مسیر برگشت از پورت 4 هم تغییر کرده است.

جدول Switch9 در مرحله ی اول:

جدول Switch9 در مرحله ی دوم:



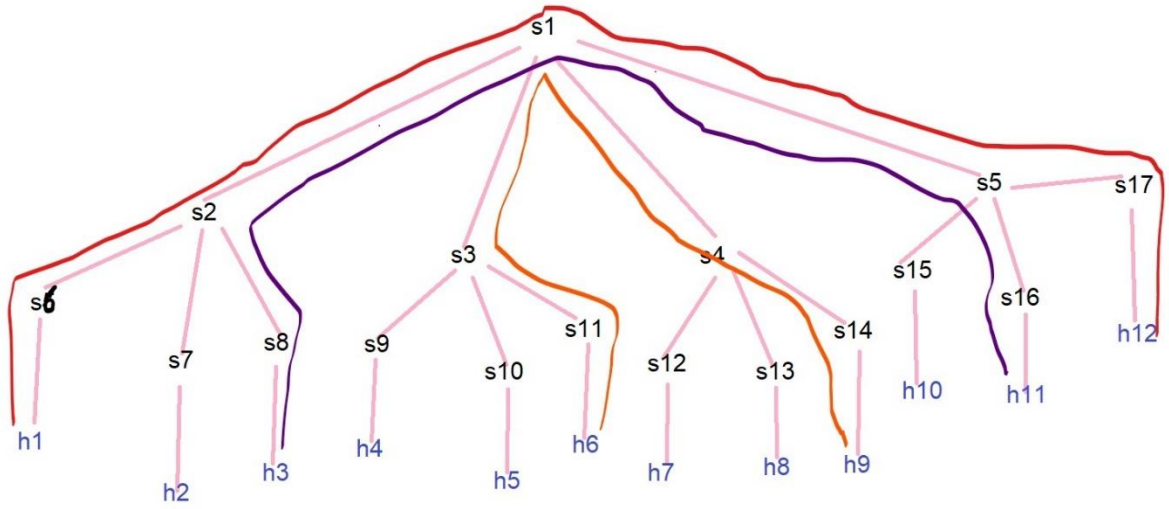
در این سوییچ هم entry های موجود برای flow ی مورد نظر تغییر کرده. و در مرحله ی اول وقتی بسته های وارد شده از پورت 1 را به پورت 5 هدایت میکند. ولی در مرحله ی دوم وقتی بسته ای از پورت 3 به سوییچ وارد بشود، آن را به پورت 5 هدایت میکند.
این هدایت های متفاوت بر اساس مسیرهای متفاوتی است که به این flow اختصاص داده شده است.
البته مسیر برگشت از پورت 5 هم تغییر کرده است.

4_2 . بله؛ از همان مسیر تعیین شده عبور میکند.
با بررسی جداول سوییچ ها و دیدن entry های ذخیره شده برای flow ی مربوطه و با بررسی تغییرات packet count و افزایش آن در سوییچ های مسیر تعیین شده، با رفرش کردن صفحه در هنگام ping گرفتن، مشخص میشود.

که البته packet count علاوه بر اینکه در جدول سوییچ ها در این صفحه ی توپولوژی قابل مشاهده است، در صفحه ی سوییچ ها هم برای هر سوییچ قابل بررسی است.

---

3 . توپولوژی من در پروژه ی قبل به این صورت بود:



سه flow ی مورد نظر را مشخص کرده ام.

(1) مسیر اول برای بسته هایی است که از h1 به h12 میروند. مسیری که باید داشته باشند، در شکل با رنگ قرمز مشخص شده است.

H1->S6->S2->S1->S5->S17->H12

(2) مسیر دوم برای بسته هایی است که از h3 به h11 میروند. مسیری که باید داشته باشند، در شکل با رنگ بنفش مشخص شده است.

H3->S8->S2->S1->S5->S16->H11

(3) مسیر سوم برای بسته هایی است که از h6 به h9 میروند. مسیری که باید داشته باشند، در شکل با رنگ نارنجی مشخص شده است.

H6->S11->S3->S1->S4->S14->H9

(چون این توپولوژی به صورت درختی است، برای هر مبدا و مقصد فقط یک مسیر وجود دارد.)

---

4 . این دو کنترلر از کنترلر های SDN هستند.

مقایسه:

Floodlight و OpenDaylight دو کنترل کننده متداول در بین کنترلرهای SDN هستند.

هرکدام از این دو کنترلر ماژولار بوده و برای خدمات جدید شبکه قابل برنامه ریزی است.

هردو open source هستند و با زبان جاوا پشتیبانی می شوند.

همچنین بین Floodlight و OpenDaylight تفاوت هایی وجود دارد. Floodlight دارای رابط کاربر مبتنی بر جاوا است در حالی که OpenDaylight از پروتکل های کنترل غیر OpenFlow پشتیبانی می کند و یک لایه abstract دارد.

این دو کنترل کننده دارای یک رفتار رقابتی هستند. موقعیت هایی وجود دارد که هر دو کنترل کننده عملکرد یکسانی دارند. موقعیت هایی وجود دارد که شبکه هایی با OpenDaylight دارای تأخیر بهتری هستند، در حالی که Floodlight می تواند بهتر از OpenDaylight از نظر از بین رفتن ها در بعضی شبکه ها عمل کند.

هر دو دارای rest api هستند. و با mininet قابل اجرا هستند.

Floodlight قدیمی تر است.

حداکثر زمان تاخیر در Floodlight بسیار بیشتر از OpenDaylight است ، به این معنی که Floodlight به زمان بیشتری برای یافتن مسیر و ارسال تصمیمی برای جریانهای تازه رسیده نیاز دارد. به طور متوسط، زمان تاخیر برای شبکه هایی که از Floodlight به عنوان کنترل کننده خود استفاده می کنند، در زمانی که ترافیک کم است، دارای تاخیر بیشتری هستند.

OpenDaylight می تواند در شبکه هایی با بار کم برای هر نوع توپولوژی عملکرد بهتری داشته باشد.

هر دو کنترل کننده در حالی که بار شبکه در حدود 50٪ از پهنای باند واقعی باشد، یکسان عمل می کنند. بجز توپولوژی tree که در آن OpenDaylight از Floodlight بهتر عمل می کند.

برای شبکه هایی که دارای ترافیک سنگین هستند ، OpenDaylight فقط برای

توپولوژی های single بهتر عمل میکند. چون loss کمتری دارد.

برای شبکه هایی که دارای ترافیک سنگین هستند ، Floodlight به دلیل تاخیر و loss کمتر، می تواند عملکرد بهتری نسبت به OpenDaylight برای توپولوژی های linear و tree داشته باشد.

OpenDaylight برای توپولوژی های single و دارای بارهای مختلف شبکه ، انتخاب بهتری است.

Floodlight برای توپولوژی های linear با بارهای مختلف شبکه انتخاب بهتری است.

Floodlight برای توپولوژی های tree با بارهای سنگین شبکه و ترافیک حساس به loss ، مانند فیلم و صدا ، انتخاب بهتری است.

OpenDaylight برای توپولوژی های tree با بارهای شبکه ای مختلف، به جز ترافیکی که به loss حساس است، گزینه بهتری است.

لینک ویدیو در iut box :

https://iutbox.iut.ac.ir/index.php/s/BmicyHGazA6wyKA