

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ



دانشگاه صنعتی اصفهان
دانشکده مهندسی برق و کامپیوتر

بررسی کاربرد قابلیت های جدید اوپن فلو در مدیریت و برنامه ریزی شبکه

گزارش پروژه کارشناسی مهندسی برق - مخابرات

آرمین مهدیلو ترکمانی

استاد راهنما

دکتر مسعود رضا هاشمی

اسفند ۱۳۹۹

فهرست مطالب

<u>صفحه</u>	<u>عنوان</u>
۱	چکیده
۲	۱ مقدمه
۲	۱-۱ شبکه نرم افزار محور (SDN)
۳	۲-۱ مزایای معماری SDN
۴	۳-۱ اجزاء تشکیل دهنده معماری SDN
۴	۱-۳-۱ لایه زیرساخت
۴	۱-۳-۲ ارتباط جنوبی
۵	۱-۳-۳ لایه کنترل
۵	۱-۳-۴ ارتباط شمالی
۵	۱-۳-۵ لایه برنامه کاربردی
۵	۴-۱ نحوه عملکرد و تفاوت آن با شبکه های سنتی
۶	۵-۱ ساختار گزارش
۷	۲ پروتکل Openflow
۷	۱-۲ Openflow
۸	۲-۲ اجزاء تشکیل دهنده سوئیچ Openflow
۸	۱-۲-۲ Openflow Ports
۱۰	۲-۲-۲ Openflow Pipeline
۱۱	۳-۲-۲ Flow Tables
۱۲	۴-۲-۲ Table-miss
۱۳	۵-۲-۲ Group Table
۱۳	۶-۲-۲ Meter Table
۱۳	۷-۲-۲ Openflow Channel
۱۶	۳-۲ بررسی وظایف مهم در خط لوله
۱۶	۱-۳-۲ Match
۱۶	۲-۳-۲ Instruction
۱۸	۳-۳-۲ Counter

۱۹	۳ ویژگی‌های نسخه جدید Openflow
۲۰	۱-۳ ویژگی‌های نسخه OF1.5
۲۱	۱-۱-۳ جداول جریان خروجی
۲۱	۲-۱-۳ مهم بودن نوع بسته در خط لوله
۲۲	۳-۱-۳ آماره‌های قابل انعطاف مرتبط با مدخل جریان (OXS)
۲۲	۴-۱-۳ ارسال خودکار آماره‌های مدخل جریان
۲۲	۵-۱-۳ عملیات Copy-Field
۲۲	۶-۱-۳ تطابق TCP flags
۲۲	۷-۱-۳ وضعیت ارتباط کنترل کننده
۲۳	۸-۱-۳ دستورات تکاملی گروه برای تغییرات انتخابی در سطرها
۲۳	۹-۱-۳ اجازه استفاده از wildcard در عملیات set-field
۲۳	۱۰-۱-۳ دسته‌های زمان‌بندی شده
۲۳	۱-۱-۳ تغییر Meter instruction به Meter action
۲۴	۴ بلوغ محصولات
۲۵	۱-۴ Open vSwitch (OvS)
۲۵	۲-۴ RYU sdn controller
۲۶	۳-۴ آزمایش پشتیبانی سوئیچ از نسخه OF1.5
۲۷	۱-۳-۴ محیط شبیه سازی
۲۷	۲-۳-۴ ایجاد عملکرد شبه دیوار آتش
۲۹	مراجع

چکیده

در سال‌های اخیر با توجه به افزایش چشم‌گیر استفاده از شبکه‌های کامپیوتری و نیازمندی این شبکه‌ها به دینامیک بالا به منظور اعمال تغییرات و برنامه‌ریزی سریع، مفهوم نسبتاً جدیدی به نام شبکه‌های مبتنی بر نرم‌افزار^۱ پدید آمده است. این شبکه‌ها با نگاهی مجدد به طراحی تجهیزات شبکه و جداسازی لایه‌های کنترلی^۲ از لایه‌های هدایت داده^۳ هر تجهیز باعث ایجاد امکان مدیر مرکزی، یکپارچه‌سازی و جداسازی بخش تصمیم‌گیرنده از پیچیدگی‌های بخش فیزیکی شده است.

در معماری این شبکه‌ها، ارتباط بخش کنترلی با بخش هدایت داده از اهمیت بالایی برخوردار است. پروتکل استاندارد اوپن‌فلو^۴ یکی از مهم‌ترین پروتکل‌های ارتباطی بین این دو لایه بوده که در حال حاضر به صورت وسیعی در عمل و همچنین در تحقیقات مورد استفاده قرار گرفته است. در این پروژه برآن شدیم که بررسی دقیقی درباره این پروتکل مهم داشته باشیم و همچنین ویژگی‌های نسخه جدید آن را بررسی کنیم. سپس با استفاده از محصولات پشتیبانی‌کننده از این پروتکل، نگاهی به نسخه جدید در عمل خواهیم داشت.

کلمات کلیدی: شبکه‌های مبتنی بر نرم‌افزار، پروتکل اوپن‌فلو، لایه کنترل‌کننده، لایه هدایت داده.
کلمات کلیدی انگلیسی:

Software Defined Network, SDN, Openflow, Control Plane, Data Plane, Southbound Protocol

¹Software Defined Network (SDN)

²Control Plane

³Data Plane

⁴Openflow

فصل اول

مقدمه

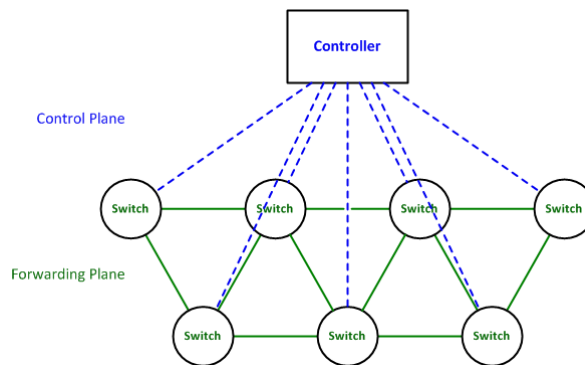
در سال‌های اخیر با توجه به افزایش چشم‌گیر استفاده از شبکه‌های کامپیوتری و نیازمندی این شبکه‌ها به دینامیک بالا به منظور اعمال تغییرات و برنامه‌ریزی سریع، مفهوم نسبتاً جدیدی به نام شبکه‌های تعریف شده بر مبنای نرم افزار یا SDN پدید آمده است.

۱-۱ شبکه نرم افزار محور (SDN)

شبکه‌های نرم افزار محور (SDN) مفهوم نو ظهوری در شبکه‌های کامپیوتری است که بر مبنای آن کنترل کننده‌های منطقی مجتمع، رفتار شبکه را کنترل می‌کنند. این گونه از معماری شبکه، فرصت‌های جدیدی به منظور ایجاد دینامیک بالاتر و تغییرات آنی و همچنین پیاده‌سازی مدل‌های مختلف امنیت را فراهم می‌آورد. در این معماری، بخش کنترل کننده^۱ تجهیزات از بخش هدایت کننده داده‌ها^۲ جدا شده و این امر موجب فراهم آوردن بستری به منظور برنامه‌ریزی مستقیم شبکه از طریق نرم افزار و انتزاعی ساختن زیرساخت شبکه از دید برنامه‌ها و سرویس‌های شبکه شده است.

^۱Control Plane

^۲Data Plane

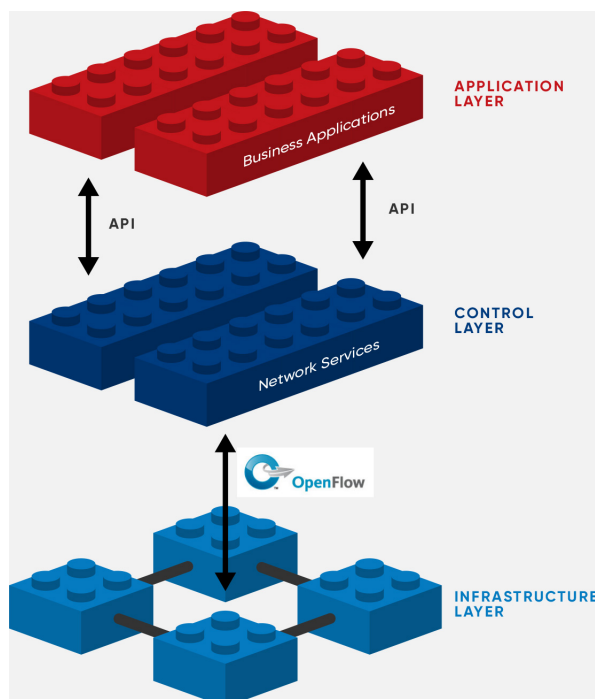


شکل ۱-۱: نمایی انتزاعی از معماری شبکه های نرم افزار محور

۲-۱ مزایای معماری SDN

- قابلیت برنامه ریزی مستقیم: با توجه به پیاده سازی بخش منطق و تصمیم گیری تجهیزات به صورت مجزا و در بستر نرم افزار، امکان برنامه ریزی مستقیم هر یک از بخش های شبکه از طریق رابط های نرم افزاری وجود دارد.
- دینامیک بالا و تغییرات لحظه ای: همانطور که انتظار می رود با پیشرفت استفاده از شبکه ها، نیازمندی به تغییرات آنی در ساختار شبکه بیش از پیش احساس می گردد. با توجه به قابلیت برنامه ریزی مستقیم تجهیزات می توان توسط رابط های نرم افزاری، تنظیمات و مسیرهای حرکت داده را به صورت خودکار و آنی تغییر داد.
- مدیریت متمرکز: با جداسازی بخش تصمیم گیرنده از بخش هدایت داده، می توان تجهیزات را به صورت متمرکز کنترل کرد. اما این مزیت خود یک عیب بزرگ نیز به شمار می رود. در صورتی که کنترل کننده مرکزی به هر دلیلی از دسترس خارج شود، تمام شبکه از کار خواهد افتاد. راه حل این مشکل، پیاده سازی دسته ای^۱ از کنترل کننده ها به منظور ایجاد قابلیت اطمینان در شبکه است.
- پایداری بالا: پایداری بالا، یکی از عوامل اصلی در اطمینان از عملکرد مناسب و مداوم شبکه است. با وجود قابلیت هایی نظیر تغییرات آنی و مدیریت مرکزی، شبکه های مبتنی بر نرم افزار قادر تشخیص هرگونه اشکال و ناهماهنگی در سطح شبکه و رفع سریع آن با پیدا کردن مسیرهای جایگزین هستند.
- اختصاصی نبودن نرم افزار و سیستم عامل های شبکه (مبتنی بر استانداردهای آزاد و عدم وابستگی به فروشنده تجهیز شبکه): در شبکه های سنتی زمانی که از تجهیزات برندهای مختلف استفاده می شد، نرم افزار و سیستم عامل های آن نیز اختصاص به همان برند خاص داشت و این باعث مختلف شدن پیکربندی های یکسان

^۱ Cluster



شکل ۱-۲: نمایی از اجزاء تشکیل دهنده شبکه های مبتنی بر نرم افزار [۲]

در برندهای متفاوت می‌شد. با ظهور پدیده شبکه‌های مبتنی بر نرم‌افزار، این مرز بین برندها برداشته شده و تمام تجهیزات با زبانی مشترک قابل پیکربندی می‌باشد.

۱-۳-۳ اجزاء تشکیل دهنده معماری SDN

با توجه به شکل ۱-۲ یک شبکه مبتنی بر نرم‌افزار از اجزاء مختلفی تشکیل شده است که در ادامه به شرح وظایف هر یک از بخش‌ها می‌پردازیم.

۱-۳-۱ لایه زیرساخت

لایه زیرساخت^۱، مجموعه‌ای از تجهیزات شبکه مانند سوئیچ‌ها، مسیریاب‌ها و سرورها هستند که وظیفه هدایت ترافیک شبکه را عهده دار می‌باشند. این لایه در واقع لایه فیزیکی کنترل شده توسط کنترل کننده‌های SDN است.

۱-۳-۲ ارتباط جنوبی

ارتباط جنوبی^۲ که بین لایه زیرساخت و لایه کنترل قرار دارد یکی از مهم‌ترین بخش‌های معماری SDN می‌باشد. از جمله پروتکل‌ها در این ارتباط می‌توان به OpenFlow، Netconf و OVSDb اشاره کرد. ما در

^۱Infrastructure Layer

^۲SouthBound Interface (SBI)

این پروژه به بررسی اجمالی پروتکل OpenFlow و قابلیت‌های نسخه‌ی جدید آن می‌پردازیم.

۱-۳-۳ لایه کنترل

لایه کنترل^۱ در واقع هسته اصلی تصمیم‌گیری‌های شبکه و مغز متفکر آن می‌باشد. بخش اعظم فعالیت شرکت‌های تولید کننده راهکارهای شبکه مبتنی بر نرم‌افزار، اختصاص به ساخت و توسعه کنترل کننده‌ها و بسترهای نرم افزاری این لایه دارد. در این لایه، وظیفه مهم تصمیم‌گیری نحوه هدایت بسته‌ها، جمع‌آوری اطلاعات شبکه، وضعیت هر یک از بخش‌ها، جزئیات توپولوژی، وضعیت آماری بخش مختلف و غیره با برنامه ریزی لایه زیرساخت توسط ارتباط جنوبی انجام می‌شود.

۱-۳-۴ ارتباط شمالی

ارتباط شمالی^۲ که بین لایه کنترل و لایه برنامه کاربردی قرار دارد، وظیفه ایجاد بستر ارتباطی به منظور برنامه ریزی کنترل کننده را به عهده دارد. از جمله مهم‌ترین پروتکل‌های ارتباط شمالی می‌توان به REST API اشاره نمود.

۱-۳-۵ لایه برنامه کاربردی

لایه برنامه کاربردی^۳، محلی برای اجرای برنامه‌های کاربردی است. این برنامه‌ها با استفاده از اطلاعاتی که از لایه کنترل کننده به آن‌ها داده می‌شود اقدام به ایجاد تغییرات در شبکه و مسیرها می‌کنند. از نمونه‌های این برنامه‌ها می‌توان به اتوماسیون شبکه^۴، مدیر و پیکربندی شبکه^۵، پایش وضعیت شبکه^۶، عیب‌یابی شبکه^۷ و امنیت شبکه^۸ اشاره کرد.

۱-۴ نحوه عملکرد و تفاوت آن با شبکه‌های سنتی

با توجه به شکل ۱-۳، در شبکه‌های سنتی، هر تجهیز دارای بخش منطق و تصمیم‌گیری بوده و با دریافت اطلاعات هر یک از تجهیزات دیگر، اقدام به هدایت داده‌ها می‌کند اما در معماری مدرن SDN، هر یک از تجهیزات ابتدا بسته‌ها را به سمت کنترل به منظور تصمیم‌گیری هدایت می‌کنند، سپس کنترل کننده با توجه به قواعد تعیین شده و ایجاد تطابق با آن‌ها، یک جریان^۹ تشکیل داده و آن را به بخش هدایت داده ارسال می‌کند،

¹Control Layer

²NorthBound Interface (NBI)

³Application Layer

⁴Network Automation

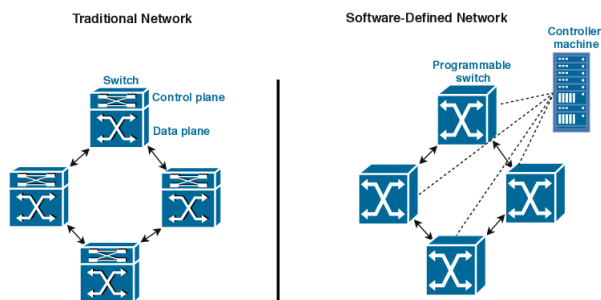
⁵Network Configuration and Management

⁶Network Monitoring

⁷Network Troubleshooting

⁸Network Security

⁹Flow



شکل ۱-۳: تفاوت شبکه مبتنی بر نرم‌افزار با شبکه سنتی

پس از آن هربار که تجهیز بسته مشابهی دریافت کرد آن را با توجه به جریان موجود در جدول جریان^۱ هدایت می‌کند.

۱-۵ ساختار گزارش

در این گزارش هدف، بررسی ویژگی‌های جدید پروتکل Openflow است. در فصل اول، به شرح معماری SDN و تفاوت‌های آن با معماری سنتی شبکه می‌پردازیم. در فصل دوم به شرح اجمالی پروتکل Openflow و اجزاء تشکیل دهنده آن بسنده خواهیم کرد سپس در فصل سوم، به توضیح ویژگی‌های افزوده شده در نسخه جدید OF1.5 خواهیم پرداخت. در فصل پایانی نیز سازگاری محصولات و نمایش نسخه جدید پروتکل در عمل را خواهیم داشت.

^۱Flow Table

فصل دوم

پروتکل Openflow

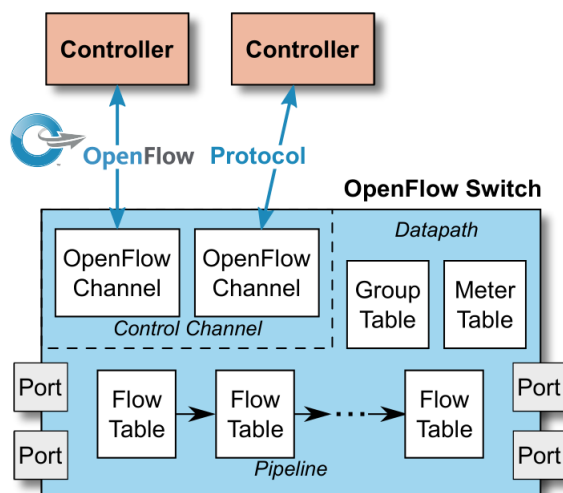
با توجه به توضیحات گفته شده در مقدمه، ارتباط کنترل کننده‌ها با بخش هدایت داده از طریق ارتباط جنوبی^۱ برقرار می‌گردد. برای برقراری این ارتباط پروتکل‌های مختلفی وجود دارد که Openflow (OF) یکی از پرکاربردترین آن‌ها است.

۱-۲ OPENFLOW

پروتکل (Openflow) که توسط بنیاد Open Networking Foundation (ONF) استاندارد سازی و توسعه داده می‌شود، یکی از مهم ترین پروتکل‌های موجود برای ارتباط کنترل کننده و سوئیچ‌ها بوده که از همان ابتدای پیدایش تحولات شبکه و حرکت به سمت تعریف بر اساس نرم‌افزار، به عنوان نمادی برای جداسازی بخش تصمیم گیری و بخش هدایت مورد توجه قرار گرفته است.

در طی سال‌های متمادی پس از عرضه و استاندارد سازی این پروتکل توسط ONF، تغییرات و به روز رسانی‌های بسیاری را تجربه کرده تا بتواند پاسخگوی نیازهای به‌روز شبکه باشد و امروز که در حال تهیه این گزارش هستیم نسخه OF1.5.1 آن توسط ONF تهیه و منتشر شده است. بسیاری از کمپانی‌های تولید کننده تجهیزات SDN

^۱ Southbound Interface



شکل ۲-۱: نمایی کلی از اجزاء اصلی یک سوئیچ Openflow [۱]

و کنترل کننده‌ها همچنان از نسخه‌های قدیمی تر استفاده می‌کنند اما در آینده نزدیک با پیشرفت سازگاری تجهیزات و کنترل کننده‌ها استفاده از نسخه‌های جدیدتر به صورت گسترده ممکن خواهد شد.

۲-۲ اجزاء تشکیل دهنده سوئیچ OPENFLOW

با توجه به شکل ۲-۱ یک سوئیچ Openflow منطبق بر نسخه OF1.5 از اجزاء مختلفی تشکیل شده است که در ادامه به معرفی هر بخش و کاربردهای آن می‌پردازیم [۱].

۱-۲-۲ OPENFLOW PORTS

در یک سوئیچ Openflow پکت‌ها از طریق درگاه‌های ورودی^۱ به خط لوله^۲ وارد می‌شوند و از طریق درگاه‌های خروجی^۳ خارج می‌شوند.

هر سوئیچ Openflow باید از نوع درگاه Physical Ports، Logical Ports و Reserved Ports پشتیبانی کند. به مجموعه درگاه‌های Physical، Logical و درگاه LOCAL از نوع درگاه‌های Reserved، درگاه‌های استاندارد گویند.

• Physical Ports: درگاه‌های فیزیکی، درواقع همان رابط‌های موجود در سخت افزار سوئیچ هستند. برای مثال، در یک سوئیچ اترنت^۴ هر درگاه فیزیکی به یکی از رابط‌های سخت افزاری نظیر می‌شود.

¹Ingress Ports

²Pipeline

³Output Ports

⁴Ethernet Switch

• Logical Ports: درگاه‌های منطقی، درگاه‌های تعریف شده توسط سوئیچ هستند که لزوماً به درگاه‌های فیزیکی نظیر نمی‌شوند. درگاه‌های منطقی مفهوم بالاتری از انتزاع هستند که برای تعریف متدهای خارج از پروتکل Openflow مانند tunnels، loopback interfaces و غیره استفاده می‌شوند.

• Reserved Ports: درگاه‌های رزرو شده که توسط پروتکل تعریف شده‌اند، به طور کلی به منظور توصیف اعمال هدایت بسته‌ها مانند ارسال به سمت کنترل کننده یا flooding یا هدایت به شیوه‌ای سنتی (non-Openflow) استفاده می‌شوند. هر سوئیچ Openflow لزوماً باید از درگاه‌های ALL، CONTROLLER، TABLE، IN_PORT، ANY و UNSET پشتیبانی کند و درگاه‌های LOCAL، NORMAL و FLOOD نیز به صورت اختیاری برای سوئیچ‌های Openflow قابل پشتیبانی می‌باشد. در ادامه به توضیح کاربرد هر یک از این درگاه‌های رزرو می‌پردازیم.

– ALL: نشان دهنده تمام درگاه‌های استاندارد است که سوئیچ قادر است از آن‌ها برای هدایت بسته‌ها استفاده کند. این درگاه فقط به صورت خروجی قابل تنظیم است و بسته‌هایی که به سمت آن هدایت می‌شوند پس از انجام پردازش‌های مرتبط با خروجی، از تمام درگاه‌های خروجی به جز درگاهی که بسته از آن آمده و درگاه‌هایی که با OFPPC_NO_FWD مشخص شده‌اند خارج می‌شود.

– CONTROLLER: به عنوان کانال ارتباطی بین سوئیچ و کنترل کننده استفاده می‌شود که می‌توان آن را به صورت ورودی یا خروجی تعریف نمود.

– TABLE: نشان دهنده آغاز خط لوله^۱ برای پردازش‌های مرتبط با Openflow است.

– IN_PORT: نشان دهنده درگاه ورودی بسته به سوئیچ می‌باشد.

– ANY: زمانی استفاده می‌شود که هیچ درگاهی در درخواست Openflow مشخص نشده باشد.

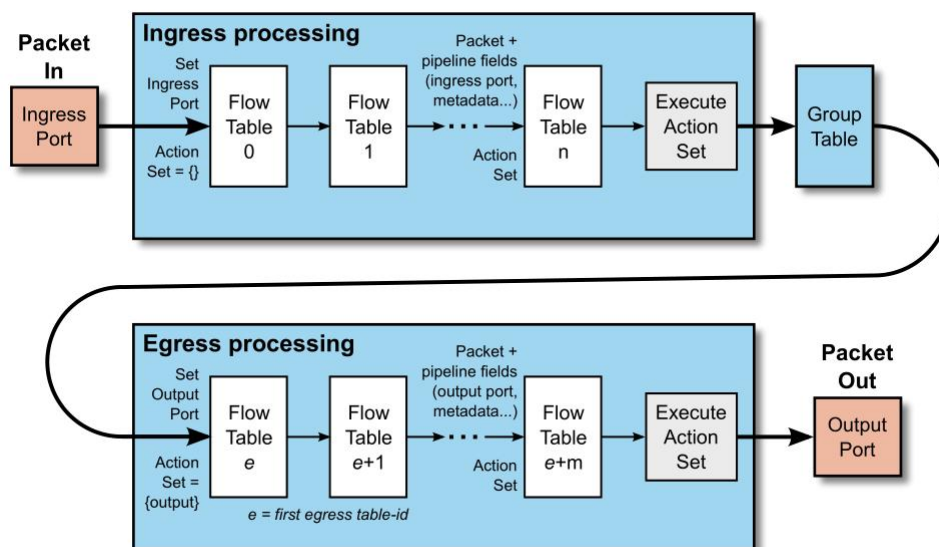
– UNSET: زمانی استفاده می‌شود که نخواهیم هیچ درگاهی در درخواست Openflow مشخص کنیم.

– LOCAL: نشان دهنده شبکه داخلی سوئیچ و بخش‌های مدیریت کننده تجهیز است که می‌توان از آن به عنوان هر دو حالت ورودی و خروجی استفاده کرد. این درگاه امکان ارتباط از خارج به سرویس‌های شبکه داخل سوئیچ Openflow را فراهم می‌کند.

– NORMAL: نشان دهنده هدایت توسط خط لوله non-OpenFlow یا همان هدایت توسط عملیات سوئیچ سنتی می‌باشد.

– FLOOD: نشان دهنده عمل Flooding در خط لوله non-Openflow می‌باشد.

^۱ pipeline



شکل ۲-۲: نمایی از جریان پردازش بسته‌ها در خط لوله [۱]

OPENFLOW PIPELINE ۲-۲-۲

پس از ورود بسته‌ها از طریق درگاه‌های معرفی شده، پردازش بر روی آن‌ها از طریق خط لوله انجام می‌گیرد. سوئیچ‌ها از لحاظ پشتیبانی Openflow به دو دسته سوئیچ‌های Openflow-only و Openflow-hybrid تقسیم می‌شوند.

در سوئیچ‌های نوع Openflow-only تمام بسته‌ها توسط خط لوله Openflow پردازش شده و به خروجی تحویل داده می‌شوند اما در سوئیچ‌های نوع Openflow-hybrid بسته‌ها می‌توانند از خط لوله Openflow یا خط لوله non-Openflow و یا هر دو به صورت سری پردازش شوند که نوع و ترتیب این پردازش‌ها با توجه به نوع پیکربندی و همچنین سازنده تجهیزات می‌تواند متفاوت باشد.

با توجه به شکل ۲-۲ هر خط لوله از اجزاء و مراحل پردازشی جداگانه تشکیل شده است که در ادامه به توضیح مختصری در باره آن‌ها می‌پردازیم.

- **Ingress Port:** این درگاه به منظور ایجاد بستری برای ورود اطلاعات به خط لوله پردازش می‌باشد که می‌تواند از یکی از انواع درگاه فیزیکی یا درگاه منطقی باشد.

- **Ingress Processing:** بخش اول پردازش بسته‌ها پس از ورود به خط لوله در اینجا انجام می‌گیرد. به هر بسته ورودی یک مجموعه تهی **Action-Set** نسبت داده می‌شود. این مجموعه در طی عبور بسته از جداول جریان توسط عملیات‌های مورد نیاز برای بسته پر شده و در انتهای پردازش، بخش **Execute Action-Set** عملیات‌ها را به صورت دسته‌ای و به ترتیب اولویت بر روی بسته اعمال می‌کند. در ارتباط

با جداول جریان^۱ و مدخل‌های ورودی^۲ در بخش بعدی به طور کامل توضیح خواهیم داد.

- **Group Table**: یکی از عناصر مهم خط لوله پردازش می‌باشد که از نسخه‌ی OF1.1 به استاندارد اضافه شده است. این جدول امکان ایجاد عملیات‌های پیچیده و خاص را بر روی بسته‌ها فراهم می‌کند و همچنین در مواردی باعث کاهش مقدار پردازش در بخش‌های قبلی و بعدی می‌شود. در بخش‌های آینده به صورت کامل کاربرد و نحوه عملکرد این واحد را شرح خواهیم داد.

- **Egress Processing**: همانطور که در شکل ۲-۲ قابل مشاهده است، این مرحله از پردازش توسط جدول جریان در مرحله آخر و قبل از ارسال بسته به خروجی قرار دارد که امکانات ویژه‌ای جهت ایجاد سناریوهای خاص فراهم می‌آورد. این ویژگی از نسخه OF1.5 به استاندارد اضافه شده که در فصل آینده در مورد ویژگی‌ها و تفاوت‌های آن با Ingress processing به صورت کامل توضیح خواهیم داد.

- **Output Port**: این درگاه به منظور ایجاد بستری برای خروج بسته‌های پردازش شده توسط خط لوله می‌باشد که می‌تواند یکی از انواع درگاه‌های تعریف شده در استاندارد برای خروجی در نظر گرفته شود.

۳-۲-۲ FLOW TABLES

با توجه به شکل ۲-۲ هر یک از مراحل Ingress/Egress Processing دارای چندین جدول جریان می‌باشند. هر جدول جریان نیز دارای یک یا چند مدخل جریان است. هر مدخل جریان نیز به صورت زیر تعریف می‌گردد:

Match Fields	Priority	Counters	Instructions	Timeouts	Cookie	Flags
--------------	----------	----------	--------------	----------	--------	-------

جدول ۲-۱: اجزاء اصلی یک مدخل جریان [۱]

- **Match Fields**: برای ایجاد تطابق با بسته‌ها کاربرد دارد که از قسمت‌های درگاه ورودی^۳، سرآیندهای بسته‌ها^۴ و به صورت اختیاری پارامترهای خط لوله^۵ مانند فراداده‌های^۶ جدول‌های پیشین تشکیل شده است.

- **Priority**: نشان دهنده اولویت بسته‌ها به منظور انجام تطابق می‌باشد.

^۱Flow Table

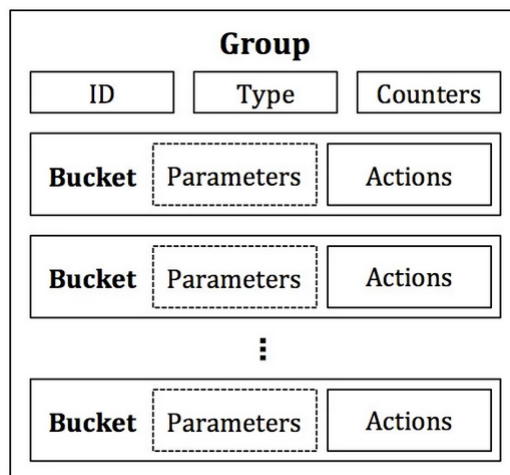
^۲Flow Entry

^۳Ingress Port

^۴Packet Headers

^۵Pipeline Fields

^۶Metadata



شکل ۲-۳: نمایی از شکل کلی جدول گروه [۱]

- **Counters:** با انجام شدن عمل تطابق شمارنده‌ها افزایش می‌یابند تا داده‌های آماری مربوط به جدول جریان را کامل کنند.
- **Instructions:** به منظور ایجاد تغییر در مجموعه Action-Set از این پارامتر در مدخل جریان می‌توان استفاده کرد.
- **Timeouts:** حداکثر زمان و مدت زمان Idle که یک مدخل جریان پس از آن باطل می‌شود در این قسمت قابل برنامه ریزی می‌باشد.
- **Cookie:** این پارامتر را کنترل کننده‌ها به منظور عملیات داخلی استفاده می‌کنند.
- **Flags:** توسط این پارامترها می‌توان نحوه مدیریت مدخل‌های جریان را در جدول جریان تغییر داد.

باید به این نکته توجه داشت که همه کنترل کننده‌ها و همه سوئیچ‌ها قابلیت پشتیبانی از همه عملیات‌های گفته شده در بالا را ندارند که این قابلیت‌ها بسته به پیاده سازی برندهای مختلف از پروتکل متفاوت می‌باشد.

TABLE-MISS ۴-۲-۲

در هر جدول جریان، در صورتی که عمل تطابق در آن صورت نگیرد، مدخل جریانی به نام Table-miss در آن وجود دارد که وضعیت بسته را برای ادامه در خط لوله مشخص می‌کند. برای مثال می‌توان بسته را دور انداخت یا به جدول جریان بعدی و یا به کنترل کننده ارسال کرد.

GROUP TABLE ۵-۲-۲

همان طور که در بخش قبلی نیز بیان شد، جدول گروه، انتزاعی به منظور پیاده سازی عملیات‌های پیچیده و خاص است که قابلیت پیاده سازی آسان با استفاده از جداول جریان و مدخل‌های آن‌ها را ندارند. هر گروه در این جدول بسته‌ها را با عنوان ورودی دریافت کرده و عملیات‌های به خصوصی را روی آن‌ها انجام می‌دهند. هر گروه دارای دسته‌ای از عملیات‌ها است که سطل^۱ نامیده می‌شوند. همان طور که در شکل ۲-۳ مشاهده می‌شود، گروه می‌تواند نوع^۲ مختلفی داشته باشد که در ادامه به آن‌ها اشاره و کاربرد آن‌ها را بررسی می‌کنیم:

• ALL: در این حالت، بسته‌های ورودی به صورت یکسان به تمام سطل‌ها وارد شده و سری عملیات^۳ بخصوصی به صورت مستقل بر روی آن‌ها انجام می‌گیرد. می‌توان از این ویژگی به منظور ایجاد، پردازش و ارسال داده‌ها به صورت Broadcast و یا Multicast استفاده کرد.

• INDIRECT: در این گروه به خصوص فقط یک سطل وجود دارد و تمام بسته‌های ورودی به گروه به آن وارد می‌شوند. کاربرد مهم این گروه، ایجاد قابلیت جمع‌آوری عملیات تکراری در جداول جریان و اجراء آن‌ها در یک مرتبه به منظور کاهش سربار پردازشی و حافظه‌ای است.

• SELECT: در این گروه، سطل‌ها به صورت وزن دار اولویت برای دریافت بسته دارند. کاربرد اصلی این گروه به منظور ایجاد تراز بار^۴ در شبکه است و واضح‌ترین الگوریتم برای ایجاد توازن در این گروه Round-Robin می‌باشد اما از الگوریتم‌های پیچیده‌تر و بهینه‌تری نیز می‌توان استفاده کرد..

METER TABLE ۶-۲-۲

در نسخه OF1.3، مفهومی به نام اندازه‌گیرها^۵ به استاندارد Openflow افزوده شد. با ترکیب این اندازه‌گیرها و بستر صف‌های خروجی می‌توان به صورت کلی نرخ دریافت و ارسال بسته از سوئیچ را اندازه‌گیری و کنترل کرد. جداول جریان قادر به ارسال بسته‌ها به سمت جداول اندازه‌گیر هستند که این خود زمینه ایجاد کنترل کیفیت سرویس^۶ را فراهم می‌آورد.

OPENFLOW CHANNEL ۷-۲-۲

کانال Openflow بستری است که اتصال بین سوئیچ منطقی را با کنترل کننده فراهم می‌کند. از طریق این بستر، کنترل کننده قادر به پیکربندی و مدیریت سوئیچ، دریافت رخدادها از سوئیچ و ارسال بسته‌ها خارج از

¹Bucket²Type³Actions⁴Load Balancing⁵Meters⁶Quality of Service (QoS)

سوئیچ می‌باشد. کانال کنترلی یک سوئیچ ممکن است از یک ارتباط کانال Openflow به یک کنترل کننده پشتیبانی کند و یا قادر به پشتیبانی از ارتباط چندگانه کانال Openflow به چند کنترل کننده باشد. نوع ارتباط بین کنترل کننده و لایه هدایت داده به نحوه پیاده سازی آن بستگی دارد اما قالب داده‌های رد و بدل شده بین این دو باید طبق استاندارد Openflow باشد. کانال Openflow می‌تواند بر بستر رمزنگاری شده TLS انجام گیرد یا به صورت مستقیم از TCP استفاده کند. پیش فرض درگاه ارتباطی برای این اتصال در نسخه‌های اولیه TCP:6633 یا TCP:976 بوده است اما با عرضه نسخه OF1.4 این درگاه توسط IANA به TCP:6653 تغییر کرد.

در بستر ارتباط فوق الذکر، جابه‌جایی داده‌ها بین کنترل کننده و سوئیچ به دو دسته زیر تقسیم می‌گردد:

- کنترل کننده به سوئیچ^۱: این پیام‌ها توسط کنترل کننده ارسال شده و ممکن است نیاز به جواب از سمت سوئیچ را داشته باشند. لیست انواع مختلف این نوع پیام به شرح زیر است:

– Features: به منظور شناسایی سوئیچ و قابلیت‌های آن توسط کنترل کننده ارسال می‌شود و سوئیچ باید به آن پاسخ دهد.

– Configuration: کنترل کننده می‌تواند پیکربندی سوئیچ را توسط این پیام بخواند^۲ یا تنظیم^۳ کند. سوئیچ باید در جواب درخواست خواندن پیکربندی پاسخ دهد.

– Modify-State: هدف از این پیام مدیریت وضعیت سوئیچ می‌باشد. کنترل کننده توسط این پیام می‌تواند مدخل جریان یا گروه اضافه، حذف یا ویرایش کند، یا سطل‌های عملیات را در سوئیچ درج یا حذف کند، یا وضعیت درگاه‌های سوئیچ و ویژگی‌های مربوط به آن‌ها را تغییر دهد.

– Read-State: این پیام‌ها به منظور جمع‌آوری طیف وسیعی از اطلاعات سوئیچ توسط کنترل کننده ارسال می‌شوند. از جمله این اطلاعات می‌توان به پیکربندی، آماره‌ها و قابلیت‌ها اشاره کرد.

– Packet-out: این دسته از پیام‌ها، باعث ارسال یک بسته از یکی درگاه‌های سوئیچ می‌شوند. این پیام‌ها می‌توانند خود شامل یک بسته کامل یا حاوی آدرس بسته ذخیره شده در سوئیچ باشند که برای ارسال آن درخواست ارسال می‌کنند

– Barrier: این پیام‌ها به منظور آگاه کردن طرفین از انجام کامل عملیات یا اطمینان از کامل شدن پیشنهادها می‌شوند.

¹ Controller-to-Switch

² Query

³ Set

– Role-Request: این دسته از پیام‌ها که بیشتر در محیط‌های چند کنترل کننده‌ای استفاده می‌شوند، برای تعیین نقش کنترل کننده و تنظیم کد شناسایی کنترل کننده می‌باشد.

– Asynchronous-Configuration: به منظور اجرای فیلتر در پیام‌های غیرهمزمان ارسال شده از سوئیچ کاربرد دارد.

• غیرهمزمان^۱: این پیام‌ها توسط سوئیچ و بدون فرمان از سمت کنترل کننده ارسال می‌شوند که بیانگر تغییر وضعیت سوئیچ یا رسیدن بسته می‌باشد. لیست انواع مختلف این نوع پیام به شرح زیر است:

– Packet-in: درخواست از کنترل کننده به منظور تعیین وضعیت بسته ورودی می‌باشد. تمام بسته‌هایی که با درگاه کنترل کننده^۲ برچسب می‌خورند توسط این پیام به کنترل کننده ارسال می‌گردند.

– Flow-Removed: مطلع کردن کنترل کننده از حذف یک مدخل جریان از جدول جریان توسط این پیام انجام می‌گیرد.

– Port-status: مطلع کردن کنترل کننده از تغییرات درگاه‌های موجود در سوئیچ توسط این پیام انجام می‌گیرد.

– Role-status: زمانی که کنترل کننده نقش خود را به عنوان کنترل کننده اصلی انتخاب می‌کند، انتظار می‌رود سوئیچ توسط این نوع پیام به کنترل کننده اصلی قبلی وضعیت جدید نقش‌ها را اطلاع رسانی کند

– Controller-Status: توسط این پورت سوئیچ وضعیت کانال Openflow را به تمام کنترل کننده‌ها گزارش می‌دهد.

– Flow-monitor: کنترل کننده می‌تواند در هر یک جداول جریان دنبال کننده‌هایی تنظیم کند تا در صورت تغییر وضعیت جدول، سوئیچ توسط این نوع پیام، کنترل کننده را آگاه سازد.

• متقارن^۳: این پیام‌ها بدون درخواست هر یک از طرفین جابه‌جا می‌شوند.

– Hello: در هنگام آغاز اتصال توسط کانال Openflow این پیام رد و بدل می‌شود.

– Echo: این پیام‌ها که به هر دو صورت درخواست و پاسخ می‌باشند از هر دو بخش کنترل کننده و سوئیچ قابل ارسال می‌باشند و لزوماً باید پاسخ دریافت کنند. هدف اصلی این پیام، بررسی برقراری

¹ Asynchronous

² CONTROLLER Port

³ Symmetric

ارتباط کنترل کننده-سوئیچ می‌باشد اما به منظور اندازه گیری تاخیر و پهنای باند این اتصال نیز استفاده می‌شود.

Error - این پیام به منظور مطلع کردن هر یک از طرفین، از بروز خطا می‌باشد.

Experimenter - این دسته از پیام‌ها، راهی استاندارد برای تجهیزات و کنترل کننده‌های Openflow می‌باشد تا بتوانند ویژگی‌های جدید را به منظور استفاده در شبکه به طرفین پیشنهاد دهند.

۲-۳ بررسی وظایف مهم در خط لوله

در طول خط لوله، سه وظیفه اصلی و مهم انجام می‌گیرد که در اینجا به بررسی آن‌ها می‌پردازیم.

۲-۳-۱ MATCH

در هنگام دریافت بسته، سوئیچ توابع به خصوصی را به منظور انجام پردازش در خط لوله اجرا می‌کند. در ابتدا سوئیچ فیلدهای بسته را به منظور جستجو و انجام عمل تطابق در جداول جریان جداسازی می‌کند. این فیلدها می‌تواند شامل فیلدهای سرآیندهای مختلف مانند Ethernet و یا TCP/IP باشد. علاوه بر سرآیندهای بسته، تطابق می‌تواند برای پورت ورودی، فراداده‌ها و دیگر فیلدهای خط لوله نیز انجام پذیرد. فراداده‌ها ممکن است به منظور انتقال اطلاعات بین جداول سوئیچ استفاده شود. تمام فیلدهای سرآیندها و فیلدهای خط لوله نمایانگر وضعیت فعلی بسته هستند. در صورتی که بسته توسط جدول‌های قبلی دچار تغییر شده باشد، تطابق بر روی مقادیر جدید انجام می‌گیرد.

یک بسته هنگامی با یکی از مدخل‌های جریان تطابق می‌یابد که تمام فیلدهای سرآیند و خط لوله آن مطابقت داشته باشد. در صورتی که یک بسته به چند مدخل جریان مطابقت داشت، مدخل جریانی که اولویت^۱ بالاتری داشته باشد عملیات را روی بسته انجام خواهد داد.

۲-۳-۲ INSTRUCTION

هر مدخل جریان دارای یک سری از دستورات است و زمانی که بسته‌ای با آن مدخل جریان تطابق یافت، دستورات روی بسته اجرا می‌گردند. هر دستور می‌تواند در مجموعه عملیات^۲ و یا در پردازش خط لوله بسته تغییر ایجاد کند. نمونه‌ای از دستورات به شرح زیر می‌باشد:

- Apply-Actions: پیاده سازی آنی تغییرات روی بسته بدون تغییر در مجموعه عملیات بسته

¹Priority

²Action-Set

- Clear-Actions: پاک کردن تمام عملیات‌های موجود در مجموعه عملیات
- Write-Actions: اضافه کردن عملیات به مجموعه عملیات بسته
- Write-Metadata: اضافه کردن فرداده به فیلد مربوطه در بسته
- Stat-Trigger: در صورتی که آمار مرتبط با یکی از جریان‌ها به حدود تعریف شده رسیده باشد به کنترل کننده یک رخداد ارسال می‌کند.
- Goto-Table: نشان دهنده جدول بعدی به منظور ادامه پردازش خط لوله می‌باشد.

Action

- همان طور که در بالا اشاره شد، هر دستور یک عملیات روی بسته انجام می‌دهد یا یک عملیات در مجموعه عملیات بسته وارد می‌کند. از مجموعه عملیات‌های ممکن روی بسته می‌توان به موارد زیر اشاره کرد:
- *Output port_no*: باعث هدایت بسته به درگاه مورد نظر به منظور انجام پردازش خروجی و خارج شدن بسته از سوئیچ می‌باشد.
 - *Group group_id*: مشخص کردن گروه به منظور هدایت بسته به آن و انجام پردازش‌های مرتبط با گروه.
 - Drop: بسته‌هایی که هیچ گونه عملیات مشخصی برای انجام نداشته باشند در نهایت رها می‌شوند.
 - *Set-Queue queue_id*: به منظور مشخص کردن صف خروجی بسته می‌باشد.
 - *Meter meter_no*: بسته‌هایی که باید به سمت اندازه‌گیرها هدایت شوند توسط این عملیات مشخص می‌گردند.
 - *Push-Tag/Pop-Tag ethertype*: به منظور تغییر نوع بسته بکار می‌رود. برای مثال می‌توان نوع سرآیند بسته را برای پیاده سازی MPLS تغییر داد.
 - *Set-Field field_type value*: توسط این عملیات می‌توان فیلدهای مختلف موجود در سرآیند بسته‌ها را تغییر داد.
 - *Copy-Field src_field_type dst_field_type*: توسط این عملیات می‌توان فیلدهای مختلف موجود در سرآیند یک بسته را به سرآیند بسته دیگر رونوشت کرد.
 - *Change-TTL ttl*: توسط این عملیات می‌توان مقدار TTL بسته را ویرایش کرد.

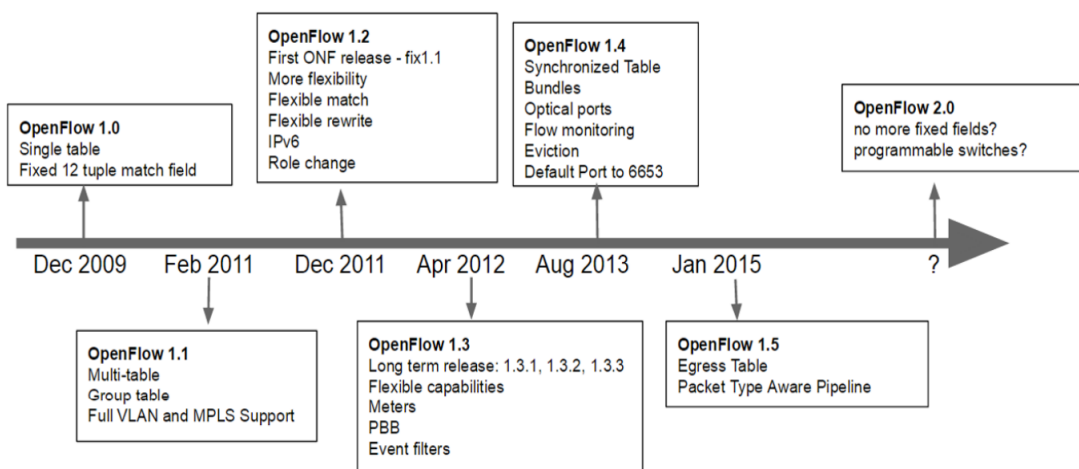
شمارنده‌ها برای هر جدول جریان، مدخل جریان، درگاه، صف، گروه، سطل گروه و اندازه‌گیر وجود دارند. در هر بخش، شمارنده‌ها، مولفه‌های مرتبط با آن بخش را شمارش کرده و به صورت داده‌های آماری به کنترل کننده ارسال می‌کنند. تمامی شمارنده‌های موجود به صورت بدون علامت بوده و نمایشگر سرریز^۱ ندارند. نمونه‌هایی از شمارنده‌های موجود برای یک مدخل جریان عبارت‌اند از: تعداد بسته‌های دریافت شده، تعداد بایت‌های دریافت شده، مدت زمان وجود مدخل بر حسب ثانیه.

^۱Overflow

فصل سوم

ویژگی‌های نسخه جدید Openflow

همانگونه که در فصل قبل گفته شد، پروتکل Openflow همواره در حال عرضه نسخه‌های جدید می‌باشد تا بتواند پاسخگوی نیازها به روز و پویای شبکه و همچنین رفع مشکلات نسخه‌های پیشین باشد. در شکل ۱-۳ تغییرات عمده و مهم این پروتکل در بازه زمانی عرضه آن با نسخه OF1.0 تا هم اکنون (سال ۲۰۲۱ میلادی) که نسخه OF1.5.1 به صورت استاندارد قابل دسترس است را مشاهده می‌کنیم.



شکل ۱-۳: روند پیشرفت پروتکل در نسخه‌های مختلف [۵]

۳-۱ ویژگی‌های نسخه OF1.5

ویژگی‌های این نسخه را می‌توان در سه دسته، طبقه‌بندی کرد: اضافه شده‌ها، بهبود یافته‌ها و تغییرات. اضافه شده‌ها، ویژگی‌های کاملاً جدیدی هستند که به پروتکل اضافه شده‌اند؛ بهبود یافته‌ها به منظور کامل کردن بخش‌های موجود در پروتکل اضافه شده‌اند و تغییرات نیز بخش‌هایی از پروتکل هستند که به صورت کامل دستخوش تغییر شده‌اند. برخی از مهم‌ترین ویژگی‌های این نسخه عبارت‌اند از (تعدادی از ویژگی‌ها به دلیل جزئی بودن ذکر نشده است):

آ اضافه شده‌ها

- جداول جریان خروجی
- مهم بودن نوع بسته در خط لوله
- آماره‌های قابل انعطاف مرتبط با مدخل جریان (OXS)
- ارسال خودکار آماره‌های مدخل جریان
- عملیات Copy-Field
- تطابق TCP flags
- وضعیت ارتباط کنترل کننده

ب بهبود یافته‌ها

- دستورات تکاملی گروه برای تغییرات انتخابی در سطرها
- اجازه استفاده از wildcard در عملیات set-field
- دسته‌های زمان‌بندی شده

ج تغییرات

- تغییر Meter instruction به Meter action

در ادامه گزارش به شرح هر یک از تغییرات و کاربردهای آن‌ها می‌پردازیم.

۳-۱-۱ جداول جریان خروجی

در نسخه‌های قبلی تمام پردازش روی بسته‌ها در حیطه‌ی درگاه ورودی انجام می‌شد. اما در نسخه OF1.5 با معرفی جداول جریان خروجی^۱، قابلیت پردازش بسته‌ها در حیطه‌ی درگاه خروجی نیز فراهم می‌گردد. زمانی که بسته برای خروجی به درگاه مربوط ارجاع داده می‌شود، پردازش این بخش در اولین جدول جریان خروجی آغاز می‌شود که طی آن عملیات‌های مورد نظر روی بسته انجام شده یا به جداول دیگر خروجی ارجاع داده می‌شود. از ویژگی‌های این مرحله از پردازش می‌توان به موارد زیر اشاره کرد:

- تمام پردازش‌های مرتبط با پورت‌های منطقی، جداول جریان ورودی و جدول گروه قبل از این بخش انجام می‌شوند.
 - تعریف رفتار جدول جریان خروجی و مدخل‌های جریان آن بسیار شبیه به ورودی است.
 - قابلیت تغییر درگاه خروجی در این جداول وجود ندارد.
 - تمام فراداده‌های مربوط به خط لوله از پردازش ورودی به پردازش خروجی منتقل می‌شوند.
 - مجموعه عملیات پردازش خروجی در ابتدا با عمل خروجی^۲ پر شده است که در برابر مجموعه عملیات ورودی که تهی می‌باشد متفاوت است.
- استفاده از پردازش قبل از خروجی از این جهت حائز اهمیت می‌باشد که در این مرحله از پردازش، درگاه خروجی و وضعیت نهایی بسته طی مراحل قبلی مشخص شده است و می‌توان عملیات‌های این بخش را با توجه به شرایط جدید انجام داد که این خود آزادی عمل بیشتری در تطابق و ایجاد تغییر در بسته‌ها را به ارمغان می‌آورد.

۳-۱-۲ مهم بودن نوع بسته در خط لوله

در نسخه‌های قبلی، تمام بسته‌ها می‌بایست از نوع Ethernet می‌بودند اما با معرفی نسخه OF1.5 قابلیت پردازش روی انواع مختلف بسته‌ها مانند بسته‌های IP یا بسته‌های PPP نیز فراهم گردید. فیلدی در خط لوله به منظور تطابق نوع بسته اضافه شده است که به عنوان پیش‌نیاز برای انجام تطابق فیلدهای سرآیند می‌باشد. اضافه شدن این قابلیت، انعطاف تطابق را بالا برده و امکان تطبیق انواع وسیعی از بسته‌ها را در اختیار قرار می‌دهد که برای مثال توسط آن می‌توان برای انواع مختلف بسته‌ها کیفیت سرویس تنظیم کرد یا سیاست‌های امنیتی گسترده وضع نمود.

¹Egress Flow Tables²Output action

۳-۱-۳ آماره‌های قابل انعطاف مرتبط با مدخل جریان (OXS)

در نسخه‌های قبلی، یک ساختار ثابت برای ذخیره آماره‌های مدخل جریان^۱ در نظر گرفته شده بود که در نسخه جدید با ساختارهای انعطاف پذیر جایگزین شده است. با این تغییر، آماره‌های ذکر شده قابلیت جمع‌آوری و ذخیره داده‌های آماری دلخواه را خواهد داشت که امکان پایش و گزارش دقیق‌تر تغییرات و وضعیت شبکه را به پروتکل می‌دهد.

۴-۱-۳ ارسال خودکار آماره‌های مدخل جریان

در نسخه‌های قبلی، کنترل کننده همواره به منظور دریافت آمار مدخل‌های جریان از سوئیچ نمونه‌برداری^۲ داده به عمل می‌آورد که باعث ایجاد سربار پردازشی در سوئیچ می‌گردید. در نسخه جدید، آماره‌های فوق الذکر، به صورت خودکار در صورت عبور از آستانه‌های تعریف شده به کنترل کننده گزارش داده می‌شوند.

۵-۱-۳ عملیات COPY-FIELD

عملیات Set-Field موجود در پروتکل امکان تنظیم یک مقدار ثابت را در فیلدهای سرآیند و خط لوله فراهم می‌کند. عملیات جدید Copy-Field امکان رونوشت فیلدهای سرآیند و خط لوله را از بسته‌ای به بسته دیگر فراهم می‌کند که این خود باعث کاهش تعداد عملیات انجام شده و کاهش سربار پردازشی می‌گردد.

۶-۱-۳ تطابق TCP FLAGS

قابلیت تطابق با پرچم‌های موجود در سرآیند بسته‌های TCP در این نسخه اضافه شده است. این قابلیت، انعطاف انجام تطابق را برای پروتکل افزایش می‌دهد و از آن می‌توان به منظور پیاده‌سازی سیاست‌های مربوط به درخواست‌ها یا وضع سیاست‌های امنیتی جهت جلوگیری از انواع حمله استفاده کرد.

۷-۱-۳ وضعیت ارتباط کنترل کننده

وضعیت ارتباط کنترل کننده، این امکان را به کنترل کننده می‌دهد تا درباره تمام ارتباطات سوئیچ با کنترل کننده‌های دیگر مطلع شود. این ویژگی به همراه قابلیت چند کنترل کننده بودن سوئیچ اجازه قسمت بندی شبکه کنترل کننده‌ها و پایش وضعیت کنترل کننده‌های دیگر را به ارمغان می‌آورد. هر کنترل کننده خود را به یک شناسه کوتاه تشخیص می‌دهد و کنترل کننده‌های دیگر را توسط آدرس URI از یکدیگر متمایز می‌کند.

^۱Flow Entry Statistics

^۲Poll

۳-۱-۸ دستورات تکاملی گروه برای تغییرات انتخابی در سطل‌ها

در نسخه‌های قبلی، به منظور ایجاد تغییر در سطل‌های موجود در گروه، تمام مجموعه سطل‌ها تحت تاثیر قرار می‌گرفت. دو دستور درج کردن^۱ به منظور اضافه کردن سطل به گروه و حذف کردن^۲ به منظور حذف سطل از گروه به مجموعه دستورات گروه اضافه گردید که باعث کاهش عملیات و سربار پردازش در انجام عملیات مرتبط با گروه می‌شود.

۳-۱-۹ اجازه استفاده از WILDCARD در عملیات SET-FIELD

در نسخه‌های قبلی، به منظور تغییر فیلد با استفاده از عملیات set-field تمام بخش‌های فیلد مورد نظر باید تنظیم می‌شد اما به اضافه شدن قابل wildcard می‌توان بخش‌های بدون تغییر فیلد را پوشاند تا نیازی به تنظیم دوباره آن نباشد.

۳-۱-۱۰ دسته‌های زمان‌بندی شده

در نسخه OF1.4، ویژگی جدیدی به نام Bundles معرفی شد که طبق آن گروهی از پیام‌های Openflow طی یک عملیات اجراء می‌شوند. این قابلیت به کنترل کننده‌ها امکان ایجاد همزمان دستورات مرتبط را می‌دهد که باعث موجب هماهنگی و همزمانی بین سری سوئیچ‌ها می‌گردد. در این دسته از دستورات، در صورت بروز خطا در یک دستور، تمام دسته با شکست مواجه می‌شود که این عامل اصلی ایجاد هماهنگی بین سوئیچ‌ها است. حال در نسخه OF1.5، ویژگی دسته‌های زمان‌بندی شده^۳ این امکان را فراهم می‌آورد که بتوان برای اجراء دستورات موجود در دسته، زمان مشخصی تنظیم کرد.

۳-۱-۱۱ تغییر METER INSTRUCTION به METER ACTION

در نسخه‌های قبلی، عمل اندازه‌گیری توسط یک دستور اندازه‌گیری انجام می‌گرفت اما در نسخه جدید این امکان توسط عملیات اندازه‌گیری صورت می‌پذیرد که این تغییر، قابلیت اضافه کردن چندین اندازه‌گیر به مدخل جریان و قابلیت اضافه کردن اندازه‌گیر به گروه‌ها را به پروتکل می‌دهد. کاربرد این قابلیت در اعمال سیاست‌های مرتبط با اندازه‌گیری‌های چندگانه بر روی ترافیک ورودی سوئیچ می‌باشد که انعطاف بالاتری را به شبکه می‌افزاید.

¹Insert

²Remove

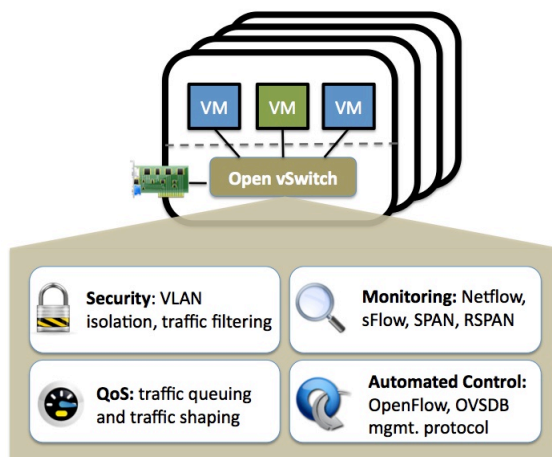
³Scheduled Bundles

فصل چهارم

بلوغ محصولات

پروتکل Openflow در مسیر پیشرفت، با مشکلات بزرگی مواجه می‌شود. یکی از مشکلات این است که شرکت‌های تولید کننده نرم‌افزار و سخت‌افزار به دلیل وسعت دامنه تغییرات برای پشتیبانی، نمی‌توانند با سرعت پیشرفت پروتکل هماهنگ شوند. و مشکل دوم نیز از آنجایی نشئت می‌گیرد که پروتکل Openflow تلاش دارد تا اکوسیستمی برای شبکه فراهم آورد که از تولید کننده سخت‌افزار مجزا است و در آن تجهیزات از شرکت‌های تولید کننده مختلف می‌توانند تحت یک کنترل کننده عمل کنند اما پس از عرضه پروتکل مشاهده می‌شود که حتی نسخه یکسانی از پروتکل که توسط چند شرکت متفاوت پیاده سازی شده‌اند نیز دارای تفاوت‌هایی در عملکرد و پیکربندی می‌باشد.

در زمان نگارش این گزارش، نسخه OF1.3 به صورت گسترده در تجهیزات و کنترل کننده‌های عملیاتی در حال استفاده است. نسخه OF1.4 در حال افزایش پشتیبانی و تعداد استفاده تجهیزات از آن است که در آینده نزدیک شاهد جایگزینی کامل OF1.3 با این نسخه خواهیم بود. اما با تغییرات عمده صورت گرفته در نسخه OF1.5 به ویژه تغییرات در خط لوله و اضافه شدن مرحله پردازشی خروجی، انتظار می‌رود روند همه گیر شدن استفاده از این نسخه به کندی صورت پذیرد. در حال حاضر سوئیچ‌های نرم‌افزاری مانند OVS و برخی از برندهای سخت افزار از این پروتکل پشتیبانی می‌کنند و تنها کنترل کننده که از این نسخه پشتیبانی می‌کند، Ryu SDN Controller



شکل ۴-۱: نمایی از قابلیت‌های سوئیچ OvS [۶]

می‌باشد. در ادامه به معرفی دو محصول پشتیبانی کننده از نسخه OF1.5 یعنی سوئیچ OvS و کنترل کننده ryu می‌پردازیم.

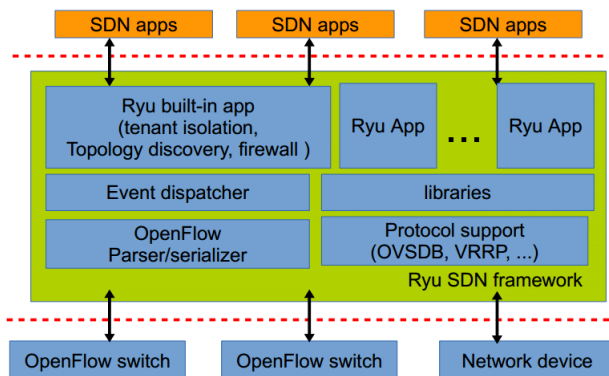
۱-۴ OPEN vSWITCH (OVS)

این سوئیچ در واقع پیاده سازی نرم افزاری منبع باز سوئیچ چندلایه شبکه است که توسعه‌ی آن توسط بنیاد آزاد لینوکس مدیریت می‌شود و تحت گواهی Apache 2.0 به صورت رایگان در دسترس می‌باشد. طراحی آن به گونه‌ای است که قابلیت ایجاد برنامه ریزی و خودکارسازی شبکه را در سطح گسترده و آماده برای بهره برداری فراهم می‌آورد. از قابلیت‌های مهم آن می‌توان به پشتیبانی از سیستم عامل‌های مرسوم مانند لینوکس^۱ و پلتفرم‌های مجازی سازی و شبکه مانند VMware و Cisco اشاره کرد. آخرین نسخه پایدار منتشر شده از این سوئیچ، نسخه v2.14 می‌باشد که قابلیت پشتیبانی از OF1.5 را به صورت پیش فرض دارد.

۲-۴ RYU SDN CONTROLLER

ری-هو (Ryu)، یک چهارچوب مولفه محور به منظور توسعه شبکه‌های مبتنی بر نرم افزار می‌باشد که قادر به تولید نرم افزارهایی برای کنترل و مدیریت شبکه‌ها است. این کنترل کننده از پروتکل‌هایی مانند Openflow، Netconf، OF-conf و غیره پشتیبانی می‌نماید. این نرم افزار به صورت آزاد و منبع باز تحت گواهی Apache 2.0 قابل دسترسی است.

^۱Linux

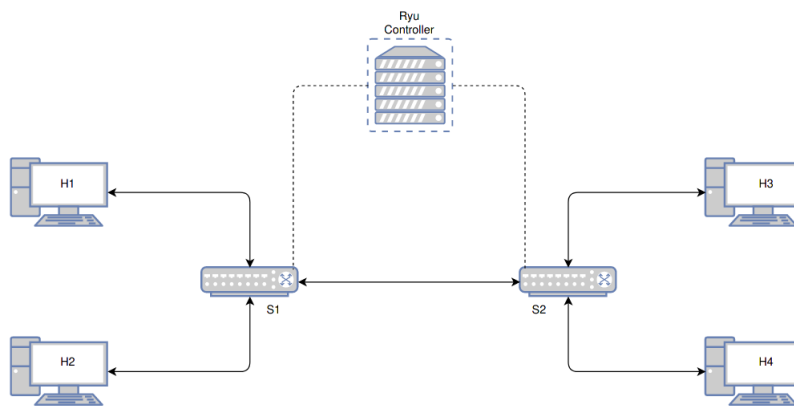


شکل ۴-۲: نمایی از اجزاء تشکیل دهنده چهارچوب نرم افزاری Ryu

توسعه در این چهارچوب توسط رابط نرم افزاری^۱ با زبان پایتون^۲ انجام می پذیرد. شکل ۴-۲ نشان دهنده اجزاء تشکیل دهنده این چهارچوب و محل قرارگیری برنامه های توسعه یافته توسط کاربر را نشان می دهد [۷].

۴-۳ آزمایش پشتیبانی سوئیچ از نسخه OF1.5

در این قسمت به بررسی پشتیبانی سوئیچ OvS نسخه v2.14.2 از پروتکل Openflow نسخه OF1.5 توسط کنترل کننده Ryu نسخه v4.34 می پردازیم. در حال حاضر تنها تجهیزات و کنترل کننده های پشتیبانی کننده از نسخه نهایی OF1.5، نرم افزارهای فوق الذکر می باشند که حتی پلتفرم های آزمایش پیروی از نسخه پروتکل^۳ نیز برای آن موجود نمی باشد. در ادامه این آزمایش به نمایش کارکرد نرم افزارهای فوق می پردازیم.



شکل ۴-۳: نمایی از سناریو شبیه سازی شده

^۱API

^۲Python

^۳Protocol Conformance Test

```

Frame 17: 82 bytes on wire (656 bits), 82 bytes captured (656 bits) on interface docker0, id 0
Ethernet II, Src: 02:42:4e:e9:49:2d, Dst: 02:42:ac:11:00:02
Internet Protocol Version 4, Src: 172.17.0.1, Dst: 172.17.0.2
Transmission Control Protocol, Src Port: 33686, Dst Port: 6653, Seq: 1, Ack: 1, Len: 16
OpenFlow 1.5
  Version: 1.5 (0x06)
  Type: OFPT_HELLO (0)
  Length: 16
  Transaction ID: 81
  Element
    Type: OFPHET_VERSIONBITMAP (1)
    Length: 8
    Bitmap: 00000040

```

شکل ۴-۴: پیام ارسالی HELLO از سوئیچ به کنترل کننده

```

Frame 25: 74 bytes on wire (592 bits), 74 bytes captured (592 bits) on interface docker0, id 0
Ethernet II, Src: 02:42:ac:11:00:02, Dst: 02:42:4e:e9:49:2d
Internet Protocol Version 4, Src: 172.17.0.2, Dst: 172.17.0.1
Transmission Control Protocol, Src Port: 6653, Dst Port: 33688, Seq: 9, Ack: 17, Len: 8
OpenFlow 1.5
  Version: 1.5 (0x06)
  Type: OFPT_FEATURES_REQUEST (5)
  Length: 8
  Transaction ID: 3211222983

```

شکل ۴-۵: پیام ارسالی FEATURES_REQUEST از کنترل کننده به سوئیچ

۴-۳-۱ محیط شبیه سازی

کنترل کننده Ryu در یک محیط کانتینری به نام Docker با آدرس آیپی^۱ 172.17.0.2 و یک برنامه کاربردی به منظور ایجاد عملکرد یک سوئیچ لایه ۲ شبکه درون آن در حال اجرا می باشد. سوئیچ های OVS نیز توسط برنامه شبیه ساز Mininet ایجاد شده اند و با کنترل کننده توسط آدرس آیپی 172.17.0.1 در ارتباط هستند و سناریو شکل ۴-۳ نشان دهنده بخش های موجود در شبیه سازی می باشد. حال با استفاده از نرم افزار شنود شبکه به نام Wireshark، بسته های رد و بدل شده بین کنترل کننده و سوئیچ را بررسی می کنیم.

همانطور که در شکل ۴-۴ مشاهده می کنیم یک نمونه از پیام ارسالی از سوئیچ به کنترل کننده می باشد که نسخه پروتکل را OF1.5 نشان می دهد و همچنین شکل ۴-۵ نیز یک نمونه از پیام ارسالی از کنترل کننده به سوئیچ است و نسخه این پیام نیز OF1.5 می باشد که نشان دهنده پشتیبانی این دو نرم افزار از نسخه جدید می باشد.

۴-۳-۲ ایجاد عملکرد شبه دیوار آتش

با استفاده از قابلیت جدید TCP_Flags Match در نسخه OF1.5 می توان سیاست های مرتبط با اتصالات TCP اعمال کرد. برای نمایش این قابلیت ابتدا به توضیح یک نوع پویش شبکه^۲ به نام Xmas Scan می پردازیم. طبق این پویش به منظور یافتن دستگاه ها و سرویس های موجود در شبکه، پرچم های FIN,PSH,URG در بسته فعال می گردند. برای جلوگیری از انجام این پویش در شبکه، می توان در سناریو بخش قبل از TCP_Flags Match به منظور یافتن بسته های مربوط به این پویش و حذف آنها اقدام کرد.

^۱IP

^۲Network Scan


```
# nping -c 1 --tcp --flags fin,psh,urg 10.0.0.2

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-02-28 21:23 +0330
SENT (0.0551s) TCP 10.0.0.1:35639 > 10.0.0.2:80 FPU ttl=64 id=42924 iplen=40 seq=432992570 win=1480
RCVD (0.0644s) TCP 10.0.0.2:80 > 10.0.0.1:35639 RA ttl=64 id=0 iplen=40 seq=0 win=0

Max rtt: 9.143ms | Min rtt: 9.143ms | Avg rtt: 9.143ms
Raw packets sent: 1 (40B) | Rcvd: 1 (40B) | Lost: 0 (0.00%)
Nping done: 1 IP address pinged in 1.13 seconds
```

شکل ۴-۶: ارسال درخواست قبل از ایجاد مدخل جریان

```
# nping -c 1 --tcp --flags fin,psh,urg 10.0.0.2

Starting Nping 0.7.91 ( https://nmap.org/nping ) at 2021-02-28 21:24 +0330
SENT (0.0466s) TCP 10.0.0.1:41695 > 10.0.0.2:80 FPU ttl=64 id=60072 iplen=40 seq=1667730679 win=1480
Max rtt: N/A | Min rtt: N/A | Avg rtt: N/A
Raw packets sent: 1 (40B) | Rcvd: 0 (0B) | Lost: 1 (100.00%)
Nping done: 1 IP address pinged in 1.12 seconds
```

شکل ۴-۷: ارسال درخواست بعد از ایجاد مدخل جریان

به این منظور، توسط کنترل کننده یک مدخل جریان در جدول جریان سوئیچ‌ها به شرح زیر ایجاد می‌کنیم:

```
priority=20,tcp,tcp_flags=fin|psh|urg actions=drop
```

با اجراء دستور زیر، نتیجه را قبل و بعد از ایجاد مدخل جریان فوق در شکل‌های ۴-۶ و ۴-۷ مشاهده می‌کنیم:

```
nping --count 1 --tcp --flags fin,psh,urg 10.0.0.2
```

همانطور که طی آزمایش مشاهده شد، بسته‌ها به سمت آدرس مورد نظر ارسال نشدند اما باقی بسته‌هایی که

پرچم‌های مربوطه را ندارد بدون مشکل ارسال و دریافت می‌شوند.

مراجع

- [1] Open Networking Foundation, *OpenFlow Switch Specification, Version 1.5.1 (Protocol version 0x06)*, 2015
- [2] Open Networking Foundation: SDN Definition,
<https://opennetworking.org/sdn-definition>, 2021
- [3] William Stallings, *Foundation of Modern Networking, 1st Edition*, 2015
- [4] Software-Defined Networks: A Systems Approach,
<https://sdn.systemsapproach.org>, 2021
- [5] Ching-Hao, Chang and Dr. Ying-Dar Lin, *OpenFlow Version Roadmap*, 2015
- [6] Open vSwitch,
<https://www.openvswitch.org>, 2021
- [7] Ryu SDN Framework,
<https://ryu-sdn.org>, 2021