

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

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 (Openflow)
۸	۲-۲ اجزاء تشکیل دهنده سوئیچ Openflow
۸	۱-۲-۲ Openflow Channel
۹	۲-۲-۲ Openflow Ports
۱۰	۳-۲-۲ Openflow Pipeline
۱۲	۴-۲-۲ Flow Tables
۱۲	۵-۲-۲ Table-miss
۱۳	۶-۲-۲ Group Table
۱۴	۷-۲-۲ Meter Table
۱۴	۳-۲ بررسی وظایف مهم در خط لوله
۱۴	۱-۳-۲ Match
۱۵	۲-۳-۲ Instruction
۱۶	۳-۳-۲ Counter

۱۷	۳ ویژگی های نسخه جدید Openflow
۱۸	۳-۱ ویژگی های نسخه OF1.5
۱۹	۳-۱-۱ جداول جریان خروجی
۲۰	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 تعریف شده بر مبنای 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 سه لایه‌ای این شبکه‌ها ارتباط بخش کنترلی با بخش هدایت داده از اهمیت بالایی برخوردار است. پروتکل استاندارد اوپن‌فلو^۴ یکی از مهم‌ترین پروتکل‌های ارتباطی بین لایه‌های کنترلی و لایه‌های هدایت داده است 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.

کلمات کلیدی: شبکه‌های نرم‌افزار محور، پروتکل اوپن‌فلو، لایه کنترلی و تصمیم‌گیر، 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

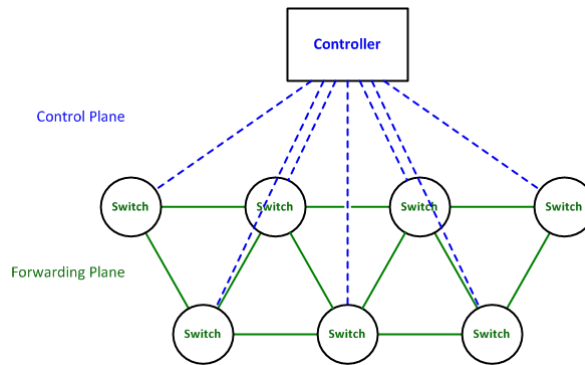
در سال‌های اخیر با توجه به افزایش چشم‌گیر استفاده از شبکه‌های کامپیوتری و نیازمندی این شبکه‌ها به دینامیک بالا به منظور اعمال تغییرات و برنامه‌ریزی سریع، مفهوم نسبتاً جدیدی به نام شبکه‌های تعریف شده بر مبنای نرم‌افزار یا SDN پدید آمده است.

۱-۱ شبکه نرم‌افزار محور (SDN)

شبکه‌های نرم‌افزار محور (SDN) مفهوم نو ظهوری در شبکه‌های کامپیوتری است که بر مبنای آن کنترل‌کننده‌های منطقی مجتمع، رفتار شبکه را کنترل می‌کنند. این گونه 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 کنترل‌کننده^۱ تجهیزات از بخش هدایت‌کننده داده‌ها^۲ جدا شده و این امر موجب فراهم آوردن بستری به منظور برنامه‌ریزی مستقیم شبکه از طریق نرم‌افزار و انتزاعی ساختن زیرساخت شبکه از دید برنامه‌ها و سرویس‌های شبکه شده است.

^۱Control Plane

^۲Data Plane

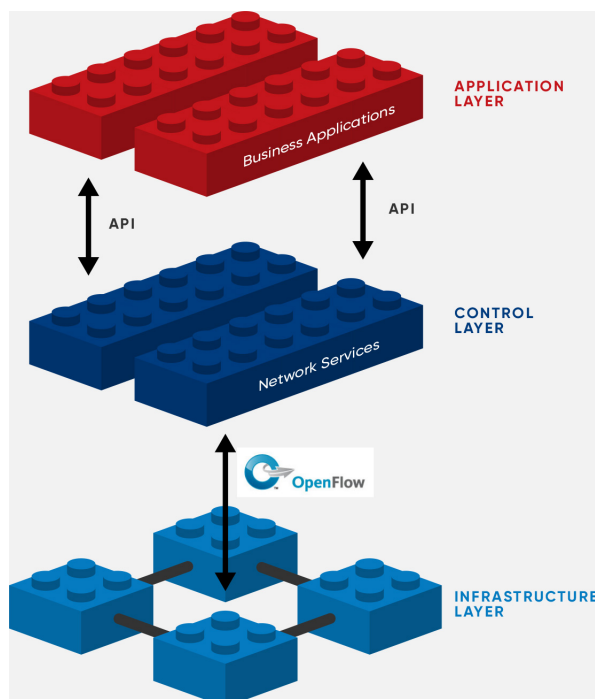


شکل ۱-۱: نمایی انتزاعی از معماری شبکه های نرم افزار محور

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 به توسط رابط های نرم افزاری تنظیمات و مسیرهای حرکت داده 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، توسط پیاده سازی دسته ای^۱ از کنترل کننده 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

در برندهای متفاوت می شود. با ظهور پدیده شبکه های مبتنی بر نرم افزار، این مرز بین برندها برداشته شده و تمام تجهیزات با زبانی مشترک قابل پیکربندی می باشد.

۱-۳-۳ اجزاء تشکیل دهنده معماری SDN

با توجه به شکل ۱-۲ یک شبکه مبتنی بر نرم افزار از اجزاء مختلفی تشکیل شده است که در ادامه به شرح وظایف هر یک از بخش ها می پردازیم.

۱-۳-۱ لایه زیرساخت

لایه زیرساخت^۱، مجموعه ای از تجهیزات شبکه مانند سوئیچ ها، مسیریاب ها و سرورها هستند که وظیفه هدایت ترافیک شبکه را عهده دار می باشند. این لایه در واقع لایه فیزیکی کنترل شده توسط کنترلرهای SDN است.

۱-۳-۲ ارتباط جنوبی

ارتباط جنوبی^۲ که بین لایه زیرساخت و لایه کنترل قرار دارد یکی از مهم ترین بخش های معماری SDN می باشد. از جمله این پروتکل ها می توان به OpenFlow، Netconf و OVSDb اشاره کرد. ما در این پروژه به

^۱Infrastructure Layer

^۲SouthBound Interface (SBI)

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 به عهده دارد. از جمله مهم ترین 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 توسط 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 از 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

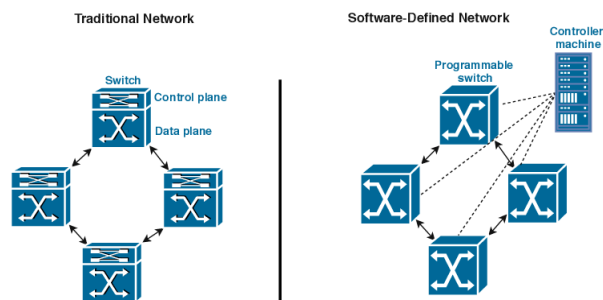
⁵Network Configuration and Management

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 توضیحات جامعی در مورد پروتکل Openflow و نحوه عملکرد آن در معماری SDN داده می‌شود. سپس در فصل دوم ویژگی‌های اضافه شده به نسخه OF1.5 مورد بررسی قرار می‌گیرد.

^۱Flow Table

فصل دوم

پروتکل Openflow

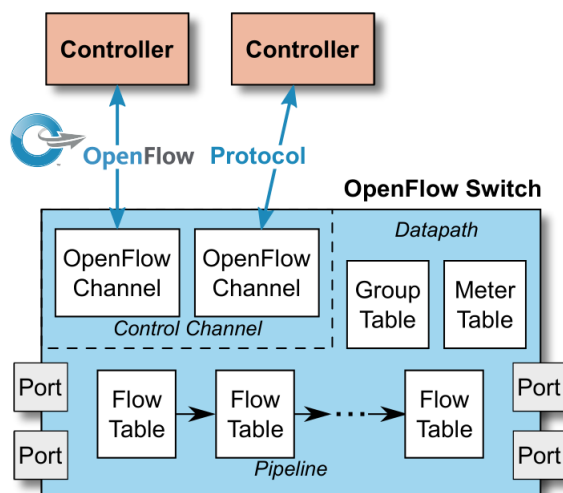
با توجه به توضیحات گفته شده در مقدمه، ارتباط کنترل کننده‌ها با بخش هدایت داده از طریق ارتباط جنوبی^۱ برقرار می‌گردد. برای برقراری این ارتباط پروتکل‌های مختلفی وجود دارد که Openflow (OF) یکی از پرکاربردترین آن‌ها است.

۱-۲ OPENFLOW (OPENFLOW)

پروتکل (Openflow) که توسط بنیاد آزاد و بدون منفعت (ONF) Open Networking Foundation 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 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

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 از اجزاء مختلفی تشکیل شده است که در ادامه به معرفی هر بخش و کاربرد آن‌ها می‌پردازیم [۱].

۱-۲-۲ OPENFLOW CHANNEL

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 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. ارتباط بین کنترل کننده و لایه هدایت داده به نحوه پیاده سازی آن 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.

۲-۲-۲ 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 Port، Logical Ports و Reserved Ports پشتیبانی کند. به مجموعه درگاه‌های Physical، Logical و درگاه LOCAL از نوع درگاه‌های Reserved را 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 اینترفیس‌های سخت افزاری نظیر می‌شود.

- 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، FLOOD و NORMAL، LOCAL، ANY، IN_PORT، TABLE پشتیبانی کند و درگاه‌های UNSET قابل پشتیبانی می‌باشد. در ادامه به توضیح کاربرد نیز به صورت اختیاری برای سوئیچ‌های Openflow قابل پشتیبانی می‌باشد. 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.

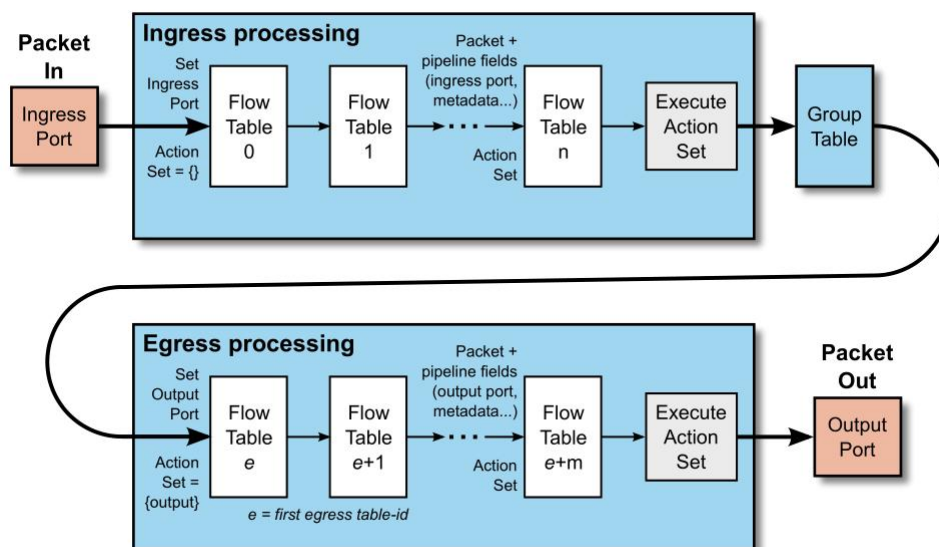
– CONTROLLER: به عنوان کانال ارتباطی بین سوئیچ و کنترل کننده استفاده می‌شود که می‌توان آن را به صورت ورودی یا خروجی تعریف نمود.

¹Ingress Ports

²Pipeline

³Output Ports

⁴Ethernet Switch



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

- 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.

۳-۲-۲ OPENFLOW PIPELINE

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

^۱pipeline

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 جداول 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 را 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.

^۱Flow Table

^۲Flow Entry

FLOW TABLES ۴-۲-۲

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

- **Match Fields:** برای ایجاد تطابق با بسته‌ها کاربرد دارد که از قسمت‌های درگاه ورودی^۱، سرآیندهای بسته‌ها^۲ و به صورت اختیاری پارامترهای خط لوله^۳ مانند فراداده‌های^۴ جدول‌های پیشین تشکیل شده است.
 - **Priority:** نشان دهنده اولویت بسته‌ها به منظور انجام تطابق می‌باشد.
 - **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 اعمال 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 اعمال گفته شده در بالا را ندارند که این قابلیت‌ها بسته به پیاده سازی برندهای مختلف از پروتکل متفاوت می‌باشد.

TABLE-MISS ۵-۲-۲

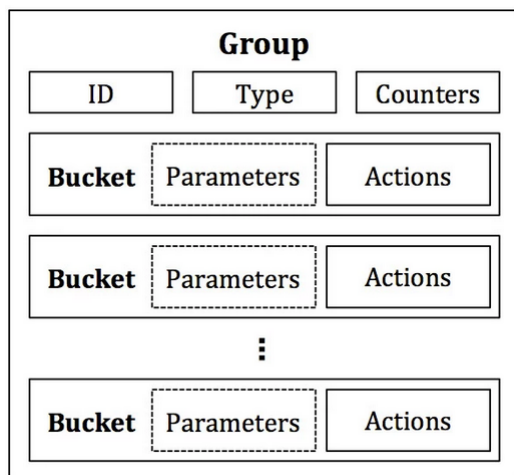
در هر جدول جریان، در صورتی که عمل تطابق صورت نگیرد، مدخل جریانی به نام Table-miss در هر جدول جریان وجود دارد که وضعیت بسته را برای ادامه در خط لوله مشخص می‌کند. برای مثال می‌توان بسته

¹Ingress Port

²Packet Headers

³Pipeline Fields

⁴Metadata



شکل ۲-۳: نمایی از شکل کلی جدول گروه

را دور انداخت یا به جدول جریان بعدی و یا به کنترل کننده ارسال کرد.

GROUP TABLE ۶-۲-۲

همان طور که در بخش قبلی نیز بیان شد، جدول گروه، انتزاعی به منظور پیاده سازی عملیات‌های پیچیده و خاص که قابلیت پیاده سازی آسان با استفاده از جداول جریان و مدخل‌های آن‌ها 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 به 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.

^۱Bucket

^۲Type

^۳Actions

• SELECT: در این گروه، سطل‌ها به صورت وزن دار اولویت برای دریافت بسته‌ها دارند. کاربرد اصلی این گروه به منظور ایجاد تراز بار^۱ 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 ۷-۲-۲

در نسخه OF1.3، مفهومی به نام اندازه‌گیرها^۲ 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 اندازه‌گیرها هستند که این خود زمینه ایجاد کنترل کیفیت سرویس^۳ را فراهم می‌آورد.

۳-۲ بررسی وظایف مهم در خط لوله

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

MATCH ۱-۳-۲

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 باشند، 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^۴ بالاتری داشته باشد عملیات را روی بسته انجام خواهد داد.

¹Load Balancing

²Meters

³Quality of Service (QoS)

⁴Priority

INSTRUCTION ۲-۳-۲

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 بسته، مجموعه عملیات^۱ و یا در پردازش خط لوله آن 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
- Clear-Actions: پاک کردن تمام عملیات‌های موجود در مجموعه عملیات
- Write-Actions: اضافه کردن عملیات به مجموعه عملیات بسته
- Write-Metadata: اضافه کردن فر داده 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 اندازه گیرها 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.

^۱ Action-Set

- *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.

۲-۳-۳ COUNTER

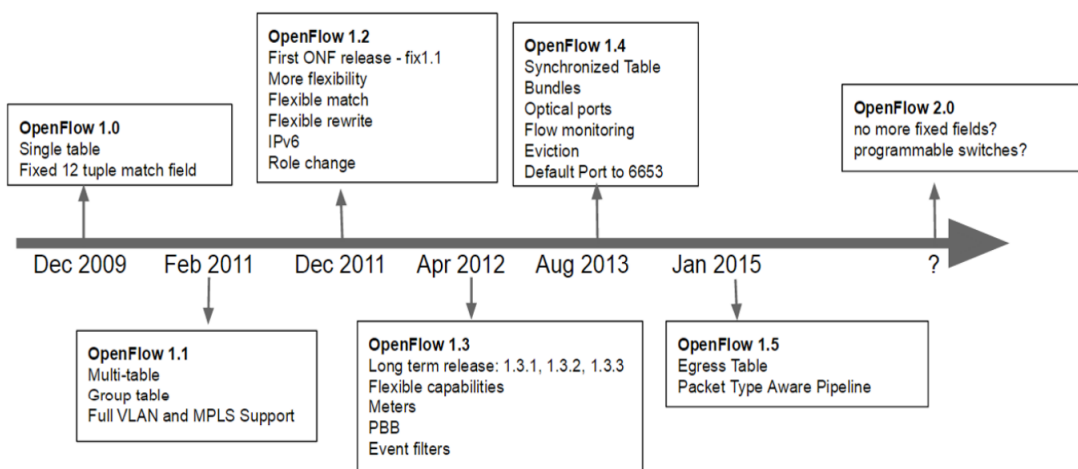
شمارنده‌ها برای هر جدول جریان، مدخل جریان، درگاه، صف، گروه، سطل گروه و اندازه‌گیر وجود دارند. در هر بخش، شمارنده‌ها مولفه‌های مرتبط با آن بخش شمرده و به صورت آمار به کنترل کننده ارسال می‌کنند. تمامی شمارنده‌های موجود به صورت بدون علامت بوده و نمایشگر سرریز^۱ ندارند. نمونه‌هایی از شمارنده‌های موجود برای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

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
- Extensible Flow Entry Statistics (OXS)
- ارسال خودکار آماره‌های مدخل جریان
- عملیات Copy-Field
- فیلدهای خط لوله Packet Register
- تطابق TCP flags
- وضعیت ارتباط کنترل کننده

ب بهبود یافته‌ها

- دستورات تکاملی گروه برای تغییرات انتخابی در سطرها
- قابلیت عملیات set-field برای تنظیم فراداده‌ها
- اجازه استفاده از wildcard در عملیات set-field
- Scheduled Bundles

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 پورت 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 عملیات خروجی 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.

¹ Egress Flow Tables

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