



### چکیده

موضوع اصلی متن این است که شبکه‌های تعریف شده توسط نرم‌افزار (SDN) یک پیشرفت مهم در تاریخ ساختاردهی شبکه‌های کامپیوتری به صورت برنامه‌پذیرتر هستند. متن به تکامل شبکه‌های برنامه‌پذیر می‌پردازد و اشتباهات و افکار غلط درباره این فناوری‌ها را اصلاح می‌کند. همچنین، رابطه بین SDN و مجازی‌سازی شبکه نیز مورد بررسی قرار می‌گیرد.

### خلاصه

#### مقدمه

پیچیدگی مدیریت شبکه‌های کامپیوتری زیاد است و چگونگی تکامل شبکه‌های SDN خواهیم پرداخت. مقاله نشان می‌دهد که شبکه‌های سنتی با تجهیزات متنوعی مانند روترها، سوئیچ‌ها و جعبه‌های وسطی مانند دیوارهای آتش و سیستم‌های تشخیص نفوذ پیچیده و مشکلات مدیریتی دارند. سپس به نحوه عملکرد SDN با جداسازی سطوح کنترل و داده و یکپارچه‌سازی سطح کنترل اشاره کرده و نقد به ایده‌های گذشته و ارتباطات میان SDN و تکنولوژی‌های مرتبط مانند مجازی‌سازی شبکه را مطرح کرده است. این متن سپس به تاریخچه فکری از شبکه‌های برنامه‌پذیر با تمرکز بر SDN پرداخته و تکامل ایده‌ها و تأثیرات فناوری در طول زمان را بیان کرده و اشتباهات شایع درباره هر یک از تکنولوژی‌ها را اصلاح می‌کند. سطح کنترل SDN، با استفاده از رابط برنامه‌نویسی اپلیکیشن (API) دقیق، به صورت مستقیم کنترل بر روی عناصر سطح داده شبکه (مانند روترها، سوئیچ‌ها و سایر جعبه‌های وسطی) را اجرا می‌کند. OpenFlow یک نمونه برجسته از چنین API ای است. یک سوئیچ OpenFlow دارای یک یا چند جدول قوانین پردازش بسته است. هر قانون یک زیرمجموعه از ترافیک را مطابقت می‌دهد و اقدامات معینی را بر روی ترافیکی که با قانون مطابقت دارد، انجام می‌دهد؛ این اقدامات شامل ریزش، ارسال یا پخش بسته به قوانینی که توسط یک برنامه کنترل نصب می‌شوند، یک سوئیچ OpenFlow می‌تواند مانند یک روتر، سوئیچ، دیواره آتش، مترجم آدرس شبکه یا چیزی در میانه رفتار کند.

#### مسیر به SDN

توسعه شبکه‌های SDN در سه مرحله اصلی شامل شبکه‌های فعال (۱۹۹۵ تا ابتدای دهه ۲۰۰۰)، جداسازی سطوح کنترل و داده (۲۰۰۱ تا ۲۰۰۷)، و استفاده از رابط OpenFlow و سیستم‌های عامل شبکه (۲۰۰۷ تا ۲۰۱۰) انجام شده است. شبکه‌های مجازی نقش مهمی در تاریخ توسعه SDN دارند. مرحله اولیه به طور مفصل به شبکه‌های فعال اختصاص یافته و نشان داده شده است که چگونه این تکنولوژی به افزایش برنامه‌پذیری شبکه کمک کرده است. ۲ مدل برنامه‌نویسی توجه بیشتری جلب کرد: capsule model و programmable router/switch. مدل اول برنامه در گرهی ورودی اجرا می‌شد و مدل دوم در خروجی گره. با کاهش هزینه محاسبات تکنولوژی موجب شد تا به سمت شبکه‌های فعال حرکت شود. در همین زمان درخواست کاربران و محققان برای پیاده‌سازی‌های وسیع‌تر سرویس‌های جدید نیز عامل دیگری برای رشد شبکه‌های فعال شد. البته همواره مخالفانی وجود داشت که نظر آن‌ها این بود نباید شبکه پیچیده شود. افزایش حجم ترافیک و تأکید بر قابلیت اطمینان شبکه موجب شد تا مفاهیمی همچون مهندسی ترافیک بروز کند و این نیازها موجب شد تا رهیافت جداسازی سطوح کنترل و داده برگزیده شود. انتقال قابلیت کنترل از تجهیزات شبکه به سرورهای جداگانه به دلیل اینکه مدیریت شبکه نیز یک فعالیت بر مبنای کل شبکه است، معناپذیر بود. دو طراحی مختلف در این حوزه انجام شده است: کنترل منطقی به صورت مرکزی با استفاده از یک رابط باز

به صفحه داده و مدیریت وضعیت توزیع شده. البته ایراداتی از جمله اینکه اگر قسمت کنترل دچار اختلال شود قسمت دیتا هم مختل نمی شود نیز وجود داشت (برخلاف قاندهی fate sharing).

در دهه‌ی میانی سال‌های ۲۰۰۰، علاقه‌مندی محققان و نهادهای تامین مالی به ایده آزمایش شبکه به مقیاس بزرگ افزایش یافت. نتیجه این انگیزه، ایجاد محیط جهانی برای نوآوری‌های شبکه (GENI) با دفتر پروژه GENI که تحت تأمین مالی NSF بود و برنامه EU FIRE بود. منتقدان این تلاش‌های متمرکز بر زیرساخت، اشاره کردند که این سرمایه‌گذاری بزرگ در زیرساخت با ایده‌های خوب برای استفاده از آن همخوانی نداشت. در این میان، یک گروه از محققان در دانشگاه استنفورد برنامه Clean Slate را ایجاد کرد و بر تجربه در مقیاس محلی و قابل مدیریت تمرکز کرد و آن شبکه‌های دانشگاهی بود. قبل از ظهور OpenFlow، ایده‌های مبنی بر SDN با تناقض مواجه بودند: تناقض بین تصویر شبکه‌های کاملاً قابل برنامه‌ریزی و واقع‌گرایی که امکان پیاده‌سازی واقعی در دنیای واقعی را فراهم می‌کند OpenFlow. تعادلی بین این دو هدف برقرار کرد، امکان اجرای وظایف بیشتری نسبت به کنترل‌کننده‌های مسیر قبلی فراهم کرد و بر پایه استفاده روزافزون از سخت‌افزارهای سوئیچ موجود، از طریق استفاده روز به روز از چیپست‌های مرسوم در سوئیچ‌های تجاری. با این حال، اعتماد به سخت‌افزار سوئیچ موجود تا حدی انعطاف پذیری را محدود کرد، اما OpenFlow تقریباً به سرعت قابل اجرا بود و این اجازه را به جنبش SDN داد تا هم واقع‌گرا و هم جسور باشد. یک سوئیچ OpenFlow دارای یک جدول قوانین پردازش بسته است، هر قانون الگویی را دارد (که بر روی بیت‌های هدر بسته مطابقت دارد)، یک لیست از اقدامات (مانند ریزش، سیلاب شدن، ارسال به یک رابط خاص، تغییر یک فیلد هدر، یا ارسال بسته به کنترل‌کننده)، مجموعه‌ای از شمارنده‌ها (برای پیگیری تعداد بایت و بسته‌ها) و یک اولویت (برای تفکیک قوانین با الگوهای همپوشانی دارد) است. هنگام دریافت یک بسته، یک سوئیچ OpenFlow قانون مطابقت با بالاترین اولویت را شناسایی می‌کند، اقدامات مرتبط را انجام می‌دهد و شمارنده‌ها را افزایش می‌دهد.

از قبل از ظهور SDN تاکنون، تجهیزات شبکه پشتیبانی از ایجاد شبکه‌های مجازی در قالب VLAN ها و شبکه‌های خصوصی مجازی کرده‌اند. با این حال، تنها مدیران شبکه می‌توانستند این شبکه‌های مجازی را ایجاد کنند و این شبکه‌های مجازی تنها محدود به اجرای پروتکل‌های موجود شبکه بودند. برخلاف شبکه‌های فعال، شبکه‌های overlay نیازی به پشتیبانی ویژه از تجهیزات شبکه یا همکاری از سوی ارائه‌دهندگان خدمات اینترنت نداشتند، که این امر باعث سهولت بسیار بیشتر در پیاده‌سازی آن‌ها شد.

### ارتباط مجازی‌سازی شبکه با SDN:

مجازی‌سازی شبکه بدون شک نیازی به SDN ندارد. به همین ترتیب، SDN (جداسازی یک کنترل مرکزی منطقی از دیتاپلین زیرین) نیز به مفهوم واقعیت مجازی شبکه اشاره نمی‌کند. با این حال، یک هم‌زیستی بین مجازی‌سازی شبکه و SDN به وجود آمده است:

SDN به عنوان فناوری ممکن‌کننده مجازی‌سازی شبکه: محاسبات ابری باعث برجسته شدن واقعیت مجازی شبکه شد، زیرا ارائه‌دهندگان ابر نیاز به راهی برای امکان به اشتراک گذاری زیرساخت شبکه یکسان برای چندین مشتری (یا "مستأجر") دارند. مجازی‌سازی شبکه برای ارزیابی و آزمایش SDN: قابلیت جدا کردن یک برنامه کنترل SDN از دیتاپلین زیرین امکان آزمایش و ارزیابی برنامه‌های کنترل SDN را در یک محیط مجازی قبل از اینکه برنامه در یک شبکه عملیاتی پیاده‌سازی شود فراهم می‌کند.

مجازی‌سازی ("تکه‌ای‌سازی") یک SDN: در شبکه‌های سنتی، مجازی‌سازی یک روتر یا سوئیچ پیچیده است، زیرا هر جزء مجازی نیاز به اجرای نمونه خود از نرم‌افزار کنترل پلین دارد. بر خلاف این، مجازی‌سازی یک سوئیچ SDN "بی‌فهم" بسیار ساده‌تر است.