KNO-1103-4501

# ارتقاء امنیت و سرعت در شبکه های SDN مبتنی بر DPDK و IPSec

عباس معنوی نژاد

کارشناسی ارشد مهندسی مخابرات - دانشگاه آلتو، هلسینکی

Abbas.mn@live.com

## چکیده:

شبکه مبتنی بر نرم‌افزار (SDN) رویکردی در معماری شبکه و روشی برای شبکه‌سازی است که هدفش بهینه‌سازی عملیات شبکه است. این مطالعه به بررسی ادغام کیت توسعه صفحه داده (DPDK) و امنیت پروتکل اینترنت (IPSec) در شبکه مبتنی بر نرم‌افزار (SDN) برای طراحی یک پروتکل ارتباطی سفارشی سازی شده با توان عملیاتی بالا و ایمن می‌پردازد. DPDK مجموعه ای از کتابخانه ها و درایورها برای پردازش سریع بسته و همچنین ویژگی‌های پیشرفته مدیریت حافظه در DPDK مانند صفحات بزرگ (Pages Huge) و دسترسی به حافظه غیر یکنواخ (NUMA) است و پژوهش انجام شده به دنبال راهکاری برای بهبود قابل توجهی در سرعت پردازش بسته‌های دارای سربارهای اضافه برای رمزنگاری است. در همین حال، IPSec از طریق رمزنگاری و احراز هویت، امنیت داده‌ها را تضمین می‌کند و یک لایه ارتباطی امن را در چارچوب SDN با عملکرد بالا فراهم می‌سازد .این پژوهش همچنین پردازش بسته برداری (VPP) را به عنوان یک روتر یا سوئیچ مجازی و سهم بالقوه آن در راهکار پیشنهادی بررسی می کند.

## کلمات کلیدی:

*ارتقاء امنیت، امنیت سایبری، شبکه مبتنی بر نرم افزار (SDN)، کیت توسعه صفحه داده (DPDK)، پردازش بسته برداری (VPP)*

**- مقدمه:**

شبکه‌های مبتنی بر نرم‌افزار (SDN)[۱]خود را به عنوان جدیدترین و مهم‌ترین معماری شبکه در صنعت ارتباطات معرفی کرده اند و به یکی از مفاهیم پرکاربرد در شبکه تبدیل شده‌اند که به منظور برقراری ارتباط با زیرساخت‌های سخت‌افزاری در شبکه از کنترل‌کننده‌های مبتنی بر نرم‌افزار یا رابط‌های برنامه‌نویسی کاربردی (API)[۲]که مجموعه‌ای از پروتکل‌ها و ابزارهایی است که به نرم‌افزارهای مختلف اجازه می‌دهد تا با یکدیگر ارتباط برقرار کنند، استفاده می‌کنند. این شبکه‌ها با توجه به امکانات گسترده‌ای که نسبت به شبکه‌های سنتی ارائه می‌دهند، نوآوری مناسبی دارند. قدرت مقیاس پذیری شبکه های SDN اجازه می‌دهد تا توابع جدید شبکه از طریق منطق مبتنی بر نرم افزار در صفحه کنترل شبکه به سادگی معرفی شوند. در SDN، کل شبکه در صفحه کنترل انتزاع شده و به مدیر شبکه یک نمای کلی از تمامی دستگاه‌های موجود در شبکه را نشان می دهد. شبکه های SDN، با جداسازی صفحه کنترل از صفحه داده، پارادایم های مدیریت شبکه را عمیقاً متحول کرده اند. این جداسازی امکان پیکربندی شبکه های پویا و قابل برنامه ریزی را با مقیاس پذیری و سفارشی سازی بالا فراهم می کند. با این حال، ماهیت متمرکز و قابلیت برنامه ریزی SDN، آسیب پذیری های امنیتی منحصر به فردی را ارائه می دهد که شبکه های سنتی کمتر با آن مواجه هستند. برای مثال ، دسترسی به هر نحوی به کنترلر اصلی SDN و یا فرمان های کنترلی در هر بخشی از مسیر آنها توسط افراد غیر مجاز، خطر تغییر سیاست های شبکه و تغییر مسیر و محتوای داده ها را به همراه دارد.[1][2]

امنیت پروتکل اینترنت (IPsec) یکی از پرکاربردترین پروتکل های رمزگذاری انتقال داده است که امروزه برای اطمینان از محرمانه بودن، صحت و یکپارچگی داده های ارسال شده طراحی شده است. ادغام IPsec[۳] در SDN با ایمن کردن سطوح کنترل و داده از طریق رمزگذاری قوی، احراز هویت و بررسی یکپارچگی، امنیت را افزایش می‌دهد. IPsec می تواند از ارتباط بین کنترل کننده SDN و دستگاه های شبکه محافظت کند و فقط از تعاملات مجاز اطمینان حاصل کند و از دستکاری و استراق سمع جلوگیری کند. با تعریف و اجرای سیاست‌های امنیتی متمرکز، IPsec تهدیدهایی مانند حملات DoS[۴]و MitM[۵]را کاهش می‌دهد، در حالی که به‌روزرسانی‌های سیاست های شبکه، پویایی و انعطاف‌پذیری آن را حفظ می‌کنند. به عبارتی، پیاده سازی IPSec در SDN علاوه بر حفاظت از داده ها، از اجرای بروزرسانی های سیاست های شبکهSDN صرفا از طریق افراد مجاز، اطمینان حاصل می کند. [3][4]

در حال حاضر، دروازه های رایج IPsec برای رمزگذاری و رمزگشایی و انتقال داده ها به پشته پروتکل سنتی لینوکس متکی هستند. این اتکا و وابستگی به هسته لینوکس، باعث کپی های متعدد داده و سوئیچینگ های مازاد در هسته می شود که نتیجه آن انتقال داده با تاخیر و سرعت پایین است.

[۱] Software-Defined Networking

[۲] **APPLICATION PROGRAMMING INTERFACE (API)**

[۳] Internet Protocol Security

[۴] denial-of-service

[۵] man-in-the-middle

برای جبران کاهش بازدهی نرخ انتقال داده هنگام ادغام IPsec درSDN ، استفاده از کیت توسعه صفحه داده(DPDK) [۶] پیشنهاد می شود. DPDK با دور زدن هسته سیستم عامل کار می کند و امکان پردازش سریع بسته ها را مستقیماً در فضای کاربر فراهم می کند. این روش کارآمد تاخیر را کاهش می دهد و توان عملیاتی بسته را افزایش می دهد که برای حفظ عملکرد در شبکه های پرسرعت بسیار مهم است. با استفاده از درایورهای حالت واکشی PMD [۷]، DPDK سربار مربوط به پردازش سنتی بسته ها، تحت تاثیر وقفه های هسته را به حداقل می رساند و بازدهی کلی شبکه را افزایش می دهد. [5]

ادغام DPDK با SDN و IPsec تضمین می کند که سربارهای امنیتی IPsec بازدهی نرخ انتقال داده را در شبکه تحت تاثیر قرار نمی دهد. توانایی DPDK برای مدیریت انتقال داده با سرعت بالا، رمزنگاری و احراز هویت قوی IPsec را بدون ایجاد تاخیرهای قابل توجه امکان پذیر می کند. این ترکیب به محیط‌های SDN اجازه می‌دهد به یک زیرساخت متعادل و کارآمد دست یابند، که در آن امنیت و سرعت به طور یکپارچه همزیستی دارند و عملکرد بالا و محافظت قوی در برابر آسیب‌پذیری‌ها را تضمین می‌کنند.

برای افزایش بیشتر عملکرد و امنیت SDN هنگام استفاده از IPsec وDPDK ، ادغام پردازش بسته برداری (VPP[۸]) می تواند بسیار سودمند باشد. VPPیک چارچوب منبع باز پردازش بسته در صفحه داده و با کارایی بالا است که در فضای کاربر اجرا می شود و قابلیت های شبکه ای پیشرفته را ارائه می دهد. در حالی که DPDK با مدیریت کارآمد بسته های ورودی/خروجی، توان عملیاتی بالا و تاخیر کم را تضمین می کند،VPP انعطاف پذیری، مقیاس پذیری و سفارشی سازی در مسیریابی داده ها به محیط SDN اضافه می کند.[6]

در زمینه SDN، VPP با ایجاد امکان ایجاد روترها، سوئیچ های نرم افزاری و سایر عملکردهای شبکه قابل تنظیم، مزایای قابل توجهی را ارائه می دهد. معماری ماژولار آن به اپراتورهای شبکه اجازه می دهد تا رفتار شبکه را مطابق با نیازهای خاص تنظیم کنند و کارایی و عملکرد کلی را افزایش دهند. VPP همچنین از استقرار سریع و مقیاس پذیری پشتیبانی می کند و سازگاری با نیازهای متغیر شبکه را آسان تر می کند. با ادغام VPP با SDN، IPsec و DPDK، محیط های شبکه می توانند از امنیت قوی، توان عملیاتی بالا و قابلیت ایجاد و تغییر لحظه ای عملکردها و توابع شبکه بدون نیاز به خاموش کردن شبکه، بهره مند شوند.

## ۲- مطالعات پیشین

شبکه‌های تعریف‌شده نرم‌افزار (SDN) مدیریت شبکه را متحول کرده است و انعطاف‌پذیری و برنامه‌پذیری بی‌سابقه‌ای را ارائه می‌دهد. با این حال، دستیابی به عملکرد بهینه و امنیت قوی در محیط‌های SDN یک نقطه کانونی برای تحقیق و توسعه است.

مطالعات متعددی به محدودیت‌های عملکرد معماری‌های قدیمی تر SDN پرداخته‌اند. یکی از زمینه های تمرکز در تحقیقاتSDN ، افزایش سرعت شبکه از طریق ادغام تکنیک های تخصصی پردازش صفحه داده است. به عنوان مثال، استفاده از کیت توسعه صفحه داده (DPDK) به دلیل توانایی آن در سرعت بخشیدن به پردازش بسته با دور زدن هسته سیستم عامل و اجرای وظایف صفحه داده به طور مستقیم در فضای کاربر، توجه قابل توجهی را به خود جلب کرده است. چندین مطالعه اثربخشی DPDK را در بهبود بازدهی پردازش بسته و کاهش تأخیر در محیط‌های SDN نشان

[۶] Data Plane Development Kit

[۷] Poll Mode Drivers

[۸] Vector Packet Processing

داده‌اند .[7][8] همچنین نویسندگان در [9] تاثیر استفاده از صفحات داده قابل برنامه ریزی را برای سرعت بخشیدن به پردازش بسته ها، مورد بررسی قرار دادن و در [10] نویسندگان با استفاده از تکنیک های سخت افزاری مانند SmartNIC، بهبود راندمان پردازش بسته ها را بررسی کردند.

در تحقیقات دیگری، الگوریتم های متعادل کننده بار متناسب با محیط های SDN نیز مورد بررسی قرار گرفته اند. در [11][12] [13] رویکردهای مبتنی بر یادگیری ماشین برای متعادل کردن بار و بهینگی پردازش بسته ها و مسیریابی بر اساس معیارهای مختلف شبکه و الزامات کاربردی بررسی شده اند. یافته های آنها بهبود استفاده از شبکه و کاهش تراکم را در مقایسه با روش های متعادل بار سنتی نشان می دهند.

. نتیجه ای این تحقیقات نشان می دهد که تمرکز صرفا بر روی یک بخش از کل فرآیند پردازش بسته، همانند کنترل های شبکه و یا متعادل سازی بار ترافیکی، توانایی کافی را برای ایجاد راندمان کافی برای مدیریت افزایش روز افزون بار ترافیکی شبکه های امروزی ندارند.

صفحه کنترل متمرکز SDN فرصت ها و چالش هایی را برای امنیت ارائه می دهد. به موازات آن، اطمینان از مکانیزم های امنیتی قوی در شبکه های SDN برای محافظت در برابر طیف گسترده ای از تهدیدات سایبری ضروری شده است. در [14][15] نویسندگان انواع حملات در شبکه های SDN و راه حل های پیشنهادی برای آنها مانند Avant-Guard و VAVE (Virtual source Address Validation Edge) و یا استفاده از رمزگذاری SSL[۹] و آنالیزهای آنتروپی را مورد بررسی قرار داده اند. نتیجه این تحقیق نشان می دهد که هر کدام از راه حل ها می توانند در مقابل یک حمله خاص محافظت انجام دهند و قادر به ارائه راهکار محافظتی کلی و یکپارچه نمی باشند و استفاده موازی از چند راه حل به طور همزمان، سربارهای بسیار زادی را در پردازش بسته ها و مسیریابی به شبکه تحمیل می کند.

در [16] نویسندگان با پیاده سازی IPSec در SDN راه حلی برای یکپارچگی و احراز هویت ترافیک کل شبکه فراهم کرده اند، که نتیجه خطر دسترسی غیرمجاز و دستکاری داده ها را کاهش می دهد.

به طور خلاصه، بررسی ادبیات تلاش های مداوم در جامعه تحقیقاتی را برای افزایش سرعت و امنیت شبکه SDN از طریق تکنیک های مختلف نشان می دهد. در حالی که مطالعات جداگانه مزایای DPDK و IPsec را به صورت مجزا بررسی کرده اند، علاقه فزاینده ای به بررسی استفاده ترکیبی از راه حل ها و تکنولوژی ها برای بهره برداری از هم افزایی آنها در کنار هم در سرعت و امنیت در شبکه هایSDN وجود دارد. ادغام IPsecدر مسیر پردازش بسته مبتنی بر DPDK، همراه با استفاده از VPP برای مسیریابی بهینه، یک رویکرد امیدوار کننده برای دستیابی به استقرار SDN با کارایی بالا و ایمن ارائه می دهد. با این حال، نیاز به تحقیقات بیشتر برای کشف مفاهیم عملی، مبادلات عملکرد، و چالش‌های استقرار مرتبط با چنین راه حل یکپارچه ای وجود دارد.

[۹] Secure Sockets Layer

## ۳- تکنولوژی های پیشنهادی

### ۱,۳- کیت توسعه صفحه داده (DPDK)

DPDKیک پروژه نرم افزاری منبع باز است که توسط بنیاد لینوکس مدیریت می شود. مجموعه‌ای از کتابخانه‌ها و درایورها را برای پردازش سریع بسته‌ها فراهم می‌کند که انتقال بسته در صفحه داده با کارایی بالا را ممکن می‌سازد. DPDK همانطور که اشاره شد، پشته شبکه هسته را دور می زند و از PMD ها و استراتژی های مدیریت حافظه بسیار پیچیده برای بهینه سازی فرآیندهای دریافت و ارسال بسته استفاده می کند. این استراتژی ها برای دستیابی به عملکرد بالا در برنامه های کاربردی شبکه بسیار مهم است. در اینجا تفکیکی بیشتر از بارزترین ویژگی های معماری مدیریت حافظه توسط DPDK آورده شده است:

الف. صفحات بزرگ (Huge Page):

DPDK از صفحات بزرگ برای تخصیص بلوک های بزرگ به هم پیوسته حافظه استفاده می کند. صفحات حافظه استاندارد معمولاً ۴ کیلوبایت حجم دارند که منجر به تعداد زیادی ورودی در جدول ترجمه صفحات حافظه می شوند. از طرف دیگر، صفحات بزرگ معمولاً ۲ مگابایت یا بزرگتر هستند (تا ۱ گیگابایت) که به طور قابل توجهی تعداد ورودی ها و جستجوها را در جدول ترجمه صفحه حافظه کاهش می دهند. این امر، سربار جستجوهای جدول ترجمه صفحه حافظه را به حداقل می رساند و خطاهای ترجمه TLB[۱۰] را کاهش می دهد و در نتیجه سرعت دسترسی به حافظه را افزایش می دهد. [17]

در جدول زیر، تفاوت تعداد صفحات را برای فضای حافظه ۲۵۶ گیگابایتی مشاهده می کنید:

**جدول ۱. مقایسه تعداد صفحات حافظه با استفاده از Huge Page**

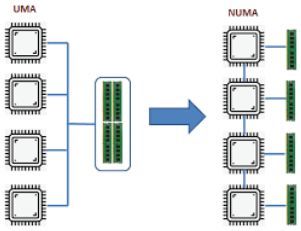
| اندازه هر صفحه | ۴KB | ۲MB | ۱ GB |
|---|---|---|---|
| تعداد صفحات | ۶۷۱۰۸۸۶۴ | ۱۳۱۰۷۲ | ۲۵۶ |

ب. به دسترسی به حافظه غیر یکنواخت (NUMA)[۱۱]

معماری دسترسی به حافظه غیر یکنواخت (NUMA) شامل چندین گره حافظه است که زمان دسترسی به حافظه بسته به گره متفاوت است. DPDK آگاه به NUMA است، به این معنی که تخصیص حافظه را بر اساس نزدیکی حافظه به هسته CPU ای که به آن دسترسی خواهد داشت، بهینه می کند. این امر تأخیر دسترسی به حافظه را کاهش می دهد و با اطمینان از تخصیص حافظه از نزدیک ترین گره ممکن، توان عملیاتی را بهبود می بخشد.

[۱۰] Table Lookaside Buffer

[۱۱] Non-Uniform Memory Access




**شکل ۱. مقایسه شماتیک معماری NUMA و غیر NUMA**

DPDK استخرهای حافظه (Memory Pools) جداگانه ای را برای هر یک از گره های NUMA نگهداری می کند. هنگامی که یک هسته CPU حافظه را درخواست می کند، DPDK آن را از استخر محلی (Local Pool) مرتبط با گره NUMA آن هسته اختصاص می دهد. این امر دسترسی به حافظه متقابل گره را به حداقل می رساند و باعث کاهش تأخیر و افزایش عملکرد می شود[17] .

ج. انتقال داده بدون کپی (Zero-Copy)

DPDK از تکنیک‌های Zero-Copy برای جلوگیری از کپی غیر ضروری داده‌ها بین بافرها استفاده می‌کند. پردازش بسته های سنتی اغلب شامل مراحل متعددی است که در آن داده ها از یک بافر به بافر دیگر کپی می شوند و سربار قابل توجهی را ایجاد می کنند. تکنیک‌های Zero-Copy تضمین می‌کنند که داده‌ها مستقیما از دیسک به سوکت ها کپی شوند، بدون اینکه وارد بافرهای هسته شود و باعث می شود چرخه های CPU مورد نیاز برای کپی کردن داده ها کاهش یابد و در نتیجه عملکرد را بهبود می بخشد. این ویژگی سربار کپی حافظه را به حداقل می‌رساند، که هنگام مدیریت حجم زیادی از ترافیک رمزگذاری شده مفید است. [19][18]

د. دسترسی مستقیم به حافظه (DMA) :[۱۲]

DPDK از DMA برای انتقال مستقیم داده ها بین کارت رابط شبکه(NIC) [۱۳]و حافظه برنامه بدون دخالتCPU استفاده می کند. این امر بارگذاری وظایف انتقال داده به بخش کنترل دسترسی مستقیم به حافظه CPU را برای سایر وظایف پردازشی آزاد می کند و با اجتناب از وقفه های CPU در آن، تأخیر را کاهش می دهد. استفاده DPDK از DMA سربار CPU را کاهش می دهد و سرعت انتقال داده را افزایش می دهد، که برای مدیریت موثر رمزگذاری/رمزگشایی پروتکل های ایمن سازی بسته های داده مانند IPsec بسیار مهم است. [20]

ه. استخرهای حافظه (Mempools) و کش محلی (Local Cache)

DPDK از استخرهای حافظه (mempool) برای مدیریت بافرهای بسته (mbufs) استفاده می کند. Mempool ها مخازن از پیش تخصیص داده شده از بافرهای با اندازه ثابت هستند که برای ذخیره داده های بسته استفاده می شوند. هنگامی که یک بسته می رسد، یک mbuf از mempool برای رسیدگی به بسته اختصاص داده می شود. این پیش تخصیص، سربار مربوط به تخصیص حافظه پویا در طول پردازش بسته را که در حالت سنتی پردازش بسته ها، تحت تاثیر وقفه های هسته سیستم عامل قرار می گیرد، کاهش می دهد.

[۱۲] Direct Memory Access
[۱۳] Network Interface Card

DPDK برای کاهش مداخله ها در استخر حافظه (Memory Pool) ، کش های کوچکی از mbuf ها را برای هر هسته نگهداری می کند. هنگامی که یک هسته به mbuf نیاز دارد، ابتدا کش محلی خود را بررسی می کند. اگر کش خالی باشد، تنها در این صورت به استخر حافظه اصلی دسترسی پیدا می کند. این باعث کاهش سربارهای اضافی دسترسی به mempool می شود و محلی بودن حافظه کش را بهبود می بخشد. [21][22]

و. ساختارهای داده بدون قفل (Lockless Data Structures)

در پردازش های چند رشته‌ای، قفل (Lock) یک مکانیسم همگام‌سازی است که برای کنترل دسترسی به منابع مشترک استفاده می‌شود و اطمینان حاصل می‌کند که تنها یک رشته می‌تواند در یک زمان به یک منبع دسترسی داشته باشد. هنگامی که یک رشته دارای قفل می شود، رشته های دیگری که نیاز به دسترسی به همان منبع دارند باید منتظر بمانند تا قفل آزاد شود، که می تواند تاخیر ایجاد کند و عملکرد کلی سیستم را کاهش دهد.

DPDK از ساختارهای داده بدون قفل برای مدیریت حافظه استفاده می‌کند، که به چندین هسته CPU اجازه می‌دهد تا به طور همزمان به استخرهای حافظه دسترسی داشته باشند و آن‌ها را بدون سربار مکانیزم‌های قفل کردن تغییر دهند. این برای دستیابی به عملکرد بالا در سیستم های چند هسته ای بسیار مهم است، زیرا از تأخیر مرتبط با قفل جلوگیری می کند.[23][24][25]

ز. تخصیص و آزادسازی دسته ای:

DPDK از تخصیص و آزادسازی انبوه mbuf ها پشتیبانی می کند که اجازه می دهد چندین بافر در یک عملیات تخصیص یا آزاد شوند. این امر باعث کاهش سربارهای اضافی مرتبط با عملیات مکرر حافظه می شود و عملکرد کلی آن را در سرعت پردازش بسته ها بهبود می بخشد.

کتابخانه امنیتی DPDK:

کتابخانه امنیتی CryptoDev در DPDK یک جزء حیاتی برای تسریع عملیات رمزنگاری است. CryptoDev از ویژگی‌های شتاب سخت‌افزاری CPUهای مدرن برای رمزنگاری استفاده می‌کند و عملکرد و کارایی پروتکل‌های ارتباطی امن مانند IPsec را افزایش می‌دهد. یکی از ویژگی های مهم آن، پشتیبانی از طیف گسترده ای از الگوریتم های رمزنگاری، از جمله AES[۱۳]،SHA[۱۴]و HMAC[۱۵] است. این انعطاف‌پذیری به CryptoDev اجازه می‌دهد تا نیازهای مختلف رمزگذاری و احراز هویت را برآورده کند، و آن را به یک انتخاب همه‌کاره برای پیاده‌سازی پروتکل های یکپارچه امنیتی مانند IPsec در محیط‌های مختلف شبکه تبدیل کند.

[۱۳] Advanced Encryption Standard
[۱۴] Secure Hashing Algorithm
[۱۵] Hash-Based Message Authentication Code

علاوه بر این، CryptoDev از پردازش موازی وظایف رمزنگاری پشتیبانی می کند و از معماری های چند هسته ای برای تقویت بیشتر عملکرد IPsec استفاده می کند. به طور کلی، CryptoDev نقشی اساسی در افزایش کارایی، مقیاس‌پذیری و امنیت و پیاده‌سازی IPsec ایفا می‌کند و آن را به یک فناوری کلیدی برای زیرساخت‌های امنیتی شبکه های مدرن تبدیل می‌کند.[26][27]

### ۲,۳- امنیت پروتکل اینترنت (IPSec)

همانظور که اشاره شد IPsec مجموعه ای از پروتکل ها است که برای اطمینان از محرمانه بودن، یکپارچگی و صحت ارتباطات داده در شبکه های IP طراحی شده است. این امر از طریق دو پروتکل اصلی به دست می‌آید؛ سربرگ احراز هویت (AH)[۱۷]، که فقط احراز هویت را فراهم می کند، و محفظه امنیتی محموله (ESP)[۱۸]، که هم احراز هویت و هم محرمانگی بسته ها را فراهم می کند. این پروتکل‌ها می‌توانند در هر دو حالت انتقال، که تنها اطلاعات اصلی رمزگذاری می شود و اطلاعات IP دست نخورده باقی می ماند و یا حالت تونل، که کل بسته IP اصلی (شامل اطلاعات IP و اطلاعات اصلی ) رمزگذاری شده و در یک بسته IP جدید وارد می شود، کار کنند. انتخاب این پروتکل ها به الزامات امنیتی شبکه بستگی دارد.

[28]

IPSec امنیت سرتاسری را با رمزگذاری و احراز هویت بسته های IP فراهم می کند. ادغام با شتاب دهنده های رمزنگاری نرم افزار DPDK برای انجام کارآمد این عملیات، ضمن حفظ امنیت قوی، حداقل کاهش عملکرد را تضمین می کند.

ادغام IPsec با DPDK در یک شبکه SDN به دلیل ویژگی های امنیتی جامع IPsec و قابلیت های عملکرد بالای DPDK یک انتخاب قانع کننده است. با توجه به ماهیت قابل برنامه ریزیSDN ، ادغام یک پروتکل همه کاره و با پشتیبانی گسترده مانند IPsec تضمین می کند که سیاست های امنیتی می توانند به طور یکنواخت در کل شبکه، از صفحه کنترل تا صفحه داده، اجرا شوند. [29] این یکنواختی در محیط SDN ضروری است، جایی که مدیریت پویا و خودکار منابع شبکه و سیاست های امنیتی یک مزیت کلیدی است.

علاوه بر این، توانایی DPDK برای رسیدگی به پردازش بسته های با توان بالا کاملاً با خواسته های IPsec هماهنگ است. همانطور که گفته شد، DPDKراندمان پردازش بسته ها را از طریق تکنیک های منحصر به فرد مدیریت حافظه بالا می برد، تأخیر را به حداقل می رساند و توان عملیاتی را به حداکثر می رساند. این قابلیت ها برای مدیریت کارآمد سربار محاسباتی مرتبط با فرآیندهای رمزگذاری و رمزگشایی IPsec بسیار مهم هستند.

[30]

با استفاده از شتاب سخت‌افزاری و پردازش چند هسته‌ای، DPDK می‌تواند عملیات رمزنگاری فشرده مورد نیاز IPsec را موازی‌سازی کند و تضمین کند که عملکرد شبکه حتی تحت بارهای سنگین نیز بالا باقی می‌ماند.

در SDN، ترکیب IPsec و DPDK یک راه حل مقیاس پذیر و امن ارائه می دهد. کنترل متمرکز و برنامه‌ریزی SDN استقرار پویا سیاست‌های IPsec را تسهیل می‌کند و تضمین می‌کند کانال‌های ارتباطی امن در زمان واقعی ایجاد و نگهداری می‌شوند. پردازش بسته با کارایی بالا DPDK

[۱۷] Authentication Header

[۱۸] Encapsulating Security Payload

تضمین می کند که این اقدامات امنیتی سرعت یا پاسخگویی شبکه را به خطر نمی اندازد. IPsec و DPDK با هم، هم افزایی قدرتمندی را ارائه می‌کنند و امنیت قوی را بدون قربانی کردن توان عملیاتی بالا و تأخیر کم مورد نیاز شبکه‌های SDN مدرن ارائه می‌کنند. این باعث می شود IPsec بهترین پروتکل برای ادغام با DPDK برای ایمن سازی محیط های SDN باشد.

این اطمینان می دهد که شبکه می تواند بارهای ترافیکی افزایش یافته را به طور موثر مدیریت کند و آن را به انتخابی ایده آل برای ایمن سازی زیرساخت های SDN تبدیل می کند.

### ۳,۳- پردازش بسته برداری (VPP)

VPP یک پلت فرم منبع باز پردازش برداری بسته در شبکه مبتنی بر نرم افزار است. VPP یک چارچوب برای پردازش بسته ارائه می دهد که از پردازش برداری برای رسیدگی به چندین بسته به طور همزمان استفاده می کند که نتیجه آن عملکرد بالا است. مقیاس پذیری VPP تأثیر عمیقی بر معماری یک شبکه دارد و بر طراحی، عملکرد، انعطاف پذیری و کارایی کلی آن تأثیر می گذارد. [31][32] یکی از مزایای پردازی برداری کاهش خطاهای حافظه می باشد.

حافظه I-cache[۹](کش دستورالعمل) نوعی حافظه پرسرعت است که در طراحی پردازنده کاربرد دارد و برای ذخیره دستورالعمل های مکرر استفاده می شود. این نوع از حافظه بخشی از سلسله مراتب حافظه است که برای افزایش عملکرد کلی سیستم طراحی شده است.

Thrashing حافظه I-cache به دلیل پردازش هایی ایجاد می شود که به دلیل استفاده بیش از حد از منابع یا تداخل در سیستم حافظه cache توسط دیگر پردازش ها و دستورات مکرر، امکان ادامه مسیر را ندارند. هنگامی که تعداد دفعات فراخوانی بسته ها زیاد شود، باعث می شود که CPU به طور مداوم دستورالعمل های جدید را بارگذاری کند، این امر باعث رخ دادن thrashing می شود.

پردازش بسته نردبانی (Scalar) معمولاً یک بسته را در یک زمان پردازش می کند، یک تابع مدیریت وقفه یک بسته واحد را از یک رابط شبکه می گیرد و آن را از طریق مجموعه ای از توابع پردازش می کند. پردازش بسته نردبانی ساده است، اما در مورد زیر ناکارآمد است؛

هنگامی که تعداد دفعات فراخوانی بسته ها زیاد شود، باعث می شود که CPU به طور مداوم دستورالعمل های جدید را بارگذاری کند، این امر باعث رخ دادن thrashing می شود. در این مدل، هر بسته دارای مجموعه‌ای از خطاهای I-cache است.

```
+---> fooA(packet1) +---> fooB(packet1) +---> fooC(packet1)
+---> fooA(packet2) +---> fooB(packet2) +---> fooC(packet2)
...
+---> fooA(packet3) +---> fooB(packet3) +---> fooC(packet3)
```

**شکل ۲. شماتیک پردازش بسته ها در پردازش بسته نردبانی**

[۹] Instruction cache

در مقابل، پردازش بسته برداری (Vectorized)، چندین بسته را در یک زمان پردازش می کند که به آنها بردار بسته ها می گویند. یک تابع مدیریت وقفه، بردار بسته ها را از یک رابط شبکه می گیرد و بردارها را از طریق مجموعه ای از توابع پردازش می کند


شکل ۳. شماتیک پردازش بسته ها در پردازش بسته برداری

پردازش بسته برداری مشکل thrashing در I-cache را که در بالا توضیح داده شد، با کاهش بارگذاری متعدد در I-cache به حداقل می رساند. کاهش خطاهای حافظه در پردازش بسته ها، از نکات کلیدی افزایش راندمان پردازش داده در یک شبکه می باشد. [33]

VPP حجم کاری پردازش بسته را از فضای هسته به فضای کاربر منتقل می کند و سپس بردار بسته ها را از طریق یک گراف پردازش بسته پردازش می کند. به جای پردازش هر بسته از طریق کل گراف پردازش، و سپس واکشی بسته دوم و پردازش آن از طریق کل نمودار گراف، VPP قبل از رفتن به گره گراف بعدی، بردار بسته ها را به طور کامل از طریق گره گراف اول پردازش می کند. بسته اول در بردار، حافظه I-cache را رزرو و آماده می کند، بنابراین بسته های باقی مانده را می تواند بسیار سریع پردازش کند، که وقفه و سربار پردازش هر بسته بعدی در بردار به شدت کاهش می یابد. این امر منجر به عملکرد بسیار بالا برای پردازش یک بسته واحد و همچنین عملکرد قابل اعتماد در پردازش تعداد زیادی بسته در طول زمان می شود. علاوه بر این، VPP اغلب در هر فرآیند بسته های بعدی را از قبل واکشی می کند، و اطمینان حاصل می کند که CPU در زمانی که بسته بعدی ازRAM[۲] واکشی می شود، متوقف نمی شود. در نتیجه، توان عملیاتی بالا و تاخیر به طور مداوم پایین است؛ شبکه پاسخگو باقی می ماند و قادر به مدیریت انتقال داده در مقیاس بزرگ است که برای برنامه هایی که به پهنای باند بالا و تاخیر کم نیاز دارند بسیار مهم است [34] .

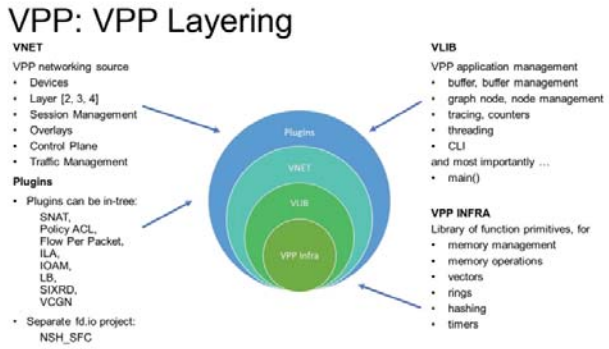
[۲] Random Access Memory

مقیاس پذیری VPP امکان گسترش یکپارچه شبکه را فراهم می کند. مقیاس پذیری افقی VPP این امکان را فراهم می آورد که بتوان شبکه را به صورت افقی با افزودن گره های پردازشی یا هسته های CPU بیشتر به زیرساخت موجود گسترش داد. همچنین طراحی مدولار VPP از افزودن تدریجی قابلیت ها و قابلیت های جدید با رشد شبکه پشتیبانی می کند. [35]

VPP با بهینه سازی استفاده از منابع موجود و کاهش نیاز به ارتقاء مداوم سخت افزار به کارایی هزینه کمک می کند. به جای جایگزینی کل سیستم ها، شبکه را می توان با افزودن هسته ها یا گره های CPU بیشتر مقیاس پذیر کرد که اغلب مقرون به صرفه تر است. استفاده بهتر از هسته های CPUو حافظه، نیاز به سخت افزار اضافی را کاهش می دهد و منجر به کاهش هزینه های سرمایه و عملیاتی می شود. این کارایی هزینه، حفظ و گسترش شبکه در طول زمان را امکان پذیرتر می کند.

استفاده از پردازش بسته برداری (VPP) در کنار DPDK مزایای قابل توجهی در افزایش عملکرد و انعطاف پذیری شبکه دارد. VPP از قابلیت‌های پردازش بسته‌های پرسرعت DPDK برای دستیابی به عملیات شبکه با تأخیر کم و بازده بالا استفاده می‌کند، در حالی که یک چارچوب بسیار ماژولار و توسعه‌پذیر برای ایجاد عملکردهای شبکه SDN سفارشی سازی شده ارائه می‌دهد. این ترکیب امکان ایجاد توابع مسیریابی، سوئیچینگ و فایروال را فراهم می کند که می توانند به صورت پویا برای برآوردن نیازهای شبکه خاص سازگار شوند. علاوه بر این، ادغام VPP با DPDK استفاده کارآمد از منابع سخت‌افزاری را تضمین می‌کند و راه‌حل‌های شبکه‌ای مقیاس‌پذیر و کارآمد را قادر می‌سازد که می‌توانند محیط‌های پیچیده و سخت را به راحتی مدیریت کنند. VPP و DPDK با هم یک ابزار قدرتمند برای ایجاد زیرساخت شبکه قوی، با کارایی بالا و ایمن در محیط‌های SDN ارائه می‌کنند. [36]

در زیر شمای کلی لایه های VPP در سطح بالا نشان داده شده است.[37]




**شکل ۴. لایه های VPP**

VPP Infra: لایه زیرساختVPP ، که حاوی کد منبع کتابخانه هسته است. این لایه عملکردهای حافظه را انجام می دهد، با بردارها و حلقه ها کار می کند، جستجوهای کلیدی را در جداول انجام می دهد و با تایمرها برای ارسال گره های گراف کار می کند.

VLIB: کتابخانه پردازش برداری. لایه vlib همچنین توابع مدیریت برنامه های مختلف را مدیریت می کند: بافر، حافظه و مدیریت گره گراف، نگهداری و ارسال شمارنده، مدیریت رشته، ردیابی بسته، پیاده سازی رابط خط فرمان اشکال زدایی.

VNET: با رابط شبکه VPP (لایه‌های ۲، ۳ و ۴) کار می‌کند و مدیریت جلسه و ترافیک را انجام می‌دهد و با دستگاه‌ها و صفحه کنترل داده کار می‌کند.

Plugins - همانطور که در نمودار بالا ذکر شده است، شامل مجموعه ای غنی از افزونه های صفحه داده است.

## ۴- معماری سیستم پیشنهادی

### ۱,۴ - بررسی اجمالی

معماری پیشنهادی با پیاده سازی IPSec در DPDK به منظور پردازش بسته های رمزگذاری شده با راندمان بالا و بهره گیری از VPP برای تخصیص بهینه منابع و مسیریابی پویا، امکان فراهم سازی زیرساختی قابل اطمینان در یک شبکه SDN را فراهم می آورد. این معماری از قابلیت‌های پردازش بسته با کارایی بالا DPDK ، ویژگی‌های امنیتی IPsec و عملکردهای شبکه‌ای ماژولار و انعطاف‌پذیر ارائه شده توسط VPP بهره می‌برد. لازم به ذکر است پیاده سازی این معماری شامل جزئیات بسیار زیادی است که از حوصله این مقاله خارج است و صرفا الگوی کلی آن مورد بررسی قرار می گیرد.

### ۴,۲- اجزای سیستم

کنترلرSDN:

- قوانین جریان و سیاست های امنیتی را مدیریت می کند.
- برای استقرار و پیکربندی عملکرد یک شبکه پویا با VPP ارتباط برقرار می کند.

DPDK و کتابخانه CryptoDev :

- عملیات ورودی/خروجی بسته ها را به طور موثر مدیریت می کند.
- از کتابخانه CryptoDev برای بارگذاری وظایف رمزنگاری به شتاب‌دهنده‌های سخت‌افزاری استفاده می‌کند و تضمین می‌کند که عملیات IPsecعملکرد را کاهش نمی‌دهد.

VPP:

- عملکردهای شبکه مانند مسیریابی، سوئیچینگ و تونل های IPsec را پیاده سازی می کند.
- قابل تنظیم از طریق کنترلر SDN برای انطباق پویا با تغییرات شبکه.

### ۳,۴- بررسی ادغام تکنولوژی های نرم افزاری پیشنهادی:

الف. پیاده سازی IPSec در DPDK

DPDK بسته های ورودی و خروجی را پردازش می کند و از کتابخانه CryptoDev برای مدیریت عملیات IPsec استفاده می کند.

پیاده سازی IPsec مستقیماً در DPDK بدون استفاده از اپلیکیشن هایی مانند Strongswan از نظر فنی امکان پذیر است اما به دلایل مختلف توصیه نمی شود. Strongswan یک پیاده سازی جامع و آزمایش شده از پروتکل های IPsec و IKE[۲] را ارائه می دهد و ویژگی هایی مانند احراز هویت، رمزگذاری و مدیریت کلید را ارائه می دهد که برای ایمن سازی ارتباطات شبکه ضروری هستند. [38][39] در حالی که DPDK برای پردازش بسته های پرسرعت عالی است، در درجه اول بر روی عملیات صفحه داده تمرکز دارد و فاقد اجرای پروتکل های سطح بالاتر لازم برای IPsec است. تلاش برای پیاده سازی IPsec از ابتدا در DPDK به تلاش قابل توجهی در زمینه توسعه، آزمایش و نگهداری نیاز دارد که به طور بالقوه آسیب پذیری های امنیتی و محدودیت های عملکرد را به بار می آورد. بنابراین، استفاده از Strongswan در کنار DPDK به شما این امکان را می دهد که از پردازش بسته با کارایی بالا و امنیت IPsec قوی بهره مند شوید و راه حل کارآمدتر و قابل اعتمادتری برای ارتباطات شبکه ایمن ارائه کنید.

strongSwan یک راه حل VPN مبتنی بر IPsec منبع باز است که برای ایمن سازی ارتباطات در شبکه های بالقوه ناامن مانند اینترنت طراحی شده است. این پروتکل از هر دو پروتکل IKEv1 و IKEv2 برای تبادل کلید پشتیبانی می کند، مکانیزم های رمزگذاری و احراز هویت قوی را ارائه می دهد و با طیف گسترده ای از سیستم عامل ها از جمله لینوکس، اندروید، FreeBSD، macOS و ویندوز سازگار است. معماری مدولار strongSwan که شامل افزونه‌های متعددی است، به آن اجازه می‌دهد تا با پشتوانه‌های مختلف احراز هویت و کتابخانه‌های رمزنگاری ادغام شود و آن را به انتخابی انعطاف‌پذیر و قابل تنظیم برای پیاده‌سازی اتصالات VPN ایمن در محیط‌های شبکه های گوناگون تبدیل کند.

strongSwan: سیاست هایIPsec ، تونل ها و کلیدهای رمزگذاری را پیکربندی و مدیریت می کند.

در اینجا مثالی از پیکربندی IPsec با strongSwan برای استفاده از حالت تونل آورده شده است. این پیکربندی خارج از کد DPDK انجام می شود، اما برای عملکرد صحیح IPsec ضروری است.

```
config setup
charondebug="ike 2, knl 2, cfg 2"

conn %default
keyexchange=ikev2
ike=aes256-sha256-modp1024!
esp=aes256-sha256!

conn tunnel
left=192.0.2.1
leftsubnet=10.0.0.0/24
right=198.51.100.1
rightsubnet=10.1.0.0/24
auto=start
type=tunnel
```

[۲] Internet Key Exchange

:left

این آدرس IP عمومی سمت محلی (local) تونل VPN است.

leftsubnet: این نشان دهنده محدوده آدرس های IP خصوصی در شبکه محلی است که از طریق تونل VPN هدایت می شوند. در این مورد، 10.0.0.0/24است، به این معنی که تمام دستگاه های دارای آدرس IP از 10.0.0.1 تا 10.0.0.254 از VPN استفاده می کنند.

right: این آدرس IP عمومی سمت راه دور تونل VPN است.

rightubnet: مشابه leftsubnet، این محدوده آدرس IP خصوصی در شبکه راه دور را که از طریق VPN قابل دسترسی خواهد بود، تعریف می کند. در اینجا، 10.1.0.0/24 است (آدرس 10.1.0.1 تا 10.1.0.254).

برای اطمینان از اینکه strongSwan از DPDK برای پردازش بسته‌ها استفاده می‌کند، باید قابلیت‌های DPDK را در مسیر داده‌های strongSwan ادغام کردکه معمولاً شامل پیکربندی VPP (پردازش بسته برداری) برای مدیریت کارآمد ارسال بسته است. در اینجا یک رویکرد ساختاریافته برای دستیابی به این هدف وجود دارد؛ باید اطمینان حاصل کرد که پردازش بسته strongSwan از VPP استفاده می کند که توسط DPDK برای مدیریت سریع بسته پشتیبانی می شود.

برای این منظور باید یک اسکریپت ایجاد شود و VPP پیکربندی طوری پیکربندی شود تا رابط های شبکه و پردازش IPsec را مدیریت کند. در زیر یک نمونه اسکریپت  برای این منظور آمده است:

```
vppctl create tap id 0 host-if-name vpp-tap0

vppctl set int state tap-0 up

vppctl set int ip address tap-0 10.0.0.1/24

vppctl ip route add 0.0.0.0/0 via 10.0.0.2 tap-0


# Configure DPDK interfaces

vppctl create host-interface name eth0

vppctl set int state host-eth0 up

vppctl set int ip address host-eth0 192.0.2.1/24


vppctl create host-interface name eth1

vppctl set int state host-eth1 up

vppctl set int ip address host-eth1 198.51.100.1/24
```

این اسکریپت VPP را برای ارسال بسته با استفاده از DPDK تنظیم می کند. رابط‌های tap و DPDK را ایجاد و پیکربندی می‌کند، آدرس‌های IP را به آنها اختصاص می‌دهد و آنها را در حالت UP قرار می‌دهد. علاوه بر این، یک مسیر پیش فرض برای رابط tap اضافه می کند. این پیکربندی VPP را برای مدیریت کارآمد ترافیک شبکه آماده می کند و امکان ادغام یکپارچه با سایر اجزای شبکه را فراهم می کند.

## ۵- نتیجه گیری

سیستم پیشنهادی IPSec، DPDK و VPP را در چارچوب SDN ادغام می‌کند تا ارتباطی ایمن و با راندمان بالا را فراهم کند. اگرچه هیچ آزمون آزمایشی انجام نشده است، مبانی نظری و نقاط قوت تکنولوژیکی این اجزا قویاً نشان می دهد که سیستم به طور موثر عمل خواهد کرد. DPDK برای تسریع پردازش بسته با دور زدن هسته و دسترسی مستقیم به سخت افزار طراحی شده است. این منجر به کاهش قابل توجه تأخیر و افزایش توان عملیاتی می‌شود .این ادغام از نظر تئوری الزامات محیط های شبکه مدرن را برآورده می کند و تضمین می کند که عملکرد و امنیت در سطوح بهینه حفظ می شوند. نقاط قوت تکنولوژیکی این تکنولوژی ها، یک مورد قانع کننده برای ادغام آنها در دستیابی به زیرساخت SDN کارآمد و ایمن است.

سیستم پیشنهادی از ادغام IPSec، DPDK و VPP در چارچوب SDN برای دستیابی به ارتباطی ایمن و با راندمان بالا استفاده می کند. ویژگی های ذاتی موضوعات مورد پژوهش نشان دهنده اثربخشی بالقوه آن است. DPDK به دلیل توانایی خود در دور زدن هسته و تکنیک های مدیریت حافظه برای ارائه پردازش سریع بسته، کاهش تأخیر و افزایش قابل توجه توان عملیاتی شناخته شده است. با یکپارچه سازیDPDK ، سیستم می تواند حجم زیادی از داده ها را با حداقل تاخیر مدیریت کند، که برای حفظ محیط های شبکه با کارایی بالا بسیار مهم است.

IPSec، با پروتکل های رمزگذاری و احراز هویت یکپارچه و قوی تضمین می کند که داده های عبوری از شبکه ایمن باقی می مانند. هنگامی که IPSecاز طریق کتابخانه Cryptodev با DPDK ادغام می شود، می تواند از شتاب سخت افزاری و عملیات رمزنگاری بهینه بهره مند شود و امنیت و عملکرد را افزایش دهد. این هم افزایی که انتقال ایمن داده ها را بدون به خطر انداختن سرعت امکان پذیر می کند، یک نیاز حیاتی برای شبکه های SDN مدرن است که در آن یکپارچگی و محرمانه بودن داده ها از اهمیت بالایی برخوردار است.

VPP با ارائه یک چارچوب پردازش بسته بسیار مقیاس پذیر و قابل برنامه ریزی، لایه دیگری از کارایی و انعطاف پذیری را اضافه می کند. این امکان ایجاد قابلیت های مسیریابی و سوئیچینگ سفارشی شده متناسب با نیازهای شبکه را فراهم می کند. هنگامی که VPP با DPDK ترکیب می شود، می تواند جریان های داده با سرعت بالا را به طور موثر اداره کند و از ماهیت پویا و سازگار SDN پشتیبانی کند. این ادغام تضمین می کند که شبکه می تواند به سرعت به الگوهای ترافیکی و الزامات تغییر پاسخ دهد و عملکرد و امنیت مطلوب را حفظ کند.

به طور خلاصه، معماری سیستم پیشنهادی نوید ارائه یک پروتکل ارتباطی ایمن و با توان عملیاتی بالا را در یک شبکه SDN می دهد. پیاده‌سازی‌های عملی و اعتبارسنجی‌های آزمایشی آینده، قابلیت این رویکرد یکپارچه را بیشتر تقویت خواهد کرد و به طور بالقوه استانداردهای جدیدی را برای عملکرد و امنیت در ارتباطات شبکه تعیین می‌کند.

## ۶- پیشنهادات

از آنجایی که تکنولوژی به کار رفته در این پروژه با رویکرد مفاهیم SDN پیاده سازی شده است، لذا این پروژه پایه ای خواهد بود برای طراحی یک شبکه امن کاملا مبتنی بر SDN. برای این منظور، نتیجه نهایی این پروژه می تواند در اختیار تیمی از برنامه نویسان شبکه قرار گیرد تا با پیاده سازی پروتکل های مسیربابی (Routing) بر روی چهارچوب طراحی شده در این پروژه، آن را تبدیل به یک چهارچوب ارتباط امن با قابلیت مسیریابی بر اساس نیاز شبکه های موجود در کشور، تبدیل نمایند. چهارچوب مذکور می تواند یک چهارچوب امن کاملا بومی انتقال داده با سرعت بالا، در شبکه ای سفارشی سازی شده باشد. لازم به ذکر است که توسعه فاز مسیریابی در چهارچوب نهایی این پروژه، خود پروژه ای بزرگ و زمانبر خواهد بود و نیاز به تیمی متخصص خواهد داشت که پس از پیاده سازی آن، در واقع شبکه امن بومی ای با قابلیت تعریف پروتکل های مسیریابی مورد نیاز مبتنی بر تکنولوژی SDN در دسترس خواهد بود که یکی از فواید آن، حذف روترها و سوئیچ های سخت افزاری از شبکه و جایگزینی آنها با این چهارچوب نرم افزاری می باشد.

### رویکردهای هوش مصنوعی برای دستیابی به امنیت شبکه در شبکه های SDN

در حال حاضر، بکارگیری رویکردهای مختلف هوش مصنوعی در اکثر سیستم‌های هوشمند موجود از اهمیت بالایی برخوردار هستند و از جمله موارد قابل بحث، امکان بهبود عملکرد شبکه های کامپیوتری توسط هوش مصنوعی است. ادغام هوش مصنوعی با SDN به ساخت برنامه‌های شبکه به صورت پیشرفته تر کمک می‌کند. مطالعه‌ای که توسط نویسندگان در [40] انجام شده است نشان می‌دهد بکارگیری الگوریتم یادگیری تقویتی[۲۲] در شبکه های SDN می تواند برای نظارت و جمع‌آوری اطلاعات لحظه ای شبکه به منظور محاسبه مسیر بهینه، موثر باشد. کنترلر به صورت دوره ای تصمیم گیرنده ای را که زمان انتخاب مسیر جدید و زمانی را که سرور تقاضای تغییر مسیر انتقال داده‌ها را درخواست می کند را تعیین می کند. همچنین در [41]، نویسندگان از الگوریتم یادگیری- Q [۲۳]در راستای بهبود سرعت انتقال داده‌ها در SDN استفاده کرده‌اند. با این حال، SDN از الگوریتم‌های اکتشافی استفاده می‌کند که همیشه مسیر بهینه را انتخاب نمی‌کنند و این مقاله به ایجاد سه الگوریتم Q-Routing با استفاده از توپولوژی شبکه SDN و مِش[۲۴]پرداخته است. دو الگوریتم هر کدام از یک متریک شبکه (تأخیر و پهنای باند) و الگوریتم سوم از معیارهای چندگانه استفاده کردند. نتایج پژوهش [41]، نشان داد که الگوریتم‌های تک متریک Q-Routing به طور میانگین عملکرد خوبی داشتند در حالی که Q-Routing با معیارهای شبکه چندگانه اینگونه نبوده است. همچنین نتایج نشان داده که Q-Routing قادر به محاسبه‌ی سریعتر مسیرها

2 Reinforcement Learning (RL)
2 Q- leraning
2 Mesh
2
3
4

نسبت به الگوریتم کوتاه‌ترین مسیر[۵]در شبکه های استاتیک و پویا بوده است. این نشان می دهد بهره گیری از تکنیک های هوش مصنوعی درشبکه های SDN، می تواند راندمان معماری پیشنهادی در این مقاله را به طرز چشمگیری بالاتر ببرد.

[2] K-Shorttest Path

[5]

منابع:

[1]. A. Haggag, “Network optimization for improved performance and speed for SDN and security analysis of SDN vulnerabilities,” *International Journal of Computer Networks and Communications Security*, vol. 5, pp. 83–90, May 2019.
[2]. A. Dawod Al-Ani, N. Ibrahim Abdullah, “Software defined networks challenges and future direction of research,” *International Journal of Research*, vol. 1, pp. 618-629, Jan. 2019.
[3]. A. Coly, M. Mbaye, Gaston Berger University, Saint-Louis, and Gaston Berger University, Saint-Louis, “S-SDS : a framework for security deployment as service in software defined networks,” May 2019.
[4]. G. Lopez-Millan, R. Marin-Lopez, and F. Pereniguez-Garcia, “Towards a standard SDN-based IPsec management framework,” *journal of Computer Standards & Interfaces*, vol. 66 , May 2019.
[5]. “15. Poll Mode Driver — Data Plane Development Kit 24.03.0 documentation.”, DPDK.org, http://doc.dpdk.org/guides-24.03/prog_guide/poll_mode_drv.html
[6]. L. Linguaglossa a, D. Rossi a, S. Pontarelli b, Telecom ParisTech, CNIT and University of Rome Tor Vergata, and Cisco Systems, Inc., “High-speed data plane and network functions virtualization by vectorizing packet processing,” *Journal of Computer Networks*, Vol. 149, pp. 187-199, Feb. 2019.
[7]. J. Pak and K. Park, “A High-Performance implementation of an IoT system using DPDK,” Journal of *Applied Sciences*, vol. 8, no. 4, p. 550, Apr. 2018.
[8]. A. Belkhiri a, M. Pepin, M. Bly, Polytechnique Montréal, and Ciena Inc., “Performance analysis of DPDK-based applications through tracing,” *Journal of Parallel and Distributed Computing*, vol. 173, pp. 1-19, Mar 2023.
[9]. S. Kaur, K. Kumar, N. Aggarwal, and University Institute of Engineering and Technology (UIET), Panjab University, Chandigarh, India, “A review on P4-Programmable data planes: Architecture, research efforts, and future directions,” *The International Journal for the Computer and Telecommunications*, vol. 170, pp. 109-129, Mar 2021.
[10]. T. Döring, H. Stubbe, K. Holzinger, and Chair of Network Architectures and Services, Department of Informatics, Technical University of Munich, Germany, “SmartNICs: Current trends in research and industry,” Seminar IITM WS 20/21, May2021.
[11]. X. Yang, L. Wang, “SDN Load Balancing Method based on K-Dijkstra,” *International Journal of Performability Engineering*, vol. 14, no. 4, pp. 709-716, Apr. 2018.
[12]. A. Kumar, D. Anand, Chandigarh University Mohali, and Chandigarh University Mohali, “Load balancing for software defined network using machine learning,” *Turkish Journal of Computer and Mathematics Education*, vol. 12, no. 12, pp. 527–535, Apr. 2021.
[13]. D. Todorov, H. Valchanov and V. Aleksieva, "Load Balancing model based on Machine Learning and Segment Routing in SDN," *2020 International Conference Automatics and Informatics (ICAI)*, Varna, Bulgaria, pp. 1-4, Oct 2020.
[14]. J. Spooner, Sh. Ying Zhu, Department of Computing and Mathematics, University of Derby, and Department of Computing and Mathematics, University of Derby, “A review of solutions for SDN-Exclusive security issues,” *International Journal of Advanced Computer Science and Applications*, Vol. 7, no. 8. 2016.
[15]. A. Pradhan and R. Mathew, “Solutions to vulnerabilities and threats in Software defined networking (SDN),” *Third International Conference on Computing and Network Communications*, vol. 171, pp. 2581–2589, Jan. 2020.
[16]. X. Yang, D. Wang, W. Tang, W. Feng, and C. Zhu, “IPSEC Cryptographic Algorithm Invocation Considering Performance and Security for SDN Southbound Interface Communication,” *IEEE Access*, vol. 8, pp. 181782–181795, Jan. 2020.
[17]. “Memory in DPDK, Part 1: General Concepts - DPDK,”, DPDK.org, Aug 2019. https://www.dpdk.org/memory-in-dpdk-part-1-general-concepts/
[18]. “Efficient data transfer through zero copy”, IBM Developer. https://developer.ibm.com/articles/j-zerocopy/
[19]. “10. IPv4 Multicast Sample Application — Data Plane Development Kit 24.03.0 documentation.”, DPDK.org, http://doc.dpdk.org/guides-24.03/sample_app_ug/ipv4_multicast.html
[20]. J. Kubálek, Brno University of Technology, “High-speed DMA packet transfer in system DPDK,” May 2018.

[21]. "13. Mempool Library — Data Plane Development Kit 24.03.0 documentation.", DPDK.org, https://doc.dpdk.org/guides/prog_guide/mempool_lib.html
[22]. "14. Mbuf Library — Data Plane Development Kit 24.03.0 documentation.", DPDK.org https://doc.dpdk.org/guides/prog_guide/mbuf_lib.html
[23]. A. Baumstark and C. Pohl, "Lock-free data structures for data stream processing," *Datenbank-Spektrum Journal*, vol. 19, pp. 209–218, Oct 2019.
[24]. J. Kong, "DPDK Optimization on arm," *Tools, Software and IDEs - Arm Community*, May 2022. https://community.arm.com/arm-community-blogs/b/tools-software-ides-blog/posts/dpdk-optimization-on-arm
[25]. "10. RCU Library — Data Plane Development Kit 24.03.0 documentation.", DPDK.org, https://doc.dpdk.org/guides/prog_guide/rcu_lib.html
[26]. "21. Cryptography Device Library — Data Plane Development Kit 24.03.0 documentation.", DPDK.org, https://doc.dpdk.org/guides/prog_guide/cryptodev_lib.html
[27]. "27. Security Library — Data Plane Development Kit 24.03.0 documentation.", DPDK.org, https://doc.dpdk.org/guides/prog_guide/rte_security.html
[28]. E. Barker, Q. Dang, S. Frankel, Karen Scarfone, and Paul Wouters, "Guide to IPSEC VPNs," National Institute of Standards and Technology, Jun. 2020.
[29]. M. Vajaranta, J. Kannisto, J. Harju, and Tampere University of Technology, "IPSEC and IKE as functions in SDN controlled network," *11th International Conference on Network and System Security*, Helsinki, Finland, pp. 521-530, Aug 2017.
[30]. O. Abolade, A. Okandeji, A. Oke, M. Osifeko, and A. Oyedeji, "Overhead effects of data encryption on TCP throughput across IPSEC secured network," Journal of *Scientific African*, vol. 13, p. e00855, Sep. 2021.
[31]. "What is vector packet processing? — Vector Packet Processor 01 documentation.", FD.io, https://fdio-vpp.readthedocs.io/en/latest/overview/whatisvpp/what-is-vector-packet-processing.html
[32]. D. Barach, L. Linguaglossa, D. Marion, P. Pfister, S. Pontarelli, and D. Rossi, "High-Speed software data plane via vectorized packet processing," *IEEE Communications Magazine*, vol. 56, no. 12, pp. 97–103, Dec. 2018.
[33]. "Scalar vs Vector packet processing — The Vector Packet Processor v24.06-rc1-0-gb3304b2b7 documentation.", FD.io, https://s3-docs.fd.io/vpp/24.06/aboutvpp/scalar-vs-vector-packet-processing.html
[34]. "VPP Technology." https://fd.io/technology/
[35]. "The Packet Processing Graph — The Vector Packet Processor v22.10-0-g07e0c05e6 documentation.", FD.io, https://docs.fd.io/vpp/22.10/aboutvpp/extensible.html?highlight=modular
[36]. "FD.io doubles packet throughput performance to terabit levels - Linux Foundation,", The Linux Foundation, Sep.13, 2022. https://www.linuxfoundation.org/press/press-release/fd-io-doubles-packet-throughput-performance-to-terabit-levels[37]. "Software Architecture — The Vector Packet Processor v24.06-rc1-0-gb3304b2b7 documentation.", FD.io, https://s3-docs.fd.io/vpp/24.06/developer/corearchitecture/softwarearchitecture.html
[38]. strongSwan Project, "StrongSwan - documentation." https://strongswan.org/documentation.html
[39]. S. Bae, Y. Chang, H. Park, M. Kim, Y. Shin, and School of Cybersecurity, Korea University, "A Performance Evaluation of IPsec with Post-Quantum Cryptography,", *25th International Conference on Information Security and Cryptology*, pp. 249-266, 2022.
[40]. M. A. Jameel, T. Kanakis, S. Turner, A. Al-Sherbaz, and W. S. Bhaya, "A Reinforcement Learning-Based Routing for Real-Time Multimedia Traffic Transmission over Software-Defined Networking," *International Journal of Electronics*, vol. 11, no. 15, p. 2441, Aug. 2022.
[41]. D. Harewood-Gill, T. Martin, Reza Nejabati, "The Performance of Q-Learning within SDN Controlled Static and Dynamic Mesh Networks,", *2020 6th IEEE Conference on Network Softwarization (NetSoft), Ghent, Belgium, pp. 185-189*, 2020.