

دانشکده مهندسی کامپیوتر

سمینار کارشناسی ارشد گرایش رایانش امن

عنوان:

|  |
| --- |
| **ارائه رويكرد تطبیق‌پذير با تنوع ترافیكي شبكه‌هاي پهن‌باند براي شناسايي حملات منع‌ خدمت توزيع‌شده**  **An Adaptive Approach with Variety Characteristic of High-Bandwidth Networks for Distributed Denial of Service Attacks Detection** |
|  |

نگارش:

**روح‌الله جهان‌افروز**

**۴۰۰۲۱۰۷۵۵**

استاد راهنما:

**دکتر رسول جلیلی**

استاد ممتحن داخلی:

**دکتر امیرحسین جهانگیر**

**بهمن ۱۴۰۱**

چکیده**:**

**بسیاری از برنامه‌های کاربردی[[1]](#endnote-2) امروزی از پروتکل‌های یکسان و مشترکی برای تبادل اطلاعات استفاده می‌کنند. برنامه‌های پیام‌رسان و مرورگر‌های وب از بسته‌های مبتنی بر پروتکل اچ.تی.تی.پی/اس[[2]](#endnote-3) برای تبادل اطلاعات استفاده می‌کنند، با این تفاوت‌ که در برنامه‌های پیام‌رسان با ارسال تعداد معینی از بسته‌های اچ.تی.تی.پی/اس در مقایسه با مرورگر‌های اینترنتی، نرخ متفاوتی از بسته‌ها را در پاسخ دریافت خواهیم کرد. لذا با ظهور برنامه‌های کاربردی مختلف شاهد بروز تنوع ترافیکی بر روی پروتکل‌های مختلف و رفتار‌های متفاوت در ترافیک شبکه هستیم. در شبکه‌های پهن‌باند[[3]](#endnote-4) با افزایش نرخ ترافیک و وجود تنوع پروتکلی زیاد، چالش‌های امنیتی نظیر تشخیص حملات‌ منع ‌خدمت[[4]](#endnote-5)، که به دلیل سادگی در پیاده‌سازی و تاثیر بسیار‌مخرب یک تهدید جدی به حساب می‌آیند، افزایش پیدا کرده‌است.** **در دهه‌های گذشته محققان روش‌های شناسایی بسیاری را برای حملات منع خدمت توزیع‌شده[[5]](#endnote-6) پیشنهاد کرده‌اند. عدم تطبیق پذیری و مقیاس‌پذیری برای استفاده در شبکه‌های پهن‌باند، از متداول‌ترین مشکلات این روش‌ها هستند. لذا برای شناسایی صحیح حملات منع ‌خدمت در شبکه‌های پهن‌باند نیاز به یک رویکردی است که شامل دو ویژگی پردازش جامع[[6]](#endnote-7) به معنای پردازش تمامی بسته‌ها و تطبیق‌پذیری[[7]](#endnote-8) به معنای قابلیت تطبیق‌پذیری با تنوع ترافیکی باشد.**

**در این پژوهش ضمن بررسی کارهای مشابه صورت گرفته در این زمینه، قصد‌ داریم رویکردی تطبیق‌پذیر با تنوع ترافیکی موجود ‌در شبکه‌های پهن‌باند برای شناسایی حملات منع ‌خدمت توزیع‌شده معرفی نماییم که ویژگی پردازش جامع ترافیک را نیز شامل شود. روش‌ پیشنهادی جریان‌ها را بر اساس اینکه برای کدام کاربرد می‌باشند دسته‌بندی کرده و برمبناي رفتار عادی ترافیک هر برنامه کاربردی، ترافیک‌هاي متخاصم را تشخیص می‌دهد. به دلیل اینکه از الگوریتم‌ها و داده‌ساختارهای فشرده و سبک با قابلیت جستجوی سریع استفاده می‌شود، سرعت بالا و استفاده بهینه از حافظه تضمین می‌شود. همچنین در روش پیشنهادی از ابزارهای تسریع عملیات پردازش بسته که در سالیان اخیر بسیار مورد استقبال قرار گرفته‌است، استفاده می‌شود و بدين صورت می‌توان سرعت پردازش بسته‌ها را تسریع‌ بخشيد که منجر به پردازش جامع تمامی بسته‌های ترافیک عبوري شبکه خواهد شد. در انتها کارایی روش ارائه شده در مقایسه با برخی دیگر از راهکارهای موجود و با ‌در نظرگرفتن معیارهایی نظیر میزان استفاده از پردازشگر و حافظه، نرخ دور انداختن[[8]](#endnote-9) بسته‌ها، و میزان تاخیر در شناسایی حملات بررسی می‌شود.**

**کلیدواژه:** حملات منع خدمت توزیع‌شده، شبکه‌های پهن‌باند، تطبیق‌پذیری با تنوع ترافیکی، سامانه‌های تشخیص نفوذ

# مقدمه

امروزه با افزایش حجم تبادلات داده‌ای در بستر اینترنت، برقراریِ ارتباطی امن و پایدار در سطح شبکه به یکی از چالش‌های اساسی پیش روی هر سازمانی تبدیل شده است. با توجه به رشد روزافزون کاربران شبکه‌های کامپیوتری، حجم درخواست‌های آن‌ها بزرگ‌تر و پیچیده‌تر می‌شود. از طرف دیگر اینترنت به جز جدایی‌ناپذیری در زندگی و تعاملات کاربران تبدیل شده و بحث دسترس‌پذیری آسان به خدمات بستر اینترنت بیش ‌از پیش مورد توجه قرار می‌گیرد، بدین معنا که ارائه‌دهندگان خدمات ارتباطی[[9]](#endnote-10) موظف هستند خدمات خود را به‌صورت شبانه‌روزی و بدون اختلال و وقفه در اختیار کارخواهان[[10]](#endnote-11) قرار دهند. درصورتی‌که این سازمان‌ها به هر دلیلی در ارائه خدمات خود دچار مشکل شوند و نتوانند به نحو مطلوب خدمات موردنظر را ارائه دهند، با چالش‌های جدی از قبیل از بین رفتن اعتماد مشتریان، خسارات سنگین مالی و از بین‌ رفتن اعتبار سازمان مواجه می‌شوند.

حملات منع خدمت، دسته‌ای از حملات در شبکه هستند که با هدف از بین بردن دسترس‌پذیری شبکه سعی در ممانعت از ارائه و انجام یک خدمت[[11]](#endnote-12) در شبکه دارند. حملات منع خدمت، پهنای باند یا ظرفیت لینک شبکه‌ را مصرف کرده و یا باعث از کار افتادن و اختلال عملکرد در یک کارپذیر[[12]](#endnote-13) یا هر دستگاه حیاتی دیگر در شبکه خواهند شد. گونه‌های مختلفی از این حملات وجود دارد که هرکدام به طریقی سعی می‌کنند دسترس‌پذیری شبکه را هدف قرار داده و یا با مصرف منابع كارپذير، مانع از ارائه خدمت به‌صورت کامل و باکیفیت به کارخواهان و كاربران قانونی شوند. حملات منع خدمت توزیع‌شده یک‌گونه مخرب‌تر از حملات منع خدمت هستند که در آن‌ها حمله‌کننده[[13]](#endnote-14) از طریق سیستم‌هایی که تحت کنترل خود مي‌آورد، حمله را انجام می‌دهد. بدین ترتیب علاوه بر حجم ترافیک سنگین‌ حملات و دشواری‌های تمییز قائل‌‌ شدن بین ترافیک بالا در عین حال قانونی شبکه[[14]](#endnote-15) و ترافیک حمله‌کننده، پیدا کردن فرد مهاجم اصلی نیز به‌مراتب دشوارتر می‌شود.

ازسویی دیگر امروزه با شبکه‌هاي پهن‌باندي مواجه هستیم که منجر به بالا رفتن نرخ‌گذر اطلاعات به ميزان بيش از 100 گيگابيت در ثانيه در بسياري از تجهيزات شبكه شده ‌است. برای شناسایی مهاجمین در چنین شرایطی نیاز به راهکاری است که با سرعت بالایی بتواند تمامی بسته‌ها را بررسی کند. همچنین به دلیل ظهور پروتکل‌ها و برنامه‌های کاربردی مختلف با حجم زیادی از داده‌ها و تنوع زیادی از پروتکل‌ها مواجه هستیم. لذا چالش بعدی تطبیق معیار تشخیص حملات با توجه به کاربرد ترافیک می‌باشد. با توجه به دلایل مطرح شده، همچنان حملات منع خدمت (توزیع‌شده) یکی از تهدید‌های بزرگ در شبکه‌های پهن‌باند محسوب می‌شوند.

این گزارش در پنج بخش تدوین‌ شده است. در بخش ۲ مفاهیم پایه مورد نیاز در این پژوهش‌ معرفی می‌شوند. ابتدا شبکه‌های پهن‌باند و ویژگی‌های آن‌ها بيان می‌شود. سپس انواع حملات منع خدمت، از نقطه ‌نظرهای مختلف مورد بررسی قرار می‌گیرند و در انتهای این بخش به توضیح مفاهیم داده‌ جريان[[15]](#endnote-16) و انگاره‌‌ها[[16]](#endnote-17)، راهکارهای افزایش سرعت پردازش بسته‌ها و معرفی راه‌گزین[[17]](#endnote-18)‌ های برنامه‌پذیر اختصاص می‌یابد. بخش ۳ به بررسی کار‌های پیشین انجام‌شده برای تشخیص حملات منع خدمت توزیع‌شده پرداخته می‌شود. در بخش ۴ روش پیشنهادی به‌ منظور بهبود تشخیص حمله در شبکه‌های پهن‌باند، بیان می‌شود و سرانجام در بخش ۵ نتیجه‌گیری، مراحل انجام پروژه و زمان‌بندی آن بیان خواهد شد.

# مفاهیم پایه

در این بخش به شرح مختصری از مفاهیم پایه مرتبط با این پژوهش‌ پرداخته خواهد شد. ابتدا شبکه‌های پهن‌باند را معرفی می‌کنیم. سپس به معرفی حملات منع خدمت و حملات منع خدمت توزیع‌شده می‌پردازیم و در پایان این بخش مفهوم داده‌ جريان و انگاره‌‌ها را شرح داده می‌شود.

### ۱.۲ شبکه‌های پهن‌باند

امروزه نرخ تبادل اطلاعات در شبکه‌های کامپیوتری بالا رفته و مفهومی به عنوان شبکه‌های پهن‌باند مطرح می‌باشد. شبکه‌های پهن‌باند دارای سه ویژگی زير می‌باشند[1]:

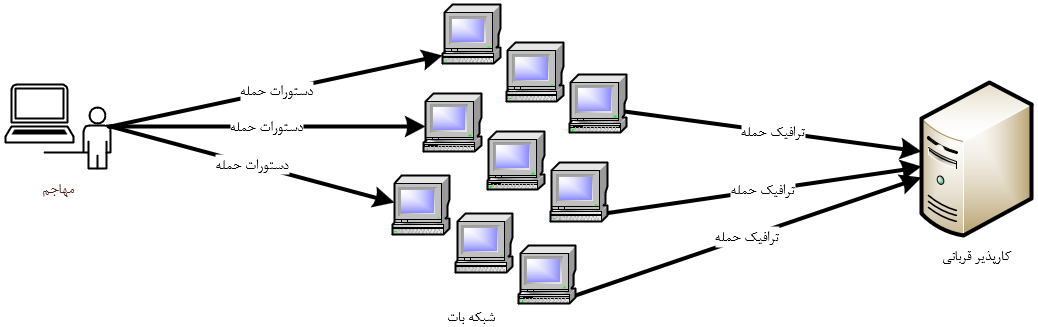
* سرعت بالا: داده‌ها و بسته‌ها با سرعت و نرخ بالایی تولید می‌شوند. برای مثال در شبکه‌های نسل پنجم اینترنت همراه[[18]](#endnote-19)، هر کاربر از قابلیت تبادل اطلاعات با سرعت ۱۵ گیگابیت بر ثانیه برخوردار مي‌باشد.
* حجم بالا:‌ اطلاعات عبوری از شبکه و داده‌هاي در حال تبادل باعث تولید حجم زیادی از فراداده[[19]](#endnote-20) می‌شوند. به عبارتی دیگر بسته‌هایی با محتوا[[20]](#endnote-21) و حجم زیادی از سرایند‌ها[[21]](#endnote-22) را خواهیم داشت. به دلیل ظهور کاربردهای مختلف و به دنبال آن پروتکل‌های مختلف و لزوم استفاده از الگوریتم‌های رمزنگاری، حجم زیادی از سرایند‌ها برای برقراری ارتباط الزامی می‌باشد كه نگهداشت فراداده‌هاي توليد شده آن‌ها هزينه زيادي را شامل مي‌شود. همچنین اطلاعاتی که کاربران در بستر اینترنت تبادل می‌کنند، می‌تواند طيف وسيعي از داده‌ها شامل فایل‌هایی حجیم و یا جرياني بي‌وقفه از بسته‌ها در هنگام مشاهده یک ویدئوي برخط يا در هنگام برگذاري يك كلاس مجازي باشد. در سال ۲۰۰۳، حجم کل داده‌های تولید شده در اینترنت حدود پنج اگزابایت بود که این میزان در سال ۲۰۰۸ سه برابر شد و به ۱۴.۷ اگزابایت رسید. درسال 2019، ترافيك توليدي كاربران به ميزان پنج اگزابایت داده در هر دو روز می‌رسید[2].
* تنوع بالا: علاوه بر ظهور پروتکل‌های مختلف که هر کدام برای کاربردی خاص می‌باشند، نحوه انتقال و دریافت بسته‌ها بین كارخواه-كارپذير و استفاده از این پروتکل‌ها وابسته به وضعیت و نوع کاربرد می‌تواند متنوع باشد. براي مثال با اينكه بيشتر برنامه‌هاي مستقر بر بستر اينترنت، داده‌ها و تبادلات خود را در قالب بسته‌هاي اچ.تي.تي.پي/اس لايه كاربرد انتقال مي‌دهند، اما محتويات اين بسته‌ها و نحوه تفسير‌ آنها براي برنامه‌هاي مختلف مي‌تواند متفاوت باشد.

باتوجه به ویژگی‌های ذکر شده برای شبکه‌های پهن‌باند، مدیریت و کنترل ترافیک در این شبکه‌ها به یکی از چالش‌های اصلی در زمینه شبکه‌های کامپیوتری تبدیل شده است.

### ۲.۲ حملات منع خدمت (توزیع‌شده)

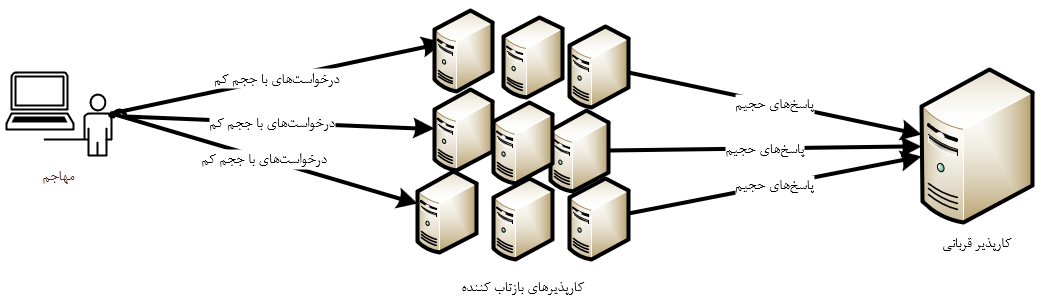
به مجموعه حملاتی که با هدف ممانعت از انجام یک خدمت صورت می‌پذیرند، حملات منع خدمت گفته می‌شود. این حملات با انگیزه‌های مختلفی نظیر ایجاد اختلال یا ممانعت از ارائه یک خدمت، از بین بردن اعتبار و مقبوليّت یک سازمان، آسیب زدن مالی و هدر دادن منابع یک سازمان، دستاوردهای سیاسی و ملی، انگیزه مالی و یا قدرت‌نمایی مهاجمین و مواردی از این دست می‌تواند صورت پذیرد. هدف اصلی در حملات منع خدمت تولید ازدحام و اختلال در مصرف منابع پردازشی سیستم (پردازشگر سیستم) یا منابع شبکه (پهنای باند) می‌باشد.

حملات منع خدمت توزیع‌شده گونه خطرناک‌تر از حملات منع خدمت می‌باشند که در آن فرد مهاجم ابتدا با پايش آسیب پذیری‌های دستگاه‌های مختلف موجود در شبکه اینترنت، شروع به نفوذ به ماشین‌های عامل[[22]](#endnote-23) متعددی می‌کند و سعی می‌کند این دستگاه‌ها را تحت کنترل خود کند. به این سیستم‌هایی که توسط فرد مهاجم از راه دور كنترل می‌شوند، ربات گفته می‌شود و این مجموعه ربات‌ها که به آنها شبکه‌ بات[[23]](#endnote-24) گفته می‌شود، دستورات را از شخص مهاجم دریافت ‌می‌کنند (شکل 1). مهاجم می‌تواند در مدت زمان کوتاهی حجم زیادی از ترافیک را به سمت كارپذير و منابع آن هدایت کند که خدمت‌دهی آن یا رویکرد شبکه را برای پاسخگویی به کاربران قانونی با اختلال مواجه می‌کند. در صورت بروز حملات منع خدمت توزیع‌شده، رهگیری مبدأ حمله یعنی نقطه‌ای که حمله از آنجا شروع شده ‌است، دشوارتر و همچنین ترافیک ایجاد شده در اثر حمله بزرگتر و مخرب‌تر می‌باشد.



شکل 1: حملات منع خدمت توزیع‌شده با استفاده از شبکه بات[4]

مشکل دیگر دفاع در برابر حملات منع خدمت توزیع‌شده، بروز حملات تقویت بازتاب[[24]](#endnote-25) است (شکل 2). در سال ۲۰۱۸، گیتهاب[[25]](#endnote-26) با استفاده از آسیب‌پذیری پروتکل ممکچ[[26]](#endnote-27)، با انعکاس چند برابر بيش‌ از ۵۰۰۰ بار و اوج ترافیک[[27]](#endnote-28) ۱.۳۵ ترابیت بر ثانیه، قربانی‌ یک حمله منع خدمت توزیع‌شده از نوع تقویت بازتابی قرار گرفت.



شکل 2: حملات منع خدمت توزیع‌شده از نوع تقویت بازتابی[4]

در فوریه ۲۰۲۰، ارایه دهندگان خدمات وب آمازون[[28]](#endnote-29) حمله‌ای با حجم ترافيك پیک ۲.۳ ترابایت بر ثانیه را تجربه کردند. در ژوئیه ۲۰۲۱، شرکت ارائه‌‌دهنده خدمات تحویل محتوا[[29]](#endnote-30)ی کلودفلر[[30]](#endnote-31) در گزارشی به محافظت از یکی از مشتریان خود در برابر حمله منع خدمت توزیع‌شده نشأت گرفته‌ شده از یک شبکه بات در ابعاد جهانی توسط بدافزار میرای با ترافیک پیک ۱۷.۲ میلیون درخواست در ثانیه، اشاره کرد. یاندکس[[31]](#endnote-32)، ارائه‌دهنده خدمات پیشگیری از حملات منع خدمت توزیع‌شده روسیه گفت که در تاریخ ۵ سپتامبر ۲۰۲۱ یک حمله منع خدمت توزیع‌شده پروتکل اچ.تی.تی.پی را که از تجهیزات شبکه میکروتک[[32]](#endnote-33) بروزنشده[[33]](#endnote-34) سرچشمه می‌گرفت، مسدود کرده است. طبق پیش‌بینی شرکت سیسکو[[34]](#endnote-35)، تعداد حملات منع خدمت توزیع‌شده به حدود ۱۶ میلیون در سال ۲۰۲۳ میلادی خواهد رسید (شکل 3). در بزرگترین حمله منع خدمت توزیع‌شده رخ داده تا به حال، ترافیک حمله به ۱.۴۴ ترابیت در ثانیه رسیده است. از طرفی در سالیان اخیر، این حملات با استفاده از پروتکل‌های جدیدتری ظاهر خود را تغییر می‌دهند. به عنوان مثال، در پایان ژوئیه ۲۰۲۰، پلیس فدرال آمریکا[[35]](#endnote-36) هشداری صادر کرد مبنی بر اینکه پروتکل برنامه‌های محدود شده[[36]](#endnote-37) و سایر پروتکل‌ها ممکن است برای انجام حملات منع خدمت توزیع‌شده مورد سوء استفاده قرار گیرند. حملات منع خدمت توزیع‌شده بر اساس بردارهای حمله[[37]](#endnote-38) جدید ممکن است تغییرات زیادی در ویژگی‌های آماری مانند سرعت بسته‌ها و فاصله بسته‌های مورد استفاده در مقایسه‌ با روش‌های سنتی داشته باشند، که اين امر باعث می‌شود روش‌های سنتی مقابله در برابر حملات مختلف کارایی لازم را نداشته باشند[3].

شکل 3: گزارش و پیش‌بینی سیسکو از مجموع حملات منع خدمت توزیع‌شده[5]

### ۳.۲ داده جريان

همانطور که در ویژگی‌های شبکه‌های پهن‌باند ذکر شد، نرخ بالای تولید اطلاعات یکی از شاخصه‌های این شبکه‌ها می‌باشد. برای پردازش بسته‌ها در این حالت،‌ دو رویکرد متفاوت وجود دارد:

* پردازش دسته‌ای[[38]](#endnote-39): در این رویکرد تمامی بسته‌ها در یک پنجره زمانی را ضبط کرده و سپس در زمان‌های بعدی پردازش مي‌شوند. از مشکلات پردازش دسته‌اي مي‌توان به تأخیر در ارسال و پردازش و نيز هزینه بسیار زیاد (برای ذخیره‌سازی) به دلیل ذخیره‌سازي اطلاعات در ابتدای‌ کار و سپس ارسال آن به مراکز دیگر، اشاره كرد.
* پردازش جریانی[[39]](#endnote-40): اکثر راهکارهای ارائه شده که در قسمت بعد بررسی مي‌شوند، مبتنی بر این رویکرد می‌باشند. این الگوریتم‌ها دو مشخصه زیر را درنظر می‌گیرند: اول این که اطلاعات به صورت جرياني از داده‌ها (بی وقفه و با سرعت بالا) در حال ارسال می‌باشند و دوم اینکه از نظر زمانی و حافظه محدودیت وجود دارد[6]. این خصیصه‌ها همان چالش‌هایی هستند که براي پردازش ترافيك در شبکه‌های پهن‌باند مطرح مي‌شوند. برای تشخیص حملات در این شبکه‌ها بايد تمامی بسته‌ها را ضبط و پردازش کرده و این کار باید با همان سرعت ورود اطلاعات[[40]](#endnote-41)و با کمترین میزان استفاده از حافظه انجام شود. الگوریتم‌های پردازش‌ جریانی در بحث پردازش اطلاعات مختلف بسیار کاربردی هستند. الگوریتم‌های مبتنی بر پردازش‌ جرياني، ابتدا مسئله را به یکی از چندین روش موجود مدل می‌کنند. یکی از این مدل‌های بسیار متداول و كاربردي ترنستيل[[41]](#endnote-42) می‌باشد. در این مدل یک داده جريان ورودي به نام *I* درنظر گرفته مي‌شود که شامل مجموعه‌ای از تاپل‌های دوتایی می‌باشد:

*I= α1, α2, α3, α4,…*

*Ɐꓰαi={( a1, v1)| a1Ɛ{0,1,…,u-1}, v1ƐR}*

*[u]= key space*

تاپل‌ها، دوتایی‌هایی هستند که شامل مقدار کلید و به‌روز‌رساني می‌باشند. آرایه‌ای به نام *A* وجود دارد که تعداد خانه‌های آن برابر *[u]* و دارای مقادیر متناظر به‌روز‌رساني براي هر کلید می‌باشد. هرگاه یک تاپل جدید*(ax, vx)* دریافت‌ شود مقدار به‌روزرساني آن با مقدار *A[ax]* جمع می‌شود:

*A[ax] += vx*

این پارامتر‌ها وابسته به مسأله داده‌ جرياني که مطرح مي‌شود، می‌توانند متفاوت باشند. دربحث پردازش بسته‌‌های دریافتی شبکه، جريان همان جریان ورودی و تاپل‌ها همان بسته‌ها می‌باشند که براي مثال کلیدشان ۵ خصیصه‌ی آدرس آي.پي[[42]](#endnote-43) مبدأ، آدرس آي.پي مقصد، شماره درگاه مبدأ، شماره درگاه[[43]](#endnote-44) مقصد و پروتكل و به‌روز‌رسانی نیز می‌تواند اندازه بسته باشد. درنتیجه برای شناسایی حملات منع خدمت، بايد آدرس‌هایی که بسته‌هایی با حجم نامتعارف ارسال می‌کنند شناسایی كرد[7].

در مسائل داده جريان، چندین نوع پاسخ برای مسائل اندازه‌گیری مختلف مطرح مي‌باشد و پس از مدل‌سازي مسئله، الگوريتم‌هايي استفاده مي‌شود که بر مبنای مدل سعی در یافتن این پاسخ دارند. برای تحليل بهتر این نوع مسائل، ابتدا مفاهيم اوليه بايد توضيح داده شود.

جريان ورودی را توالي از بسته‌هايي به شكل تاپل‌ شامل شناسه جریان متناظر و اندازه آن بسته در نظر گرفته مي‌شود

*Ƒ=* تعداد کل جریان‌های متمایز شناسه جریان*=(srcIP, srcport, dstIP, dstport, protocol)*

داده جريان ورودی *= (f1,c1),…, (ft,ct),…*

پاسخ‌هاي مسائل يكي از انواع زیر می‌باشند[8]:

* سایز هر جریان[[44]](#endnote-45): خواسته اين مسائل، یافتن سایز جریان یا تعداد بسته‌های دریافت شده متعلق به جریان ƒ مي‌باشد که با *nƒ* نشان داده مي‌شود. سایز تمامی بسته های دریافتی نیز می‌باشد.
* لحظه جریان[[45]](#endnote-46): در لحظه دلخواهي، وضعیت جريان با استفاده از تابع *g* در لحظه [[46]](#endnote-47)*g* مطلوب است، که به صورت زیر می‌تواند تعریف شود:

, ƒ [1, Ƒ]

این تابع وابسته به این‌که به چه صورت تعریف شده باشد، می‌تواند وضعیت کلی از ترافیک شبکه را به صورت عددی بیان کند.

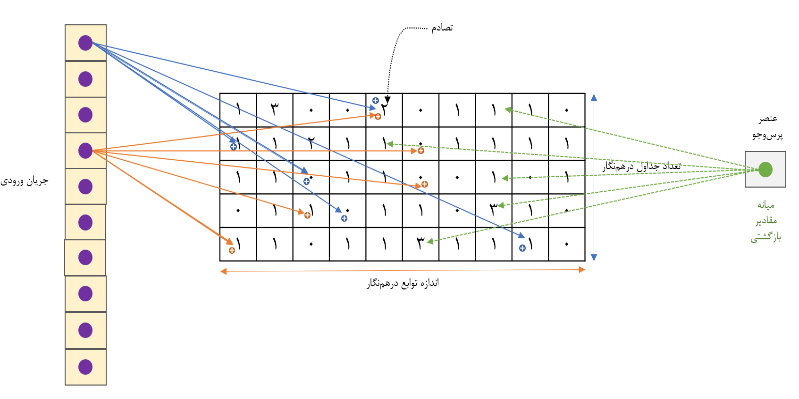
* شاخص[[47]](#endnote-48): شاخص‌ها جریان‌‌هایی هستند که اندازه آنها بر لحظه جریان Lg بیشترین تاثیر را می‌گذارد. به عبارتی دیگر:

که α مقدار آستانه از پیش تعریف شده بین صفر و یک می‌باشد.

### ۴.۲ انگاره

برای حل مسائل داده‌ جریان، راهکارهای متفاوتی را می‌توان استفاده كرد. یکی از راهکارها بدين صورت می‌باشد که به دليل اينكه با حجم زیادی از اطلاعات روبرو هستیم، تنها بخشی از داده‌های ورودی به عنوان نمونه انتخاب شوند و عمليات پردازش تنها روي آن‌ها صورت گيرد. اين روش نمونه برداری[[48]](#endnote-49) ناميده می‌شود[9]. نمونه‌‌برداری دقت پایینی خواهد داشت. به منظور بالابردن دقت، پردازش تمامي بسته‌ها الزامي مي‌باشد. اما بررسی همه بسته‌ها نیز نیازمند حجم زیادی از منابع پردازشی و زمان می‌باشد. برای حل این مشکل، الگوریتم‌هایی به نام انگاره ارائه شده‌اند که از یک داده ساختار فشرده برای ذخیره سازی اطلاعات داده‌های ورودی استفاده می‌کنند. انواع مختلفی از این الگوریتم‌ها در پژوهش‌های مختلف ارائه شده است که هر کدام سعی در حل یکی از انواع مسائل داده جريان دارند. در ذیل چند مورد از پر‌استفاده‌ترین آن‌ها معرفي خواهند شد:

* انگاره شمارشي[[49]](#endnote-50): از یک جدول KxH تشکیل شده است که شامل K تابع درهم‌نگار[[50]](#endnote-51) مي‌‌باشد. H نیز اندازه توابع درهم‌نگار در یک سطر می‌باشد. این ساختار را الگوریتم‌های انگاره دیگر نیز استفاده می‌کنند. چون از توابع درهم‌نگار استفاده می‌شود لذا امکان تصادم[[51]](#endnote-52) وجود خواهد داشت‌. مقادير بازگشتی تخميني خواهند بود و درنتيجه به آنها داده‌ساختارهای آماری احتمالاتی مي‌گویند. اما بايستي نرخ خطای قابل قبول و کرانداری ارائه دهند. از این الگوریتم براي يافتن پاسخ مسائل شاخص استفاده می‌شود[10].

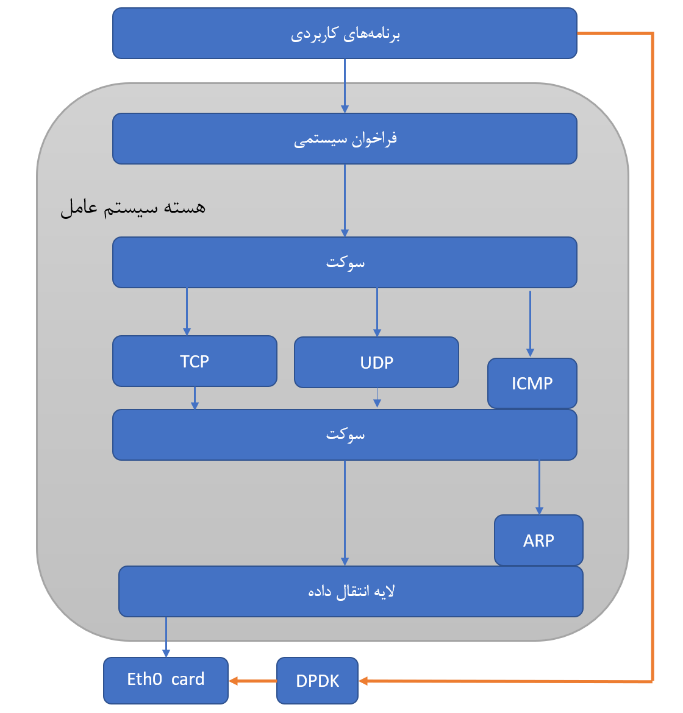


شکل 4: انگاره شمارشی[11]

* انگاره شمارشي كمينه[[52]](#endnote-53): همانند انگاره شمارشي مي‌باشد اما سعي دارد مرتبه فضایی را کاهش دهد[12].
* انگاره عمومي[[53]](#endnote-54): یک دسته جدیدي از انگاره‌ها با هدف ارائه داده‌ساختاری قابل استفاده برای حل تمامی انواع مسائل داده جريان مي‌باشند. يونيومان[[54]](#endnote-55) یکی از این الگوریتم‌ها می‌باشد[13].

### ۵.۲ پردازش سریع‌ بسته‌‌ها

هنگامی که یک بسته از طریق واسط‌هاي شبكه یک سیستم دریافت می‌شود تا پردازش آن، مراحل مختلفی را طی خواهدکرد. بنا به کاربرد، بسته‌ها از دستگاه‌های مختلفی عبور داده می‌شوند. برمبنای پشته پروتکل تی‌.‌سی.پی‌/آی.پی[[55]](#endnote-56) که هسته[[56]](#endnote-57) تمامی توزیع‌های مختلف سیستم عامل لینوکس از آن پشتیبانی می‌کنند، بسته‌ها از دریافت تا پردازش بخش‌های مختلف آن‌ها مراحل مختلفی را پشت ‌سر خواهند گذاشت و درنهایت در صورت نیاز بازارسال خواهند شد. رویدادهای مهم در هنگام دریافت یک بسته توسط ماشین بدين شرح مي‌باشد:

* بسته توسط کارت شبکه[[57]](#endnote-58) ماشین دریافت می‌شود (وقفه کارت شبکه) .

شکل : مراحل ضبط و پردازش بسته[13]

* کارت ‌شبکه از طریق دی.ام.ای[[58]](#endnote-59)، بسته را در فضای حافظه در یک بافر قرار می‌دهد.
* کارت ‌شبکه سیگنالی به پردازنده ارسال می‌کند و آن را برای پردازش بسته بیدار می‌کند (وقفه نرم افزاری).
* پردازنده اطلاعات مورد نیازش را خوانده و در صورت نیاز در فضای بافر تعیین شده می‌نویسد.
* در صورت نیاز، بسته برای پردازش‌های بیشتر به پشته پروتکلی هسته برای انجام پردازش‌‌های مختلف (مثل بررسی آدرس‌ آی.‌پی برای تطبیق با آدرس‌های متناظر لیست کنترل دسترسی) فرستاده می‌شود.
* در نهایت اگر برنامه کاربردی در سطح کاربر باشد، محتویات بسته از فضای هسته به فضای کاربر انتقال داده خواهد شد. در غیر اینصورت، بسته در همان فضای هسته خواهد ماند.

تمامی این مراحل بایستی در سطح هسته انجام شده ولی پردازش بسته توسط کاربر در لایه کاربرد صورت می‌گیرد. این مراحل به دلیل وقفه‌هایی که انجام می‌شود، سربار زیادی خواهند داشت و در شبکه‌های پهن‌باند که با حجم زیادی از بسته‌ها مواجه هستیم، باعث اتلاف وقت زیادی خواهند شد.

دی‌.پی.دی‌.کی[[59]](#endnote-60) و ایکس.دی.پی[[60]](#endnote-61) از ابزار‌های موجود برای تسریع عملیات پردازش بسته می‌باشند. دی.پی.دی.کی ابزار نرم افزاری می‌باشد که در سال ۲۰۰۹ توسط اینتل[[61]](#endnote-62) توسعه داده شد. ولی بعدها به صورت یک پروژه متن باز[[62]](#endnote-63) درآمد. به طور خلاصه یک ابزار دورزدن هسته در هنگام دریافت بسته در شبکه‌ می‌باشد که وقفه‌های مختلف مربوط به هسته را حذف می‌کند و لذا تمام عملیات پردازش بسته را مي‌توان در سطح کاربر انجام داد و درنهايت عملیات دریافت و پردازش بسته را تا‌ حد خوبی می‌تواند تسریع ‌بخشد. هدف این فناوري استفاده از قابلیت پردازش چندهسته‌ای پردازنده‌های معمولی ایکس۸۶[[63]](#endnote-64) برای بهبود سرعت پردازشی کارپذیرها می‌باشد. بدین صورت ما نرخ پردازشی برابر هنگام استفاده از پردازنده‌های مخصوص كارپذيرها و یا مدار‌های مجتمع با کاربرد خاص[[64]](#endnote-65) و مدار مجتمع دیجیتال برنامه‌پذیر[[65]](#endnote-66) ، با صرف هزینه‌ای بسیار کمتر، خواهیم داشت. از چندین پردازنده برای محاسبات مربوط به سطح داده و از بقیه هسته‌ها برای امور کنترلی و خدمات دیگر استفاده می‌کند. به صورت جزئي‌تر، چندین صف بر روی هر واسط شبكه تعریف می‌کند و هسته‌ها با حالت سرکشی[[66]](#endnote-67) به این صف‌ها الصاق می‌شوند. از این ابزار در کاربردهای مختلفی در مواقعی که حجم زیادی از ورودی/خروجی مطرح می‌باشد از حیطه شبکه و امنيت آن، پردازش و راهگزيني در ابرها، بهبود کارایی حافظه‌ها، توابع مجازی شبکه[[67]](#endnote-68)، مخابرات و تلکام استفاده می‌شود. البته به غیر از مورد اشاره شده که ویژگی اصلی این ابزار می‌باشد، امکانات مختلف دیگری مانند رمزگذاری و فشرده‌سازی به کمک رابط‌های برنامه‌نویسی‌اش نیز ارائه می‌دهد[14].

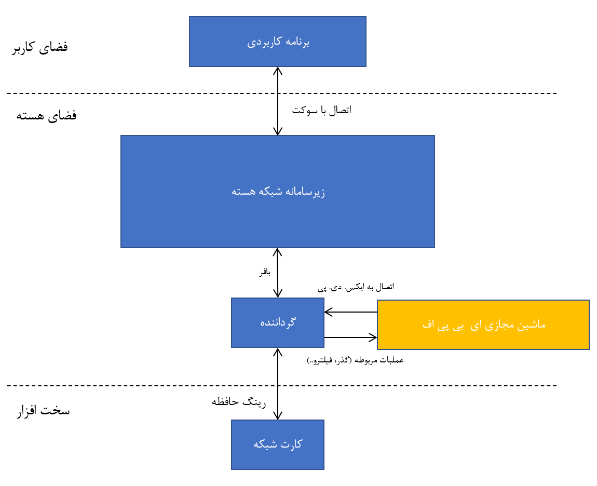
ایکس.دی.پی یکی از مؤلفه‌های جدید هسته می‌باشد که پردازش بسته را به صورت خوبي بهبود می‌بخشد. روش‌هایی مثل دی.‌پی.دی. کی هسته را دور مي‌زنند و تمام عملیات پردازش بسته در فضای کاربر صورت می‌گیرد. همچنین کارت ‌شبکه را بايد توسط یک گرداننده[[68]](#endnote-69) سطح کاربر کنترل ‌نمود. پردازش شبکه در سطح کاربر با وجود مزاياي زيادش، معایب زیر را نيز به همراه خواهد داشت:

* به دليل اينکه سیستم عامل یک لایه انتزاعی برای ارتباط با منابع سخت‌افزاری می‌باشد لذا برنامه‌‌های سطح کاربر برای تعامل با آنها بایستی گرداننده‌های مربوطه را خودشان توسعه دهند.
* برنامه‌های سطح کاربر می‌بایست درصورت نیاز عملکرد‌هایی که توسط هسته ایجاد می‌شد را پیاده‌سازی کنند.
* برنامه‌ها به صورت ایزوله اجرا می‌شوند که نحوه تعامل آنها با دیگر بخش‌های سیستم‌عامل را دشوار می‌کند.

به طور خلاصه ایکس. دی. پی، برنامه‌‌های شبکه سطح کاربر (پالايش، نگاشت، مسیریابی و...) را به جای انتقال به سطح کاربر، به فضاي هسته می‌برد. ایکس.دی.پی امكان اجراي برنامه به محض ورود بسته به کارت ‌شبکه و پیش از حرکت به سمت زیرسیستم شبکه‌ی هسته را فراهم مي‌كند که منجر به افزایش قابل توجه سرعت پردازش بسته می‌شود. اجرای برنامه در سطح هسته با استفاده از بی. پی. اف[[69]](#endnote-70) میسر می‌شود[15].

بی.پی.اف یک ماشین مجازی است که تنها مخصوص پردازش پالايش ترافيك می‌باشد. یکی از ابزارهایی که از بی.پی.اف استفاده می‌کند، تی.‌سی.‌پی ‌دامپ[[70]](#endnote-71) می‌باشد. عبارت پالايش مربوطه توسط یک کامپایلر به بایت‌کد[[71]](#endnote-72) بی.پی.اف تبدیل خواهد شد. از آنجایی که بی.پی.اف یک ماشین مجازی می‌باشد، محیطی به منظور اجرای برنامه‌ها در آن که علاوه بر بایت‌کد شامل یک مدل حافظه مبتنی بر بسته (دستورالعمل‌های بارگذاری به طور ضمنی بر روی بسته موردنظر انجام می‌شود) ، ثبات‌ها[[72]](#endnote-73) (A و X یعنی انباشتگر[[73]](#endnote-74) و ثبات اندیس[[74]](#endnote-75)) ، یک حافظه موقت و یک شمارنده برنامه[[75]](#endnote-76) ضمنی نيز می‌باشد را تعریف می‌کند. هسته لینوکس از نسخه ۲.۵ به بعد از بی.پی.اف پشتیبانی می‌کند. در سال ۲۰۱۱، مفسر بی.پی.اف به یک کامپایلر درجا[[76]](#endnote-77) تغییر داده شد. این کار باعث شد که هسته به جای تفسیر برنامه‌های بی.پی.اف، قادر باشد که آن‌ها را به معماری هدف های مختلف مانند ایکس‌۸۶، میپس[[77]](#endnote-78)، و آرم[[78]](#endnote-79) تبدیل کند. این‌ امر به معرفی بی.پی.اف توزیع یافته[[79]](#endnote-80) در سال ۲۰۱۴ و کنار گذاشته ‌شدن بی.پی.اف سنتی منجر شد. ویژگی‌های جدید شامل موارد زیر می‌باشد:

* از ویژگی‌های معماری ۶۴-بیتی مثل ثبات‌ها و تعداد آن‌ها و کد‌های عملیاتی[[80]](#endnote-81) بیشتر بهره مي‌برد.
* از زیرسیستم شبکه جدا شده است و امکان استفاده در کاربرد‌های دیگر میسر می‌شود.
* نگاشت‌ها به عنوان راهی برای تبادل داده بین سطح کاربر و هسته مورد استفاده قرار می‌گیرند.
* استفاده از توابع کمکی که در سطح هسته اجرا می‌شوند. امکان فراخواني فراخوان سيستمي[[81]](#endnote-82) در برنامه‌های بی.پی.اف نیز وجود دارد.
* زنجیره سازی تعداد برنامه بیشتر بی.پی.اف نيز امكان‌پذير خواهد بود.

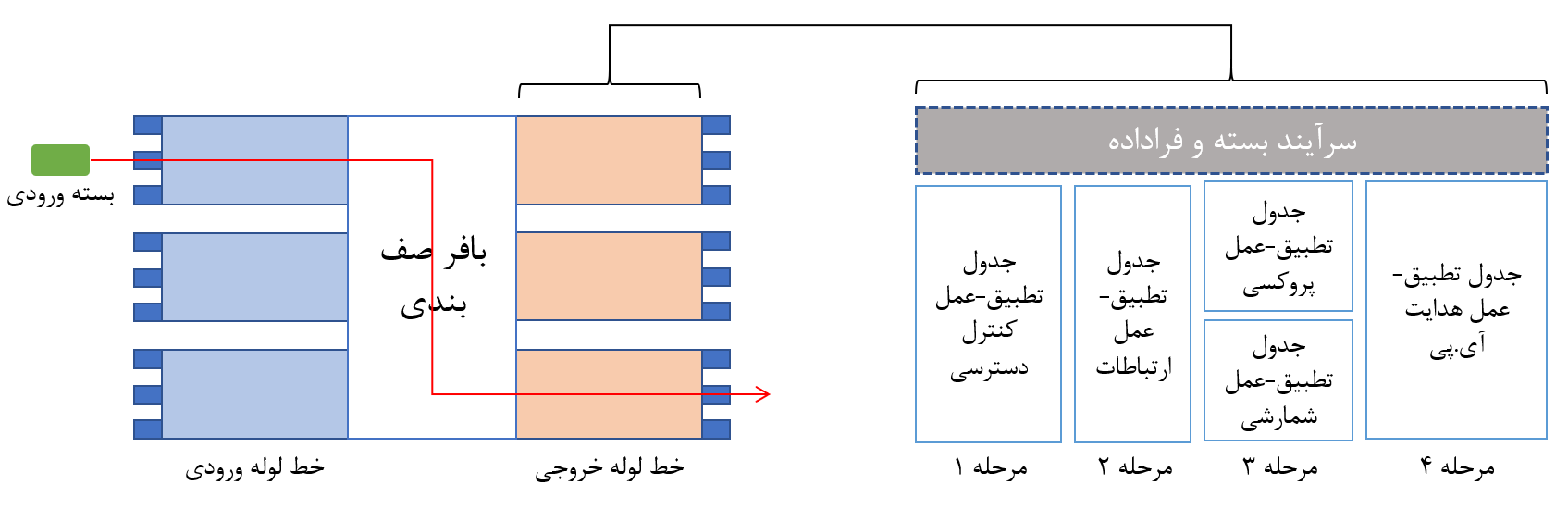
برخی از صف‌های کارت ‌شبکه هنوز به هسته متصل هستند، در حالی که برخی دیگر به یک برنامه فضای کاربر متصل هستند که در مورد حذف‌ شدن یا نشدن یک بسته تصمیم می‌گیرد. با این کار، میزان ترافیکی که به زیرسیستم شبکه هسته می‌رسد به میزان قابل توجهی کاهش می‌یابد. برای این کار بایستی یک نقطه بازرسی[[82]](#endnote-83) در پشته هسته تعریف کرد که هرگاه بسته‌ای در کارت ‌شبکه دریافت شد، آن را به فضای کاربر بفرستد و درآنجا تصمیم می‌گیرد که بسته دور انداخته شود[[83]](#endnote-84) یا اجازه عبور به لایه‌های بالاتر پشته را صادر کند. لذا نیاز به مکانیزمی بود که امکان اجرای کدهای سطح کاربر را در هسته فراهم کند. به همین دلیل از بی.پی.اف توزیع یافته استفاده شد.

شکل : ایکس.دی.پی بر روی هسته سيستم عامل[15]

ایکس.دی.پی بسته‌های دریافتی را به برنامه بی.پی.اف هدایت می‌کند. در آنجا می‌توان بسته‌ها را ویرایش و يا هدايت[[84]](#endnote-85) کرد. از توابع کمکی می‌توان برای انجام محاسبات و پردازش بسته‌ها بدون نیاز به فراخوان سیستمی استفاده کرد. همچنین با استفاده از داده‌ساختار‌های نگاشت امکان ذخیره داده‌ها به صورت دائمي را خواهيم داشت. در نهايت با استفاده از ویژگی‌های از پیش تعبیه‌شده در ایکس.دی.پی می‌توان عمل مورد نظر را بر روی بسته انجام داد[16].

### ۶.۲ راه‌گزين‌‌های برنامه‌پذیر

استفاده از یک كنترل‌كننده به عنوان مرکزی که تمام اطلاعات به آن فرستاده می‌شود و سپس در آنجا بر مبنای الگوریتم پیاده شده بر روی آن، تصمیم می‌گیرد که جلوی ترافیک را بگیرد یا نه، یکی از مشکلات برخی روش‌های تشخیص پیشین بود. این روش با تأخیر زیادی همراه است و همچنین می‌تواند یک نقطه آسیب پذیر واحد برای مهاجمین فراهم کند. اما امروزه با معرفی راهگزين‌هاي برنامه پذیر[[85]](#endnote-86)، راهگزين‌‌های معمولی نیز، با استفاده از برنامه‌هایی که بر روی آنها با استفاده از زبان‌هایی مثل پی.4[[86]](#endnote-87) توسعه داده مي‌شوند، توانایی پردازش داده را تا حد زیادی خواهند داشت. یک راهگزين‌ برنامه‌پذیر مبتنی بر مدار‌های مجتمع با کاربرد خاص، چندین خط لوله شامل واسط‌هاي ورودی و خروجی را شامل می‌شود و بسته‌ها مراحل مختلفی را در طول خط لوله برای پردازش سپری می‌کنند. هر کدام از این مراحل نیز منابع اختصاصی خود یعنی: ثبات‌ها برای ذخیره‌سازی، جداول تطبیق-عمل، و واحد‌های منطق ریاضی به منظور پردازش را شامل می‌شوند. توسط زبان پی.‌۴ امکان شخصی‌سازی جداول تطبیق-عمل به منظور انجام تغییر روی بسته‌ها میسر خواهد بود. در مجموع راهگزين‌‌های برنامه‌پذیر مبتنی بر مدار‌های مجتمع با کاربرد خاص، دو برتری بهینه ‌بودن سرعت پردازشی به نسبت هزینه مصرفی و مصرف برق و انعطاف‌پذیری در برابر حملات جدید را در مقایسه با سخت‌افزار‌های دیگر ارائه مي‌دهند[17].



شکل 7: معماري راه‌گزین برنامه‌پذير[17]

# کارهای ‌پیشین

**به‌طورکلی پژوهش‌های انجام‌شده در حوزه حملات منع خدمت توزیع‌شده را می‌توان در سه دسته پیشگیری از وقوع حمله، تشخیص حمله و کاهش اثر حمله[[87]](#endnote-88) تقسیم‌بندی کرد. از آنجایی‌که تمرکز این گزارش بر پژوهش‌های موجود در حوزه تشخیص حملات منع خدمت توزیع‌شده می‌باشد،** در ادامه به بررسی چند روش اخيراً معرفی‌شده تشخیص حملات منع خدمت توزیع‌شده در شبکه‌های کامپیوتری می‎‌پردازیم. علاوه بر شيوه‌ي دسته‌بندي‌ای كه در ادامه استفاده مي‌كنيم، الگوریتم‌های تشخیص را می‌توان بر اساس اینکه‌ در کدام ناحیه از شبکه سعی به تشخیص مهاجم دارند نیز طبقه بندی کرد، که شامل سه گروه می‌شوند:

* شناسایی در مبدأ: از توانایی تشخیص همه حملات برخوردار نمی‌باشند.
* شناسایی در مقصد(قربانی): نیاز به منابع بیشتری دارند و ممکن است با تأخیر هم همراه باشند.
* شناسایی در مسیر‌های میانی[[88]](#endnote-89).

### ۱.۳ روش‌های مبتنی بر امضا

طبقه‌بندی و فیلترکردن ترافیک شبکه با استفاده از امضا، معمولاً در سیستم‌های تشخیص نفوذ[[89]](#endnote-90) استفاده می‌شود. در این روش، مقادیر فیلد‌های مختلف بسته‌های عبوری با یکسری از امضاها(الگو‌ها)ی بسته‌های مهاجم به منظور تشخیص حمله، مقایسه می‌شوند. اگرچه این روش‌ها الگو‌های حمله‌ی مشاهده‌شده‌ی قبلی را با سرعت بالایی شناسایی می‌کنند، اما قادر به شناسایی حملات جدید نمی‌باشند.

* تقسيم‌كننده و كاهنده ترافيك حملات منع ‌خدمت توزيع‌شده مبتني بر امضا با استفاده از راهگزين‌‌هاي برنامه‌پذير سطح‌ داده: در روش ارائه‌شده در سال 2021، ديموليانس و همكاران سعی می‌کنند امضاهای مهاجم را بدست آورند و تعداد حداقل بهینه خط قوانین به منظور مقابله با آنها را تولید کنند[18]. مشکل روش، عدم کارایی در شناسایی حملات متنوع می‌باشد. همچنين در مورد نحوه ياددهي مجدد مدل‌هاي طبقه‌بندي كننده توضيحي ارائه نمي‌دهد.
* روش تشخيص مبتني بر جريان در شبكه‌هاي با سرعت بالا براي شناسايي حملات با توليد امضاي سازگار با اسنورت[[90]](#endnote-91): در روش ارائه شده در سال 2020 توسط ارلاكر و همكاران، از دسته‌بندي جريان مبتني ‌بر آي‌.پي.فيكس[[91]](#endnote-92) استفاده مي‌شود، كه اطلاعات بيشتري علاوه بر اطلاعات آماري متداول مي‌تواند استخراج كند، همچنين با استفاده از برخي روش‌ها امكان بررسي محتواي داده‌اي نيز ميسر خواهد بود[19]. مشکل این روش عدم کارایی در شناسایی حملات مختلف می‌باشد.

### ۲.۳ روش‌های مبتنی بر ناهنجاري[[92]](#endnote-93)

اين دسته از روش‌ها با ضبط‌كردن و بررسي ترافيك عادي شبكه در يك بازه‌ زماني، رفتار عادي شبكه را شبيه‌سازي مي‌كنند و هرگونه رفتار مغاير با اين مدل را به عنوان ناهنجاري و بروز حمله در نظر مي‌گيرند. یادگیری ماشین به‌عنوان یکی از روش‌های کارآمد مبتنی بر ناهنجاري می‌باشد، که امروزه به صورت گسترده‌ای مورد استقبال پژوهشگران قرارگرفته است. در حوزه تشخیص حملات منع خدمت توزیع‌شده نیز از این روش استفاده می‌شود. گونه‌های مختلفی از الگوریتم‌های یادگیری ماشين نظیر استفاده از ماشین بردار پشتیبان، بیز ساده، نزدیک‌ترین همسایه، شبکه عصبی و شبکه‌های عصبی ژرف، نگاشت خودسازمان‌ده و مواردی از این قبیل به‌منظور انجام طبقه‌بندي جریان ترافيك و تشخيص ناهنجاري در آن مورد استفاده قرار می‌گیرند.

* رویکرد تشخيص ناهنجاري به كمك راهگزين‌‌‌های برنامه‌پذیر برای شناسایی و مقابله با حملات منع خدمت توزیع‌شده: این روش به نام جاکن در سال ۲۰۲۱ توسط لیو و همکاران با استفاده از انگاره‌های عمومی و پیاده‌سازی آنها روی راهگزين‌‌های برنامه‌پذیر به منظور جمع‌‌آوری اطلاعات توسط همین دستگاه‌ها، ارائه شد. یک کنترل‌کننده مرکزی از این اطلاعات برای تشخیص حملات استفاده می‌کند. همچنین الگوریتم‌هایی به منظور رفع مخاطره برروی این راهگزين‌‌ها می‌توان پیاده کرد. به دلیل این که با استفاده از زبان پی‌۴، الگوریتم‌های تشخیص و رفع مخاطره پیاده مي‌شود، لذا این روش مبتنی بر معماری خاصی از راهگزين‌‌ها نمی‌باشد[20]. عدم وارسي محتواي كامل بسته‌ها و عدم استفاده از وارسي‌كننده عميق بسته[[93]](#endnote-94)، مشكل اصلي جاكن مي‌باشد. به همین دلیل به بحث تنوع پروتکلی و اینکه مقادير آستانه‌ برای برنامه‌های کاربردی مختلف می‌تواند متفاوت باشد، اشاره‌ای نکرده است.
* روش بلادرنگ تطبيق‌پذير مبتني بر انگاره مخصوص شبكه‌هاي ارائه دهنده خدمات اينترنتي: آر.تي.‌سد[[94]](#endnote-95) در سال 2021 توسط شي و همكاران پيشنهاد شد. از نامتوازن بودن مقادير يك ويژگی‌ برای یک آدرس مقصد مشخص، قربانی‌ بودن آن را تشخیص می‌دهد[3]. به دلیل استفاده از انگاره‌ها، آر.تي.‌سد از نظر مرتبه فضایی بسیار بهینه می‌باشد. اما مسئله نحوه انتخاب این ویژگی‌ها برای برنامه‌های کاربردی مختلف به صورت پویا را درنظر نمي‌گيرد.
* روش توزيع شده مقابله با حملات منع خدمت توزيع‌شده به كمك شبكه‌هاي عصبي: روشي به نام دفاع هوشمند در سال 2022 توسط ماينني و همكاران ارائه شد كه از شبکه‌های عصبی عمیق در سمت لبه مشتری و شبکه‌های عصبی عمیق با الگوریتم‌های پیشرفته‌تر در سمت فراهم کننده اینترنت برای شناسایی حملات استفاده می‌کند[19]. اما مشکل این روش، عدم ارائه راهكاري بهينه به منظور آموزش مجدد شبكه‌ها در شبكه‌هاي پهن‌باند مي‌باشد.
* روش بلادرنگ مبتني بر ناهنجاري براي تشخيص رخنه در شبكه‌هاي با سرعت بالا: روشي توسط ويگاس و همكاران در سال 2019 ارائه شد كه با درنظرگرفتن بسته‌‌ها به عنوان جریان، ویژگی‌های آن‌ها را استخراج می‌کند و سپس از روی آنها حمله را تشخیص می‌دهد. يك بخش اعتمادسازي دارد كه ميزان قابل اعتمادبودن گروه‌‌بندي ارائه شده توسط طبقه‌بندي‌كننده را بررسي مي‌كند و اگر از مقدار حداقلي پايين بود، به كمك يك شخص مدير آن را برچسب‌گذاري مي‌كند و سپس مدل طبقه‌بندي‌كننده را به صورت افزايشي به روز مي‌كند[20]. به دليل مداخله انسان برای برچسب‌گذاري برخی جریان‌ها، در شبكه‌هاي پهن‌باند با مشكل مواجه مي‌شود و سربار بالا و دقت پايينی خواهد داشت.

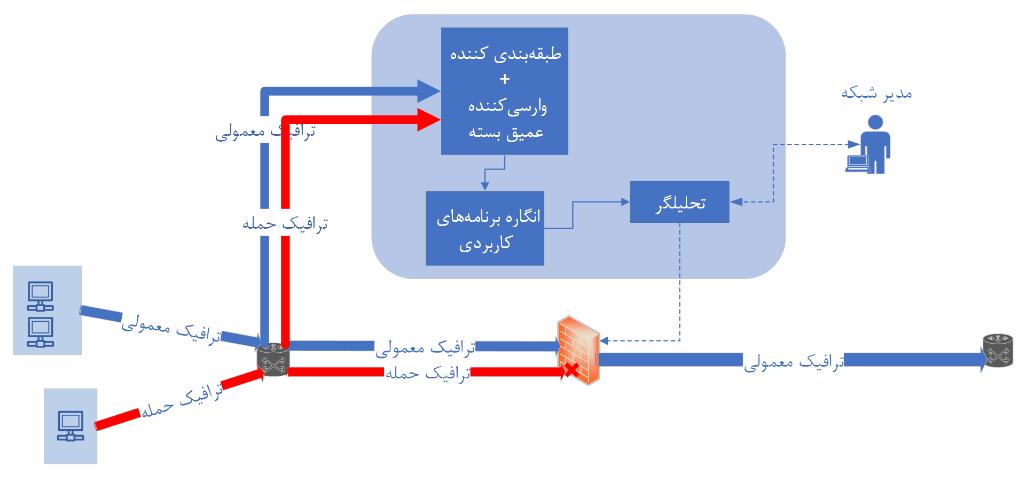
# روش پیشنهادی

در بخش قبل برخی روش‌های مبتنی بر امضا و مدل‌سازي به منظور تشخیص حملات منع خدمت توزیع‌شده معرفی شدند. به عنوان نتیجه می‌توان گفت، روش‌های شناسایی و مقابله با حملات منع خدمت توزیع‌شده، ویژگی‌های ترافیک را از سه منظر بررسی می‌کنند (ترافیک را از سه منظر مشاهده می‌کنند) و سعی در مقابله دارند:

* **بسته**: با استفاده از ويژگی‌ها در سطح بسته (برای مثال یک بسته اچ.‌‌ تي.‌تي‌.پي با اندازه خاص) حملات را تشخیص می‌دهند.
* **جریان**: در صورت مشاهده ویژگی‌های نامتعارف برای یک جریان (برای مثال داشتن اندازه‌ای بیش از ۱۰۰ کیلوبایت) حملات را تشخیص می‌دهند.
* **رفتار** **کاربر**: با استفاده از نامتعارف بودن رفتار ترافیک کاربری سعی می‌کنند حملات را تشخیص دهند. به طور مثال کاربری که در مدت زمان مشخصی، بیش از ۱۰ درخواست به منابع مختلف یک سایت ارسال کند.

در هر یک از منظرها طبق اطلاعاتی که جمع‌‌آوري مي‌كنند، ترافیک نامتعارف را با استفاده از رويكرد‌هاي مبتني بر امضا يا با استفاده از رويكرد‌هاي مدل‌سازي و تشخيص ناهنجاري، مي‌توانند حملات را تشخيص دهند. اما عدم سازگاري با تنّوع پروتکلی برنامه‌هاي كاربردي مختلف، از مشكلات آن‌ها مي‌باشد و در هنگام تشخيص حملات دچار خطا مي‌شوند. در واقع ترافیک برای هر کاربرد می‌تواند الگوی مختلفی داشته باشد و برای همه کاربردها نمی‌توان یک الگو، مرز و شناسه برای حالت متعارف آن تعریف نمود. برای حل این مشکل به شناسایی کاربردهای مختلف می‌پردازند که از روش‌هايی مانند وارسي عمیق بسته یا یادگیری ماشین استفاده می‌شود و سپس با استفاده از نتايج آنها ترافیک را دسته‌بندی کرده و در هر کدام برای تشخیص الگو‌های نامتعارف، تنظیمات متفاوتی (مانند مقادیر آستانه متفاوت برای حجم بسته‌‌ها) به کار می‌برند. اما در شبکه‌های پهن‌باند با مشکلی به نام تنوع ترافیکی بالا مواجه هستیم و از طرفی با توجه به نرخ‌ بالاي توليد ترافیک بایستی در کمترین زمان ممکن، کم ‌‌هزینه‌ترین راهکار را استفاده كنيم. راهکارهای مبتنی بر یادگیری ماشین و استفاده از وارسی‌کننده عمیق بسته، سربار محاسباتی زیاد دارند.

همانطور كه گفته شد، نکته‌ای که در پژوهش های پیشین نادیده گرفته می‌شد، مربوط به مؤلفه سوم شبکه‌های پهن‌باند یا تنوع بالای ترافیک می‌باشد. در روش‌های پیشین مولفه‌های اول و دوم یعنی اینکه داده‌ها با سرعت زیادی در حال تولید هستند و با حجم زیادی از سرآيندها و محتوا روربرو هستيم را تنها درنظر گرفته بودند. لذا روشی كه ارائه می‌دهیم تمامی این سه مورد را با تمرکز بیشتر بر روی ویژگی سوم به عنوان مسئله اصلی، هدف قرار می‌دهد. روش ارائه شده، از ویژگی‌ پردازش جامع برخوردار می‌باشد، یعنی تمامی بسته‌ها را یک و تنها یکبار بررسی می‌کند. بدین صورت روشی بسیار سریع و با دقت بالا و تطبیق‌پذیر با مشخصات ترافیکی شبكه‌هاي پهن‌باند مي‌باشد (علاوه بر معیارهای متداولی مثل سرعت-نرخ‌گذر بالا و تأخیر کم که خواسته همه روش‌های قبلی بوده است) که راه‌حل نویني نيز به شمار مي‌رود.



شکل 8: معماری کلی روش پیشنهادی

روش پيشنهادي ما بدین صورت خواهد بود كه با استفاده از راهكارهاي ارائه شده در پژوهش‌های پیشین و همچنين استفاده از يك زیرسامانه وارسی‌کننده عمیق بسته، جریان‌ها را برچسب‌گذاري مي‌كنيم و جریان‌های شبیه به هم از نظر رفتار را در یک گروه قرار مي‌دهيم. این اطلاعات در داده‌ساختار‌های انگاره که بر روی راهگزين‌‌های برنامه‌پذیر می‌باشند و توسط مدیر شبکه کنترل می‌شوند، برای هر برنامه‌ کاربردی و پروتکل متناظر به صورت جدا ذخیره می‌شود. یک قسمت تحلیلگر وجود دارد که با استفاده از این ویژگی‌‌های آماری و مشاهده رفتار متداول هر پروتکل و یا برنامه‌ كاربردي در بازه‌های زمانی مختلف، این اطلاعات را با مقادیر آستانه‌ای که از قبل به دست آورده و نشان‌دهنده حداکثر بی‌نظمی قابل چشم‌پوشی در شبکه می‌باشد، مقایسه می‌کند و در صورت مشاهده مغایرت آن جریان را به عنوان یک حمله تشخیص داده و سعی می‌کند خط‌مشی امنیتی معادل آن را تولید ‌کند و به عنوان خروجی به یک دیوار آتش ارسال کند. برای بهبود سرعت پردازش‌ بسته‌ها و حذف وقفه‌های زمانبر نیز از ابزارهای مخصوص پردازش سریع بسته‌ها (مانند دی.پی.دی.کی و ایکس.دی.پی) استفاده می‌شود. همچنین کارایی روش ارائه شده در مقایسه با دیگر راهکارها و با ‌درنظرگرفتن معیارهایی نظیر میزان استفاده از پردازشگر و حافظه، نرخ دورانداختن بسته‌ها، و میزان تاخیر در شناسایی حملات در مواجهه با ترافیک‌های حجیم و ترافیک‌های تولیدی از انواع مختلف حملات منع خدمت، بررسی خواهد‌شد.

# نتيجه‌گيري

در این نوشتار به معرفی شبکه‌های پهن‌باند و ویژگی‌های این شبکه‌ها، حملات منع خدمت توزیع‌شده، روش‌ها و الگوریتم‌های پردازش داده جریان و معرفی مفاهیم و واژه‌های به کاررفته در اين‌ زمينه، مانند محاسبات و مسائل داده جريان و موضوعاتی دیگر پرداخته شد. سپس برخی پژوهش‌های انجام‌شده درزمینه‌ی تشخیص حملات منع خدمت توزیع‌شده در شبکه‌های پهن‌باند مورد بررسی قرار گرفت و مشکلات پیاده‌سازی و عملکردی و چالش‌های حل‌نشده آنها بیان شد. در پايان روش پیشنهادی سریع با دقت بالا و بهینه از نظر میزان مصرف منابع و سازگار با تنوع ترافیکی به منظور شناسایی حملات منع خدمت توزیع‌شده در بستر شبکه‌‌های پهن‌باند به صورت مختصر توضيح‌ داده شد. در جدول 1 زمان‌بندی لازم برای بخش‌های مختلف این پژوهش آورده شده‌است.

جدول 1: مراحل انجام و پیشبرد پروژه

|  |  |  |
| --- | --- | --- |
| فعالیت | میزان پیشرفت | تخمین زمان باقی‌مانده |
| ۱. مطالعه و بررسی مفاهیم | ۹۰٪ | ۱ هفته |
| ۲. تحلیل و بررسی کارهای پیشین | ۷۰٪ | 3 هفته |
| ۳. ارائه و امکان‎سنجی روش پیشنهادی | ۱۰٪ | 6 هفته |
| ۴. پیاده‌سازی روش پیشنهادی | ۱۰٪ | 8 هفته |
| ۵. ارزیابی روش پیشنهادی | ۰٪ | 6 هفته |
| ۶. جمع‌بندی و تدوین پایان‌نامه | ۰٪ | 6 هفته |

### كتاب‌نامه

[1] M. Noferesti and R. Jalili, ‘ACoPE: An adaptive semi-supervised learning approach for complex-policy enforcement in high-bandwidth networks’, *Computer Networks*, vol. 166, p. 106943, Jan. 2020.

[2] R. K. Deka, D. K. Bhattacharyya, and J. K. Kalita, ‘Active learning to detect DDoS attack using ranked features’, *Computer Communications*, vol. 145, pp. 203–222, Sep. 2019.

[3] H. Shi, G. Cheng, Y. Hu, F. Wang, and H. Ding, ‘RT-SAD: Real-Time Sketch-Based Adaptive DDoS Detection for ISP Network’, *Security and Communication Networks*, vol. 2021, pp. 1–10, Jul. 2021.

[4] R. Vishwakarma and A. K. Jain, ‘A survey of DDoS attacking techniques and defence mechanisms in the IoT network’, *Telecommun Syst*, vol. 73, no. 1, pp. 3–25, Jan. 2020.

[5] ‘Cisco Annual Internet Report (2018–2023) White Paper’, Mar. 2020. [Online]. Available: https://www.a10networks.com/blog/5-most-famous-ddos-attacks/

[6] B. Zhao, X. Li, B. Tian, Z. Mei, and W. Wu, ‘DHS: Adaptive Memory Layout Organization of Sketch Slots for Fast and Accurate Data Stream Processing’, in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, Virtual Event Singapore, pp. 2285–2293, Aug. 2021.

[7] B. Krishnamurthy, S. Sen, Y. Zhang, and Y. Chen, ‘Sketch-based change detection: methods, evaluation, and applications’, in *Proceedings of the 2003 ACM SIGCOMM conference on Internet measurement - IMC ’03*, Miami Beach, FL, USA, p. 234, 2003.

[8] Q. Xiao, Z. Tang, and S. Chen, ‘Universal Online Sketch for Tracking Heavy Hitters and Estimating Moments of Data Streams’, in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, Toronto, ON, Canada, pp. 974–983, Jul. 2020.

[9] V. Sivaraman, S. Narayana, O. Rottenstreich, S. Muthukrishnan, and J. Rexford, ‘Heavy-Hitter Detection Entirely in the Data Plane’, in *Proceedings of the Symposium on SDN Research*, Santa Clara CA USA, pp. 164–176, Apr. 2017.

[10] M. Charikar, K. Chen, and M. Farach-Colton, ‘Finding Frequent Items in Data Streams’, in *Automata, Languages and Programming*, vol. 2380, P. Widmayer, S. Eidenbenz, F. Triguero, R. Morales, R. Conejo, and M. Hennessy, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 693–703, 2002.

[11] S. Muthukrishnan, ‘Data Streams: Algorithms and Applications’, *FNT in Theoretical Computer Science*, vol. 1, no. 2, pp. 117–236, 2005.

[12] G. Cormode and S. Muthukrishnan, ‘An improved data stream summary: the count-min sketch and its applications’, *Journal of Algorithms*, vol. 55, no. 1, pp. 58–75, Apr. 2005.

[13] Z. Liu, A. Manousis, G. Vorsanger, V. Sekar, and V. Braverman, ‘One Sketch to Rule Them All: Rethinking Network Flow Monitoring with UnivMon’, in *Proceedings of the 2016 ACM SIGCOMM Conference*, Florianopolis Brazil, pp. 101–114, Aug. 2016.

[14] H. Zhu, *Data Plane Development Kit (DPDK): A Software Optimization Guide to the User Space-based Network Applications*, 1st Edition. CRC Press, 2020.

[15] T. Høiland-Jørgensen *et al.*, ‘The eXpress data path: fast programmable packet processing in the operating system kernel’, in *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*, Heraklion Greece, pp. 54–66, Dec. 2018.

[16] M. Fleming, ‘A thorough introduction to eBPF’, *LWN.net Linux Weekly News*, Dec. 02, 2017. https://lwn.net/Articles/740157/

[17] M. Zhang *et al.*, ‘Poseidon: Mitigating Volumetric DDoS Attacks with Programmable Switches’, in *Proceedings 2020 Network and Distributed System Security Symposium*, San Diego, CA, 2020.

[18] M. Dimolianis, A. Pavlidis, and V. Maglaris, ‘Signature-Based Traffic Classification and Mitigation for DDoS Attacks Using Programmable Network Data Planes’, *IEEE Access*, vol. 9, pp. 113061–113076, 2021.

[19] F. Erlacher and F. Dressler, ‘On High-Speed Flow-Based Intrusion Detection Using Snort-Compatible Signatures’, *IEEE Trans. Dependable and Secure Comput.*, vol. 19, no. 1, pp. 495–506, Jan. 2022.

[20] Z. Liu *et al.*, ‘Jaqen: A High-Performance Switch-Native Approach for Detecting and Mitigating Volumetric DDoS Attacks with Programmable Switches’, in *30th USENIX Security Symposium (USENIX Security 21)*, pp. 3829–3846, 2021.

[21] S. Myneni, A. Chowdhary, D. Huang, and A. Alshamrani, ‘SmartDefense: A distributed deep defense against DDoS attacks with edge computing’, *Computer Networks*, vol. 209, p. 108874, May 2022.

[22] E. Viegas, A. Santin, A. Bessani, and N. Neves, ‘BigFlow: Real-time and reliable anomaly-based intrusion detection for high-speed networks’, *Future Generation Computer Systems*, vol. 93, pp. 473–485.

### 

### واژه‌نامه

1. Application [↑](#endnote-ref-2)
2. HTTP/S [↑](#endnote-ref-3)
3. High-Bandwidth networks [↑](#endnote-ref-4)
4. Denial of Service [↑](#endnote-ref-5)
5. Distributed Denial Of Service(DDOS) [↑](#endnote-ref-6)
6. Comprehensive Processing [↑](#endnote-ref-7)
7. Adaptive Learning [↑](#endnote-ref-8)
8. Drop Rate [↑](#endnote-ref-9)
9. Internet Service Provider [↑](#endnote-ref-10)
10. Client [↑](#endnote-ref-11)
11. Service [↑](#endnote-ref-12)
12. Server [↑](#endnote-ref-13)
13. Attacker [↑](#endnote-ref-14)
14. Flash Coward [↑](#endnote-ref-15)
15. Data Streaming [↑](#endnote-ref-16)
16. Sketch [↑](#endnote-ref-17)
17. Switch [↑](#endnote-ref-18)
18. 5G [↑](#endnote-ref-19)
19. Metadata [↑](#endnote-ref-20)
20. Payload [↑](#endnote-ref-21)
21. Header [↑](#endnote-ref-22)
22. Agent Machine [↑](#endnote-ref-23)
23. Botnet [↑](#endnote-ref-24)
24. Amplification DDoS Attack [↑](#endnote-ref-25)
25. GitHub [↑](#endnote-ref-26)
26. Memcached Distributed Caching Memory System [↑](#endnote-ref-27)
27. Traffic peak [↑](#endnote-ref-28)
28. Amazon Web Services [↑](#endnote-ref-29)
29. Content Delivery Network (CDN) [↑](#endnote-ref-30)
30. Cloudflare [↑](#endnote-ref-31)
31. Yandex [↑](#endnote-ref-32)
32. MikroTik [↑](#endnote-ref-33)
33. Unpatched [↑](#endnote-ref-34)
34. Cisco [↑](#endnote-ref-35)
35. FBI [↑](#endnote-ref-36)
36. Constrained Application Protocol(CAP) [↑](#endnote-ref-37)
37. Attack Vector [↑](#endnote-ref-38)
38. Batch Processing [↑](#endnote-ref-39)
39. Stream Processing [↑](#endnote-ref-40)
40. Line Rate Processing [↑](#endnote-ref-41)
41. Turnstile Model [↑](#endnote-ref-42)
42. Internet Protocol (IP) [↑](#endnote-ref-43)
43. Port [↑](#endnote-ref-44)
44. Per flow size [↑](#endnote-ref-45)
45. Flow Moment [↑](#endnote-ref-46)
46. Moment-g [↑](#endnote-ref-47)
47. Heavy Hitter [↑](#endnote-ref-48)
48. Sampling [↑](#endnote-ref-49)
49. Count-Sketch [↑](#endnote-ref-50)
50. Hash Function [↑](#endnote-ref-51)
51. Hash Collision [↑](#endnote-ref-52)
52. Count Min Sketch [↑](#endnote-ref-53)
53. Universal Sketch [↑](#endnote-ref-54)
54. Univmon sketch [↑](#endnote-ref-55)
55. TCP/IP Stack [↑](#endnote-ref-56)
56. Kernel [↑](#endnote-ref-57)
57. Network Interface Card (NIC) [↑](#endnote-ref-58)
58. Direct Memory Access (DMA) [↑](#endnote-ref-59)
59. Data Plane Development Kit(DPDK) [↑](#endnote-ref-60)
60. eXpress Data Path(XDP) [↑](#endnote-ref-61)
61. Intel [↑](#endnote-ref-62)
62. Open Source Project [↑](#endnote-ref-63)
63. X86 Processor Architecture [↑](#endnote-ref-64)
64. Application Specific Integrated Circuit (ASIC) [↑](#endnote-ref-65)
65. Field Programmable Gate Array (FPGA) [↑](#endnote-ref-66)
66. Polling Mode [↑](#endnote-ref-67)
67. Network Function Virtualization(NFV) [↑](#endnote-ref-68)
68. Driver [↑](#endnote-ref-69)
69. Berkely Packet Filter (BPF) [↑](#endnote-ref-70)
70. Tcpdump [↑](#endnote-ref-71)
71. Bytecode [↑](#endnote-ref-72)
72. Register [↑](#endnote-ref-73)
73. Accumulator [↑](#endnote-ref-74)
74. Index Register [↑](#endnote-ref-75)
75. Program Counter [↑](#endnote-ref-76)
76. Just-In-Time Compiler (JIT) [↑](#endnote-ref-77)
77. MIPS Architecture [↑](#endnote-ref-78)
78. ARM Architecture [↑](#endnote-ref-79)
79. Extended BPF (EBPF) [↑](#endnote-ref-80)
80. Opcode [↑](#endnote-ref-81)
81. System Call [↑](#endnote-ref-82)
82. Checkpoint [↑](#endnote-ref-83)
83. Drop [↑](#endnote-ref-84)
84. Forward [↑](#endnote-ref-85)
85. Data-Plane Switch [↑](#endnote-ref-86)
86. P4 [↑](#endnote-ref-87)
87. Attack Mitigation [↑](#endnote-ref-88)
88. Middlebox [↑](#endnote-ref-89)
89. Network Intrusion Detection [↑](#endnote-ref-90)
90. Snort Intrusion Detection System [↑](#endnote-ref-91)
91. IPFIX [↑](#endnote-ref-92)
92. Anomaly [↑](#endnote-ref-93)
93. Deep Packet Inspection(DPI) [↑](#endnote-ref-94)
94. RT-SAD [↑](#endnote-ref-95)