

ارائه‌ی ابزار توزیع شده تولید ترافیک متنوع (DVTG) با توان عملیاتی بالا به منظور شبیه‌سازی رفتار اینترنت

سارا ارشد^۱، مسعود نریمانی زمان آبادی^۲، علیرضا نوروزی^۳

sara.arshad.86@gmail.com، دانشگاه کامپیوتر، دانشگاه تربیت مدرس،

m.narimani@stu.qom.ac.ir، دانشجوی دکترای فناوری اطلاعات، دانشگاه قم،

norwozi@ce.sharif.ir، دانشکده مهندسی کامپیوتر، دانشگاه صنعتی شریف،

چکیده

رشد روزافزون سرویس‌ها و کاربران اینترنت نیاز به مدل‌سازی ترافیک آن را به منظور بررسی معایب و نقاط ضعف سرویس‌ها افزایش می‌دهد. مدل‌سازی رفتار ترافیکی و شرایط اینترنت برای تحلیل و پیش‌بینی کارایی و ارزیابی کیفیت سرویس شبکه‌های کامپیوتری و عناصر آن مانند سوئیچ و مسیریاب و دیواره‌آتش ضروری است. تولید این ترافیک دو نیازمندی اساسی دارد: حجم بالا و تنوع پروتکل‌ها. از طرفی سکوهایی که قادرند حجم بالای ترافیک را بصورت تنها و مستقل تولید کنند هزینه بالایی دارند. این مقاله ابزار DVTG را معرفی می‌کند که قادر است به صورت توزیع شده حجم بالایی از ترافیک شبکه را، که شامل گستره‌ی متنوعی از پروتکل‌های شبکه می‌شود، تولید کند. این ابزار با بهره‌گیری از تکنیک‌های آماری در لایه کاربرد و ابزارهای موجود در لایه‌های دیگر، رفتار واقعی ترافیک شبکه را شبیه‌سازی کرده و قادر است کارایی شبکه و عناصر آن را ارزیابی کند. ارزیابی‌ها نشان می‌دهد با استفاده از ده کامپیوتر با کارت شبکه یک گیگابایت بر ثانیه، عامل‌های تولید ترافیک می‌توانند بیش از نه گیگابایت بر ثانیه ترافیک متنوع تولید کنند که سربراری کمتر از ده درصد دارد.

کلیدواژه

ارزیابی عنصر شبکه، مدل‌سازی ترافیک اینترنت، ابزار تولید ترافیک، ابزار توزیع شده تولید ترافیک متنوع شبکه

مقدمه

برای ارزیابی رفتار و کارایی میزبان‌ها مناسب هستند. اما اغلب برای تست روترها یا سوئیچ‌ها سنگین می‌باشند. معمولاً تمرکز آنها تنها بر روی یک نوع پروتکل کاربردی می‌باشد. (۲) ابزارهای سطح جریان^۱ زمانی مورد استفاده قرار می‌گیرند که ترافیک مورد نیاز فقط در سطح جریان باشد. (۳) ابزارهای سطح بسته^۲ این ابزارها مبتنی بر زمان بین خروج بسته‌ها و اندازه‌ی بسته می‌باشند که معمولاً با استفاده از توابع توزیع تعیین می‌شوند. ابزارهای سطح بسته به عنوان ابزارهای اندازه‌گیری فعال نیز مورد استفاده قرار می‌گیرند. به عنوان مثال برای محاسبه‌ی تأخیر و اعوجاج. (۴) ابزارهای حلقه-بسته^۳ (مبتنی بر رفتار) و چند سطحی^۴ این ابزارها تعاملات بین لایه‌های پشت‌پشتی پروتکلی را در نظر می‌گیرند. اغلب پیچیده هستند به همین دلیل در تحقیقات تجربی چندان مورد استفاده قرار نمی‌گیرند. همچنین ابزارهای حلقه-بسته با استفاده از اطلاعات ردیابی شبکه و با تکیه بر مدل‌هایی مانند مارکوف به تولید ترافیک می‌پردازند. در این ابزارها ممکن

اینترنت شبکه‌ای از سیستم‌های ناهمگن و بار ترافیکی آن ترکیبی از منابع پیچیده است. با توسعه‌ی برنامه‌های کاربردی، فناوری‌های دسترسی به اینترنت، پروتکل‌ها و سرویس‌های جدید به سرعت به این پیچیدگی‌ها افزوده می‌شود. استفاده‌ی وسیع از چنین سیستم‌هایی خطر شکست سرویس‌ها و بنابراین نیاز به ابزارهای تولید و تست ترافیک را افزایش داده‌است. این ابزارها در حوزه‌های مختلف شبکه کاربرد داشته و مورد مطالعه و بررسی قرار گرفته‌اند. از جمله می‌توان به تست سرور برنامه‌های کاربردی، تست‌های امنیتی و تست دستگاه‌ها و تجهیزات شبکه اشاره کرد. از نظر سطح انتزاعی ترافیک، ابزارها به چهار گروه تقسیم‌بندی شده‌اند [۱]:

(۱) ابزارهای سطح کاربرد: تمرکز این ابزارها تولید درخواست‌های لایه‌ی کاربرد است. ترافیک تولیدی توسط آنها ویژگی‌های آماری یکسانی با برنامه‌ی کاربردی مدل شده دارد. این ابزارها

^۴ Packet level generators
^۳ Closed-loop
^۲ Multi-level generators

^۱ Application level generators
^۲ Host
^۳ Flow level generators

کاربرد اهمیت بیشتری نسبت به آزمون شبکه دارد [۴]. بنابراین ابزار تولید ترافیک برای تولید ترافیک واقعی از شبکه باید بتواند انواع پروتکل‌های لایه‌ی کاربرد را پشتیبانی و ترافیک ترکیبی لازم را تولید کند. هر یک از پروتکل‌های کاربرد رفتارهای مختلفی از خود نشان می‌دهند و ممکن است نیاز به انجام وظایف متفاوتی باشد. همچنین یک مسئله‌ی مهم در مورد چنین برنامه‌هایی نیازمندی کیفیت سرویسی است که کاربران انتظار دارند. بنابراین سازمان‌ها نیاز دارند که ویژگی‌هایی مانند زمان پاسخ و توان عملیاتی را مورد ارزیابی قرار دهند.

یک ابزار تولید ترافیک خوب ابزاری است که بار ترکیبی برای بررسی رفتار کاربران برنامه‌های کاربردی تولید می‌کند و معیارهایی مانند زمان پاسخ و توان عملیاتی برنامه‌های کاربردی را گزارش می‌دهد. بار تولیدی باید ویژگی‌های بار واقعی که می‌تواند بر روی کارایی تأثیر داشته باشد، حفظ کند. به عنوان مثال، ترکیب انواع درخواست‌هایی که به سیستم ارسال می‌شود، الگوی ورود کاربران به سیستم، زمان همزمان‌سازی نشست‌ها و غیره. در این راستا هدف مقاله‌ی حاضر تولید ابزاری با ویژگی‌های عنوان شده در سطح بسته و کاربرد است. برای نیل به این مقصود از ابزارهای مناسب سطح بسته‌ی موجود برای لایه‌های پایین‌تر استفاده شده و ترافیک لایه‌ی کاربرد مدل‌سازی گردیده است. در بخش‌های بعد، ابتدا مروری بر پژوهش‌های پیشین در این زمینه خواهیم داشت. سپس به توصیف راهکار پیشنهادی و ابزار پیاده‌سازی شده (DVTG) می‌پردازیم. در ادامه نتایج ارزیابی راهکار پیشنهادی ارائه شده و در بخش آخر نیز نتیجه‌گیری بیان می‌شود.

مروری بر راهکارهای ارائه شده

ابزارهای تولید ترافیک موجود را می‌توان به دو دسته‌ی کلی تجاری و غیرتجاری تقسیم کرد. ابزارهای تجاری ممکن است سخت‌افزاری، نرم‌افزاری و یا ترکیبی از آنها باشند. ابزارهای غیرتجاری (متن باز) فقط نرم‌افزاری هستند. ابزارهای نرم‌افزاری ارزان‌تر و انعطاف‌پذیر بوده و می‌توانند با اجرا بر روی میزبان‌های مختلف برای شبکه‌های با مقیاس بالای امروزی مناسب‌تر باشند. اما از جهت کارایی وابسته به فاکتورهایی مانند سرعت و بار پردازنده، سیستم عامل و فعالیت‌های IO می‌باشند. از جمله ابزارهای نرم‌افزاری تجاری می‌توان به Omnicor-LANTraffic V2 [۵]، Cat Karat packet builder [۶] اشاره کرد. در مقابل، ابزارهای سخت‌افزاری وابسته به سرعت پردازنده و سیستم عامل نیستند، اما گران بوده و انعطاف‌پذیری ابزارهای نرم‌افزاری را

است رفتار تولیدکننده‌ی ترافیک در طول اجرا متناظر با مشاهداتش تغییر کند.

اکثر ابزارهای گروه ۲ و ۳ ترافیک لایه‌ی انتقال تولید می‌کنند و وابسته به کاربرد نیستند. در مقابل، ابزارهای گروه ۱ و ۴ بیشتر به پروتکل‌های لایه‌ی کاربرد می‌پردازند. تولید بارکاری واقعی نیازمند پشتیبانی از هر دو سطح جریان و بسته است و نیز باید توانایی تولید ترافیک با ویژگی‌های خاصی که معمولاً توسط برنامه‌های کاربردی انتخاب می‌شوند، داشته باشند. مانند توان عملیاتی، اعوجاج، تأخیر، نرخ از دست دادن بسته‌ها و زمان رفت و برگشت بسته^۷. این ویژگی‌ها اغلب در ابزارهای سطح بسته قابل محاسبه هستند؛ در صورتی که ابزارهای سطح لایه‌ی کاربرد یا حلقه- بسته اکثراً با استفاده از ابزارهای ردیابی مانند Tcpdump [۲] برخی از انواع این نوع اطلاعات را استخراج می‌کنند و همین مسئله باعث پیچیدگی استفاده از آنها می‌شود. یک ابزار تولید ترافیک خوب باید بتواند در کنار تست کارایی شبکه و اندازه-گیری پارامترهای ذکر شده، امکان تولید سناریوهای واقعی با دستکاری سرآیندها در سطح پروتکل‌های بالاتر مانند RTP^۸ داشته باشد.

ویژگی‌های ترافیک سطح بسته در قالب «زمان بین بسته»^۹ و «اندازه بسته» بیان می‌شود. استفاده از این رویکردها چندین مزیت دارد. اول اینکه قابل توسعه برای مطالعه‌ی پروتکل‌های لایه‌ی کاربرد و ترکیبی از آنها است. بعلاوه مشاهده‌ی ترافیک در سطح بسته امکان تحلیل عمیق شبکه را فراهم می‌کند. زیرا اغلب دستگاه‌های سوئیچینگ بر پایه‌ی بسته‌ها کار می‌کنند. همچنین مشکلاتی مانند تأخیر، اعوجاج و از دست دادن بسته‌ها در این سطح رخ می‌دهد. بنابراین برای ارزیابی تجهیزات شبکه و کارایی آن مناسب هستند. این ترافیک حتی با وجود رمزگذاری‌هایی که در سطح کاربرد انجام می‌شود، قابل مشاهده است. بنابراین برای تست امنیت شبکه و تشخیص ناهنجاری نیز مناسبند [۳]. اما این ابزارها حلقه-باز^{۱۰} بوده و مستقل از مشاهدات شبکه‌ای که میزبان هدف در آن قرار دارد عمل می‌کنند. چنین ابزارهایی هیچ بازخوردی از شبکه نمی‌گیرند. در حالی که در دنیای واقعی بسیاری از برنامه‌های کاربردی نیاز به بازخورد دارند که منجر به تغییر شرایط و رفتار موجودیت‌ها می‌شود. به عنوان مثال در اتصال TCP با تشخیص ازدحام نرخ ارسال ترافیک کاهش می‌یابد. بنابراین اکثر ابزارهایی که ترافیک لایه‌ی کاربرد تولید می‌کنند از رویکرد حلقه-بسته بهره می‌گیرند.

امروزه شبکه‌ها انواع مختلف ترافیک لایه‌ی کاربرد را انتقال می‌دهند. در بسیاری از سیستم‌های توزیع شده تولید بارکاری سطح

^{۱۰} Packet/payload size

^{۱۱} Open-loop

^{۱۲} Distributed Variant Traffic Generator

^۷ Round-Trip Time (RTT)

^۸ Real-time Transport Protocol

^۹ inter-departure-time Inter-packet-time or

می‌کند. اما این ابزار اجازه‌ی سفارشی کردن محتویات بسته‌ها را نمی‌دهد. بنابراین نمی‌توان آن را برای توسعه‌ی پروتکل‌های لایه-۳ کاربرد استفاده کرد؛ همانند ابزارهای مبتنی بر ردیابی مانند TCPReplay [۲۲] که امکان تغییر محتوا را ندارند.

D-ITG [۲۳] می‌تواند ترافیکی مبتنی بر مدل‌های تصادفی بر مبنای PS و IDT تولید کند. همچنین قادر به تولید مدل‌های پیچیده‌تری برای تقلید رفتار پروتکل‌های سطح کاربرد می‌باشد. این ابزار امکان استفاده از الگوهای ترافیکی واقعی و تنظیم پارامترهای لازم در سطوح مختلف [۳] و نیز امکان ارزیابی کارایی شبکه را فراهم می‌کند.

Kolahi و همکاران [۲۴] چهار ابزار Iperf، Netperf [۲۵]، D-ITG و IP Traffic [۱۲] را در تولید ترافیک TCP مقایسه کرده‌اند. طبق گزارش‌های این مقاله Iperf بالاترین و IP Traffic پایین‌ترین توان عملیاتی را دارد. اما زمانی که اندازه‌ی بسته‌ها کوچک‌تر باشد، IP Traffic بهتر عمل می‌کند. در حالی که نویسندگان D-ITG در مقایسه‌ی آن با Mtools [۲۶]، C/RUDE، MGEN، Iperf، UDP generator [۲۷] در شرایط مختلف، در تولید ترافیک UDP نشان داده‌اند که ابزارشان توان عملیاتی بالاتری نسبت به بقیه دارد [۲۳، ۲۸، ۲۹].

در [۳۰] چهار ابزار Iperf [۳۱]، D-ITG، MGEN [۳۲] و RUDE/CRUDE [۳۳] در محیط لینوکس مورد مطالعه قرار گرفته است. طبق بررسی‌های انجام شده در کتابخانه‌های مختلف، مانند IEEE و Google Scholar، محبوب‌ترین ابزار Iperf است. D-ITG جدیدترین ابزار بوده و به صورت فعال در روزرسانی‌ها عمل می‌کند. C/RUDE با وجود عدم ارائه‌ی نسخه-۳ جدید از محبوبیت خوبی برخوردار است. Iperf در مورد تأخیر و MGEN در مورد اعوجاج اطلاعاتی به دست نمی‌دهد. C/RUDE کمترین واریانس را در زمان بین خروج بسته‌ها دارد. بنابراین قادر است تأخیر شبکه را با دقت بیشتری محاسبه کند. اما برای رسیدن به این هدف مصرف پردازنده‌ی بالاتری را متحمل می‌شود. C/RUDE تنها از پروتکل UDP پشتیبانی می‌کند. D-ITG از پروتکل‌های بیشتری در لایه‌ی انتقال پشتیبانی می‌کند، از جمله DCCP و SCTP. این ابزار همچنین علاوه بر IP از پروتکل ICMP نیز در لایه‌ی اینترنت و پروتکل‌های DNS، Telnet، VoIP در لایه‌ی کاربرد پشتیبانی می‌کند.

[۳۰] از ابزار Stress [۳۴] به منظور ایجاد استرس در پردازنده، حافظه و بافر لینوکس استفاده کرده و کارایی ابزارها را تحت شرایط اشباع منابع مقایسه کرده است که خلاصه‌ی نتایج آن در جدول ۲ آمده است.

ندارند. IxNetwork [۷]، Omnicor-NuStreams [۸]، Spirent-Avalanche [۹] و BreakingPoint Elite [10] از جمله ابزارهای سخت‌افزاری تجاری هستند. در سال‌های اخیر راه حل سومی مبتنی بر پردازنده‌های شبکه معرفی شده است که از ویژگی‌های دو دسته‌ی بالا توأم استفاده می‌کند؛ ابزارهایی مانند LANForge [۱۱] و IP Traffic - Test & Measure [۱۲].

ابزارهای رایگان و متن باز در تحقیقات آزمایشگاهی بسیار مورد استفاده قرار می‌گیرند. تمرکز اکثر این ابزارها تولید ترافیک در سطح بسته بوده و تعداد آنها بسیار زیاد است. در جدول ۱ به برخی از آنها اشاره شده است. در ادامه نیز به توصیف مختصری در مورد برخی از آنها می‌پردازیم.

Ostinato [۱۳] ابزاری متن باز و انعطاف‌پذیر برای توسعه‌ی پروتکل‌های جدید می‌باشد. همچنین در حال حاضر دامنه‌ی قابل توجهی از پروتکل‌ها را در لایه‌های ۲ تا ۷ شبکه پشتیبانی می‌کند و بر روی انواع سیستم‌های عامل قابل اجرا است. اما کارایی آن در تولید ترافیک با نرخ بالا ضعیف گزارش شده است [۱۴]. همچنین این ابزار گزارشی از پارامترهای کیفیت سرویس شبکه ارائه نمی‌دهد و برای ضبط بسته‌ها و ارائه‌ی گزارش‌ها، نیاز به نرم‌افزارهای دیگری مانند Wireshark [۱۵] دارد.

Mausezahn [۱۶] (MZ) ابزاری رایگان، متن باز و مبتنی بر لینوکس است که امکان تولید فریم‌های معتبر و نامعتبر را در لایه‌های ۲ تا ۴ فراهم می‌کند. محتوای بسته یا فریم مستقیماً اعلام یا از فایل خوانده می‌شود. در غیر این صورت تهی خواهد بود. به علاوه می‌توان تا اندازه‌ی دلخواه بایت‌های تصادفی تولید کرد. Mausezahn دارای دو حالت مستقیم و تعاملی است. در حالت تعاملی امکان تولید و ارسال چندین جریان از انواع مختلف بسته‌ها به صورت موازی فراهم شده است. اما نسبت به حالت مستقیم دارای سرعت پایین‌تری می‌باشد. این ابزار اساساً برای ارزیابی شبکه‌های چندپخش‌ی^۳ یا VoIP و نیز برای بررسی‌های امنیتی در برابر حملات طراحی شده است. اما در حال حاضر در حالت تعاملی ترافیک VoIP قابل تولید نیست. در مورد پارامترهای شبکه، در حالت مستقیم، نرخ از دست دادن و برهم خوردن ترتیب بسته و اعوجاج فقط برای ترافیک VoIP (مبتنی بر UDP) محاسبه می‌شود. از جمله مطالعاتی که این ابزار را مورد استفاده قرار داده‌اند می‌توان به [۲۰-۱۷] اشاره کرد.

Harpoon [۲۱] یک ابزار سطح جریان می‌باشد که می‌تواند ترافیک را مبتنی بر داده‌های از پیش ثبت شده یا توصیف مبتنی بر پارامتر تولید کند. Harpoon اطلاعات آماری را از داده‌های ثبت شده استخراج می‌کند. اگر شبکه‌ای برای جمع‌آوری اطلاعات در دسترس نباشد، از توزیع‌های آماری عمومی استفاده

جدول ۱: برخی ابزارهای تولید ترافیک متن باز

نام ابزار	پروتکل های پشتیبانی شده	لایسنس	سیستم عامل	ویژگی ها
MGEN	IPv4/v6, TCP, UDP	متن باز	Unix, Windows	تولید داده های آماری برای توان عملیاتی، نرخ از دست دادن بسته ها و تأخیر. بسته های ارسالی را لاگ نمی کند و فقط دریافتی ها را با استفاده از ابزارهای دیگری لاگ می کند. TRPR مانند استفاده شود. opnet و ns-2 می تواند در محیط های شبیه سازی مانند
R/Crude	IPv4, UDP	متن باز	Linux, Solaris, FreeBSD	محاسبه تأخیر، نرخ از دست دادن، اعوجاج، توان عملیاتی و زمان بین ورود. امکان تولید جریانی از بسته ها و یا بسته هایی با اندازه و زمان بین بسته (که در یک فایل ردیاب توصیف می شود) که بطور همزمان قابل تولید هستند. عمل می کند اما بر محدودیت دقت آن ناشی از استفاده از تایمر MGEN شبیه غلبه کرده است. سیستم
Iperf	IPv4/v6, TCP, UDP	متن باز	Linux, Unix, Windows	سادگی. دارای کتابخانه ای برای سادگی برنامه نویسی سوکت. UDP قابلیت آزمون چند بخشی بر روی یک سرور وجود ندارد. TCP و UDP امکان تولید همزمان ترافیک TCP. توانایی تعیین اندازه پنجره اندازه گیری پارامترهای شبکه مانند نرخ از دست دادن بسته ها، اعوجاج و کارایی. TCP اندازه گیری بیشینه ی پهنای باند را پشتیبانی می کند. CBR فقط
D-ITG	IPv4/v6, TCP, UDP, SCTP, DCCP, ICMP, DNS, Telnet, VoIP, Games	متن باز	Linux, Windows	مورد استفاده در پیاده سازی سرویس های امنیتی و آزمایش و بررسی آنها مانند حمله DoS. آزمایش الگوریتم های مسیریابی متمرکز شده. محاسبه ی دامنه ی وسیعی از پارامترهای شبکه مانند تأخیر، تأخیر یک طرفه، نرخ از اعوجاج و توان عملیاتی. RTT دست دادن بسته ها، TCP و UDP استفاده از توزیع های احتمالی مختلف برای تولید ترافیک به طور همزمان چندین جریان می تواند تولید کند. TTL و TOS امکان تنظیم فیلدهای
Netperf	IPv4/v6, TCP, UDP	متن باز	Unix, Linux, Windows	امکان استفاده در انواع مختلف شبکه ها. و ارسال مجدد بسته های از دست رفته. UDP ردیابی بسته های ارسالی از طریق ارزیابی توان عملیاتی و زمان پاسخگویی به میزبان. ارزیابی پارامترهای شبکه مانند کارایی، نرخ از دست دادن بسته ها، زمان پاسخ، توان عملیاتی و میزان مشغول بودن پردازنده. امکان اندازه گیری میزان بهره وری پردازنده.
Mausezahn	ARP, BPDU, PVST, CDP, LLDP, IP, IGMP, UDP, TCP, ICMP, DNS, RTP RX-mode for jitter measurement, Syslog, VLAN, MPLS	متن باز	Linux	و بررسی امنیتی سیستم استفاده VoIP عموماً برای تست شبکه های چندبخشی یا می شود. امکان اندازه گیری اعوجاج. تست استرس. IDS تست نفوذ فایروال و DoS حملات. پیدا کردن باگها در نرم افزار یا تجهیزات شبکه. و اسکن پورت. ping sweep شناسایی حملات با استفاده از (... و malformed رفتار شبکه در شرایط غیر عادی (تست استرس، بسته های

جدول ۲: خلاصه‌ی نتایج کمی ابزارهای تولید ترافیک [۳۰]

	Throughput	Delay	Resource Consumption	Impact of Resource shortage
Iperf	Very Good	Poor	Good	Bad
C/RUDE	Very Good	Very Good	Very Bad	Good
D-ITG	Good	Good	Good	Bad
MGEN	Good	Good	Bad	Good

حال حاضر تنها ترافیک TCP و با رویکرد حلقه-باز تولید می‌کند. در مورد اندازه‌گیری پارامترهای کیفیت سرویس نیز صحبت نشده‌است و همچنین هنوز در دسترس قرار داده نشده‌است. یک دسته از ابزارهای نرم‌افزاری، ابزارهای مبتنی بر هسته هستند. انتظار می‌رود این ابزارها به دلیل دسترسی مستقیم به سخت‌افزار نسبت به ابزارهای مبتنی بر فضای کاربر، کارایی بهتری ارائه دهند. از جمله این ابزارها می‌توان به KUTE [۳۸] اشاره کرد. اما از این ابزار کارایی پایینی گزارش شده است و چندان مورد استفاده قرار نگرفته است. ابزار Pktgen [۳۹] نیز مبتنی بر هسته می‌باشد، اما تنها بر روی پردازنده‌ی شبکه قابل اجرا است. هر یک از ابزارها مزایا و معایب خود را دارند و هیچ یک از آن‌ها تمامی معیارهای عنوان شده را فراهم نمی‌کنند. در محیط ویندوز Iperf و Netperf در تولید ترافیک TCP تا حدی کارایی بهتری ارائه می‌دهند. همچنین Netperf در اجراهای مختلف واریانس کمتر و پایداری بیشتری دارد [۱۴]. D-ITG نسبت به ابزارهای سطح بسته‌ی دیگر تعداد پروتکل‌های بیشتری را در لایه‌های مختلف پشتیبانی می‌کند. این ابزار در برخی تحقیقات برای تولید ترافیک پروتکل‌های جدید مورد استفاده قرار گرفته است. به عنوان مثال [۴۰] ابزاری مبتنی بر D-ITG طراحی کرده و تلاش کرده است با استفاده از فایل‌های لاگ و یک مولفه‌ی تحلیل‌گر این ابزار را توسعه داده و از آن برای تولید انواع ترافیک واقعی در سطح کاربرد استفاده کند، به طوری که بتواند علاوه بر تولید چنین ترافیکی امکان تنظیم بسته و تولید اطلاعات آماری و آزمون کارایی شبکه را نیز داشته باشد که اکثر ابزارهای تولیدکننده‌ی ترافیک لایه‌ی کاربرد امکان آن را ندارند. برای این کار از ویژگی‌های ابزارهای مبتنی بر ردیابی و مبتنی بر تحلیل آماری استفاده کرده و محیطی تشکیل داده‌اند که می‌تواند حلقه-باز و یا حلقه-بسته باشد. زیرا یک رویکرد ترکیبی می‌تواند نماینده بهتری از سیستم‌های واقعی باشد. همچنین این ابزار در [۴۱] برای تولید ترافیک بازی مورد استفاده قرار گرفته است. هدف برخی دیگر از ابزارها تنها تولید ترافیک لایه‌ی کاربرد بوده و مبتنی بر رفتار یا حلقه-بسته عمل می‌کنند مانند Swing [۴۲]، GenSyn [۴۳]، GlobeTraff [۴۴]، NetSpec [۴۵] و Trafgen [۴۶]. این ابزارها ترافیک واقعی شبکه در سطح لایه‌ی

طبق [۳۵] ابزار Hpbench [۳۶]، در تولید ترافیک UDP، توان عملیاتی بالاتری نسبت به Iperf و D-ITG دارد. اگرچه Hpbench اساساً برای شبکه‌های با کارایی بالا طراحی شده است، کارایی خوبی در گیگابیت اترنت نداشته اما بر روی Myrinet و QsNet نتایج خوبی ارائه می‌دهد [۳۷]. مسئله‌ی دیگر این است که ابزار Hpbench نرخ از دست دادن بسته‌های TCP و نیز اعوجاج را گزارش نمی‌کند. همچنین تنها بر روی سیستم عامل لینوکس قابل اجرا است.

[۱] به بررسی سطح کارایی و دقت MGEN، C/RUDE، TG و D-ITG می‌پردازد و نشان می‌دهد که هیچ یک نمی‌توانند به اندازه‌ی پیشینه‌ی پهنای باند لینک (1Gbps) ترافیک تولید کنند. D-ITG که بالاترین توان عملیاتی را ارائه داده است، در حدود 620Mbps به اشباع می‌رسد. همچنین اجرای برخی وظایف داخلی (کاری که توسط خود ابزار انجام می‌شود مانند محاسبات، ثبت وقایع و ...) و خارجی (کار دیگری به جز تولید ترافیک که بر روی ماشین مبدأ در حال اجرا توسط پردازنده می‌باشد) در کارایی ابزارها تأثیر می‌گذارد. زیرا این وظایف منابع را مصرف کرده، دسترس‌پذیری پردازنده و گذرگاه‌ها را کاهش می‌دهند و باعث کاهش نرخ ترافیک تولیدی می‌شوند. همچنین در زمانبندی پردازنده، دقت زمان بین خروج بسته‌ها را کم می‌کنند. برای غلبه بر چنین مشکلی D-ITG و MGEN از مکانیزم سرشماری استفاده می‌کنند. ثبت اطلاعات نیز باعث کاهش توان عملیاتی می‌شود. برای حل این مسئله D-ITG اطلاعات مربوط به بسته‌ها را موقتاً در بافر حافظه ذخیره می‌کند و پس از تکمیل کار به دیسک منتقل می‌کند. همچنین D-ITG و TG برای کاهش مقدار داده از فایل‌های باینری به جای فایل‌های متنی استفاده می‌کنند. بعلاوه استفاده از ریسمان‌ها برای افزایش مقیاس‌پذیری و ارسال ترافیک از چندین منبع نیز تأثیر مستقیم در کارایی این ابزارها دارد.

پیشرفت‌های سخت‌افزاری امکان موزایی‌سازی در تولید بسته را فراهم می‌کند که توسط هیچ یک از ابزارهای موجود در نظر گرفته نشده است. [۱۴] به ارائه‌ی یک ابزار متن‌باز تولید ترافیک سطح بسته با استفاده از این ویژگی سخت‌افزارهای جدید می‌پردازد و طبق گزارش‌ها به نتایج مطلوبی رسیده است. اما در

- ۹ مقیاس پذیری
- ۱۰ امکان اجرا بر روی انواع سیستم‌های عامل
- ۱۱ اندازه‌گیری دقیق پارامترهای شبکه
- ۱۲ تداوم پشتیبانی و توسعه‌ی پروتکل‌های جدید

همان‌طور که در بخش قبل بحث شد، تاکنون ابزارهای مختلف و مفیدی در سطح لایه‌ی پیوند داده، لایه‌ی شبکه و لایه‌ی انتقال ارائه شده و در تحقیقات بسیاری مورد استفاده قرار گرفته‌اند. برخی از این ابزارها گزارش‌های مفید و مورد نیاز این لایه‌ها را برای انواع آزمون‌ها فراهم می‌کنند. بنابراین برای تولید این نوع ترافیک‌ها از ابزارهای موجود استفاده شده است. آزمایش‌های مختلفی به منظور انتخاب ابزار مناسب انجام شده است. این آزمایش‌ها بر روی دو سیستم Ubuntu 12.04 TLS هر یک با دو پردازنده با سرعت 3.33GHZ، حافظه‌ی 1G و اترنت 1000Mbps و همچنین آزمایش دیگری بر روی دو سیستم Ubuntu 12.04 TLS ۳۲ بیتی هر یک با یک پردازنده 2.5GHZ و حافظه‌ی 512M انجام شده است. ابزارها از نظر توان عملیاتی و نرخ از دست دادن بسته‌ها در تولید ترافیک UDP و TCP مورد مقایسه قرار گرفته‌اند. قابل ذکر است آزمایش‌ها برای بسته‌هایی با محتوایی به اندازه‌ی ۶۴، ۲۵۶، ۱۰۲۴ و ۱۴۶۰ بایت به مدت ۵ ثانیه انجام شده است. برخی از نتایج آزمون‌ها در نمودارهای ۱ تا ۳ آورده شده است. (به دلیل اینکه برخی ابزارها میزان توان عملیاتی را گزارش نمی‌دهند از ابزار Wireshark برای این هدف استفاده شده است).

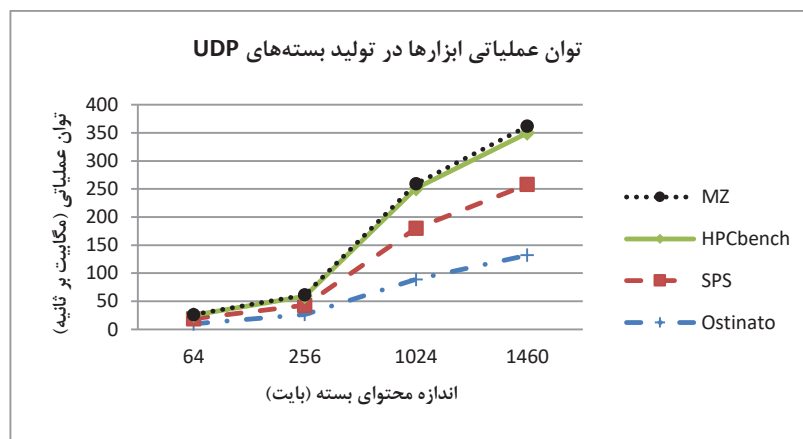
کاربرد تولید می‌کنند. اما برای تست کارایی شبکه مناسب نیستند. زیرا به بررسی برخی یا هیچ یک از پارامترهای کیفیت سرویس شبکه نمی‌پردازند. آنها برای چنین بررسی‌هایی نیاز به استفاده از ابزارهای دیگر دارند.

انتخاب راهکارهای مناسب

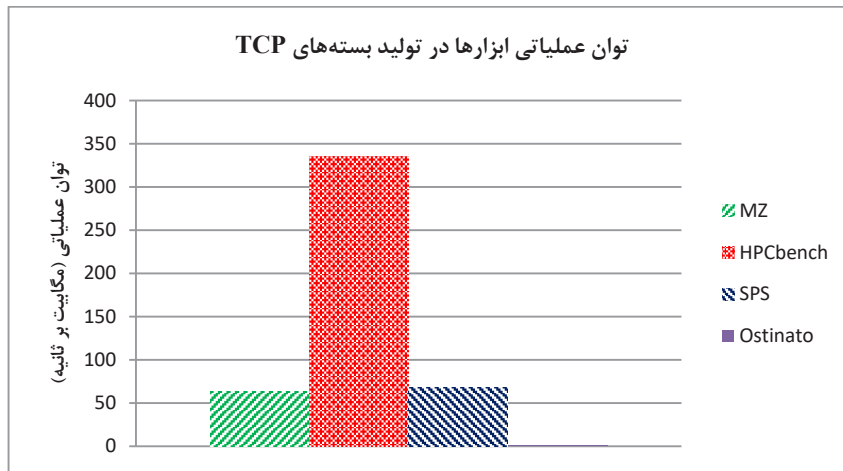
در این بخش، معیارهای انتخاب راهکار مناسب برای تولید انواع ترافیک‌ها در سطوح مختلف و ارزیابی تجهیزات شبکه ارائه خواهد شد.

به طور کلی یک ابزار تولید ترافیک خوب باید دارای ویژگی‌های زیر باشد:

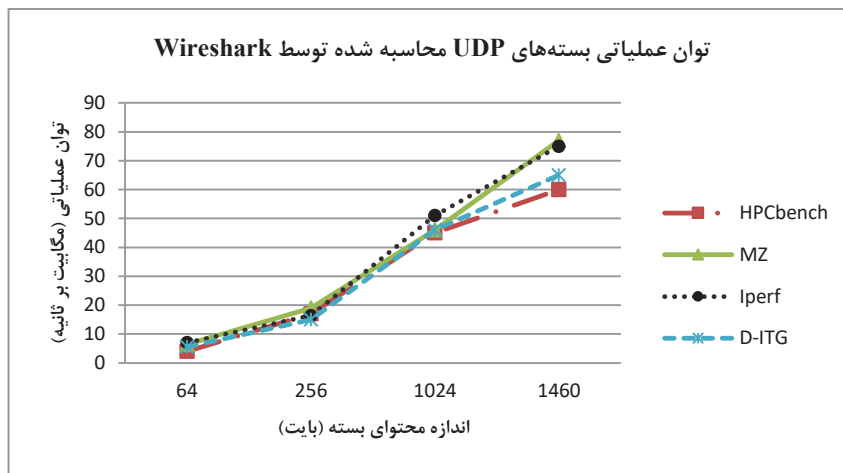
- ۱ آزمون کارایی شبکه از نظر پارامترهایی مانند نرخ از دست دادن بسته، اعوجاج، RTT، توان عملیاتی و تأخیر
- ۲ آزمون تجهیزات مختلف شبکه مانند روتر، فایروال و...
- ۳ آزمون تمام لایه‌های پشته‌ی پروتکلی
- ۴ تحلیل و تولید گزارش
- ۵ تولید ترافیک واقعی
- ۶ امکان تولید مجدد ترافیک جهت ارزیابی سناریوهای مختلف
- ۷ امکان تولید نرخ انبوهی از ترافیک در لینک‌های سرعت بالا
- ۸ تولید ترافیک حمله



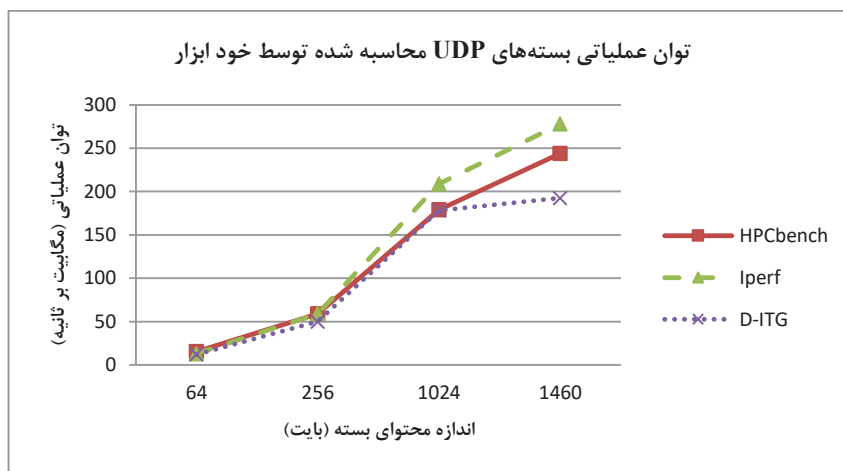
نمودار ۱: مقایسه‌ی توان عملیاتی در تولید ترافیک UDP در آزمایش اول



نمودار ۲: مقایسه‌ی توان عملیاتی در تولید ترافیک TCP در آزمایش اول



نمودار ۳: مقایسه‌ی توان عملیاتی در تولید ترافیک UDP در آزمایش دوم



نمودار ۴: مقایسه‌ی توان عملیاتی در تولید ترافیک UDP در آزمایش دوم توسط ابزارها

اما ویژگی‌های مثبت بسیاری از جمله تولید انواع مختلف پروتکل‌ها و ارائه‌ی گزارش‌های کامل دارد که بسیار مورد استفاده قرار می‌گیرند.

مطالعات انجام شده بر روی مقالات مختلف نشان می‌دهد این ابزارها بسیاری از ویژگی‌های ضروری نشست‌های لایه‌ی کاربرد را در نظر نگرفته‌اند. در نتیجه برای تولید بسته‌ها در لایه‌ی انتقال از ابزار D-ITG استفاده شد که به راحتی قابل جایگزینی است. علت انتخاب نشدن ابزارهای دیگر این است که با وجود سرعت بالاتر بسیاری ویژگی‌های دیگر لازم برای مدل‌سازی را دارا نبودند. به عنوان مثال Iperf اندازه‌ی تعیین شده برای بسته‌ها را که پارامتری مهم در مدل‌سازی است تغییر می‌دهد. یا MZ اندازه‌ی بزرگتر از ۱۰۰۰ را نمی‌پذیرد. همچنین D-ITG گزارش کامل‌تری نسبت به باقی ابزارها ارائه می‌دهد و با استفاده از موازی‌سازی، به صورت چندریسمانی یا با اجرا بر روی چند سیستم، می‌توان آن را نیز افزایش داد. به علاوه از نظر پشتیبانی و بروز رسانی فعال می‌باشد. اما به دلیل عدم امکان تعیین مقدار پارامترهای آماری طبق داده‌های واقعی، پروتکل‌های لایه کاربرد آن مورد استفاده قرار نگرفت.

MZ نیز به دلیل سرعت بالای خود و نیز تولید ترافیک لایه‌ی ۲ در تولید این نوع ترافیک مورد استفاده قرار گرفت. ابزار Httpperf [۴۸] نیز ابزاری قوی است که در بسیاری از مقالات برای تولید ترافیک Http[32] مورد استفاده قرار گرفته است [۴۹-۵۱]. این ابزار گزارش‌های خوبی از ترافیک ارسالی ارائه می‌دهد که در ابزار پیشنهادی مورد استفاده قرار گرفته است.

در نمودارهای ۱ و ۲ ابزارهای MZ، HPCbench، SPS [۴۷] و Ostinato مورد مقایسه قرار گرفته‌اند. از بین این ابزارها تنها HPCbench توان عملیاتی را گزارش می‌کند. بنابراین نتایج این نمودار طبق Wrieshark بدست آمده است. طبق نمودار ۱ در تولید ترافیک UDP، MZ بالاترین توان عملیاتی را دارا است. HPCbench بسیار نزدیک به آن بوده و Ostinato دارای کمترین توان عملیاتی است. اما طبق نمودار ۲ در تولید ترافیک TCP ابزار HPCbench توان بسیار بالاتری نسبت به بقیه دارد. MZ و SPS توان تقریباً یکسانی ارائه داده‌اند.

در نمودار ۳ ابزارها بر روی سیستم دوم مورد مقایسه قرار گرفته‌اند. طبق این نمودار MZ و Iperf دارای بالاترین توان عملیاتی می‌باشند و HPCbench و D-ITG دارای توان نزدیک به هم هستند. اما طبق نمودار ۴ که نتایج ارائه شده توسط خود ابزارها می‌باشد (MZ نتیجه‌ای گزارش نمی‌دهد)، Iperf بالاترین توان عملیاتی را دارد. به علاوه واضح است که نتایج ارائه شده در Wireshark تفاوت قابل ملاحظه‌ای با نتایج ابزارها دارد. به عنوان مثال در نمودار ۳ توان D-ITG نسبت به HPCbench بالاتر است که این نتیجه در نمودار ۴ بالعکس است.

در کل ابزارهای MZ و Iperf در تولید ترافیک UDP بالاترین توان عملیاتی را دارند و HPCbench در تولید ترافیک TCP نتایج بهتری ارائه کرده است. اما با توجه به عدم دقت Wireshark نمی‌توان مقایسه‌ی دقیقی داشت. همچنین MZ می‌تواند با تولید ترافیک حمله برای تست انواع تجهیزات امنیتی مورد استفاده قرار گیرد. با وجود اینکه D-ITG توان کمتری ارائه کرده است،

جدول ۳: ابزارهای مورد استفاده و پروتکل‌های هر یک در DVTG

ابزار مورد استفاده	پروتکل
MZ	ARP, BPDU, Direct access link
D-ITG	TCP, UDP, SCTP, DCCP, ICMP
Httpperf	Http, Https
پیاپی سازی طبق داده‌های آماری	P2P, Web, Streaming, VoIP, FTP, Email

مجموعه‌ای از پروتکل‌ها هستند. شکل ۱ شمای کلی از معماری ابزار را نمایش می‌دهد.

مولفه‌ی مدیریت

این مولفه وظیفه‌ی هماهنگی و مدیریت تولید ترافیک و ارائه‌ی گزارش‌های دریافتی از بخش‌های دیگر و یکپارچه‌سازی آن‌ها را بر عهده دارد. مدیر عامل‌های ارسال‌کننده و دریافت‌کننده را انتخاب و به آن‌ها اعلام می‌کند که آماده‌ی ارسال یا دریافت بسته باشند. تمامی ویژگی‌های ترافیک ارسالی نیز از جمله تعداد بسته‌های ارسالی، اندازه بسته‌ها، زمان بین ورود بسته، تعداد

معماری پیشنهادی

مدلی که برای ابزار ارزیابی ارائه شده است به صورت توزیع شده و دارای سه مؤلفه می‌باشد: مدیریت، ارسال‌کننده و دریافت‌کننده. هدف مؤلفه‌ی مدیریت متمرکزسازی امور می‌باشد. ارسال‌کننده‌ها و دریافت‌کننده‌ها به هر تعداد می‌توانند در سیستم موجود باشند. هر ارسال‌کننده می‌تواند برای چندین دریافت‌کننده ترافیک ارسال کند. هر مؤلفه دارای یک گرداننده^{۱۸} است که وظیفه‌ی برقراری ارتباط مؤلفه را با سایر اجزا بر عهده دارد. عامل‌های ارسال و دریافت هر کدام مسئول یک یا

- تعداد بسته‌های ارسالی
- میانگین تأخیر
- میانگین اعوجاج
- پهنای باند
- نرخ ارسال بسته‌ها
- نرخ از دست دادن بسته‌ها
- تعداد و نرخ اتصالات و پاسخ در مورد پروتکل Http[32]

مؤلفه‌ی دریافت

مدیر به تمامی گره‌هایی که برای دریافت ترافیک انتخاب شده‌اند، سیگنالی ارسال و به آنها اعلام می‌کند که آماده‌ی دریافت جریان باشند. چنانچه گره‌های مذکور آماده‌ی دریافت نباشند، به عنوان مثال گذرگاه دریافت مشغول باشد و یا اتصال با شبکه برقرار نباشد، خطای مربوطه به مدیر اطلاع داده می‌شود. هر مؤلفه‌ی دریافت برای هر یک از انواع ترافیک دریافتی عامل مخصوص به آن نوع ترافیک را دارا می‌باشد. هر عامل می‌تواند اطلاعات مربوط به جریان دریافتی خود را ثبت و گزارش‌های لازم را تولید و در صورت لزوم برای مؤلفه‌ی مدیریت ارسال کند.

اتصالات و... توسط مدیر تعیین می‌شود. در مورد پروتکل‌های لایه-ی کاربرد، به جای تعداد بسته‌ها، مقدار داده‌هایی که باید ارسال گردد در بخش مدیریت تعیین می‌شود. همچنین مدیر گزارش‌های مختلف از ترافیک ارسالی را جمع‌آوری و نمایش می‌دهد.

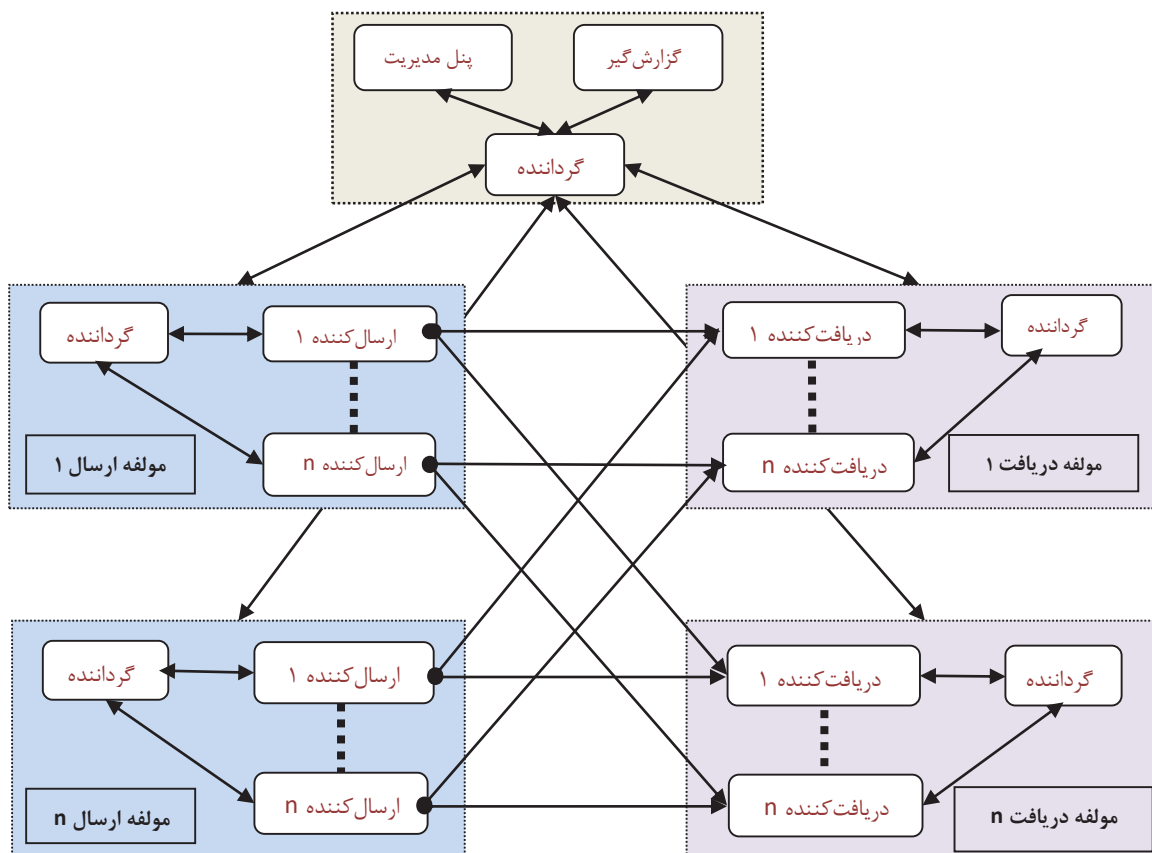
مؤلفه‌ی ارسال

عامل ارسال ترافیک پس از دریافت نوع ترافیک ارسالی و تمامی اطلاعات لازم از مدیر، شروع به ارسال جریان برای عامل‌های دریافت مشخص شده می‌نماید. مؤلفه‌ی ارسال ممکن است کاربر یا سرویس‌دهنده باشد که توسط مدیر تعیین می‌شود. در ادامه جزئیات بخش‌های مختلف این مؤلفه تشریح خواهد شد.

گزارش‌گیری

هر یک از ارسال‌کننده‌های ترافیک گزارش خود را برای مدیر ارسال می‌کنند. گزارشات شامل پارامترهای زیر می‌باشد:

- ارسال‌کننده
- دریافت‌کننده
- نوع پروتکل
- مدت زمان ارسال



شکل ۱: معماری پیشنهادی برای ابزار تولید ترافیک توزیع شده

روش پیشنهادی برای تولید ترافیک

به منظور تولید ترافیک در لایه‌های مختلف، پس از مطالعه و بررسی رفتار هر پروتکل در بستر شبکه، روش تولید داده آن انتخاب شد. در ادامه، در مورد روش تولید ترافیک پروتکل‌های مهم توضیح داده خواهد شد.

تولید ترافیک از سمت کاربر به سرویس دهنده

این بخش از ابزار به ارسال ترافیک حجیم از سمت کاربر به سمت سرویس دهنده می‌پردازد. به عنوان مثال ارسال تعداد زیادی اتصالات TCP و بررسی وضعیت شبکه و گزارش‌گیری از آن. این بخش شامل ارسال ترافیک از سه لایه‌ی پیوند داده‌ها، انتقال و کاربرد می‌باشد. هر عامل ارسال ترافیک می‌تواند برای هر تعداد عامل دریافت، ترافیک ارسال کند.

۱) ترافیک لایه پیوند داده‌ها

ترافیک لایه‌ی پیوند داده‌ها شامل پروتکل‌های ARP، BPD و ارسال مستقیم داده‌های خام با استفاده از ابزار MZ تولید می‌شود و تمامی ویژگی‌های این ابزار در تولید ترافیک لحاظ شده است. تعداد فریم‌ها، طول فریم، آدرس مبدأ و مقصد قابل تعریف است.

۲) ترافیک لایه انتقال

ترافیک لایه‌ی انتقال با استفاده از ابزار D-ITG تولید و ارسال می‌شود. ویژگی‌های مختلف از هر نوع ترافیک در این ابزار قابل تنظیم می‌باشد. شامل تعداد بسته‌ها، طول بسته‌ها (ثابت یا متغیر با استفاده از انواع توابع توزیع آماری)، سرعت ارسال (ثابت یا متغیر با استفاده از انواع توابع توزیع آماری) و... محتوای بسته‌ها می‌تواند تصادفی تولید گردد و یا از فایل خوانده شود. همچنین می‌توان با تعریف تعداد ریسمان‌های پردازشی بیشتر (فراخوانی همزمان ابزار به تعداد ریسمان‌ها) و در نتیجه استفاده از توان تمام پردازنده‌های سیستم سرعت تولید و ارسال ترافیک را افزایش داد. پروتکل‌های قابل تولید توسط این ابزار شامل TCP، UDP، SCTP، DCCP و ICMP می‌باشد.

۳) ترافیک لایه‌ی کاربرد

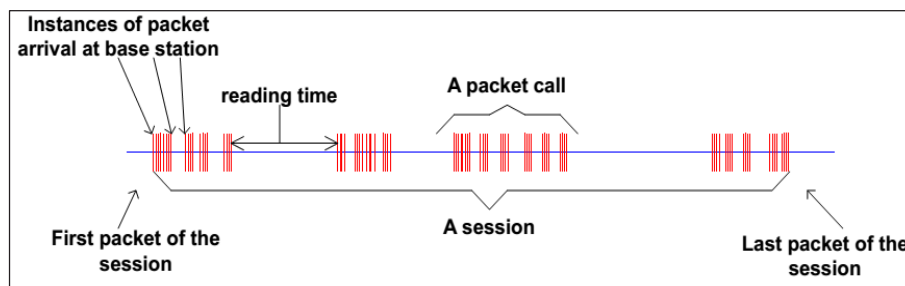
از جمله آزمون‌های کارایی شبکه، آزمون اتصالات Http و ظرفیت و زمان پاسخگویی سرویس دهنده‌ی وب می‌باشد. در این راستا بخشی از ابزار به تولید ترافیک Http/Https با استفاده از ابزار Httpperf که در تحقیقات بسیاری مورد استفاده قرار گرفته است می‌پردازد. این ابزار بسیاری از ویژگی‌های این نوع ترافیک را در نظر گرفته و پارامترهایی از جمله کل تعداد اتصالات، تعداد فراخوانی‌ها به ازای هر اتصال، اندازه‌ی بافر مبدأ و مقصد، نسخه‌ی Http، مدت ارسال، مدت زمان اتصال، روش رمزگذاری SSL و ... قابل تنظیم می‌باشد.

تولید ترافیک از سمت سرویس دهنده به کاربر

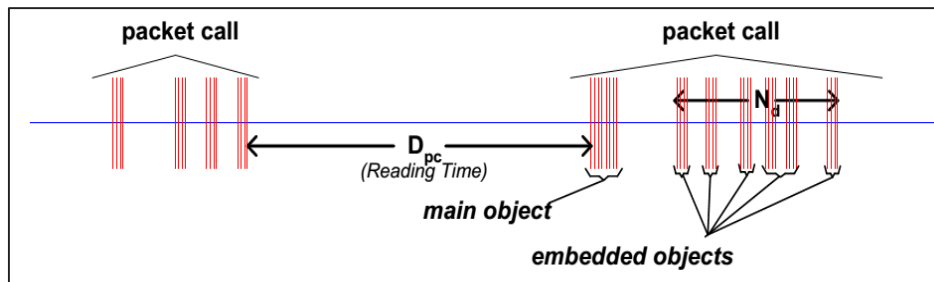
مدل‌سازی ترافیک و ایجاد محیطی مشابه محیط واقعی شبکه مسئله‌ای مهم در پیش‌بینی رفتار کاربران، عیب‌یابی و ارزیابی کارایی می‌باشد. چنانچه مدل ارائه شده دقیق نباشد ممکن است موجب نتیجه‌گیری‌های اشتباه از نتایج شبیه‌سازی گردد. مدل‌سازی ترافیک، به‌کارگیری ریاضی و فرآیندهای تصادفی برای بیان رفتار ترافیک می‌باشد. تاکنون مدل‌های بسیاری برای انواع ترافیک موجود در اینترنت ارائه و بررسی شده است. با تکیه بر این مطالعات و بررسی آنها در مورد هر یک از انواع ترافیک لایه‌ی کاربرد که بیشترین سهم مصرف پهنای باند را دارند مدلی ارائه و پیاده‌سازی شده‌است که در ادامه به توصیف هر یک می‌پردازیم.

۱) Web

یک صفحه‌ی وب یک سند HTML به همراه تعدادی اشیای تعبیه شده بر روی آن می‌باشد. شکل ۲ یک نشست جستجوی وب معمولی را نمایش می‌دهد. هر نشست به دوره‌های On و Off که نشان‌دهنده‌ی زمان دانلود صفحات (packet call) و زمان بین دانلودها (reading time) است تقسیم می‌شود. محتوای هر packet call به صورت شکل ۳ می‌باشد. مرورگر وب سرویس‌دهی به درخواست کاربر را با واکنشی صفحه html اولیه آغاز و سپس آن را برای بدست آوردن اشیای تعبیه شده تجزیه می‌کند.



شکل ۲: شمای کلی یک نشست وب [۵۲]



شکل ۳: محتوای صفحات [۵۲]

برای دانلود اشیای تعبیه شده در صفحه‌ی درخواستی توسط مرورگر ارسال می‌شود و به صورت فرآیندی خودمشابه مدل می‌گردد.

پارامترهای لازم برای مدل‌سازی وب به صورت زیر و توزیع‌های تصادفی مورد استفاده به صورت جدول ۴ می‌باشد:

- اندازه‌ی شی اصلی یا سند Html صفحه
- اندازه‌ی اشیای تعبیه شده در صفحه
- تعداد اشیای تعبیه شده
- زمان بین دانلود
- زمان تجزیه

اغلب ورود درخواست به سرورها با استفاده از فرآیند پواسون مدل می‌شود. اما مطالعات بسیاری نشان داده است که برای برخی از انواع ترافیک این فرآیند مدل مناسبی برای نرخ ورود و یا زمان بین ورود ارائه نمی‌دهد. طبق [۵۳] استفاده از توزیع پواسون زمانی مناسب است که شروع ارسال توسط کاربر آغاز گردد. در غیر این صورت فرآیندهای خودمشابه^{۱۹} برای مدل کردن رفتار ترافیک مناسب‌تر خواهند بود. در این نوع فرآیند، ترافیک در بازه‌های زمانی مختلف رفتار آماری مشابهی دارد. ترافیک وب خودمشابه می‌باشد و شروع نشست توسط کاربر صورت می‌گیرد. بنابراین نرخ ورود درخواست پواسون و در نتیجه زمان بین ورود درخواست‌ها به سرور نمایی خواهد بود. اما درخواست‌های بعدی

جدول ۴: توابع توزیع و پارامترهای Web [۵۳]

Component	Distribution	Parameters	PDF
Main object size	Truncated lognormal	Mean = 10710 bytes Std.Dev. = 25032 bytes Min = 100 bytes Max = 2 Mbyte	$f_x = \frac{1}{\sqrt{2\pi\sigma x}} \exp\left[\frac{-(\ln x - \mu)^2}{2\sigma^2}\right], x > 0$
Embedded object size	Truncated lognormal	Mean = 7758 bytes Std.Dev. = 126168 bytes Min = 50 bytes Max = 2 Mbyte	$f_x = \frac{1}{\sqrt{2\pi\sigma x}} \exp\left[\frac{-(\ln x - \mu)^2}{2\sigma^2}\right], x > 0$
Number of embedded objects	Truncated pareto	Mean = 5.64 Max = 53	$f_x = \frac{\alpha k^\alpha}{x^{\alpha+1}}, k \leq x \leq m$ $f_x = \left(\frac{k}{m}\right)^\alpha, x = m$ $\alpha = 1.1, k = 2, m = 55$
Reading time	Exponential	Mean = 30 sec	$f_x = \lambda e^{-\lambda x}, x \geq 0$
Parsing time	Exponential	Mean = 0.13 sec	$f_x = \lambda e^{-\lambda x}, x \geq 0$

Packet call وابسته به نسخه‌ی Http استفاده شده توسط سرویس‌دهنده و مرورگر وب می‌باشد. در نسخه Http/1.0 برای صفحه‌ی اصلی و هر یک از اشیای تعبیه شده یک اتصال TCP مجزا استفاده می‌شود (Http/1.0-burst). اکثر مرورگرها اشیای

پروتکل لایه‌ی انتقال مورد استفاده برای صفحات وب TCP می‌باشد. طبق [۵۲] اندازه‌ی بسته‌های TCP مورد استفاده برای صفحات وب بیشتر ۵۷۶ یا ۱۵۰۰ بیتی هستند. اما طبق [۵۴] متوسط اندازه‌ی بسته‌ها ۷۵۸ بایت است. ترافیک TCP داخل هر

^{۱۹} Self-similar

پایدار برای دانلود اشیا استفاده می‌شوند. به این معنی که اشیا به صورت سریال از طریق یک اتصال TCP ارسال می‌شوند. طبق همان مرجع نسبت استفاده از هر یک از این دو نسخه ۵۰٪-۵۰٪ است.

تعبیه شده را با استفاده از چندین اتصال TCP همزمان دانلود می‌کنند. بیشینه‌ی تعداد اتصالات همزمان قابل تنظیم است. طبق [۵۲] که مربوط به سال ۲۰۰۴ می‌باشد اکثر مرورگرها از ۴ اتصال همزمان استفاده می‌کنند. در Http/1.1 اتصالات TCP

جدول ۵: توابع توزیع و پارامترهای P2P [۵۵]

Component	Distribution	Parameters	PDF
Session inter-arrival time	Second-order Hyper-Exponential		$f(x; \theta_1, \theta_2) = pf_1(x; \theta_1) + (1-p)f_2(x; \theta_2)$ $\theta_1 = 0.115, \theta_2 = 0.75, p = 0.8$
Session size	Truncated Lognormal	Mean = 32MB Std.Dev.= 99.4MB Max = 1652MB	$f_x = \frac{1}{\sqrt{2\pi}\sigma x} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right], x > 0$

جریان‌های صدا و تصویر با روش‌های مختلف کدگذاری یا فشرده می‌شوند. تحلیل‌ها بر روی جریان‌های داده‌ای با روش کدگذاری MPEG2، وجود سه نوع فریم I، B و P را در اندازه‌های مختلف نشان داده‌است که دارای همبستگی بسیار پیچیده‌ای هستند. شکل ۴ یک نمونه از ساختار جریان را نمایش می‌دهد. در پایین-ترین سطح ۱۵ فریم دیده می‌شود که تشکیل یک GOP می‌دهند. هر صحنه از ویدئو از تعدادی GOP تشکیل می‌شود. پارامترهای مهم و ضروری برای مدلسازی جریان‌های صدا و تصویر توزیع طول صحنه، توزیع اندازه‌ی فریم و توابع همبستگی بین اندازه‌ی فریم‌ها و ساختار جریان می‌باشد.

طبق مطالعات انجام شده، بیشتر ترافیک Streaming مربوط به Flash می‌باشد که با روش H.264 کدگذاری می‌شود. ساختار کلی H.264 شبیه MPEG2 است اما از نظر توزیع اندازه‌ی فریم و ساختار جریان دارای تفاوت‌هایی هستند. در این روش کدگذاری از دو نوع فریم دیگر Si و Sp برای سوئیچ بین جریان‌ها استفاده می‌شود. اما به دلیل جدید بودن این دو مفهوم و اینکه در مورد پیاده‌سازی آنها بحثی صورت نگرفته است، هیچ مدل ترافیکی در مورد آنها در مقالات ارائه نشده است. طبق بررسی‌های انجام شده در [۵۶] با توجه به نتایج بررسی‌ها رفتار ترافیکی MPEG2 و H.264 بسیار شبیه به هم می‌باشد. بنابراین مدل ارائه شده مبتنی بر MPEG می‌باشد.

۲ P2P

از میان پروتکل‌های P2P، پروتکل BitTorrent بیشترین میزان محبوبیت را در میان کاربران دارد. بنابراین در اینجا مدلی مبتنی بر این پروتکل ارائه شده است. محتوای توزیع شده در BitTorrent به تعداد زیادی piece تقسیم می‌شود. پروتکل BitTorrent از Swarming برای توزیع بار بین peerها استفاده می‌کند. هر swarm شامل حداقل یک tracker است که آدرس آی‌پی peerها را برای همدیگر فراهم می‌کند و خود در توزیع محتوا شرکت نمی‌کند. peerها به دو دسته‌ی leecher و seederها تقسیم می‌شوند. Seedها کل محتوای مورد نظر را دارند و فقط به آپلود می‌پردازند. Leacherها در ابتدا هیچ محتوایی در دسترس ندارند و تنها به دانلود می‌پردازند اما با گذشت زمان و جمع‌آوری بخش-هایی از محتوا آپلود نیز انجام می‌دهند. پارامترهای لازم برای مدل‌سازی این پروتکل عبارتند از [۵۵]:

- زمان بین نشست
- اندازه‌ی نشست

اندازه‌ی pieceها معمولاً 256 کیلوبایت می‌باشد. طبق [۵۴] میانگین اندازه‌ی بسته‌های P2P برابر با ۵۸۷ بایت است.

۳ Streaming

مدل ترافیک جریان‌های داده‌ای صدا و تصویر با توجه به ثابت (CBR) یا متغیر (VBR) بودن نرخ ارسال بیت متفاوت است. CBR ممکن است در برخی محیط‌ها مانند شبکه‌های اتصال‌گرا همچون ATM مفید باشد. اما در شبکه‌های Packet Switched روش VBR کارا تر می‌باشد. بنابراین مدل ارائه شده مبتنی بر جریان‌های VBR است.

۵ رویکردهای دیگر: شامل مدل‌های مختلفی مانند فرآیند $M/G/\infty$ و TES. M/G/∞ تنها برای SRD مناسب است و TES نیاز به دسترسی به تستر تستر دارد.

مدل ارائه شده در اینجا مبتنی بر مدل AR می‌باشد. زیرا این روش در تحقیقات بسیاری مورد استفاده قرار گرفته و طبق [۵۷] با در نظر گرفتن توابع توزیع مناسب برای اندازه‌ی فریم‌ها می‌تواند کارایی بهتری از خود نشان دهد. در عین حال پیچیدگی روش‌های دیگر را ندارد. توابع توزیع هر یک از پارامترها در جدول ۶ آمده است.

طبق توضیحات [۵۸] اثرات همبستگی بین فریم‌های P و B نسبت به I قابل چشم‌پوشی است. تابع همبستگی فریم‌های I به صورت زیر تعریف می‌شود:

$X_I(n)$ اندازه‌ی n امین فریم I:

$$X_I(n) = a_1 X_I(n-1) + a_2 X_I(n-2) + \varepsilon(n)$$

$$\varepsilon(n) = N(0, \sigma_\varepsilon)$$

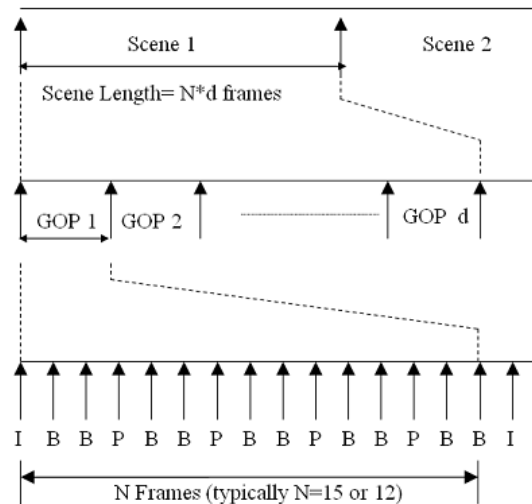
۴ FTP

در پروتکل انتقال فایل، هر نشست شامل دنباله‌ای از انتقال فایل - ها است که با فاصله‌ی زمانی reading time منتقل می‌شوند. پروتکل لایه‌ی انتقال آن TCP بوده و متوسط اندازه‌ی بسته‌ها طبق [۵۲] مشابه وب و طبق [۵۴] به طور متوسط ۹۰۲ بایت می‌باشد.

پارامترهای مدل‌سازی این نوع ترافیک عبارتند از [۵۲, ۵۹, ۶۰]:

- زمان بین نشست
- اندازه‌ی نشست یا فایل
- اندازه‌ی اتصالات داخل نشست
- تأخیر بین اتصالات

جدول ۷ توابع توزیع و مقادیر مناسب پارامترها برای پروتکل FTP را نمایش می‌دهد.



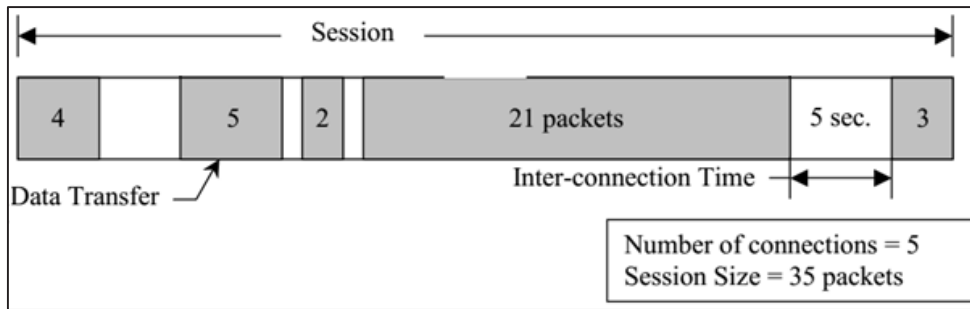
شکل ۴: شمای کلی ساختار جریان [۵۶]

انواع مختلف مدل‌های موجود به پنج دسته‌ی زیر قابل تقسیم است [۵۷]:

- ۱ مدل‌های AR: در این مدل اندازه‌ی فریم بعدی در یک جریان ویدئو با تابعی براساس اندازه‌ی فریم‌های قبلی در یک پنجره‌ی زمانی بدست می‌آید. مشکل این مدل‌ها این است که مدل یکتایی برای آن ارائه نشده است.
- ۲ مدل‌های مبتنی بر فرآیندهای مارکو: فرآیندها یا زنجیره‌های مارکو برای تعیین اندازه‌ی فریم/GOP یا نرخ بیت مورد استفاده قرار می‌گیرند. این مدل‌ها کارایی و دقت مناسبی دارند اما تقسیم دقیق حالت-های ویدئو به زنجیر مارکو به دلیل طبیعت پویای ترافیک ویدئو مشکل است.
- ۳ مدل‌های ARIMA کسری و خودمشابه: این مدل‌ها وابستگی طولانی مدت (LRD) ترافیک ویدئو را ثبت می‌کنند. این مدل‌ها پیچیدگی محاسباتی بالایی دارند و ویژگی‌های وابستگی کوتاه مدت (SRD) ترافیک ویدئو را در نظر نمی‌گیرند.
- ۴ مدل‌های Wavelet: این تکنیک‌ها برای ثبت LRD و SRD استفاده می‌شوند. در کل این روش‌ها دقت بالایی دارند اما پیاده‌سازی آن پیچیده بوده و نیازمند تعیین دقیق تعداد سطوح تجزیه است.

جدول ۶: توابع توزیع و پارامترهای Streaming [۵۸]

Component	Distribution	Parameters	PDF
Inter-session time	Exponential	Mean = 10 sec	$f_x = \lambda e^{-\lambda x}, x \geq 0$
Video size	Log Normal	Mean = 43k frame	$f_x = \frac{1}{\sqrt{2\pi\sigma x}} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right], x > 0$ $\sigma = 0.63, \mu = 10.5$
Scene size	Geometric	Mean = 100 frame	$f_x = (1-p)^{x-1} p, x \in N$
I frame size	Gamma	Mean = 8976 bytes Std.Dev. = 1392 bytes $\sigma_x = 21$ $a_1 = 0.7, a_2 = 0.05$	$f_x = \frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-\frac{x}{\theta}}, x > 0$
B frame size	Gamma	Mean = 7680 bytes Std.Dev. = 624 bytes	$f_x = \frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-\frac{x}{\theta}}, x > 0$
P frame size	Gamma	Mean = 3216 bytes Std.Dev. = 432 bytes	$f_x = \frac{1}{\Gamma(k)\theta^k} x^{k-1} e^{-\frac{x}{\theta}}, x > 0$



شکل ۵: شمای کلی یک نشست FTP [۵۲, ۵۹, ۶۰]

جدول ۷: توابع توزیع و پارامترهای FTP [۵۲, ۵۹, ۶۰]

Component	Distribution	Parameters	PDF
Session inter-arrival time	Exponential		$f_x = \lambda e^{-\lambda x}, x \geq 0$ $\lambda = 0.006$
Session size	Truncated Lognormal	Mean = 2MB Std.Dev. = 0.722 MB Max = 5MB	$f_x = \frac{1}{\sqrt{2\pi\sigma x}} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right], x > 0$
Connection size	Truncated Lognormal	Mean = 1MB Std.Dev. = 0.35 MB	$f_x = \frac{1}{\sqrt{2\pi\sigma x}} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right], x > 0$
Inter-connection time	Truncated Lognormal	Mean = 3.27 sec Std.Dev. = 2.16 sec Min = 4s Max = 180s	$f_x = \frac{1}{\sqrt{2\pi\sigma x}} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right], x > 0$

جدول ۸: توابع توزیع و پارامترهای Email [۵۲، ۶۰]

Component	Distribution	Parameters	PDF
Session inter-arrival time	Exponential	Mean = 7.6 sec	$f_x = \lambda e^{-\lambda x}, x \geq 0$
Message size	Truncated Lognormal	Mean = 7.63 KB Std.Dev = 1.68 KB	$f_x = \frac{1}{\sqrt{2\pi}\sigma x} \exp\left[-\frac{(\ln x - \mu)^2}{2\sigma^2}\right], x > 0$

مدل سازی مبتنی بر دو نوع کدگذاری G.711 و G.729 می باشد. هر یک از این کدگذاری ها دارای اندازه ی بسته های مختلف می باشند و در مدل ارائه شده طبق [۶۱] در نظر گرفته شده اند.

۵ Email

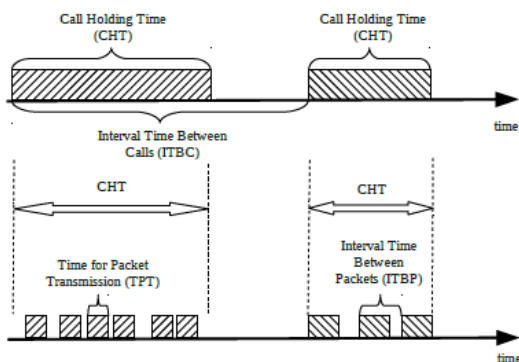
پارامترهای مدل سازی ترافیک email عبارتند از [۵۲، ۶۰]:

- زمان بین نشست
- اندازه ی پیام

۶ VoIP

در سناریوی VoIP چهار متغیر زیر را داریم [۶۱]:

۱. زمان بین Call های متوالی یا زمان نشست
۲. Call holding time
۳. زمان انتقال بسته ها: این زمان ثابت بوده و وابسته به طول بسته ها می باشد
۴. زمان بین انتقال بسته ها



شکل ۶: شکل کلی ترافیک VoIP [۶۱]

جدول ۹: توابع توزیع و پارامترهای VoIP [۶۱]

Component	Distribution	Parameters	PDF
Inter-call time	Exponential	Mean = 1.125 sec	$f_x = \lambda e^{-\lambda x}, x \geq 0$
Call holding time	Pareto type 2 (lomax)		$f_x = \frac{\alpha}{\lambda} \left[1 + \frac{x}{\lambda}\right]^{-\alpha}$ $\alpha = 2.16, \lambda = 63.43$

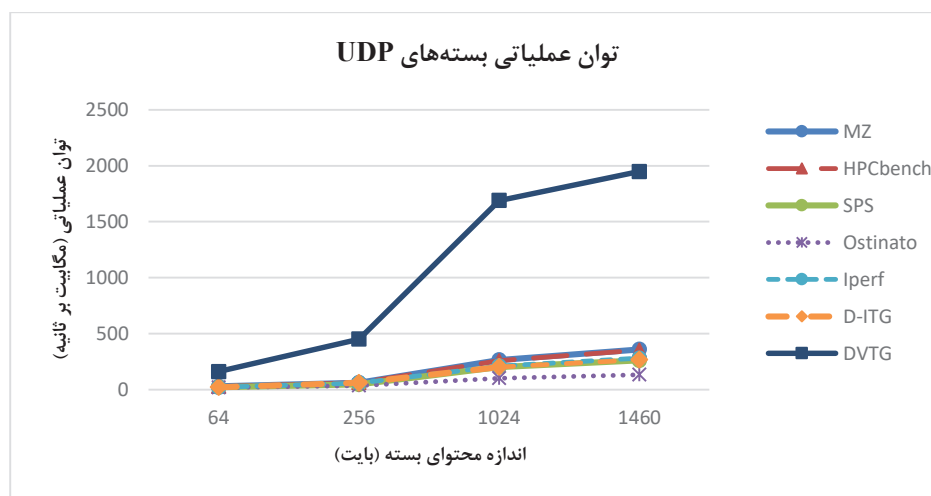
همانطور که در بخش های قبل عنوان شد، برای تولید ترافیک لایه ی انتقال از ابزار D-ITG، لایه ی دوم از MZ و برای پروتکل های Http و Https از Httpperf استفاده شده است. این ابزار با شبیه سازی رفتار کاربران و برنامه های کاربردی، ترافیک واقعی پروتکل های مربوطه را تولید می کند. مابقی پروتکل ها شامل پروتکل های پر ترافیک لایه ی کاربرد با استفاده از پارامترهای توصیف شده در بخش قبل شبیه سازی شده است. ابزار پیاده سازی شده، علاوه بر این که از قابلیت های موازی سازی خود ابزارها استفاده کرده است، با فراهم کردن بستر توزیعی قادر است از کل ظرفیت یک شبکه برای تولید ترافیک استفاده کند. همانطور که در بخش معماری پیشنهادی بیان شد، مولفه های توزیع شده در بستر شبکه به صورت متمرکز مدیریت می شوند.

پیاده سازی

به منظور ارائه یک راهکار عملی برای یافته های این پژوهش، ابزار DVTG توسعه داده شد. از جمله ویژگی های ابزار ارائه شده عبارتند از: (۱) پشتیبانی از انواع پروتکل ها در لایه های مختلف شبکه، (۲) ارائه ی گزارش در مورد معیارهایی مانند زمان پاسخ و توان عملیاتی، (۳) حفظ ویژگی های بار ترافیکی واقعی که می تواند بر روی کارایی تأثیر داشته باشد، (۴) امکان تولید ترافیک ترکیبی، (۵) معماری توزیع شده ی ابزار برای تولید حجم بالایی از ترافیک و ایجاد محیطی مشابه محیط واقعی، (۶) امکان ارسال و دریافت همزمان ترافیک توسط عامل ها. این ابزار بر روی سیستم Ubuntu 12.04 TLS با دو پردازنده با سرعت 3.33GHz و حافظه ی ۲ گیگابایت و با زبان برنامه نویسی جاوا پیاده سازی شده است.

جدول ۱۰. مقایسه ی ویژگی های ابزار پیشنهادی و ابزارهای موجود

DVTG	D-ITG	Iperf	Ostinato	SPS	HPCbench	MZ	
✓	✓	✓	✓	✓	✓	✓	TCP
✓	✓	✓	✓	✓	✓	✓	UDP
✓	✓	-	✓	✓	-	✓	ICMP
✓	-	-	✓	✓	-	✓	پروتکل های لایه ۲
✓	-	-	-	-	-	-	Http/Https
✓	-	-	-	-	-	-	Web
✓	-	-	-	-	-	-	Email
✓	✓	-	✓	-	-	-	VOIP
✓	-	-	-	-	-	-	FTP
✓	-	-	-	-	-	-	P2P
✓	-	-	✓	-	-	-	Streaming
✓	✓	-	-	-	-	-	مقیاس پذیری
✓	✓	-	-	-	-	-	توزیع پذیری
✓	-	-	-	-	-	-	حفظ ویژگی های بار ترافیکی لایه کاربر
✓	-	-	-	-	-	-	تولید همزمان ترافیک از چند پروتکل
✓	-	-	-	-	-	✓	امکان تولید فریم های نامعتبر
-	-	-	-	-	✓	-	قابلیت نمایش منابع مصرفی
<ul style="list-style-type: none"> • تأخیر • تأخیر یکطرفه • نرخ از دست • دادن بسته • RTT • اعوجاج • توان عملیاتی 	<ul style="list-style-type: none"> • تأخیر • تأخیر یکطرفه • نرخ از دست • دادن بسته • RTT • اعوجاج • توان عملیاتی 	<ul style="list-style-type: none"> • نرخ از دست • دادن بسته ها • اعوجاج • توان عملیاتی 	-	-	<ul style="list-style-type: none"> • میزان تأخیر • توان عملیاتی • بین دو طرف • RTT 	<ul style="list-style-type: none"> • نرخ از دست • دادن بسته • بر هم خوردن • ترتیب بسته • اعوجاج 	پارامترهای گزارش شده



نمودار ۵: مقایسه ی توان عملیاتی تولید ترافیک UDP ابزار پیشنهادی (DVTG) با ابزارهای موجود

ارزیابی

در تحقیقات آتی، تلاش ما بر این است که کاستی‌ها و محدودیت‌های ابزار DVTG را هموار کنیم. از جمله این محدودیت‌ها می‌توان به عدم امکان تولید ترافیک لایه ۲ در محیط غیر از لینوکس - به دلیل استفاده از ابزار MZ که فقط در محیط لینوکس قابل اجرا است - اشاره کرد. برخی چالش‌ها در اینگونه تحقیقات مطرح هستند که عمومی بوده و مختص این پژوهش نیست. از جمله سرشار ناشی از راه‌اندازی پیاده‌سازی محیط توزیع‌شده و تفاوت محتوای ترافیک شبیه‌سازی شده با ترافیک واقعی. یکی دیگر از کارهایی که می‌تواند به کاربردی‌تر شدن ابزار کمک کند، تولید ترافیک مربوط به پروتکل‌های پرکاربرد دیگر مانند ترافیک بازی‌های اینترنتی است.

یکی از ویژگی‌های اصلی ابزار DVTG امکان تولید حجم بالایی از ترافیک از طریق ارسال موازی ترافیک است. برای ارزیابی آن شبکه‌ای شامل ۱۰ گره در نظر گرفته شد. این ارزیابی‌ها با روشی مشابه روش ارزیابی ابزارهای موجود، که در بخش انتخاب راهکار مناسب توضیح داده شد، انجام شد. ارزیابی‌ها نشان داد توان عملیاتی ابزار پیشنهادی تقریباً ۹ برابر توان عملیاتی ابزارهای موجود می‌باشد. همچنین این امکان وجود دارد که حجم بسیار بالای ترافیک، از انواع ترافیک مشابه ترافیک اینترنت شبیه‌سازی گردد. نمودار ۵ نتایج این ارزیابی را نشان می‌دهد.

نتیجه‌گیری

در این مقاله به بررسی اهمیت تولید ترافیک واقعی برای انواع آزمون‌های شبکه پرداخته شد. تولیدکننده‌های ترافیک در حوزه‌های مختلف شبکه از جمله آزمون کارایی شبکه و دستگاه‌های شبکه (شامل Middleboxها و تحلیل شبکه‌های ناهمگن از نظر توان عملیاتی، نرخ از دست دادن بسته، تأخیر، اعوجاج)، آزمون امنیت (به عنوان مثال دیوار آتش^۴IDS، بارکاری پیش‌زمینه و بدخواه)، آزمون کیفیت سرویس، آزمون پروتکل‌های جدید (مثلاً در سطح لایه‌ی کاربرد)، اندازه‌گیری پهنای باند در دسترس، آزمون ظرفیت دستگاه‌ها و بررسی الگوریتم‌های جدید، مورد مطالعه و بررسی قرار گرفته‌اند. تاکنون تعداد زیادی از این ابزارها ارائه شده‌اند اما هیچ یک از آن‌ها تمامی ویژگی‌های لازم را ندارند. به خصوص ابزارهایی که ترافیک لایه‌ی کاربرد تولید می‌کنند بسیار محدود بوده و اکثر آنها در دسترس نبوده و نیز تعداد کمی از پروتکل‌ها را پشتیبانی می‌کنند.

ابزار پیشنهادی در این مقاله، DVTG، با استفاده از تکنیک‌های آماری موجود، مجموعه‌ای را فراهم کرده است که قادر است انواع ترافیک‌ها را در لایه‌های مختلف و برای پروتکل‌های معروف تولید کند. ترافیک تولیدی رفتار واقعی کاربران اینترنت را شبیه‌سازی خواهد کرد. در مقایسه با ابزارهای رایگان موجود این ابزار از تنوع پروتکلی بیشتری در لایه‌های مختلف برخوردار بوده و با ارائه‌ی معماری توزیع‌شده می‌تواند به طور همزمان ترکیبی از انواع ترافیک را تولید و ارسال کند. همچنین استفاده از ابزارهای خوب موجود و اضافه کردن ویژگی‌های مثبت آن‌ها به این ابزار به راحتی امکان‌پذیر می‌باشد. خصیصه‌ی توزیع‌پذیری این ابزار این امکان را فراهم می‌کند که بتوان حجم زیادی از ترافیک شبکه را تولید کرد. افزایش حجم ترافیک تقریباً با افزایش تعداد مولفه‌های تولید ترافیک در شبکه نسبت خطی دارد.

منابع

- [1] A. Botta, A. Dainotti, and A. Pescapé, "Do you trust your software-based traffic generator?," IEEE Communications Magazine, vol. 48, pp. 158-165, 2010.
- [2] V. Jacobson, C. Leres, and S. McCanne, "tcpdump, 1989," ed.
- [3] G. Aceto, A. Botta, W. De Donato, P. Marchetta, A. Pescapé, and G. Ventre, "Open Source Platforms for Internet Monitoring and Measurement," in Signal Image Technology and Internet Based Systems (SITIS), 2012 Eighth International Conference on, 2012, pp. 563-570.
- [4] C. Mueller, M. Horie, and S. W. Neville, "A distributed application-level IT system workload generator," in Advanced Information Networking and Applications Workshops, 2009. WAINA'09. International Conference on, 2009, pp. 1100-1105.
- [5] LANTraffic V2. Available: http://www.omnicor.com/Xtramus_packet_generators_LAN_STD.aspx
- [6] Cat Karat packet builder Available: <http://packetbuilder.net>
- [7] IxNetwork Available: <https://www.ixiacom.com/products/ixnetwork>
- [8] NuStreams Available: http://www.omnicor.com/Ethernet_switch_tester_NuStreams_P9M.aspx
- [9] Spirent-Avalanche Available: <https://www.spirent.com/Products/Avalanche>
- [10] BreakingPoint Elite Available: <https://www.ixiacom.com/products/breakingpoint>
- [11] LANForge Available: <http://www.candelatech.com/>
- [12] IP Traffic - Test & Measure Available: <http://www.zti-communications.com/iptraffic/>
- [13] "Ostinato".

- [31] Iperf. Available: <http://dast.nlanr.net/Projects/Iperf/>
- [32] MGEN. Available: <http://cs.itd.nrl.navy.mil/work/mgen/index.php>
- [33] RUDE. Available: <http://rude.sourceforge.net/>
- [34] A. Waterland. (July 2012). Stess. Available: <https://people.seas.harvard.edu/~apw/stress/>
- [35] H. Englund, "Evaluation of traffic generation tools and implementation of test system," Master Thesis, Umeå University, 2011.
- [36] B. Huang, M. Bauer, and M. Katchabaw, "Hpcbench-a Linux-based network benchmark for high performance networks," in 19th International Symposium on High performance Computing Systems and applications (HPCS ,2005 ,(05'pp. 65-71.
- [37] B. Huang, "Network Performance Studies in High Performance Computing Environments," Master Thesis, The University of Western Ontario, 2004.
- [38] S. Zander, D. Kennedy, and G. Armitage, "KUTE-A high performance kernel-based UDP traffic engine," CAIA (Center for Advanced Internet Architectures) Technical Report, 2005.
- [39] R. Olsson, "Pktgen the linux packet generator," in Proceedings of the Linux Symposium, Ottawa, Canada, 2005, pp. 11-24.
- [40] A. Dainotti, A. Botta, and A. Pescapè, "A tool for the generation of realistic network workload for emerging networking scenarios," Computer Networks vol. 56, pp. 3531-3547, 2012.
- [41] M. Suznjevic, I. Stupar, and M. Matijasevic, "Traffic modeling of player action categories in a MMORPG ",in Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques, 2011, pp. 280-287.
- [42] K. V. Vishwanath and A. Vahdat, "Swing: Realistic and responsive network traffic generation," IEEE/ACM Transactions on Networking, vol. 17 , pp. 712-725, 2009.
- [43] P. E. Heegaard, "GenSyn-a Java based generator of synthetic Internet traffic linking user behaviour models to real network protocols," in ITC Specialist Seminar on IP Traffic Measurement, Modeling and Management, 2000.
- [44] K. V .Katsaros, G. Xylomenos, and G. C. Polyzos, "GlobeTraff: a traffic workload generator for the performance evaluation of future Internet architectures," in 2012 5th International Conference on New Technologies, Mobility and Security (NTMS), 2012, pp. 1-5.
- [45] NetSpec. Available: <http://www.netspec.com/>
- [46] E. L. Helvey, "Trafgen: An Efficient Approach to Statistically Accurate Artificial Network Traffic Generation," Ohio University, 1998.
- [14] N. Bonelli, A. Di Pietro, S. Giordano, and G. Prociassi, "Flexible High Performance Traffic Generation on Commodity Multi-core Platforms," in Traffic Monitoring and Analysis, ed: Springer, 2012, pp. 157-170.
- [15] Wireshark. Available: <https://www.wireshark.org/>
- [16] Mausezahn Available: <http://www.perihel.at/sec/mz/>
- [17] N. Muruganadham, R. Umesh, and D. Kaleeswaran, "A Novel Approach for Preventing Selective Jamming Alerts".
- [18] M. Fernandez, "Evaluating OpenFlow Controller Paradigms," in ICN 2013, The Twelfth International Conference on Networks, 2013, pp. 151-157.
- [19] A. R. Khakpour, J. W. Hulst, Z. Ge, A. X. Liu, D. Pei, and J. Wang, "Denial of Firewalling".
- [20] A. R. Khakpour, J. W. Hulst, Z. Ge, A. X. Liu, D. Pei, and J. Wang, "Firewall fingerprinting," in INFOCOM, 2012 Proceedings IEEE, 2012, pp. 1728-1736.
- [21] J. Sommers, H. Kim, and P. Barford, "Harpoon: a flow-level traffic generator for router and network tests," in ACM SIGMETRICS Performance Evaluation Review, 2004, pp. 392-392.
- [22] TCPReplay. Available: <http://tcpreplay.synfin.net>
- [23] S. Avallone, S. Guadagno, D. Emma, A. Pescapè, and G. Ventre, "D-ITG distributed internet traffic generator," in Quantitative Evaluation of Systems, 2004. QEST 2004. Proceedings. First International Conference on the, 2004, pp. 316-317.
- [24] S. S. Kolahi, S. Narayan, D. D. Nguyen, and Y. Sunarto, "Performance monitoring of various network traffic generators," in Computer Modelling and Simulation (UKSim), 2011 UkSim 13th International Conference on, 2011, pp. 501-506.
- [25] Netperf. Available: <http://www.netperf.org/netperf/>
- [26] A. Pescapè, M. D'Arienzo, S. Romano, M. Esposito, S. Avallone, and G. Ventre, "'Mtools"-IEEE Network, Software Tools for Networking 2002, Vol. 16 No. 5 pag. 3," ed: ISSN.
- [27] Udpgenerator. Available: <http://www.citi.umich.edu/projects/qbone/generator.html>
- [28] S. Avallone, D. Emma, A. Pescape, and G. Ventre, "High Performance Internet Traffic Generators," Journal of Supercomputing, vol. 35, pp. 5-26, 2006.
- [29] S. Avallone, D. Emma, A. Pescapè, and G. Ventre, "A distributed multiplatform architecture for traffic generation," in International Symposium on Performance Evaluation of Computer and Telecommunication Systems, San Jose, CA, USA, 2004.
- [30] G. Z. Papadopoulos, "Experimental Assessment of Traffic Generators," Master thesis, 2012.

- [56] J. Doggen and F. Van der Schueren, "Design and Simulation of a H. 264 AVC Video Streaming Model," Department of Applied Engineering, University College of Antwerp, Antwerp, 2008.
- [57] S. Tanwir and H. Perros, "A Survey of VBR Video Traffic Models," 2013.
- [58] D. Liu, E. I. Sára, and W. Sun, "Nested autoregressive processes for MPEG-encoded video traffic modeling," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, pp. 169-183, 2001.
- [59] J. Ishac, "FTP traffic generator," Citeseer2001.
- [60] D. Sivchenko, "Internet Traffic Performance in High Speed Trains1," 2004.
- [61] C. I. de Mattos, E. P. Ribeiro, and C. M. Pedrosa, "A new model for VoIP traffic generation," in *International telecommunication symposium. Brazilian Telecommunication Society, Manaus, Brazil*, 2010.
- [47] Simple Packet Sender (SPS). Available: <https://sites.google.com/site/simplepacketsender/>
- [48] D. Mosberger and T. Jin, "httpperf—a tool for measuring web server performance," *ACM SIGMETRICS Performance Evaluation Review*, vol. 26, pp. 31-37, 1998.
- [49] D. Cotroneo, D. Di Leo, F. Fucci, and R. Natella, "SABRINE: State-Based Robustness Testing of Operating Systems," 2013.
- [50] R. Hou, T. Jiang, L. Zhang, P. Qi, J. Dong, H. Wang, et al., "Cost effective data center servers," in *HPCA*, 2013, pp. 179-187.
- [51] A. N. Pais Cereghetti, "Global evaluation of CDNs performance using PlanetLab," 2013.
- [52] "cdma2000 Evaluation Methodology (Revision 0)," 3GPP22004.
- [53] V. Paxson and S. Floyd, "Wide area traffic: the failure of Poisson modeling," *IEEE/ACM Transactions on Networking (ToN)*, vol. 3, pp. 226-244, 1995.
- [54] IPOQUE, "Internet Study 2008/2009," Research Report2009.
- [55] D. Erman, D. Ilie, and A. Popescu, "BitTorrent traffic characteristics," in *Computing in the Global Information Technology, 2006. ICCGI'06. International Multi-Conference on*, 2006, pp. 42-42.

