



۸۱۰۱۰۱۴۹۰

۸۱۰۱۰۱۵۲۶

شبکه‌های کامپیوتری
کسری کاشانی، مرضیه موسوی



پروژه ۲ فاز اول

۱ مقدمه

در این پروژه، به مقایسه عملکرد پروتکل‌های TCP Reno، TCP Tahoe و TCP Vegas در NS-2 بر اساس سه معیار RTT، Throughput و Packet Loss پرداختیم.

۲ هدف و صورت مسئله

هدف اصلی این پروژه، پیاده‌سازی و شبیه‌سازی سه گونه‌ی TCP شامل Reno، Tahoe و Vegas و مقایسه‌ی رفتار آن‌ها روی یک توپولوژی مشخص است. طبق صورت پروژه، این مقایسه باید بر اساس سه متريک انجام شود: RTT، Throughput و نرخ از دست‌رفتن بسته‌ها یعنی Packet Loss، و خروجی نهایی نيز شامل سه نمودار (هر کدام با ۶ خط: ۲ جریان برای هر ۳ نوع TCP) خواهد بود.

۳ محیط اجرا و ابزار

شبیه‌سازی در Ubuntu و با NS-2.35 انجام شد. مدل شبیه‌سازی NS-2 را event-driven (رویداد-محور) می‌دانیم. زمان بر اساس صف رویدادها پیش می‌رود. اسکریپت توپولوژی و سناریو با OTcl نوشته شد و استخراج داده‌ها و رسم نمودارها با ابزارهای python، gnuplot، awk، bash، awk (برای نمودار میله‌ای) انجام شد.

۴ شرایط و الزامات دقیق صورت پروژه

در این بخش، دقیقاً مواردی که در صورت پروژه الزام شده‌اند لیست می‌شوند و در ادامه نشان داده می‌شود که چگونه رعایت شده‌اند:

۱.۴ توپولوژی و جریان‌ها

- وجود دو جریان دائمی: یکی از گره ۱ به ۵ و دیگری از گره ۲ به ۶، با ارسال مداوم داده (همیشه داده برای ارسال وجود دارد).
- تأخیر لینک‌های دارای delay variable باید به صورت تصادفی بین ۵ تا ۲۵ میلی‌ثانیه در نظر گرفته شود.
- اندازه صفت در روتراها برابر ۱۰ بسته باشد.
- مقدار TTL برابر ۶۴ تنظیم شود.
- اندازه بسته TCP برابر پیش‌فرض NS2 یعنی ۱۰۰۰ بایت باقی بماند.
- استفاده از IPv4 برای لایه شبکه.

۲.۴ تنظیمات شبیه‌سازی

- شبیه‌سازی باید ۱۰ مرتبه اجرا شود و هر بار مدت زمان شبیه‌سازی ۱۰۰۰ ثانیه باشد؛ سپس میانگین داده‌ها گزارش شود.
- فرض شود جریان‌ها دائما در حال ارسال هستند و از دست‌رفتن بسته روی لینک رخ نمی‌دهد (از دست‌رفتن تنها به دلیل محدود بودن بافر روتراها است).
- ظرفیت یا پهنه‌ای باند لینک‌ها ثابت باقی بماند.
- برای پارامترهایی مانند initial window و max window از مقادیر پیش‌فرض NS2 استفاده شود (یا اگر چیزی پیش‌فرض نبود، پیش‌فرض انتخاب‌شده توضیح داده شود).

۵ ساختار پروژه و فایل‌ها

این پروژه به صورت ماژولار پیاده‌سازی شده و فایل‌ها نقش‌های مشخصی دارند:

۱.۵ اسکریپت اصلی شبیه‌سازی

- tcl/ca_tcp.tcl: تعریف توپولوژی، لینک‌ها، صفت، TTL، دو جریان، فعال‌سازی گونه‌ی TCP (یکی از Tahoe/Reno/Vegas)، تولید فایل‌های trace لازم.

۲.۵ استخراج داده از خروجی trace

- استخراج گذردهی بر حسب زمان (معمولاً مبتنی بر بایت یا بسته‌ی دریافت شده در مقصد و تبدیل به Mbps در بازه‌های زمانی).
- استخراج RTT بر حسب زمان از فایل‌های مرتبط با نمونه‌های RTT.
- استخراج تعداد Drop بر ثانیه (یا نرخ از دست رفتن بر حسب زمان) از روی .drop.
- میانگین‌گیری ۱۰ اجرای مستقل برای تولید فایل‌های avg.

۳.۵ اجرای خودکار و ساخت داده‌های نهایی

- اجرای ۱۰ مرتبه شبیه‌سازی برای هر نوع TCP و تولید فایل‌های trace و خروجی‌های میانی.
- فراخوانی scripts/build_data.sh برای تولید فایل‌های داده در مسیر /data و سپس میانگین‌گیری ۱۰ اجرای مستقل.

۴.۵ رسم نمودارها

- فایل‌های gnuplot برای نمودارهای اصلی ۶-خطی:
 - plots/plot_throughput.gp
 - plots/plot_rtt.gp
 - plots/plot_loss.gp
- نسخه‌های zoom از نمودارهای اصلی (۰ تا ۵۰ ثانیه):
 - plots/plot_throughput_zoom50.gp
 - plots/plot_rtt_zoom50.gp
 - plots/plot_loss_zoom50.gp
- تولید ۹ نمودار تفکیک شده (هر TCP جداگانه برای هر متريک) و ۹ نمودار zoom برای آنها:

scripts/plot_split_all.sh □

scripts/plot_split_zoom50.sh □

plots/plot_one_*.gp □

plots/plot_one_*_zoom50.gp □

۵.۵ نمودار میله‌ای نرخ اتلاف متوسط

- تولید فایل داده‌ی خلاصه‌ی نرخ اتلاف متوسط هر TCP برای دو scripts/make_loss_bar_data.sh جریان.

- رسم نمودار میله‌ای Average Packet Loss Rate (پس از نصب scripts/plot_loss_bar.py .(matplotlib

نحوه‌ی جایگزین با plots/plot_loss_bar.gp •

۶ روش انجام کار

این بخش، مسیر رسیدن به خروجی‌ها را مرحله به مرحله توضیح می‌دهد:

۱.۶ مرحله ۱: آماده‌سازی محیط

- اطمینان از در دسترس بودن فرمان ns در PATH یا اجرای مستقیم باینتری NS2
- آماده‌سازی پوشش‌های پروژه: plots/, data/, trace/, scripts/, tcl/

۲.۶ مرحله ۲: اجرای شبیه‌سازی ۱۰ مرتبه برای هر نوع TCP

- طبق الزام صورت پروژه، برای هر یک از Tahoe/Reno/Vegas شبیه‌سازی ۱۰ بار و هر بار به مدت ۱۰۰۰ ثانیه اجرا می‌شود. این کار با scripts/run_all.sh انجام شد و فایل‌های trace به شکل کلی زیر تولید شدند: trace/Tahoe_run10.tr ... trace/Tahoe_run1.tr (و مشابه برای Reno و Vegas)، همچنین فایل‌های trace مرتبط با هر جریان.

۳.۶ مرحله ۳: استخراج متريک‌ها و ميانگين‌گيري ۱۰ اجرا

در اين مرحله از فایل‌های trace، سه متريک استخراج و به بازه‌های زمانی (ثانيه‌ای) تبدیل شد، سپس با avg10.awk در اين مرحله از فایل‌های trace، سه متريک استخراج و به بازه‌های زمانی (ثانيه‌ای) تبدیل شد، سپس با avg10.awk ميانگين ۱۰ اجرای مستقل محاسبه و در data/*_avg.dat ذخیره شد.

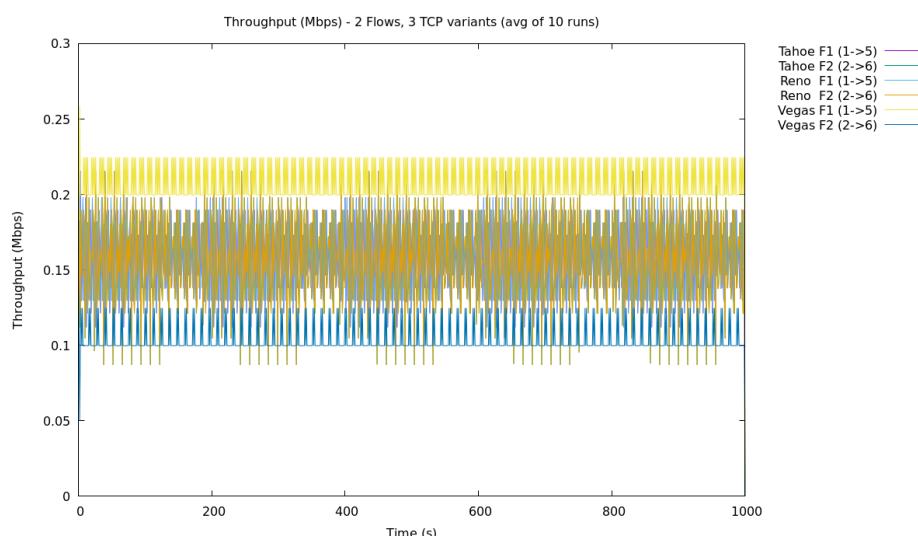
۴.۶ مرحله ۴: تولید نمودارهای موردنیاز

طبق صورت پروژه، سه نمودار اصلی با ۶ خط تولید شد (۲ جریان برای هر ۳ پروتکل). سپس برای خوانایی بیشتر، ۹ نمودار تفکیک شده (هر TCP جداگانه) و نسخه‌های zoom ۰ تا ۵۰ ثانیه برای هر کدام ساخته شد.

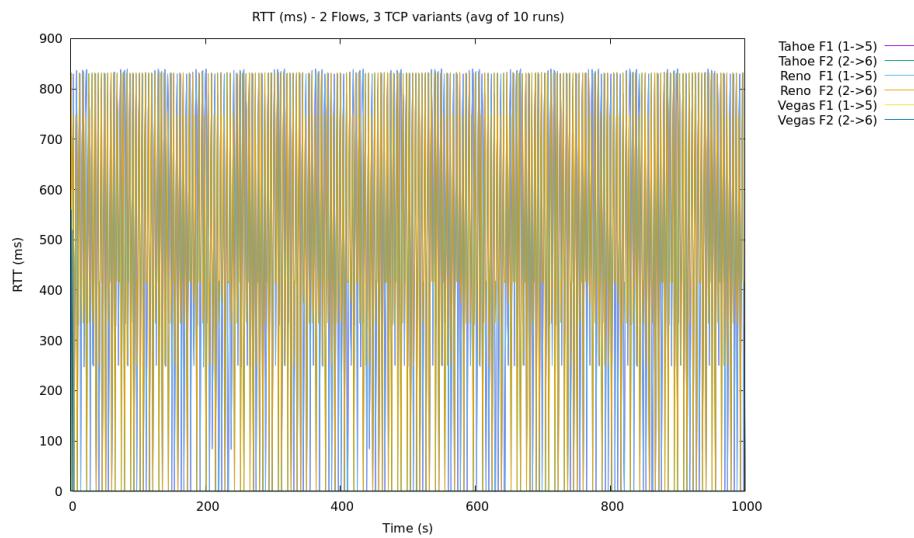
۷ نتایج و نمودارها

در اين بخش، عکس تمام نمودارهای خروجی قرار دارند:

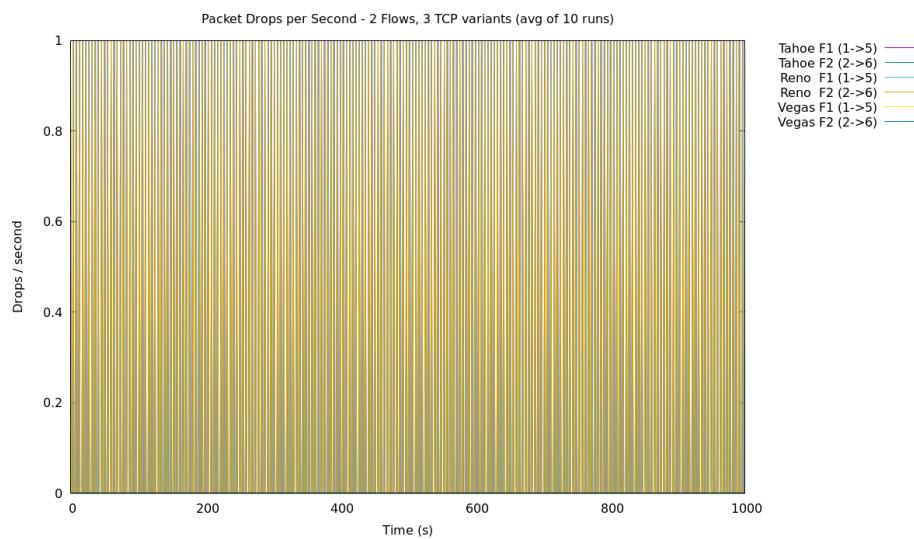
۱.۷ نمودارهای اصلی ۶-خطی (كل بازه)



شکل ۱: نمودار throughput ۶-خطی، ۰ تا ۱۰۰۰ ثانیه

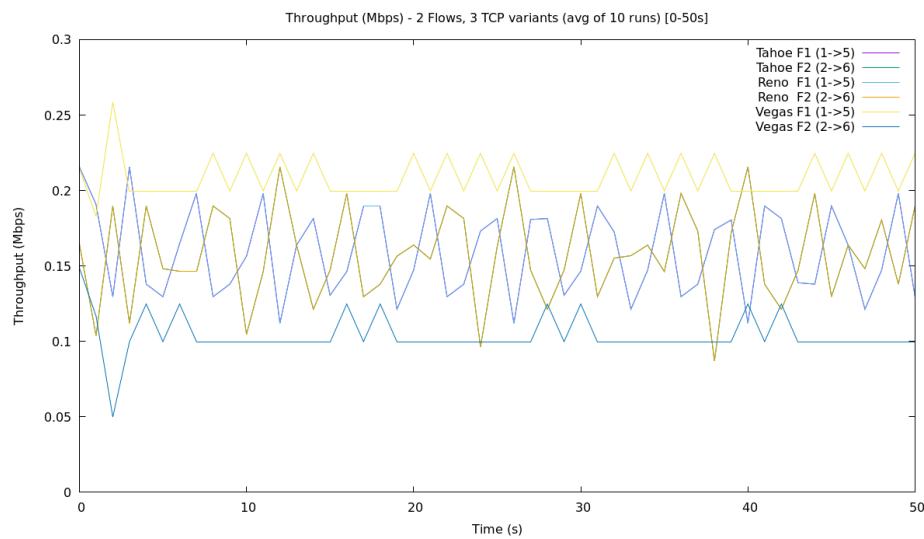


شکل ۲: نمودار RTT ۶ خطی، ۰ تا ۱۰۰۰ ثانیه

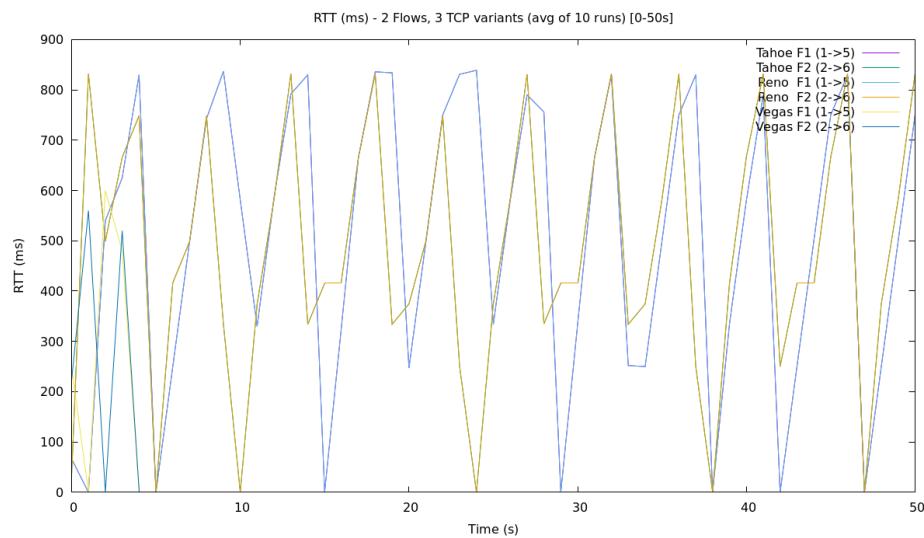


شکل ۳: نمودار loss ۶ خطی، ۰ تا ۱۰۰۰ ثانیه

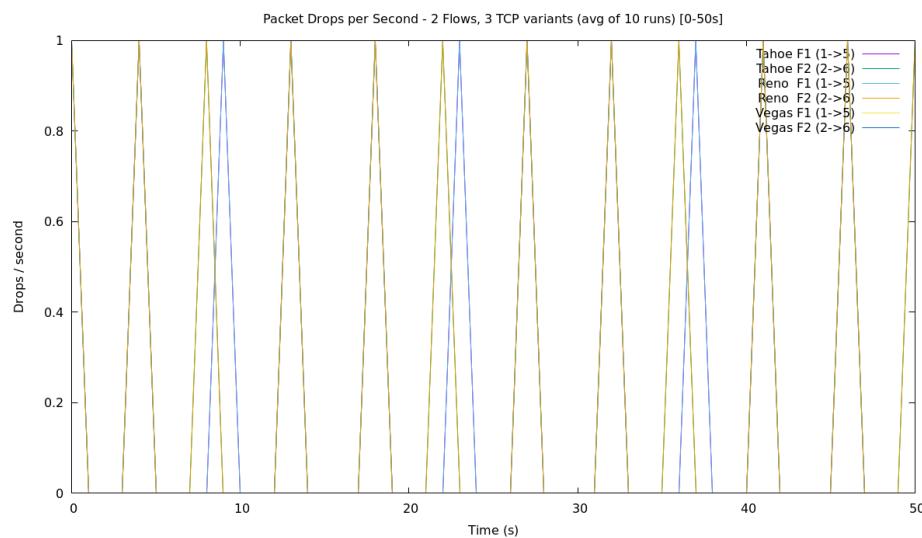
۲.۷ نمودارهای اصلی ۶-خطی زومشده (۰ تا ۵۰ ثانیه)



شکل ۴: نمودار throughput ۶-خطی زومشده، ۰ تا ۵۰ ثانیه

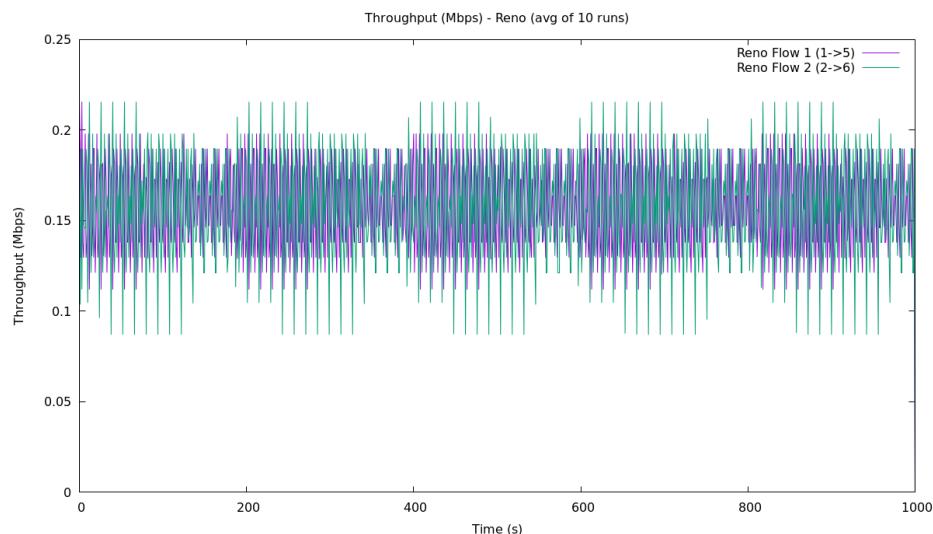


شکل ۵: نمودار RTT ۶-خطی زومشده، ۰ تا ۵۰ ثانیه

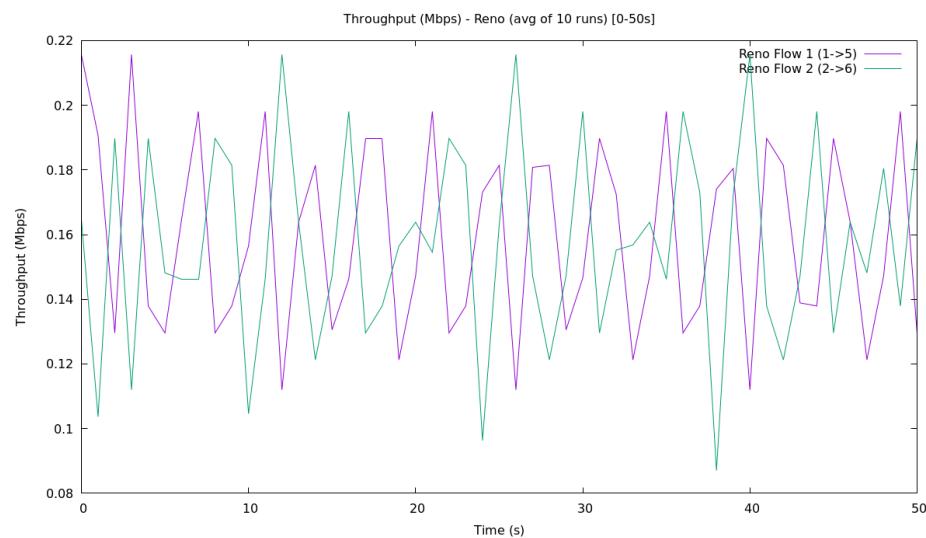


شکل ۶: نمودار loss-خطی زومشده، ۰ تا ۵۰ ثانیه

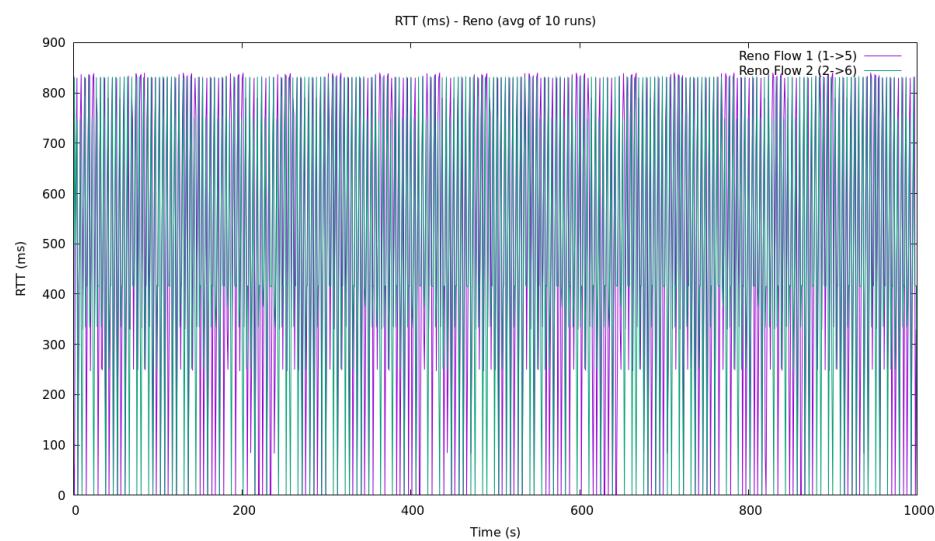
۳.۷ نمودارهای تفکیکی برای Reno (کل و زوم)



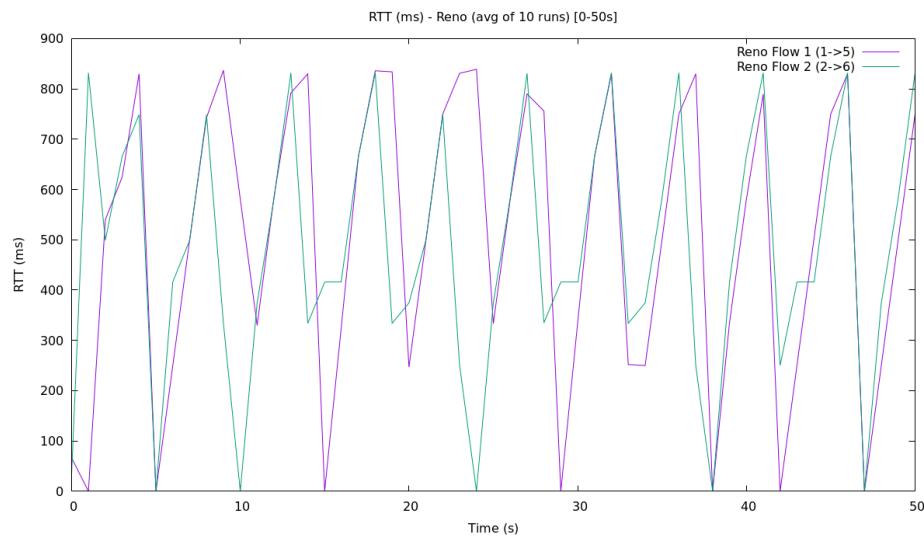
شکل ۷: نمودار throughput برای Reno، ۰ تا ۱۰۰۰ ثانیه



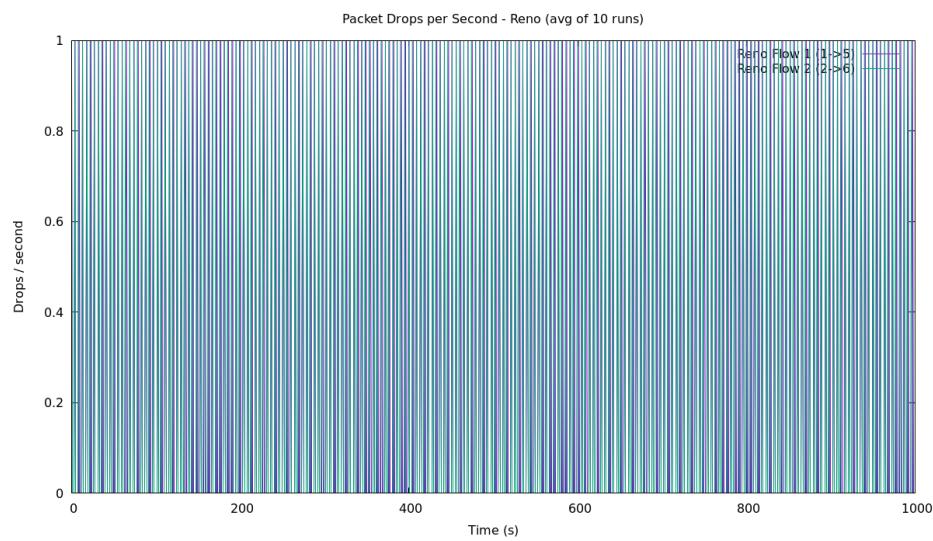
شکل ۸: نمودار throughput برای Reno زومشده، ۰ تا ۵۰ ثانیه



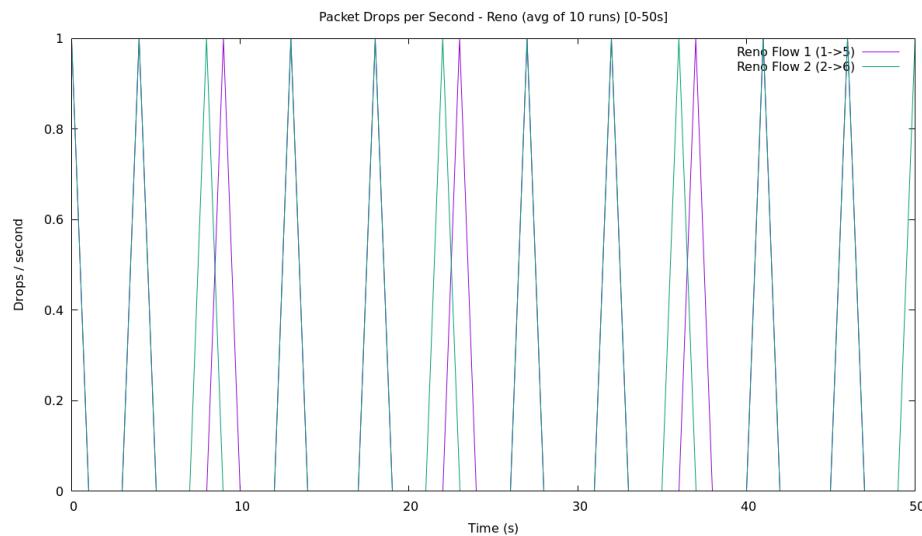
شکل ۹: نمودار RTT برای Reno، ۰ تا ۱۰۰۰ ثانیه



شکل ۱۰: نمودار RTT برای Reno زوم شده، ۰ تا ۵۰ ثانیه

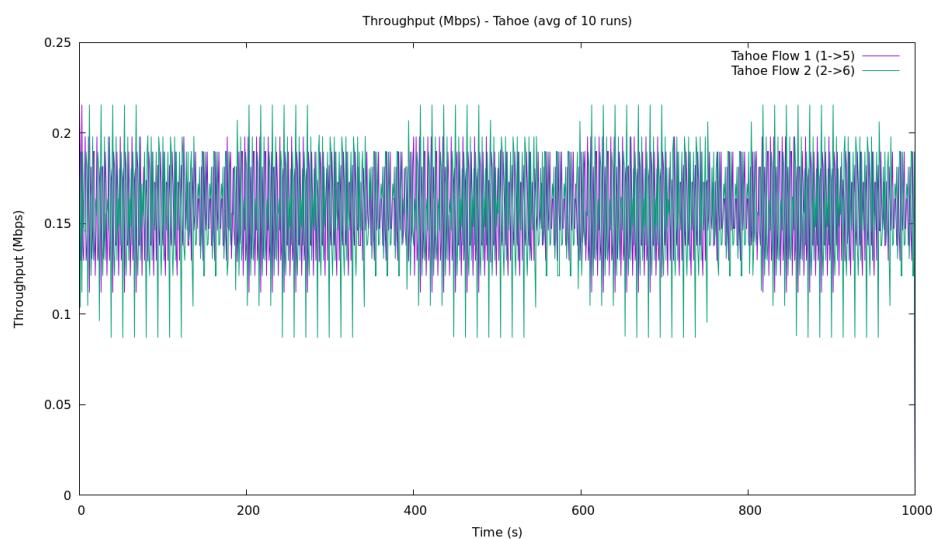


شکل ۱۱: نمودار loss برای Reno، ۰ تا ۱۰۰۰ ثانیه

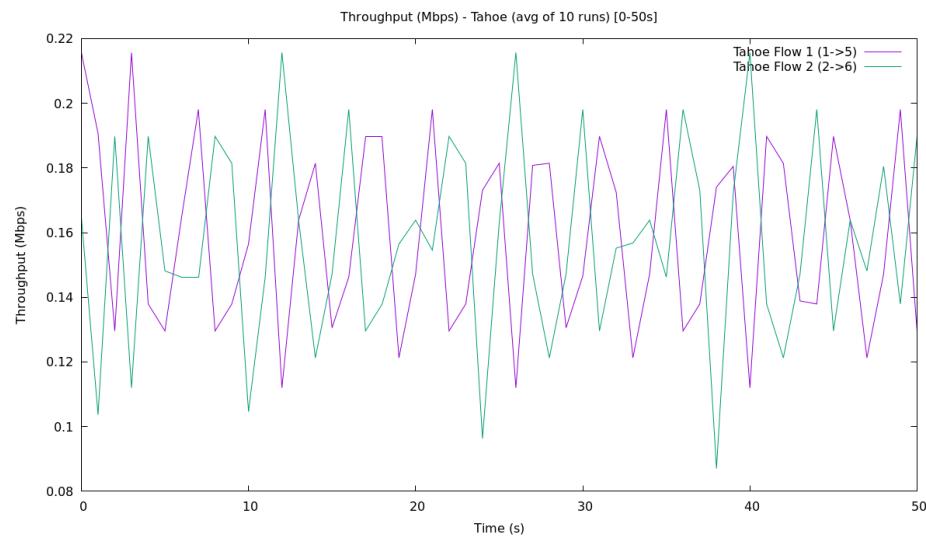


شکل ۱۲: نمودار loss برای Reno زومشده، ۰ تا ۵۰ ثانیه

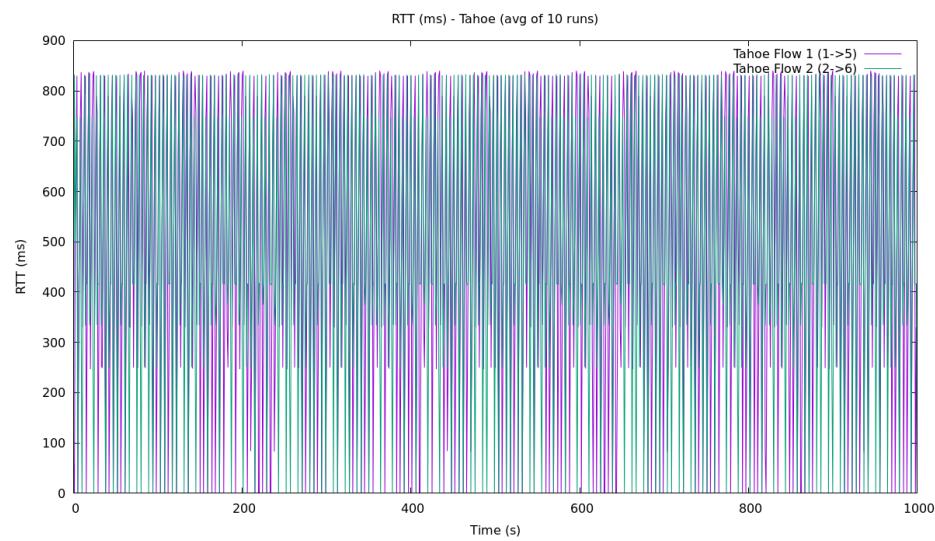
۴.۷ نمودارهای تفکیکی برای Tahoe (کل و زوم)



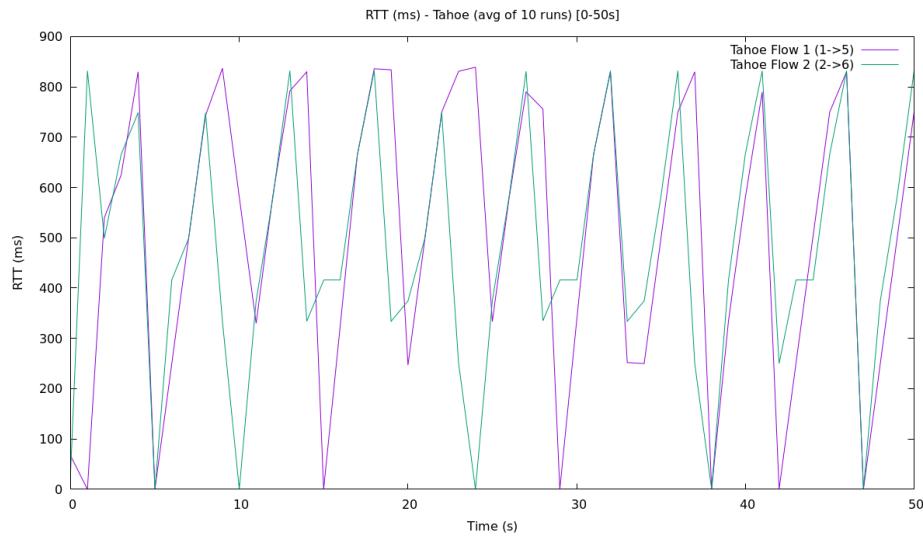
شکل ۱۳: نمودار throughput برای Tahoe، ۰ تا ۱۰۰۰ ثانیه



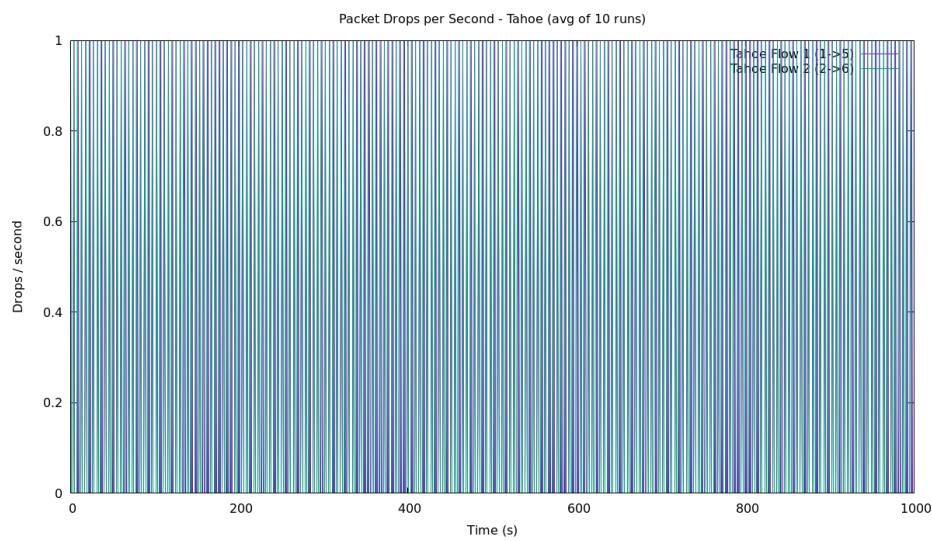
شکل ۱۴: نمودار throughput برای Tahoe زومشده، ۰ تا ۵۰ ثانیه



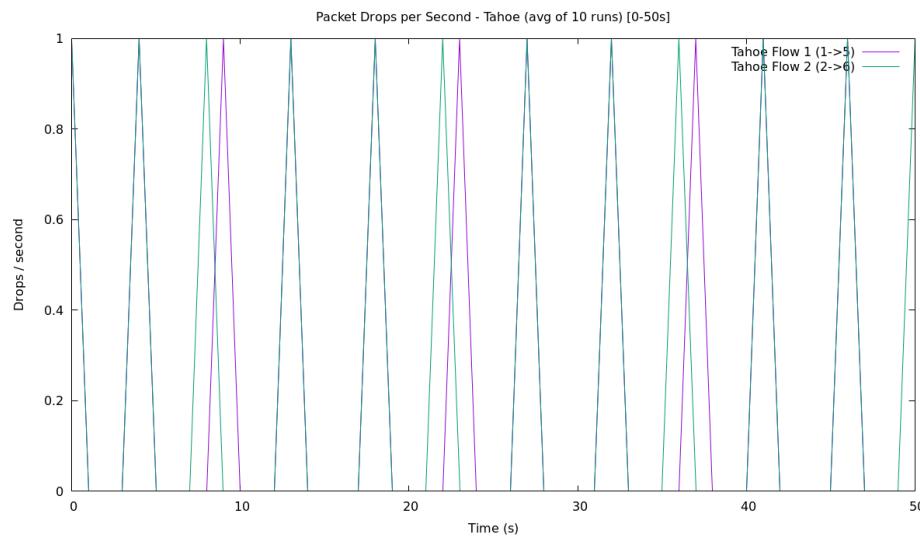
شکل ۱۵: نمودار RTT برای Tahoe، ۰ تا ۱۰۰۰ ثانیه



شکل ۱۶: نمودار RTT برای Tahoe زومشده، ۰ تا ۵۰ ثانیه

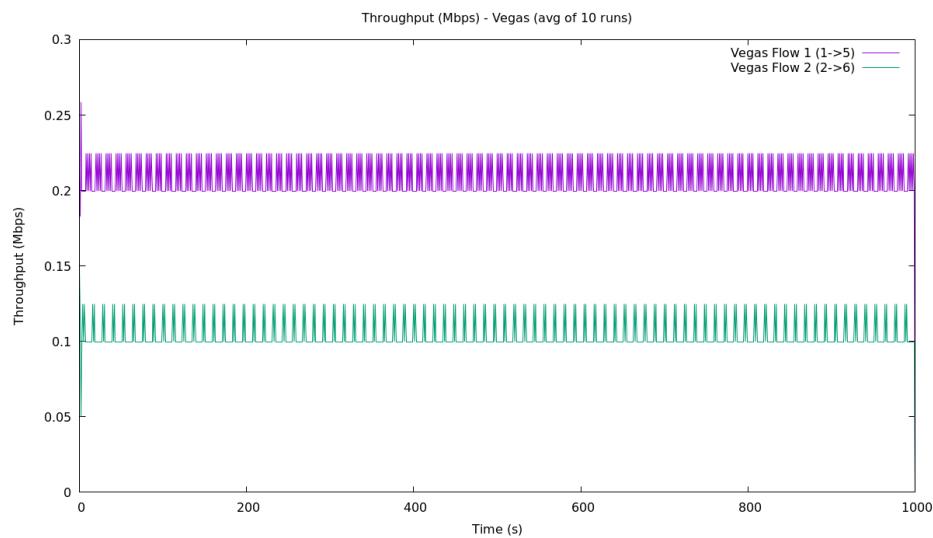


شکل ۱۷: نمودار loss برای Tahoe، ۰ تا ۱۰۰۰ ثانیه

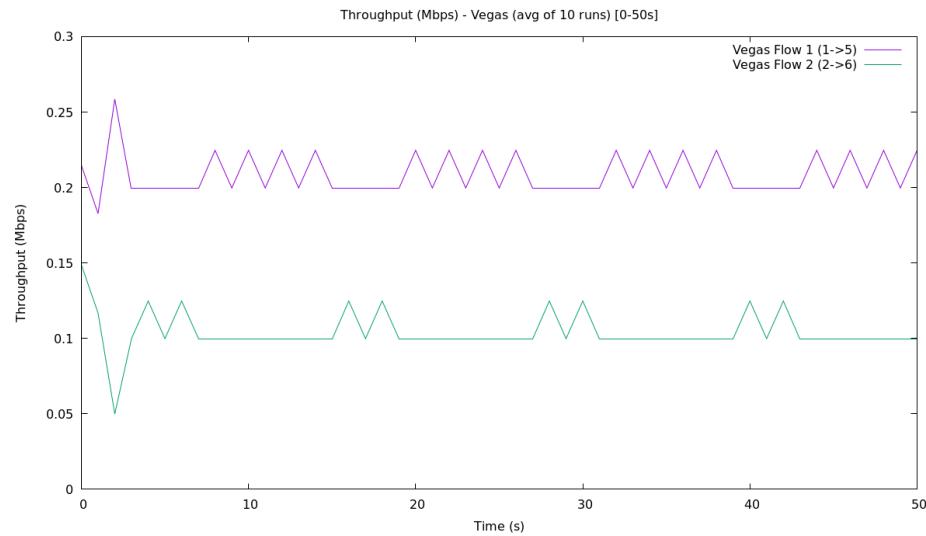


شکل ۱۸: نمودار loss برای Tahoe زومشده، تا ۵۰ ثانیه

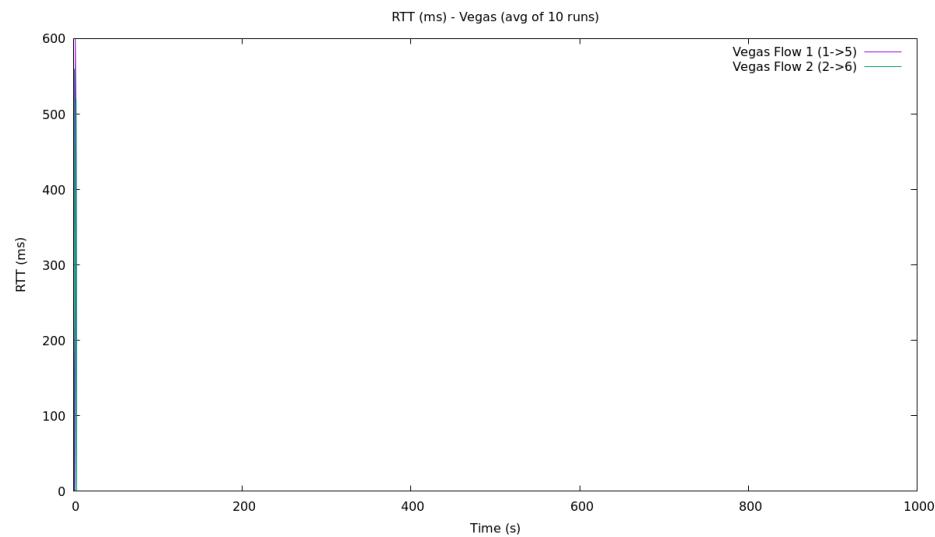
۵.۷ نمودارهای تفکیکی برای Vegas (کل و زوم)



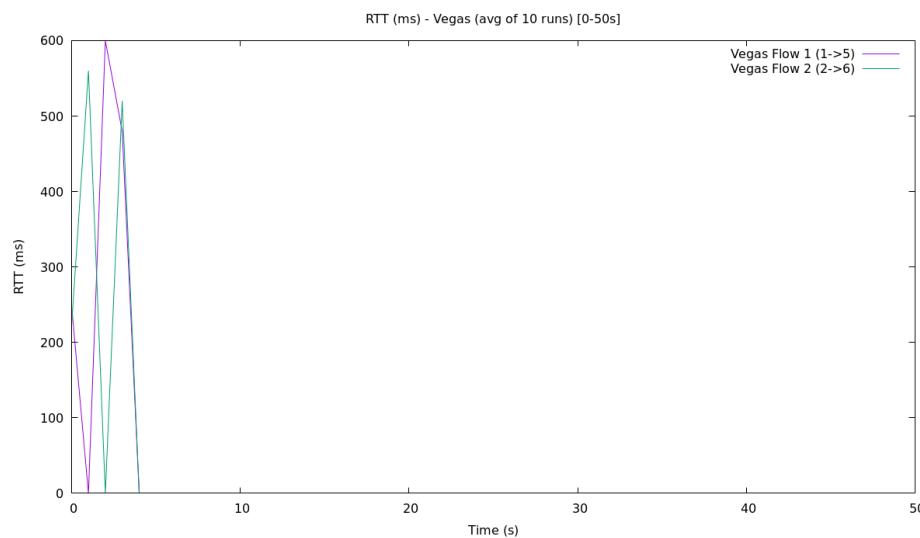
شکل ۱۹: نمودار throughput برای Vegas، تا ۱۰۰۰ ثانیه



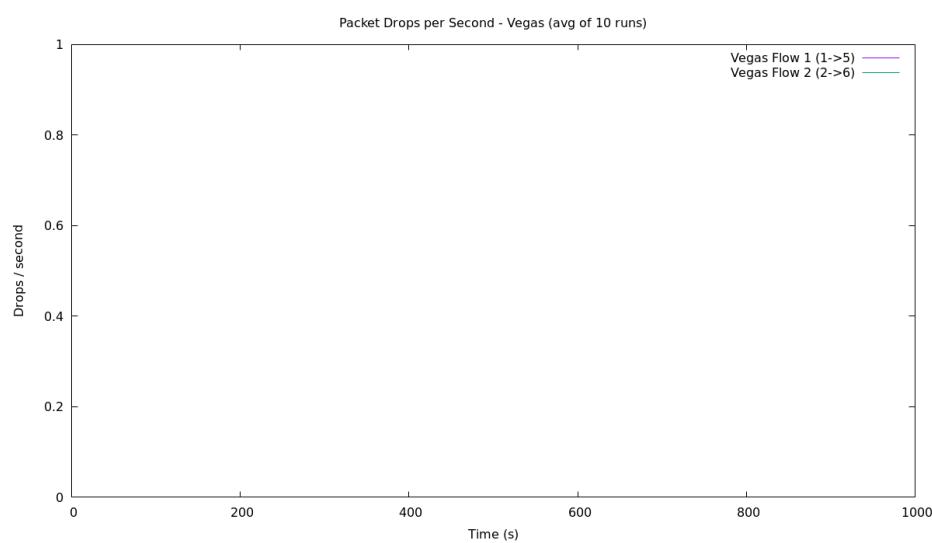
شکل ۲۰: نمودار برای throughput Vegas زوم شده، ۰ تا ۵۰ ثانیه



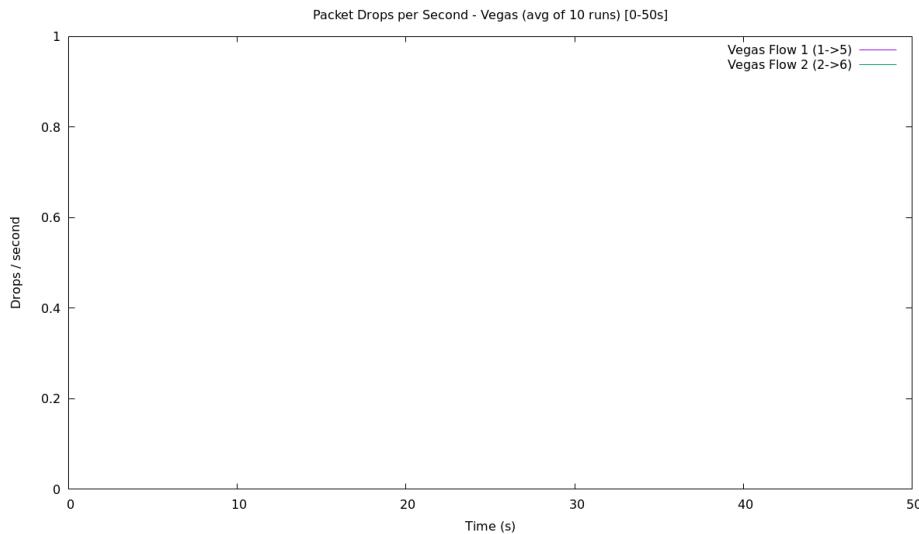
شکل ۲۱: نمودار برای RTT Vegas، ۰ تا ۱۰۰۰ ثانیه



شکل ۲۲: نمودار RTT برای Vegas زومشده، ۰ تا ۵۰ ثانیه

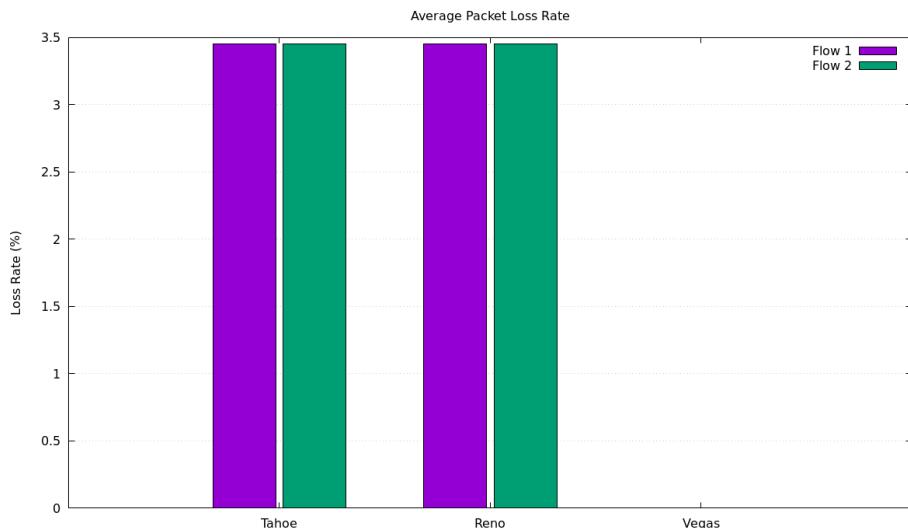


شکل ۲۳: نمودار loss برای Vegas، ۰ تا ۱۰۰۰ ثانیه



شکل ۲۴: نمودار loss برای Vegas رومشده، ۰ تا ۵۰ ثانیه

۶.۷ نمودار میله‌ای نرخ اتلاف متوسط



شکل ۲۵: نمودار میله‌ای برای نرخ loss شدن هر دو جریان هر سه پروتکل

۸ تحلیل نتایج و علت یابی

در این بخش، نتایج به تفکیک هر متريک تحلیل می‌شوند. دقیقاً بر مبنای نمودارهای تولید شده و مقادیر مشاهده شده در خروجی‌ها نوشته شده‌اند:

۱.۸ تحلیل Throughput

۱.۱.۸ مشاهدات

در نمودارهای شکل ۱ و شکل ۴، رفتار کلی چنین دیده می‌شود:

- گذردهی هر جریان در طول زمان نوسان دارد و این نوسان در دو خانواده loss-based (یعنی Tahoe/Reno) و Vegas Flow 1 به شکل تقریبی در بازه‌ی حدود ۰.۰۲۰ تا ۰.۰۲۵ Mbps قرار دارد و Vegas Flow 2 به شکل تقریبی در بازه‌ی حدود ۰.۱۰ تا ۰.۱۲۵ Mbps (در نمودار تفکیکی Vegas این دو سطح تقریباً ثابت دیده می‌شوند).
- در Vegas، گذردهی‌ها بسیار پایدارتر و پله‌ای یا تنابی اما منظم هستند (در نمودارهای شکل ۱۹ و ۲۰ کاملاً مشخص است).
- سطح گذردهی Vegas Flow 1 در گذردهی‌ها حول مقادیر میانی تر (حدود ۰.۱۴ تا ۰.۱۹ Mbps) است و Reno و Tahoe (نمودارهای شکل ۷ و ۱۳) گذردهی‌ها برای جریان‌ها) نوسان می‌کنند و افتهای تیز بیشتری دارند.

۲.۱.۸ علت یابی

تفاوت بنیادی Vegas با Tahoe/Reno در این است که:

- Reno و Tahoe عموماً packet loss هستند؛ یعنی سیگنال اصلی ازدحام را از رخ دادن (یا duplicate ACK(timeout)) می‌گیرند. پس تا زمانی که صفحات پر نشده و drop رخ نداده، ممکن است پنجره رشد کند و شبکه را به آستانه‌ی ازدحام نزدیک کند؛ سپس با drop کاهش می‌یابد و گذردهی افت می‌کند. این چرخه باعث نوسان‌های واضح در گذردهی می‌شود.
- Vegas رویکرد delay-based دارد؛ یعنی با پایش RTT و افزایش تأخیر (نشانه‌ی رشد صفحات)، قبل از رسیدن به مرحله‌ی drop، نرخ ارسال یا پنجره را تنظیم می‌کند تا صفحات در محدوده‌ی کوچک‌تری نگه داشته شوند. نتیجه‌ی عملی این رفتار، گذردهی پایدارتر و جلوگیری از سقوط‌های ناشی از loss است.

بنابراین، پایدار بودن و منظم بودن Throughput Vegas در باوجود drop (که در بخش Loss واضح است) هم‌راستا است.

۲.۸ RTT تحلیل

۱.۲.۸ مشاهدات

نمودارهای شکل ۲ و ۵ (۶-خطی) و نیز نمودارهای تفکیکی برای هر TCP نشان می‌دهند:

- در Reno و Tahoe، RTT جهش‌های بزرگ و تکرارشونده دارد و تا حوالی ۸۰۰ ns بالا می‌رود (در شکل‌های ۹ و ۱۵ واضح است).
- در نمودارهای زومشده (مثل نمودار شکل ۱۰ و ۱۵)، این جهش‌ها به شکل دندانه‌دار و رفت‌وبرگشتی دیده می‌شوند که با چرخه‌ی پر و خالی شدن صفحه‌ها هم خوان است.
- در Vegas، نمودار شکل ۲۱ نشان می‌دهد در کل بازه، RTT به شکل بسیار محدود و فشرده ظاهر شده و در نمودار شکل ۲۲ نیز فقط چند نمونه‌ی ابتدای اتصال دیده می‌شود و بعد عملاً نمونه‌ای ثبت نشده است.

۲.۲.۸ علت یابی

این بخش دو علت را جدا می‌کند: یکی علت رفتاری پروتکل و یکی علت ماهیت داده‌ی رسم شده:

لایه ۱: علت رفتاری پروتکل

- در Reno/Tahoe پروتکل تا مرز ازدحام جلو می‌رود، صفحه‌ای روترا پر می‌شوند، زمان انتظار در صف (queueing delay) زیاد می‌شود و RTT رشد می‌کند. سپس با loss و کاهش پنجره، صفحه تخلیه شده و در آلت می‌کند؛ بنابراین RTT نوسانی و بزرگ می‌شود.
- در Vegas اما هدف این است که با کنترل بر اساس delay، صفحه را کوچک نگه دارد؛ پس انتظار می‌رود RTT نسبتاً پایدارتر باشد و به جهش‌های شدید نرسد.

لایه ۲: علت ماهیت داده‌ی رسم شده از خود نمودارهای Vegas مشخص است که سری زمانی RTT بعد از چند ثانیه تقریباً داده‌ی قابل رسم ندارد و به همین دلیل در بازه‌ی ۰ تا ۱۰۰۰ عملای خطی دیده نمی‌شود یا فقط نزدیک زمان صفحه فشرده می‌شود. این معمولاً یکی از این دو معنی را دارد (و هر دو با خروجی فعلی سازگارند):

- یا مکانیزم ثبت RTT sample در سناریو برای Vegas بسیار کم تعداد و محدود اتفاق افتاده (مثلاً فقط در فاز ابتدایی اتصال).
- یا روش استخراج RTT (فایل‌های trace/rtt) برای Vegas بعد از شروع، رکورد جدیدی تولید نکرده است و بنابراین ابزار رسم، چیزی برای نمایش ندارد.

لذا نتیجه‌ی مهم این است که نمودار Vegas RTT در کل بازه، اطلاعات نمونه‌ای محدود دارد و در زوم فقط همان نمونه‌های ابتدای اتصال قابل مشاهده است. این دقیقا همان چیزی است که در نمودارهای شکل ۲۱ و ۲۲ می‌بینیم.

۳.۸ تحلیل Packet Loss

۱.۳.۸ مشاهدات

نمودارهای شکل ۳ و ۶ (خطی) و نمودارهای تفکیکی نشان می‌دهند:

- در Reno و Tahoe، dropها به صورت ضربه‌های متناوب و پر تکرار مشاهده می‌شوند (به شکل خطوط عمودی متراکم در نمودارهای ۱۰۰۰ ثانیه‌ای مثل نمودارهای شکل ۱۱ و ۱۷).
- در زوم ۵۰ تا ۱۰۰ ثانیه (نمودارهای شکل ۱۲ و ۱۸ و همچنین شکل ۶)، الگوی افت‌ها به شکل پالس‌های تکی (اغلب مقدار ۰ یا ۱ drop/sec) دیده می‌شود.
- در Vegas، نمودارهای شکل ۲۳ و ۲۴ عملای هیچ drop را نشان نمی‌دهند (نمودار تقریبا خالی است).

۲.۳.۸ نرخ اتلاف متوسط (نمودار میله‌ای)

نمودار میله‌ای شکل ۲۵ و داده‌ی خلاصه‌ی تولیدشده نشان می‌دهد:

- نرخ اتلاف متوسط برای Reno و Tahoe تقریبا یکسان و حدود ۳.۴۵٪ برای هر دو جریان است.
- نرخ اتلاف متوسط برای Vegas برابر ۰٪ گزارش شده است.

این نتیجه با نمودارهای زمانی نیز سازگار است: Tahoe/Reno پالس‌های drop دارند، اما Vegas علماً ندارد.

۳.۳.۸ علت یابی

چرا Tahoe/Reno اتلاف دارند؟ طبق صورت پروژه، از دست رفتن روی لینک‌ها نداریم و اتلاف تنها به دلیل محدود بودن بافر روترا رخ می‌دهد. همچنین اندازه صفحه ۱۰ بسته است. در Tahoe/Reno چون پروتکل‌ها به صورت loss-based رفتار می‌کنند، پنجره تا جایی رشد می‌کند که صفحه پر شود و در نتیجه بسته‌ها در روتر drop شوند؛ سپس TCP واکنش نشان می‌دهد و پنجره را کم می‌کند. این چرخه باعث ایجاد drop‌های دوره‌ای می‌شود. چرا Vegas اتلاف ندارد؟ قبل از رسیدن صفحه به حالت اشباع، با دیدن افزایش RTT (تاخیر صفحه) نرخ ارسال را تنظیم می‌کند؛ بنابراین معمولاً اجازه نمی‌دهد بافر ۱۰ تایی پر شود و بسته drop شود. نتیجه‌های قابل مشاهده در خروجی‌های Vegas تقریباً صفر و گذردهی پایدارتر دارد.

۹ جمع‌بندی مقایسه‌ای نهایی

با کنار هم گذاشتن سه معیار، یک تصویر کلی از رفتار سه پروتکل به دست می‌آید:

- Reno و Tahoe رفتار بسیار نزدیک به هم نشان می‌دهند (هم در نرخ اتلاف متوسط و هم در الگوی نوسان Throughput و RTT). این هم‌رفتاری در خروجی کاملاً مشهود است: نرخ اتلاف متوسط هر دو تقریباً برابر گزارش شده و نمودارهای زمانی نیز الگوی مشابه دارند.
- Vegas از نظر Loss بهترین نتیجه را دارد (تقریباً صفر)، و از نظر Throughput نیز پایدارتر و منظم‌تر دیده می‌شود.
- Tahoe/Reno در RTT جهش‌های شدید دارد که نشان‌دهنده‌ی تشکیل صف و ازدحام است؛ در Vegas نمودار RTT نمونه‌های محدود‌تری نشان می‌دهد و در سطح نمایش‌داده شده، ازدحام و صف شدید به سبک مشاهده نمی‌شود.