



دانشکده کامپیوتر- دانشگاه صنعتی شریف

مدیریت جریان داده با روش مبتنی بر اعتبار

درس:

مدار های واسط

استاد:

جناب آقای دکتر فصحتی

دانشجو:

نیکا قادری

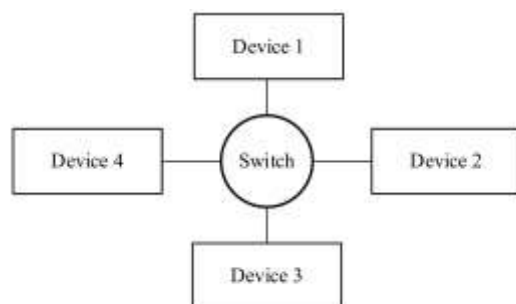
دی ماه ۱۴۰۳

فهرست مطالب

۳	مقدمه
۵	بخش اول: روش شناسی
۵	معماری شبکه
۶	ورودی‌ها
۶	مدیریت جریان داده با روش Credit-Based
۷	انواع بسته
۷	اولویت‌بندی بسته‌ها
۷	تشخیص و تصحیح خطا
۹	فایل‌های لاگ
۱۰	دستگاه‌ها
۱۱	سوئیچ
۱۳	کنترلر
۱۵	بخش دوم: اجرا
۱۵	نمونه اول
۲۰	نمونه دوم
۲۷	بخش سوم: بررسی کد
۲۷	Device.py
۲۷	تابع check_alert
۲۸	تابع process_incoming
۳۰	تابع send_packets
۳۱	Switch.py
۳۱	تابع broadcast
۳۱	تابع process_packet
۳۲	تابع restore_buffers
۳۳	تابع listen
۳۵	Controller.py
۳۷	سخن پایانی

مقدمه

در این گزارش، رویکردی جهت مدیریت جریان داده با استفاده از credit-based flow control را شبیه سازی می کنیم. در این سامانه چهار عدد دستگاه وجود دارد که می توانند بسته ارسال یا دریافت کنند اما ارتباط مستقیمی بین آنها وجود ندارد، بلکه یک سویچ (switch) وجود دارد که انتقال بین بسته ها را کنترل می کند. هر دستگاه، نرخ معینی برای ارسال بسته به هر یک از دستگاه های دیگر دارد. وقتی دستگاهی، یک بسته دریافت می کند، ابتدا بسته در بافر آن ذخیره می شود. سپس بعد از زمان معینی این بسته پردازش می شود و از بافر پاک می شود.



فرضیات به شکل زیر است:

- (۱) برای سادگی فرض کنید سویچ ایدئال است و کلیه فرایندهای داخلی آن بدون هیچ محدودیت زمانی و حافظه ای انجام می شود.
- (۲) در این معماری دو نوع متفاوت بسته موجود است؛ ولی طول آنها یکسان و برابر با ۵۱۲ بیت است.
- (۳) بافرهای دریافت در دستگاه های ۱، ۲، ۳ و ۴ به ترتیب برابر با ۱، ۱، ۲ و ۴ کیلوبایت است.
- (۴) بافرهای ارسال ایدئال فرض شده و محدودیتی ندارند.
- (۵) کلیه لینک ها Full-Duplex است که تأخیر ارسال و دریافت در آنها ایدئال است و می توان از آنها صرف نظر کنید.
- (۶) نرخ ارسال از دستگاه ۱ به دستگاه های ۳، ۲ و ۴ به ترتیب ۱۰، ۲۰ و ۳۰ بسته در ثانیه است.
- (۷) نرخ ارسال از دستگاه ۲ به دستگاه های ۱، ۳ و ۴ به ترتیب ۱۰، ۲۰ و ۳۰ بسته در ثانیه است.
- (۸) نرخ ارسال از دستگاه ۳ به دستگاه های ۱، ۲ و ۴ به ترتیب ۱۰، ۲۰ و ۳۰ بسته در ثانیه است.
- (۹) نرخ ارسال از دستگاه ۴ به دستگاه های ۱، ۲ و ۳ به ترتیب ۱۰، ۲۰ و ۳۰ بسته در ثانیه است.
- (۱۰) فرض بر این است که کلیه بسته ها بدون خطا دریافت می شوند.
- (۱۱) نرخ پردازش در کلیه دستگاه ها یکسان و برابر با ۱۰ بسته در ثانیه است.

این شبیه سازی قرار است به سوالات صفحه بعد پاسخ دهد.

۱) اگر فرض کنید که اولویت هر دو بسته یکسان است و در هر دستگاه به همان تعداد که بسته از نوع ۱ داریم، بسته از نوع ۲ هم موجود است.

۲) اگر فرض کنید اولویت بسته ۱ دو برابر اولویت بسته ۲ است و در هر دستگاه به ازای هر ۴ بسته ۱، یک بسته ۲ هم موجود است.

برای هر کدام از موارد بالا ابتدا الگوریتم مدیریت جریان داده با روش مبتنی بر اعتبار را طراحی کنید و کارکرد آن را توضیح دهید. سپس شبیه‌ساز آن را پیاده‌سازی کنید و برای مدت زمان ۲۰ ثانیه آن را اجرا کنید. در هر ثانیه وضعیت بافرهای دریافت هر دستگاه و میزان اعتبار باقی‌مانده برای هر نوع بسته در هر دستگاه را در خروجی چاپ کنید. دقت شود در تمام مراحل باید سرریز بافرهای خود را چک کنید و هر جا که سرریز بافر رخ دهد، پیاده‌سازی یا الگوریتم شما به درستی کار نمی‌کند و نیاز به تصحیح آن است.

برای توضیح راه حل، گزارش را به سه بخش تقسیم کردم:

۱. توضیح فرضیات، روش‌های استفاده شده و عملکرد کلی برنامه.

۲. نمونه‌هایی از اجرای برنامه و تحلیل خروجی‌ها.

۳. تجزیه و تحلیل کد نوشته شده.

در ادامه به بررسی بخش‌های مختلف می‌پردازیم.

بخش اول: روش شناسی

معماری شبکه

ابتدا به معرفی عناصر و فایل‌هایی موجود در قالب کلی برنامه می‌پردازیم:

بسته: هر بسته شامل موارد زیر است:

- شناسه یا Id
- طول پیام (بسته‌های معمولی ۵۱۲ بیت دارند؛ اما من بسته‌های دیگری را نیز برای الگوریتم تصحیح طراحی کردم که بعداً توضیح می‌دهم).
- دستگاه مقصد
- نوع پیام (تایپ ۱ یا تایپ ۲)

دستگاه: هر دستگاه یک فایل مخصوص به خود را دارد. وظیفه هر دستگاه این است که در هر ثانیه، تعداد مشخصی بسته برای سوئیچ بفرستد، که نرخ ارسال آنها برای هر یک از دستگاه‌های دیگر طبق صورت سوال مشخص شده است. همچنین، هر دستگاه می‌تواند تعدادی بسته از سوئیچ دریافت کند و در بافر دریافتش – که مقداری محدود دارد – بنویسد. سپس با سرعت ۱۰ بسته بر ثانیه بسته‌ها را پردازش می‌کند. یعنی در هر ثانیه حداکثر ۱۰ بسته از روی بافرش پاک می‌کند.

سوئیچ: در هر لحظه باید بررسی کند که آیا بسته‌ای از سمت دستگاه‌های دیگر رسیده است یا خیر. اگر بسته موجود بود، حافظه بافر دستگاه مقصد را چک می‌کند. اگر حافظه به مقدار کافی وجود داشت، پیام را برای دستگاه مقصد می‌فرستد. در غیر این صورت، بسته را دراپ می‌کند.

کنترلر: این فایل نوشته شده است تا ارتباطات را راه اندازی کند؛ یعنی ورودی‌های مناسب را از کاربر بگیرد، از کلاس‌های مختلف نمونه بگیرد و رشته/thread های لازم را بسازد. در بخش سوم دقیق‌تر به ساختار این فایل می‌پردازیم.

حال که عناصر شبکه معرفی شدند، پروتکل و الگوریتم استفاده‌شده را بررسی می‌کنیم.

ورودی‌ها

ابتدا از کاربر خواسته می‌شود که زمان شبیه‌سازی را انتخاب کند. سپس از او خواسته می‌شود تا بین سوال ۱ و ۲، یکی را انتخاب کند. در صورتی که کاربر سوال ۲ را انتخاب کرد، درخواست می‌شود تا روشی را که برای اولویت‌بندی بسته‌ها در نظر گرفته، نیز مشخص کند. در ادامه بیشتر در این مورد توضیح خواهیم داد.

مدیریت جریان داده با روش Credit-Based

ارسال و دریافت پیام‌ها با استفاده از صف داده/Queue های مختلف انجام می‌شود. کنترلر، به هر دستگاه یک switch queue می‌دهد تا بسته‌های ارسالی خود را در آن بگذارد. همچنین از هر دستگاه یک received_packets می‌گیرد تا دستگاه بتواند بسته‌های ورودی از سمت سوئیچ را داشته باشد. در نهایت این queue ها توسط کنترلر، با سوئیچ به اشتراک گذاشته می‌شوند تا سوئیچ به همه queue ها دسترسی داشته باشد.

میزان credit/کردیت هر دستگاه، برابر با میزان حافظه باقی مانده از بافر دریافتش است. (واحد: ۱ بیت) در هر ثانیه، کردیت همه دستگاه‌ها ۱۰ بسته افزایش می‌یابد یا به مقدار اولیه خودش می‌رسد (هر بسته: ۵۱۲ بیت).

همچنین سوئیچ به این مقادیر نیز دسترسی دارد (منطق کردیت‌ها را در سوئیچ پیاده‌سازی کردم). موقع ارسال بسته به دستگاه‌ها، سوئیچ فقط وقتی بسته را می‌فرستد که مقصد به اندازه کافی کردیت داشته باشد و اگر بسته ارسال شد، از کردیت مقصد کم می‌کند.

انواع بسته

دو نوع بسته داریم، که بسته به شرایط برنامه به نسبت معینی تولید می‌شوند. برای مثال، در سوال یک، تعداد بسته از نوع اول با نوع دوم برابر است؛ و به این صورت پیاده‌سازی شده که دستگاه‌ها یکی در میان بسته نوع یک و دو ارسال می‌کنند تا فراوانی آنها در بسته‌های ارسالی برابر بماند. یا مثلاً به ازای هر ۴ بسته از نوع ۱، یک بسته از نوع دوم ارسال شود.

اولویت‌بندی بسته‌ها

در سوال ۱، که اولویت دو بسته یکسان است، بسته‌های ورودی با همان ترتیبی که وارد شدند، نیز پردازش می‌شوند. اما در سوال ۲، که اولویت بسته نوع اول دو برابر بسته نوع دوم است، از سه روش مختلف برای پیاده‌سازی این مفهوم استفاده شده است، که یکی از آنها توسط کاربر انتخاب می‌شود:

۱. داده‌های نوع یک، همیشه بر داده‌های نوع دوم اولویت داشته باشند و در هر ثانیه، ابتدا بسته‌های نوع اول پردازش شوند.

۲. اولویت‌بندی فقط در موقعیت رقابتی وجود دارد. فرض کنید بافر فضای کمی دارد و ما دو بسته از انواع مختلف داریم. در این زمان بسته نوع اول انتخاب می‌شود تا مطمئن شویم به مقصد فرستاده می‌شود.

۳. به ازای هر دو بسته نوع یک، یک بسته نوع دو به مقصد فرستاده شود. به عبارتی، بیشتر بسته‌های نوع یک انتخاب می‌شوند و فراوانی آنها در بافر مقصد، تقریباً دو برابر بسته‌های نوع دو است.

تشخیص و تصحیح خطا

هنگام فرستادن بسته‌ها، ممکن است بافر مقصد به اندازه کافی فضا نداشته باشد. در این صورت برنامه بسته را دراپ می‌کند و به کاربر اطلاع می‌دهد. در اینجا تشخیص خطا اتفاق می‌افتد. از آنجایی که این رویکرد چندان مطلوبی نیست، با اعمال تغییراتی در شبکه سعی می‌شود تا این ایراد برطرف شود:

۱. مشکل اول اینجاست که نمی‌خواهیم وقتی بسته‌ای دراپ می‌شود، کلاً حذف شود و هیچ‌گاه به مقصد نرسد. به همین دلیل، بعد از اطلاع به کاربر، سامانه دوباره این بسته را به صف داده تزریق می‌کند (requeue) تا بتواند در آینده به مقصد ارسال شود.

۲. وقتی بافر مقصد در یک دستگاه پر می‌شود، دیگر از طرف دستگاه‌های دیگر نمی‌توان به آن بسته‌ای فرستاد. بنابراین سوئیچ پیامی به بقیه دستگاه‌ها می‌فرستد تا آن‌ها به طور موقت ارسال بسته را به دستگاه مقصد متوقف کنند.

۳. از آنجائیکه دوست نداریم بافرها سریع پر شوند و overflow داشته باشیم، از رویکردی به نام backpressure استفاده می‌کنیم. به این صورت که:

- سوئیچ کردیت‌ها را بررسی/observe می‌کند و مرتب آن‌ها را با مقدار threshold خودشان چک می‌کند (در واقع اگر میزان فضای خالی یک بافر از threshold آن کمتر شد، سوئیچ باید اقداماتی انجام دهد. مقدار threshold به طور دلخواه تعیین می‌شود، مثلاً ۳۵ درصد بافر).
 - اگر کردیت یک دستگاه از threshold آن کمتر شد، سوئیچ پیامی به بقیه دستگاه‌ها می‌فرستد و اطلاع می‌دهد که کردیت مقصد کم است و باید در ارسال بسته صرفه‌جویی کنند.
- دستگاه‌های دیگر نیز هر بار که این پیام را می‌گیرند، سرعت ارسال بسته به آن مقصد خاص را نصف می‌کنند. در نتیجه تا حد امکان سعی می‌شود از رخداد overflow جلوگیری شود.
- در هر ثانیه، هر دستگاه ۱۰ بسته را پردازش می‌کند و این بسته‌ها از بافرها پاک می‌شوند و به کردیت دستگاه اضافه می‌شود. اگر کردیت از threshold بیشتر شد، سوئیچ باری دیگر به همه دستگاه‌های دیگر پیامی مخابره می‌کند و به آنها اطلاع می‌دهد که می‌توانند سرعت ارسال خود را اضافه کنند. آن‌ها نیز به "آرامی" سرعت خود را افزایش می‌دهند تا به مقدار اولیه برسند.

فایل‌های لاگ

به منظور خوانایی بهتر، نتایج شبیه‌سازی در دو فایل log ذخیره شده‌اند. شرح این دو فایل به صورت زیر است:

Simulation.log: پیام‌هایی را که سوئیچ می‌فرستد، ثبت می‌کند. برای مثال:

- اطلاعات هر بسته‌ای که برای دستگاهی فرستاده می‌شود. مثل فرستنده، گیرنده و ...
- کردیت باقی‌مانده برای هر دستگاه با ارسال هر بسته
- اطلاعات بسته‌هایی که دراپ می‌شوند و به صف داده بر می‌گردند.
- سیگنال‌های Backpressure که در جهت کاهش سرعت ارسال بسته‌ها فرستاده می‌شوند.
- سیگنال‌های Critical Backpressure که وقتی ارسال می‌شوند که ظرفیت بافر مقصد به صفر برسد.
- سیگنال‌های Restore که کردیت دستگاه‌ها را زیاد می‌کنند.

Memory.log: پیام‌های مربوط به ۴ دستگاه موجود را ثبت می‌کند. شامل:

- اطلاعات بسته‌هایی که دستگاه در هر زمان به سوئیچ می‌فرستد.
- پیام‌های Backpressure که از سوئیچ می‌گیرد و نحوه تاثیر آن بر روی برنامه خود. مثلاً سرعت ارسال را چه قدر تغییر می‌دهند.
- پیام‌های Critical Backpressure دریافتی از سوئیچ و نحوه تاثیر آن‌ها روی دستگاه.
- پیام‌های Restore دریافتی از سوئیچ و نحوه تاثیر آن‌ها روی دستگاه.
- وضعیت بافر دریافتی که با آن بسته‌های موجود در بافر را مشاهده می‌کنیم.
- بسته‌هایی که در هر ثانیه پردازش می‌شوند و از بافر خارج می‌شوند، نیز ثبت می‌شوند.

دستگاه‌ها

ساختار دستگاه‌ها بسیار شبیه به هم است و فقط در شماره یا id خود با هم تفاوت دارند. برای همین در این قسمت به معرفی کلاس "دستگاه" یا "device" می‌پردازیم.

مقادیر ثابت:

- TRANSMISSION_RATES: آرایه‌ای که نرخ ارسال‌های اولیه را در خود نگه می‌دارد.
- PROCESS_RATE: نرخ پردازش بسته در هر ثانیه (۱۰).

اتریبیوت‌های کلاس دستگاه:

هر دستگاه اطلاعاتی را در خودش نگه می‌دارد که در __init__ مقدار دهی یا initialize می‌شوند:

- شماره یا id
- Received Packets که پیام‌های دریافتی از سوئیچ در آن قرار می‌گیرند.
- Switch Queue که پیام‌های ارسالی به آن در سوئیچ قرار می‌گیرند.
- Running که اگر False باشد سیستم خاموش می‌شود و اگر True باشد دستگاه به عملکرد عادی خود

ادامه می‌دهد.

- Current Rates که نشان‌دهنده نرخ فعلی ارسال داده به دستگاه‌های مختلف است. در ابتدا با یک کپی از TRANSMISSION_RATES مقداردهی می‌شود.

- Logger که پیام‌ها و اعلانات دستگاه روی آن نوشته می‌شوند. Memory Logger به هر دستگاه داده

می‌شود.

- Ratio: نسبت فراوانی بسته‌های نوع یک به نوع دو در آن ذخیره می‌شود.

- Ratio Counter: ابتدا صفر است. موقع ارسال بسته‌ها، اگر بسته از نوع اول باشد یک واحد به این مقدار اضافه می‌شود. اگر از نوع دوم باشد، مقدار Ratio از این شمارنده کم می‌شود. قبل از ارسال هر بسته، با توجه به مقدار کنونی Ratio Counter و مقایسه آن با صفر، تصمیم می‌گیریم که چه نوع بسته‌ای بفرستیم.
- Duration: مدت زمان اجرای برنامه را نشان می‌دهد.

توابع به کار رفته:

- Check Alerts: همواره Received Packets را چک می‌کند و به دنبال پیام‌های Backpressure, Critical Backpressure و Restore می‌گردد. در صورت دریافت این پیام‌ها، سرعت انتقال داده را تنظیم می‌کند.
- Process Incoming: ابتدا محتویات بافر را چاپ می‌کند و سپس، تا ۱۰ بسته از بافر خود خارج کرده، پردازش می‌کند و در فایل لاگ شماره آنها را می‌نویسد. این کار را در هر ثانیه تکرار می‌کند.
- Send Packets: در هر ثانیه، با توجه به نرخ ارسال، یک سری بسته می‌سازد و آنها را برای سوئیچ می‌فرستد.

و بدین صورت فایل دستگاه پیاده‌سازی می‌شود.

سوئیچ

مقادیر ثابت:

- BUFFER_SIZES: آرایه‌ای که اندازه بافرهای دریافتی هر دستگاه را بر حسب تعداد بیت ذخیره می‌کند.
- PROCESS_RATE: سرعت پردازش بسته برای دستگاه‌ها
- DEVICES_NUMBER: تعداد دستگاه‌ها
- PROGRAM_START_TIME: زمان شروع برنامه

اترپیوت‌های کلاس سوئیچ:

- Incoming Queues: صف داده، شامل بسته‌هایی که بقیه دستگاه‌ها به سوئیچ ارسال می‌کنند.
- Outgoing Queues: بسته‌هایی را که سوئیچ به دستگاه‌ها ارسال می‌کند، نگه می‌دارد.
- STATE: حالت برنامه. در واقع همان سوال اول و سوال دوم در فایل میان‌ترم است.
- Priority Mode: نحوه پیاده‌سازی اولویت که توسط کاربر مشخص می‌شود.
- Buffers: کردیت‌های بافرها را در هر لحظه نگه می‌دارد. در ابتدا با BUFFER_SIZES مقداردهی می‌شود.

- Running: تا وقتی True است، توابع فعال هستند. بدین گونه زمان اجرای برنامه را کنترل می‌کنیم.
- Logger: پیام‌ها و اعلانات سوئیچ روی آن نوشته می‌شوند. همان Simulation Logger است.
- Lock: برای مدیریت اختلال/concurrency برنامه است، هنگامی که چند Thread می‌خواهند هم‌زمان

از کلاس سوئیچ استفاده کنند.

توابع به‌کار رفته:

- Broadcast: این تابع یک دستگاه مقصد و یک پیام می‌گیرد و پیام را به بقیه دستگاه‌ها مخابره می‌کند. هنگام فرستادن سیگنال‌های کنترل سرعت از آن استفاده شده است.
- Process Packet: وظیفه دارد بسته‌های دریافتی را بگیرد و در صورت کردیت کافی به دستگاه مقصد بفرستد. در غیر این صورت، عدم ارسال را روی لاگ می‌نویسد و بسته را به صف داده خود بر می‌گرداند. همچنین در هر مرحله با توجه به کردیت باقی‌مانده می‌تواند خطاب به بقیه دستگاه‌ها Backpressure, Critical Restore و Backpressure (سیگنال‌های کنترل سرعت) ارسال کند.

- Restore Buffers: چون دستگاه‌ها هر ثانیه تعدادی بسته را پردازش می‌کنند، کردیت‌های هر دستگاه باید در هر ثانیه آپدیت شوند. وظیفه تابع همین است. همچنین بسته به نیاز می‌تواند سیگنال‌های کنترل سرعت هم بفرستد. همچنین از مقادیر افزوده‌شده به هر کردیت و اندازه نهایی بافرها در آن لحظه لاگ می‌گیرد.
- Listen: این تابع بررسی می‌کند که چه پکت‌هایی به سوئیچ ارسال می‌شوند و آن‌ها را به ترتیب مناسب پردازش می‌کند. با توجه به شرایط انتخاب شده، ممکن است اولویت بسته‌ها برابر باشد. در این شرایط تغییری در صف داده ورودی نمی‌دهد (state = 1).

اگر داشته باشیم state = 2:

Priority Mode = 1: همیشه ابتدا بسته‌های نوع اول برای هر دستگاه پردازش می‌شوند و بعد بسته‌های نوع دوم.

Priority Mode = 2: فقط هرگاه کردیت مقصد کم بود (> 10 /)، بسته‌های نوع یک به دو ترجیح داده شوند.

Priority Mode = 3: همیشه به ازای هر دو بسته نوع یک، یک بسته نوع دو بفرستیم. به عبارتی، نسبت

فراوانی بسته‌های نوع یکی که به مقصد می‌فرستیم به بسته‌های نوع دو، ۲ باشد. به این صورت اولویت ۱/۲ پیاده‌سازی می‌شود.

در آخر تابع Process Packet فراخوانی می‌شود.

در نتیجه منطق سوئیچ نیز کامل می‌شود.

کنترلر

وظیفه راه‌اندازی thread ها را برعهده دارد.

توابع به کار رفته:

- Process: این تابع ارتباطی به شبکه ندارد و تنها برای خوانایی بیشتر فایل‌های لاگ نوشته شده است.
- در واقع یک level جدید به نام process به لاگر اضافه می‌کند.
- Get Simulation Duration: زمان اجرا را از کاربر می‌گیرد.
- Get Simulation Ratio: ابتدا state برنامه را ورودی می‌گیرد. اگر این مقدار دو باشد، یعنی برای بسته‌ها اولویت متفاوت داریم و باید تابع Get Priority Option فراخوانده شود.
- و باید مقدار Priority Option نیز از کاربر ورودی گرفته شود.
- Get Priority Option: نحوه پیاده‌سازی منطق "اولویت‌بندی" در این تابع توسط کاربر انتخاب می‌شود.
- Setup Loggers: فایل‌های لاگ را خالی می‌کند و لاگرها را تنظیم می‌کند، به طور مثال فرمت پیام‌های نوشته شده در این تابع مشخص می‌شود.
- Stop Simulation: فلگ‌های Running را در سوئیچ و دستگاه‌ها False می‌کند تا غیرفعال شوند.
- Main: با فراخوانی توابع بالا ابتدا ورودی را از کاربر می‌گیرد. سپس از کلاس دستگاه‌ها و سوئیچ نمونه می‌گیرد. برای سوئیچ، دو آرایه داریم: ورودی و خروجی. هر کدام شامل ۴ صف داده هستند که هر صف داده به یک دستگاه تعلق دارد. هر Queue در آرایه ورودی سوئیچ به خروجی دستگاه متناظرش مشترک است و هر Queue در آرایه خروجی سوئیچ با صف داده ورودی دستگاه مشترک است.
- در مرحله بعد تعدادی thread ساخته می‌شود تا توابع زیر اجرا شوند:
 - Listen و Restore Buffers در سوئیچ
 - Send Packets، Process Incoming و Check Alerts در تمام دستگاه‌ها
- سپس این thread ها اجرا می‌شوند و بعد از گذشت زمانی که به ثانیه در Duration مشخص شده، اجرای همه آن‌ها متوقف می‌شود و کنترلر برنامه را تمام می‌کند.

منطق برنامه توضیح داده شد. حالا به بررسی نمونه‌هایی از اجرای آن می‌پردازیم.

بخش دوم: اجرا

نمونه اول

"سوال یک" را اجرا می‌کنیم. در این حالت state باید یک باشد.

```
Enter the simulation duration in seconds (positive integer):
20
Enter the simulation state
1: Packets are equal in priority and number.
2: Packets from type 1 have double priority and are 4 times more common than packets of type 2.
1
```

بعد از بیست ثانیه:

```
Enter the simulation duration in seconds (positive integer):
20
Enter the simulation state
1: Packets are equal in priority and number.
2: Packets from type 1 have double priority and are 4 times more common than packets of type 2.
1
Simulation completed.
```

می‌توان روند اجرای برنامه را در فایل simulation.log مشاهده کرد:

```

[20:11:22] [MainThread] [INFO] Switch is running. Ready to process packets...
[20:11:25] [MainThread] [INFO] Starting simulation...
[20:11:25] [SwitchListener] [INFO] Switch: Listening for incoming packets...
[20:11:25] [BufferRestorer] [INFO] Switch: Buffer restoration thread started.
[20:11:26] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 7688 bits.
[20:11:26] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 7688 bits.
[20:11:26] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 1 sent. Remaining buffer for Device 1: 7168 bits.
[20:11:26] [SwitchListener] [INFO] Switch: type1 packet from Device 4 to Device 1 sent. Remaining buffer for Device 1: 6656 bits.
[20:11:26] [SwitchListener] [INFO] Switch: type2 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 7168 bits.
[20:11:26] [SwitchListener] [INFO] Switch: type2 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 6144 bits.
[20:11:26] [SwitchListener] [INFO] Switch: type2 packet from Device 3 to Device 1 sent. Remaining buffer for Device 1: 5632 bits.
[20:11:26] [SwitchListener] [INFO] Switch: type2 packet from Device 4 to Device 1 sent. Remaining buffer for Device 1: 5120 bits.
[20:11:26] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 6656 bits.
[20:11:26] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 4608 bits.
[20:11:26] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 1 sent. Remaining buffer for Device 1: 4096 bits.
[20:11:26] [SwitchListener] [INFO] Switch: type1 packet from Device 4 to Device 1 sent. Remaining buffer for Device 1: 3584 bits.
[20:11:26] [SwitchListener] [INFO] Switch: type2 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 6144 bits.
[20:11:26] [SwitchListener] [INFO] Switch: type2 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 3072 bits.
[20:11:26] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.
[20:11:26] [SwitchListener] [INFO] Switch: type2 packet from Device 3 to Device 1 sent. Remaining buffer for Device 1: 2560 bits.
[20:11:26] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.
[20:11:26] [SwitchListener] [INFO] Switch: type2 packet from Device 4 to Device 1 sent. Remaining buffer for Device 1: 2048 bits.
[20:11:26] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.
[20:11:26] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 5632 bits.
[20:11:26] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 1536 bits.
[20:11:26] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.

```

در ابتدا که فضا زیاد است اکثر بسته‌ها با نرخ عادی فرستاده می‌شوند. برای مثال اولین ارسال را در نظر بگیرید. ساینز

بافر دستگاه اول $1024 * 8 = 8192$ بیت بوده که پس از ارسال یک بسته ۵۱۲ بیتی، به ۷۶۸۰ بیت می‌رسد (این بسته

روی memory log هم نوشته می‌شود).

```

[20:11:28] [SwitchListener] [INFO] Switch: type2 packet from Device 1 to Device 4 sent. Remaining buffer for Device 4: 11264 bits.
[20:11:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[20:11:28] [SwitchListener] [INFO] Switch: type2 packet from Device 2 to Device 4 sent. Remaining buffer for Device 4: 10752 bits.
[20:11:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[20:11:28] [SwitchListener] [INFO] Switch: type2 packet from Device 3 to Device 4 sent. Remaining buffer for Device 4: 10240 bits.
[20:11:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[20:11:28] [SwitchListener] [WARNING] Switch: Packet from Device 4 to Device 3 dropped due to buffer overflow. Buffer space remainin
[20:11:28] [SwitchListener] [INFO] Switch: Re-queued packet 1958 from Device 4 back to incoming queue.
[20:11:28] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 4 sent. Remaining buffer for Device 4: 9728 bits.
[20:11:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[20:11:28] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 4 sent. Remaining buffer for Device 4: 9216 bits.
[20:11:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[20:11:28] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 4 sent. Remaining buffer for Device 4: 8704 bits.
[20:11:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[20:11:28] [SwitchListener] [WARNING] Switch: Packet from Device 4 to Device 3 dropped due to buffer overflow. Buffer space remainin

```

با گذشت زمان و کمتر شدن ظرفیت بافر برای دستگاه‌ها، به تدریج پیام‌های Backpressure فرستاده می‌شود.


```

[28:11:28] [SwitchListener] [WARNING] Switch: Packet from Device 4 to Device 3 dropped due to buffer overflow. Buffer space remainin
[28:11:28] [SwitchListener] [INFO] Switch: Re-queued packet 5122 from Device 4 back to incoming queue.
[28:11:28] [SwitchListener] [INFO] Switch: type2 packet from Device 1 to Device 4 sent. Remaining buffer for Device 4: 8192 bits.
[28:11:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[28:11:28] [SwitchListener] [INFO] Switch: type2 packet from Device 2 to Device 4 sent. Remaining buffer for Device 4: 7688 bits.
[28:11:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[28:11:28] [SwitchListener] [INFO] Switch: type2 packet from Device 3 to Device 4 sent. Remaining buffer for Device 4: 7188 bits.
[28:11:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[28:11:28] [SwitchListener] [WARNING] Switch: Packet from Device 4 to Device 3 dropped due to buffer overflow. Buffer space remainin
[28:11:28] [SwitchListener] [INFO] Switch: Re-queued packet 6956 from Device 4 back to incoming queue.
[28:11:28] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 4 sent. Remaining buffer for Device 4: 6656 bits.
[28:11:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[28:11:28] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 4 sent. Remaining buffer for Device 4: 6144 bits.
[28:11:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[28:11:28] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 4 sent. Remaining buffer for Device 4: 5632 bits.
[28:11:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[28:11:28] [SwitchListener] [WARNING] Switch: Packet from Device 4 to Device 3 dropped due to buffer overflow. Buffer space remainin
[28:11:28] [SwitchListener] [INFO] Switch: Re-queued packet 1349 from Device 4 back to incoming queue.
[28:11:28] [SwitchListener] [INFO] Switch: type2 packet from Device 1 to Device 4 sent. Remaining buffer for Device 4: 5128 bits.
[28:11:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[28:11:28] [SwitchListener] [INFO] Switch: type2 packet from Device 2 to Device 4 sent. Remaining buffer for Device 4: 4688 bits.
[28:11:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[28:11:28] [SwitchListener] [INFO] Switch: type2 packet from Device 3 to Device 4 sent. Remaining buffer for Device 4: 4096 bits.
[28:11:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[28:11:28] [SwitchListener] [WARNING] Switch: Packet from Device 4 to Device 3 dropped due to buffer overflow. Buffer space remainin
[28:11:28] [SwitchListener] [INFO] Switch: Re-queued packet 9861 from Device 4 back to incoming queue.

```

همچنین از آنجایی که موقع ارسال داده، نرخ‌های ارسال در هر ثانیه توسط دستگاه‌ها چک می‌شوند و ممکن است در

این میان بافری پر شده باشد، ممکن است بسته‌هایی داشته باشیم که دراپ شوند. در این صورت دوباره که Queue برگردانده می‌شوند.

```

[20:11:26] [SwitchListener] [INFO] Switch: Re-queued packet 8238 from Device 3 back to incoming queue.
[20:11:26] [SwitchListener] [WARNING] Switch: Packet from Device 4 to Device 2 dropped due to buffer overflow. Buffer space remain
[20:11:26] [SwitchListener] [INFO] Switch: Re-queued packet 2084 from Device 4 back to incoming queue.
[20:11:26] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 3 sent. Remaining buffer for Device 3: 7688 bits.
[20:11:26] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 3 sent. Remaining buffer for Device 3: 7168 bits.
[20:11:26] [SwitchListener] [WARNING] Switch: Packet from Device 3 to Device 2 dropped due to buffer overflow. Buffer space remain
[20:11:26] [SwitchListener] [INFO] Switch: Re-queued packet 3437 from Device 3 back to incoming queue.
[20:11:26] [SwitchListener] [WARNING] Switch: Packet from Device 4 to Device 2 dropped due to buffer overflow. Buffer space remain
[20:11:26] [SwitchListener] [INFO] Switch: Re-queued packet 1876 from Device 4 back to incoming queue.
[20:11:26] [BufferRestorer] [PROCESS] Switch: Restored credit for Device 1 by 640.0 bytes. Current credit size: 640.0 bytes.
[20:11:26] [BufferRestorer] [PROCESS] Switch: Sent RESTORE signal for Device 1 as buffer usage is below the threshold.
[20:11:26] [BufferRestorer] [PROCESS] Switch: Restored credit for Device 2 by 640.0 bytes. Current credit size: 640.0 bytes.
[20:11:26] [BufferRestorer] [PROCESS] Switch: Sent RESTORE signal for Device 2 as buffer usage is below the threshold.
[20:11:26] [BufferRestorer] [PROCESS] Switch: Restored credit for Device 3 by 640.0 bytes. Current credit size: 1536.0 bytes.
[20:11:26] [BufferRestorer] [PROCESS] Switch: Sent RESTORE signal for Device 3 as buffer usage is below the threshold.
[20:11:26] [BufferRestorer] [PROCESS] Switch: Restored credit for Device 4 by 0.0 bytes. Current credit size: 4096.0 bytes.
[20:11:26] [BufferRestorer] [PROCESS] Switch: Sent RESTORE signal for Device 4 as buffer usage is below the threshold.
[20:11:27] [SwitchListener] [INFO] Switch: type2 packet from Device 1 to Device 3 sent. Remaining buffer for Device 3: 11776 bits.
[20:11:27] [SwitchListener] [INFO] Switch: type2 packet from Device 2 to Device 3 sent. Remaining buffer for Device 3: 11264 bits.
[20:11:27] [SwitchListener] [INFO] Switch: type2 packet from Device 3 to Device 2 sent. Remaining buffer for Device 2: 4608 bits.
[20:11:27] [SwitchListener] [INFO] Switch: type2 packet from Device 4 to Device 2 sent. Remaining buffer for Device 2: 4096 bits.
[20:11:27] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 3 sent. Remaining buffer for Device 3: 10752 bits.
[20:11:27] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 3 sent. Remaining buffer for Device 3: 10240 bits.
[20:11:27] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 2 sent. Remaining buffer for Device 2: 3584 bits.
[20:11:27] [SwitchListener] [INFO] Switch: type1 packet from Device 4 to Device 2 sent. Remaining buffer for Device 2: 3072 bits.

```

باخالی شدن فضا در هر بافر، مقداری ظرفیت خالی می‌شود و این باعث می‌شود تا بعد از restore شدن، بسته‌ها دوباره بتوانند ارسال شوند.

حال محتویات memory.log را بررسی می‌کنیم:

```
[20:11:26] [Device4AlertHandler] [WARNING] Device 4: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 10.
[20:11:26] [Device3AlertHandler] [WARNING] Device 3: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 5.
[20:11:26] [Device1AlertHandler] [WARNING] Device 1: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 2.
[20:11:26] [Device4AlertHandler] [WARNING] Device 4: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 5.
[20:11:26] [Device1AlertHandler] [WARNING] Device 1: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 1.
[20:11:26] [Device4AlertHandler] [WARNING] Device 4: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 2.
[20:11:26] [Device3AlertHandler] [WARNING] Device 3: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 2.
[20:11:26] [Device3AlertHandler] [WARNING] Device 3: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 1.
[20:11:26] [Device4AlertHandler] [WARNING] Device 4: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 1.
[20:11:26] [Device1AlertHandler] [CRITICAL] Device 1: Received CRITICAL_BACKPRESSURE signal. Stopping transmission to Device 2.
[20:11:26] [Device4AlertHandler] [CRITICAL] Device 4: Received CRITICAL_BACKPRESSURE signal. Stopping transmission to Device 2.
[20:11:26] [Device3AlertHandler] [CRITICAL] Device 3: Received CRITICAL_BACKPRESSURE signal. Stopping transmission to Device 2.
[20:11:26] [Device3AlertHandler] [INFO] Device 3: Received RESTORE signal. Speeding up transmission to Device 1 to 1.
[20:11:26] [Device1Sender] [INFO] Device 1: Sent 50 packets to the switch.
[20:11:26] [Device1AlertHandler] [INFO] Device 1: Received RESTORE signal. Speeding up transmission to Device 2 to 1.
[20:11:26] [Device4AlertHandler] [INFO] Device 4: Received RESTORE signal. Speeding up transmission to Device 1 to 1.
[20:11:26] [Device2AlertHandler] [INFO] Device 2: Received RESTORE signal. Speeding up transmission to Device 1 to 1.
[20:11:26] [Device1Processor] [INFO] Buffer Status: Device 1: Buffer Content (IDs and Types):
[{'id': 7451, 'type': 'type1'}, {'id': 3355, 'type': 'type1'}, {'id': 4855, 'type': 'type1'}, {'id': 3797, 'type': 'type2'}, {'id': 4188, 'type': 'type2'}, {'id': 3169, 'type': 'type2'}, {'id': 7957, 'type': 'type1'}, {'id': 3652, 'type': 'type1'}, {'id': 3159, 'type': 'type1'}, {'id': 8977, 'type': 'type2'}, {'id': 9841, 'type': 'type2'}, {'id': 4490, 'type': 'type2'}, {'id': 6597, 'type': 'type1'}, {'id': 1720, 'type': 'type1'}, {'id': 3168, 'type': 'type1'}, {'id': 2812, 'type': 'type2'}] Total Packets: 16
[20:11:26] [Device3AlertHandler] [INFO] Device 3: Received RESTORE signal. Speeding up transmission to Device 2 to 1.
[20:11:26] [Device2Processor] [INFO] Buffer Status: Device 2: Buffer Content (IDs and Types):
[{'id': 5261, 'type': 'type1'}, {'id': 6176, 'type': 'type2'}, {'id': 4527, 'type': 'type1'}, {'id': 7669, 'type': 'type2'}, {'id': 4677, 'type': 'type1'}, {'id': 5116, 'type': 'type2'}, {'id': 7202, 'type': 'type1'}, {'id': 3985, 'type': 'type2'}, {'id': 2244, 'type': 'type1'}, {'id': 8865, 'type': 'type2'}, {'id': 8499, 'type': 'type1'}, {'id': 6550, 'type': 'type1'}, {'id': 3944, 'type': 'type2'}, {'id': 8488, 'type': 'type2'}, {'id': 5610, 'type': 'type1'}, {'id': 2656, 'type': 'type1'}] Total Packets: 16
```

مشاهده می‌شود که تعدادی سیگنال‌های کنترل سرعت داریم. محتویات بافرها در هر ثانیه نمایش داده می‌شود و

همچنین پکت‌هایی که از روی بافر برداشته می‌شوند، نیز چاپ می‌شوند.


```
[20:11:30] [Device3Processor] [INFO] Buffer Status: Device 3: Buffer Content (IDs and Types):
[{'id': 7129, 'type': 'type2'}, {'id': 3851, 'type': 'type2'}, {'id': 1668, 'type': 'type1'}, {'id': 7207, 'type': 'type1'}, {'id': 1352, 'type': 'type2'}, {'id': 1177, 'type': 'type2'}, {'id': 2652, 'type': 'type1'}, {'id': 9035, 'type': 'type1'}, {'id': 1713, 'type': 'type2'}, {'id': 3579, 'type': 'type2'}, {'id': 7808, 'type': 'type1'}, {'id': 7514, 'type': 'type2'}, {'id': 7244, 'type': 'type1'}, {'id': 9861, 'type': 'type2'}, {'id': 1989, 'type': 'type1'}, {'id': 5848, 'type': 'type2'}, {'id': 8434, 'type': 'type1'}, {'id': 5713, 'type': 'type2'}, {'id': 2541, 'type': 'type1'}, {'id': 8042, 'type': 'type2'}, {'id': 3177, 'type': 'type1'}, {'id': 5396, 'type': 'type2'}, {'id': 1056, 'type': 'type2'}, {'id': 6598, 'type': 'type1'}, {'id': 6954, 'type': 'type2'}, {'id': 8132, 'type': 'type1'}, {'id': 9613, 'type': 'type2'}, {'id': 3405, 'type': 'type1'}, {'id': 8451, 'type': 'type2'}, {'id': 2407, 'type': 'type1'}, {'id': 2176, 'type': 'type2'}, {'id': 3883, 'type': 'type1'}, {'id': 9878, 'type': 'type2'}, {'id': 4051, 'type': 'type2'}, {'id': 7383, 'type': 'type1'}, {'id': 6618, 'type': 'type1'}, {'id': 8138, 'type': 'type1'}, {'id': 6895, 'type': 'type2'}, {'id': 4881, 'type': 'type2'}, {'id': 8582, 'type': 'type2'}, {'id': 2718, 'type': 'type1'}, {'id': 3597, 'type': 'type1'}] Total Packets: 42.
[20:11:30] [Device4Processor] [PROCESS] Device 4: Processed packets: [3391, 1558, 6875, 4849, 4589, 6603, 8728, 5645, 1693, 4855]
[20:11:30] [Device2Processor] [PROCESS] Device 2: Processed packets: [5762, 4184, 2885, 2519, 5798, 1726, 3493, 2611, 8238, 2884]
[20:11:30] [Device3Processor] [PROCESS] Device 3: Processed packets: [7129, 3851, 1668, 7207, 1352, 1177, 2652, 9035, 1713, 3579]
[20:11:30] [Device3AlertHandler] [CRITICAL] Device 3: Received CRITICAL_BACKPRESSURE signal. Stopping transmission to Device 4.
[20:11:30] [Device2AlertHandler] [CRITICAL] Device 2: Received CRITICAL_BACKPRESSURE signal. Stopping transmission to Device 4.
[20:11:30] [Device1AlertHandler] [CRITICAL] Device 1: Received CRITICAL_BACKPRESSURE signal. Stopping transmission to Device 4.
[20:11:30] [Device1Sender] [INFO] Device 1: Sent 2 packets to the switch.
[20:11:31] [Device3AlertHandler] [INFO] Device 3: Received RESTORE signal. Speeding up transmission to Device 1 to 2.
[20:11:31] [Device3AlertHandler] [INFO] Device 3: Received RESTORE signal. Speeding up transmission to Device 2 to 3.
[20:11:31] [Device3AlertHandler] [INFO] Device 3: Received RESTORE signal. Speeding up transmission to Device 4 to 1.
[20:11:31] [Device4AlertHandler] [INFO] Device 4: Received RESTORE signal. Speeding up transmission to Device 1 to 2.
[20:11:31] [Device4AlertHandler] [INFO] Device 4: Received RESTORE signal. Speeding up transmission to Device 2 to 3.
[20:11:31] [Device4AlertHandler] [INFO] Device 4: Received RESTORE signal. Speeding up transmission to Device 3 to 1.
[20:11:31] [Device2AlertHandler] [INFO] Device 2: Received RESTORE signal. Speeding up transmission to Device 1 to 2.
[20:11:31] [Device2AlertHandler] [INFO] Device 2: Received RESTORE signal. Speeding up transmission to Device 3 to 1.
[20:11:31] [Device2AlertHandler] [INFO] Device 2: Received RESTORE signal. Speeding up transmission to Device 4 to 1.
[20:11:31] [Device1AlertHandler] [INFO] Device 1: Received RESTORE signal. Speeding up transmission to Device 2 to 3.
[20:11:31] [Device1AlertHandler] [INFO] Device 1: Received RESTORE signal. Speeding up transmission to Device 3 to 1.
```

همچنین یک سری سیگنال Critical Backpressure داریم که سرعت ارسال مربوطه را صفر می‌کنند. وقتی هم که Restore اتفاق می‌افتد اجازه داریم سرعت را افزایش دهیم. این کار به آرامی انجام می‌شود؛ یعنی هر بار یک واحد به سرعت مربوطه اضافه می‌شود.

نمونه دوم

اکنون "سوال دوم" را شبیه‌سازی می‌کنیم. گزینه سوم برای اولویت‌دهی انتخاب شده است.

```
Enter the simulation duration in seconds (positive integer):
20
Enter the simulation state
1: Packets are equal in priority and number.
2: Packets from type 1 have double priority and are 4 times more common than packets of type 2.
2
Choose priority management option:
1) Packets of type 1 are always processed before packets of type 2.
2) Only when competing for remaining buffer space, packets of type 1 are preferred to type 2.
3) Switch processes 2 packets of type 1 for every 1 packet of type 2.
Enter 1, 2, or 3:
1
```

توجه کنید که هر دستگاه باید نسبت پیام‌های نوع ۱ به نوع ۲ را رعایت کند. به عبارت دیگر، به ازای هر ۴ پیام از نوع

۱، ۱ پیام از نوع ۲ داریم. شکل صفحه بعد نمونه‌ای از فراوانی پیام‌های ارسال شده برای دستگاه اول بررسی می‌کند (فایل

simulation.log). با نگاه به تایپ بسته‌ها متوجه برقراری این نسبت می‌شویم.

```
[17:18:27] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 7688 bits.
[17:18:27] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 7688 bits.
[17:18:27] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 1 sent. Remaining buffer for Device 1: 7168 bits.
[17:18:27] [SwitchListener] [INFO] Switch: type1 packet from Device 4 to Device 1 sent. Remaining buffer for Device 1: 6656 bits.
[17:18:27] [SwitchListener] [INFO] Switch: type2 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 7168 bits.
[17:18:27] [SwitchListener] [INFO] Switch: type2 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 6144 bits.
[17:18:27] [SwitchListener] [INFO] Switch: type2 packet from Device 3 to Device 1 sent. Remaining buffer for Device 1: 5632 bits.
[17:18:27] [SwitchListener] [INFO] Switch: type2 packet from Device 4 to Device 1 sent. Remaining buffer for Device 1: 5128 bits.
[17:18:28] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 6656 bits.
[17:18:28] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 6144 bits.
[17:18:28] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 1 sent. Remaining buffer for Device 1: 5696 bits.
[17:18:28] [SwitchListener] [INFO] Switch: type1 packet from Device 4 to Device 1 sent. Remaining buffer for Device 1: 5184 bits.
[17:18:28] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 6144 bits.
[17:18:28] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 5696 bits.
[17:18:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.
[17:18:28] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 1 sent. Remaining buffer for Device 1: 5184 bits.
[17:18:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.
[17:18:28] [SwitchListener] [INFO] Switch: type1 packet from Device 4 to Device 1 sent. Remaining buffer for Device 1: 4672 bits.
[17:18:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.
[17:18:28] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 5632 bits.
[17:18:28] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 5136 bits.
[17:18:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.
[17:18:28] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 1 sent. Remaining buffer for Device 1: 4624 bits.
[17:18:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.
[17:18:28] [SwitchListener] [INFO] Switch: type1 packet from Device 4 to Device 1 sent. Remaining buffer for Device 1: 4112 bits.
[17:18:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.
[17:18:28] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 5128 bits.
```

همچنین سوئیچ بسته‌های نوع یک را به نوع دوم ترجیح می‌دهد. از آنجایی که زمان ارسال بسته‌ها دقیقاً با هم یکی

نیست و هر بار سوئیچ آرایه‌ای از بسته‌ها را بررسی می‌کند، ممکن است در بافرهای مقصد، بسته‌های نوع دو قبل از نوع

یک قرار بگیرند. برای مثال، فرض کنید محتویات صف داده‌ای که سوئیچ در سه چرخه متوالی بررسی می‌کند، به صورت

زیر باشد:

Cycle 1: { (id=1014,type1), (id=2026,type2), (id=2014,type1), (id=1026,type1) }

Cycle 2: { (id=1004,type2), (id=2006,type1), (id=2004,type1) }

Cycle 3: { (id=1010,type1), (id=2020,type1), (id=2010,type2), (id=1020,type2) }

اگر Priority Mode = 1، سوئیچ در هر چرخه داده‌ها را به شکل زیر مرتب می‌کند:

Cycle 1: { (id=1014,type1), (id=2014,type1), (id=1026,type1), (id=2026,type2)}

Cycle 2: { (id=2006,type1), (id=2004,type1), (id=1004,type2) }

Cycle 3: { (id=1010,type1), (id=2020,type1), (id=2010,type2), (id=1020,type2)}

و آن‌ها را با این ترتیب به دستگاه مقصد می‌فرستد. از آنجایی که این پردازش داخل خود سوئیچ انجام می‌شود و خیلی محسوس نیست، قطعه کد مربوط به آن را استثنائاً اینجا آورده‌ایم تا طرز کار آن مشخص باشد. کد صفحه بعد در واقع بخشی از تابع Listen است که قبل از پردازش بسته‌ها، آن‌ها را مرتب می‌کند.

```
if self.STATE == 1:
    pass
else:
    for target_device, packets in packets_to_process.items():
        if self.PRIORITY_MODE == 1:
            packets.sort(key=lambda pkt: pkt[1]["type"] == 2)
        elif self.PRIORITY_MODE == 2:
            buffer_usage = self.buffers[target_device]
            if buffer_usage < 0.18 * BUFFER_SIZES[target_device]:
                packets.sort(key=lambda pkt: pkt[1]["type"] == 2)
        elif self.PRIORITY_MODE == 3:
            with self.lock:
                type1_packets = [pkt for pkt in packets if pkt[1]["type"] == "type1"]
                type2_packets = [pkt for pkt in packets if pkt[1]["type"] == "type2"]

                combined_packets = []
                type1_index, type2_index = 0, 0

                while type1_index < len(type1_packets) or type2_index < len(type2_packets):
                    if type1_index < len(type1_packets):
                        combined_packets.append(type1_packets[type1_index])
                        type1_index += 1
                    if type1_index < len(type1_packets):
                        combined_packets.append(type1_packets[type1_index])
                        type1_index += 1
                    if type2_index < len(type2_packets):
                        combined_packets.append(type2_packets[type2_index])
                        type2_index += 1

                packets[:] = combined_packets
```

مانند حالت قبل می‌توان اطلاعاتی از جریان انتقال داده را با نگاه به فایل simulation به دست آورد:


```

[17:10:30] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[17:10:30] [SwitchListener] [WARNING] Switch: Packet from Device 4 to Device 3 dropped due to buffer overflow. Buffer space remain
[17:10:30] [SwitchListener] [INFO] Switch: Re-queued packet 6862 from Device 4 back to incoming queue.
[17:10:30] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 4 sent. Remaining buffer for Device 4: 6656 bits.
[17:10:30] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[17:10:30] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 4 sent. Remaining buffer for Device 4: 6144 bits.
[17:10:30] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[17:10:30] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 4 sent. Remaining buffer for Device 4: 5632 bits.
[17:10:30] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.
[17:10:30] [SwitchListener] [WARNING] Switch: Packet from Device 4 to Device 3 dropped due to buffer overflow. Buffer space remain
[17:10:30] [SwitchListener] [INFO] Switch: Re-queued packet 7878 from Device 4 back to incoming queue.
[17:10:30] [SwitchListener] [INFO] Switch: type2 packet from Device 1 to Device 4 sent. Remaining buffer for Device 4: 5120 bits.
[17:10:30] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 4 due to high buffer utilization.

```

Handwritten notes on the first log: "Queue" with an upward arrow pointing to the re-queue messages, and "Overflow" with a downward arrow pointing to the dropped packet messages.

```

[17:10:29] [SwitchListener] [WARNING] Switch: Packet from Device 4 to Device 3 dropped due to buffer overflow. Buffer space remain
[17:10:29] [SwitchListener] [INFO] Switch: Re-queued packet 9989 from Device 4 back to incoming queue.
[17:10:29] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 4 sent. Remaining buffer for Device 4: 21504 bits.
[17:10:29] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 4 sent. Remaining buffer for Device 4: 28992 bits.
[17:10:29] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 4 sent. Remaining buffer for Device 4: 26480 bits.
[17:10:29] [SwitchListener] [WARNING] Switch: Packet from Device 4 to Device 3 dropped due to buffer overflow. Buffer space remain
[17:10:29] [SwitchListener] [INFO] Switch: Re-queued packet 6285 from Device 4 back to incoming queue.
[17:10:29] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 4 sent. Remaining buffer for Device 4: 9968 bits.
[17:10:29] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 4 sent. Remaining buffer for Device 4: 17536 bits.
[17:10:29] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 4 sent. Remaining buffer for Device 4: 15944 bits.
[17:10:29] [SwitchListener] [WARNING] Switch: Packet from Device 4 to Device 3 dropped due to buffer overflow. Buffer space remain
[17:10:29] [SwitchListener] [INFO] Switch: Re-queued packet 4498 from Device 4 back to incoming queue.
[17:10:29] [BufferRestorer] [PROCESS] Switch: Restored credit for Device 1 by 384.0 bytes. Current credit size: 1024.0 bytes.
[17:10:29] [BufferRestorer] [PROCESS] Switch: Sent RESTORE signal for Device 1 as buffer usage is below the threshold.
[17:10:29] [BufferRestorer] [PROCESS] Switch: Restored credit for Device 2 by 640.0 bytes. Current credit size: 640.0 bytes.
[17:10:29] [BufferRestorer] [PROCESS] Switch: Sent RESTORE signal for Device 2 as buffer usage is below the threshold.
[17:10:29] [BufferRestorer] [PROCESS] Switch: Restored credit for Device 3 by 640.0 bytes. Current credit size: 640.0 bytes.
[17:10:29] [BufferRestorer] [PROCESS] Switch: Sent RESTORE signal for Device 3 as buffer usage is below the threshold.
[17:10:29] [BufferRestorer] [PROCESS] Switch: Restored credit for Device 4 by 640.0 bytes. Current credit size: 3008.0 bytes.
[17:10:29] [BufferRestorer] [PROCESS] Switch: Sent RESTORE signal for Device 4 as buffer usage is below the threshold.
[17:10:29] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 4 sent. Remaining buffer for Device 4: 23552 bits.
[17:10:29] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 4 sent. Remaining buffer for Device 4: 23040 bits.

```

Handwritten notes on the second log: "Queue" with an upward arrow pointing to the re-queue message, and "Overflow" with a downward arrow pointing to the dropped packet messages. A large "S" is drawn around the BufferRestorer messages, and a "64" is written next to the credit restoration values.

```

[17:10:24] [MainThread] [INFO] Switch is running. Ready to process packets...
[17:10:27] [MainThread] [INFO] Starting simulation...
[17:10:27] [SwitchListener] [INFO] Switch: Listening for incoming packets...
[17:10:27] [BufferRestorer] [INFO] Switch: Buffer restoration thread started.
[17:10:27] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 7680 bits.
[17:10:27] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 7680 bits.
[17:10:27] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 1 sent. Remaining buffer for Device 1: 7168 bits.
[17:10:27] [SwitchListener] [INFO] Switch: type1 packet from Device 4 to Device 1 sent. Remaining buffer for Device 1: 6656 bits.
[17:10:27] [SwitchListener] [INFO] Switch: type2 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 7168 bits.
[17:10:27] [SwitchListener] [INFO] Switch: type2 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 6144 bits.
[17:10:27] [SwitchListener] [INFO] Switch: type2 packet from Device 3 to Device 1 sent. Remaining buffer for Device 1: 5632 bits.
[17:10:27] [SwitchListener] [INFO] Switch: type2 packet from Device 4 to Device 1 sent. Remaining buffer for Device 1: 5120 bits.
[17:10:28] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 6656 bits.
[17:10:28] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 6080 bits.
[17:10:28] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 1 sent. Remaining buffer for Device 1: 5600 bits.
[17:10:28] [SwitchListener] [INFO] Switch: type1 packet from Device 4 to Device 1 sent. Remaining buffer for Device 1: 5088 bits.
[17:10:28] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 4608 bits.
[17:10:28] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 4096 bits.
[17:10:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.
[17:10:28] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 1 sent. Remaining buffer for Device 1: 2560 bits.
[17:10:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.
[17:10:28] [SwitchListener] [INFO] Switch: type1 packet from Device 4 to Device 1 sent. Remaining buffer for Device 1: 2048 bits.
[17:10:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.
[17:10:28] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 5632 bits.
[17:10:28] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 1936 bits.
[17:10:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.
[17:10:28] [SwitchListener] [INFO] Switch: type1 packet from Device 3 to Device 1 sent. Remaining buffer for Device 1: 1808 bits.
[17:10:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.
[17:10:28] [SwitchListener] [INFO] Switch: type1 packet from Device 4 to Device 1 sent. Remaining buffer for Device 1: 512 bits.
[17:10:28] [SwitchListener] [WARNING] Switch: Backpressure signal sent to devices for Device 1 due to high buffer utilization.
[17:10:28] [SwitchListener] [INFO] Switch: type1 packet from Device 1 to Device 2 sent. Remaining buffer for Device 2: 5120 bits.
[17:10:28] [SwitchListener] [INFO] Switch: type1 packet from Device 2 to Device 1 sent. Remaining buffer for Device 1: 0 bits.
[17:10:28] [SwitchListener] [CRITICAL] Switch: Critical backpressure signal sent to all devices to stop sending to Device 1 due to

```

دو نکته:

- همچنان امکان دارد بافرها پر شوند و با پیغام overflow مواجه شویم که البته در این حالت بسته از دست نمی رود (طبق اقداماتی که در بخش تصحیح خطا انجام شد). چرا همچنان به این حالت می‌رسیم؟ چون با queue کردن بسته‌ها، صف داده کوچک نمی‌شود و در سایکل بعدی همچنان لازم است تا این بسته‌ها فرستاده شوند. اما چون بافرها همچنان پر هستند، تا وقتی عملیات restore فضای بافرها را بیشتر نکند همچنان نمی‌توانیم بسته‌ها را ارسال کنیم. همچنین در نظر بگیرید که ممکن است دستگاه‌های مختلف بسته‌های خود را هم‌زمان ارسال کرده باشند (هم‌زمان برای سوئیچ).

اگر در حین پردازش بسته‌ها، بافر یکی از دستگاه‌ها پر شود، تمام بسته‌های بعدی در آن سایکل که به آن

دستگاه فرستاده می‌شوند نیز queue می‌شوند و نمی‌توانیم آن‌ها را بفرستیم.

- هنگام پردازش بسته‌ها و بازیابی حافظه، بسته به این که تا الان بافر چه مقدار حافظه دارد، افزایش کردیت می‌تواند متفاوت باشد. به عبارت دیگر، در هر ثانیه "ماکزیمم" ۱۰ بسته پردازش می‌کنیم. برای مثال در تصویرهای بالا، دستگاه اول، $384 \times 8 / 512 = 6$ بسته پردازش کرده و کاملاً خالی شده است در حالی که دستگاه‌های دیگر همان ۱۰ بسته را پردازش کرده اند ($640 \times 8 / 512$).
- لازم به ذکر است که دستگاه ۴ حتی با وجود پاک کردن ۱۰ بسته، خالی نشده است و بافر آن هنوز تعدادی بسته دارد.

در آخر فایل memory.log را نیز بررسی می‌کنیم:

The screenshot shows a log file with the following entries (with some lines highlighted by red boxes):

```

[17:10:28] [Device1AlertHandler] [WARNING] Device 1: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 5.
[17:10:28] [Device4AlertHandler] [WARNING] Device 4: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 10.
[17:10:28] [Device4AlertHandler] [WARNING] Device 4: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 5.
[17:10:28] [Device4AlertHandler] [WARNING] Device 4: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 1.
[17:10:28] [Device1AlertHandler] [WARNING] Device 1: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 5.
[17:10:28] [Device3AlertHandler] [WARNING] Device 3: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 2.
[17:10:28] [Device1AlertHandler] [WARNING] Device 1: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 1.
[17:10:28] [Device1AlertHandler] [WARNING] Device 1: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 1.
[17:10:28] [Device3AlertHandler] [WARNING] Device 3: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 1.
[17:10:28] [Device4AlertHandler] [WARNING] Device 4: Received BACKPRESSURE signal. Slowing down transmission to Device 2 to 1.
[17:10:28] [Device1AlertHandler] [CRITICAL] Device 1: Received CRITICAL_BACKPRESSURE signal. Stopping transmission to Device 2.
[17:10:28] [Device4AlertHandler] [CRITICAL] Device 4: Received CRITICAL_BACKPRESSURE signal. Stopping transmission to Device 2.
[17:10:28] [Device3AlertHandler] [CRITICAL] Device 3: Received CRITICAL_BACKPRESSURE signal. Stopping transmission to Device 2.
[17:10:28] [Device2AlertHandler] [INFO] Device 2: Received RESTORE signal. Speeding up transmission to Device 1 to 1.
[17:10:28] [Device4AlertHandler] [INFO] Device 4: Received RESTORE signal. Speeding up transmission to Device 1 to 1.
[17:10:28] [Device4AlertHandler] [INFO] Device 4: Received RESTORE signal. Speeding up transmission to Device 2 to 1.
[17:10:28] [Device1AlertHandler] [INFO] Device 1: Received RESTORE signal. Speeding up transmission to Device 2 to 1.
[17:10:28] [Device3AlertHandler] [INFO] Device 3: Received RESTORE signal. Speeding up transmission to Device 1 to 1.
[17:10:28] [Device1Sender] [INFO] Device 1: Sent 51 packets to the switch.
[17:10:28] [Device3AlertHandler] [INFO] Device 3: Received RESTORE signal. Speeding up transmission to Device 2 to 1.
[17:10:28] [Device1Processor] [INFO] Buffer Status: Device 1: Buffer Content (IDs and Types):
[{'id': 5335, 'type': 'type1'}, {'id': 6470, 'type': 'type1'}, {'id': 4822, 'type': 'type1'}, {'id': 8907, 'type': 'type2'}, {'id': 3108, 'type': 'type2'}, {'id': 9842, 'type': 'type2'}, {'id': 4305, 'type': 'type1'}, {'id': 9813, 'type': 'type1'}, {'id': 7538, 'type': 'type1'}, {'id': 1778, 'type': 'type1'}, {'id': 7763, 'type': 'type1'}, {'id': 3584, 'type': 'type1'}, {'id': 3482, 'type': 'type1'}, {'id': 6827, 'type': 'type1'}, {'id': 8881, 'type': 'type1'}, {'id': 214, 'type': 'type1'}] Total Packets: 16
[17:10:28] [Device2Sender] [INFO] Device 2: Sent 51 packets to the switch.
[17:10:28] [Device2Processor] [INFO] Buffer Status: Device 2: Buffer Content (IDs and Types):
[{'id': 2996, 'type': 'type1'}, {'id': 9278, 'type': 'type2'}, {'id': 6849, 'type': 'type1'}, {'id': 3883, 'type': 'type1'}, {'id': 2234, 'type': 'type1'}, {'id': 2352, 'type': 'type1'}, {'id': 3456, 'type': 'type2'}, {'id': 6642, 'type': 'type1'}, {'id': 7775, 'type': 'type1'}, {'id': 2916, 'type': 'type1'}, {'id': 7216, 'type': 'type1'}, {'id': 6954, 'type': 'type1'}, {'id': 9535, 'type': 'type2'}, {'id': 1668, 'type': 'type2'}, {'id': 8486, 'type': 'type1'}, {'id': 4221, 'type': 'type1'}] Total Packets: 16

```

Handwritten Persian annotations on the log:

- سریع ارسال به این دستگاه‌ها (Fast delivery to these devices) - points to the first group of WARNING messages.
- در هر مرحله سرعت را با افزایش می‌دهد (Increases speed in each stage) - points to the INFO messages about speeding up transmission.
- بافر دستگاه ۱ در این لحظه پر شده است (Device 1 buffer is full at this moment) - points to the Buffer Status message for Device 1.

[17:10:30] [Device2Sender] [INFO] Device 2: Sent 2 packets to the switch.

[17:10:30] [Device4AlertHandler] [INFO] Device 4: Received RESTORE signal. Speeding up transmission to Device 1 to 3.

[17:10:30] [Device4AlertHandler] [INFO] Device 4: Received RESTORE signal. Speeding up transmission to Device 2 to 2.

[17:10:30] [Device4AlertHandler] [INFO] Device 4: Received RESTORE signal. Speeding up transmission to Device 3 to 1.

[17:10:30] [Device2AlertHandler] [INFO] Device 2: Received RESTORE signal. Speeding up transmission to Device 1 to 3.

[17:10:30] [Device2AlertHandler] [INFO] Device 2: Received RESTORE signal. Speeding up transmission to Device 3 to 1.

[17:10:30] [Device2AlertHandler] [INFO] Device 2: Received RESTORE signal. Speeding up transmission to Device 4 to 1.

[17:10:30] [Device3Sender] [INFO] Device 3: Sent 3 packets to the switch.

[17:10:30] [Device1AlertHandler] [INFO] Device 1: Received RESTORE signal. Speeding up transmission to Device 2 to 2.

[17:10:30] [Device3AlertHandler] [INFO] Device 3: Received RESTORE signal. Speeding up transmission to Device 1 to 3.

[17:10:30] [Device4Sender] [INFO] Device 4: Sent 6 packets to the switch.

[17:10:30] [Device1AlertHandler] [INFO] Device 1: Received RESTORE signal. Speeding up transmission to Device 3 to 1.

[17:10:30] [Device1AlertHandler] [INFO] Device 1: Received RESTORE signal. Speeding up transmission to Device 4 to 1.

[17:10:30] [Device4Processor] [INFO] Buffer Status: Device 4: Buffer Content (IDs and Types):

{'id': 9044, 'type': 'type1'}, {'id': 1221, 'type': 'type1'}, {'id': 7471, 'type': 'type1'}, {'id': 1854, 'type': 'type1'}, {'id': 4728, 'type': 'type1'}, {'id': 9772, 'type': 'type1'}, {'id': 8829, 'type': 'type1'}, {'id': 6475, 'type': 'type1'}, {'id': 4477, 'type': 'type2'}, {'id': 3306, 'type': 'type2'}, {'id': 1198, 'type': 'type2'}, {'id': 7303, 'type': 'type1'}, {'id': 8131, 'type': 'type1'}, {'id': 8228, 'type': 'type1'}, {'id': 9163, 'type': 'type1'}, {'id': 9266, 'type': 'type1'}, {'id': 7793, 'type': 'type1'}, {'id': 3825, 'type': 'type1'}, {'id': 7439, 'type': 'type1'}, {'id': 8596, 'type': 'type1'}, {'id': 9781, 'type': 'type1'}, {'id': 8847, 'type': 'type1'}, {'id': 9483, 'type': 'type1'}, {'id': 2246, 'type': 'type2'}, {'id': 9067, 'type': 'type2'}, {'id': 7276, 'type': 'type2'}, {'id': 6557, 'type': 'type1'}, {'id': 8649, 'type': 'type1'}, {'id': 7286, 'type': 'type1'}, {'id': 5696, 'type': 'type1'}, {'id': 3234, 'type': 'type1'}, {'id': 5458, 'type': 'type1'}, {'id': 2768, 'type': 'type1'}, {'id': 2543, 'type': 'type1'}, {'id': 8994, 'type': 'type1'}, {'id': 5872, 'type': 'type1'}, {'id': 3658, 'type': 'type1'}, {'id': 2116, 'type': 'type1'}, {'id': 5578, 'type': 'type2'}, {'id': 7473, 'type': 'type2'}, {'id': 8233, 'type': 'type2'}, {'id': 5219, 'type': 'type1'}, {'id': 5578, 'type': 'type1'}, {'id': 6848, 'type': 'type1'}, {'id': 2272, 'type': 'type1'}, {'id': 7719, 'type': 'type1'}, {'id': 3738, 'type': 'type1'}, {'id': 1807, 'type': 'type1'}, {'id': 7219, 'type': 'type1'}, {'id': 3819, 'type': 'type1'}, {'id': 4596, 'type': 'type1'}, {'id': 2819, 'type': 'type1'}, {'id': 7488, 'type': 'type1'}, {'id': 6593, 'type': 'type2'}, {'id': 2648, 'type': 'type2'}, {'id': 3688, 'type': 'type2'}, {'id': 9018, 'type': 'type1'}, {'id': 9883, 'type': 'type1'}, {'id': 1617, 'type': 'type1'}, {'id': 9593, 'type': 'type1'}, {'id': 4649, 'type': 'type1'}, {'id': 3314, 'type': 'type1'}, {'id': 5122, 'type': 'type1'}, {'id': 7028, 'type': 'type1'}] Total Packets: 64

[17:10:30] [Device3AlertHandler] [INFO] Device 3: Received RESTORE signal. Speeding up transmission to Device 2 to 2.

{'id': 5335, 'type': 'type1'}, {'id': 6478, 'type': 'type1'}, {'id': 4822, 'type': 'type1'}, {'id': 8907, 'type': 'type2'}, {'id': 3108, 'type': 'type2'}, {'id': 9842, 'type': 'type2'}, {'id': 4305, 'type': 'type1'}, {'id': 9813, 'type': 'type1'}, {'id': 7938, 'type': 'type1'}, {'id': 1778, 'type': 'type1'}, {'id': 7763, 'type': 'type1'}, {'id': 3584, 'type': 'type1'}, {'id': 3682, 'type': 'type1'}, {'id': 6827, 'type': 'type1'}, {'id': 6881, 'type': 'type1'}, {'id': 2146, 'type': 'type1'}] Total Packets: 16

[17:10:28] [Device2Sender] [INFO] Device 2: Sent 51 packets to the switch.

[17:10:28] [Device2Processor] [INFO] Buffer Status: Device 2: Buffer Content (IDs and Types):

{'id': 2996, 'type': 'type1'}, {'id': 9278, 'type': 'type2'}, {'id': 6849, 'type': 'type1'}, {'id': 3883, 'type': 'type1'}, {'id': 2214, 'type': 'type1'}, {'id': 2352, 'type': 'type1'}, {'id': 3456, 'type': 'type2'}, {'id': 6642, 'type': 'type1'}, {'id': 7775, 'type': 'type1'}, {'id': 2916, 'type': 'type1'}, {'id': 7216, 'type': 'type1'}, {'id': 4054, 'type': 'type1'}, {'id': 4535, 'type': 'type2'}, {'id': 1660, 'type': 'type2'}, {'id': 8486, 'type': 'type1'}, {'id': 4221, 'type': 'type1'}] Total Packets: 16

[17:10:28] [Device1Processor] [PROCESS] Device 1: Processed packets: [5335, 6478, 4822, 8907, 3108, 9842, 4305, 9813, 7938, 1778].

[17:10:28] [Device2Processor] [PROCESS] Device 2: Processed packets: [2996, 9278, 6849, 3883, 2214, 2352, 3456, 6642, 7775, 2916].

[17:10:28] [Device3Processor] [INFO] Buffer Status: Device 3: Buffer Content (IDs and Types):

{'id': 9422, 'type': 'type1'}, {'id': 7134, 'type': 'type1'}, {'id': 4861, 'type': 'type2'}, {'id': 8827, 'type': 'type2'}, {'id': 5688, 'type': 'type1'}, {'id': 7915, 'type': 'type1'}, {'id': 7640, 'type': 'type1'}, {'id': 1239, 'type': 'type1'}, {'id': 3193, 'type': 'type1'}, {'id': 3971, 'type': 'type1'}, {'id': 2188, 'type': 'type1'}, {'id': 2769, 'type': 'type1'}, {'id': 3858, 'type': 'type2'}, {'id': 7517, 'type': 'type2'}, {'id': 4194, 'type': 'type1'}, {'id': 1121, 'type': 'type1'}, {'id': 5236, 'type': 'type1'}, {'id': 8997, 'type': 'type1'}] Total Packets: 18

[17:10:28] [Device3Sender] [INFO] Device 3: Sent 32 packets to the switch.

[17:10:28] [Device3Processor] [PROCESS] Device 3: Processed packets: [9422, 7134, 4861, 8827, 5688, 7915, 7640, 1239, 3193, 3971].

بخش سوم: بررسی کد

در این بخش، نحوه پیاده‌سازی توضیحات بخش اول نشان داده می‌شود. سه کلاس switch, controller و device به حالت کلی (چون کد دستگاه‌ها یکسان است) بررسی می‌شوند.

Device.py

تابع check_alert

```
def check_alerts(self):
    start_time = time.time()
    while self.running and time.time() - start_time < self.DURATION:
        try:
            all_packets = list(self.received_packets.queue)

            for packet in all_packets:
                if packet['id'] in ["BACKPRESSURE", "RESTORE", "CRITICAL_BACKPRESSURE"]:
                    self.received_packets.queue.remove(packet)

                target_device = packet.get('target', None)
                current_rate = self.current_rates[target_device]

                if packet['id'] == "BACKPRESSURE":
                    self.current_rates[target_device] = max(1, current_rate // 2)
                    if current_rate != 1:
                        self.logger.warning(
                            f"Device {self.device_id}: Received BACKPRESSURE signal. Slowing down "
                            f"transmission to Device {target_device} to {self.current_rates[target_device]}."
                        )
                elif packet['id'] == "RESTORE":
                    self.current_rates[target_device] = min(
                        TRANSMISSION_RATES[self.device_id][target_device], current_rate + 1
                    )
                if current_rate != TRANSMISSION_RATES[self.device_id][target_device]:
                    self.logger.info(
                        f"Device {self.device_id}: Received RESTORE signal. Speeding up transmission to "
                        f"Device {target_device} to {self.current_rates[target_device]}."
                    )
```



```

elif packet['id'] == "CRITICAL_BACKPRESSURE":
    self.current_rates[target_device] = 0
    if current_rate != 0:
        self.logger.critical(
            f"Device {self.device_id}: Received CRITICAL_BACKPRESSURE signal. Stopping "
            f"transmission to Device {target_device}."
        )
except Exception as e:
    self.logger.error(f"Device {self.device_id}: Error in handling alert: {e}")

time.sleep(0.0001)

```

در واقع در این تابع، فقط پیام‌های کنترل سرعت از received_queue برداشته می‌شود. توجه کنید که چون حجم این پیام‌ها صفر بیت در نظر گرفته می‌شود، روی کردیت دستگاه تاثیری نمی‌گذارند.

تابع process_incoming

این تابع دو بخش دارد. بخش اول، محتویات کنونی بافر را در فایل لاگ می‌نویسد:

```

def process_incoming(self):
    start_time = time.time()
    while self.running and time.time() - start_time < self.DURATION:

        buffer_contents = list(self.received_packets.queue)
        filtered_buffer = [packet for packet in buffer_contents if packet['id'] not in ["BACKPRESSURE", "RESTORE",
                                                                                        "CRITICAL_BACKPRESSURE"]]

        if filtered_buffer:
            buffer_details = [{"id": packet['id'], "type": packet['type']} for packet in filtered_buffer]
            buffer_size = len(buffer_details)

            buffer_details_str = str(buffer_details)
            max_line_length = 120
            wrapped_buffer_details = '\n'.join(
                [buffer_details_str[i:i + max_line_length] for i in range(0, len(buffer_details_str),
                                                                    max_line_length)]
            )

            self.logger.info(
                f"Buffer Status: Device {self.device_id}: Buffer Content (IDs and Types):\n{wrapped_buffer_details}"
                f"Total Packets: {buffer_size}"
            )

```

و بخش دوم طراحی شده تا تعدادی بسته را در هر ثانیه پردازش کند و آن‌ها را از بافر خارج سازد. اطلاعات بسته‌های

پردازش شده نیز در فایل لاگ نوشته می‌شود:

```

processed_packets = []
counter = 0
while True:
    try:
        if self.received_packets.empty():
            break
        packet = self.received_packets.get_nowait()

        if packet['id'] in ["BACKPRESSURE", "RESTORE", "CRITICAL_BACKPRESSURE"]:
            self.received_packets.put(packet)
            continue
        counter += 1
        processed_packets.append(packet)
        if counter == PROCESS_RATE:
            break
    except Empty:
        break

if processed_packets:
    packet_ids = [p['id'] for p in processed_packets]
    self.logger.process(f"Device {self.device_id}: Processed packets: {packet_ids}.")

time.sleep(1)

```

```
def send_packets(self):
    start_time = time.time()
    packets_to_send = []

    while self.running and time.time() - start_time < self.DURATION:
        for target_device, rate in self.current_rates.items():
            for _ in range(rate):
                if self.ratio_counter <= 8:
                    packet_type = 'type1'
                    self.ratio_counter += 1
                else:
                    packet_type = 'type2'
                    self.ratio_counter -= self.RATIO

                packet_id = random.randint(1000, 9999)
                packet = {
                    "id": packet_id,
                    "size": 512,
                    "target": target_device,
                    "type": packet_type
                }
                packets_to_send.append(packet)

        for packet in packets_to_send:
            self.switch_queue.put(packet)

        self.logger.info(
            f"Device {self.device_id}: Sent {len(packets_to_send)} packets to the switch."
        )

        packets_to_send.clear()
        time.sleep(1)
```

توجه کنید که با توجه به پروتکل تصحیح خطا، بسته به نرخ ارسال کنونی ممکن است تعداد بسته‌های ارسالی در مجموع دچار تغییراتی شود. همچنین نحوه استفاده از ratio_counter این تضمین را می‌دهد که نسبت فراوانی بسته‌های ارسالی هر دستگاه رعایت شود.

```
def broadcast(self, message, exclude=None):
    if exclude is None:
        exclude = []
    for device_id in self.outgoing_queues:
        if device_id not in exclude:
            self.outgoing_queues[device_id].put(message)
```

که عملکرد این تابع در بخش اول توضیح داده شد.

توجه کنید که این تابع علاوه بر چک کردن این که فضای خالی داریم یا نه، باید در صورت نیاز پیام‌های کنترل سرعت را نیز بفرستد. همچنین چون چند thread ممکن است همزمان به این تابع دسترسی داشته باشند و مقادیر برنامه را تغییر دهند، از self.lock استفاده شده است. همچنین برای مخابره پیام‌ها از تابع broadcast استفاده کردیم.

```
def process_packet(self, source_device, packet):
    target_device = packet['target']
    packet_size = packet['size']

    BACKPRESSURE_THRESHOLD = 0.4 * BUFFER_SIZES[target_device]
    CRITICAL_THRESHOLD = 0
    packet_type = packet.get('type', 'unknown')

    with self.lock:
        if self.buffers[target_device] >= packet_size:
            self.buffers[target_device] -= packet_size
            self.outgoing_queues[target_device].put(packet)
            self.logger.info(
                f"Switch: {packet_type} packet from Device {source_device} to Device {target_device} sent. "
                f"Remaining buffer for Device {target_device}: {self.buffers[target_device]} bits."
            )

        if self.buffers[target_device] < BACKPRESSURE_THRESHOLD and self.buffers[target_device] != 0:
            backpressure_packet = {"id": "BACKPRESSURE", "size": 0, "target": target_device}
            self.broadcast(backpressure_packet, exclude=[target_device])

            self.logger.warning(
                f"Switch: Backpressure signal sent to devices for Device {target_device} "
                f"due to high buffer utilization."
            )
```

```

if self.buffers[target_device] == CRITICAL_THRESHOLD:
    critical_backpressure_packet = {"id": "CRITICAL_BACKPRESSURE", "size": 0, "target": target_device}
    self.broadcast(critical_backpressure_packet, exclude=[target_device])
    self.logger.critical(
        f"Switch: Critical backpressure signal sent to all devices to stop sending to Device "
        f"{target_device}"
        f"due to buffer overflow."
    )

else:
    self.logger.warning(
        f"Switch: Packet from Device {source_device} to Device {target_device} dropped due to buffer "
        f"overflow."
        f"Buffer space remaining: {self.buffers[target_device]} bits, Packet size: {packet_size} bits."
    )
    self.incoming_queues[source_device].put(packet)
    self.logger.info(
        f"Switch: Re-queued packet {packet['id']} from Device {source_device} back to incoming queue."
    )

```

تابع `restore_buffers`

این تابع از سمت سوئیچ اجرا می‌شود و سبب افزایش کردیت‌های هر دستگاه می‌شود. از آنجائی که همچنان ممکن است ظرفیت کمی برای بافرها داشته باشیم، این تابع می‌تواند در صورت نیاز، مثل `process_packet` برای دستگاه‌ها سیگنال‌های کنترلی بفرستد.


```

def restore_buffers(self):
    self.logger.info("Switch: Buffer restoration thread started.")
    while self.running:
        time.sleep(1)
        for device_id in self.buffers.keys():
            restored_size = PROCESS_RATE * 512
            before_restore = self.buffers[device_id]
            self.buffers[device_id] = min(self.buffers[device_id] + restored_size, BUFFER_SIZES[device_id])
            restored = self.buffers[device_id] - before_restore
            self.logger.process(
                f"Switch: Restored credit for Device {device_id} by {restored / 8} bytes. "
                f"Current credit size: {self.buffers[device_id] / 8} bytes."
            )

            BACKPRESSURE_THRESHOLD = 0.15 * BUFFER_SIZES[device_id]
            if self.buffers[device_id] >= BACKPRESSURE_THRESHOLD:
                restore_packet = {"id": "RESTORE", "size": 0, "target": device_id}
                self.broadcast(restore_packet, exclude=[device_id])
                self.logger.process(
                    f"Switch: Sent RESTORE signal for Device {device_id} as buffer usage is below the threshold."
                )
            elif self.buffers[device_id] < BACKPRESSURE_THRESHOLD:
                backpressure_packet = {"id": "BACKPRESSURE", "size": 0, "target": device_id}
                self.broadcast(backpressure_packet, exclude=[device_id])
                self.logger.process(
                    f"Switch: Continued backpressure for Device {device_id} due to high buffer utilization."
                )

```

تابع listen

وظیفه این تابع پردازش بسته‌های ارسالی می‌باشد. در هر چرخه، با استفاده از صف داده‌ای که تا الان دارد، ترتیبی را که بسته‌ها باید پردازش شوند، مشخص می‌کند. سپس برای هر بسته تابع process_packet را برای صدا می‌زند.

```

def listen(self):
    self.logger.info("Switch: Listening for incoming packets...")
    while self.running:
        packets_to_process = {}

        for device_id, q in self.incoming_queues.items():
            if not q.empty():
                packet = q.get()
                target_device = packet["target"]
                if target_device not in packets_to_process:
                    packets_to_process[target_device] = []
                packets_to_process[target_device].append((device_id, packet))

        if self.STATE == 1:
            pass
        else:
            for target_device, packets in packets_to_process.items():
                if self.PRIORITY_MODE == 1:
                    packets.sort(key=lambda pkt: pkt[1]["type"] == 2)
                elif self.PRIORITY_MODE == 2:
                    buffer_usage = self.buffers[target_device]
                    if buffer_usage < 0.10 * BUFFER_SIZES[target_device]:
                        packets.sort(key=lambda pkt: pkt[1]["type"] == 2)
                elif self.PRIORITY_MODE == 3:
                    with self.lock:
                        type1_packets = [pkt for pkt in packets if pkt[1]["type"] == "type1"]
                        type2_packets = [pkt for pkt in packets if pkt[1]["type"] == "type2"]

                        combined_packets = []
                        type1_index, type2_index = 0, 0

                        while type1_index < len(type1_packets) or type2_index < len(type2_packets):

```

```

                            while type1_index < len(type1_packets) or type2_index < len(type2_packets):
                                if type1_index < len(type1_packets):
                                    combined_packets.append(type1_packets[type1_index])
                                    type1_index += 1
                                if type1_index < len(type1_packets):
                                    combined_packets.append(type1_packets[type1_index])
                                    type1_index += 1
                                if type2_index < len(type2_packets):
                                    combined_packets.append(type2_packets[type2_index])
                                    type2_index += 1

                            packets[:] = combined_packets

            for target_device, packets in packets_to_process.items():
                for device_id, packet in packets:
                    self.process_packet(device_id, packet)

```

Controller.py

همان طور که در بخش اول توضیح داده شد این کلاس جزو منطق اصلی برنامه نیست و برای راه اندازی و مقداردهی های اولیه نوشته شده است. در نتیجه از توضیح جزئیات آن خودداری می کنیم و تنها به بیان نکات قابل توجه آن می پردازیم:

```
if __name__ == "__main__":

    get_simulation_duration()
    get_simulation_RATIO()

    simulation_logger.info("Starting simulation...")

    incoming_queues = {
        1: queue.Queue(),
        2: queue.Queue(),
        3: queue.Queue(),
        4: queue.Queue()
    }

    devices = [
        Device1(1, incoming_queues[1], memory_logger, RATIO, DURATION),
        Device2(2, incoming_queues[2], memory_logger, RATIO, DURATION),
        Device3(3, incoming_queues[3], memory_logger, RATIO, DURATION),
        Device4(4, incoming_queues[4], memory_logger, RATIO, DURATION)
    ]

    outgoing_queues = {
        1: devices[0].received_packets,
        2: devices[1].received_packets,
        3: devices[2].received_packets,
        4: devices[3].received_packets
    }

    switch = Switch(incoming_queues, outgoing_queues, simulation_logger, STATE, PRIORITY_OPTION)
```

در ابتدا چهار صف داده برای خروجی دستگاه ها ساخته می شود و در آرایه incoming_queues ذخیره می شود. توجه کنید که خروجی دستگاه ها، در واقع ورودی سوئیچ است. در مرحله بعدی، از هر دستگاه نمونه گرفته شده است. می توانستیم به جای استفاده از ۴ کلاس متفاوت برای دستگاه ها، فقط یک فایل device.py داشته باشیم و اینجا چهار بار از آن نمونه بگیریم. فرقی نمی کرد.

در مرحله بعدی، آرایه received_packets را از هر کدام از این دستگاه‌ها می‌گیریم. این‌ها خروجی سوئیچ هستند. در نتیجه در آرایه outgoing_queues قرار داده می‌شوند. سپس از کلاس سوئیچ یک نمونه گرفته و آن را با queue‌هایی که داریم مقداردهی می‌کنیم.

```
switch_thread = threading.Thread(target=switch.listen, name="SwitchListener")
buffer_thread = threading.Thread(target=switch.restore_buffers, name="BufferRestorer")

device_threads = []
for device in devices:
    device_thread_sender = threading.Thread(target=device.send_packets, name=f"Device{device.device_id}Sender")
    device_thread_processor = threading.Thread(target=device.process_incoming,
                                              name=f"Device{device.device_id}Processor")
    device_thread_alert = threading.Thread(target=device.check_alerts, name=f"Device{device.device_id}AlertHandler")
    device_threads.append(device_thread_sender)
    device_threads.append(device_thread_processor)
    device_threads.append(device_thread_alert)

switch_thread.start()
buffer_thread.start()
for thread in device_threads:
    thread.start()
```

در مرحله بعد، thread‌های لازم ساخته شده و اجرای آن‌ها شروع می‌شود. به دلیل زیاد بودن thread‌های مربوط به دستگاه‌ها، آن‌ها را در device_thread ذخیره کردیم.

```
time.sleep(DURATION)

stop_simulation(switch, devices, simulation_logger)

switch_thread.join()
buffer_thread.join()
for thread in device_threads:
    thread.join()

simulation_logger.info("Simulation completed.")
print("Simulation completed.")
```

بعد از سپری شدن زمان اجرا، کنترلر فلگ running را در بقیه کلاس‌ها False می‌کند و به اجرای آن‌ها خاتمه می‌دهد؛ با join مطمئن می‌شود که همه thread‌ها خاتمه یافته‌اند و بدین ترتیب اجرای برنامه تمام می‌شود.

سخن پایانی

در این پروژه، هدف اصلی طراحی و شبیه‌سازی سیستمی بود که بتواند جریان داده‌ها را با استفاده از کنترل مبتنی بر اعتبار (Credit-Based Flow Control) بهینه‌سازی نماید و از مشکلاتی نظیر سرریز (Overflow) در بافرها جلوگیری کند. با توجه به چالش‌های موجود در مدیریت منابع محدود سیستم و تضمین عملکرد پایدار، تمرکز ویژه‌ای بر روی کاهش تعداد دفعات وقوع سرریز و مدیریت هوشمندانه بافرها قرار گرفت.

در این راستا، مکانیزم‌های مختلفی برای اولویت‌بندی بسته‌ها و تخصیص بهینه منابع پیاده‌سازی شدند. از جمله این مکانیزم‌ها می‌توان به موارد زیر اشاره کرد:

- ✓ اولویت‌بندی بسته‌ها بر اساس نوع: اطمینان از ارسال بسته‌های حیاتی‌تر در اولویت بالا.
 - ✓ مدیریت حالت بافرها: شناسایی شرایط بحرانی بافرها و ارسال سیگنال‌هایی نظیر "Backpressure" و "Restore" برای تنظیم نرخ ارسال بسته‌ها.
 - ✓ مدل‌های مختلف کنترل جریان: استفاده از روش‌هایی نظیر پردازش ۲ بسته نوع ۱ در برابر ۱ بسته نوع ۲ (۲:۱) برای دستیابی به تعادل بهتر میان انواع مختلف داده‌ها.
- این پیاده‌سازی‌ها به طور چشمگیری باعث کاهش تعداد دفعات سرریز بافرها و بهبود پایداری سیستم شدند. با استفاده از نظارت لحظه‌ای (Logging) و الگوریتم‌های بازخوردی، توانستیم نرخ ارسال داده‌ها را به صورت پویا تنظیم کنیم تا سیستم تحت بار بالا نیز به صورت پایدار عمل کند.
- یکی از دستاوردهای مهم این پروژه، شناسایی و درک بهتر از نحوه برخورد با محدودیت‌های منابع در سیستم‌های پیچیده بود. همچنین، طراحی سیستمی که بتواند بدون نیاز به توقف کامل یا از دست رفتن داده‌ها، خود را در برابر شرایط بحرانی تنظیم کند، از دیگر جنبه‌های این پروژه محسوب می‌شود.

برای مشاهده روند توسعه و پیشرفت الگوریتم های استفاده شده و منطق کلی، می توانید به مخزن گیت هاب پروژه مراجعه کنید:

<https://github.com/NikaGhaderi/Credit-Based-Flow-Control-Simulation.git>

پایان