



تمرین کامپیوتری شماره 3



عنوان: پیاده سازی شبیه ساز شبکه برای بررسی الگوریتم های مسیریابی

درس: شبکه های کامپیوتری

استاد راهنما: دکتر ناصر یزدانی^۱

رشته: مهندسی کامپیوتر

دستیاران آموزشی: سهراب مرادی^۲، عرفان احمدی^۳ و هستی کریمی^۴

نیمسال دوم سال تحصیلی 1402-03

^۱ نشانی پست الکترونیکی: yazdani@ut.ac.ir

^۲ نشانی پست الکترونیکی: m.moradi1998@ut.ac.ir

^۳ نشانی پست الکترونیکی: erfah.ahmadi@ut.ac.ir

^۴ نشانی پست الکترونیکی: hasti.karimi@ut.ac.ir

عنوان پروژه

پیاده سازی شبیه ساز شبکه برای بررسی الگوریتم های مسیریابی

هدف

در این تمرین قصد داریم با استفاده از فریمورک Qt به پیاده سازی یک شبیه سازی رویداد محور (Event Driven) بپردازیم تا با استفاده از آن پاسخگویی الگوریتم های مسیریابی را در شرایط مختلف بررسی کنیم، همچنین با ساختار روترها و فرآیند های موجود در آنها، برخی از توپولوژی های شبکه و توزیع های آماری تولید داده در دنیای واقعی آشنا خواهید شد.

0- ابزار

این تمرین تماماً با استفاده از زبان برنامه نویسی C++ پیاده سازی خواهد شد لذا از کتابخانه ها و فریمورک های مطرح و محبوب این زبان استفاده خواهیم کرد. ابزار مورد نیاز به شرح زیر است:

- Qt framework

1- مقدمه

- اهمیت الگوریتم های مسیریابی

الگوریتم های مسیریابی سنگ بنای ارتباطات شبکه هستند که کارآمدترین مسیر را برای انتقال بسته های داده در سراسر شبکه از مبدأ به مقصد تعیین می کنند. اهمیت این الگوریتم ها در توانایی آنها برای بهینه سازی عملکرد شبکه با کاهش تأخیر، متعادل کردن بارهای ترافیک و جلوگیری از ازدست دادن بسته ها نهفته است. آنها برای حصول اطمینان از اینکه شبکه می تواند مقیاس رو به رشدی از کاربران و دستگاه ها را بدون کاهش کیفیت خدمات در خود جای دهد، ضروری هستند.

- اهمیت توپولوژی

توپولوژی ستون فقرات (backbone) یک شبکه به زیرساخت مرکزی اشاره دارد که بخش های مختلف شبکه را مانند ستون فقرات در بدن انسان به هم متصل می کند و انتخاب توپولوژی مناسب برای زیرساخت شبکه بسیار مهم است زیرا بخش عمده ای از ترافیک داده را مدیریت می کند و به عنوان مسیر اصلی برای ارتباطات بین شبکه ای عمل می کند. توپولوژی زیر ساخت با طراحی خوب، انتقال داده با سرعت بالا، قابلیت اطمینان و تحمل خطا را تضمین می کند. همچنین مدیریت شبکه را ساده می کند و با افزایش تقاضای شبکه امکان گسترش آسان تر را فراهم می کند.

- شبیه سازی شبکه های تحت بار سنگین

شبیه سازی شبکه یک عمل حیاتی برای نظارت بر الگوریتم های مسیریابی، به ویژه در شرایط بار سنگین شبکه است. با ایجاد یک مدل مجازی از یک شبکه، مهندسان شبکه می توانند رفتار الگوریتم های مسیریابی را بدون ریسک و هزینه آزمایش بر روی یک شبکه واقعی تجزیه و تحلیل کنند. شبیه سازی امکان مشاهده نحوه عملکرد الگوریتم ها در شرایط استرس (ترافیک بالا)، شناسایی تنگناهای بالقوه و ارزیابی استراتژی های مسیریابی مختلف را فراهم می کند. این عمل برای توسعه شبکه های قوی که می توانند عملکرد و قابلیت اطمینان را حتی در صورت مواجهه با حجم ترافیک بالا حفظ کنند، ضروری است.

- **توزیع‌های آماری در شبیه‌سازی شبکه**

توزیع‌های آماری نقش محوری در شبیه‌سازی شبکه ایفا می‌کنند زیرا به مدل‌سازی تصادفی و متغیر بودن ترافیک شبکه کمک می‌کنند. انواع مختلف ترافیک، مانند مرور وب، پخش ویدئو، یا تماس‌های VoIP، دارای الگوهای متمایزی هستند که می‌توانند با استفاده از توزیع‌های آماری مناسب نمایش داده شوند. برای مثال، زمان‌های بین ورود و اندازه بسته‌ها را می‌توان با استفاده از توزیع‌های نمایی یا پارتو مدل‌سازی کرد تا ماهیت انفجاری ترافیک شبکه‌های واقعی را منعکس کند. دخیل کردن توزیع‌های آماری در شبیه‌سازی شبکه اجازه می‌دهد تا نمایش دقیق‌تری از رفتار شبکه در شرایط بار مختلف ارائه شود.

- **ویژگی‌های روتر در شبیه‌سازی شبکه**

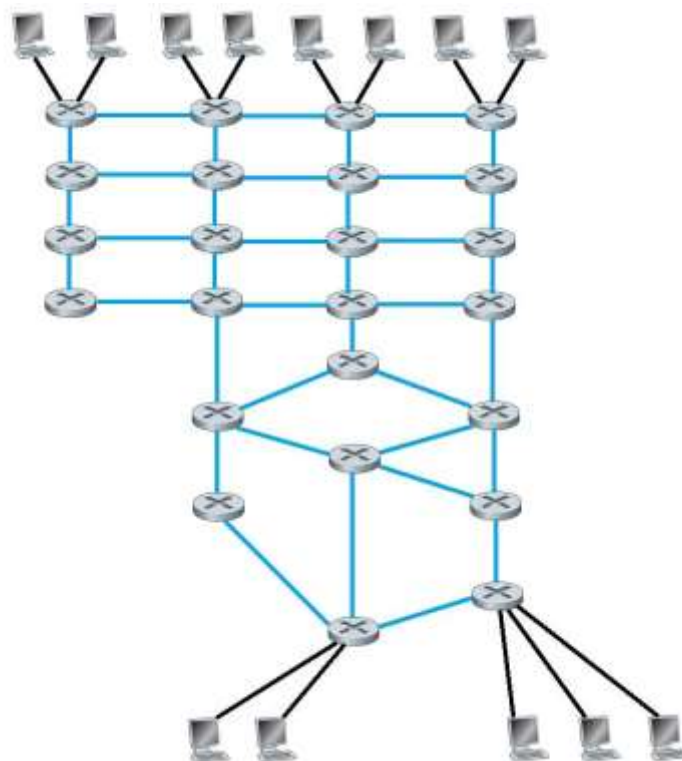
ویژگی‌های روتر در شبیه‌سازی شبکه نیز به همانند توزیع‌های آماری مهم هستند. ویژگی‌هایی مانند اندازه بافر، سیاست‌های مدیریت صف و پروتکل‌های مسیریابی (مانند OSPF یا RIP) مستقیماً بر نحوه مدیریت بسته‌ها توسط روترها تأثیر می‌گذارند، مخصوصاً تحت بار سنگین. شبیه‌سازی روترها با پیکربندی‌های واقعی، تجزیه و تحلیل عملکرد شبکه، شناسایی تنگناهای بالقوه و آزمایش الگوریتم‌ها یا سیاست‌های مسیریابی جدید را ممکن می‌سازد.

- **اهمیت در شبیه‌سازی**

ترکیب توزیع‌های آماری و ویژگی‌های روتر در شبیه‌سازی شبکه به چند دلیل حیاتی است:

- 1) تضمین می‌کند که شبیه‌سازی رفتار شبکه‌های واقعی را از نزدیک منعکس می‌کند.
- 2) به محققان و دانشجویان اجازه می‌دهد تا تأثیر الگوهای مختلف ترافیک را بر عملکرد شبکه مشاهده کنند.
- 3) به ارزیابی اثربخشی الگوریتم‌های مسیریابی و پیکربندی‌های مسیریاب تحت شرایط استرس، مانند زمان اوج استفاده یا حملات DDoS کمک می‌کند.
- 4) با در نظر گرفتن این عوامل، شبیه‌سازی‌ها می‌توانند بینش‌های ارزشمندی را در مورد طراحی و مدیریت شبکه‌های قوی و کارآمد ارائه دهند.

2- توضیح تمرین



شکل 1 - ساختار شبکه مورد بحث تمرین

حل این تمرین نیازمند پیاده سازی 5 بخش مختلف است:

1- **سیستم کنترل رویداد:** این سیستم باید در **بازه های زمانی قابل تنظیم** (از خارج برنامه، به عنوان مثال هر 10 ثانیه) تولید سیگنال کند (سیگنالی رو در سطح برنامه emit کند)، از این سیگنال برای هماهنگی بخش های مختلف برنامه استفاده می شود. تولید های داده بر اساس سیگنال های تولید شده از این ماژول کنترل خواهند شد. همچنین این ماژول کل مدت زمان شبیه سازی را نیز مدیریت خواهد کرد.

2- **سیستم تولید داده:** این سیستم در هر سیکل (رویداد تولید شده در سیستم کنترل رویداد) اقدام به تولید داده و ارسال آن بر بستر شبکه می کند. **تولید داده توسط این سیستم باید بر بستر یک توزیع آماری خاص انجام شود که در ادامه توضیح داده شده اند**، همچنین این موضوع که هر گروه باید از کدام توزیع آماری استفاده کند در ادامه صورت مسئله ی تمرین ذکر شده است. این سیستم دارای زیر بخش هایی است که نقش کاربر در اینترنت را بازی میکنند، این زیر بخش ها به **gateway** ها متصل هستند و بسته های داده را به سمت دیگر کاربران موجود در شبکه ارسال میکنند (**اینکه کدام بسته به سمت کدام کاربر ارسال شود می تواند به صورت رندوم صورت گیرد اما باید توجه شود که بسته ای از شخص A به سمت خودش و باقی کاربران موجود در همین subnet نباید ارسال شود**).

3- روتر: موجودیت روتر که عمل مسیریابی در شبکه را انجام میدهد، روترها دارای ویژگی‌های زیر خواهند بود:

1- جدول مسیریابی (بر اساس نوع الگوریتم مورد استفاده). اجباری

2- پورت‌ها. اجباری

3- ورژن IP. اجباری

4- بافر با اندازه‌ی نامحدود. اجباری

5- بافر با اندازه‌ی محدود. امتیازی

6- DHCP امتیازی

7- داینامیک روتینگ پروتکل. امتیازی

تمامی ویژگی‌های ذکر شده در بخش تعاریف تمرین توضیح داده شده‌اند. بدیهی است که روترها حتماً باید دارای IP باشند!

4- موجودیت Cluster: در پیاده‌سازی backbone شبکه، ما از دو cluster استفاده می‌کنیم تا بتوانیم الگوریتم‌های مسیریابی خارج از

شبکه‌ی محلی را نیز پوشش بدهیم، هر کلاستر شامل تعدادی روتر است (تعداد روترها جهت پیاده‌سازی در شکل 1 مشخص شده است) که این روترها در قالب یک توپولوژی خاص به یکدیگر متصل شده‌اند، همچنین روترهای مشخصی نیز باید به عنوان درگاه این کلاستر معرفی شوند. یکی از کلاسترها از توپولوژی Mesh و دیگری از توپولوژی Ring-Star (یک توپولوژی ابداعی، ترکیبی از توپولوژی Ring و Star) استفاده خواهد کرد (توضیحات توپولوژی‌ها + شکل در بخش تعاریف تمرین وجود دارند). برای ساده‌سازی شرایط تمرین، طول لینک‌ها در ارتباطات میان روترها (چه داخلی و چه خارجی) یکسان در نظر گرفته خواهد شد. لازم است تا شبکه‌ی Mesh شامل 16 روتر و شبکه‌ی Ring-Star شامل 8 روتر باشد.

5- موجودیت Packet: هر بسته باید تمام گام‌های طی شده در شبکه از مبدا تا مقصد (IP روترها)، تعداد سیکل‌های انتظار در بافر و به

طور کلی تعداد سیکل‌هایی که طول کشیده تا از مبدا به مقصد برسد را در خود نگه دارد تا بتوانیم در نهایت مقایسه‌ی خوبی بین الگوریتم‌های مختلف انجام بدهیم. همچنین packet‌ها در هنگام انتقال از روتری با IP Version متفاوت باید مجدداً بسته‌بندی شود. (به عنوان مثال بسته در هنگام انتقال از یک روتر IPv6 به روتر IPv4 باید با ساختار IPv4 بسته‌بندی شود).

در انتهای تمرین، بررسی خواهد شد که کدام الگوریتم در نقطه‌ی اوج بار شبکه عملکرد بهتری را از خود به جای گذاشته است. فاکتورهای مورد بررسی عبارتند از:

1- مقایسه‌ی تعداد گام‌ها به صورت میانگین کل مدت زمان و اوج بار بین دو الگوریتم مسیریابی.

2- مقایسه‌ی مجموع کل، میانگین و کمینه‌ی بیشینه‌ی انتظار در بافرهای روترها.

3- کدام الگوریتم از فضای گسترده‌تری از شبکه استفاده کرده است (بر اساس مسیر حرکت بسته بین روترها).

نحوه‌ی تخصیص توزیع آماری بدین شرح است:

1- گروه‌هایی که مجموع رقم انتهایی شماره دانشجویی آنها زوج است از توزیع پواسون برای تولید داده استفاده کنند

2- گروه‌هایی که مجموع رقم انتهایی شماره دانشجویی آنها فرد است از توزیع پارتو برای تولید داده استفاده کنند.

3- برنامه نویسی و نکات پیاده سازی

کد شما باید ساختارمند و تماماً ماژولار باشد، برای پیاده سازی سیستم کنترل رویداد از **Signal-Slot System** در **Qt** استفاده کنید. لازم هست تا تمامی کلاس های شما از **QObject** ارث بری کنند. برای شبیه سازی عملکرد موازی سیستم (بسته ها نباید به ترتیب عملیات انجام دهند، زیرا در این صورت هیچگاه کارکرد واقعی شبکه شبیه سازی نخواهد شد) می توانید هر روتر را به **Thread** مخصوص به خودش اجرا کنید، برای این منظور باید از تابع **moveToThread** در **QObject** استفاده کنید، توجه داشته باشید که تمامی توابعی که برای کلاس روتر پیاده سازی میکنید باید **Reentrancy** داشته باشند و برخی توابع نیز نیاز است تا **Thread-Safe** باشند، همچنین امکان استفاده از **Qt Concurrent** و دیگر ابزار های **Qt** که کمک به شبیه سازی حالت موازی می کنند مجاز است. برای ساده سازی تمرین، استفاده از تمامی کتابخانه های فریمورک **Qt** مجاز است و میتوانید از هر کتابخانه ای که فرآیند حل تمرین را برای شما ساده تر کند استفاده کنید.

- 1- برنامه ی شما باید به ازای هر روتر یک **Thread** جدید ایجاد کند.
- 2- سیستم تولید داده و سیستم کنترل رویداد بهتر است که هرکدام در یک **Thread** باشند.
- 3- در پیاده سازی شبیه ساز های شبکه مبحث **Memory Management** بسیار مهم است، برای انتقال بسته ها میان روتر ها، یک بسته در سیستم تولید داده تولید شود و یک **Pointer** از آن بین روتر ها و ترد های مختلف در جریان باشد (تا از کپی ها ناخواسته جلوگیری شود)، از آنجا که هر بسته در یک لحظه فقط در یک روتر قرار دارد بنابراین مشکلی از بابت **Concurrency Problems** نخواهد داشت (حداقل در این تمرین مشکلی به وجود نخواهد آمد). پیشنهاد میشود تا از **std::unique_ptr** استفاده کنید.
- 4- پیشنهاد میشود تا روابط **Object Trees & Ownership** میان **QObject** ها مطالعه شود تا به مدیریت بهتر حافظه کمک کند.
- 5- برای پیاده سازی راحل تر توپولوژی ها، میتوانید از روی شکل 1 روتر ها را شماره گذاری کنید و با یک شمارنده ی **static** به هر نود یک **id** اختصاص داده و از روی شماره گذاری خودتان مشخص کنید که هر روتر به کدام روتر های دیگر متصل است.
- 6- نیاز است تا ویژگی های روتر دقیقاً همانند آنچه در بخش تعاریف آمده است پیاده شود، برای ساده سازی تمرین ویژگی ها در ساده ترین سطح شان تعریف شده اند.

4- تعاریف

• الگوریتم Dijkstra

Initialization ○

هزینه 0 را به گره شروع اختصاص دهید و برای تمامی گره های دیگر بی نهایت را تعیین کنید.

Main Loop ○

تمامی گره ها را به عنوان گره های بازدید نشده علامت گذاری کنید. تا زمانی که گره های بازدید نشده وجود داشته باشند، گره بازدید نشده با کمترین هزینه را به عنوان گره فعلی انتخاب کنید. برای هر همسایه گره فعلی، هزینه کل برای رسیدن به همسایه از طریق گره فعلی را محاسبه کنید. اگر این هزینه کمتر از هزینه فعلی همسایه باشد، هزینه همسایه را با این هزینه کمتر به روز کنید.

- Termination

زمانی که تمامی گره‌ها بازدید شده‌اند یا اگر گره مقصد بازدید شده باشد، الگوریتم پایان می‌یابد.

- Path Reconstruction

پس از اجرای الگوریتم، می‌توان کوتاه‌ترین مسیر از گره شروع به هر گره دیگری را با بازگشت از گره مقصد به گره شروع با استفاده از گره‌های قبلی ثبت شده بازسازی کرد.

- الگوریتم Bellman-Ford

- Initialization

هزینه 0 را به گره مبدأ و بینهایت به تمامی گره‌های دیگر اختصاص دهید. آرایه هزینه را برای پیگیری برآوردهای کوتاه‌ترین فاصله فعلی از گره مبدأ به هر گره مقصد مقداردهی اولیه کنید.

- Relaxation

این مرحله را برای تمامی یالها (v, u) در گراف به تعداد $|V| - 1$ بار تکرار کنید (در اینجا $|V|$ تعداد رئوس در گراف است) برای هر یال (v, u) ، اگر هزینه رسیدن به v از طریق u کمتر از برآورد هزینه فعلی برای v باشد، برآورد هزینه برای v را به‌روز کنید.

- Path Reconstruction

پس از $|V| - 1$ بار تکرار، کوتاه‌ترین مسیر از گره مبدأ به هر گره دیگری با بازگشت از گره مقصد به گره مبدأ با استفاده از گره‌های قبلی ثبت‌شده بازسازی می‌شود.

- الگوریتم iBGP

- روابط همسایگی

در iBGP، روترها داخل یک AS یکسان روابط همسایگی با یکدیگر برقرار می‌کنند تا اطلاعات مسیریابی را تبادل کنند. این روابط همسایگی معمولاً با استفاده از اتصالات TCP بین روترهای BGP برقرار می‌شوند.

- انتشار مسیر

هنگامی که یک روتر یک مسیر جدید یا مسیر موجود را به‌روز می‌کند، این اطلاعات را به همسایگان iBGP خود منتشر می‌کند. لذا این امکان را فراهم می‌کند که تمام روترها در داخل AS جداول مسیریابی متناسبی داشته باشند.

- انعکاس مسیر

در شبکه‌های بزرگ، اغلب از انعکاس مسیر iBGP برای کاهش تعداد نشست‌های مجاورت iBGP مورد نیاز استفاده می‌شود. انعکاس‌دهنده‌های مسیر به مقیاس‌پذیری شبکه‌های iBGP با اجازه دادن به روترها برای انعکاس مسیرها به روترهای دیگر بدون نیاز به یک شبکه کامل از اتصالات iBGP کمک می‌کنند.

- انتخاب مسیر

روترهای iBGP از الگوریتم انتخاب مسیر BGP برای تعیین بهترین مسیر به یک مقصد بر اساس ویژگی‌های مختلفی مانند طول مسیر AS، ترجیح محلی و کد منشأ استفاده می‌کنند.
- جلوگیری از حلقه

روترهای iBGP باید اطمینان حاصل کنند که حلقه‌های مسیریابی رخ ندهند. برای جلوگیری از حلقه‌های مسیریابی، روترهای iBGP مسیرهای یادگرفته شده از یک همسایه iBGP را به همسایه iBGP دیگری اعلام نمی‌کنند.
- پایداری و همگرایی

الگوریتم iBGP برای ارائه مسیریابی پایدار و قابل اعتماد در یک AS طراحی شده است. این کمک میکند تا اطمینان حاصل شود که تمام روترها در شبکه اطلاعات مسیریابی متناسبی دارند و تغییرات در توپولوژی شبکه به سرعت به تمام روترها منتقل میشوند.
- الگوریتم eBGP
 - روابط همسایگی

روترهای eBGP ارتباطات همسایگی را با روترها در ASهای مختلف برقرار می‌کنند تا اطلاعات مسیریابی را تبادل کنند. این ارتباطات همسایگی به طور معمول با استفاده از اتصالات TCP بین روترهای BGP برقرار می‌شوند.
 - تبلیغ مسیر

هنگامی که یک روتر eBGP مسیر جدیدی را یاد می‌گیرد یا مسیر موجود را به‌روز می‌کند، این اطلاعات را به همسایگان eBGP خود در ASهای دیگر اعلام می‌کند. این ویژگی امکان را برای گسترش اطلاعات مسیریابی بین ASها فراهم می‌کند.
 - مسیر AS

یکی از ویژگی‌های کلیدی در مسیریابی eBGP مسیر AS است. مسیر AS یک دنباله از شماره‌های AS است که یک مسیر برای رسیدن به مقصد طی کرده است. از آن برای جلوگیری از حلقه‌های مسیره‌ای استفاده می‌شود و اطلاعاتی در مورد مسیری که توسط مسیر طی شده است ارائه می‌دهد.
 - انتخاب مسیر

روترهای eBGP از الگوریتم انتخاب مسیر BGP برای تعیین بهترین مسیر به یک مقصد بر اساس ویژگی‌هایی مانند طول مسیر AS، اولویت محلی و کد منشأ استفاده می‌کنند. روترهای eBGP مسیرهای با مسیر AS کوتاه‌تر را ترجیح می‌دهند.
 - جلوگیری از حلقه

روترهای eBGP با عدم اعلام مسیرهای یادگرفته شده از یک همسایه eBGP به همسایه eBGP دیگر، از ایجاد حلقه‌های مسیره‌ای جلوگیری می‌کنند. این اطمینان حاصل می‌کند که حلقه‌های مسیریابی بین ASهای مختلف رخ ندهد.

○ پایداری و اتصال

الگوریتم eBGP برای اطمینان از مسیریابی پایدار و قابل اعتماد بین دامنه‌ها در اینترنت بسیار حیاتی است. eBGP امکان را برای تبادل اطلاعات مسیریابی بین ASهای مختلف و حفظ اتصال بین شبکه‌ها فراهم می‌کند.

● تعریف توزیع پواسون

توزیع پواسون یک توزیع احتمال است که تعداد وقایعی را که در یک بازه زمانی یا فضای ثابت روی می‌دهند، با فرض نرخ ثابت و وقایعی که از هم مستقل هستند، توصیف می‌کند.

○ کاربرد در شبکه‌های کامپیوتری

در شبکه‌های کامپیوتری، توزیع پواسون می‌تواند برای مدل‌سازی وقایع تصادفی مانند ورود بسته‌ها، خطاها در انتقال داده یا درخواست‌ها به یک سرور استفاده شود.

○ تعیین پارامتر

برای استفاده از توزیع پواسون، باید میانگین نرخ وقوع وقایع در شبکه را تعیین کنید. این پارامتر با نماد λ نشان داده می‌شود و میانگین تعداد وقایع در یک بازه مشخص است.

○ محاسبه احتمالات

پس از داشتن پارامتر λ ، می‌توانید احتمال وقوع یک تعداد خاص از وقایع در یک بازه زمانی مشخص را با استفاده از تابع چگالی احتمال پواسون محاسبه کنید:

$$P(X = k) = (e^{-\lambda} * \lambda^k) / k!$$

که:

– مقدار $P(X = k)$ احتمال وقوع k واقعه است

– مقدار e ، پایه لگاریتم طبیعی است.

– مقدار λ ، میانگین نرخ وقوع وقایع است.

– مقدار k ، تعداد وقایع است.

– مقدار $k!$ ، فاکتوریل k را نشان می‌دهد.

○ تفسیر نتایج

با استفاده از توزیع پواسون، می‌توانید احتمال وقوع اعداد مختلف از وقایع در شبکه در یک بازه زمانی خاص را تحلیل و تفسیر کنید. این اطلاعات می‌تواند در برنامه‌ریزی ظرفیت، بهینه‌سازی شبکه و پیش‌بینی رفتار شبکه مفید باشد.

○ مثال

به عنوان مثال، می‌توانید از توزیع پواسون برای تخمین احتمال ورود یک تعداد خاص از بسته‌ها به یک روتر در یک بازه زمانی خاص بر اساس نرخ میانگین ورود λ استفاده کنید.

○ تأیید و تنظیمات

اهمیت دارد که فرضیات توزیع پواسون را در زمینه داده‌های خاص شبکه که در حال تحلیل هستید، تأیید کنید. اگر داده‌ها با توزیع پواسون خوب همخوانی نداشته باشند، ممکن است نیاز به تنظیمات یا مدل‌های جایگزین برای یک تحلیل دقیق‌تر باشد.

● تعریف توزیع پارتو

توزیع پارتو یک توزیع احتمال قدرتی است که توزیع ثروت یا درآمد را در اقتصاد توصیف می‌کند. در زمینه شبکه‌های کامپیوتری، توزیع پارتو می‌تواند برای مدل‌سازی توزیع اندازه بسته‌ها، اندازه فایل‌ها یا ترافیک شبکه استفاده شود.

○ کاربرد در شبکه‌های کامپیوتری

در شبکه‌های کامپیوتری، توزیع پارتو به طور معمول برای نمایش توزیع ناهموار اندازه انتقال داده‌ها یا حجم‌های ترافیک استفاده می‌شود. این توزیع فرض می‌کند که تعداد کمی از عناصر موجب بیشترین حجم کلی می‌شوند.

○ تعیین پارامترها

توزیع پارتو توسط دو پارامتر تعیین می‌شود: α (پارامتر شکل) و x_m (پارامتر مقیاس). این پارامترها شکل و مقیاس توزیع را تعیین می‌کنند.

○ محاسبه احتمالات

تابع چگالی احتمال (PDF) توزیع پارتو به صورت زیر است:

$$f(x; \alpha, x_m) = (\alpha * x_m^\alpha) / x^{(\alpha+1)}$$

که:

– مقدار $f(x; \alpha, x_m)$ تابع چگالی احتمال توزیع پارتو در x است.

– مقدار α پارامتر شکل است.

– مقدار x_m پارامتر مقیاس است.

– مقدار x متغیر است.

○ تفسیر نتایج

با استفاده از توزیع پارتو، ادمین‌های شبکه می‌توانند توزیع اندازه‌ها یا حجم‌های ترافیک داده را تجزیه و تحلیل کرده و درک کنند. این اطلاعات می‌تواند برای برنامه‌ریزی ظرفیت، تخصیص منابع و بهینه‌سازی شبکه مفید باشد.

○ مثال

به عنوان مثال، توزیع پارتو می‌تواند برای مدل‌سازی توزیع اندازه فایل‌هایی که در یک شبکه منتقل می‌شوند، استفاده شود و به تعیین اندازه فایل‌هایی که بیشترین میزان ترافیک شبکه را تشکیل می‌دهند کمک کند.

○ تأیید و تنظیمات

اهمیت دارد که اعتبارسنجی کنید که توزیع پارتو به درستی داده‌ها را در زمینه خاص شبکه کامپیوتری نشان می‌دهد یا خیر. تنظیمات به پارامترها یا در نظر گرفتن توزیع‌های جایگزین ممکن است برای یک تجزیه و تحلیل دقیق‌تر لازم باشد.

ویژگی‌های روتر:

جدول روتینگ

جدول روتینگ یکی از مؤلفه‌های اصلی در هر روتر است که به مدیریت ترافیک شبکه کمک می‌کند. جدول روتینگ داده‌ها و اطلاعاتی را در خود ذخیره می‌کند که برای تعیین بهترین مناسب ترین مسیر برای ارسال بسته‌های داده به مقصد خاصی استفاده می‌شود. در این جدول، اطلاعاتی که وجود دارد شامل موارد زیر است:

1. **آدرس شبکه مقصد (Destination Network Address)** این آدرس نشان‌دهنده شبکه‌ای است که بسته‌ها باید به آن فرستاده شوند. معمولاً به صورت آدرس IP و ماسک شبکه نمایش داده می‌شود.
2. **ماسک شبکه (Subnet Mask)** این ماسک کمک می‌کند تا روتر بتواند تشخیص دهد که کدام بخش از آدرس IP مربوط به شبکه است و کدام بخش مربوط به میزبان داخل آن شبکه.
3. **دروازه (Gateway)** دروازه یا همان Next Hop، آدرسی است که بسته‌ها باید ابتدا به آن فرستاده شوند تا از آنجا به مقصد نهایی هدایت شوند. این آدرس معمولاً آدرس IP روتری است که بسته‌ها باید ابتدا به آن بروند.
4. **متریک (Metric)** متریک یک عدد است که برای تعیین "هزینه" یا "فاصله" استفاده می‌شود تا بهترین مسیر به مقصد تعیین شود.
5. **پروتکل (Protocol)** نشان می‌دهد که این مسیر از طریق کدام پروتکل مسیریابی در جدول روتینگ قرار گرفته است. این می‌تواند شامل پروتکل‌های مسیریابی مانند RIP، OSPF یا BGP باشد.

زمانی که یک روتر بسته‌ای را دریافت می‌کند، ابتدا آدرس مقصد بسته را بررسی می‌کند و سپس با مقایسه آن با موارد ذخیره‌شده در جدول روتینگ، بهترین مسیر را برای ارسال بسته تعیین می‌کند. این فرآیند شامل بررسی تمام مسیرهای ممکن و انتخاب مسیری است که کمترین متریک را دارد یا بر اساس سیاست‌های تعیین‌شده دیگر انتخاب می‌شود.

پورت‌ها

پورت‌های یک روتر نقطه ورود و خروج ارتباطات در شبکه است. این پورت‌ها برای اتصال به دستگاه‌های دیگر یا به شبکه خارجی استفاده می‌شوند و وظیفه ارسال و دریافت داده‌ها را بر عهده دارند.

برای این تمرین نیاز است تا پورت‌ها شماره گذاری شوند و هر پورت بافر مخصوص به خود را داشته باشد. (در واقع روتر قبلی بداند که قرار است بسته را به کدام بافر در روتر مقصد push کند)

DHCP

Dynamic Host Configuration Protocol (DHCP) یک پروتکل شبکه است که استفاده می‌شود تا به دستگاه‌هایی که به یک شبکه متصل شده‌اند، به طور خودکار آدرس IP و تنظیمات دیگر شبکه از جمله ماسک زیرشبکه، آدرس دروازه، و آدرس DNS را اختصاص دهد.

یکی از وظایف اصلی DHCP در روتر این است که به دستگاه‌هایی که به شبکه متصل شده‌اند، به طور خودکار آدرس IP اختصاص دهد. این امر از دستکاری دستی و مداوم در تنظیمات آدرس IP برای هر دستگاه جلوگیری می‌کند و به مدیران شبکه اجازه می‌دهد تا شبکه را به راحتی مدیریت کنند.

Dynamic Routing Protocols

ویژگی Dynamic Routing Protocols به روتر این قابلیت را می‌دهد که به طور خودکار و پویا اطلاعات مسیریابی را در شبکه به اشتراک بگذارد و به تغییرات در توپولوژی شبکه و شرایط مسیریابی پاسخ دهد. ویژگی Dynamic Routing Protocols به شبکه‌ها این امکان را می‌دهد که به طور اتوماتیک و به روز شونده باشند و از طریق همکاری میان روترها به بهینه‌سازی مسیرهای شبکه بپردازند.

IPv6

IPv6 در روترها به طریق مشابهی با IPv4 کار می‌کند اما با تفاوت‌هایی در ساختار آدرس‌دهی و پردازش داده‌ها. در ادامه، نحوه کارکرد IPv6 در روترها به شرح زیر است:

1. آدرس‌دهی IPv6

- در IPv6، آدرس‌ها به صورت 128 بیتی نمایش داده می‌شوند که باعث افزایش انتقال پذیری آدرس‌ها و افزایش تعداد دستگاه‌ها در شبکه می‌شود. روترها مسئول تخصیص آدرس‌های IPv6 به دستگاه‌های متصل به شبکه هستند و از پروتکل‌هایی مانند DHCPv6 استفاده می‌کنند تا آدرس‌ها را به دستگاه‌ها اختصاص دهند.

2. پردازش بسته‌های IPv6

- بسته‌های داده IPv6 شامل سرآیندهایی هستند که شامل آدرس‌های IPv6 مبدأ و مقصد می‌شوند. روترها به طور مشابهی با IPv4، از اطلاعات در سرآیند برای تصمیم‌گیری در مورد مسیریابی بسته‌ها استفاده می‌کنند و بر اساس جدول مسیریابی خود، بسته‌ها را به مقصد مورد نظر هدایت می‌کنند.

درباره نحوه **encapsulation** در کتاب توضیحات لازمه آورده شده است.

بافر

ویژگی‌های بافر در روترها عمدتاً شامل موارد زیر می‌شود:

1. اندازه بافر: (Buffer Size)

- این ویژگی مشخص می‌کند که چه تعداد داده را می‌توان در بافر ذخیره کرد. اندازه بافر بر اساس نیازهای شبکه و ظرفیت دستگاه تنظیم می‌شود.

2. نحوه مدیریت بافر: (Buffer Management)

- این ویژگی مشخص می‌کند که چگونه داده‌ها در بافر مدیریت می‌شوند. این شامل روش‌های مختلفی از جمله FIFO (First In, First Out), Tail Drop, WRED (Weighted Random Early Detection), RED (Random Early Detection), و First Out می‌شود که هر کدام ویژگی‌ها و عملکرد خاص خود را دارند.

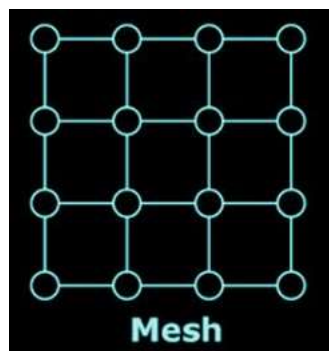
3. نحوه نگهداری داده‌ها: (Data Retention)

- این ویژگی مشخص می‌کند که چگونه داده‌ها در بافر نگهداری می‌شوند و چقدر مدت زمانی در بافر باقی می‌مانند. برخی از سیاست‌های نگهداری می‌توانند شامل FIFO یا LIFO (Last In, First Out) باشند.

توپولوژی‌های شبکه:

1. توپولوژی مش: (Mesh Topology)

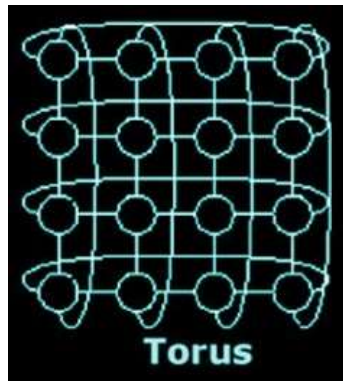
در توپولوژی مش، هر دستگاه به دستگاه‌های مجاور خود در راستای محورهای مختصات وصل می‌باشد. به شکل زیر:



شکل 2 - توپولوژی مش

2. توپولوژی تورس: (Torus Topology)

این توپولوژی الهام گرفته از توپولوژی مش می باشد با این تفاوت که رئوسی که در ابتدا و انتهای هر سطر و ستون قرار دارند، به هم دیگر وصل هستند. به طور ساده، می توان توپولوژی تورس را به صورت یک شبکه مربعی دوبعدی تصور کرد که مرزهای آن به یکدیگر متصل هستند.



شکل 3 - توپولوژی تورس

3. توپولوژی رینگ استار: (Ring Star Topology)

در توپولوژی رینگ، دستگاه ها به صورت دایره ای مرتبط هستند، به طوری که هر دستگاه با دو دستگاه مجاور خود ارتباط دارد.



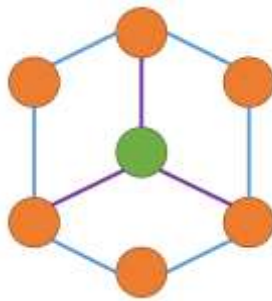
شکل 4 - توپولوژی رینگ

در توپولوژی استار، یک دستگاه مرکزی وجود دارد که با تمامی دستگاه های دیگر به صورت مستقیم متصل است. دستگاه های دیگر فقط به دستگاه مرکزی متصل هستند و نه به یکدیگر.



شکل 5 - توپولوژی استار

حال در توپولوژی رینگ استار، ترکیب این دو توپولوژی را استفاده میکنیم. برای مثال شکل زیر را در نظر بگیرید:



شکل 6 - توپولوژی رینگ استار

یک شبکه متشکل از تعدادی نود با توپولوژی رینگ میسازیم و در مرکز این رینگ یک نود دیگر قرار میدهیم، حال نود وسط به صورت یکی در میان به نود های موجود در رینگ متصل میشود.

5- بخش های امتیازی

- 1- موارد 5 و 6 و 7 از ویژگی های روتر امتیازی هستند.
- 2- به جای پیاده سازی backbone ای که شامل cluster 2 است میتوانید از cluster 3 استفاده کنید، کلاستر سوم از باید از توپولوژی تورس استفاده کند. این کلاستر جدید باید حداقل شامل 6 نود باشد.

جمع بندی و نکات پایانی

- مهلت تحویل: **1403/03/14**
- پروژه در گروه‌های **2 نفره** انجام می‌شود. (گروه بندی در سامانه ایلرن نیز انجام می‌شود و تحویل تمرین به صورت گروهی خواهد بود)
- هر ۲ نفر می‌بایست کار را تقسیم کنند. همچنین از **Git** برای ساختن **branch** و تقسیم **issue** ها استفاده نمایید. (با استفاده از **commit** ها و تعیین **issue** ها میزان مشارکت هر نفر مشخص می‌شود). بعد از انجام این کار کدها را در یک **repository** به نام **CN_CHomeworks_3** در اکانت‌های **GitHub/GitLab** خود قرار دهید (به صورت **private**). همچنین در یک فایل **README.md** می‌توانید **report** و داکيومنت خود را کامل کنید و در کنار **repository** قرار دهید. در نهایت لینک این **repository** را در محل پاسخ تمرین قرار دهید. (از فرستادن فایل به صورت زیپ جدا خودداری نمایید). اکانت تی ای‌های این تمرین رو به **Repo** خودتون به عنوان **Maintainer** به پروژه اضافه کنید.

Git account: @TheSohrabX

- دقت کنید گزارش نهایی شما می‌بایست همانند یک **Document** باشد و شامل توضیح کد و ساختار کد، همچنین نتیجه نهایی اجرای کد و اسکرین شات‌های دقیق از تمام مراحل باشد. (در فایل **Readme.md** کنار فایل اصلی خود و در **Repo** مربوطه قرار دهید). این نکته حائز اهمیت است که فایل **PDF** به هیچ عنوان مورد پذیرش قرار نخواهد گرفت.
- ساختار صحیح و تمیزی کد برنامه، بخشی از نمره‌ی این پروژه شما خواهد بود. بنابراین در طراحی ساختار برنامه دقت به خرج دهید.
- برای هر قسمت کد، گزارش دقیق و شفاف بنویسید. کدهای ضمیمه شده بدون گزارش مربوطه نمره‌ای نخواهند داشت.
- هدف این تمرین یادگیری شماست. لطفاً تمرین را خودتان انجام دهید. در صورت مشاهده‌ی مشابهت بین کدهای دو گروه، مطابقت سیاست درس با گروه متقلب و تقلب دهنده **برخورد خواهد شد**. (استفاده از ابزارهای هوش مصنوعی، توجیهی برای شباهت پاسخ دو گروه نمی‌باشد).
- سؤالات خود را تا حد ممکن در **فروم درس** مطرح کنید تا سایر دانشجویان نیز از پاسخ آن بهره‌مند شوند؛ در صورتی که قصد مطرح کردن سؤال خاص‌تری دارید، از طریق ایمیل‌های زیر ارتباط برقرار کنید. **توجه داشته باشید** که سایر شبکه‌های اجتماعی راه ارتباطی رسمی با دستیاران آموزشی نیست و دستیاران آموزشی موظف به پاسخگویی در محیط‌های غیررسمی نیستند.

m.moradi1998@ut.ac.ir ○

erfan.ahmadi@ut.ac.ir ○

hasti.karimi@ut.ac.ir ○

موفق باشید