



مقدمه

وقتی دو یا چند شبکه به هم متصل می‌شوند، شبکه‌ای به هم پیوسته پدید می‌آید. مسیریاب‌های شبکه یا روترها شبکه‌ها را به هم متصل و بسته‌های داده را از شبکه‌ای به شبکه دیگر هدایت می‌کنند. آن‌ها مشخص می‌کنند که بسته‌های داده از چه مسیریایی باید بگذرند تا به مقصد برسند؛ فرآیندی که به آن مسیریابی می‌گویند. اما تعیین مسیر درست و بهینه در شبکه‌هایی که همبندی پیچیده‌ای دارند کار ساده‌ای نیست. لذا روترها از پروتکل‌های خاصی بهره می‌گیرند تا مناسب‌ترین مسیر از مبدا به مقصد را تعیین کنند.

در این پروژه هدف ما اجرای پروتکل مسیریابی Distance-Vector در پایتون است. برای این پیاده‌سازی از مفاهیم مختلفی از جمله الگوریتم بلمن-فورد، برنامه نویسی سوکت و multi-threading استفاده می‌شود.

مسیریابی بردار فاصله (Distance vector routing) نوعی پروتکل مسیریابی پویا (داینامیک) است. در این پروتکل، هر روتر در حافظه خود یک جدول مسیریابی تشکیل می‌دهد که کمکش می‌کند کوتاه‌ترین مسیر از گره‌ای به گره دیگر را در شبکه بیابد. هر روتر در شبکه با دیگر روترهای همسایه‌اش در تعامل است. وقتی یکی از روترها درباره تغییرات شبکه یا مسیرهای جدید اطلاعات تازه‌ای به دست می‌آورد آن را به دیگر روترهای همسایه، یعنی روترهایی که مستقیماً با آن‌ها اتصال دارد نیز اطلاع می‌دهد. روترهای همسایه پس از دریافت این اطلاعات، جدول مسیریابی‌شان را به‌روزرسانی و کوتاه‌ترین مسیر در شبکه را محاسبه می‌کنند. سپس آن‌ها نیز اطلاعات جدول مسیریابی‌شان را با روترهای مجاور خود به اشتراک می‌نهند و این رویه همچنان تکرار و اطلاعات تمام روترها مرتباً همگرا می‌شود. (convergence) ضمناً در مسیریابی بردار فاصله، حتی وقتی تغییری در شبکه رخ نداده است نیز همه روترها در فواصل زمانی معین اطلاعات جدول مسیریابی‌شان را با روترهای مجاورشان به اشتراک می‌نهند. مثلاً طبق پروتکل RIP که نوعی پروتکل مسیریابی بردار فاصله است، روترها هر ۳۰ ثانیه اطلاعات جدول مسیریابی‌شان را به روترهای مجاور اعلام می‌کنند.

پروتکل مسیریابی بردار فاصله نوعی پروتکل مسیریابی درون‌دامنه‌ای (Intradomain) است. مسیریابی درون‌دامنه‌ای یعنی مسیریابی درون سامانه خودگردان (Autonomous system) و منظور از سامانه خودگردان، خوشه‌ای از روترها و شبکه‌هاست که با مجموعه پروتکل‌های مشابهی کار می‌کنند و مدیریت واحدی دارند.

پروتکل مسیریابی بردار فاصله برای تعیین کوتاه‌ترین مسیر در شبکه از الگوریتم بلمن-فورد بهره می‌برد. همانطور که گفته شد، هر مسیری که گره‌ها را به هم متصل می‌کند مسافت یا اصطلاحاً هزینه‌ای دارد. پس برای محاسبه کوتاه‌ترین فاصله یا مسافت باید مراحل زیر انجام شود:

۱. ابتدا هزینه یا فاصله یک گره با خودش ۰ در نظر گرفته می‌شود.

۲. اگر یک گره مستقیماً به گره دیگری متصل نباشد، هزینه آن بی‌نهایت فرض می‌شود.

کمترین فاصله بین دو گره در یک گراف چنین به دست می‌آید:

$$D_{ij} = \min \{(C_{i1} + D_{1j}), (C_{i2} + D_{2j}), \dots, (C_{iN} + D_{Nj})\}$$

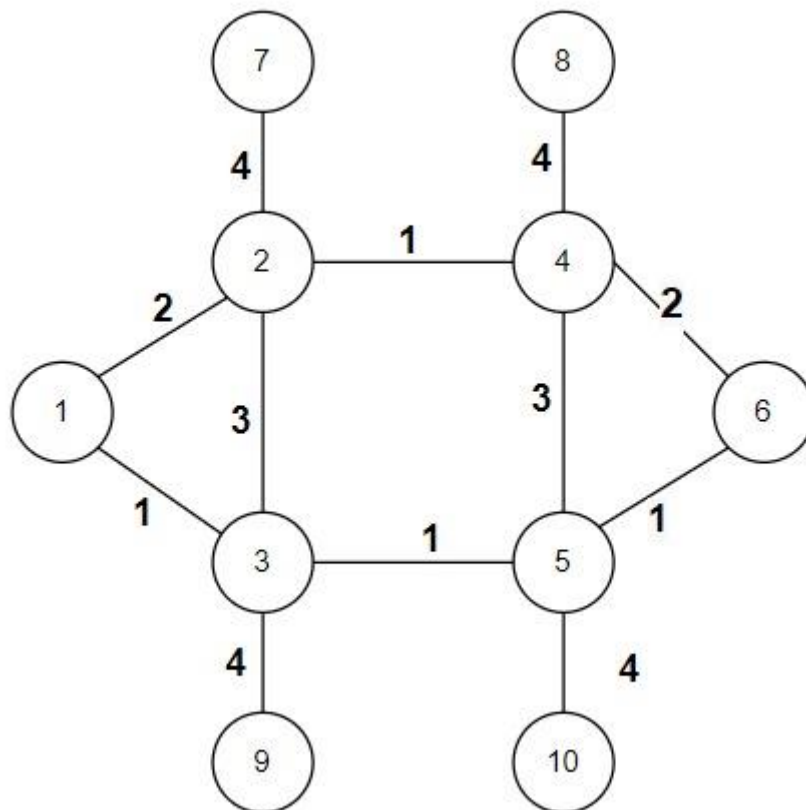
▪ منظور از D_{ij} کوتاهترین فاصله از گره i تا گره j است.

▪ منظور از C_{i1} هزینه گره i تا گره ۱ است.

الگوریتم بلمن-فورد فرض می‌کند که شما تمام اطلاعات اولیه همه گره‌ها را یکجا دارید. لذا الگوریتم بلمن-فورد می‌توانست همزمان (synchronous) برای هر گره مسیری تعیین کند. اما در الگوریتم مسیریابی بردار فاصله باید برای روترها در سامانه خودگردان، جدول مسیریابی ایجاد شود. ممکن است در سامانه خودگردان تغییراتی رخ دهد؛ مثلاً روتر جدیدی به شبکه اضافه شود یا یکی از شبکه‌ها خدماتش به یکی از روترها را متوقف کند یا یکی از لینک‌ها دچار نقص شود. در چنین مواقعی، جدول مسیریابی همه روترها باید غیرهمزمان (asynchronous) یعنی یکی پس از دیگری به‌روزرسانی شود، چون هر روتر باید جدول مسیریابی خود را بر پایه اطلاعات دریافتی از روترهای مجاور به‌روزرسانی کند.

پیاده‌سازی

برای انجام پیاده‌سازی از شکل زیر به‌عنوان توپولوژی مسیریاب‌ها در شبکه استفاده کنید. هزینه‌های لینک‌ها روی شکل مشخص است.



در پیاده سازی می‌توانید از دو کلاس به نام `Path` و `Neighbor` استفاده کنید.

Neighbor : پیوند بین هر دو گره که مستقیماً به یکدیگر متصل هستند.

Path: پیوند بین هر دو گره که ممکن است مستقیماً به یکدیگر متصل شوند یا نباشند.

برنامه شما باید شامل چندین رشته باشد، برخی از آنها دائمی هستند (در طول برنامه فعال هستند) و برخی از آنها موقتی هستند (فقط برای مدتی فعال هستند).

برنامه توسعه داده شده توسط شما باید قابلیت‌های زیر را داشته باشد :

- ✓ نمایش کوتاه ترین مسیرها از یک روتر به سایر روترها در شبکه
- ✓ نمایش جدول مسیریابی هر یک از روترها
- ✓ ویرایش هزینه لینک بین هر یک از جفت روترها
- ✓ اضافه کردن یک روتر و لینک‌های آن با هزینه مشخص برای ارتباط با سایر روترها
- ✓ حذف یک روتر و تمامی لینک‌های متصل به آن
- ✓ خروج از برنامه

بدیهی است هر زمان که روتر ایجاد می‌شود، هزینه پیوند خود را علاوه بر `Distance Vector` برای همه همسایگان ارسال می‌کند. هنگامی که همسایگان این بسته را دریافت می‌کنند، هزینه پیوند به این روتر را از "بی نهایت" به مقدار مشخص شده تغییر می‌دهند.

تغییر در هزینه‌های پیوند: وقتی هزینه‌های پیوند از طریق یک رابط مشخص تغییر می‌کرد، راهی برای همسایه دیگر وجود ندارد تا از این تغییر مطلع شوند لذا برای ثابت کردن هزینه پیوند در هر دو جهت، به محض ایجاد تغییر، روتر هزینه پیوند به روز شده را علاوه بر `Distance Vector` برای همسایه مربوطه ارسال می‌کند. سپس همسایه می‌تواند هزینه را به مقدار جدید به روز کند.

قوانین

۱. کد توسعه داده شده را به همراه گزارش کاملی از الگوریتم و نحوه اجرای کد را در مهلت مقرر در سامانه بارگذاری کنید.
۲. انجام پروژه به صورت انفرادی می‌باشد.
۳. از زبان برنامه نویسی پایتون و هر یک از کتابخانه‌های آن می‌توانید استفاده کنید.
۴. در مورد زمان ارائه پروژه‌ها اطلاع رسانی خواهد شد.
۵. مشخص شدن تقلب به منزله‌ی نمره صفر می‌باشد.

موفق باشید.