



تمرین برنامه‌نویسی دوم^۱ شبکه‌های کامپیوتری

بهار ۱۳۹۱

مدرس: مهدی خرازی

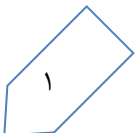
در تمرین بعد شما باید یک مسیریاب ساده پیاده‌سازی کنید. این مسیریاب از پروتکل OSPF پشتیبانی خواهد کرد. در این تمرین، مقدمات پیاده‌سازی مسیریاب در تمرین بعد فراهم می‌شود.

۱. مقدمه

در کلاس درس با سوئیچ و مسیریاب آشنا شدیم. دیدیم که یک سوئیچ در یک شبکه‌ی مستقل کار می‌کند و بسته را مستقیماً به گره‌ی مقصد می‌رساند. ولی مسیریاب در لایه‌ی سوم مدل OSI و بین چندین زیرشبکه کار می‌کند و بسته را از زیرشبکه‌ی مبدأ، به سمت زیرشبکه‌ی مقصد هدایت می‌کند. برای این کار، مسیریاب نیاز به جدولی دارد که توسط آن، از روی آدرس مقصد یک بسته، next hop را تعیین کند.

این جدول به‌طور کلی به دو روش ساخته می‌شود: ایستا و پویا. در روش ایستا، در هر مسیریاب، باید به ازای هر زیرشبکه‌ی مقصد، یک رکورد تعریف کرده و برای آن next hop را تعیین کنیم. اما در روش پویا، مسیریاب‌ها به صورت خودکار این جدول را می‌سازند و به‌نگام می‌کنند. از مزایای مسیریابی پویا این است که مسیریاب‌ها با خارج شدن مسیریاب‌های دیگر یا اتصالات از سرویس، نیازی به تنظیم دوباره به‌دست مدیر سیستم ندارند و جداولشان به صورت خودکار به‌نگام می‌شود.

دو روش کلی معروف link-state و distance vector برای مسیریابی پویا وجود دارد که با آنها در کلاس آشنا شده‌ایم. پروتکل‌های OSPF و RIP، دو پروتکل هستند که به ترتیب مربوط به این دو روش می‌باشند. در این تمرین می‌خواهیم بخشی از یک مسیریاب را بسازیم که در آینده از پروتکل مبتنی بر OSPF پشتیبانی می‌کند.



^۱ با تشکر از بهنام مومنی، مهدی احمدی‌نژاد، کامیار اللهوردی، سجاد فولادی، علی محمد ربانی، مهرداد مرادی و بهنام حاتمی

۲. پروتکل OSPF

پروتکل مورد استفاده در مسیریاب‌ها، مبتنی بر نسخه‌ی دوم پروتکل OSPF است. شماره‌ی در نظر گرفته شده برای پروتکل OSPF در لایه‌ی IP، ۸۹ می‌باشد. در این مستند، خلاصه‌ای از این پروتکل و قسمت‌هایی از آن که باید پیاده‌سازی شوند آورده شده است. اما برای کسب اطلاعات تکمیلی، مطالعه‌ی منابع زیر مفید است:

- 1- <http://tools.ietf.org/html/rfc2328>
- 2- <http://fengnet.com/book/OSPFandISIS/ch02.html>
- 3- http://en.wikipedia.org/wiki/Open_Shortest_Path_First
- 4- http://en.wikipedia.org/wiki/Link-state_advertisement

در این تمرین، فرض می‌شود که همه‌ی مسیریاب‌ها در یک ناحیه قرار دارند. مکانیزم‌های تأیید هویت^۲ نیز در این تمرین استفاده نخواهند شد. بستر پیاده‌سازی تمرین، شبکه‌ی Ethernet است که از شبکه‌های با قابلیت broadcast محسوب می‌شود.

۲-۱. اصطلاحات کلیدی

۲-۱-۱. ناحیه‌ی خودمختار^۳

گروهی از مسیریاب‌ها که با استفاده از یک پروتکل مسیریابی مشترک، اطلاعات مسیریابی را با یکدیگر رد و بدل می‌کنند.

۲-۱-۲. ناحیه‌ی مسیریابی^۴

گروهی شامل تعدادی از شبکه‌های یک ناحیه‌ی خودمختار که می‌خواهیم توپولوژی آن از سایر شبکه‌های AS مخفی بماند. پروتکل OSPF می‌تواند برای تعدادی ناحیه در یک AS کار کند و اطلاعات هر ناحیه را مخفی نگه دارد. برای کاهش پیچیدگی، در این تمرین فرض می‌شود کل شبکه‌ها و مسیریاب‌هایی که روی آن‌ها پروتکل را اجرا می‌کنیم در یک ناحیه قرار دارند.

۲-۱-۳. Router ID

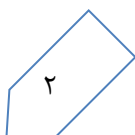
عددی ۳۲ بیتی و یکتا که به هر مسیریاب که در یک ناحیه‌ی خودمختار با OSPF کار می‌کند نسبت داده می‌شود. هر مسیریاب خود را با این عدد به سایرین معرفی می‌کند.

۲-۱-۴. شبکه‌ی Broadcast

شبکه‌ای که قابلیت اتصال به بیش از دو مسیریاب را دارا باشد و بتواند یک پیغام فیزیکی را به دست تمامی روترهای متصل برساند. در پروتکل Hello، مسیریاب‌های همسایه با استفاده از این قابلیت در شبکه‌های broadcast به صورت پویا شناسایی می‌شوند. شبکه‌ی Ethernet نمونه‌ای از یک شبکه‌ی broadcast است.

۲-۱-۵. Interface

اتصال بین یک مسیریاب و هر یک از شبکه‌های متصل به آن. هر interface یک آدرس IP و mask مختص به خود دارد. هر interface وضعیتی را که در آن قرار دارد نگه می‌دارد.



- 2 Authentication
- 3 Autonomous System (AS)
- 4 Area

۲-۱-۶. مسیریاب‌های همسایه

دو مسیریاب که به یک شبکه‌ی مشترک مستقیماً متصلند. رابطه‌ی همسایگی برای مسیریاب‌ها در پروتکل Hello و در این تمرین به صورت پویا و با استفاده از قابلیت broadcast در Ethernet تعیین می‌شود.

۲-۱-۷. مجاورت

رابطه‌ای که بین مسیریاب‌های همسایه‌ی انتخاب شده برای تبادل اطلاعات مسیریابی شکل می‌گیرد. به طور کلی، لزوماً هر دو مسیریاب همسایه با یکدیگر مجاور نخواهند شد. اما در این تمرین به دلیل عدم وجود Authentication و فرض بر قرار گرفتن همه‌ی مسیریاب‌ها در یک ناحیه، رابطه‌ی همسایگی، هم‌ارز با رابطه‌ی مجاورت است.

۲-۱-۸. Link state advertisement

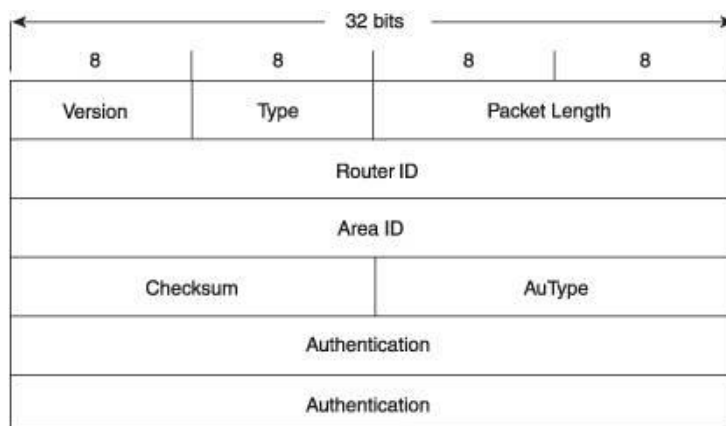
داده‌ای که وضعیت فعلی یک مسیریاب یا شبکه را بازنمایی می‌کند. برای یک مسیریاب، این مفهوم یعنی وضعیت فعلی interface های خود و وضعیت فعلی مسیریاب‌های مجاورش. هر LSA^۵ در ناحیه‌ی مسیریابی و بین مسیریاب‌ها پخش می‌شود. از جمع شدن LSAهای همه‌ی شبکه‌ها و مسیریاب‌هاست که پایگاه داده‌ی پروتکل OSPF^۶ شکل می‌گیرد.

۲-۱-۹. مسیریاب برگزیده^۷

مسیریابی است که در هر شبکه‌ی broadcast با بیش از دو مسیریاب وجود دارد. وظیفه‌ی اصلی این مسیریاب تولید LSA برای شبکه است. این مسیریاب در روند پروتکل HELLO انتخاب می‌شود.

۲-۲. سرآیند اصلی

پروتکل OSPFv2، پنج نوع بسته‌ی اصلی دارد که برای برقراری ارتباط و ارسال و دریافت اطلاعات بین مسیریاب‌ها استفاده می‌شوند. این پنج بسته همگی سرآیندی مطابق شکل زیر دارند:



تصویر شماره ۱ - سرآیند اصلی و مشترک بسته‌های OSPF

در هر پنج نوع بسته، این سرآیند ۲۴ بیتی مشابه است و تفاوت اصلی آنها در مقدار فیلد Type می‌باشد. در هر نوع از این بسته‌ها، Payload متفاوت است که باید با مراجعه به منابع داده شده، ساختار آنها را با جزئیات مطالعه کنید. اما ۳ نوع از این بسته‌ها، شامل بسته‌ی مهم دیگری تحت عنوان LSA می‌شوند.

با توجه به اینکه در این تمرین از Authentication استفاده نمی‌شود، فیلدهای مربوط به Authentication و Checksum و همچنین Optionهایی که استفاده نمی‌شوند باید دارای مقدار 0 باشند.

⁵ Link State Advertisement

⁶ Link-state database

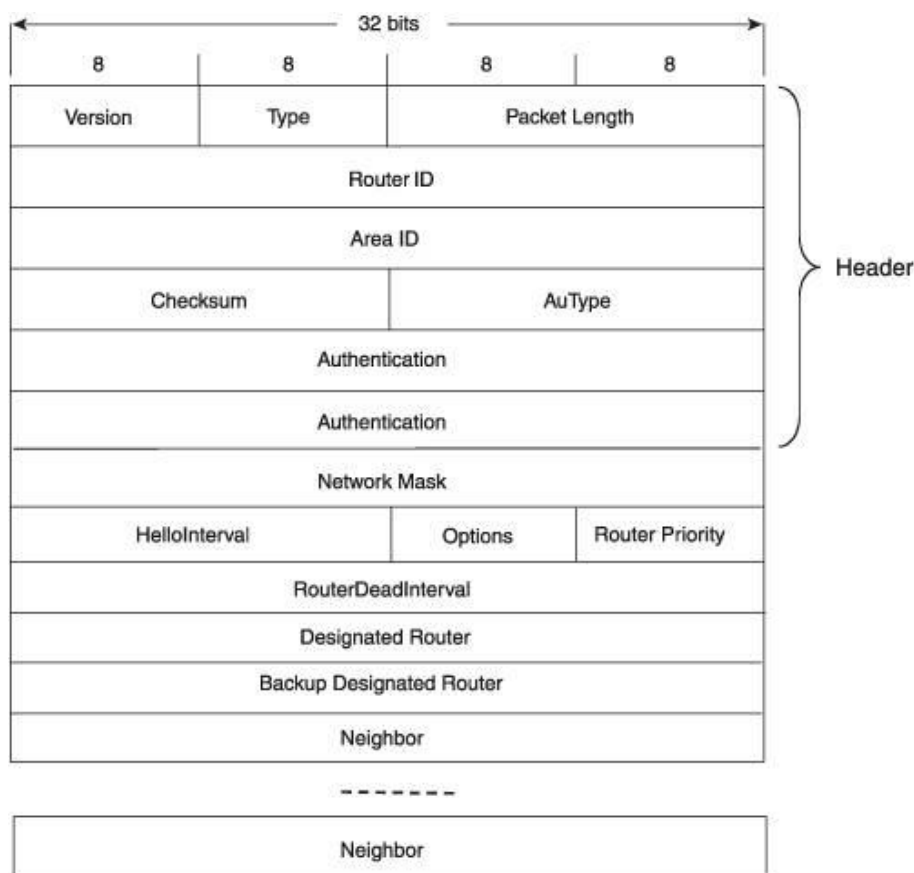
⁷ Designated router

۳-۲. پروتکل HELLO

هنگامی که پروتکل OSPF بر روی یک مسیریاب شروع به کار می‌کند، همسایه‌ها باید پیدا شوند و همچنین روابط مجاورت تعیین گردند. وظایف پروتکل HELLO عبارتند از:

- یافتن مسیریاب‌های همسایه،
- انجام Three-way Handshake برای اطمینان از وجود رابطه‌ی دو طرفه بین دو مسیریاب همسایه،
- تبادل اطلاعات لازم برای تعیین اینکه آیا این دو مسیریاب همسایه می‌توانند مجاور باشند یا خیر (که در این تمرین این قسمت وجود ندارد)،
- تشخیص همسایه‌ها و مجاوران خراب با استفاده از مکانیزم Keepalive بعد از تعیین روابط مجاورت.

یک بسته‌ی HELLO به صورت زیر است:



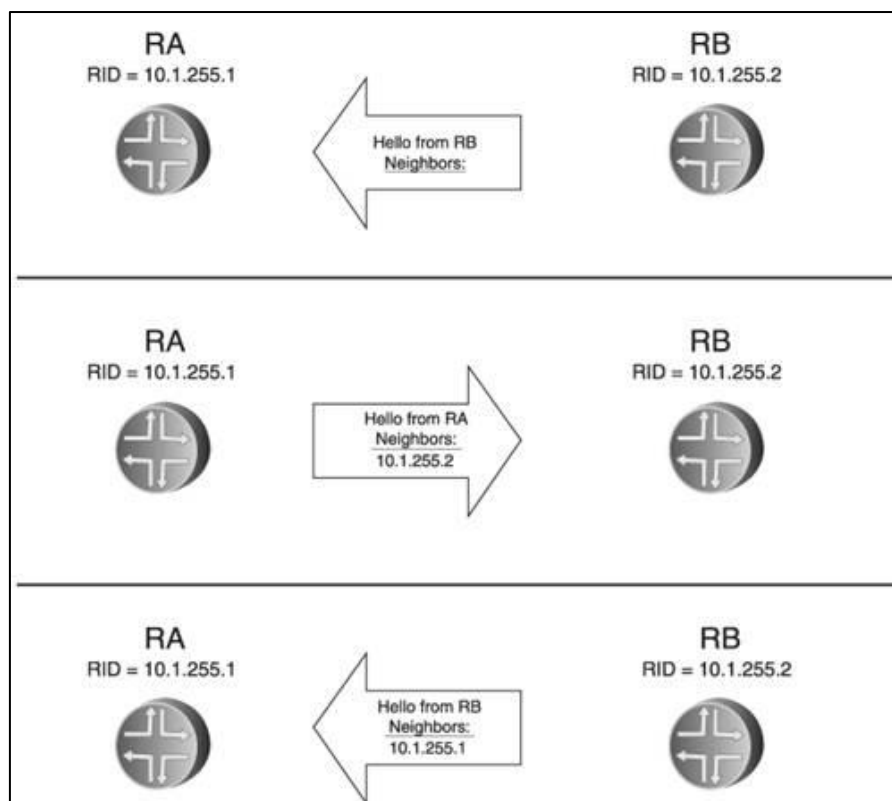
تصویر شماره ۲ - ساختار بسته‌های HELLO

بخش‌های مربوط به پروتکل HELLO به صورت زیر است:

- **Network Mask**: یک مقدار ۳۲ بیتی است که تعیین کننده‌ی mask مربوط به زیرشبکه‌ای است که بسته روی آن ارسال می‌شود. برای مثال اگر بسته روی زیر شبکه‌ی 192.168.0.1/24 ارسال شود، مقدار این فیلد برابر با 0xffffffff خواهد بود.
- **Hello Interval**: تعیین می‌کند که مسیریاب ارسال کننده‌ی بسته‌ی HELLO در چه بازه‌های زمانی اقدام به ارسال مجدد بسته‌ی HELLO می‌کند.
- **Options**: مشخص می‌کند که چه قابلیت‌های اختیاری توسط مسیریاب پشتیبانی می‌شود.
- **Router Priority**: عدد هشت بیتی است که در فرآیند انتخاب مسیریاب برگزیده مورد استفاده قرار می‌گیرد.

- **RouterDeadInterval**: تعیین کننده‌ی زمانی است که همسایه‌های مسیریاب باید برای پیام HELLO منتظر بمانند؛ پیش از آنکه آن را به عنوان یک روتر خراب در نظر بگیرند.
- **Designated Router و Backup Designated Router**: این دو فیلد هم در فرآیند تعیین مسیریاب برگزیده مورد استفاده قرار می‌گیرند. این فرآیند در جای خود توضیح داده خواهد شد.
- **Neighbor**: لیستی از Router ID هایی است که مسیریاب ارسال کننده از آنها پیام HELLO دریافت کرده است. از این لیست برای عملیات Three-way Handshake استفاده می‌شود.

بعد از این که یک مسیریاب به همسایه‌های خود پیام HELLO را ارسال کرد، باید از برقراری ارتباط دوطرفه بین خود و آنها مطمئن شود. برای اینکار، عملیات Three-Way Handshake صورت می‌گیرد. برای مثال به شکل زیر دقت کنید:



تصویر شماره ۳ - عملیات Three-Way Handshake برای برقراری ارتباط دوطرفه

روتر RB به تازگی به شبکه اضافه شده است و از روتر RA بی‌خبر است. این روتر یک بسته‌ی HELLO را به همسایه‌های خود ارسال می‌کند. به دلیل اینکه این روتر به تازگی به شبکه پیوسته است، لیست همسایه‌های آن خالی است. RA با دریافت این HELLO، یک بسته‌ی HELLO به RB ارسال می‌کند و در لیست همسایه‌های این بسته، RB را هم قرار می‌دهد. در سومین بسته‌ی HELLO، که توسط RB ارسال می‌شود، RA در لیست همسایه‌ها قرار داده می‌شود و به این ترتیب دو طرف از سلامت یکدیگر با خبر خواهند شد!

به طور کلی بسته‌های HELLO همیشه با فاصله زمانی ثابت HelloInterval ثانیه از هم، از هر interface ارسال می‌شوند. با ارسال هر بسته‌ی HELLO یک تایمر فعال می‌شود. با رسیدن این تایمر به HelloInterval، بسته‌ی HELLO بعدی ارسال می‌شود. حتی در صورتی که مسیریاب بسته‌ی HELLO از یکی از همسایگان دریافت کند، پس از بهنگام کردن لیست همسایگان، تا زمانی که تایمرش به این زمان نرسد، بسته‌ی HELLO ارسال نمی‌کند.

هر مسیریاب برای هر همسایه‌اش یک تایمر عدم فعالیت^۸ دارد. وظیفه‌ی این تایمر این است که اگر یک همسایه وجود خود را با بسته‌های HELLO در یک بازه‌ی زمانی بزرگ‌تر از RouterDeadInterval اعلام نکرد، آن را از لیست همسایگانش خارج کند. بنابراین اگر بسته‌های HELLO در مسیر گم شوند، تا زمانی که تایمر گیرنده به RouterDeadInterval همسایه‌اش نرسیده باشد، مشکلی ایجاد نمی‌شود.

بعد از عملیات Three-way Handshake، باید Designated Router و Designated Backup Router انتخاب شوند که در ادامه توضیح داده خواهد شد.

۲-۴. انتخاب Designated Router و Backup Designated Router

پروتکل OSPF بر اساس انتخاب یک Designated Router کار می‌کند. این DR با سایر مسیریاب‌های شبکه ارتباط مجاورت برقرار می‌کند و وظیفه‌ی اصلی آن ارسال Network LSA به عنوان نقشه‌ی شبکه برای سایرین است.

برای کاهش اثرات مربوط به خارج شدن DR از شبکه، یک مسیریاب دیگر هم به عنوان مسیریاب برگزیده‌ی پشتیبان انتخاب می‌شود. در صورت خارج شدن DR از شبکه، این مسیریاب به طور خودکار جایگزین DR می‌شود.

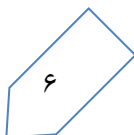
در صورتی که در بخش قبل به بسته‌ی HELLO توجه کنیم، می‌بینیم که سه فیلد Router Priority، Designated Router و Backup Designated Router برای استفاده در این فرآیند وجود دارند. هرکدام از interface‌های مسیریاب دارای یک Priority هستند که عددی بین 0 تا 255 است. مقدار این Priority می‌تواند به صورت دستی تنظیم شود؛ در غیر این صورت عموماً مسیریاب‌ها دارای یک مقدار پیش‌فرض هستند. دو فیلد دیگر هم آدرس IP مربوط به DR و BDR را در خود دارند. در صورتی که هنوز BDR و DR انتخاب نشده باشند، مقدار این فیلد 0.0.0.0 است.

۲-۴-۱. شروع

هنگامی که یک مسیریاب جدید روی شبکه فعال می‌شود، مقادیر مربوط به DR و BDR واسطه‌های خود را برابر با 0.0.0.0 قرار می‌دهد. با این کار مشخص می‌شود که واسط هنوز DR و BDR انتخاب یا کشف نکرده است. همچنین یک تایمر انتظار^۹ برابر با زمان RouterDeadInterval برای واسطه‌های خود فعال می‌کند.

با رخداد هر یک از سه شرط زیر، عملیات انتخاب DR و BDR آغاز می‌شود:

۱. تایمر انتظار واسط به پایان برسد و هنوز DR و BDR برای شبکه کشف نشده باشد.
۲. در بسته‌های HELLO دریافتی، وجود یا عدم وجود BDR مشخص شود. این شرط در صورتی اتفاق می‌افتد که یک همسایه خود را به عنوان BDR معرفی کند و یا یک همسایه خود را DR معرفی کند و ضمناً اعلام کند BDR وجود ندارد.
۳. وضعیت یکی از همسایگان تغییر کند. یعنی یا ارتباط دو طرفه با آن قطع شده باشد. یا ارتباط دو طرفه‌ی جدیدی شکل گرفته باشد. یا مسیریابی که خود را DR یا BDR اعلام می‌کرده، دیگر این کار را نکند. و یا مسیریابی که خود را DR یا BDR اعلام نمی‌کرده، شروع به اعلام خود به عنوان DR یا BDR کند.



⁸ Inactivity Timer

⁹ Wait Timer

۲-۴-۲. مراحل انتخاب

مسیریابی لیستی از همسایه‌هایی را که می‌توانند به عنوان یک DR یا BDR انتخاب شوند تعیین می‌کند. مسیریابی‌هایی در این لیست قرار می‌گیرند که مقدار Priority آنها بزرگتر از صفر باشد. خود مسیریاب هم در این لیست قرار دارد، تنها در صورتی که مقدار Priority آن بزرگتر از صفر باشد.

۱. مقادیر فعلی DR و BDR واسط در نظر گرفته می‌شوند. این مقادیر بعداً برای مقایسه مورد نیازند. (به عنوان مثال، اگر به دلیل تغییر وضعیت یکی از همسایگان عملیات انتخاب شروع شود، مقادیر DR و BDR برابر با 0.0.0.0 نیستند.)

۲. در لیست تعیین شده، از بین مسیریابی‌هایی که خود را BDR اعلام کرده‌اند، مسیریابی که دارای بیشترین مقدار Priority است، به عنوان BDR انتخاب می‌شود. در صورتی که Priority‌ها مساوی باشند، مسیریابی انتخاب می‌شود که ID بزرگتری دارد. اگر هیچ مسیریابی خود را BDR اعلام نکرده بود، همین روش انتخاب روی همه‌ی مسیریاب‌های لیست که خود را DR اعلام نکرده‌اند، اعمال می‌شود.

۳. در لیست تعیین شده، از بین مسیریابی‌هایی که خود را DR اعلام کرده‌اند، مسیریابی که دارای بیشترین مقدار Priority است، به عنوان DR انتخاب می‌شود. در صورتی که Priority‌ها مساوی باشند، مسیریابی انتخاب می‌شود که ID بزرگتری دارد. اگر هیچ مسیریابی خود را DR اعلام نکرده بود، همان مسیریابی که به عنوان BDR انتخاب شد، به عنوان DR نیز انتخاب می‌شود.

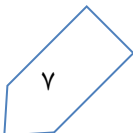
۴. اگر خود مسیریاب در این فرآیند به عنوان BDR یا DR جدید انتخاب شد، و یا در گذشته BDR بوده و دیگر نیست و یا در گذشته DR بوده و دیگر نیست، مراحل ۲ و ۳ تکرار می‌شوند و بعد از آن مرحله‌ی ۵ اجرا می‌شود. این مرحله مکانیزمی است برای آنکه یک مسیریاب نتواند خود را هم به عنوان BDR و هم DR انتخاب و معرفی کند. مثلاً اگر مسیریاب به عنوان DR انتخاب شد، در اجرای دوباره‌ی مرحله‌ی ۲، اجازه‌ی انتخاب شدن به عنوان BDR را ندارد.

۵. در نتیجه‌ی محاسبات، ممکن است خود مسیریاب به عنوان DR یا BDR انتخاب شود. در هر صورت، وضعیت بعدی مسیریاب با توجه به انتخاب شدن یا نشدنش به عنوان DR و BDR تنظیم می‌شود.

در انتخاب BDR و DR، ممکن است یک مسیریاب هم به عنوان DR و هم به عنوان BDR انتخاب شود. البته این امکان برای مسیریابی که الگوریتم را اجرا کرده وجود ندارد. در این الگوریتم لزوماً DR بیشترین Priority و BDR دومین بیشترین Priority را ندارد. الگوریتم طوری کار می‌کند که نیاز کمتری به تغییر DR باشد.

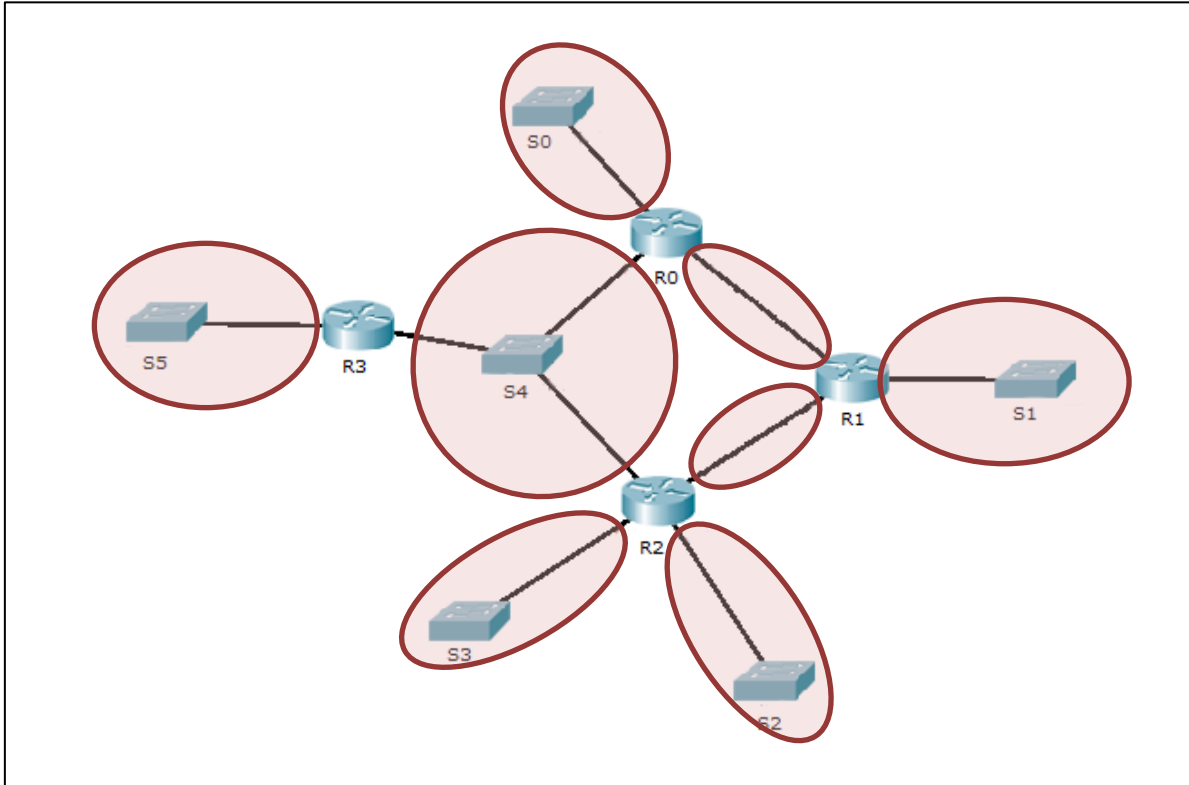
اگر مسیریابی که الگوریتم را اجرا می‌کند، خود مجاز به انتخاب شدن به عنوان DR نباشد، این امکان وجود دارد که در انتهای اجرای الگوریتم، DR و BDR انتخاب نشوند. در ضمن، اگر این مسیریاب تنها مسیریاب با Priority بیشتر از 0 باشد که به شبکه متصل است، خود را به عنوان DR انتخاب می‌کند و BDR انتخاب نمی‌شود.

بعد از انتخاب DR، همه‌ی مسیریاب‌های متصل به آن، برای دریافت و ارسال اطلاعات با آن مجاور می‌شوند.



۳. برنامه‌ی مسیریاب

هدف نهایی این برنامه ، استفاده از پروتکل توصیف شده در این تمرین و تمرین بعد برای دریافت بسته‌ها و هدایت آنها به سمت شبکه‌ی مقصد است. در واقع شما در این تمرین برنامه‌ی مربوط به هر مسیریاب را تا انتهای عملیات انتخاب DR و BDR خواهید نوشت. به هر مسیریاب، تعدادی مسیریاب دیگر متصل است.



تصویر شماره ۴ - توپولوژی شبکه‌ی نمونه

۳-۱. شبکه‌های مستقیماً متصل

هر مسیریاب تعدادی interface دارد. برای پیدا کردن این تعداد می‌توانید از متغیر `countOfInterfaces` که در کلاس `Machine` تعریف شده است استفاده کنید.

پروتکل مسیریابی OSPF برای شروع به کار روی هر مسیریاب، به داشتن آدرس و Mask شبکه‌هایی که مستقیماً به آن مسیریاب متصل هستند نیاز دارد. برای این منظور، باید با استفاده از IP Address و Subnet Mask هر یک از Interfaceها، Network Address و Network Mask شبکه‌های متصل به مسیریاب را محاسبه و نگهداری کنید. در کلاس `Interface` می‌توانید متغیرهای مورد نیاز برای این کار را پیدا کنید.

مثلاً اگر برای `iface[0]` داشته باشیم:

IP/Mask = 213.233.168.17/29

می‌توانیم نتیجه بگیریم که یکی از شبکه‌های متصل به مسیریاب، شبکه‌ی زیر است:

IP/Mask = 213.233.168.16/29

۳-۲. مقادیر اولیه

هر مسیریاب یک ID یکتا دارد که یک عدد ۳۲ بیتی است و با آن خود را به سایر مسیریاب‌ها معرفی می‌کند. برای این کار، مسیریاب شما باید با استفاده از `iface[0].getIp()` آدرس IP اولین `interface` اش را استفاده کند. باید دقت داشته باشید که این عدد برای هر مسیریاب ثابت باشد و برای بسته‌های مختلفی که روی `interface`‌های مختلف ارسال می‌کند، متفاوت نباشد.

بقیه‌ی مقادیر اولیه مورد نیاز، به شکل زیر از طریق `getCustomInformation()` قابل دسترسی هستند.

```
<area-id>
<hello-interval>
<router-dead-interval>
<iface[0]-priority>
<iface[1]-priority>
...
<iface[n]-priority>
```

۳-۳. خروجی

خروجی‌های چاپی برنامه شما در عملکرد سیستم خودکار داوری تأثیری ندارد. لذا می‌توانید هر آنچه را که نیاز دارید چاپ کنید.

۴. محیط

در این تمرین نیز، مانند تمرین قبل، از محیط پرتو برای اجرای برنامه شما استفاده خواهد شد. برای اطلاعات بیشتر به مستند «[راهنمای چارچوب کاربر](#)» مراجعه فرمایید.

۵. انتظارات

شما باید برنامه‌ی مربوط به مسیریاب را به شکلی در این تمرین پیاده‌سازی کنید که در تمرین بعد امکان تکمیل آن وجود داشته باشد.

۶. نکات ضروری

- در صورتیکه هر مشکل یا پرسشی داشتید که فکر می‌کنید پاسخ آن برای همه مفید خواهد بود، لطفاً آن را به گروه پستی درس ارسال کنید.
- از فرستادن جواب تمرین به گروه پستی جداً خودداری کنید.
- فرستادن کل یا قسمتی از برنامه‌تان برای افراد دیگر، یا استفاده از کل یا قسمتی از برنامه‌ی فردی دیگر به نام خود، تقلب محسوب می‌شود.
- پس از اتمام کارتان لازم است که پوشه `user` را به همراه `Makefile` فشرده کرده و بر روی سیستم خودکار داوری^{۱۰} `upload` کنید.