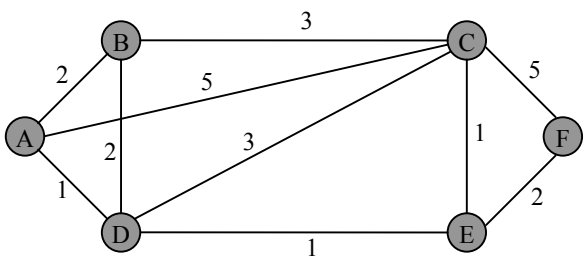
۱) مفاهیم اولیهٔ مسیریابی

در فصل قبل اشاره شد که مسیریاب ابزاری است که ارتباط دو یا چند شبکه را برقرار می‌نماید. مجدداً با مراجعه به فصل قبلی به شکلهای (٤-۳) و (٥-۳) دقت کنید. در این دو شکل ، کامپیوترهای S1 تا S5 نقش مسیریاب را ایفا می‌کنند؛ مجموعهٔ این مسیریابها و کانالهای فیزیکی مابین آنها "زیرساخت ارتباطی"[۱] شبکه را تشکیل می‌دهد. در نشان دادن زیرساخت ارتباطی از یک شبکه ، تمامی ماشینهای میزبان حذف خواهند شد چراکه این ماشینها هیچ تاثیری در برقراری ارتباط و حمل ترافیک بسته‌ها نداشته و به عنوان استفاده‌کنندهٔ نهایی[۲] مطرح هستند.

در شکل (۱-٤) به زیرساخت ارتباطی یک شبکهٔ فرضی که در قالب یک گراف نشان داده شده است ، دقت کنید. در این گراف گره‌های A تا F مسیریابها هستند و خطوط ارتباطی بین هر دو گره (لبه[۳]) نشان‌دهندهٔ وجود یک کانال فیزیکی بین آنها می‌باشد. اعدادی که روی هر لبه نوشته شده است پارامتر هزینهٔ رسیدن از یک گره به گره دیگر خواهد بود. در ساده‌ترین حالت می‌توان معیار هزینه را زمان تاخیر در نظر گرفت. البته پارامتر تاخیر را باید ترکیبی از "تاخیر فیزیکی انتشار" و "زمان پردازش" یعنی زمان اجرای الگوریتم مسیریابی بر روی یک بسته در نظر گرفت. مجموع این دو زمان می‌تواند به عنوان پارامتر هزینه در نظر گرفته شود. البته در بخشهای آتی خواهیم دید که معیار هزینه در برخی از مسیریابها بر اساس پارامترهای پیچیده‌تری همانند امنیت ، سیاست و اقتصاد ارزیابی می‌شود.

حال فرض کنید بسته‌ای وارد مسیریاب A شده تا پس از طی مسیر ، به F تحویل داده شود. اصلیترین وظیفهٔ الگوریتمهای مسیریابی ، پیدا کردن مسیری بهینه از A به F می‌باشد به‌گونه‌ای که هزینهٔ کل مسیر به حداقل برسد.



شکل (۱-٤) زیرساخت ارتباطی یک شبکهٔ فرضی

---

[۱] Communication Subnet
[۲] End User
[۳] Edge

به‌گونه‌ای که از شکل مشخص است دوازده مسیر متفاوت برای رسیدن یک بسته از A به F وجود دارد که در زیر فهرست شده اند:

A→B→C →F	≺	با هزینهٔ ۱۰	≺	A→D→C →F	با هزینهٔ ۹
A→B→C →E→F	≺	با هزینهٔ ۸	≺	A→D →E→F	با هزینهٔ ۴
A→B→C→D→E→F	≺	با هزینهٔ ۱۱	≺	A→D→C→ E→F	با هزینهٔ ۷
A →C →F	≺	با هزینهٔ ۱۰	≺	A→D→B→C →F	با هزینهٔ ۱۱
A→ C →E→F	≺	با هزینهٔ ۸	≺	A→D→B→C→ E→F	با هزینهٔ ۹
A→ C→D→E→F	≺	با هزینهٔ ۱۱	≺	A→ D→C→E→F	با هزینهٔ ۷

از بین این مسیرها بهترین مسیر برای رسیدن از A به F ، مسیر A→D →E→F خواهد بود.
دو مسئلهٔ مهم در مسیریابی مطرح است:
الف) هر یک از مسیریابها چگونه از پارامتر هزینهٔ کل کانالهای شبکه مطلع شوند ، تا بتوانند گراف زیرساخت ارتباطی شبکه را تشکیل داده و بهترین مسیر را انتخاب نمایند؟
ب) چه الگوریتمی برای یافتن بهترین مسیر انتخاب شود تا از لحاظ سرعت پردازش و تصمیم‌گیری ، بهینه بوده و بسته ها را با تاخیر و انتظار مواجه نکند؟ (یعنی از لحاظ پیچیدگی زمانی الگوریتم بهینه باشد).

بهتر است قبل از ادامهٔ بحث به اصطلاحات کلیدی از فصل قبل که به صورت خلاصه در جدول (۲-٤) ارائه شده است دقت کنید.

۱-۱) روشهای هدایت بسته‌های اطلاعاتی در شبکه‌های کامپیوتری

دو روش برای انتقال بسته‌های اطلاعاتی در شبکه‌های کامپیوتری مطرح است که هر کدام از آنها در شبکه‌های امروزی به نحوی مورد استفاده قرار می‌گیرد:
الف) روش "مدار مجازی"[۱] که اختصاراً روش VC گفته می‌شود.
ب) روش "دیتاگرام"[۲]

[۱] Virtual Circuit
[۲] Datagram
۲ مفهوم روش دیتاگرام در مسیریابی را با مفهوم دیتاگرام به معنای "یک واحد اطلاعاتی" در لایهٔ اینترنت اشتباه نکنید.

آدرسهای MAC : آدرسهائی هستند که در لایه فیزیکی (لایهٔ اوّل) تعریف می‌شوند و فقط برای انتقال فریمها روی کانال مورد استفاده قرار می‌گیرند. در حقیقت این آدرسها روشی برای تحریکِ سخت‌افزار کارت شبکه هستند تا اطلاعات را از روی کانال مشترک بردارد. بنابراین چگونگی تعریف این آدرسها و اصول آدرس‌دهی و اندازهٔ این آدرسها (برحسب بایت) شدیداً به پروتکل و توپولوژی شبکه وابسته است.

آدرسهای IP : آدرسهای جهانی و منحصر به فرد که یک ماشین را فارغ از نوع سخت‌افزار و نرم‌افزار آن ، مشخص می‌نماید.

بستهٔ IP : یک واحد اطلاعاتی با اندازه محدود (ولی متغیر) که باید در زیرساخت ارتباطی یک شبکه از مبدأ به سمت مقصد هدایت شود.

در هدایت بسته‌های اطلاعاتی از یک مسیریاب به مسیریاب دیگر آدرسهای MAC دائماً تغییر می‌کنند ولی آدرسهای IP ثابت و جهانی هستند. مسیریاب بر اساس این آدرسها هدایت بسته را به سمت مقصد انجام می‌دهد.

مسیریاب : ابزاری است که تعدادی ورودی و خروجی داشته و بسته‌های اطلاعاتی را از ورودی تحویل گرفته و بر اساس آدرسهای جهانی یکی از کانالهای خروجی را برای انتقال بسته انتخاب می‌نماید ، به نحوی که بسته را به مقصد نزدیک نماید.

شرایط توپولوژیکی شبکه : مجموعهٔ مسیریابها و کانالهای فیزیکی مابین آنها در زیرساخت ارتباطی یک شبکه ، توپولوژی آن شبکه را تشکیل می‌دهد. با توجه به آنکه با ورود یک مسیریاب جدید به شبکه یا خرابی یکی از کانالهای ارتباطی یا حذف یک مسیریاب ، توپولوژی زیرساخت ارتباطی تغییر خواهد کرد لذا توپولوژی شبکه متغیر با زمان خواهد بود.

شرایط ترافیکی شبکه : تعداد متوسط بسته‌های اطلاعاتی را که در واحد زمان روی یک کانال ارسال (یا دریافت) می‌شود ، ترافیک آن کانال گویند. چون تولید بسته‌های اطلاعاتی توسط ماشینهای میزبان کاملاً تصادفی و نامعین است بنابراین ترافیک در شبکه نیز کاملاً متغیر با زمان خواهد بود.

گام یا Hop : به عبور بسته از یک مسیریاب ، گام و به تعداد مسیریابهایی که یک بسته در طی مسیر خود به سمت مقصد می‌پیماید "تعداد گام" گفته می‌شود.

ازدحام یا Congestion : زمانی که تعداد متوسط بسته‌های ورودی به یک مسیریاب از تعداد متوسط بسته‌های خروجی از آن بیشتر شود ، ازدحام رخ داده و تاخیر ارسال بسته‌ها در آن مسیریاب شروع به افزایش خواهد کرد. هرگاه تاخیر به حدی برسد که طول عمر بسته‌ها تمام شود ، اصطلاحاً بن‌بست –Deadlock– پدید آمده و مسیریاب عملاً مسدود شده است.

جدول (۲-٤) تعریف برخی از اصطلاحات کلیدی در مبحث مسیریابی

در روش مدار مجازی قبل از شروع به ارسال بسته‌های اطلاعاتی از یک ماشین ، ابتدا یک مسیر بین مبدأ و مقصد برقرار می‌شود؛ بدینصورت که مبدأ ابتدا با ارسال یک بستهٔ کنترلی خاص با یک شمارهٔ ویژه بر روی شبکه اعلام می‌کند که خواستار برقراری ارتباط با یک مقصد خاص می‌باشد. هر مسیریاب که آن بسته را دریافت کند ضمن پیداکردن یک مسیر مناسب برای آن بسته شمارهٔ آن را در جدولی درج میکند و از آن به بعد هر بسته‌ای که با این شماره وارد شود از همان مسیری که برای بستهٔ اول انتخاب شده بود به سمت مقصد هدایت می‌شود. بنابراین تمامی بسته‌های اطلاعاتی که بعد از برقراری یک مسیر ، از مبدأ به سمت مقصد ارسال می‌شوند نیاز به مسیریابی جداگانه نخواهند داشت. به این مسیر که فقط یکبار ایجاد می‌شود ، "مدار مجازی" گفته می‌شود. "مدار مجازی" تا وقتی با اطلاع طرفین ارتباط و اعلام به مسیریابهای واقع بر روی مدار خاتمه داده نشود ، برقرار خواهد ماند. از آنجایی که در روش VC تمام بسته های اطلاعاتی از یک مسیر واحد حرکت می‌کنند ، این تضمین وجود دارد که بسته‌های اطلاعاتی در مقصد به همان ترتیبی که در مبدأ ارسال شده‌اند ، دریافت شوند.

خصوصیات روش VC را می‌توان به صورت زیر خلاصه کرد:

- برای ارسال بسته‌های اطلاعاتی به آدرسهای جهانی مبدأ و مقصد نیازی نیست بلکه فقط به شمارهٔ VC نیاز است.
- برای هدایت بسته‌های اطلاعاتی از مبدأ به سمت مقصد نیاز به اجرای الگوریتم مسیریابی به ازای تک‌تک بسته‌ها نیست بلکه فقط یک جستجو در جدول هر مسیریاب انجام می‌شود.
- بسته‌ها الزاماً به ترتیب به مقصد خواهند رسید.
- احتمال گم شدن بسته‌ها ناشی از اشتباه در عمل مسیریابی در شبکه وجود ندارد.

در روش دیتاگرام هر ماشین میزبان پس از آنکه بسته‌ای را تولید کرد تحویل اولین مسیریاب در دسترس می‌دهد. مسیریابها مختارند بر اساس شرایط ترافیکی و توپولوژیکی زیرساخت ارتباطی شبکه ، مسیری را برای آن بسته انتخاب کرده و آن بسته را روی آن مسیر ارسال نمایند. بنابراین هیچ مسیر ثابت و از قبل مشخصی برای بسته‌های متوالی وجود ندارد. یعنی وقتی دو بسته از یک مبدأ تولید و به سمت یک مقصدِ واحد ارسال می‌شود ممکن است مسیرهای متفاوتی را طی نمایند؛ در ضمن ممکن است بسته‌ها به ترتیبی که تولید می‌شوند به مقصد نرسند.

خصوصیات روش دیتاگرام را می‌توان به صورت زیر خلاصه کرد:

- هر بستهٔ اطلاعاتی به آدرسهای جهانی مبدأ و مقصد نیازمند است.
- برای هر بسته باید مسیریابی جداگانه انجام شود.

- توزیع و هدایت بسته‌ها روی مسیرهای متفاوت ، بر اساس شرایط توپولوژیکی و ترافیکی لحظه‌ای شبکه خواهد بود.
- چون بسته‌ها به ترتیب نمی‌رسند باید فرآیندی برای تنظیم ترتیب بسته‌ها اتخاذ شود.
- در لایهٔ بالاتر باید نظارتهای ویژه بر گم شدن یا دوبله شدن بسته‌ها انجام شود.

۲-۱) انواع الگوریتمهای مسیریابی

الگوریتمهای مسیریابی را با دو دیدگاه میتوان دسته‌بندی کرد :
الف) از دیدگاه روش تصمیم‌گیری و میزان هوشمندی الگوریتم
ب) از دیدگاه چگونگی جمع‌آوری و پردازش اطلاعاتِ زیرساخت ارتباطی شبکه

- با دیدگاه اول الگوریتمهای مسیریابی را میتوان به دو دستهٔ **"ایستا"** و **"پویا"** تقسیم‌بندی کرد. در الگوریتمهای **ایستا** هیچ اعتنایی به شرایط توپولوژیکی و ترافیک لحظه‌ای شبکه نمی‌شود. معمولاً در این الگوریتمها برای هدایت یک بسته ، هر مسیریاب از جداولی استفاده می‌کند که در هنگام برپایی شبکه تنظیم شده و در طول زمان ثابت است. در هنگام وقوع هرگونه تغییر در توپولوژی زیرساختِ شبکه ، این جداول باید توسط مسئول شبکه بصورت دستی مجدداً تنظیم شود. اگرچه این الگوریتمها بسیار سریعند ولی چون ترافیک لحظه‌ای شبکه متغیر است ، نمی‌توانند بهترین مسیرها را انتخاب نمایند و هرگونه تغییر در توپولوژی زیرساخت ارتباطی شبکه ، یک مشکل عمده و جدی ایجاد خواهد کرد.

در الگوریتمهای **پویا** مسیریابی بر اساس آخرین وضعیت توپولوژیکی و ترافیک شبکه انجام می‌شود. جداول مسیریابی در این نوع الگوریتمها هر T ثانیه یکبار به‌هنگام می‌شود. این الگوریتمها بر اساس وضعیت فعلی شبکه تصمیم‌گیری می‌نمایند ولی ممکن است پیچیدگی این الگوریتمها به قدری زیاد باشد که زمان تصمیم‌گیری برای انتخاب بهترین مسیر ، طولانی شده و منجر به تاخیرهای بحرانی شده و نهایتاً به ازدحام بیانجامد؛ بهمین دلیل در مسیریابهای سریع از تکنیکهای چندپردازنده‌ای و پردازش موازی استفاده می‌شود.

- از دیدگاه دوم الگوریتمهای مسیریابی به دو دستهٔ "سراسری / متمرکز"[۱] و "غیرمتمرکز"[۲] تقسیم می‌شود.

در "الگوریتمهای سراسری" هر مسیریاب باید اطلاعات کاملی از زیرساخت ارتباطی شبکه داشته باشد. یعنی هر مسیریاب باید تمامی مسیریابهای دیگر ، ارتباطات بین آنها و هزینهٔ هر

[۱] Global Routing Algorithm
[۲] Decentralized Routing Algorithm

خط را دقیقاً شناسایی نماید. سپس با جمع‌آوری این اطلاعات "ساختمان دادهٔ" مربوط به گراف زیرساخت شبکه را تشکیل بدهد. در چنین شرایطی برای یافتن بهترین مسیر بین هر دو مسیریاب ، از الگوریتمهای کوتاهترین مسیر نظیر "الگوریتم دایجکسترا"[۱] استفاده می‌شود. به چنین الگوریتمهایی که برای مسیریابی به اطلاعات کاملی از زیرساخت شبکه و هزینهٔ ارتباط بین هر دو مسیریاب نیازمندند ، اختصاراً الگوریتمهای LS[۲] گفته می‌شود و در مسیریابهای مدرن و جدید از آن استفاده می‌شود.

در الگوریتمهای "غیر متمرکز" ، مسیریاب اطلاعات کاملی از زیرساخت شبکه ندارد بلکه فقط قادر است هزینهٔ ارتباط با مسیریابهایی که بطور مستقیم و فیزیکی با آنها در ارتباط است محاسبه و ارزیابی نماید. سپس در فواصل زمانی منظم ، هر مسیریاب جدول مسیریابی خود را برای مسیریابهای مجاور ، ارسال می‌نماید. مسیریاب با دریافت این جداول و مقادیری که خودش مستقیماً اندازه‌گیری کرده ، با یک الگوریتم بسیار ساده جدول خودش را به‌هنگام می‌نماید و برای هدایت هر بسته ، از آن استفاده می‌کند. در این الگوریتمها برای مسیریابی هر بسته ، فقط یک جستجو در جدول مسیریابی کافی است و در نتیجه پیچیدگی زمانی بسیار مناسبی دارد چراکه درگیر اجرای الگوریتمهای وقتگیری شبیه "دایْجْکِسترا" نخواهند شد. به این نوع الگوریتمها به اختصار "الگوریتمهای DV"[۳] گفته می‌شود.

این روشها را با تفصیل بیشتری بررسی می‌کنیم ولی قبل از آن یکی از روشهای ایستا را که در برخی از موارد خاص کاربرد دارد ، معرفی می‌نماییم.

۳-۱) روش ارسال سیل‌آسا[۴]

این روش که برای ارسال بسته‌های همگانی (فراگیر[۵]) کاربرد دارد سریعترین الگوریتم برای ارسال اطلاعات به یک مقصد در شبکه به شمار می‌رود. طریقهٔ ارسال در این روش آنست که هر مسیریاب با دریافت این‌گونه بسته‌ها موظف است آنرا روی تمامی مسیرهای خروجی خود (به غیر از مسیری که بسته را از آن دریافت کرده است) ارسال نماید. در چنین حالتی این تضمین وجود دارد که اولاً هر بستهٔ اطلاعاتی به تمامی مسیریابهای زیرشبکه خواهد رسید. ثانیاً هر بسته در سریعترین زمان ممکن به مقصد می‌رسد.

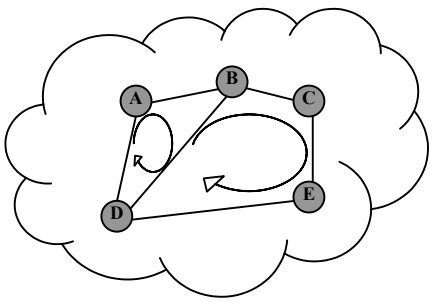
---

[۱] Dijkstra Shortest Path Algorithm
[۲] Link State Algorithms
[۳] Distance Vector Algorithms
[۴] Flooding Algorithm
[۵] Broadcast

از روش سیل‌آسا در موارد خاص و برای ارسال پیامهای فراگیر و کنترلی (مثل اعلام جداول مسیریابی) استفاده می‌شود؛ زیرا استفاده از این روش ، کل شبکه را در ترافیک زائد و بیهوده غرق خواهد کرد و بنابراین روشی قابل اتکا و عمومی برای مسیریابی نخواهد بود.

روش سیل‌آسا مشکل عمده‌ای دارد که باید رفع شود:

اگر قاعده براین باشد که همهٔ مسیریابها یک بستهٔ نوع فراگیر را روی تمامی خروجیهای خود ارسال نمایند ، ممکن است پس از چند لحظه خودشان آن بسته را دریافت کرده و چون مجدداً آنرا روی خروجی‌های خود ارسال می‌کنند این عمل تا بی‌نهایت ادامه خواهد یافت و در یک دور باطل کلّ شبکه از این بسته‌ها پر شده و روند ارسال هیچگاه متوقف نمی‌شود و عملاً شبکَه از کار خواهد افتاد. در شکل (۳-٤) حلقه‌های موجود در زیرساخت ارتباطی شبکه باعث شده است که روند تکرار بسته‌های فراگیر هیچگاه خاتمه نیابد.



شکل (۳-٤) حلقه‌های بینهایت در روش سیل آسا

برای رفع این مشکل دو راه حل وجود دارد:

- قرار دادن شمارهٔ شناسایی برای هر بسته[1]: در این روش برای هر بستهٔ فراگیر ، یک شمارهٔ منحصر به فرد و یکتا درج میشود و مسیریابی که بسته‌ای را یکبار دریافت کند شمارهٔ آنرا در جدولی ثبت می‌نماید. با دریافت یک بستهٔ فراگیر ، شمارهٔ شناسایی آنرا در جدول جستجو می‌کند و در صورت وجود ، آنرا حذف می‌نماید.
- قرار دادن طول عمر برای بسته‌ها : در این روش یک فیلد شمارنده در سرآیند بسته قرار داده می‌شود و به ازای عبور بسته از هر مسیریاب یکی از آن کم شده و وقتی این شماره به صفر رسید آن بسته از شبکه حذف خواهد شد.

[1] Selective Flooding

۲) الگوریتمهای LS

در الگوریتمهای LS هر مسیریاب باید پنج عمل زیر را انجام بدهد:

الف) مسیریابهای مجاور خود را که بصورت فیزیکی به آنها متصل است شناسایی کرده و آدرس آنها را بدست آورد.

ب) تاخیر (یا بطور کلی هزینهٔ) مسیریابهای مجاور خود را اندازه گیری نماید.

ج) یک بسته بسازد و تمام اطلاعاتی که از مسیریابهای مجاور خود دارد در آن قرار بدهد.

د) بستهٔ ساخته شده را به روش "سیل‌آسا" برای تمامی مسیریابهای شبکه ارسال نماید و همچنین بسته‌هایی را که از مسیریابهای دیگر می‌رسد دریافت و ذخیره کند.

ه) با استفاده از الگوریتمی مناسب ، بهینه‌ترین مسیر را بین هر دو مسیریاب در شبکه ، پیدا نماید.

هر یک از مراحل پنج‌گانهٔ فوق را بطور مجزا توضیح می‌دهیم:

۲-۱) شناسایی مسیریابهای مجاور

وقتی یک مسیریاب شروع به کار می‌کند (به عبارت ساده بوت می‌شود) اولین کاری که باید انجام بدهد شناسایی مسیریابهای مجاور خود می‌باشد. مسیریاب این کار را با ارسال یک بستهٔ خاص به نام "بستهٔ سلام" یا HELLO Packet روی تمامی خروجی‌های خود انجام می‌دهد. مسیریابهایی که از طریق یک کانال فیزیکی مستقیم به آن وصلند ضمن پاسخ به این بسته آدرس جهانی (آدرس IP) خود را اعلام می‌نمایند. پس از دریافت بسته‌های پاسخ این اطلاعات در جدولی درج می‌شود.

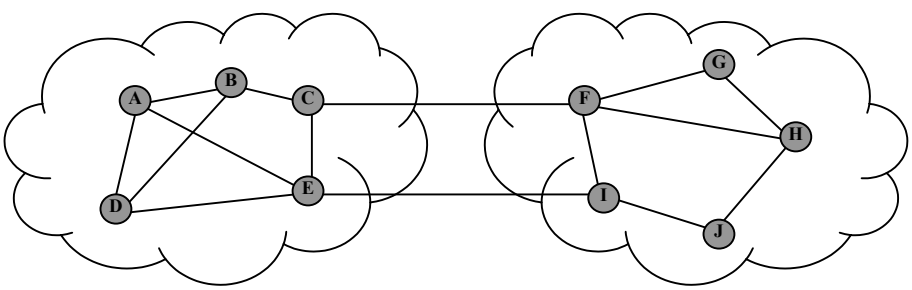
۲-۲) اندازه‌گیری هزینه

هر مسیریاب موظف است تاخیر هر یک از خطوط خروجی خود را اندازه‌گیری نماید؛ مسیریاب این کار را از طریق ارسال یک بستهٔ خاص به نام Echo Packet روی تمام خطوط خروجی خود انجام می‌دهد. تمام مسیریابهای گیرندهٔ این بسته ، با ارسال بسته‌ای به نام Echo Reply به فرستنده پاسخ می‌دهند. اگر مسیریاب موظف باشد که با دریافت بستهٔ Echo خارج از نوبت و به سرعت به آن پاسخ بدهد ، "زمان رفت و برگشت"[1] این بسته فقط تاخیر فیزیکی بین دو مسیریاب را به عنوان معیار هزینه مشخص می‌کند؛ مسیریاب این زمان را با استفاده از یک زمان‌سنج اندازه‌گیری کرده و آنرا بر ۲ تقسیم و در جدولی درج می‌نماید. در این حالت

[1] Round Trip Time

"میزان بار" و زمان انتظار بسته‌های به صف شده و منتظرِ پردازش در مسیریاب ، به حساب نمی‌آید که چندان صحیح نیست. مسیریاب می‌تواند بسته‌های Echo را همانند بسته‌های معمولی به انتهای صف بفرستد و پس از فرا رسیدن نوبت پردازشِ بسته ، به آن پاسخ بدهد؛ در این حالت معیار دقیقتری از تاخیر بدست می‌آید.

لازم به ذکر است که برای برخی از کانالهای فیزیکی مثل کانالهای ماهواره‌ای زمان تاخیر انتشار سیگنال بسیار زیاد و نامتعارف است چراکه زمان ارسال یک سیگنال به ارتفاع حدود ۳۷۰۰۰ کیلومتری[1] و برگشت آن به زمین چیزی حدود ۲۷۰میلی‌ثانیه طول میکشد. بنابراین ارسال یک بستهٔ Echo و برگشتِ پاسخ آن ، با چشمپوشی از زمان تاخیرِ پردازش ، چیزی حدود ۵۴۰ میلی ثانیه خواهد شد که در دنیای شبکه‌های کامپیوتری زمان بسیار زیادی محسوب می‌شود. بهترین حالت ، کانالهای فیبر نوری هستند که به ازای ۱۰۰۰کیلومتر حدود ۳ میلی ثانیه تاخیر دارند. بنابراین نمی‌توان از تاخیر فیزیکی انتشار سیگنال چشمپوشی کرد.
از طرف دیگر تاخیر پردازش برای مسیریابهایی که در برخی از زمانها با ترافیک بالایی روبرو می‌شوند ممکن است از چند ده میلی ثانیه تا چندین ثانیه! طول بکشد تا جایی که حتی طول عمر بسته به پایان رسیده و منجر به حذف آن شود. بنابراین زمان پردازش را نیز باید در معیار هزینه به حساب آورد ولی منظور کردن این پارامتر در معیار هزینه مشکلی دیگر پدید می‌آورد که برای رفع آن باید الگوریتم را پیچیده‌تر کرد. برای تشریح این مسئله به شکل (٤-٤) دقت کنید.



شبکهٔ I **شبکهٔ II**

شکل (٤-٤) دو شبکه متفاوت که با دو کانال ارتباطی به هم متصلند

[1] **به ارتفاع حدود ۳۷۰۰۰ کیلومتری از سطح زمین که ماهواره‌های مخابراتی در آن مدار قرار می‌گیرند مدار Geosynchrone گفته می‌شود.**

در شکل (٤-٤) تمام مسیریابها زمان تاخیر انتشار و میزان بار (زمان انتظار پردازش برای هر بسته) را به عنوان پارامتر هزینه در نظر می‌گیرند. فرض کنید تمامی مسیریابها با اجرای الگوریتم کوتاهترین مسیر ، مسیر بهینه از شبکهٔ I به شبکهٔ II را از طریق خط CF تشخیص بدهند و تمامی بسته‌ها را به این مسیر هدایت نمایند؛ چراکه این خط کمترین تاخیر و ترافیک را داشته است. با این کار در عرض چند ثانیه خط CF با ترافیک زیادی روبرو شده و تاخیر آن زیاد خواهد شد. با به‌هنگام شدن جدول مسیریابی ، کانال CF یک خط با تاخیر زیاد و "بد" گزارش می‌شود و از آن به بعد تمامی مسیریابها از خط EI بعنوان بهترین مسیر از شبکهٔ I به شبکهٔ II استفاده خواهند کرد. مجدداً خط EI با بار زیادی مواجه شده و به بدترین مسیر تبدیل می‌شود و این روند "نوسان"[1] تا بینهایت تکرار خواهد شد. این قضیه بسیار طبیعی است که اگر همهٔ مسیریابها از بهترین مسیر برای هدایت بسته‌ها استفاده کنند ، پس از مدت کوتاهی بهترین مسیر به بدترین مسیر تبدیل خواهد شد. برای اصلاح این مشکل میتوان یا از معیار زمان انتظار و پردازش چشمپوشی کرد یا آنکه روشی انتخاب کرد که درصدی از ترافیک بسته‌ها روی خطوط غیر بهینه توزیع شود. مثلاً وقتی مسیریاب C بهترین کانال را خط CF تشخیص میدهد ۷۰٪ از بسته ها را روی CF و ۳۰٪ باقیمانده را روی خط CE ارسال کند تا از طریق EI به سمت شبکهٔ II هدایت شوند. پیاده‌سازی چنین روشی پیچیدگی زمانی الگوریتم را افزایش می‌دهد.

۳-۲) تشکیل بسته‌های LS

هر مسیریاب موظف است پس از جمع‌آوری اطلاعات لازم از مسیریابهای مجاور خود بسته‌ای از این اطلاعات تشکیل بدهد؛ به این بسته ، "بستهٔ LS"[2] گفته می‌شود. در حالت کلّی این بسته باید شامل اطلاعات زیر باشد:

الف) آدرس جهانی مسیریاب تولیدکنندهٔ بسته

ب) یک شمارهٔ ترتیب (تا بسته‌های تکراری از بسته‌های جدید تشخیص داده شوند.)

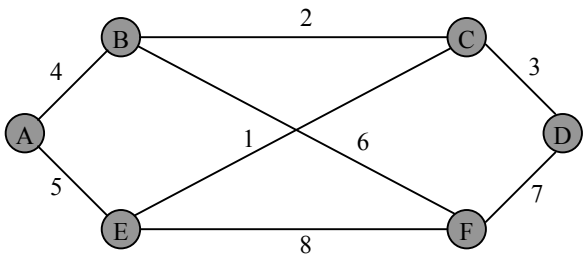
ج) طول عمر بسته (تا اطلاعات بسته ، زمان انقضای اعتبار داشته باشد.)

د) آدرس جهانی مسیریابهای مجاور و هزینهٔ تخمینی

برای روشن شدن قضیه به شکل (٤-٥) دقت کنید. در این شکل شش مسیریاب A تا F و خطوط ارتباطی مابین آنها و هزینه هر خط نشان داده شده است. قالب کلّی جدولی که هر

[1] Oscillation
[2] Link State Packet

مسیریاب باید بسازد در شکل (٦-٤) آورده شده است. ساختن این بسته‌ها مشکل نیست بلکه قسمت مشکل مسئله ، زمان تولید و توزیع آنها بر روی شبکه میباشد.
یک راه حل آنست که بسته‌ها در فواصل زمانی منظم برای مسیریابهای دیگر ارسال شود؛ در این حالت در بازه‌های زمانی مشخص ، کلّ شبکه برای لحظاتی غرق در بسته‌های LS خواهد شد. راه حل دوم آنست وقتی این جداول ارسال شوند که یک تغییر اساسی در زیرساخت شبکه رخ بدهد (مثل اضافه شدن یا حذف یک مسیریاب از شبکه).
وظیفهٔ فیلدهای "شماره ترتیب" و "طول عمر" در بخش بعدی مشخص شده است.



شکل (٥-٤) یک زیرساخت از یک شبکهٔ فرضی

A	
Seq.	
Age	
B	4
E	5

B	
Seq.	
Age	
A	4
C	2
F	6

C	
Seq.	
Age	
B	2
D	3
F	1

D	
Seq.	
Age	
C	3
F	7

E	
Seq.	
Age	
A	5
C	1
F	8

F	
Seq.	
Age	
B	6
D	7
E	8

شکل (٦-٤) بسته‌های LS

۴-۲) توزیع بسته‌های LS روی شبکه

یکی از مسائل مهم در این نوع الگوریتمِ مسیریابی ، چگونگی توزیع بسته‌های LS روی شبکه است. قاعدهٔ کلی ارسال بسته‌های LS ، ارسال به روش "سیل‌آسا" است. برای آنکه این بسته‌ها در یک حلقهٔ بینهایت تکرار نشوند ، هر بسته دارای یک شمارهٔ ترتیب است که با ورود

به هر مسیریاب ابتدا بررسی می‌شود که آیا این بسته قبلاً دریافت شده است یا آنکه یک بستهٔ جدید است. در صورت جدید بودن ، مسیریاب ضمن درج اطلاعات درون بسته در یک جدول موقت ، آنرا روی تمامی خروجی‌های خود (به غیر از کانالی که بسته را از آن دریافت کرده) ارسال می‌نماید. بنابراین هر مسیریاب موظف است که شمارهٔ ترتیب بسته‌های LS خود را که در هر مرحله ارسال می‌کند ، حفظ نماید و در هر بار ارسال ، یکی به آن اضافه کرده تا در مراحل بعدی ، بسته‌های LS با شمارهٔ جدید و غیر تکراری ارسال شود. شمارهٔ ترتیب معمولاً ۳۲ بیتی است و اگر شمارهٔ بسته‌ها از صفر شروع و به فرض در هر ثانیه یکی به آن اضافه شود تکرار شمارهٔ یک بسته ، ۱۳۷ سال طول می‌کشد!

وقتی مسیریاب بسته‌ای را با یک شمارهٔ خاص دریافت کرد دیگر بسته‌های با شمارهٔ کوچکتر از آن شماره را دریافت نخواهد کرد. بعنوان مثال اگر مسیریابی یک بستهٔ LS را از یک مسیریاب خاص با شمارهٔ ترتیب ٦٥٤٣٢٠ دریافت کند و آنرا در جدولش درج نماید ، از آن به بعد بسته‌هایی را که شمارهٔ کمتر از ٦٥٤٣٢١ داشته باشد ، از آن مسیریاب قبول نخواهد کرد. همین مسئله مشکل بزرگی برای مسیریابهایی که به ناگاه از کار می‌افتند و می‌خواهند مجدداً به شبکه وارد شوند ، پدید خواهد آورد؛ چونکه یک مسیریاب که از شبکه خارج شده و می‌خواهد مجدداً به شبکه وارد شود مجبور است شمارهٔ بسته‌ها را از صفر شروع کند و هیچ مسیریابی آنرا نخواهد پذیرفت. برای رفع این مشکل برای هر بسته یک طول عمر در نظر گرفته می‌شود که به ازای هر ثانیه یک واحد از آن کم می‌شود؛ هرگاه مسیریابی بسته‌ای را دریافت و آنرا در جدولی درج نماید ولی در خلال طول عمر بسته (مثلاً ۱۰ دقیقه معادل ٦٠٠ ثانیه) بستهٔ جدیدی دریافت نکند آن مسیریاب از جدول مسیریابی حذف خواهد شد. با این قاعده وقتی مسیریابی که از جدول حذف شده ، بخواهد به شبکه برگردد بسته‌های ارسالی از طرف او با هر شماره‌ای دریافت خواهد شد.با این روش مشکل دریافت بسته‌های تکراری حل خواهد شد.

مسئلهٔ بحرانی دیگر آنست که اگر یک مسیریاب به هر دلیلی اطلاعات غلط ارائه کند ، کلّ مسیریابی شبکه با اشکال مواجه شده و تمام جداول مسیریابی با اطلاعات آلوده تنظیم خواهد شد. این مشکل زمانی بروز می‌کند که یک مسیریاب به اشتباه اعلام کند که کانالی فیزیکی با مسیریاب دیگر دارد در حالی که نداشته باشد؛ یا کانالی فیزیکی با یک مسیریاب داشته باشد ولی اعلام نکند. گاهی این مشکلات بصورت عمدی توسط اخلالگران بوجود می‌آید چراکه ایجاد بسته‌های LS بصورت مصنوعی و تزریق آن به شبکه توسط کاربران فقط نیاز به برنامه‌نویسی دارد. برای رفع این مشکل ، در مسیریابهای مدرن بسته‌های LS زمانی پذیرفته می‌شود که قبل از آن ، هویت ارسال کنندهٔ بسته احراز شود.

۵-۲) محاسبهٔ مسیرهای جدید

وقتی یک مسیریاب بسته‌های LS را از تمامی مسیریابهای شبکه دریافت کرد ، می‌تواند ساختمان دادهٔ گراف زیرشبکه را تشکیل داده و بر اساس آن اقدام به پیدا کردن بهترین مسیرها (یعنی با کمترین هزینه) بین هر دو گره بنماید.

بگونه‌ای که اشاره شد برای یافتن بهترین مسیر بین دو گره در یک گراف ، می‌توان از الگوریتم دایجْکِسترا بهره برد. معرفی این الگوریتم در اینجا خالی از لطف نیست چراکه بسیاری از مسیریابهای مدرن از آن استفاده می‌کنند؛ این الگوریتم در سال ۱۹۵۹ معرفی شد.

فرض کنید الگوریتم بخواهد بهترین مسیر بین گره‌های V1 و V2 را پیدا کند.

- گراف زیرشبکه را تشکیل داده و گره‌های V1 و V2 را مشخص کنید. فرض کنید گراف زیرشبکه با استفاده از "ماتریس همجواری"[1] نشان داده شود. در این ماتریس عنصر a[i.j] هزینهٔ رسیدن از گره Vi به گره Vj می‌باشد. اگر هیچ مسیر مستقیمی بین گره Vi و گره Vj وجود نداشته باشد مقدار a[i.j] ، بینهایت در نظر گرفته می‌شود.

```
int dist[MAX_NODES] [MAX_NODES];
```

- برای هر گره از گراف یک "رکورد حالت" ایجاد کنید که شامل سه فیلد باشد:
  - فیلد اول نشان‌دهندهٔ گره قبلی در مسیر است؛ این فیلد را predecessor می‌نامیم.
  - فیلد دوم نشان‌دهندهٔ مجموع هزینهٔ رسیدن از گره V1 به این گره می‌باشد؛ این فیلد را length می‌نامیم.
  - فیلد سوم حالت گره را مشخص می‌کند که می‌تواند دو حالت "دائمی" (Permanent) و "موقت" (Tentative) داشته باشد. این فیلد را "برچسب" (label) می‌نامیم.

این رکوردها را می‌توان در زبان C بصورت زیر تعریف کرد:

```
struct state {
    int predecessor;
    int length;
    enum {permanent, tentative} label;
    } State[MAX_NODES];
```

- برای تمامی گره‌ها ، متغیرهای حالت را بصورت زیر مقداردهی اولیه نمایید:

```
State[i].prdecessor=NULL;
State[i].length=INFINITY;
```

[1] Adjacency Matrix

```
State[i].label=tentative;
```

- گره t=V1 را به عنوان نقطهٔ کار انتخاب نمایید.
- (*) برچسب گره **t** را به صورت دائمی درآورید. گره‌ای که برچسب آن دائمی است دیگر هیچگاه برچسب آن تغییر نخواهد کرد.
- برای تمامی "گره‌های Vi مجاور با **t** که برچسبشان موقت" است ، رکوردهای حالت باید طبق رابطهٔ زیر تغییر کنند.

```
if (State[t].length+dist[t][i]<State[i].length) {
    State[i].prdecessor=t;
    State[i].length= State[t].length+dist[t][i];
    }
```

یعنی برای تمامی گره‌های Vi مجاور با **t** که برچسبشان موقت است ، بررسی می‌شود که اگر هزینهٔ رسیدن از V1 به Vi از طریق گره **t** کمتر از هزینهٔ مسیر قبلی آنها به V1 است ، اصلاح لازم روی رکورد حالت آن گره انجام شود.

- از بین تمامی گره‌هایی که برچسبشان موقتی است گره‌ای که کمترین هزینه را تا V1 دارد پیدا کرده و بعنوان نقطهٔ کار t انتخاب کنید.
- اگر **t** به گره مقصد (V2) نرسیده است به مرحلهٔ (*) برگردید.
- اگر **t** به گره مقصد رسید ، از آن شروع کرده و گره ماقبل آنرا از رکورد حالت استخراج کرده و اینکار را تکرار کنید تا به گره مبدأ برگردید.

برای درک راحتتر الگوریتم فوق به مثال شکل (۷-٤) دقت کنید. در این مثال فرض شده است می‌خواهیم بهترین مسیر از A به D را پیدا کنیم.

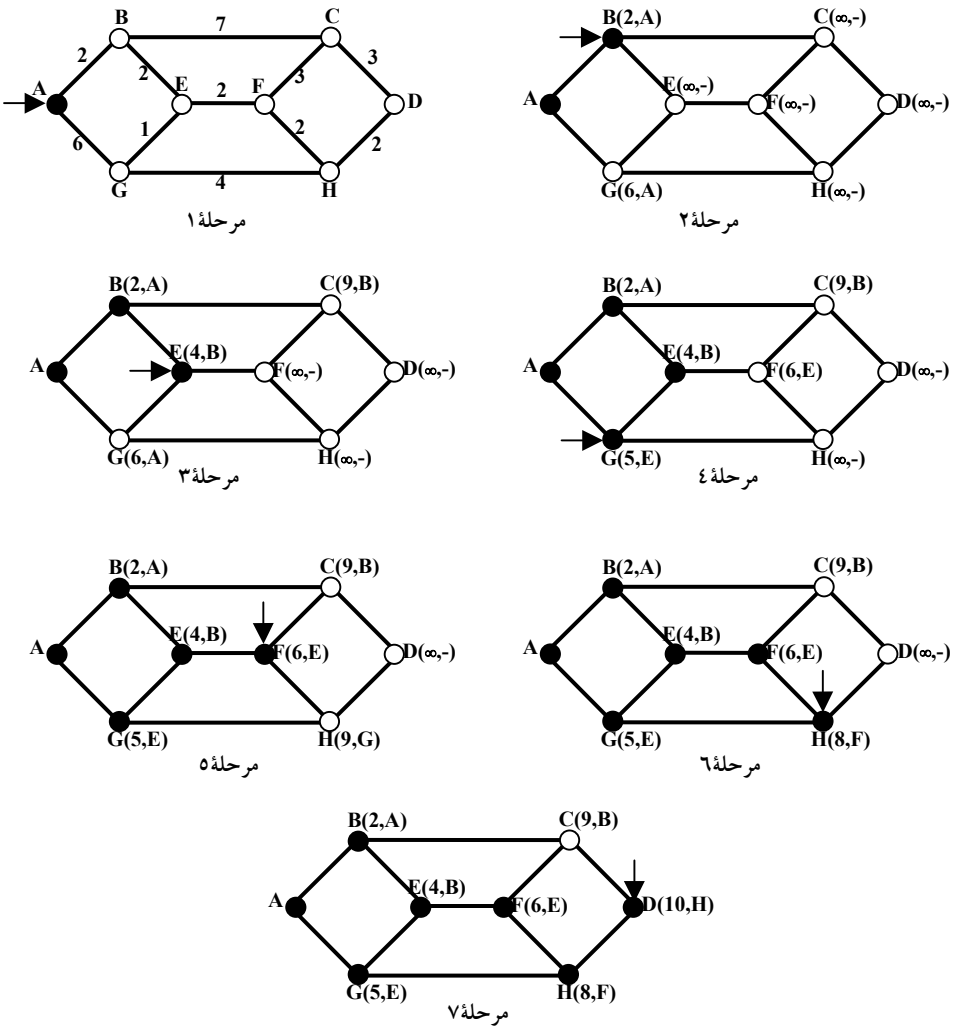
در مرحلهٔ ۱ گراف زیرساخت ارتباطی یک شبکهٔ فرضی به همراه هزینهٔ هر ارتباط به تصویر کشیده شده است. گره شروع به عنوان نقطهٔ کار انتخاب شده و حالت آن بصورت "دائمی" علامت خورده است. (گره‌های دائمی با دایرهٔ توپر نشان داده شده است)

در مرحلهٔ ۲ هزینهٔ گره‌های مجاور A (یعنی B و G) محاسبه شده و رکورد حالت آنها اصلاح شده است. سپس از بین تمام گره‌های با حالت "موقت" ، گره B که کمترین هزینه را تا A داشته به عنوان گره نقطهٔ کار انتخاب و حالت آن بصورت دائمی علامت زده شده است.

در مرحلهٔ ۳ هزینهٔ گره‌های مجاور B که علامت موقتی دارند (یعنی C و E) تا A محاسبه شده و رکورد حالت آنها اصلاح شده است. از بین تمام گره‌های با حالت "موقت" ، گره E که

کمترین هزینه را تا A داشته به عنوان گره نقطهٔ کار انتخاب و حالت آن بصورت دائمی علامت زده شده است.

در مرحلهٔ ٤ هزینهٔ گره‌های مجاور E که علامت موقتی دارند (یعنی G و F) تا A محاسبه شده و رکورد حالت آنها اصلاح شده است. (به چگونگی اصلاح رکورد حالت در گره G در این مرحله دقت کنید) از بین تمام گره‌های با حالت "موقت" ، گره G که کمترین هزینه را تا A داشته به عنوان گره نقطهٔ کار انتخاب و حالت آن بصورت دائمی علامت زده شده است.



شکل (۷-٤) مراحل پیدا کردن کوتاهترین مسیر از A به F

در مرحلهٔ ۵ هزینهٔ گره‌های مجاور G که علامت موقتی دارند (یعنی H) تا A محاسبه و رکورد حالت آن اصلاح شده است. از بین تمام گره‌های با حالت "موقت" ، گره F که کمترین هزینه را تا A داشته به عنوان گره نقطهٔ کار انتخاب و حالت آن بصورت دائمی علامت زده شده است. (دقت کنید که F مجاور G نیست)

در مرحلهٔ ٦ هزینهٔ گره‌های مجاور F که علامت موقتی دارند (یعنی C و H) تا A محاسبه و رکورد حالت گره H اصلاح شده است. از بین تمام گره‌های با حالت "موقت" ، گره H که کمترین هزینه را تا A داشته به عنوان گره نقطهٔ کار انتخاب و حالت آن بصورت دائمی علامت زده شده است.

در مرحلهٔ ۷ هزینهٔ گره‌های مجاور H که علامت موقتی دارند (یعنی D) تا A محاسبه و رکورد حالت آن اصلاح شده است. از بین تمام گره‌های با حالت "موقت" ، گره D که کمترین هزینه را تا A داشته به عنوان گره نقطهٔ کار انتخاب می‌شود ولی چون D گره مقصد است الگوریتم در این مرحله پایان می‌یابد. برای پیدا کردن مسیر ، از رکورد حالتِ گره D شروع کرده و گره قبلی آن را پیدا می‌کنیم ، از این گره مجدداً گره قبلیِ آن پیدا می‌شود و این کار ادامه می‌یابد تا به نقطهٔ شروع برسیم.

در جدول (۷-٤) زیربرنامهٔ کوتاهترین مسیر بر طبق الگوریتم دایْجْکِسترا به زبان C ارائه شده است.

برای شبکه‌هایی که دارای n مسیریاب و هر مسیریاب دارای حداکثر k کانال ورودی/خروجی است ، در بدترین حالت به فضای n×k رکورد اطلاعاتی برای ذخیره‌سازی جداول LS ، نیاز خواهد بود که برای شبکه‌های وسیع با هزاران مسیریاب ، مشکل‌ساز خواهد بود.

در بسیاری از مسیریابهای مدرن از روشهای مبتنی بر الگوریتم LS ، استفاده شده است. برخی از این پروتکلها عبارتند از:

- پروتکل OSPF[1] در برخی مسیریابهای تجاری CISCO
- پروتکل IS-IS[2] که توسط DECnet ارائه شده است.
- پروتکل NLSP که توسط شرکت Novell معرفی و ارائه شده است.
- پروتکل CLNP که توسط ISO معرفی و ارائه شده است.

در ادامه الگوریتمهای DV را معرفی خواهیم کرد.

[1] Open Shortest Path First
[2] Intermediate System-Intermediate System

```
 #define MAX_NODES 100  /*تعریف تعداد حداکثر گره‌های گراف*/
#define INFINITY 1000000000 /*تعریف یک عدد بسیار بزرگ به عنوان هزینهٔ بی‌نهایت */
int n, dist[MAX_NODES][ MAX_NODES]; /*تعریف ماتریس همجواری */
                                    /* n تعداد گره‌های گراف می‌باشد */
void shortest_path(int s, int t, int path[])
    { struct state {
          int predecessor; /*گره قبلی در مسیر*/
          int length; /*هزینهٔ گره تا مبدأ */
          enum {permanent, tentative} label; /*حالت گره */
          } State[MAX_NODES]; /*تشکیل رکوردهای حالت برای هر گره درون یک آرایه*/
int  i, k, min;
struct state *p;
for(p  = &State[0];p<&State[n];p++) { /*مقداردهی اولیه به رکوردهای حالت */
      p->predecessor=NULL;
      p->length=INFINITY;
      p->label=tentative;
}
State[t].length=0;
State[t].label=permanent;
k=t;  /* k گره نقطهٔ کار برای شروع می‌باشد */
do {
    for(i=0;i<n;i++)
      if(dist[k][i]!=0 && State[i].label==tentative){
         if(State[k].length+dist[k][i]<State[i].length){
                                          /*آیا مسیر بهتری از گره فعلی به گره مبدا وجود دارد؟*/
           State[I].predecessor=k;
           State[i].length= State[k].length+dist[k][i];
        }
      }
k=0;   min=INFINITY; /*یافتن گره‌ای از بین گره‌های با علامت "موقتی" که کمترین هزینه را دارد */
    for(i=0;i<n;i++)
      if(State[i].label==tentative && State[i].length<min){
      min= State[i].length;
      k=i;
      }
      State[k].label=permanent;
      } while (k!=s);
/* path[] قرار دادن مسیر بهینه از آخر به اول در آرایهٔ */
      i=0;  k=s;
      do {
      path[i++]=k; k=State[k].predecessor;
      } while (k>=0);
      }
```

جدول (۷-٤) الگوریتم دایجکسترا برای پیدا کردن بهترین مسیر بین دو گره در یک گراف

۳) الگوریتمهای DV[1]

یکی از روشهای پویا در مسیریابی ، روش "بردار فاصله" یا DV است که در سالهای اولیه راه اندازی شبکهٔ ARPA مورد استفاده قرارگرفت و پس از عمومی شدن اینترنت تحت نام پروتکل RIP عرضه شد و هنوز هم در مسیریابهای کوچک مورد استفاده قرار می‌گیرد و در نسخه های جدید ویندوز NT پشتیبانی می‌شود . نامهای متفاوتی برای این روش ارائه شده که همهٔ آنها از یک الگوریتم استفاده می‌کنند. این نامها عبارتند از:

- پروتکل RIP
- الگوریتم مسیریابی Bellman–Ford
- الگوریتم مسیریابی Ford-Fulkerson
- الگوریتم Distance Vector Routing

در این روش بر خلاف الگوریتم LS ، هر مسیریاب بدون آنکه اطلاعی از هزینهٔ خطوط ارتباطی در زیرشبکه داشته باشد ، جدولی را در حافظه خود نگه می‌دارد که جدول مسیریابی[2] نام دارد. در این جدول به ازای هر مسیریاب در زیرشبکه یک رکورد وجود دارد؛ هر رکورد دارای دو فیلد مجزا است:

الف: **فیلد مسیر** : این فیلد خط خروجی مناسب برای رسیدن به یک مسیریابِ خاص در شبکه را مشخص می‌کند.

ب: **فیلد مقدار تقریبی هزینه** : این فیلد هزینهٔ تقریبی رسیدن یک بسته تا مسیریابِ مقصد را مشخص می‌نماید.

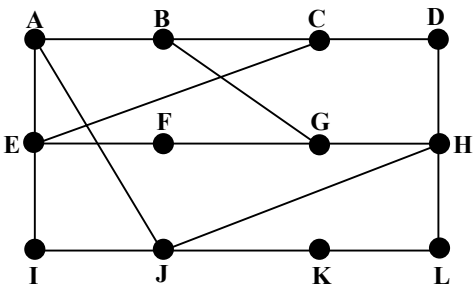
برای روشن شدن قضیه به شکل (۸–٤) که زیرساخت ارتباطی یک شبکهٔ فرضی را نشان می‌دهد ، دقت کنید. در این مثال تعداد ۱۲ مسیریاب با نامهای A تا L زیرساخت ارتباطی شبکه را تشکیل داده‌اند؛ کانالهایی که بین این ۱۲ مسیریاب وجود دارد در شکل مشخص است ولی هیچیک از مسیریابها اطلاعی از هزینهٔ هر یک از خطوط ارتباطی ندارند ، بهمین دلیل مقادیر هزینهٔ هر خط در شکل نشان داده نشده است. جدول (۹–٤) جدول مسیریابی مربوط به J است و هر سطر نشان می‌دهد که اگر J بخواهد بسته‌ای را به یک مسیریاب دیگر بفرستد از چه خطی باید استفاده کند. بعنوان مثال اگر مسیریاب J خواست برای G بسته‌ای ارسال کند ، با

[1] در برخی از کتابها ، الگوریتمهای DV به "الگوریتم بردار فاصله" ترجمه شده است.

[2] Routing Table

مراجعه به این جدول نتیجه می‌گیرد که باید آنرا به سمتِ H ارسال نماید و هزینهٔ تقریبی رسیدن بسته به G تقریباً ۱۸ است. حال شاید بپرسید در این روش معیار هزینه و واحد آن چیست؟ پاسخ آنست که معیار هزینه میتواند تاخیر یا "تعداد گام" (Hop) در نظر گرفته شود؛ در چنین حالاتی واحد تاخیر ، میلی‌ثانیه و واحد گام ، "تعداد" خواهد بود. در این مثال معیار هزینه ، زمانِ تاخیر بر حسب میلی‌ثانیه انتخاب شده است.

تا اینجا به این نکته اشاره کردیم که هر یک از مسیریابها جدولی در حافظه خود تشکیل می‌دهند ولی سوال اصلی اینجاست که این جداول چگونه ایجاد و به‌هنگام می‌شوند. چرا که در زمانهای متفاوت شرایط ترافیکی و توپولوژیکی شبکه عوض شده و بالطبع این جداول باید با زمان تغییر داده شود تا همیشه بهترین وضعیت را برای مسیریابی ارائه بدهد.



شکل (۸-٤) زیرساخت ارتباطی یک شبکهٔ فرضی با دوازده مسیریاب

	هزینهٔ تقریبی	خط
A	8	A
B	20	A
C	28	I
D	20	H
E	17	I
F	30	I
G	18	H
H	12	H
I	10	I
J	0	—
K	6	K
L	15	K

جدول (۹-٤) جدول مسیریابی مربوط به مسیریاب J

در روش DV اصول کار بصورت زیر خلاصه می‌شود:

- هر مسیریاب موظف است هزینهٔ خطوطی را که بصورت فیزیکی با مسیریابهای دیگر دارد ، محاسبه کرده و در جدول خود درج نماید. هزینهٔ خطوطی که مسیریاب با آنها در ارتباطِ مستقیم نیست ، در این جدول بینهایت در نظر گرفته می‌شود.
- هر مسیریاب موظف است در بازه‌های زمانی مشخص ، ستون هزینه از جدول مسیریابی خودش را برای مسیریابهای مجاور ارسال نماید، ("یعنی فقط برای مسیریابهائی که با آن در ارتباط است نه تمام مسیریابها "). بنابراین هر مسیریاب در فواصل T ثانیه‌ای ، اطلاعاتی را از مسیریابهای مجاور دریافت می‌کند که جدید است و می‌تواند بر اساس آن ، جدول مسیریابی خود را به‌هنگام کند.
- هر مسیریاب موظف است پس از دریافت جداول مسیریابی از مسیریابهای مجاور ، جدول خود را طبق یک الگوریتم بسیار ساده به‌هنگام نماید. (این الگوریتم با یک مثال ، تشریح خواهد شد)

برای مشخص شدن چگونگی به‌هنگام شدن جداول مسیریابی ، به مثال قبلی مراجعه کنید. فرض کنید مسیریابهای مجاور J (یعنی مسیریابهای A ، I ، H ، K) جداول مسیریابی خود را برای J ارسال کرده باشند. این جداول در شکل (۱۰-٤) نشان داده شده‌اند.

مسیریاب J می‌تواند تخمین بزند که از لحاظ زمانی تا مسیریابهای A ، I ، H و K چقدر تاخیر وجود دارد . این تخمین بسادگی از طریق ارسال بسته های Echo و دریافت پاسخ آن ومحاسبهٔ زمان رفت و برگشت آن امکانپذیر است . فرض کنید با این روش مسیریاب J مقادیر تاخیر را بصورت زیر ارزیابی کرده باشد:

- J تا A ← ۸ میلی‌ثانیه
- J تا I ← ۱۰ میلی‌ثانیه
- J تا H ← ۱۲ میلی‌ثانیه
- J تا K ← ٦ میلی‌ثانیه

حال فرض کنید پس از رسیدن چهار جدول فوق به J و اندازه گیری J از مقدار تاخیر تا مسیریابهای A ، I ، H و K ، مسیریاب J بخواهد در جدول مسیریابی خود ، بهترین کانال را برای ارسال بسته به هر یک از مسیریابهای A تا L بیابد. بعنوان مثال J می‌خواهد بداند بهترین مسیر برای رسیدن به G کدام است . ابتدا به جدول رسیده از A مراجعه می‌کند؛ A ادعا کرده است که برای رسیدن به G ، تاخیری معادل ۱۸ میلی‌ثانیه دارد. پس اگر J بخواهد ازطریق A بسته‌ای برای G بفرستد معادل ۲٦ میلی‌ثانیه تاخیر خواهد داشت؛ (یعنی ۸ میلی‌ثانیه از J به A و ۱۸ میلی‌ثانیه از A به G بنابر ادعای A). J این مقدار را بطور موقتی در حافظه ذخیره می‌نماید.

	A	I	H	K
A	0	24	20	21
B	12	36	31	28
C	25	18	19	36
D	40	27	8	24
E	14	7	30	22
F	23	20	19	40
G	18	31	6	31
H	17	20	0	19
I	21	0	14	22
J	9	11	7	10
K	24	22	22	0
L	29	33	9	9

شکل (۱۰-٤) جداول ارسالی توسط مسیریابهای مجاور J

حال به جدول رسیده از I مراجعه می‌کند. I ادعا کرده است که تا G معادل ۳۱ میلی‌ثانیه تاخیر دارد ، بنابراین اگر J بخواهد از طریق I به G بسته ای بفرستد معادل ٤۱ میلی‌ثانیه تاخیر خواهد داشت. (۱۰ میلی‌ثانیه از J به I و ۳۱ میلی ثانیه از I به G) این مقدار نیز بطور موقتی در حافظه ذخیره می‌شود.

در جدول رسیده از H ، تاخیر زمانی تا G معادل ٦ میلی‌ثانیه ادعا شده ، بنابراین اگر J بخواهد از طریق H به G بسته ای بفرستد معادل ۱۸ میلی‌ثانیه تاخیر خواهد داشت. (۱۲ میلی‌ثانیه از J به H و ٦ میلی ثانیه از H به G)

بهمین ترتیب هزینه رسیدن به G از طریق K معادل ۳۷ (۳۱+٦) میلی‌ثانیه محاسبه و در حافظه ذخیره می‌شود .

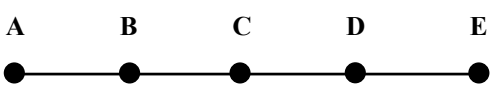
با محاسبات فوق J می‌تواند با پیدا کردن حداقل مقدار از بین مقادیر محاسبه شده ، بهترین مسیر برای ارسال یک بسته به G را پیدا کند. در مثال فوق از بین چهار مقدار ذخیره شده در حافظه ، هزینهٔ ارسال از طریق مسیریابِ H حداقل خواهد بود و تاخیری معادل ۱۸ میلی‌ثانیه خواهد داشت. مسیریاب J در جدول خَود در رکورد متناظر با G آدرس مسیریاب H و هزینهٔ ۱۸ را درج می‌کند.

برای تمام مسیریابهای دیگر این روند تکرار و بهترین مسیر برای رسیدن به یکایک آنها محاسبه شده و در جدول جدید درج خواهد شد.

این جدول تا زمان به‌هنگام‌سازی بعدی ، که جداول جدید از مسیریابهای مجاور می‌رسند قابل استناد بوده و بهترین مسیر را برای ارسال بسته‌ها تعیین می‌کند. بعنوان مثال برای ارسال بسته‌ای به مسیریاب F با استناد به جدول محاسبه شده ، باید آن بسته تحویل مسیریاب I شود.

این روش در عین سادگی، پویا است و تغییرات ترافیکی شبکه با زمان را در جداول مسیریابی دخالت خواهد داد. از طرفی حجم جدولی که بایستی هر مسیریاب در حافظه خود نگه دارد از درجه ۱ است یعنی به ازای n مسیریاب فقط n رکورد کافی است در حالی که برای نگهداری جداول مسیریابی در روش LS درجهٔ ۲ است.(یعنی در بدترین حالت به یک ماتریس با n×n رکورد نیاز می‌باشد)

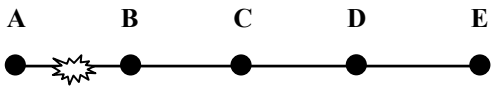
پروتکلهای DV از یک مشکل عمده رنج می‌برند و آن "عدم همگرایی سریع جداول مسیریابی" در هنگام خرابی یک مسیریاب یا یک کانال ارتباطی می‌باشد. این مشکل "شمارش تا بینهایت"[1] نام گرفته است. این اشکال زمانی پیش خواهد آمد که یکی از مسیریابها دچار خرابی شود یا آنکه مسیر ارتباطی او با دیگران قطع شود . بعنوان مثال شکل (۱۱-٤) را در نظر بگیرید؛ در این شکل ، A تا E مسیریابها هستند و هر کدام برای رسیدن به دیگری فقط یک مسیر در اختیار دارند. بعنوان مثال هزینه A تا B ، ۱ میلی ثانیه و از A تا E ، ٤ میلی ثانیه است. در حالت عادی جداول مسیریابی هر یک از مسیریابها ، طبق جداولِ شکل (۱۱-٤) تنظیم خواهد شد.



A	0,-	1,A	2,B	3,C	4,D
B	1,B	0,-	1,B	2,C	3,D
C	2,B	1,C	0,-	1,C	2,D
D	3,B	2,C	1,D	0,-	1,D
E	4,B	3,C	2,D	1,E	0,-

شکل (۱۱-٤) زیرساخت ارتباطی از یک شبکهٔ فرضی و جداول مسیریابی

[1] Count to infinity



هزینهٔ رسیدن به A در هنگام بروز خرابی	A	∞,A	2,B	3,C	4,D
هزینهٔ رسیدن به A پس از اولین بهنگام‌سازی	A	3,C	2,B	3,C	4,D
هزینهٔ رسیدن به A پس از دومین بهنگام‌سازی	A	3,C	4,B	3,C	4,D
هزینهٔ رسیدن به A پس از سومین بهنگام‌سازی	A	5,C	4,B	5,C	4,D
هزینهٔ رسیدن به A پس از چهارمین	A	5,C	6,B	5,C	6,D
هزینهٔ رسیدن به A پس از پنجمین بهنگام‌سازی	A	7,C	6,B	7,C	6,D
هزینهٔ رسیدن به A پس از ششمین بهنگام‌سازی	A	7,C	8,B	7,C	8,D
هزینهٔ رسیدن به A پس از بهنگام‌سازی n ام	A	….	….	….	….
∞	A	∞	∞	∞	∞

شکل (۱۲-٤) عدم همگرایی سریع جدول مسیریابی در هنگام وقوع یک خرابی

حال فرض کنید ناگهان خط ارتباطی A بـه B قطـع شـود. بنـابراین در ایـن حـالت هزینـهٔ مسیریاب B به A بینهایت خواهد شد و هزینهٔ ارسال بسته به A در جداول هر یک از مسیریابها ، طبق روند شکل (۱۲-٤) تغییر خواهد کرد. دقت کنید که پس آنکه ارتباط B با A قطع شد ، B در جدول خود ، هزینهٔ رسیدن به A را مقدار ∞ درج می‌کند ولـی پـس از گذشـت T ثانیـه جدول مسیریابی از C می‌رسد و C اعلام کرده که مقدار هزینه تـا A مقـدار ۲ اسـت زیـرا C نمی‌داند که کانال B به A قطع شده است و در نتیجه همان مقدار قبل از وقوع خرابی را اعـلام می‌کند. B که در جدولش مقدار هزینه تا A را بینهایت درج کرده است به خیال آنکه C بـه A مسیری مجزا دارد (با هزینه ۲) در جدول خـود هزینـه رسـیدن بـه A را از ∞ بـه ۳ تبدیـل می‌کند. (۱ واحد تا C و ۲ واحد از C به A طبق ادعای C) بنابراین B فرض کرده که از C به A کانالی مستقل وجود دارد و هزینه آنهم ۲ است در حالی که چنین فرضـی اشـتباه اسـت. در T ثانیه بعد مجدداً جداول مسیریابی بین همسایه ها رد و بدل می‌شود . C جـدول B را گرفتـه و متوجه می‌شود که هزینه رسیدن به A مقدار ۳ شده است بنـابراین بـا احتسـاب مقـدار جدیـد هزینه رسیدن از C به A در جدولش ٤ می‌شود . در T ثانیه بعد ، مسیریاب B با دریافت جدول مسیریابی C ، باز هم به اشتباه هزینه رسیدن بـه A را در جـدول خـودش ۵ درج می‌کنـد. (۱ واحد تا C + ٤ واحد از C به A طبق ادعای C). این روند تا بی نهایت ادامه دارد و بطور مداوم بین مسیریابها اطلاعات غلط در مورد A مبادله می‌شود . مشکل از آن جائی است کـه C ، D و E نمی‌دانند که تنها مسیرشان به A از طریق B است و به B اعلام می‌کنند که راهی به A دارنـد

و هزینهٔ آن را اعلام می‌کنند ، در حالیکه که تمامی این راهها همانی است که فعـلاً قطـع شـده است !

روشهای گوناگونی برای حل این مسأله پیشنهاد شده که عمدتاً پیچیده‌اند یا مقـرون بـه صرفـه نیستند. ساده تریـن راه‌حـل آن اسـت کـه وقتـی یـک مسـیریاب می‌خواهـد اطلاعـاتی را بـه همسایه‌هایش بدهد هزینه رسیدن به آنهایی را که قطعاً باید از همان مسیریاب بگذرند را اعلام نمی‌کند. (یا ∞ اعلام می‌کنند) بعنوان مثـال C چـون می‌دانـد مسـیر A از B مـی‌گـذرد وقتـی خواست جدول مسیریابی خود را به B اعلام کنـد هزینـه رسـیدن بـه A را همیشـه ∞ اعـلام می‌کند چرا که برای رسیدن به A قطعاً باید از B عبور کرد؛ در ایـن حـالت جـداول مسـیریابی سریعاً اصلاح خواهد شد.[1]

الگوریتمهای DV برای مسیریابی در یک شبکهٔ کوچک (با حداکـثر ۳۰ مسـیریاب) هنـوز کاربرد دارد ولی مسیریابهای جدید ، به دلیل نقص یاد شده و روشهای بهتری که ابداع شـده‌اند به سمت روشهای اصلاح شده تری مثل روش OSPF رفته‌اند . ایـن روشـها سـریعتر و دقیقـتر جداول مسیریابی را نسبت به تغییرات توپولوژیکی و ترافیکی شبکه تنظیم می‌کنند و در ضمـن می‌توانند بار ترافیک را روی چند مسیر که نزدیک به بهینگی هستند تقسیم نمایند چرا که اگـر همه بار روی بهترین مسیر ارسال شود در اندک زمانی بهترین تبدیل به بدترین مسـیر خواهـد شد.

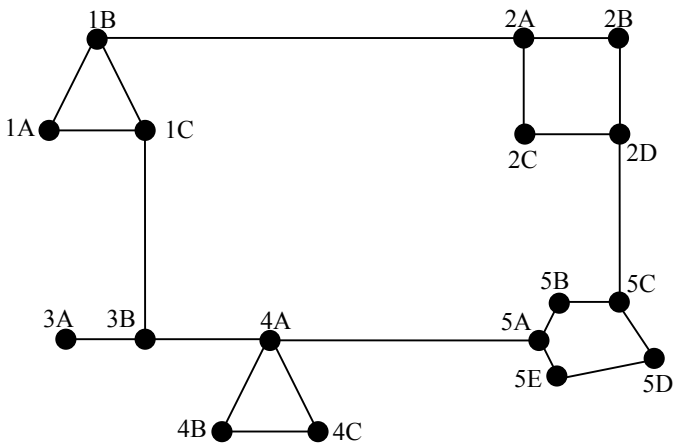
در بخشهای آتی باز هم به سراغ نسخه‌های کـاربردی و پیاده‌سـازی شـدهٔ ایـن الگوریتمـها خواهیم آمد.

۴) مسیریابی سلسله‌مراتبی[2]

در الگوریتمهای مسیریابی LS و DV هر مسیریاب باید جدولی را به‌عنوان جدول مسیریابی تشکیل بدهد. وقتی یک شبکه رشد می‌کند و شـبکه‌های محلـی و مسـیریابها اضافـه می‌شـوند حجم جداول مسیریابی و زمان لازم برای تعیین مسیر یک بسته افزایش پیدا می‌کند؛ تا جاییکـه ممکن است این زمان به تاخیرهای بحرانی بیانجامد و عملاً کـارآیی شـبکه کـاهش چشـمگیر داشته باشد. در شکل (۱۳-٤) به زیرساخت ارتباطی از یک شبکهٔ فرضی دقـت کنیـد. در ایـن شبکه ۱۷ مسیریاب وجود دارد و مسیریابی به روش DV انجام می‌شود. در این مثال معیار هزینه

[1] به این راه حل Split Horzon گفته می شود.

[2] Hierarchical Routing



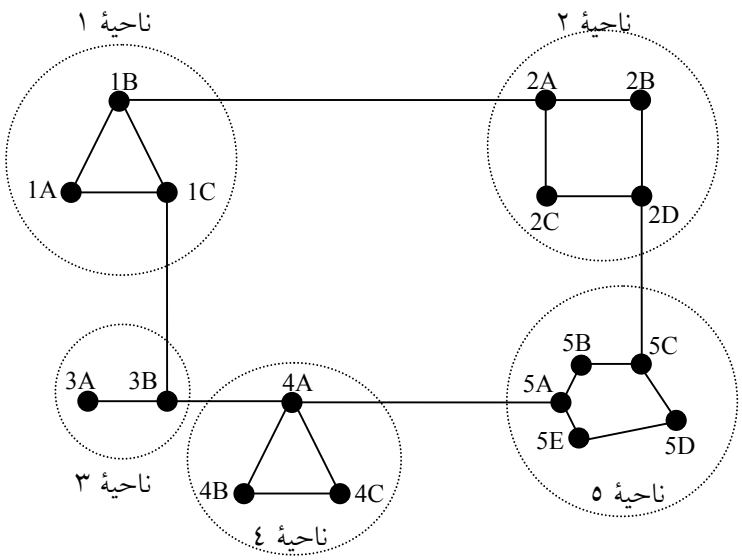
شکل (۱۳-٤) زیرساخت ارتباطی از یک شبکهٔ فرضی و جداول مسیریابی

مقصد	خطّ	هزینه
1A	–	-
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

شکل (۱٤-٤) جدول مسیریابی 1A

"تعداد گام" می‌باشد. (وقتی معیار هزینه تعداد گام است ، براحتی هزینهٔ هر خط را ثابت و مساوی ۱ فرض کنید) در شکل (۱۴-٤) جدول مسیریابی 1A نشان داده شده است و بگونه‌ای که ملاحظه می‌شود دارای ۱۷ رکورد است. تمام مسیریابها دارای چنین جدولی هستند و در فواصل منظم باید آن را برای مسیریابهای مجاور خود ارسال نمایند.

حال فرض کنید که تعداد صدهزار مسیریاب در شبکه موجود باشد. (صدهزارمین شبکهٔ دنیا که به اینترنت پیوست در سال ۱۹۹٦ ثبت شده است!) در چنین حالتی مسیریاب قادر به ذخیره ، پردازش و ارسال جداول مسیریابی برای دیگر مسیریابها نخواهد بود.



شکل (۱۵-٤) زیرساخت ارتباطی از یک شبکهٔ فرضی و جداول مسیریابی

مقصد	خطّ	هزینه
1A	–	–
1B	1B	1
1C	1C	1
Region	1B	2
Region	1C	2
Region	1C	3
Region	1C	4

شکل (۱٦-٤) جدول مسیریابی 1A

در مسیریابی سلسله‌مراتبی ، مسیریابها در گروههایی به نام **"ناحیه"**[1] دسته بندی می‌شوند. هر مسیریاب فقط **"نواحی"** و مسیریابهای درون ناحیهٔ خود را می‌شناسد و هیچ اطلاعی از مسیریابهای درون نواحی دیگر ندارد. در شکل (۱۵-٤) ، شبکهٔ مثال قبل در قالب پنج ناحیه تقسیم بندی شده و در شکل (۱٦-٤) جدول مسیریابِ 1A به تصویر کشیده شده است. در این جدول به ازای هر "ناحیه" و هر مسیریاب درون ناحیه ، یک رکورد در حافظه نگهداری می‌شود. به عنوان مثال مسیریاب 1A برای ارسال یک بسته به مسیریابی که در ناحیهٔ ۲ واقع است ، آنرا به سمت مسیریاب 1B هدایت می‌نماید؛ طبق این جدول برای ارسال بسته به نواحی ۳ و ٤ و ۵ ، باید از طریق 1C اقدام شود.

به گونه‌ای که از جدول (۱٦-٤) مشهود است با استفاده از مسیریابی سلسله‌مراتبی تعداد رکوردهای جدول مسیریابی از ۱۷ به ۷ رکورد کاهش یافته است. (۳ رکورد برای مسیریابهای درون ناحیه و ٤ رکورد برای بقیهٔ نواحی)

حال فرض کنید شبکه‌ای با ۷۲۰ مسیریاب ایجاد شده است؛ در روش معمولی (DV) جدول مسیریابی دارای ۷۲۰ رکورد خواهد بود. اگر این شبکه به ۲٤ ناحیه تقسیم و در هر ناحیه ۳۰ مسیریاب تعریف شود ، در این حالت مسیریاب نیاز به ۳۰ رکورد به ازای هر مسیریاب در درون ناحیه و ۲۳ رکورد به ازای ۲۳ ناحیهٔ دیگر دارد. (جمعاً ۵۳ رکورد)

برای ساده‌تر شدن جداول مسیریابی می‌توان از روش **"سلسله‌مراتبی سه‌سطحی"** استفاده کرد. در این روش کل شبکه به صورت زیر تقسیم‌بندی می‌شود:

- کل شبکه به چندین "دسته"[2] تقسیم می‌شود.
- هر "دسته" به چند "ناحیه" تقسیم می‌شود.
- هر ناحیه در برگیرندهٔ چند مسیریاب است.

به عنوان مثال شبکه‌ای با ۷۲۰ مسیریاب به ۸ "دسته" ، هر دسته دارای ۹ "ناحیه" و هر ناحیه دارای ۱۰ مسیریاب باشد. در این حالت هر مسیریاب به حداکثر ۲۵ رکورد احتیاج دارد:

۱. ۱۰ رکورد برای مسیریابهای هم‌ناحیه
۲. ۸ رکورد برای بقیهٔ نواحی (۸=۱-۹)
۳. ۷ رکورد برای بقیهٔ دسته‌ها (۷=۱-۸)

---

[1] Region
[2] Cluster

سلسله‌مراتب در مسیریابی می‌تواند بسته به بزرگی شبکه ، تا چندین سطح ادامه یابد. بعنوان مثال یک شبکه می‌تواند بصورت زیر تقسیم‌بندی شود:

- کل شبکه به تعدادی "ناحیه" تقسیم شود. Region
- هر ناحیه به تعدادی "دسته" تقسیم شود. Cluster
- هر دسته به تعدادی "حوزه" تقسیم شود. Zone
- هر حوزه به تعدادی "گروه" تقسیم شود. Group
- در هر گروه تعدادی مسیریاب تعریف شود.

برای آنکه بتوانید میزان صرفه‌جویی در اندازهٔ جداول مسیریابی سلسله‌مراتبی را ارزیابی کنید به جدول (۱۷-٤) دقت کنید. در این جدول تعداد رکوردی که هر مسیریاب ، باید در حافظه نگهداری کند ، ارائه شده است. (در یک شبکه با ۷۲۰ مسیریاب)

تنها اشکالی که می‌توان برای روش سلسله‌مراتبی برشمرد ، آنست که چون در این روش کل توپولوژی زیرشبکه برای هر مسیریاب مشخص نیست ، لذا ممکن است مسیر انتخابی برای ارسال بسته به یک مسیریابِ خاصّ درون یک ناحیه بهینه نباشد ولی در مجموع این روش به حالت بهینه نزدیک بوده و تغییرات توپولوژیکی و ترافیکی در جداول مسیریابی تاثیر داده خواهد شد.

در شبکهٔ اینترنت از روش سلسله‌مراتبی در مسیریابی استفاده می‌شود و پروتکلهای متفاوتی برای مسیریابی در درون یک "ناحیه" و مسیریابی بین نواحی تعریف و ارائه شده است.

	تعداد ناحیه Regions	تعداد دسته Clusters	تعداد حوزه Zones	تعداد مسیریاب	تعداد رکورد در جدول
مسیریابی DV بدون سلسله‌مراتب	۱	–	–	۷۲۰	۷۲۰
مسیریابی DV با سلسله‌مراتب دوسطحی	۲٤	–	–	۳۰	۵۳
مسیریابی DV با سلسله‌مراتب سه‌سطحی	۹	۸	–	۱۰	۲۵
مسیریابی DV با سلسله‌مراتب سه‌سطحی	۹	۵	٤	٤	۱۹

جدول (۱۷-٤) مقایسهٔ اندازهٔ جدول مسیریابی در روشهای سلسله‌مراتبی

۵) مسیریابی در اینترنت

اینترنت مجموعه‌ای از "شبکه‌های خودمختار[۱] و مستقل" است که به نحوی به هم متصل شده‌اند. "شبکهٔ خودمختار" که اختصاراً AS[۲] نامیده می‌شود ، شبکه‌ای است که تحت نظارت و سرپرستی یک مجموعه یا سازمان خاص پیاده و اداره می‌شود. مثلاً یک دانشگاه می‌تواند برای خود یک شبکهٔ AS مهیّا نماید بگونه‌ای که هر دانشکده دارای یک شبکهٔ محلی باشد و این شبکه‌های محلی طبق صلاحدید مسئول شبکه (از طریق نصب مسیریاب) به هم متصل شوند. اضافه شدن یک شبکهٔ محلی یا حتی یک ماشین میزبان به شبکه باید با مجوز و نظارت مسئول شبکه انجام شود. شکل (۱۸-٤) می‌تواند مثالی از یک شبکهٔ AS باشد. این مسئله واضح است که برای برقراری ارتباط بین زیرشبکه‌های درون یک AS ، باید مسیریابی انجام شود؛ مسئول شبکه موظف است "الگوهای زیرشبکه"[۳] را طراحی کرده و مسیریابها را بگونه‌ای تنظیم نماید که مسیریابی در درون شبکهٔ تحت نظارتِ او به درستی انجام شود.

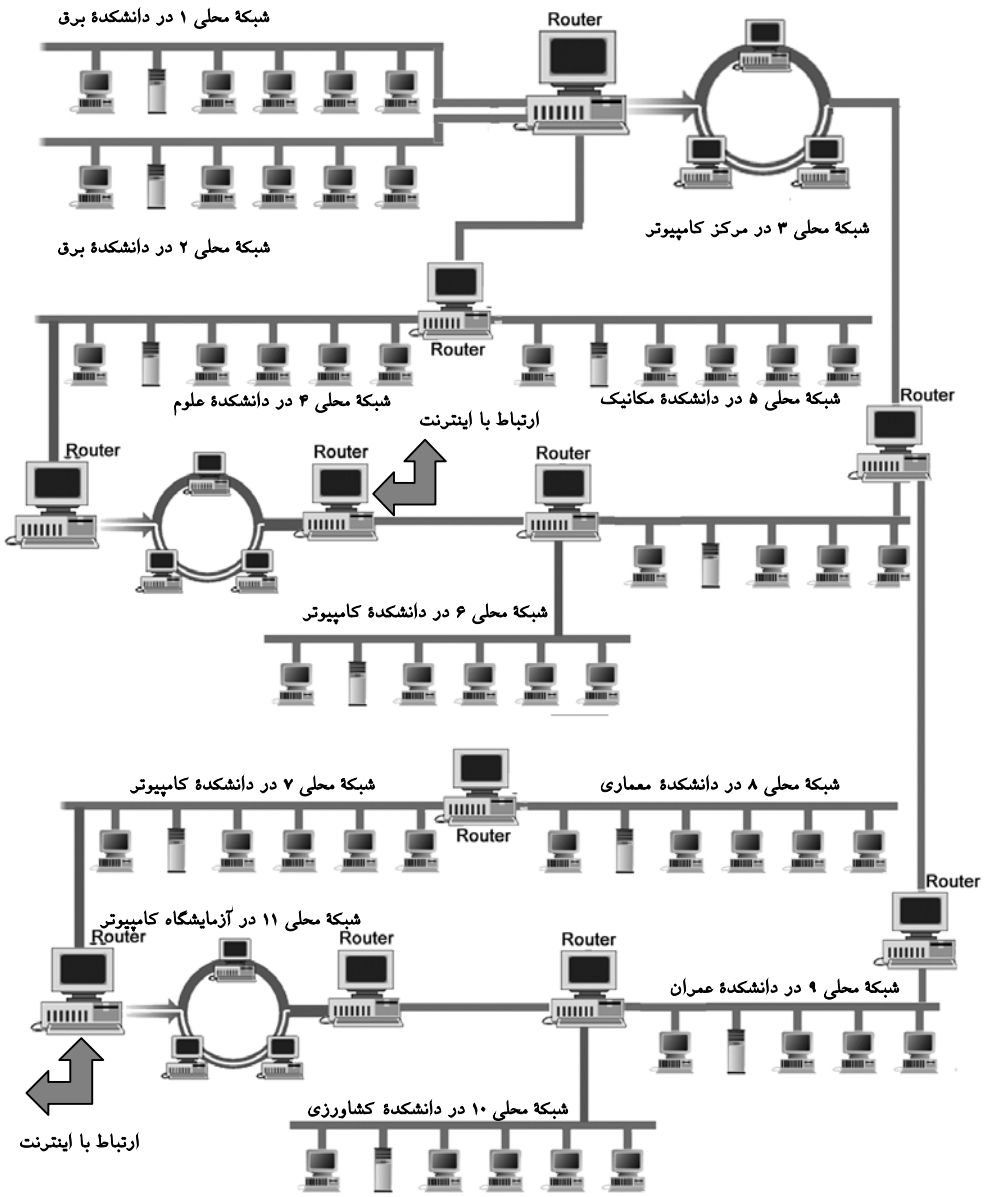
مسئول شبکهٔ خودمختار می‌تواند بر روی شبکهٔ تحت نظارت خود "حاکمیّت"[٤] داشته باشد یعنی می‌تواند بر روی تک‌تک اجزای شبکه (ماشینهای میزبان)، توپولوژی کل شبکه ، سیستم عامل ، طراحی زیرساخت ارتباطی و طریقهٔ اتصال شبکه‌های محلی و نوع پروتکل مسیریابی اعمال نفوذ کرده و نظرات خود را پیاده نماید.

مسیریابی بسته‌های IP در درون یک شبکهٔ خودمختار بیشتر تابع پارامترهایی نظیر سرعت و قابل اعتماد بودن الگوریتم مسیریابی است. حال فرض کنید شبکه‌های خودمختار از طریق یک زیرساخت ارتباطی بسیار سریع و جهانی به هم متصل شوند.[۵] مسیریابی بسته‌های اطلاعاتی بر روی شاهراه‌هایی که شبکه‌های AS را بهم متصل کرده ، مسائلی کاملاً متفاوت با مسیریابی در درون یک شبکهٔ خودمختار دارد. در مسیریابی بین شبکه‌های AS ، مسائلی نظیر امنیت ، پرداخت حقّ اشتراک و سیاست نیز می‌تواند در انتخاب بهترین مسیر دخیل باشد.

به گونه‌ای که در مبحث آدرس‌دهی اشاره شد یک شبکهٔ عظیم می‌تواند خودش از چندین زیرشبکه تشکیل شده باشد که مسیریابی بسته‌ها در درون این شبکه بر اساس الگوهای زیرشبکه انجام می‌شود ولی از دیدگاه شبکهٔ اینترنت ، کل این شبکه (با تمام ماشینهای میزبان و مسیریابهای داخلی) فقط یک "گره" تلقی شده و مسیریابهای بیرونی آنرا یک سیستم واحد می‌بینند.[٦] در حقیقت کلاسهای آدرس IP به شبکه‌های خودمختار اختصاص داده می‌شود.

---

[۱] Autonomous
[۲] Autonomous System
[۳] **به فصل قبلی مراجعه کنید.**
[٤] Authority
[۵] **در فصل قبل اشاره شد که به زیرساخت ارتباطی شبکه‌های جهانی ستون فقرات یا** Backbone **می‌گویند.**
[٦] **این شبکه با قسمت** NetID **در آدرس** IP **مشخص می‌شود.**



شکل (۱۸-٤) مثالی از یک سیستم خودمختار (AS) در یک دانشگاه

قبل از معرفی روشهای مسیریابی در اینترنت ، ارائهٔ یک مثال ساده می‌تواند به فهم کلیّت موضوع کمک کند:

کشورمان ایران را در نظر بگیرید؛ در این کشور مسیرهای هوایی زیادی بین مراکز استان و برخی از شهرستانها وجود دارد. بنابراین یک فرد ایرانی وقتی می‌خواهد از نقطه‌ای به نقطهٔ دیگر سفر کند با یک ارزیابی ساده (از میزان هزینه و زمان) مسیر خود را انتخاب و بر اساس آن سفرش را آغاز می‌نماید. یک ایرانی مشکل خاصی برای سفر آزادانه در درون کشور خود ندارد و براحتی می‌تواند بین هر دو نقطه سفر کند؛ ضوابط حاکم بر راههای هوایی و زمینی کشور نیز توسط دولت تبیین می‌شود. حال فرض کنید یک ایرانی در شیراز بخواهد به شهر کلن در آلمان سفر کند؛ برای انتخاب مسیر خود : اولاً نیاز به اخذ مجوزهای لازم از کشور مبدأ و مقصد دارد. ثانیاً چون پرواز مستقیم بین این دو شهر وجود ندارد ، باید با یک پرواز داخلی خود را به یک فرودگاه بین‌المللی (مثلاً مهرآباد تهران) رسانده و از طریق یک پرواز خارجی به یکی از شهرهای آلمان (برلین ، فرانکفورت) سفر کند و پس از ورود به آلمان طبق ضوابط دولت آلمان و از یک مسیر مناسب به کلن آلمان سفر کند. تفاوتهای ویژه در پروازهای داخلی و پروازهای خارجی از لحاظ سطح کنترل مدارک ، اخذ هزینه و مسائل امنیتی وجود دارد. یک پرواز خارجی ممکن است در طول مسیر از کشورهای ثالثی عبور کند و این کشورها بر اساس تعرفه‌هایی برای عبور یک پرواز خارجی اخذ هزینه کنند یا آنکه بدلایل امنیتی اصلاً اجازهٔ عبور صادر نکنند.

این مثال تا حدود زیادی به انواع مسیریابی در شبکهٔ اینترنت شباهت دارد. بگونه‌ای که در شکل (۱۸-٤) مشهود است مسیرهای داخلی زیادی در درون شبکهٔ دانشگاه وجود دارد که بسته‌های IP می‌توانند از طریق آنها بین هر دو ماشین درون این شبکه مبادله شوند ولی برای ارسال بسته‌ها به خارج از شبکه ، ابتدا بسته‌ها باید به سمت یکی از دو مسیریاب که با دنیای خارج در ارتباطند هدایت شوند (مسیریابی درونی)؛ سپس این دو مسیریاب بسته‌ها را روی زیرساخت خارجی شبکه به جریان می‌اندازند. مسیریابهایی که در درون شبکهٔ AS نصب شده‌اند و نقش برقراری ارتباطات داخلی را برعهده دارند به نام "دروازه‌های درونی"[۱] مشهورند.[۲] مسیریابهایی که ارتباط دو شبکهٔ خودمختار متفاوت را برقرار می‌کنند و تمامی ارتباطات بین‌شبکه‌ای از طریق آنها انجام می‌شود به نام "دروازه‌های مرزی"[۳] شناخته می‌شوند. به شکل (۱۹-٤) دقت کنید. در این شکل زیرساخت ارتباطی چهار شبکهٔ AS که هر کدام با اهداف ویژه ، تحت نظارت و سرپرستی یک سازمان خاص پیاده‌سازی شده‌اند ، به تصویر

[۱] Interior Gateway

[۲] **"دروازه یا Gateway" نام سنتی و اولیّه مسیریاب بوده است.**

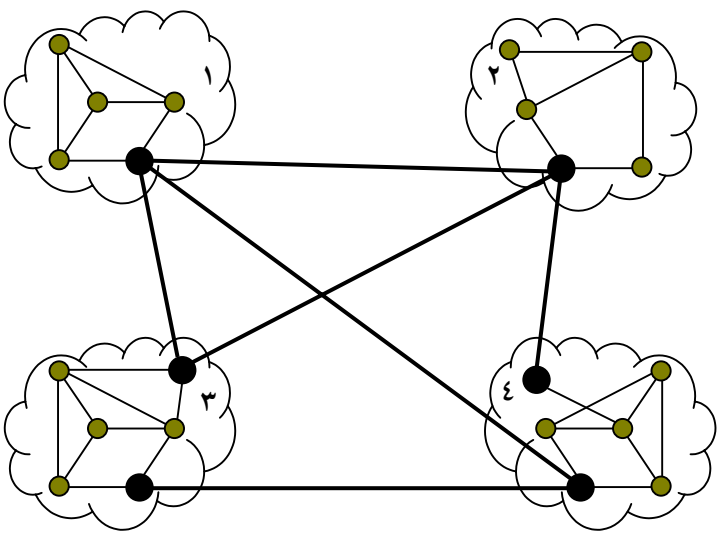
[۳] Border Gateway

کشیده شده است. از بین مسیریابهای این چهار شبکه فقط تعداد محدودی از مسیریابها ، ارتباطات بین شبکه‌ای را برعهده دارند. این "مسیریابهای مرزی" با رنگ تیره‌تری نسبت به مسیریابهای درونی نشان داده شده‌اند. مسیریابهای مرزی و ساختار ارتباطی بین آنها تابع قواعد "مسیریابی برونی" و مسیریابهای داخلی تابع الگوریتمهای "مسیریابی درونی" است که می‌تواند کاملاً با هم متفاوت باشد. به مسیریابهای مرزی ، "مسیریابهای BGP" نیز گفته می‌شود.

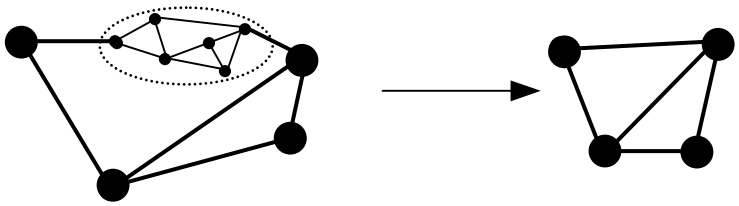
در ساختار مثال فوق اگر یک ماشین میزبان در شبکهٔ ۱ بخواهد بسته‌ای برای ماشین دیگر در شبکهٔ ٤ بفرستد سه مرحله مسیریابی لازم است:

- مسیریابی در درون شبکهٔ ۱ تا رسیدن بسته به مسیریاب مرزی
- مسیریابی روی خطوط ارتباطی بین‌شبکه‌ای تا رسیدن به شبکهٔ ٤
- مسیریابی درون شبکهٔ ٤ تا رسیدن به ماشین مقصد

ممکن است بین دو مسیریاب مرزی یک شبکه قرار گرفته باشد که کلّ آن را می‌توان یک خطّ واحد در نظر گرفت. در شکل (۲۰-٤) این موضوع به تصویر کشیده شده است.



شکل (۱۹-٤) مثالی از چهار شبکهٔ AS متصل به هم

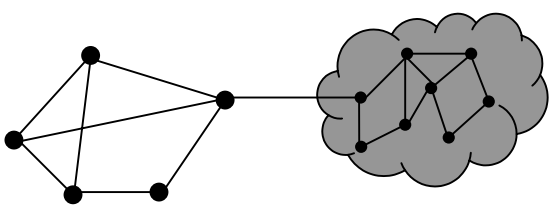


شکل (۲۰-٤) ساختار ارتباطی یک شبکهٔ فرضی

مجموعهٔ اینترنت از دهها هزار شبکهٔ خودمختار تشکیل شده که از طریق یک ساختار ارتباطی به نام "ستون فقرات" به هم متصل شده‌اند. در ادامه طریقهٔ اتصال این شبکه‌ها به اینترنت تشریح شده است.

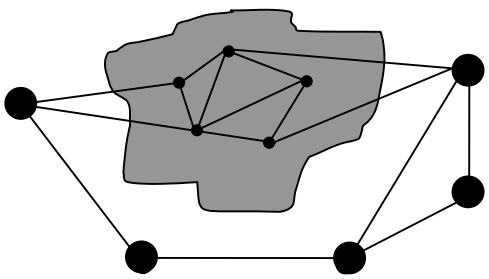
سه دسته شبکه می‌توانند با مسیریابهای BGP در ارتباط باشند:

- **"شبکه‌های پایانی"**- Stub - : این نوع از شبکه ها فقط با یک مسیریاب نوع BGP در ارتباطند و بنابراین نمی‌توانند در ستون فقرات اینترنت نقش ایفا کنند و کمکی به توزیع ترافیک بر روی شبکهٔ اینترنت نمی‌کنند. معمولاً برای وصل شبکه‌های پایانی به یکی از مسیریابهای BGP باید هزینهٔ قابل توجهی در هر ماه پرداخت شود. اکثر شبکه‌های متصل به اینترنت در ایران بخاطر عدم وجود ستون‌فقرات ارتباطی سریع بین شهرها و استانهای مختلف کشور ، از نوع شبکه‌های پایانی -Stub- بشمار می‌روند.



شکل (۲۱-٤) ساختار ارتباطی یک شبکهٔ پایانی

➢ **"شبکه‌های چندارتباطی"**[۱]: این گونه از شبکه‌ها بین مسیریابهای نوع BGP واقعند و می‌توانند برای توزیع و حمل ترافیک در شبکهٔ اینترنت مورد استفاده قرار بگیرند مگر آنکه بدلایل امنیتی ، تمایل به چنین کاری نداشته باشند.



شکل (۲۲-٤) ساختار ارتباطی یک شبکهٔ چندارتباطی

➢ **"شبکه‌های ترانزیت"**[۲] : این گونه شبکه ها که به نحوی به روی ستون فقرات شبکه اینترنت واقعند وظیفه عمده‌ای در حمل و توزیع بسته های IP بعهده دارند. (همانند شبکهٔ NSFNet در آمریکا)

با مقدمهٔ فوق باید برخی از الگوریتمهای کاربردی مسیریابی درونی و برونی در شبکهٔ اینترنت را معرفی کنیم.

۶) پروتکل RIP در مسیریابی درونی

پروتکل RIP[۳] یکی از پروتکلهای پرسابقه در مسیریابی درونی در شبکهٔ اینترنت بشمار می‌رود و ابداع کنندهٔ اصلی آن شرکت "زیراکس"[٤] است. شاید نقطهٔ عطف گسترش این پروتکل سال ۱۹۸۲ بود که بهمراه نسخهٔ "یونیکس برکلی" (BSD)[٥] ، نرم‌افزار پیاده شدهٔ

[۱] Mulitconnected Networks
[۲] Transit Networks
[۳] Routing Information Protocol
[٤] Xerox Network System
[٥] Berkeley Software Distribution

پروتکل RIP به بازار ارائه شد. بعد از آن ، در نسخه‌های بعدی یونیکس ، پروتکل RIP نیز بهینه‌سازی شد.[۱]

پروتکل RIP ذاتاً مبتنی بر الگوریتم بردار فاصله (DV) است؛ با این مشخصه که ، معیار هزینه در این پروتکل "تعداد گام" است یعنی هزینهٔ هر خط بین دو مسیریاب ثابت و مساوی ۱ است. با این معیار وقتی هزینهٔ مسیری ٤ اعلام می‌شود بدین معناست که ٤ مسیریاب در سر راه این مسیر وجود دارد. (برای درک دقیق این پروتکل ، باید الگوریتم DV را به دقّت بررسی کنید.)

در پروتکل RIP جداول مسیریابی هر ۳۰ ثانیه یکبار بین مسیریابهای مجاور مبادله می‌شوند و اگر مسیریابی این جداول را از یک همسایه به مدت ۱۸۰ ثانیه (۳ دقیقه) دریافت نکند ، متوجه یک خرابی در آن همسایه شده و ضمن درج ∞ در جدول خود ، آنرا به تمام مسیریابهای مجاور خود اعلان می‌کند. ارسال جداول مسیریابی به همسایه‌ها در فواصل ۳۰ ثانیه‌ای ، **"اعلان"**[۲] گفته می‌شود.

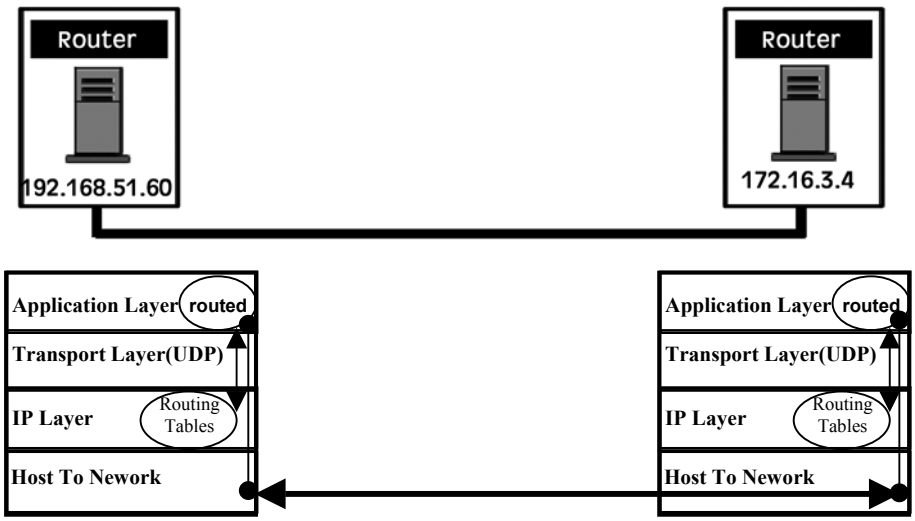
در پروتکل اولیهٔ RIP ، حداکثر تعداد طول مسیر به ۱۵ محدود شده بود یعنی در یک زیرساخت ارتباطی با هر تعداد مسیریاب ، طول بزرگترین مسیر نباید از ۱۵ زیادتر شود. در این پروتکل هر مسیریاب می‌تواند با ارسال یک "پیام تقاضا"[۳] از همسایهٔ خود در مورد هزینهٔ رسیدن به یک مسیریاب خاص در شبکه سوال نماید.

RIP از پورت شمارهٔ ۲۵۰ و پروتکل UDP برای مبادلهٔ جداول مسیریابی استفاده می‌نماید. (موضوع شمارهٔ پورت و طریقهٔ برنامه‌نویسی تحت شبکهٔ اینترنت در فصول بعدی بررسی می‌شوند.) در شکل (۲۳-٤) ساختار پروتکل RIP که در لایهٔ کاربرد از مدل TCP/IP تعریف می‌شود ، نشان داده شده است. بگونه‌ای که مشاهده می‌شود جداول مسیریابی در لایهٔ دوم و برای مسیریابی بسته‌های IP ، مورد استفاده هستند ولی مبادلهٔ جداول و عملیاتِ به‌هنگام‌سازی توسط یک برنامهٔ کاربردی در لایهٔ چهارم انجام می‌شود. نام این برنامه در سیستم عامل یونیکس routed[٤] می‌باشد و با استفاده از برنامه‌نویسی سوکت که در فصلی مجزا به آن خواهیم پرداخت نوشته شده است.

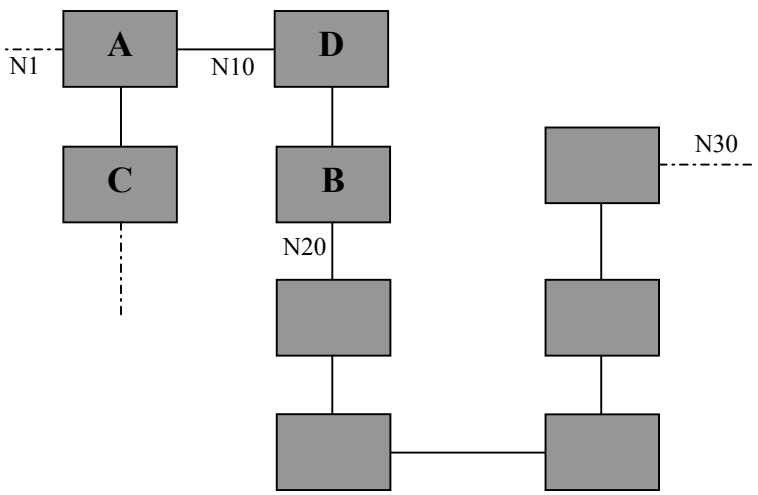
هرچند پروتکل RIP دقیقاً مشابه الگوریتمی است که در روش DV به آن پرداختیم ولی در اینجا به مثالی دیگر خواهیم پرداخت. در شکل (۲٤-٤) بخشی از زیرساخت ارتباطی یک شبکهٔ AS نشان داده شده و طبق معمول ماشینهای میزبان حذف شده‌اند. خطوط نقطه‌چین به

---

[۱] RIP Version 2
[۲] Advertisement
[۳] Request Message
[٤] **کلمهٔ routed را route dee تلفظ کنید چراکه مخفف** Routing Daemon است.



(۲۳-٤) پروتکل RIP در لایهٔ کاربرد



شکل (۲٤-٤) بخشی از زیرساخت ارتباطی یک شبکهٔ AS

معنای آنست که گراف شبکه ادامه دارد ولی در تصویر نشان داده نشده است.
در جدول (۲۵-٤) بخشی از جدول مسیریابی D نشان داده شده است.

مسیریاب مقصد	مسیریاب بعدی	تعداد گام تا مقصد
N1	A	2
N20	B	2
N30	B	7
N10	-	1
....	....	....

جدول (۲۵-٤) بخشی از جدول مسیریاب D

فرض کنید در لحظهٔ *t=t0* جدول مسیریابی (۲٦-٤) ، توسط مسیریاب A به D اعلان شده باشد.

مسیریاب مقصد	مسیریاب بعدی	تعداد گام تا مقصد
N1	-	1
N10	-	1
N30	C	4
....	....	....

جدول (۲٦-٤) بخشی از جدول اعلان شده توسط مسیریاب A به D

با استفاده از این جدول و طبق الگوریتمی که در روش DV اشاره شد ، D نتیجه می‌گیرد که برای رسیدن به شبکهٔ N30 مسیر کوتاهتری از طریق A وجود دارد و بنابراین جدول خود را به صورت جدول (۲۷-٤) اصلاح می‌کند.

مسیریاب مقصد	مسیریاب بعدی	تعداد گام تا مقصد
N1	A	2
N20	B	2
N30	A	5
N10	-	1
....	....	....

جدول (۲۷-٤) بخشی از جدول مسیریاب D پس از اصلاح

قالب پیامها در پروتکل RIP در شکل (۲۸-٤) نشان داده شده است. در "سرآیند پیام"[۱] سه فیلد زیر تعریف شده است:

- **Command** : این فیلد نوع پیام را مشخص می‌کند :

۱ : پیام از نوع تقاضا است.[۲] اگر در بدنهٔ پیام آدرس IP درج شده باشد بدین معناست که هزینهٔ یک مسیریاب خاص تقاضا شده است ولی اگر هیچ آدرسی درج نشده باشد یعنی کل جدول مسیریابی باید برای تقاضا دهنده ارسال شود.

۲ : پیام از نوع پاسخ است.[۳] یعنی درون بدنهٔ پیام کلّ یا قسمتی از جدول مسیریابی قرار گرفته است.

- **Version** : این فیلد نسخهٔ پروتکل RIP که بسته بر طبق آن تولید و ارسال شده است را معین می‌کند. (۱ ، ۲ و ...)
- **Reserved** : این فیلد بدون استفاده است و با صفر پر می‌شود.

بدنهٔ پیام در این پروتکل شامل ۳ فیلد مهم است :

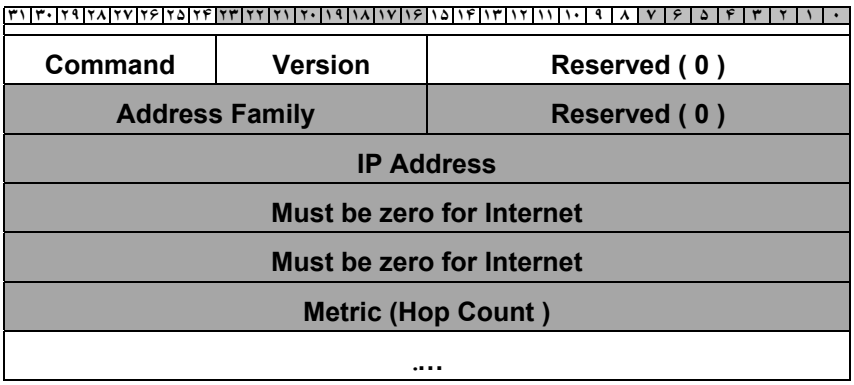
- **Address Family** : در این فیلد برای شبکهٔ اینترنت مقدار ثابت **۲** قرار می‌گیرد.
- **IP Address** : آدرس IP مسیریاب را تعیین می‌کند.
- **Metric** : معیار هزینهٔ رسیدن به مسیریاب مشخص شده در فیلد قبلی را بر حسب گام تعیین می‌کند. در پروتکل RIP حداکثر مقدار این فیلد ۱۵ است. " اگر یک مسیریاب در دسترس نباشد ، به جای مقدار ∞ در این فیلد مقدار **۱٦** قرار می‌گیرد.

بدنهٔ پیام (قسمت تیره‌رنگ در قالب پیام)می‌تواند به ازای هر رکورد در جدول مسیریابی تکرار شود.

در مثالهای این فصل ، تمام شبکه‌ها و مسیریابها با اسامی نمادین (همانند A ، B ،N1 و N30) نشان داده شده‌اند در حالی واضح است که شبکه‌ها و مسیریابها با آدرس IP تعیین می‌شوند. در جدول (۲۹-٤) حالت واقعی از یک جدول مسیریابی که در حافظهٔ مسیریاب نگهداری می‌شود ، آورده شده است. این جدول در سیستم عامل یونیکس با استفاده از دستور netstat –rn بدست آمده است.

---

[۱] Message Header
[۲] Request Message
[۳] Response Message



شکل (۲۸-٤) قالب پیامها در پروتکل RIP

Destination	Gateway	Flags	Ref	Use	Interface
127.0.0.1	127.0.0.1	UH	0	26492	lo0
192.168.2.	192.168.2.5	U	2	13	fa0
193.55.114.	193.55.114.6	U	3	58503	le0
192.168.3.	192.168.3.5	U	2	25	qaa0
224.0.0.0	193.55.114.6	U	3	0	le0
default	193.55.114.129	UG	0	143454	

جدول (۲۹-٤) حالت واقعی از یک جدول مسیریابی

آدرس شبکه‌های مقصد که در این جدول درج شده‌اند به شرح زیرند :

♦ **192.168.2. و 193.55.114. و 192.168.3. :** این سه آدرس ، سه شبکه با آدرس کلاس C را تعیین می‌کند. در هنگام ارسال یک بسته برای یک ماشین ، آدرس IP ماشین مقصد ، بررسی شده و اگر قسمت "مشخصهٔ شبکهٔ"[1] آن یکی از سه مقدار بالا باشد ، بسته به سمت مسیریابی هدایت می‌شود که آدرس آن در قسمت Gateway درج شده است. آدرس نمادین کانال خروجی که بسته باید روی آن ارسال شود (یعنی آدرس نمادین کارت شبکه) در قسمت Interface مشخص شده است.

[1] NetID

♦ **127.0.0.1 :** این آدرس همان آدرس "بازگشت" است که در فصل قبل توضیح داده شد. ارسال بسته‌ای به این مقصد باعث بازگشت آن به تولیدکنندهٔ آن خواهد شد.

♦ **224.0.0.0 :** این آدرس یک آدرس کلاس D است و برای پخش همگانی کاربرد دارد.

♦ **default :** هرگاه بسته‌ای برای یک ماشین میزبان ارسال شود ولی آدرس شبکهٔ مقصد آن جزو شبکه‌های مشخص شده در جدول نباشد آن بسته به سمت مسیریابی هدایت می‌شود که آدرس آن در قسمت Gateway درج شده است.

به گونه‌ای که اشاره شد با وجود تمام مشکلاتی که در این پروتکل وجود دارد ، به دلیل سادگی و سرعت ، هنوز هم در شبکه‌های کوچک مورد استفاده قرار می‌گیرد و در نسخه های جدید ویندوز NT پشتیبانی می‌شود.

۷) پروتکل OSPF در مسیریابی درونی

بکارگیری پروتکل RIP در شبکه‌های کامپیوتری بیشتر به دلیل شرایط زمان بوده است. در دههٔ هفتاد و هشتاد حافظه و پردازنده‌های سریع ، گران قیمت بودند و پیاده‌سازی الگوریتمهای مسیریابی مبتنی بر روشهایی نظیر LS که هم به حافظه و هم به پردازندهٔ سریع نیاز دارند ، مقرون به صرفه نبود. از طرفی شبکه‌ها نیز آنقدر توسعه نیافته بودند که نیاز به الگوریتمهای بهینه‌تر احساس شود. با گسترش اینترنت و توسعهٔ شبکه‌های خودمختار در اواخر دههٔ هشتاد ، کاستیهای پروتکل RIP نمود بیشتری پیدا کرد و با سریع شدن پردازنده‌ها و ارزان شدن سخت‌افزار ، نیاز به طراحی یک پروتکل بهینه ، IETF[۱] را واداشت تا در سال ۱۹۹۰ ، OSPF[۲] را به عنوان یک پروتکل استاندارد ارائه نماید. مسیریابهای زیادی مبتنی بر این پروتکل به بازار عرضه شده‌اند و احتمال می‌رود که در آینده تبدیل به مهمترین پروتکل مسیریابی درونی در شبکه‌های AS شود.

ابتدا مشخصه‌های پروتکل OSPF را در مقایسه با پروتکل RIP ارائه می‌کنیم:

---

[۱] Internet Engineering Task Force

استانداردسازی تمام پروتکلها و مستندات شبکهٔ اینترنت توسط IETF که کمیته‌ای بین‌المللی است انجام می‌شود.

[۲] Open Shortest Path First

کلمهٔ Open در نام این پروتکل به معنای آنست که این پروتکل بطور عام در اختیار تمام دنیا قرار دارد و طراحان مسیریاب می‌توانند بدون هیچ مشکلی از آن استفاده کنند؛ یعنی امتیاز این پروتکل در اختیار گروه یا شرکت خاصی نیست. (بر خلاف پروتکلی مثل IGRP که امتیاز آن متعلق به شرکت CISCO در آمریکاست و کسی حق استفادهٔ بدون مجوز از آنرا ندارد. (RFC1247)

- بر خلاف پروتکل RIP ، در این پروتکل از الگوریتم LS برای محاسبهٔ بهترین مسیر استفاده می‌شود و بنابراین مشکل "شمارش تا بینهایت" وجود ندارد.
- برخلاف پروتکل RIP ، در این پروتکل معیار هزینه فقط "تعداد گام" نیست بلکه می‌تواند چندین معیار هزینه را در انتخاب بهترین مسیر در نظر بگیرد.
- بر خلاف پروتکل RIP ، در این پروتکل حجم بار و ترافیک یک مسیریاب در محاسبهٔ بهترین مسیر دخالت داده می‌شود و در ضمن در هنگام خرابی یک مسیریاب ، جداول مسیریابی سریعاً همگرا می‌شود.
- بر خلاف پروتکل RIP ، در این پروتکل ، فیلد Type of Service در بستهٔ IP می‌تواند در نظر گرفته شود و بر اساس نوع سرویس درخواستی ، برای یک بسته مسیر مناسب انتخاب گردد.
- بر خلاف پروتکل RIP ، در پروتکل OSPF تمام بسته‌های ارسالی برای یک مقصد خاص ، روی بهترین مسیر هدایت نمی‌شود بلکه درصدی از بسته‌ها روی مسیرهایی که از لحاظ حداقل هزینه در رتبهٔ ۲ ، ۳ و ... قرار دارند ارسال می‌شود تا پدیدهٔ "نوسان" که قبلاً به آن اشاره شد رخ ندهد. به این کار **"موازنهٔ بار"**[1] گفته می‌شود.
- بر خلاف پروتکل RIP ، در این پروتکل از مسیریابی سلسله‌مراتبی پشتیبانی می‌شود.
- بر خلاف پروتکل RIP ، در این پروتکل مسیریابها جداول مسیریابی را از دیگر مسیریابها قبول نمی‌کنند مگر آنکه هویت ارسال کنندهٔ آن احراز شود. به همین دلیل مسئول شبکه برای هر مسیریاب یک "کلمهٔ عبور" تعیین می‌کند تا کاربران اخلالگر نتوانند با برنامه‌نویسی ، جداول مسیریابی مصنوعی تولید کرده و با ارسال آنها ، مسیریابی در شبکه را با مشکل مواجه کنند.

قبل از مطالعهٔ این بخش روش سلسله‌مراتبی در مسیریابی و الگوریتم LS را به دقّت بررسی کنید ، زیرا OSPF مبتنی بر این دو مفهوم است.

در ادبیات پروتکل OSPF ، سلسله‌مراتب تعیین شده برای نواحی بصورت زیر می‌باشد:

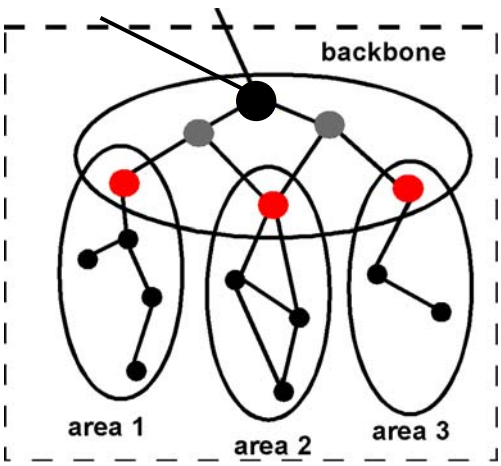
- یک شبکهٔ خودمختار (AS) به تعدادی "ناحیه"[2] تقسیم می‌شود. تمام مسیریابهای درون یک ناحیه باید مسیریابهای هم‌ناحیهٔ خود و هزینهٔ ارتباط بین آنها را بدانند و در جدولی ذخیره کنند. در لحظات به‌هنگام‌سازی ، این جداول برای تمام مسیریابهای هم‌ناحیه ارسال خواهد

[1] Load Balancing
[2] Area

شد. مسیریاب هیچ اطلاعی از وضعیت مسیریابهای درون نواحی دیگر ندارد. در شکل (۳۰-٤) این مسیریابها با علامت ● نشان داده شده است.

➢ درون هر ناحیه یک یا چند مسیریاب وجود دارند که ارتباط بین نواحی را برقرار می‌کنند؛ به آنها ، "مسیریابهای مرزی"[۱] گفته می‌شود. مجموعهٔ مسیریابهای مرزی و مسیریابهایی که در خارج از هر ناحیه نقش توزیع ترافیک بین نواحی را بر عهده دارند (بهمراه ساختار ارتباطی بین این مسیریابها) **"ستون‌فقرات"** شبکهٔ AS را تشکیل می‌دهد. در شکل (۳۰-٤) ، مسیریابهای مرزی با علامت ● نشان داده شده است.

➢ درون ستون‌فقراتِ شبکهٔ AS ممکن است مسیریابهایی وجود داشته باشند که با دیگر شبکه‌های AS در ارتباط باشد. به این مسیریابها "دروازه‌های مرزی"[۲] یا BGP گفته می‌شود. در شکل (۳۰-٤) این مسیریاب با دایرهٔ بزرگ و تیره‌رنگ نشان داده شده است.



شکل (۳۰-٤) سلسله‌مراتب مسیریابی در پروتکل OSPF

در پروتکل OSPF جداول زیر توسط مسیریابها "اعلان" می‌شود:

➢ **جدول مسیریابی محلی درون یک ناحیه**[۳] : این جداول ، محتوی اطلاعاتی در مورد گراف هزینهٔ ناحیه‌ای است که یک مسیریاب به آن متعلق است و توسط هر مسیریاب درون آن ناحیه، به تمام مسیریابها اعلان می‌شود.

---

۱ Area Border Routers
۲ Boundary Gateway
۳ Router Links Advertisement

- **جدول مسیریابی شبکه درون یک ناحیه**[۱] : این جداول که محتوی اطلاعاتی در مورد مسیریابها و کانالهای بین آنها در یک شبکه است ، توسط مسیریابهای درون یک ناحیه به تمامی مسیریابها اعلان می‌شود.
- **جدول خلاصهٔ مسیریابیِ مسیریابهای مرزی**[۲] : این جداول محتوی اطلاعاتی خلاصه ، در مورد مسیرهای موجود در خارج از نواحی است و توسط مسیریابهای مرزی به تمامی مسیریابهای نواحی مختلف اعلان می‌شود.
- **جدول مسیریابی شبکه**[۳] : این جداول محتوی اطلاعاتی در مورد مسیریابها و کانالهای بین آنها در خارج از شبکهٔ AS است و توسط مسیریابهای واقع بر ستون فقرات شبکهٔ AS به تمامی مسیریابهای نواحی مختلف اعلان می‌شود ولی فقط در مسیریابهای مرزی مورد استفاده قرار می‌گیرد.

بر خلاف RIP که از پروتکل UDP در لایهٔ انتقال استفاده می‌کند و به عنوان یک برنامهٔ کاربردی مطرح است ، پروتکل OSPF مستقیماً از پروتکل IP استفاده می‌کند و به عنوان یک برنامه در لایهٔ انتقال انجام وظیفه می‌نماید. اگر در فیلد "پروتکل" از بستهٔ IP عدد ۸۹ قرار گرفته باشد ، آن بسته تحویل پروسهٔ OSPF در لایهٔ سوم می‌شود. تعریف پروتکل OSPF در لایهٔ سوم (بجای لایهٔ چهارم) ، سرعت عمل آنرا افزایش داده است.

پیامهای تعریف شده در پروتکل OSPF متنوع و زیاد هستند ولی برای ارائهٔ یک دید کلی از امکاناتی که این پروتکل ارائه می‌دهد ، انواع و قالب این پیامها و فیلدهای تعریف شده در آن را بطور اجمالی معرفی می‌نماییم. بررسی کلّی برخی از فیلدهای این پیامها می‌تواند مشخصات کلی این پروتکل را روشنتر کند.

پنج نوع پیام در پروتکل OSPF تعریف شده است:

- پیام **"سلام"** : وقتی یک مسیریاب روشن و بوت می‌شود موظف است به تمام مسیریابهایی که مستقیماً به آنها متصل است[۴] یک پیام "سلام" بفرستد تا آنها از حضور این مسیریاب در شبکه مطلع شوند. قالب این پیام در شکل (۳۱-٤) نشان داده شده است و به گونه ای که دیده می‌شود سرآیند این پیام دو قسمت است:

---

۱ Router Links Advertisement
۲ Summary Links Advertisement
۳ Autonomous System Extended Links Advertisement
٤ Point To Point Adjacent Router

قسمت ٢٤ بایتی اول که به رنگ تیره نشان داده شده است. این قسمت از سرآیند ، در تمام پیامهای دیگر نیز وجود دارد و شامل اطلاعات مهمی است که به آن اشاره خواهیم کرد.
قسمت دوم سرآیند ، فقط مختص به "پیام سلام" می‌باشد.

- پیام Link State Update : هر مسیریاب موظف است که در بازه‌های زمانی مشخص ، جدول مسیریابی خودش را به روش سیل‌آسا به اطلاع دیگر مسیریابهای هم‌ناحیه برساند. این کار را با ارسال این پیام انجام می‌دهد؛ در ضمن هر مسیریاب وقتی هزینهٔ یکی از خطوط مستقیم او تغییر کرد یا مسیریاب مجاورش از شبکه بیرون رفت (یا به شبکه برگشت) سریعاً با این پیام آنرا به اطلاع دیگران می‌رساند. قالب این پیام در شکل (٣٢-٤) نشان داده شده است.

- پیام Database Description : هر مسیریاب به ازای تک‌تک رکوردهای هزینه که در یک بانک اطلاعاتی درون حافظهٔ اصلی ذخیره کرده ، یک فیلد شمارهٔ ترتیب در نظر می‌گیرد. مسیریاب ارسال کننده این پیام ، شمارهٔ ترتیب و تمام رکوردهایی را که در بانک اطلاعاتی خود ذخیره کرده است ، ارسال می‌نماید. گیرندهٔ این پیام با مقایسهٔ شماره‌های ترتیب رکوردها با رکوردهایی که در بانک اطلاعاتی خود دارد می‌تواند رکوردهای قدیمیتر را با رکوردهای جدید جایگزین نماید. قالب این پیام در شکل (٣٣-٤) نشان داده شده است .

- پیام Link State Request : با این پیام هر مسیریاب می‌تواند اطلاعات جدول مسیریابی را از از یک مسیریاب خاص تقاضا نماید. با این کار مسیریاب می‌تواند ضمن درخواست جدول مسیریابی همسایه‌های خود و مقایسهٔ شمارهٔ ترتیب رکوردهای آن اقدام به تازه‌سازی جدول خود نماید. قالب این پیام در شکل (٣٤-٤) نشان داده شده است .

- پیام Link State Ack : این پیام توسط گیرندهٔ پیام Link State Update برای ارسال کنندهٔ آن ارسال می‌شود و به منظور تصدیق دریافت جدول مسیریابی می‌باشد.

به گونه‌ای که اشاره شد تمام پیامهای تعریف شده در پروتکل OSPF دارای سرآیند ٢٤ بایتی هستند و در اشکال (٣١-٤) تا (٣٤-٤) به رنگ تیره نشان داده شده است. این فیلدها به شرح زیرند:

- **فیلد Version** : نسخهٔ پروتکل OSPF را تعیین می‌کند. (فعلاً این شماره ۱ است)
- **فیلد Type** : این فیلد یکی از انواع پنج‌گانهٔ پیام را که در بالا معرفی شد ، مشخص می‌کند. معنای مقادیر مختلف این فیلد عبارتند از :

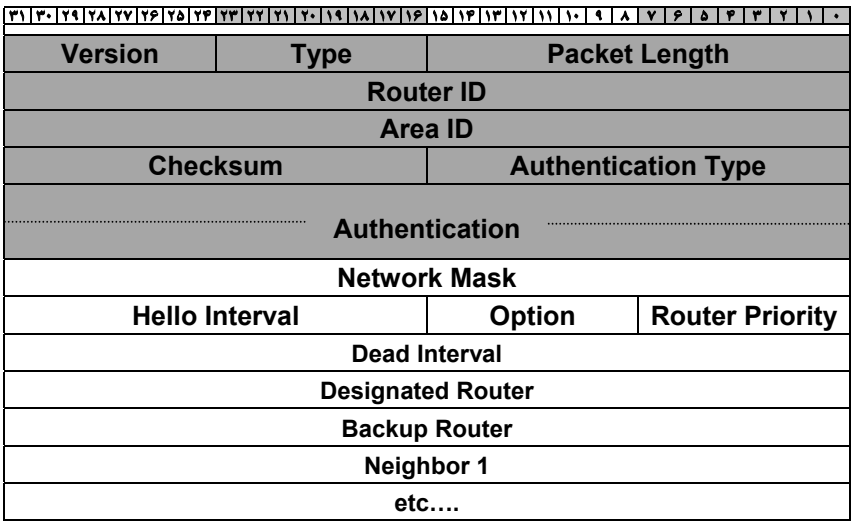
۱	**Hello Message**
۲	**Database Description Message**
۳	**Link State Request Message**
٤	**Link State Update Message**
٥	**Link State Acknowledgement Message**

- **فیلد Packet Length** : این فیلد طول کل پیام را بر حسب بایت مشخص می‌نماید. (طول پیام شامل قسمت سرآیند پیام نیز هست)
- **فیلد Router ID** : این فیلد چهاربایتی "مشخصهٔ مسیریاب"[1] ارسال‌کنندهٔ پیام را تعیین می‌نماید.
- **فیلد Area ID** : این فیلد چهاربایتی "مشخصهٔ ناحیه‌ای" که مسیریاب ارسال‌کنندهٔ پیام به آن ناحیه متعلق است ، تعیین می‌نماید.
- **فیلد Checksum** : این فیلد ، یک کد کشف خطای ١٦ بیتی از کل پیام (شامل سرآیند و بدنهٔ پیام) است که با روشی که برای بسته‌های IP گفته شد ، محاسبه می‌شود.
- **فیلد Authentication Type** : این فیلد که فعلاً دو مقدار برای آن تعریف شده مشخص می‌کند که آیا برای این پیام ، احراز هویت انجام شود یا خیر :

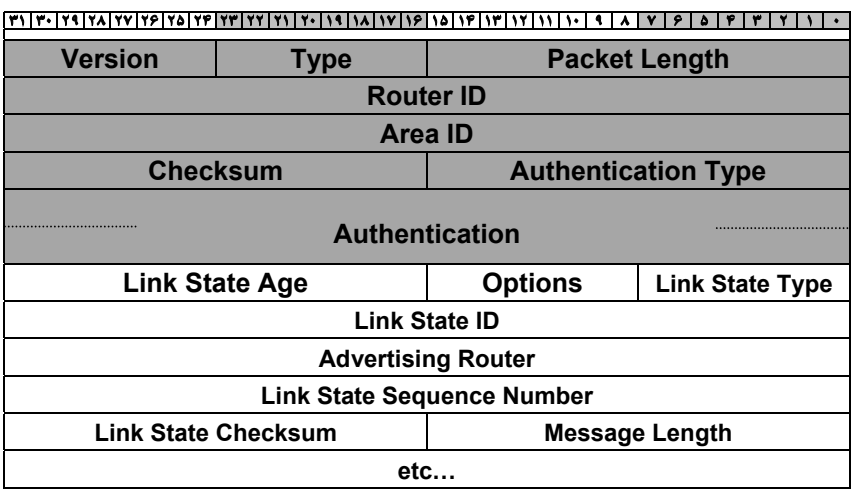
  مقدار ۰ : بدون احراز هویت (با این مقدار فیلد ٦٤ بیتی بعدی اهمیتی نخواهد داشت.)

  مقدار ۱ : احراز هویت وجود دارد و مقدار فیلد ٦٤ بیتی بعدی کلمهٔ عبور مسیریاب را تعیین می‌کند.
- فیلد Authentication : این فیلد در حقیقت کلمهٔ عبور مسیریاب به شمار می‌آید ودر هنگام نصب و تنظیم شبکه ، توسط مسئول آن تعیین می‌شود. مسیریابهای دریافت کنندهٔ این بسته به شرطی آنرا می‌پذیرند که کلمهٔ عبور آن مسیریاب معتبر و تعریف‌شده باشد.

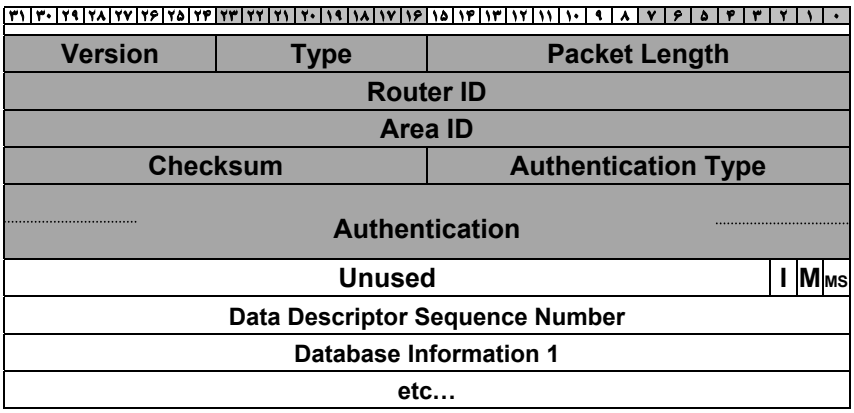
[1] Router Identifier



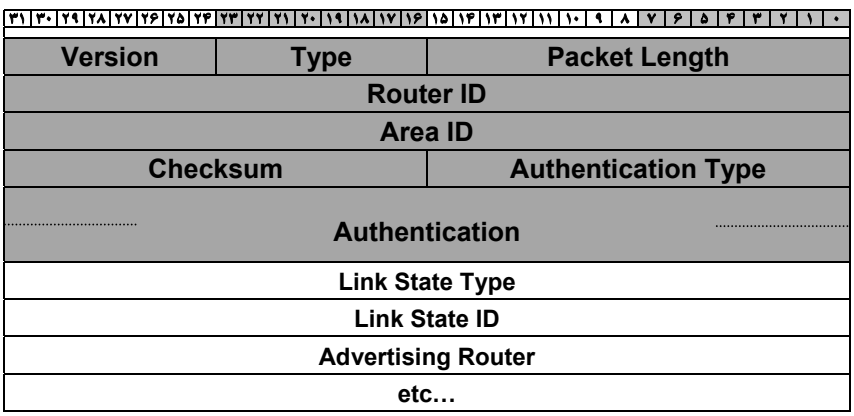
شکل (۳۱-٤) قالب پیام **"سلام"** -Hello Packet- در پروتکل OSPF



شکل (۳۲-٤) قالب پیام Link State Update در پروتکل OSPF



شکل (۳۳-٤) قالب پیام ارسال بانک اطلاعاتی در پروتکل OSPF



شکل (۳٤-٤) قالب پیام "تقاضا" در پروتکل OSPF

دقت کنید که هدف از ارائهٔ قالب پیامهای پروتکل OSPF ، آشنایی کلی با ساختار آنها و مقایسه با نکاتی است که در مورد تشکیل بستههای LS در بخشهای قبلی به آن اشاره شد. برای آشنایی دقیق با این پروتکل نیاز به مستندات IETF دارید که خود به اندازهٔ یک کتاب ، مفصل است. مراجع مختلفی که میتوانید به آنها مراجعه کنید در آخر این کتاب آمده است. در ضمن از سایت زیر هم میتوانید استفاده کنید:

http://www.cis.ohio-state.edu/hypertext/information/rfc.html

۸) پروتکل BGP : پروتکل مسیریابی برونی[۱]

همانگونه که مؤکداً اشاره کردیم در شبکه‌ای مثل اینترنت ، مسیریابی در درون یک شبکهٔ خودمختار (AS) با مسیریابی خارج از شبکه‌های خودمختار ، کاملاً متفاوت است و از پارامترهای متنوع و متناقضی تبعیت می‌کند. "مسیریابی برونی" نه تنها تابع شرایط ترافیکی ، توپولوژیکی ، پهنای باند و سرعت پردازش مسیریابها است ، بلکه از یکسری سیاستهای اقتصادی ، امنیتی و ملّی نیز تاثیر می‌پذیرد. شاید این نکته جالب باشد که مسیریابهائی که بین شبکه‌های خودمختار عمل می‌کنند نه تنها بایستی در مورد بهترین مسیر از لحاظ تاخیر کمتر و سرعت بیشتر تصمیم بگیرند بلکه بایستی در این تصمیم گیری مسائل سیاسی ، اقتصادی و امنیتی را نیز بعنوان جزئی از الگوریتم دخالت بدهند. بعنوان مثال یک مسیریاب در آمریکا شاید نخواهد بسته های IP که مبداً آنها مراکز نظامی و امنیتی آمریکا هستند از مسیرهائی عبور کنند که کشورهایی مثل کره شمالی ، عراق و لیبی در آن مسیر قرار دارند؛ یا مثلاً امنیت اقتصادی ایجاب می‌کند ، بسته‌های IP که مبداً آن IBM است بهیچوجه از مسیریاب Microsoft عبور ننمایند.

الگوریتمهای مسیریابی بین شبکه‌های خودمختار در اینترنت ، BGP[۲] نامیده می‌شود و تمامی این پارامترها را در تصمیم گیری لحاظ می‌کند. شبکه‌های خودمختار را یک ستون‌فقرات خارجی به هم متصل کرده که مسیریابهای نوع BGP بر روی آن قرار گرفته‌اند. بنابراین مسیریابهای نوع BGP از طریق ستون فقرات شبکهٔ اینترنت با مسیریابهای دیگر از نوع BGP در ارتباطند . ممکن است بین دو مسیریاب از نوع BGP یک شبکهٔ خودمختار قرار گرفته باشد که کل آن شبکه یک خط واحد در نظر گرفته می‌شود (به شکل (۲۰-٤) دقت کنید). حال به اختصار بررسی می‌کنیم که این مسیریابها چگونه عمل می‌نمایند:

در پروتکل BGP بجای آنکه جداول مسیریابی و هزینه‌ها بین مسیریابهای مجاور مبادله شود ، در بازه‌های زمانی T ثانیه‌ای ، فهرستی از مسیرهای کامل بین هر دو مسیریاب در شبکه ، برای مسیریابهای مجاور ارسال می‌شود. (بدون تعیین هزینه)

برای روشن شدن قضیه فرض کنید در شکل (۳٤-٤) مسیریاب F از مسیریابهای مجاور خود یعنی B ، I ، G و E ، اطلاعاتی در مورد D بصورت زیر دریافت کند:

تعیین مسیر رسیده از B From B : "I use BCD"

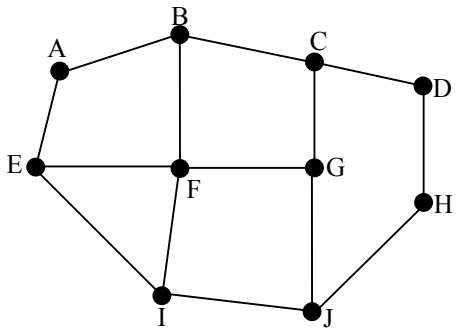
تعیین مسیر رسیده از G From G : "I use GCD"

تعیین مسیر رسیده از I From G : "I use IFGCD"

تعیین مسیر رسیده از E From G: "I use EFGCD"

---

[۱] The Exterior Gateway Routing Protocol
[۲] Border Gateway Protocol



شکل (۳۵-٤) ساختار فرضی از ارتباط بین مسیریابهای BGP

البته از B و G و I و E مسیرهای دیگری که به سایر مسیریابها وجود دارد ، اعلان می‌شود ولی برای سادگی در مثال فوق ، فقط مسیرهائی که به D ختم می‌شوند ارائه شده است. مسیریاب F برای یافتن راهی به D چهار مسیر پیشنهاد شده ، توسط همسایگانش را بررسی می‌کند و همان ابتدا مسیرهای I و E را کنار می‌گذارد چرا که این مسیرها از خود F عبور می‌کنند و بنابراین مسیری مستقل محسوب نمی‌شوند.

بنابراین F برای رسیدن به D فقط دو راه خواهد داشت که یکی BCD و دیگری GCD است و بایستی یکی از این دو مسیر را انتخاب کند. در اینجاست که باید تمامی پارامترهای ترافیکی ، توپولوژیکی ، اقتصادی ، سیاسی و امنیتی برای انتخاب بهینه‌ترین مسیر ، محاسبه شود. بهمین دلیل نحوه **"امتیازدهی"**[۱] به مسیرها بعنوان بخشی ازپروتکل BGP محسوب نمی‌شود و به ملاحظات مدیریتی هر مسیریاب وابسته است و توسط مسئول شبکه تنظیم می‌شود.

الگوریتم BGP مشکل "شمارش تا بینهایت"[۲] را نخواهد داشت : بعنوان مثال فرض کنید خط بین F و G قطع شود. F اطلاعاتی را از سه مسیریاب مجاور دریافت می‌کند که اعلان کرده‌اند برای رسیدن به D ، از مسیرهای BCD ، IFGCD ، EFGCD استفاده میکنند و چون در بین مسیرهای IFGCD ، EFGCD ، خط FG قرار دارد ، بنابراین نمی‌توانند به عنوان مسیرهایی مستقل مورد استفاده قرار گیرند و تنها مسیر باقیمانده BCD خواهد بود.

---

۱ Scoring Function
۲ Count To Infinity

الگوریتمهائی که در تبادل اطلاعات با همسایگان مسیرهای کامل را به اطلاع یکدیگر می‌رسانند اولاً مشکل "شمارش تا بینهایت" را نخواهد داشت؛ ثانیاً مسیریابهای دیگر میتوانند بر روی کل مسیر ، بررسی‌های امنیتی ، اقتصادی ، سیاسی و ملی انجام دهند و بر اساس این پارامترها مسیر مناسب را انتخاب نمایند.

در ادبیات پروتکل BGP ، مسیریابهای مجاور "مسیریابهای همتا"[۱] نامیده شده است. اطلاعات مسیریابی (فهرست مسیرها) در قالب پیامهایی بین مسیریابهای همتا مبادله می‌شود. در پروتکل BGP چهار نوع پیام تعریف شده است:

- **پیام OPEN** : مسیریابهای همتا از طریق پروتکل TCP و شمارهٔ پورت ۱۷۹ با یکدیگر مبادلهٔ اطلاعات می‌کنند. (در فصل بعد خواهید دید که استفاده از پروتکل TCP علیرغم اندکی کاهش سرعت انتقال ، صحت داده‌ها را تضمین می‌کند.) وقتی یک مسیریاب به شبکه وارد می‌شود با ارسال این پیام برای مسیریابهای همتای خود ، ضمن اعلام موجودیت ، خود را معرفی می‌کند تا دیگران بتوانند هویت او را احراز کنند. اگر مسیریابهای همتا او را بپذیرند ، در پاسخ پیام KEEPALIVE را به عنوان تصدیق ، برخواهند گرداند.
- **پیام KEEPALIVE** : این پیام دو نوع کاربرد دارد:

کاربرد اول زمانی است که یک مسیریاب با پیام OPEN در شبکه اعلام حضور می‌کند؛ در این حالت مسیریابهای همتا پس از احراز هویت ، این پیام را به عنوان پیغام تصدیق (Ack) برمی‌گردانند.

کاربرد دوم زمانی است که یک مسیریاب BGP در موعد اعلان "فهرست مسیرها" ، چیزی برای ارسال نداشته باشد؛ در این حالت با ارسال این پیام اعلام می‌کند که در شبکه حضور داشته و فعال است. اگر مسیریابی در یک بازهٔ زمانی مشخص هیچ پیامی از مسیریاب همتای خود دریافت نکند ، فرض خواهد کرد که آن مسیریاب به هر دلیلی از شبکه خارج شده است و بر این اساس فهرست مسیرهای خود را اصلاح خواهد کرد.

- **پیام NOTIFICATION** : با این پیام یک مسیریاب به همتای خود اعلام می‌کند که در دریافت پیام قبلی او خطایی رخ داده است. در حقیقت این پیام نقش "اعلان عدم تصدیق" (NAck[۲]) را بازی می‌کند.
- **پیام UPDATE** : مسیریاب با این پیام ، مسیر مورد استفاده‌اش برای رسیدن به یک مقصد خاص در شبکه را ، به اطلاع همتای خود می‌رساند. در ضمن اگر مسیری که قبلاً اعلان شده ، دیگر معتبر و در دسترس نباشد با این پیام به اطلاع مسیریابهای همتا می‌رسد.

---

[۱] Peer BGP Router
[۲] Not Acknowledged

برای اطلاع بیشتر از جزئیات پروتکل BGP به مستندات RFC-1263 و RFC-1654 مراجعه نمائید.

۹) مراجع این فصل

مجموعهٔ مراجع زیر می‌توانند برای دست آوردن جزییات دقیق و تحقیق جامع در مورد پروتکلهای معرفی شده در این فصل مفید واقع شوند.

مراجع مفید در مبحث مسیریابی

Cisco97	"Interior Gateway Routing Protocol and Enhanced IGRP " http://www.cisco.com/univercd/cc/td/doc/csintwk/ito_doc/55182.htm
Halabi 97	B. Halabi, Internet Routing Architectures, Cisco Systems Publishing, Indianapolis, 1997.
Huitema	C. Huiteman, Routing in the Internet, Prentice Hall, New Jersey, 1995.
RFC1058	"Routing Information Protocol," Hedrick, C.L.; 1988
RFC1074	"NSFNET Backbone SPF-Based Interior Gateway Protocol," Rekhter, J.; 1988
RFC1136	"Administrative Domains and Routing Domains: A Model for Routing in the Internet," Hares, S.; Katz, D.; 1989
RFC1163	"Border Gateway Protocol (BGP)," Lougheed, K.; Rekhter, Y.; 1990
RFC1164	"Application of the Border Gateway Protocol in the Internet," Honig, J.C.; Katz, D.; Mathis, M.; Rekhter, Y.; Yu, J.Y.; 1990
RFC1195	"Use of OSI IS-IS for Routing in TCP/IP and Dual Environments," Callon, R.W.; 1990
RFC1222	"Advancing the NSFNET Routing Architecture," Braun, H.W.; Rekhter, Y.; 1991
RFC1247	"OSPF version 2," Moy, J.; 1991
RFC1256	S. Deering, "ICMP Router Discovery Messages," RFC 1256, Sept. 1991.
RFC1267	"A Border Gateway Protocol 3 (BGP-3)," Lougheed, K.; Rekhter, Y.; 1991
RFC1584	J. Moy, "Multicast Extensions to OSPF," RFC 1584, March 1994.
RFC1723	G. Malkin, RIP Version 2 - Carrying Additional Information. RFC 1723, November 1994.
RFC1771	Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," RFC 1771, March 1995.
RFC1772	Y. Rekhter and P. Gross, "Application of the Border Gateway Protocol in the Internet," RFC 1772, March 1995.
RFC1773	P. Traina, "Experience with the BGP-4 protocol," RFC 1773, March 1995
RFC2002	C. Perkins, "IP Mobility Support," RFC 2002, 1996.
RFC2178	J. Moy, "Open Shortest Path First Version 2", RFC 2178, July 1997.
RFC823	"DARPA Internet Gateway," Hinden, R.M.; Sheltzer, A.; 1982
RFC827	"Exterior Gateway Protocol (EGP)," Rosen, E.C.; 1982
RFC888	"STUB Exterior Gateway Protocol," Seamonson, L.; Rosen, E.C.; 1984
RFC904	"Exterior Gateway Protocol Formal Specification," Mills, D.L.; 1984
RFC911	"EGP Gateway under Berkeley UNIX 4.2," Kirton, P.; 1984

مراجع مفید در مبحث کارایی و سیاستهای مسیریابی

RFC1102	"Policy Routing in Internet Protocols," Clark, D.D.; 1989
RFC1104	"Models of Policy-Based Routing," Braun, H.W.; 1989
RFC1124	"Policy Issues in Interconnecting Networks," Leiner, B.M.; 1989
RFC1125	"Policy Requirements for Inter-Administrative Domain Routing," Estrin, D.; 1989
RFC1245	"OSPF Protocol Analysis," Moy, J., ed; 1991
RFC1246	"Experience with the OSPF Protocol," Moy, J., ed.; 1991
RFC1254	"Gateway Congestion Control Survey," Mankin, A.; Ramakrishnan, K.K, eds.; 1991