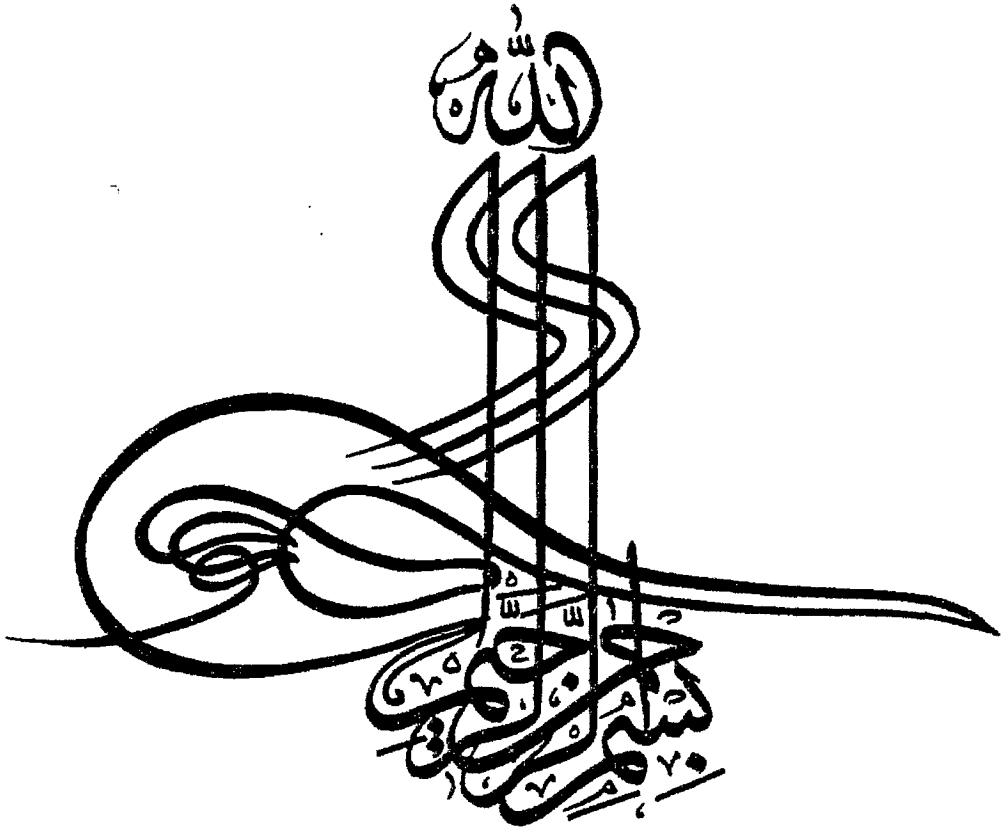


١٠٨٣٤

۸۷/۹/۱۰۰۷۲۱
۸۷/۱۰/۱۵

دانشگاه شهید بهشتی
دانشکده مهندسی برق و کامپیوتر

بکارگیری الگوریتم‌های بر پایه جرگه مورچه‌ها برای راه‌یابی در شبکه‌های رایانه‌ای

پایان نامه کارشناسی ارشد مهندسی کامپیوتر
گرایش معماری کامپیوتر

شهاب الدین رحمتی‌زاده

۱۳۸۷/۱۰/۱ - ۶

استاد راهنما:
دکتر حامد شاه‌حسینی

استاد مشاور:
دکتر مقصود عباسپور

۱۳۸۷

۱۰۸۳۴۷

دانشگاه شهید بهشتی

دانشکده مهندسی برق و کامپیوتر

پایان نامه کارشناسی ارشد مهندسی کامپیوتر- گرایش معماری‌کامپیوتر

تحت عنوان:

بکارگیری الگوریتم‌های بر پایه جرگه مورچه‌ها برای راه‌یابی در شبکه‌های رایانه‌ای

در تاریخ ۱۳۸۷/۶/۲۶ پایان نامه دانشجو، شهاب الدین رحمتی‌زاده، توسط کمیته تخصصی داوران مورد بررسی و تصویب نهائی قرار گرفت.

۱- استاد راهنما: جناب آقای دکتر حامد شاه‌حسینی

۲- استاد مشاور: جناب آقای دکتر مقصود عباسپور

۳-استاد داور (داخلی): جناب آقای دکتر کیوان ناوی

۱۳۸۷/۱۰/ -۶

۴- استاد داور (خارجی): سرکار خانم دکتر چیترا دادخواه

۵- نماینده تحصیلات تکمیلی: جناب آقای دکتر علیرضا رضازاده ولوجردی

به نام آنکه جان را فکرت آموخت

وَ أَوْحَى رَبُّکَ إِلَى النَّحْلِ أَنِ اتخِذِي مِنَ الْجِبَالِ بُيُوتاً وَ مِنَ الشجَرِ وَ مِمَّا يَعْرِشونَ (نحل ۶۸)

و پروردگارت به زنبور عسل وحی کرد که از کوه‌ها و درختان و آنچه برمی‌افرازند خانه بساز.

هوشمندی در تمام طبیعت جاری و ساری است. خیلی وقت‌ها برای حل مشکلات می‌بایست به طبیعت نگریست. برای نمونه، مورچه‌ها توانایی یافتن کوتاه‌ترین راه از لانه به توده غذای را دارند. همچنین مدل رفتاری زنبورها در علوم هوافضا کاربردهایی پیدا کرده است و پژوهشگران برای تنظیم دما سعی دارند از روش موجود در گیاه کلم بهره ببرند. در پایان‌نامه‌ای که پیش رو دارید برای حل مشکل راه‌یابی در شبکه‌های رایانه‌ای از رفتار جرگه مورچه‌ها و زنبورها الهام گرفته شده است.

نخست پروردگار خویش را سپاس می‌گزارم که هر چه دارم لایق آن نیستم و بی لطف و منت او توفیقی میسر نمی‌گردد. سپس می‌خواهم زیباترین واژه‌های عشق را نثار پدر و مادر خود نمایم که این‌چنین صبورانه راهگشای زندگیم بوده‌اند و خود را تا آخر عمر مدیون آنان می‌دانم.

در ادامه لازم می‌دانم از زحمات، تلاش‌ها و راهنمایی‌های بسیار گرانقدر استاد گرامیم جناب آقای دکتر حامد شاه‌حسینی صمیمانه سپاسگزاری نمایم. بی‌شک بدون کمک‌های دلسوزانه ایشان تحقق این امر امکان پذیر نبود.

در پایان بجاست از زحمات خواهر عزیزم و همچنین دوستان بسیار ارزشمندم، علی بصیری، علی دانشپورمقدم، حامد هادی‌پور، احمد محمودی، آزاده منصوری، سعید گرگین، پیمان لؤلؤئیان، حسین ترکمن، سعید طلعتیان، علی صیانتی و نیما بهنود و نیز مسئولان محترم دانشکده که موجبات دلگرمی حقیر را فرآهم نمودند تشکر نمایم.

بار درخت علم ندانم مگر عمل　　با علم اگر عمل نکنی شاخ بی بری

هر علم را که کار نبندی چه فایده　　چشم از برای آن بود آخر که بنگری

شهاب الدین رحمتی‌زاده

تابستان ۸۷

کلیه حقوق مادی مترتب بر نتایج مطالعات،

ابتکارات و نوآوریهای ناشی از تحقیق موضوع

این پایان نامه متعلق به دانشگاه شهید بهشتی

می باشد.

به نام خدا

نام ونام خانوادگی: شهاب الدین رحمتی‌زاده

عنوان پایان نامه: بکارگیری الگوریتم‌های بر پایه جرگه مورچه‌ها برای راه‌یابی در شبکه‌های رایانه‌ای

استاد/اساتید راهنما: دکتر حامد شاه‌حسینی

اینجانب شهاب الدین رحمتی‌زاده تهیه کننده پایان نامه کارشناسی ارشد/دکتری حاضر خود را ملزم به حفظ امانت داری و قدردانی از زحمات سایر محققین و نویسندگان بنا بر قانون Copyright می‌دانم. بدین وسیله اعلام می نمایم که مسئولیت کلیه مطالب درج شده با اینجانب می‌باشد و در صورت استفاده از اشکال؛ جداول، و مطالب سایر منابع، بلافاصله مرجع آن ذکر شده و سایر مطالب از کار تحقیقاتی اینجانب استخراج گشته است و امانتداری را به صورت کامل رعایت نموده ام. در صورتی که خلاف این مطلب ثابت شود، مسئولیت کلیه عواقب قانونی با شخص اینجانب می‌باشد.

نام ونام خانوادگی دانشجو: شهاب الدین رحمتی‌زاده

امضاء وتاریخ:

فصل اول الگوریتم‌های راه‌یابی ۱

۱-۱ مقدمه ۱

۱-۲ راه‌یابی بردار فاصله ۲

۱-۳ الگوریتم‌های حالت لینک ۴

۱-۴ راه‌یابی در اینترنت ۶

۱-۴-۱ پروتکل RIP جهت راه‌یابی دروازه‌های درونی ۶

۱-۴-۲ پروتکل OSPF جهت راه‌یابی دروازه‌های درونی ۷

۱-۴-۳ پروتکل BGP جهت راه‌یابی دروازه‌های بیرونی ۹

۱-۵ خلاصه ۱۰

فصل دوم بهینه سازی با بهره‌گیری از جرگه مورچه‌ها ۱۲

۲-۱ مقدمه ۱۲

۲-۲ هوش گروهی ۱۲

۲-۳ یافتن راه کوتاه‌تر توسط جرگه مورچه‌ها ۱۳

۲-۴ فرآیند بهینه‌سازی توسط جرگه مورچه‌ها ۱۴

۲-۵ مورچه ساختگی ۱۵

۲-۶ مورچه‌های طبیعی، مورچه‌های ساختگی ۱۶

۲-۷ برخی از کاربردهای الگوریتم‌های بهینه سازی جرگه مورچه‌ها ۱۸

۲-۷-۱ مشکل فروشنده دوره‌گرد ۱۹

۲-۷-۲ مشکل واگذاری مربعی ۱۹

۲-۷-۳ مشکل برنامه‌ریزی کار ۲۰

۲-۷-۴ مشکل راه‌یابی وسایل نقلیه ۲۰

۲-۷-۵ مشکل رنگ آمیزی گراف ۲۱

۲-۸ خلاصه ۲۲

فصل سوم راه‌یابی بر پایه جرگه مورچه‌ها ۲۳

۳-۱ مقدمه ۲۳

۳-۲ راه‌یابی در شبکه‌های اتصال‌گرا ۲۳

۳-۳ راه‌یابی در شبکه‌های بدون اتصال ۲۶

۳-۴ الگوریتم AntNet ۲۸

۳-۴-۱ جدول راه‌یابی در الگوریتم AntNet ۲۸

۳-۴-۲ جدول آمار ترافیک محلی در الگوریتم AntNet ۲۹

۳-۴-۳ گزینش گره مقصد برای مورچه‌های پیشرو ۳۰

۳-۴-۴ چگونگی گزینش پرش بعدی ۳۱

۳-۴-۵ چگونگی کارکرد مورچه‌های پسرو ۳۲

۳-۴-۶ تشخیص و چگونگی برخورد با حلقه ۳۵

۳-۵ مقایسه راه‌یابی بر پایه مورچه با روش‌های مرسوم ۳۶

۳-۵-۱ بررسی چگونگی دستیابی به آگاهی‌های راه‌یابی ۳۶

۳-۵-۲ بررسی سربار در الگوریتم‌ها ۳۷

۳-۵-۳ بررسی ایستایی و پویایی در الگوریتم‌ها ۳۷

۳-۶ نتایج AntNet ۳۸

۳-۷ خلاصه ۳۹

فصل چهارم نسخه‌های بهبودیافته الگوریتم ANTNET ۴۰

۴-۱ مقدمه ۴۰

۴-۲ تغییر اولویت‌ها ۴۰

۴-۳ افزایش کارایی ۴۱

۴-۳-۱ ارزش‌دهی نخستین هوشمندانه جدول‌های راه‌یابی ۴۱

۴-۳-۲ بهنگام سازی هوشمندانه در صورت بروز یک خرابی ۴۲

۴-۳-۳ بکاربردن پارامتری برای جلوگیری از گیرافتادگی ۴۳

۴-۳-۴ دو روش برای گزینش گره بعدی؛ بختانه و قطعی ۴۳

۴-۳-۵ کنترل بر روی تعداد مورچه درون شبکه ۴۳

۴-۳-۶ خودکشی مورچه‌های گیر افتاده در حلقه ۴۴

۴-۴ افزایش سرعت همگرایی الگوریتم ۴۴

۴-۵ روش‌هایی جهت کنترل فرومون ۴۵

۴-۵-۱ تبخیر فرومون ۴۶
۴-۵-۲ سالخوردگی ۴۶
۴-۵-۳ محدودیت مقدار فرومون ۴۶

۴-۶ بهینه‌سازی جرگه مورچه‌های چندگانه ۴۷

۴-۷ زدودن پیشینه مورچه ۴۷
۴-۷-۱ الگوریتم AntNet-CHTT ۴۸

۴-۸ خلاصه ۵۰

فصل پنجم حل دشواری‌های بهینه‌سازی به یاری زنبورها ۵۱

۵-۱ مقدمه ۵۱

۵-۲ زنبورهای طبیعی ۵۱

۵-۳ بهینه‌سازی جرگه زنبورها ۵۲
۵-۳-۱ الگوریتم ABC ۵۳

۵-۴ خلاصه ۵۵

فصل ششم الگوریتم راه‌یابی مورچه-زنبور پیشنهادی ۵۶

۶-۱ مقدمه ۵۶

۶-۲ الگوریتم راه‌یابی مورچه-زنبور ۵۶
۶-۲-۱ جدول راه‌یابی ۵۷
۶-۲-۲ فرآیند کاری زنبورهای سه‌گانه ۵۸

۶-۳ خلاصه ۵۹

فصل هفتم آزمایش‌های انجام گرفته ۶۱

۷-۱ مقدمه ۶۱

۷-۲ بررسی توزیع زنبورهای سه‌گانه تحت یک توپولوژی ثابت ۶۲

۷-۳ آزمایش‌هایی برای سنجش تاثیر پارامتر β ۶۵

۷-۴ بررسی کارکرد الگوریتم در برخورد با خرابی دو گره ۷۱
۷-۵ تغییری در الگوریتم مورچه-زنبور و بررسی کارکرد آن ۷۳
۷-۶ خلاصه ۷۷
فصل هشتم نتیجه‌گیری و پیشنهادهایی برای کارهای آتی ۷۸
۸-۱ مقدمه ۷۸
۸-۲ نتیجه‌گیری ۷۸
۸-۳ پیشنهادهایی برای کارهای آتی ۷۹
۸-۴ خلاصه ۷۹

چکیده:

در چند سال سپری شده، دانشمندان با الگو برداری از رفتار جرگه[1] حشره‌های اجتماعی در طبیعت به معرفی الگوریتم‌هایی پرداخته‌اند. مورچه‌ها و زنبورها دو گونه از حشره‌های اجتماعی هستند که رفتارشان بسیار زیاد مورد توجه پژوهشگران قرار گرفته است. یک مورچه هنگام حرکت یک ماده شیمیایی به نام فرومون از خود باقی می‌گذارد که مورچه‌های کاوشگر دیگر چنین جاپاهایی را دنبال می‌کنند. بهینه‌سازی جرگه مورچه‌ها[2] دیدگاهی است که به تازگی برای گشایش مشکل‌های بهینه‌سازی مطرح شده است. این بحث بر پایه بر جا گذاشتن ماده‌ای به نام فرومون و دنبال کردن رفتار مورچه‌ها در دنیای واقعی بوده است. بهبود در حل بسیاری از دشواری‌ها مانند مشکل فروشنده دوره‌گرد[3]، برنامه‌ریزی کار[4]، رنگ آمیزی گراف[5] و راه‌یابی[6] در شبکه‌های رایانه‌ای به‌وسیله الگو برداری از چگونگی رفتار هوشمندانه جرگه مورچه‌ها فراهم گشته است. دستاوردهایی که به کمک الگوریتم‌های بهینه‌سازی جرگه مورچه‌ها بدست می‌آید به خوبی سایر روش‌های هیوریستیک[7] می‌باشد.

راه‌یابی، روشی برای گذر داده‌ها از مبدا[8] به مقصد[9] می‌باشد. این فرآیند به کمک یک دنباله‌ای از گره‌های شبکه انجام می‌پذیرد. مشکلی که بایستی هر الگوریتم راه‌یابی بگشاید آن است که رفت و آمد (ترافیک) داده‌ها از مبدا به مقصد را به گونه‌ای راهنمایی کند تا هزینه‌ها کمینه شوند و کارایی شبکه بیشینه گردد. در الگوریتم‌های بر پایه جرگه مورچه‌ها، مورچه‌ها از خود فرومون بجا می‌گذارند و در پی آن جدول‌های راه‌یابی در هر گره از شبکه بهنگام می‌شود. در این پایان‌نامه ابتدا الگوریتم‌های بر پایه جرگه مورچه‌ها برای راه‌یابی در شبکه‌های کامپیوتری مورد بررسی قرار گرفته و سپس با توجه به شیوه کارکرد زنبورها در طبیعت، الگوریتم جدیدی به نام مورچه-زنبور برای راه‌یابی در شبکه‌های رایانه‌ای معرفی می‌گردد. چگونگی کارکرد این الگوریتم بر اساس یک مدلی از همکاری مورچه‌ها و زنبورهای ساختگی می‌باشد. در انتها با شبیه‌سازی‌های انجام گرفته به مقایسه الگوریتم پیشنهادی جدید با دیگر روش‌ها پرداخته و مشکلات و موانع موجود بررسی خواهند شد.

---

[1] Colony
[2] Ant Colony Optimization (ACO)
[3] Traveling Salesman Problem (TSP)
[4] Job Scheduling Problem (JSP)
[5] Graph Coloring Problem (GCP)
[6] Routing
[7] Heuristic
[8] Source
[9] Destination

فصل اول
الگوریتم‌های راه‌یابی

۱-۱ مقدمه

الگوریتم راه‌یابی، یک فرمول یا فرآیند ریاضیاتی است که با اعمال آن بر روی جدول راه‌یابی، میزان مطلوبیت یا میزان بهینگی راه‌های مختلف محاسبه شده و بر اساس آن مطلوب‌ترین و بهینه‌ترین راه بدست می‌آید.

در فرآیند راه‌یابی دو مورد مهم مطرح است[۱]:

- هر یک از راه‌یاب‌ها چگونه از پارامتر هزینه کل ارتباط‌های شبکه آگاه شوند تا بتوانند گراف زیر ساخت ارتباطی شبکه را تشکیل داده و بهترین راه را گزینش نمایند.
- چه الگوریتمی برای یافتن بهترین راه گزینش شود تا از جهت سرعت پردازش و تصمیم‌گیری، بهینه بوده و بسته‌ها را با تاخیر و انتظار مواجه نکند.

می‌توان الگوریتم‌های راه‌یابی را در دو گروه سراسری[1] و نا متمرکز[2] دسته‌بندی نمود. یک الگوریتم راه‌یابی سراسری، کم هزینه‌ترین راه بین مبدا و مقصد را به کمک دانش کامل و سراسری از شبکه محاسبه می‌کند. بنابراین می‌بایست ابتدا اطلاعات تمامی لینک‌ها و هزینه‌های هر یک گردآوری شده سپس محاسبات صورت پذیرد. در عمل این نوع الگوریتم‌ها به الگوریتم‌های حالت لینک[3] شناخته می‌شوند.

در یک الگوریتم راه‌یابی نامتمرکز، محاسبه کم هزینه‌ترین راه از طریق اجرای یک روش تکرار شونده صورت می‌پذیرد. هیچ گره‌ای آگاهی کاملی از ارتباطات و هزینه لینک‌های شبکه ندارد و تنها از هزینه‌ی لینک‌های مستقیم خود آگاه می‌باشد از این رو با ارسال مرتب اطلاعات برای همسایه‌ها و انجام محاسبات، راه بهینه بدست می‌آید. به این نوع الگوریتم‌ها، الگوریتم‌های بردار فاصله[4] گفته می‌شود زیرا هر راه‌یاب یک بردار ارزیابی هزینه را نگهداری می‌کند[۲].

---

[1] Global
[2] Decentralized
[3] Link State Algorithms
[4] Distance Vector Algorithms

تقسیم‌بندی فوق از دیدگاه چگونگی جمع‌آوری و پردازش اطلاعات زیرساخت ارتباطی شبکه انجام گرفت. می‌توان الگوریتم‌های راه‌یابی را از دیدگاه روش تصمیم‌گیری و میزان هوشمندی نیز تقسیم‌بندی نمود. از این دیدگاه الگوریتم‌های راه‌یابی به دو دسته زیر تقسیم می‌شوند

- الگوریتم‌های ایستا[1]
- الگوریتم‌های پویا[2]

در الگوریتم‌های ایستا هیچ توجه‌ای به شرایط توپولوژیکی و ترافیک لحظه‌ای شبکه نمی‌شود. در این الگوریتم‌ها برای هدایت یک بسته[3]، هر راه‌یاب از جدولی استفاده می‌کند که در هنگام پیکربندی راه‌یاب‌ها تنظیم شده و در طول زمان ثابت می‌ماند. در هنگام وقوع هرگونه تغییر در توپولوژی زیر ساخت شبکه، این جدول دوباره می‌بایست توسط مدیر شبکه به صورت دستی تنظیم شود.

در الگوریتم‌های پویا، راه‌یابی بر اساس آخرین وضعیت توپولوژیکی و ترافیکی شبکه انجام می‌شود و جدول‌های راه‌یابی در این نوع الگوریتم‌ها هر T ثانیه یکبار بهنگام می‌شود. این الگوریتم‌ها بر اساس وضعیت فعلی شبکه تصمیم گیری می‌نمایند ولی ممکن است پیچیدگی این الگوریتم‌ها به‌قدری زیاد باشد که زمان تصمیم‌گیری برای گزینش بهترین راه، خودش طولانی و منجر به تاخیرهای بحرانی بشود.

۱-۲ راه‌یابی بردار فاصله

در راه‌یابی بردار فاصله هر راه‌یاب یک جدول راه‌یابی دارد که به ازای هر راه‌یاب موجود در زیر شبکه، یک درایه در آن درج شده است. هر درایه دارای دو بخش است: خط خروجی مناسب برای رسیدن به مقصد مورد نظر و تخمینی از زمان یا فاصله رسیدن بدان مقصد. معیار هزینه می‌تواند تعداد گام[4]، تاخیر، تعداد کل بسته‌های به صف شده در آن راه و یا چیزی شبیه به این‌ها باشد.

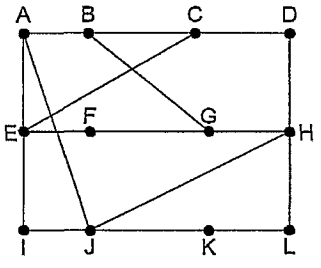
---

[1] Static Algorithms
[2] Dynamic Algorithms
[3] Packet
[4] Number of Hops

فرض بر آن است که هر راه‌یاب فاصله خود تا هر یک از همسایه‌هایش را می‌داند. اگر معیار به‌کار رفته تعداد گام باشد، فاصله هر راه‌یاب از همسایه‌هایش یک است. اگر معیار، طول صف[1] باشد راه‌یاب می‌تواند به‌سادگی هر یک از صف‌ها را بررسی نماید. اگر معیار تاخیر باشد راه‌یاب می‌تواند با ارسال بسته‌های خاصی به نام اکو[2] و دریافت بسته پاسخ (که گیرنده به آن مهر زمان[3] زده و به سرعت بر می‌گرداند) این تاخیر را به سادگی اندازه بگیرد.

فرض کنید از میزان تاخیر به عنوان معیار بهینگی استفاده شود. هر راه‌یاب تاخیر خود تا هر یک از همسایه‌ها را بداند. هر T میلی ثانیه یکبار، تمام راه‌یاب‌ها برای همسایه‌های خود فهرستی از تاخیر تخمینی در رسیدن به هر یک از راه‌یاب‌های مقصد را ارسال می‌نمایند. همچنین فهرست مشابهی را از همسایه‌های خود دریافت می‌کنند. تصور کنید یکی از این جدول‌ها از طرف راه‌یاب X دریافت شود و نماد X_i مشخص کننده میزان تاخیر برای رسیدن به راه‌یاب i باشد. اگر این راه‌یاب بداند که تاخیر خودش تا X ، معادل m میلی‌ثانیه است می‌تواند نتیجه بگیرد که با تاخیر $X_i + m$ میلی ثانیه به راه‌یاب i می‌رسد. با انجام این محاسبه برای تمام جدول‌هایی که از همسایه‌ها می‌رسند راه‌یاب می‌تواند نتیجه بگیرد که کدام یک از راه‌ها بهتر هستند و مقدار تخمینی تاخیر و همچنین خط متناظر با بهترین راه را در جدول جدید خود درج می‌نماید[۳].

در شکل ۱-۱ هر یک از راه‌یاب‌های A، I، H و K جدول‌های راه‌یابی خود را برای J ارسال کرده‌اند و راه‌یاب J نیز بر اساس روش بیان شده جدول خود را بهنگام کرده است.



	A	I	H	K		
A	0	24	20	21	8	A
B	12	36	31	28	20	A
C	25	18	19	36	28	I
D	40	27	8	24	20	H
E	14	7	30	22	17	I
F	23	20	19	40	30	I
G	18	31	6	31	18	H
H	17	20	0	19	12	H
I	21	0	14	22	10	I
J	9	11	7	10	0	–
K	24	22	22	0	6	K
L	29	33	9	9	15	K

شکل ۱-۱ :**یک شبکه نمونه و جدول‌های راه‌یابی بردارفاصله[۳]**

[1] Queue length
[2] Echo
[3] Timestamp

۱-۳ الگوریتم‌های حالت لینک

دومشکل اساسی موجب زوال یافتن الگوریتم‌های بردار فاصله اولیه گردید: نخست آنکه در این الگوریتم معیار تاخیر، طول صف در نظر گرفته می‌شد و پهنای باند هر یک از خطوط در محاسبات گزینش بهترین راه، دخالت داده نمی‌شد. البته این امکان وجود داشت که بتوان معیار تاخیر را به پهنای باند تغییر داد ولی مشکل دیگری ایجاد می‌شد و آن هم همگرایی[1] الگوریتم بود که بسیار طولانی می‌شد.

در الگوریتم‌های حالت لینک هر راه‌یاب باید پنج عمل زیر را انجام بدهد:

- راه‌یاب‌های همسایه خود را که به صورت فیزیکی به آن‌ها متصل است شناسایی کرده و آدرس آن‌ها را بدست آورد.
- هزینه راه‌یاب‌های همسایه خود را اندازه‌گیری نماید.
- یک بسته بسازد و تمام اطلاعاتی که از راه‌یاب‌های همسایه خود دارد در آن قرار دهد.
- بسته ساخته شده را به روش سیل‌آسا[2] برای تمام راه‌یاب‌های شبکه ارسال نماید و همچنین بسته‌هایی را که از راه‌یاب‌های دیگر می‌رسد دریافت و ذخیره کند.
- گراف شبکه را تشکیل داده و با استفاده از الگوریتمی مناسب، بهینه‌ترین راه بین دو راه‌یاب در شبکه را پیدا نماید.

هر راه‌یاب موظف است تمامی راه‌یاب‌های همسایه خود را شناسایی کرده و هزینه هر یک از خطوط خروجی خود را بدست آورد. راه‌یاب باید پس از جمع‌آوری اطلاعات لازم، بسته‌ای از این اطلاعات تشکیل بدهد. به این بسته، بسته حالت لینک[3] گفته می‌شود. در حالت کلی این بسته شامل اطلاعات زیر است:

- آدرس جهانی[4] راه‌یاب تولید کننده بسته
- یک شماره ترتیب (تا بسته‌های تکراری از بسته‌های جدید تشخیص داده شوند.)

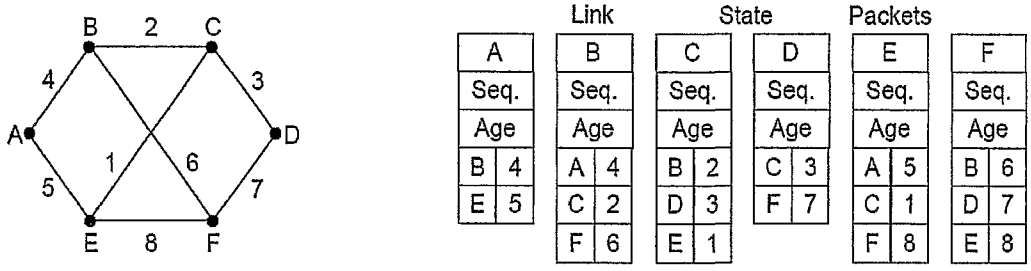
---

[1] Convergence
[2] Flooding
[3] Link State Packet
[4] IP Address

- طول عمر بسته (تا اطلاعات بسته، زمان پایان اعتبار داشته باشند.)

- آدرس جهانی راه‌یاب‌های همسایه و هزینه تخمینی

یکی از مسایل مهم در این نوع الگوریتم راهیابی، چگونگی توزیع بسته‌های حالت لینک روی شبکه است. قاعده کلی ارسال بسته‌های حالت لینک ، ارسال به روش سیل‌آسا است. برای آنکه این بسته‌ها در یک حلقه بی‌نهایت تکرار نشوند، هر بسته دارای یک شماره ترتیب است که با ورود به هر راه‌یاب ابتدا بررسی می‌شود که آیا این بسته قبلا دریافت شده است یا آنکه یک بسته جدید است. بنابراین هر راه‌یاب موظف است که شماره ترتیب بسته‌های حالت لینک 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 حالت لینک با شماره جدید و غیر تکراری ارسال شود. در شکل ۱-۲ بسته‌های حالت لینک را برای شبکه نمونه داده شده ملاحظه می‌کنید.



شکل ۱-۲ : یک شبکه نمونه و جدول راه‌یابی حالت لینک[۳]

وقتی راه‌یاب بسته‌ای را با یک شماره خاص دریافت کرد دیگر بسته‌های با شماره کوچکتر از آن شماره را دریافت نخواهد کرد. همین موضوع مشکل بزرگی برای راه‌یاب‌هایی که به ناگاه از کار می‌افتند و می‌خواهند مجدد به شبکه وارد شوند، پدید خواهد آورد؛ زیرا یک راه‌یاب که از شبکه خارج شده و می‌خواهد دوباره به شبکه بازگردد چاره‌ای ندارد جز اینکه شماره بسته‌ها را از صفر شروع کند و نتیجه آن می‌شود که هیچ راه‌یابی آن را نخواهد پذیرفت. جهت 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 راه‌یابی بسته‌ای را دریافت و آن را در جدول درج نماید ولی در خلال طول عمر بسته، بسته جدیدی دریافت نکند آن راه‌یاب از جدول راه‌یابی حذف خواهد شد. با این روش این مشکل نیز حل می‌شود.

وقتی یک راه‌یاب بسته‌های حالت لینک را از تمام راه‌یاب‌های شبکه دریافت کرد، می‌تواند ساختمان‌داده گراف زیر شبکه را تشکیل داده و بر اساس آن اقدام به پیدا کردن بهترین راه‌ها (یعنی با کمترین هزینه) بین هر دو گره بنماید. برای یافتن بهترین راه بین دو گره در یک گراف، می‌توان از الگوریتم دایجسترا[1] بهره برد.

حجم جدولی که هر راه‌یاب در روش بردارفاصله در حافظه خود نگه می‌دارد از درجه یک است یعنی به ازای n راه‌یاب فقط n رکورد کافی است در حالی که برای نگه‌داری جدول‌ها راه‌یابی در روش حالت لینک درجه دو است یعنی در بدترین حالت به یک ماتریس $n \times n$ رکورد نیاز می‌باشد[۱].

۱-۴ راه‌یابی در اینترنت

شبکه اینترنت از تعداد بسیار زیادی سیستم‌های خودگردان[2] تشکیل شده است. هر سیستم خود گردان که برای سادگی AS نیز نامیده می‌شود توسط سازمان و یا نهاد خاصی پیاده و اداره می‌شود و طبیعی است که آن سازمان برای راه‌یابی بسته‌ها در درون شبکه، از الگوریتم مورد نظر خود بهره بگیرد. از دیدگاه اینترنت، جزئیات درونی یک شبکه AS قابل رؤیت نیست. با وجود آنکه جزئیات درونی هر AS مستقل از دیگری است ولی داشتن یک استاندارد برای راه‌یابی در درون یک AS، پیاده‌سازی مرز AS ها را آسان می‌کند. راه‌یابی در درون یک AS اصطلاحا پروتکل دروازه درونی[3] و الگوریتم راه‌یابی بین ASها پروتکل دروازه بیرونی[4] نامیده می‌شود[۳].

راه‌یاب‌هایی که در درون شبکه AS نصب شده‌اند و نقش برقراری ارتباط داخلی را بر عهده دارند به نام دروازه‌های درونی مشهورند و راهیاب‌هایی که ارتباط دو شبکه خودگردان متفاوت را برقرار می‌کنند و ارتباط‌های بین شبکه‌ای از طریق آنها انجام می‌شود به نام راه‌یاب‌های مرزی یا دروازه‌های مرزی شناخته می‌شوند.

۱-۴-۱ پروتکل RIP جهت راه‌یابی دروازه‌های درونی

اولین پروتکل دروازه درونی اینترنت، یک پروتکل بردار فاصله با نام RIP[5] بود. اگر چه این پروتکل در سیستم‌های کوچک به خوبی کار می‌کند ولی با بزرگ شدن ASها، کارایی خود را از دست می‌دهد. این پروتکل همچنین از

---

[1] Dijkstra
[2] Autonomous System
[3] Interior Gateway Protocol
[4] Exterior Gateway Protocol
[5] Routing Information Protocol

مشکل شمارش تا بی‌نهایت رنج می‌برد و نسبت به خرابی‌ها همگرایی کندی دارد. این مشکل از آنجا ناشی می‌گردد که در صورت خراب شدن یک راه‌یاب، همه راه‌یاب‌ها هزینه جدید را بر اساس مقادیری محاسبه می‌کنند که قدیمی و اشتباه است البته به تدریج ارزش فاصله درج شده در جدول تمام راه‌یاب‌ها به سمت بی‌نهایت رشد می‌کند ولی تعداد داد و ستدهای جدول‌ها به عددی که برای مقدار بی‌نهایت در نظر گرفته شده بستگی دارد. اصل این مشکل از آنجا ناشی می‌گردد که وقتی راه‌یاب الف به راه‌یاب ب می‌گوید که راهی به جایی دارد، ب به هیچ وجه نمی‌تواند بفهمد آیا خودش هم بر روی این راه قرار گرفته است یا خیر.

پروتکل RIP ذاتا مبتنی بر الگوریتم بردار فاصله است؛ با این مشخصه که معیار هزینه در این پروتکل تعداد گام می‌باشد.

در پروتکل RIP جدول‌های راه‌یابی هر ۳۰ ثانیه یکبار بین راه‌یاب‌های همسایه مبادله می‌شوند و اگر راه‌یابی این جدول‌ها را از یک همسایه به مدت ۱۸۰ ثانیه دریافت نکند، متوجه یک خرابی در آن همسایه شده و ضمن درج ∞ در جدول خود، آن را به تمام راه‌یاب‌های همسایه خود اعلان می‌کند. RIP از پورت شماره ۵۲۰ و پروتکل UDP[1] برای داد و ستد جدول‌ها استفاده می‌نماید.[۴]

پرؤتکل RIP دقیقا مشابه الگوریتمی کار می‌کند که در روش بردار فاصله به آن پرداختیم.

۱-۴-۲ پروتکل OSPF جهت راه‌یابی دروازه‌های درونی

پروتکل OSPF[2] نیز شبیه RIP برای راه‌یابی دروازه‌های درونی مورد استفاده قرار می‌گیرد. این پروتکل یک پروتکل حالت لینک است و برای ارسال اطلاعات از روش ارسال سیل‌آسا بهره می‌برد. همچنین این پروتکل از الگوریتم دایجسترا جهت یافتن کم‌هزینه‌ترین راه استفاده می‌کند. در این پروتکل، هر راه‌یاب نقشه کاملی از توپولوژی درونی سیستم خود گردان را تشکیل می‌دهد. سپس راه‌یاب الگوریتم کوتاه‌ترین راه دایجسترا را به صورت محلی اجرا می‌کند تا درخت کوتاه‌ترین راه به تمامی زیر شبکه‌ها با ریشه خود را تشکیل دهد[۲].

مشخصه‌های پروتکل OSPF در مقایسه با پروتکل RIP را می‌توان اینگونه برشمرد[۱]:

---

[1] User Datagram Protocol
[2] Open Shortest Path First

- در این پروتکل بر خلاف پروتکل **RIP** از روش الگوریتم‌های حالت لینک برای محاسبه بهترین راه استفاده می‌شود و بنابراین مشکل شمارش تا بی‌نهایت وجود ندارد.
- در این پروتکل معیار هزینه فقط تعداد گام نیست بلکه می‌توان چندین معیار هزینه را در گزینش بهترین راه در نظر بگیرد.
- در این پروتکل حجم بار و ترافیک یک راه‌یاب در محاسبه بهترین راه دخالت داده می‌شود و همچنین در هنگام خرابی یک راه‌یاب جدول‌های راه‌یابی با سرعت مناسبی همگرا می‌شوند.
- در این پروتکل فیلد نوع سرویس در بسته IP، می‌تواند در نظر گرفته شود و بر اساس آن، برای یک بسته راه مناسب گزینش گردد.
- در این پروتکل تمام بسته‌های ارسالی برای یک مقصد خاص، روی بهترین راه هدایت نمی‌شوند بلکه درصدی از بسته‌ها روی راه‌هایی که از لحاظ کم هزینه‌ترین، در رتبه ۲ ، ۳ و... قرار دارند فرستاده می‌شوند. به این کار موازنه بار[1] گفته می‌شود.
- در این پروتکل راه‌یاب‌ها جدول‌های راه‌یابی را از دیگر راه‌یاب‌ها قبول نمی‌کنند مگر آنکه هویت ارسال کننده آن آشکار گردد.

بسیاری از ASها در اینترنت، خودشان بسیار عظیم هستند و مدیریت آنها ساده نیست.OSPF این امکان رافراهم آورده که بتوان چنین شبکه‌هایی را به تعدادی ناحیه‌ی شماره گذاری شده[2] تقسیم کرد. هر ناحیه خود یک شبکه یا مجموعه‌ای از شبکه‌های بهم پیوسته همسایه است.

در هر شبکه خود گردان، ناحیه‌ای به نام ستون فقرات وجود دارد که ناحیه صفر نامیده می‌شود. تمام نواحی به ستون فقرات متصل می‌شوند لذا به راحتی می‌توان به کمک ستون فقرات از هر ناحیه در شبکه AS به ناحیه دیگر رفت.

---

[1] Load Balancing
[2] Numbered Area

درون یک ناحیه، هر یک از راه‌یاب‌ها نسخه مشابهی از پایگاه اطلاعاتی در خصوص راه‌ها و هزینه‌ها در اختیار دارند و الگوریتم محاسبه کوتاه‌ترین راه آن‌ها یکسان است. همچنین هر راه‌یاب باید کوتاه‌ترین راه‌ها از خودش به تمام راه‌یاب‌های دیگر ناحیه را محاسبه نماید. از جمله، هر راه‌یاب باید کوتاه‌ترین راه از خود تا یک راه‌یاب واقع بر روی ستون فقرات را پیدا کند. یک راه‌یاب که در مرز دو ناحیه واقع است باید پایگاه اطلاعاتی هر دو ناحیه را در اختیار داشته باشد و الگوریتم کوتاه‌ترین راه را جداگانه بر روی آن‌ها اجرا کند.

جمع‌بندی مطالب فوق را می‌توان این‌گونه بیان کرد: به کمک روش ارسال سیل‌آسا، هر راه‌یاب به تمام اعضای ناحیه‌خود، از همسایه‌هایش و هزینه رسیدن به آن‌ها خبر می‌دهد. این اطلاعات امکان آن را فراهم می‌آورد تا یکایک راه‌یاب‌ها بتوانند گراف ناحیه خود را تشکیل داده و کوتاه‌ترین راه‌ها را محاسبه نمایند. ستون فقرات نیز همین کار را می‌کند. علاوه بر این، راه‌یاب‌های واقع بر روی ستون فقرات، اطلاعات ارسالی از راه‌یاب‌های مرزی هر ناحیه را هم می‌پذیرند تا بتوانند کوتاه‌ترین راه‌ها از ستون فقرات تا دیگر راه‌یاب‌ها را محاسبه نمایند. این اطلاعات دوباره به راه‌یاب‌های مرزی هر ناحیه بازگردانده می‌شود تا آن‌ها نیز در درون ناحیه خودشان اعلام نمایند. به کمک این اطلاعات، یک راه‌یاب که می‌خواهد بسته‌ای را به خارج ناحیه خود ارسال کند، بهترین راه‌یاب مرزی متصل به ستون فقرات را به‌عنوان دروازه خروجی بسته برمی‌گزینند.

۱-۴-۳ پروتکل BGP جهت راه‌یابی دروازه‌های بیرونی

در درون یک AS تک، پروتکل راه‌یابی OSPF گزینه مناسبی است. بین ASها از پروتکل متفاوتی به نام BGP[1] استفاده می‌شود. به دلیل آنکه اهداف پروتکل راه‌یابی درونی با پروتکل راه‌یابی بیرونی فرق می‌کند، به پروتکل متفاوتی برای راه‌یابی بین ASها نیاز است. پروتکل راه‌یابی درونی باید به سریع‌ترین وجه ممکن بسته‌ها را از مبدا به مقصد برساند و اهمیتی به سیاست‌های جانبی ندارد. پروتکل‌های راه‌یابی بیرونی بایستی در حد گسترده‌ای ملاحظات سیاسی را مد نظر قرار دهند.

از دیدگاه یک راه‌یاب BGP، کل جهان از چندین AS و لینک‌های ارتباطی بین آن‌ها تشکیل شده است. دو AS وقتی متصل در نظر گرفته می‌شوند که بین راه‌یاب‌های مرزی هر یک از آنها حداقل یک لینک وجود داشته باشد. با توجه به تاکید BGP بر حمل ترافیک، هر شبکه AS در یکی از سه رده زیر قرار می‌گیرد:

---

[1] Border Gateway Protocol