

## مقدمه الگوریتمهای مسیریابی

در هریک از سه قرن گذشته فناوری خاصی رونق داشته باشد قرن هجدهم زمان توسعه سیستم های مکانیکی بزرگ به همراه انقلاب صنعتی بود. قرن نوزدهم عصر موتور بخار بود. قرن بیستم زمان جمع آوری، پردازش، و توزیع اطلاعات بود در بین سایر پیشرفت ها، شاهد نصب شبکه های جهانی تلفن، اختراع رادیو و تلویزیون، تولید و رشد بی سابقه صنعت کامپیوتر و پرتاب ماهواره های ارتباطی بوده ایم.

با پیشرفت فناوری این موارد در حال همگرایی است و تفاوت هایی بین جمع آوری، انتقال ذخیره و پردازش اطلاعات به شدت در حال محو شدن است سازمان هایی با صدها شعبه در نقاط مختلف جغرافیایی، ب فشردن کلید وضعیت فعلی را حتی در دورترین نقاط بررسی می کنند. با افزایش قدرت جمع آوری، پردازش و توزیع اطلاعات، تقاضای پردازش اطلاعات پیچیده تر نیز افزایش می یابد

-

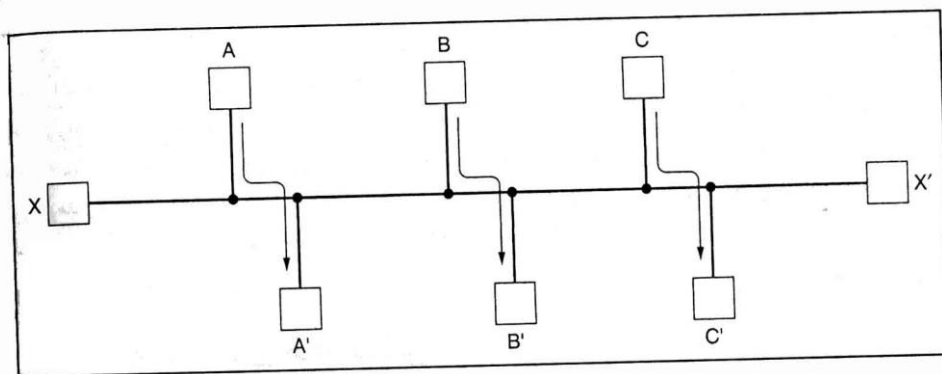
## الگوریتمهای مسیر یابی

وظیفه اصلی لایه شبکه ، هدایت بسته‌ها از ماشین منبع به ماشین مقصد است در اغلب زیر شبکه‌ها ، بسته‌ها باید چند جهش انجام دهند. تا به مقصد برسند. برای شبکه‌های پخش، استثنایی وجود دارد، وای در اینجا نیز اگر منبع و مقصد در یک شبکه نباشد مسیر یابی مشکل محسوب می‌شود. الگوریتم‌هایی که مسیرها و ساختمان داده‌های مربوط به آن را انتخاب می‌کنند، موضوع مهم را طراحی لایه شبکه اند.

الگوریتم مسیر یابی بخشی از نرم افزار لایه شبکه است که تعیین می‌کند بسته ورودی باید به کدام خط خروجی منتقل شود. اگر زیر شبکه از داده‌ها گرام‌ها استفاده کند، این تصمیم‌گیری دوباره باید برای هر بسته ورودی تکرار شود ،چون تا آن موقع امکان دارد بهترین مسیر، تغییر کند اگر زیر شبکه از مدارهای مجازی استفاده کند ، تصمیمات مسیر یابی وقتی اتخاذ می‌شوند که مدار مجازی جدیدی استفاده گردد. از آن پس ، بسته‌های داده‌ها فقط از مسیر ایجاد شده قبلی منتقل می‌شوند.حالت دوم گاهی مسیر یابی تماس دارد ، زیرا مسیر در طول مدت

تمسا کاربرد باقی می ماند ( مثل کار کردن با پایانه یا انتقال فایل ) صرف نظر از این که آیا مسیرها برای هر بسته به طور مستقل انتخاب میشوند یا فقط وقتی که اتصال جدیدی برقرار می شود انتخاب می گردند، خواصی وجود دارند. که در الگوریتم های مسیر یابی مطلوب اند صحت ، سهولت تحمل عیب، پایداری ، عدالت و بهینگی سخت و سهولت نیازی به توضیح ندارند، اما نیاز به تحمل عیب چندان روشن نیست. انتظار می رود که شبکه های بزرگ ، سال ها بدون عیب کلی سیستم به کار خود ادامه دهند. در این مدت ممکن است اشکالات سخت افزاری و نرم افزاری گوناگونی به وجود آید. میزبان ها مسیر یاب ها مسیر یاب ها بدون نیاز به توقف انجام کارها در مسیر یاب ها و راه اندازی مجدد شبکه در هر بار متلاشی شدن مسیریاباز عهده تغییرات در توپولوژی و ترافیک برآید.

پایداری نیز برای الگوریتم مسیر یابی هدف مهمی است. الگوریتم های مسیر یابی وجود دارند که هرگز وجود دارند که هرگز به حالت پایداری نمی رسند. مدت زمان اجرای آن بی تاثیر است عدالت و بهینگی ممکن است ساده به نظر می رسند یقینا کسی با آن مخالف نیست. اما همان طور که روشن است اهداف متناقضی دارند به عنوان مثال از این تناقض ، شکل ۱ را ببینید. فرض کنید ترافیک کافی بین



تضاد بین عدالت و بهینگی.

قبل از اینکه به متوازن کردن عدالت و بهینگی پردازیم . باید تصمیم بگیریم که چه چیزی را بهینه کنیم . بدیهی است تاخیر بسته باید کمینه شود ولی توان شبکه باید بیشینه شود. علاوه بر این این دو هدف نیز با هم تضاد دارند، زیرا عملکرد هر سیستم صف بندی در حد ظرفیت تاخیر صف بندی را زیاد می کند. اغلب شبکه ها سعی میکنند تعداد جهشهای بسته های را کمینه نمایند زیرا کاهش تعداد جهش موجب بهبود تاخیر و نیز کاهش میزان پهنای باند مصرفی است که منجر به بهبود توان عملیاتی می شود.

الگوریتم های مسیر یابی به می توانند به دو دسته تقسیم شوند غیر وفقی و وفقی الگوریتم های غیر وفقی تصمیمات مسیر یابی خود را بر اندازه گیری یا تخمین توپولوژی و ترافیک فعلی بنا نمی نهند بلکه برای انتخاب مسری جهت

رسیدن از I به J برای تمام I را به تمام J از قبل محاسبه می‌شود در حالت OFF-LINE و هنگام راه اندازی شبکه به مسیر یابها بار می‌شود این روند گاهی مسیر یابی ایستا نام دارد.

برعکس الگوریتم‌های وقفی تصمیمات مسیر یابی خود را براساس تغییرات توپولوژی و ترافیک تغییر می‌دهند الگوریتم‌های وقفی ، وقتی که مسیرها را عوض می‌کنند. مثلاً هر ثانیه وقتی بار تغییر می‌کند، با وقتی توپولوژی تغییر می‌کند از نظر جایی که اطلاعات را می‌گیرند مثلاً محلی از مسیریابهمجوار یا تمام مسیریابومعیارهایی که برای بهینه سازی مورد استفاده قرار می‌گیرند. (مثلاً ، محلی از مسیریاب همجواریا تمام مسیر یابها و معیارهایی که برای بهینه سازی مورد استفاده قرار می‌گیرند (مثلاً فاصله ، تعداد جهشها یا زمان انتقال تقریبی با یکدیگر متفاوت‌اند . در بخش‌های بعدی الگوریتم‌های الگوریتم‌های گوناگونی را چه ایستا و چه پویا ،مورد بررسی قرار می‌دهیم.

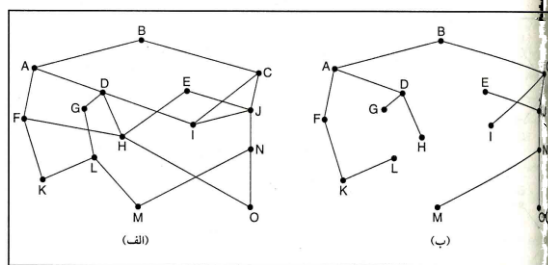
#### اصل بهینگی

قبل از پرداختن به الگوریتم توجه به مهم است که صرف نظر از توپولوژی شبکه وتر افیکی ، می‌توان حکمی کلی راجع به مسیرهای بهینه ارائه کرد این حکم را به

عنوان اصل بهینگی شناخته می‌شود. این اصل بیا می‌کند که اگر مسیریاب J از مسیریاب I به مسیریاب K در مسیریاب بهینه‌ای شناخته می‌کند آنگاه مسر بهینه‌ای از J و K نیز در مسیر مشابهی قرار می‌گیرد. برای مشاهده این موضوع

ر مسیری بهتر از  
ود بخشد، و حکم

لایه شبکه ۳۲۹



(الف) زیر شبکه. (ب) درخت بایگانی برای مسیریاب B.

بخشی از مسیر  
وجود داشت می‌ت  
ما را می‌گوید؟

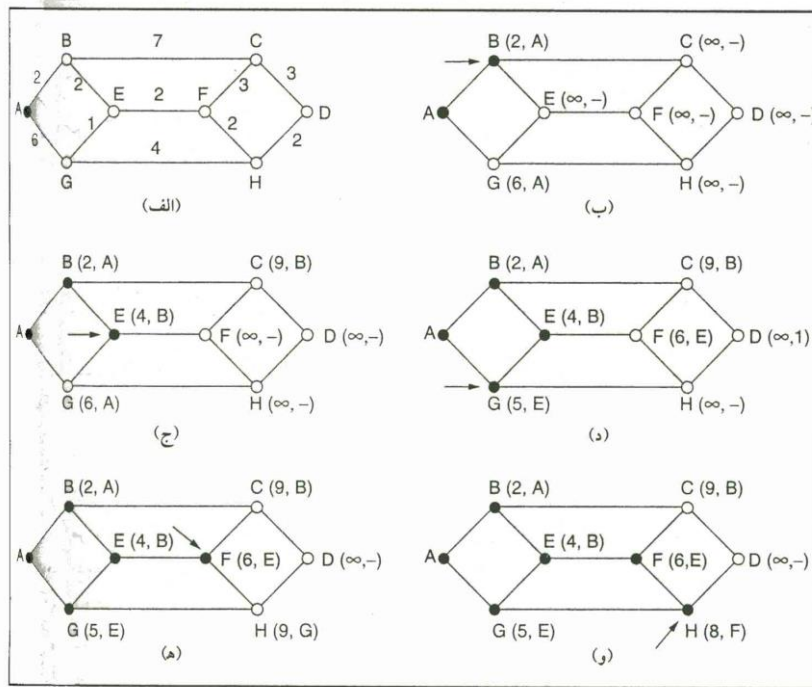
از اصل بهینگی می‌توان نتیجه گرفت که مجموعه‌ای از مسیرهای بهینه از تمام منابع به مقصدی معین، درختی را تشکیل می‌دهد که ریشه اش مقصد است چنین درختی، درخت بایگانی نام دارد. شکل ۲ در این درخت مقیاس فاصله تعداد جهش‌ها است توجه داشته باشید. که درخت‌های دیگری با همان طول مسیر وجود داشته باشند هدف الگوریتم‌های مسیر یابی، یافتن درخت‌های بایگانی و استفاده از آنها برای تمام مسیر یاب‌ها است.

چون درخت بایگانی یک درخت است، فاقد هرگونه حلقه است. لذا هر بسته در تعداد مشخصی از جهش‌های دریافت می‌شود. در عمل همیشه به این سادگی

نیست. در اثنای کار، پیوندهای و مسیر یابی می‌توانند به طرف پایین بروند و به طرف بالا برگردند. بنابراین امکان دارد مسیر یاب‌های مختلف راجع به توپولوژی فعلی ایده‌های متفاوتی داشته باشند. همچنین سوال دیگری که مطرح بود این بود که آیا هر مسیر یاب مجبور است به طور انفرادی اطلاعات مورد نیاز جهت محاسبه درخت بایگانی را به دست آورد یا این اطلاعات توسط وسایل دیگری جمع آوری می‌شوند در ادامه به طور مختصر به این موضوع می‌پردازیم با این وجود، اصل بهینگی و درخت بایگانی‌های معیارهایی را تهیه کردند که سایر الگوریتم‌های مسیر یابی می‌توانند بر اساس آنها ارزیابی شوند.

#### مسیر یابی کوتاه ترین مسیر

مطالعه الگوریتم‌های مسیر یابی را با تکنیکی که به طور گسترده به شکل‌های مختلفی به کار می‌رود شروع می‌کنیم، زیرا الگوریتم ساده‌ای است و درک آن آسان است. ایده، ساختن گرافی از زیر شبکه است، به طوری که، هر گره گراف نشان دهنده مسیر یاب است و هر یال نشان دهنده خط ارتباطی است (که اغلب پیوند نام دارد). برای انتخاب مسیری بین دو مسیر یاب معین، الگوریتم، کوتاهترین مسیر بین آنها را در گراف می‌یابد.



در مورد

مسیر ،

یکسان است

حالت بد

در مورد

مسیر ،

یکسان است

حالت بد

با مقیاس

پنج مرحله اول برای محاسبه کوتاهترین مسیر از A به D می باشد، پیکان ها گره کاری را نشان می دهند.

علاوه بر جهشها و فاصله فیزیکی معیارهای دیگری نیز قابل استفاده اند به عنوان مثال هریال می تواند به میانگین تاخیر صف بندی و انتقال برای بعضی از بسته های آزمایشی برچسب گذاری شود. با این برچسب گذاری، کوتاهترین مسیر به جای مسیری به جای مسیری که با کمترین یال یا فاصله سریع تر مسیر است.

در حالت کلی، برچسب های یالها باید به صورت تابعی از فاصله، پهنای باند، میانگین ترافیک هزینه ارتباط میانگین طول صف تاخیر اندازه گیری شده و سایر عوامل محاسبه شود. با تغییر تابع وزنی، الگوریتم، کوتاهترین مسیر وزن دار را براساس هریک از معیارهای فوق یا ترکیبی از آنها محاسبه می کند.



الگوریتم‌های متعددی برای محاسبه کوتاهترین مسیریابی در گره‌گراف شناسایی شده‌اند یکی از این الگوریتم‌های به دیکسترا ۱۹۹۵ نسبت داده می‌شود. هر گره دارای برچسب‌هایی در پیرامون است که فاصله آن تا گره منبع، از طریق بهترین مسیر شناخته شده نیست لذا تمام گره‌ها دارای برچسب بی نهایت هستند. با ادامه اجرای الگوریتم و پیدا شدن مسیرها، امکان دارد برچسب‌ها تغییر کنند تا مسیرهای بهتری منعکس نمایند. برچسب ممکن است موقتی یا دائمی باشد. در آغاز، تمام برچسب‌ها موقتی‌اند وقتی مشخص شد که برچسبی کوتاهترین مسیر بین منبع به آن گره تمام برچسب‌ها موقتی اندوقتی مشخص شد که برچسبی کوتاهترین مسیر بین منبع به آن گره را نمایش می‌دهد، دائمی می‌شود و از آن پس تغییر نمی‌کند.

برای اینکه که مشخص شود الگوریتم برچسب‌گذاری چگونه کار می‌کند. گراف وزن دار بدون جهت شکل ۳ الف را در نظر بگیرید. که وزن‌ها، مثلاً فاصله را نشان می‌دهد می‌خواهیم کوتاهترین مسیر از A به D را بیابیم. با علامت‌گذاری گره A به عنوان گره ثابت که به صورت دایره پر نشان شده است. شروع می‌کنیم. سپس نوبت، تمام همجوار A همجوار A گره کاری را تست می‌کنیم. هر کدام را با

فاصله آن به A مجدداً برچسب می‌دهیم. هر وقت گره‌ای مجدداً برچسب دهی شد، آن را با گره اس که کار از آنجا آغاز شد برچسب می‌دهیم به این ترتیب می‌توانیم مسیر نهایی را بازسازی کنیم. با بررسی تمام گره‌ها همجوار A تمام گره‌هایی را که در کل گراف به طور موقت برچسب دهی شدند بررسی می‌کنیم و گره‌ای که دارای کوچک‌ترین برچسب است دائمی می‌کنیم. (شکل ۳-ب) این گروه به عنوان گره کاری جدید انتخاب می‌شود.

اکنون از B شروع می‌کنیم و تمام گره‌هایی همجوار آن را مورد بررسی قرار می‌دهیم. اگر مجموع برچسب در B و فاصله B تا گره‌ای که باید در نظر گرفته شود کمتر از برچسب موجود در آن گره باشد کوتاهترین مسیر پیدا شده، این گره مجدداً برچسب گذاری می‌شود.

پس از این تمام گره‌ها همجوار گره کاری بررسی شدند و گره‌های موقتی تغییر کردند، کل گراف مورد جست‌وجو قرار می‌گیرد تا گره‌ای موقتی با کمترین مقدار برچسب گذاری می‌شود

برای پی بردن به عملکرد الگوریتم شکل ۳ ج را ببینید در این شکل، E دائمی است فرض کنید مسیر AXYZA کوتاهتر از ABE باشد دو امکان وجود دارد: یا گره

Z به عنوان گره دائمی منظور شده است یا نشده است اگر دائمی باشد E تاکنون بررسی شده است در سیکلی بعد از آن که Z دائمی شد. لذا  $AXYZE$  از دید ما خارج نبوده است و نمی‌تواند مسیر کوتاهتری باشد

اکنون حالتی را در نظر بگیرید که هنوز بر چسب Z موقتی باشد. برچسب موجود در Z بزرگتر یا مساوی برچسب در E است که در این حالت  $XYZE$  نسبت به  $ABC$  مسیر کوتاهتری نیست، یا کمتر از E است که در این حالت Z و E تاکنون بررسی مورد جستجو قرار می‌گیرد.

این الگوریتم در شکل ۴ آمده است متغیرهایی عمومی N و DIST گراف را توصیف می‌کنند و قبل از فراخوانی **SHORTEST PATH** مقدار می‌گیرند. تنها بین برنامه والگوریتمی که تشریح شد این است که کوتاهترین مانند کوتاهترین مسیر از S به T محاسبه شده است. چون کوتاهترین مسیر از T به S در گراف بدون جهت است مهم نیست که از کدام طرف شروع کنیم مگر اینکه کوتاهترین مسیر متعددی وجود داشته باشد که در آن حالت جست و جستجوی معکوس مسیر دیگری را انتخاب می‌نماید. دلیل جستجوی معکوس این است که هرگره با گره قبلی خود (به جای گره بعدی) برچسب گذاری می‌شود. هنگام کپی کردن

```

#define MAX_NODES 1024          /* maximum number of nodes */
#define INFINITY 1000000000    /* a number larger than every maximum path */
int n, dist[MAX_NODES][MAX_NODES]; /* dist[i][j] is the distance from i to j */

void shortest_path(int s, int t, int path[])
{ struct state {                /* the path being worked on */
  int predecessor;             /* previous node */
  int length;                  /* length from source to this node */
  enum {permanent, tentative} label; /* label state */
} state[MAX_NODES];

int i, k, min;
struct state *p;

for (p = &state[0]; p < &state[n]; p++) { /* initialize state */
  p->predecessor = -1;
  p->length = INFINITY;
  p->label = tentative;
}
state[t].length = 0; state[t].label = permanent;
k = t; /* k is the initial working node */
do { /* Is there a better path from k? */
  for (i = 0; i < n; i++) /* this graph has n nodes */
    if (dist[k][i] != 0 && state[i].label == tentative) {
      if (state[k].length + dist[k][i] < state[i].length) {
        state[i].predecessor = k;
        state[i].length = state[k].length + dist[k][i];
      }
    }

  /* Find the tentatively labeled node with the smallest label. */
  k = 0; min = INFINITY;
  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);

/* Copy the path into the output array. */
i = 0; k = s;
do {path[i++] = k; k = state[k].predecessor; } while (k >= 0);
}

```

الگوریتم دیکسترا برای محاسبه کوتاه‌ترین مسیر در گراف.

تکنیک دیگر برای محدود کردن الگوریتم غرق کردن این است که بسته هایی که

تاکنون ارسال شده‌اند مشخص باشند، تا مجددا ارسال نگردند یک روش انجام این

کار این است که مسیریابمنبع ، در بسته هایی که از میزبانهایش دریافت می کند شماره ترتیبی را قرار دهد در این صورت هر مسیریاببه ازای هر مسیریابمنبع به لیستی نیاز دارد تا مشخص کند کدام شماره ترتیب هایی که تاکنون از منبع ارسال شدند دریافت گردیدند. اگر بسته ورودی در آن لیست موجود باشد: ارسال نشده است.

برای جلوگیری از رشد بی رویه لیست، هر لیست باید دارای شمارنده ای به نام  $K$  باشد، معنایش این است که تمام شماره ترتیبها از ۱ تا  $K$  مشاهده شده اند وقتی بسته ای دریافت می شود، به راحتی می توان تشخیص داد که این آیا تکراری است یا خیر اگر تکراری باشد، از آن صرف نظر می گردد. علاوه بر این، به لیست کامل کمتر از  $K$  نیازی نیست، زیرا  $K$  آن را خلاصه می کند.

شکل خاصی از الگوریتم غرق کردن که عملی تر است غرق کردن انتخابی نام دارد. در این الگوریتم، مسیر یابها هر بسته ورودی را به تمام خطوط خروجی نمی فرستند ، فقط به خط هایی می فرستند که تقریباً درجهت درستی منتقل می شوند کمتر اتفاق می افتد بسته ای که می خواهد به غرب برود به خطی در

قسمت شرق ارسال شود، مگر این که توپولوژی ویژه‌ای به کار گرفته شود  
و مسیریاب به این حقیقت مطمئن باشد.

الگوریتم غرق کردن، در اغلب کاربردها عملی نیست، اما کاربردهایی دارد به  
عنوان مثال در کاربردهای نظامی، که لازم است در هر لحظه بیت هایی برای  
بسیاری از مسیر یابها ارسال شود، الگوریتم غرق کردن توانمند نوسازی شوند  
سومین کاربرد غرق کردن همواره کوتاهترین مسیر را انتخاب می‌کند، زیرا تمام  
مسیرهای ممکن را به طور موازی آزمایش می‌کند در نتیجه هیچ الگوریتم دیگری  
نمی‌تواند تاخیر کمتری ایجاد نماید. اگر سربار حاصل از خود فرایند غرق کردن را  
نادیده بگیریم.

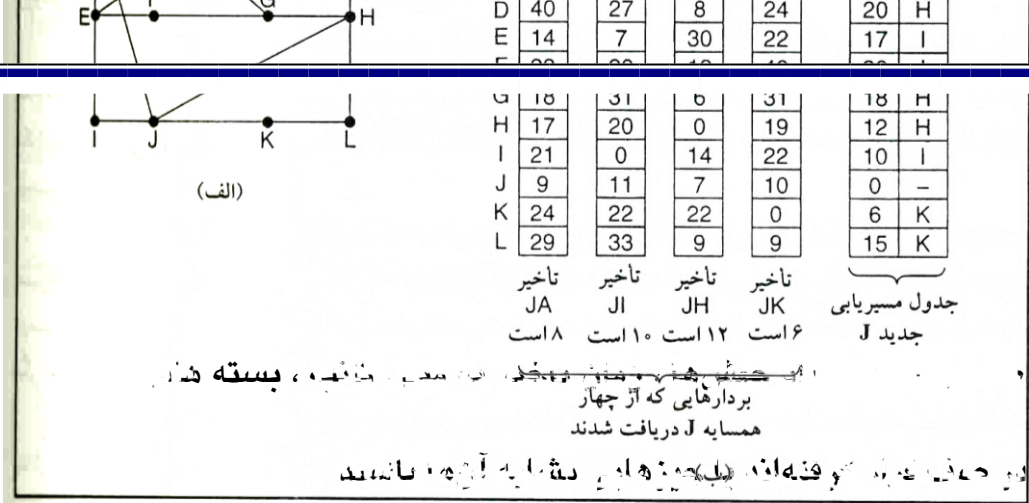
#### مسیر یابی بردار فاصله

شبکه هایی کامپیوتری مدرن به جای الگوریتمهای مسیر یابی ایستا از الگوریتم  
مسیریابی پویا استفاده می‌کنند، زیرا الگوریتمهای ایستا بار فعلی شبکه را در  
نظر نمی‌گیرند و دو الگوریتم پویا به نامهای مسیر یابی بردار فاصله و مسیر

یابی حالت پیوند، عمومیت بیشتری دارند در این بخش به الگوریتم مسیر یابی بردار فاصله و در بخش بعدی به الگوریتم مسیر یابی حالت پیوند می پردازیم. در الگوریتمهای مسیریابی بردار فاصله هر مسیریابجدول یا برداری دارد که بهترین فاصله به هر مقصد را نگهداری می کند خطی را که برای رسیدن به آن مقصد لازم است مشخص می کند. این جدولها از طریق تبادل اطلاعات با همسایهها بازسازی می شوند.

الگوریتم مسیر یابی بردار فاصله به اسامی دیگر نیز خوانده می شود. از جمله الگوریتم مسیر یابی بلمن -فورد و الگوریتم و الگوریتم فورد - فورکرسون که نامگذاری آنها را نام مخترعین آنها بلمن ۱۹۷۵- فورد و فورکرسون، ۱۹۶۲ اقتباس شده است. این الگوریتم مسیر یابی ARPANET اولیه بود و تحت نام RIP در اینترنت مورد استفاده قرار گرفت.

در مسیر یابی بردار فاصله ، هر مسیر باب دارای جدول است که به ازای هر مسیر در زیر شبکه یک وارده دارد این وارده دو بخش است : خط خروجی پیشنهادی برای استفاده از آن مقصد و تخمینی از زمان یا فاصله به آن مقصد مقیاس مورد



شکل ۹-۵ (الف) زیر شبکه، (ب) ورودی از  $K, H, I, A$  و جدول مسیریابی جدیدی برای  $J$ .

فرض می شود که مسیریاب فاصله خود تا هر همسایه اش را می داند و اگر مقیاس ، جهش باشد، فاصله فقط یک جهش است اگر مقیاس طول صف باشد مسیر باب هر صف را بررسی می کند اگر مقیاس تاخیر باشد، مسیر باب می تواند آنرا مستقیماً با بسته ECHO خاصی از هر طرف گیرنده ارسال می شود اندازه گیری کند.

به عنوان مثال ، فرض کنید تاخیر به عنوان مقیاس به کار می رود و مسیریاب، تاخیر سایه خودش را می داند . هر مسیریاب در هر  $T$  میلی ثانیه لیستی از تاخیرهای تخمینی خود را به هر مقصد را ارسال می کند و لیست مشابهی از هر همسایه خود دریافت می کند فرض کنید یکی از این جدولها از همسایه ها  $X$  می رسد، به طوری که  $X$  زمان رسیدن به مسیریاب  $I$  باشد که  $X$  آن را تخمین زده است اگر مسیریاب بداند تاخیر تا  $X$  برابر با  $M$  میلی ثانیه باشد، می داند که اگر بخواهد از طریق  $X$  به مسیریاب  $I$  برسد  $X+M$  میلی ثانیه طول می کشد. با انجام این محاسبات برای هر همسایه های مسیریابی می تواند بهترین تخمین را تشخیص



دهد و می‌تواند از این تخمین و خط متناظر در جدول مسیر یابی جدید استفاده نماید توجه داشته باشید و که جدول مسیر یابی قبلی، در محاسبه به کار نمی‌آید. این فرآیند بازسازی در شکل ۵ آمده است بخش الف زیر شبکه‌ای را نشان می‌دهد چهار ستون اول بخش (ب) بردارهایی تاخیری را که از همسایه‌هایی مسیریاب J

برابر	(الف)					(ب)					آمده‌ان
کنید	A	B	C	D	E	A	B	C	D	E	۲۵
	•	•	•	•	•	•	•	•	•	•	شروع
	1	•	•	•	•	1	2	3	4	•	پس از ۱ مبادله
	1	2	•	•	•	3	2	3	4	•	پس از ۲ مبادله
	1	2	3	•	•	5	4	3	4	•	پس از ۳ مبادله
۶۱۲	1	2	3	4	•	5	6	5	6	•	پس از ۴ مبادله
	1	2	3	4	•	7	6	7	6	•	پس از ۵ مبادله
						7	8	7	8	•	پس از ۶ مبادله
						⋮					
						•	•	•	•	•	تاخیر
											میلی ث

شکل ۱۰-۵ مسئله بی‌نهایت‌گرایی.

چگونگی محاسبه مسیر جدید از J به G را در نظر بگیرید J می‌داند که می‌تواند با ۸ میلی ثانیه تاخیر به A برسد و A با ۱۸ میلی ثانیه به G می‌رسد لذا J می‌داند که اگر بخواند از طریق A به G برسد ۲۶ میلی ثانیه طول می‌کشد. به طور مشابه به تاخیر رسیدن به J را در جدول، ۱۸ میلی ثانیه ثبت می‌کند و آن، از طریق H است محاسبه مشابهی برای تمام مقصدها صورت می‌گیرد به طوری که

جدول مسیر یابی جدید را به صورت آخرین به صورت آخرین ستون شکل در می‌آید.

### مسئله بی نهایت گرایی

مسیر یابی بردار فاصله از نظرو تئوری کار می‌کند، اما در عمل مشکل جدی دارد با این که پاسخ صحیح می‌دهد، ولی به کندی عمل میکند به ویژه به خبرهای خوب، واکنش سریع ولی به خبرهای بد واکنش نشان می‌دهد مسیر یابی را در نظر بگیرید که بهترین مسیر آن را به  $X$  بزرگ باشد، ادگر در مبادله بعدی ، همسایه  $A$  ناگهان تاخیر اندکی به  $X$  را گزارش کند، مسیریاباز خطی که به  $A$  می‌آید برای ارسال ترافیک به  $X$  استفاده می‌کند در یک مبادله بردار، اخبار خوب پردازش می‌شوند.

برای مشاهده چگونگی انتشار خبرهای خوب، زیر شبکه پنج گره‌ای خطی شکل ۶ رادر نظر بگیرید، که در آن تعداد جهش‌ها به عنوان مقیاس است فرض کنید  $A$  از

همان اول از کار افتاد و تمام مسیر یاب‌های دیگر این را می‌دانند به عبارت دیگر تمام آنها تاخیرهای رسیدن به  $A$  را بخ صورت بی نهایت ضبط کرده اند وقتی  $A$  را به کار می‌افتد . سایر مسیر یاب‌ها از طریق مبادله بردار ، آگاه مس شوند برای سهولت فرض کنیم زنگ بزرگی وجود دارد که برای شروع همزمان مبادله بردار در تمام مسیر یاب‌ها به صدا در می‌آید در زمان مبادله نخست  $B$  می‌فهمد که همسایه چپ آن تا  $A$  آن را تاخیری ندارد صفر است سپس  $B$  در جدول مسیر یابی خود ثبت می‌کند که  $A$  تا همسایه چپ ، یک جهش فاصله دارد سایر مسیر یاب‌ها فکر می‌کنند که  $A$  هنوز از کار افتاده است در این لحظه وارده هایی جدول مسیر یابی  $A$  در سطر دوم شکل ۶ برابر است الف لذا جدول مسیر یابی را بازسازی می‌کند تا مسیری به طول ۲ را نشان دهد اما  $D$  و  $E$  تاکنون خبرهای جدید را نشنیده اندن بدیهی است که خبرهای جدید با سرعت یک جهش در هر مبادله بخش می‌شود در زیر شبکه‌های که طولانی ترین مسیر که ان به طول  $N$  جهش است. در  $N$  مبادله هرکسی از خطوط از خطوط و مسیریاب هایی که تازه فعال شده‌اند باخبر می‌شود.

اکنون وضعیت شکل ۶ (ب) را در نظر می‌گیریم در این شکل تمام خطوط و مسیریاب‌ها در آغاز فعال‌اند و فاصله مسیر یاب‌های A تا E, D, C, B, به ترتیب عبارتند از ۱ و ۳ و ۲ و ۴ ناگهان A از کار می‌افتد یا خط بین A, B قطع می‌شود از دید B فرقی نمی‌کند که کدامیک اتفاق افتاده است.

در مبادله اولین بسته، B چیزی از A نمی‌شنود خوشبختانه C می‌گوید نگران نباشید من مسیری به طول ۲ به A دارم لذا B می‌داند که مسیر C از طریق خود B می‌داند که C ممکن است ده خط خروجی داشته باشد. که هر کدام دارای مسیرهای مستقلی به A هستند که طول آنها ۲ است در نتیجه B فکر می‌کند که می‌تواند از طریق C به A برسد با مسیرهای به طول ۳ در مبادله اول E, D وارده‌های خود را برای A بازسازی میکنند.

در مبادله دوم C در می‌یابد که هر یک از همسایه‌هایش ادعا میکنند که طول مسیر آنها را به A برابر با ۳ است یکی از آنها به طور تصادفی انتخاب می‌کند و فاصله جدید به A را برابر با ۴ منظور می‌کند سطر سوم از شکل ۶ الف مبادله‌های بعدی نتایج بقیه شکل ۶ الف را تولید میکنند.

از این شکل پیدا است که چرا خبرهای بد کندی ارسال می‌شوند : هیچ مسیر یابی مقداری بیش از کمترین مقدار تمام همسایه هایش را ندارد گاهی تمام مسیر یابها بی نهایت بار کار می‌کنند. به همین دلیل ، عاقلانه است که بی نهایت را برابر با طولانی ترین مسیر به علاوه ۱ قرار داد اگر مقیاس تاخیر زمان باشد و حد بالایی تعریف شده‌ای وجود ندارد لذا برای با طولانی ترین مسیر با تاخیر طولانی مثل مسیر از کار افتاده رفتار نشود وجود ندارد لذا برای اینکه با مسیری با تاخیر طولانی، مثل مسیر از کار افتاده نشود، نیاز به حد بالایی است لذا این مسئله بی نهایت گرایی نام دارد تلاش زیادی برای حل آن انجام شد ، ولی هیچ کدام موفق نبوده اند. مسئله مهم این است که وقتی  $X$  به  $Y$  می‌گوید مسیری در اختیار دارد،  $Y$  نمی‌تواند بفهمد که آیا خودش در آن مسیر قرار دارد یا خیر .

#### مسیر یابی حالت پیوند

مسیر یابی فاصله تا سال ۱۹۷۹ در ARPANET مورد استفاده قرار گرفت و از آن پس جای خود را به مسیر یابی حالت پیوند داد. و مشکل عمده موجب مرگ آن شد. یکی از این که مقیاس تاخیر، طول صف بود و هنگام انتخاب مسیریابها پهنای باند را در نظر نمی‌گرفت در آغاز تمام خطها 56KBPS بودند لذا پهنای

باند موضوع مهمی نبود اما وقتی بعضی از خطوط به 235KBPS و بعضی دیگر به ۱/۵۵ MBPS تغییر یافتند عدم توجه به پهنای باند را به عنوان مقیاس در نظر گرفت اما مشکل دوم نیز وجود داشت، یعنی الگوریتم برای همگرا شدن به زمان زیادی نیاز دارد. بی نهایت گرایی به این دلایل الگوریتم دیگری به نام مسیریابی حالت پیوند جای آن را گرفت اکنون شکلهای گوناگونی از مسیر یابی حالت پیوند مورد استفاده قرار میگیرد.

ایده مسیر یابی حالت پیوند ساده است و در پنج بخش بیان می شود هر مسیریاب باید:

- ۱- همسایه هایش را تشخیص داده و آدرس شبکه ها آنها را بداند.
  - ۲- تاخیر با هزینه تا همسایه هایش را اندازه گیری کند.
  - ۳- ایجاد بسته ای که اطلاعات به دست آمده از همسایه ها را نگهداری کند.
  - ۴- این بسته ها را به تمام مسیریاب ها ارسال نماید.
  - ۵- کوتاهترین مسیر به هر مسیر دیگر را محاسبه کند.
- در نتیجه کل توپولوژی و تمام تاخیرها به طور آزمایشی اندازه گیری می شود و به مسیر یاب های دیگر توزیع می گردد. سپس الگوریتم های دیکسترا

می‌تواند برای یافتن کوتاهترین مسیرها را به مسیر یاب‌ها دیر مورد استفاده قرار گیرد هر یک از پنج مرحله را به تفصیل مورد بررسی قرار می‌دهیم.

کسب اطلاعاتی راجع به همسایه‌ها

وقتی مسیر فعال شد، اولین کارش این است که همسایه‌اش را بشناسد این کار با ارسال بسته HELLO ویژه‌ای به هر خط نقطه به نقطه انجام می‌شود. انتظار می‌رود مسیریاب‌طرف دیگر پاسخی بدهد و خود را معرفی کند این اسامی باید منحصر به فرد باشند زیرا وقتی مسیریاب دور، می‌یابد به F متصل اند باید مشخص کند که آیا منظور هر سه، همان F است یا خیر؟

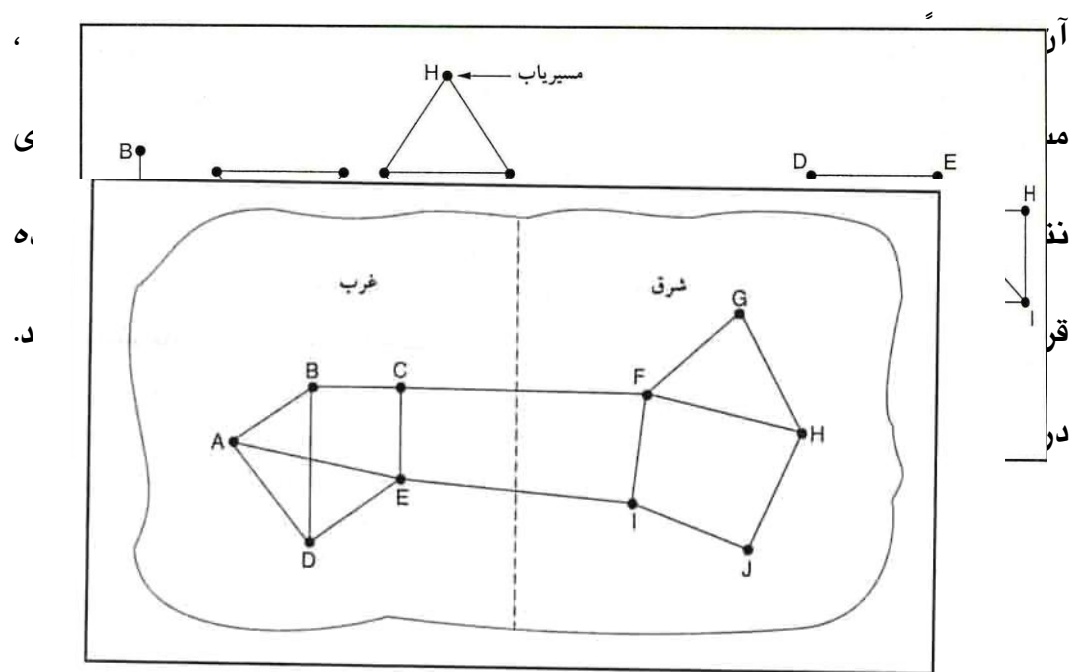
وقتی دو یا چند مسیریاب با شبکه‌های محلی را به هم متصل باشند. وضعیت کمی پیچیده تر است. شکل ۷ الف شبکه محلی را با سه مسیریاب A, C, F نشان می‌دهد که مستقیماً به آن متصل‌اند هر کدام از این مسیریاب‌ها به یک یا چند مسیریاب دیگر متصل شده‌اند.

یک روش مدل سازی شبکه محلی این است که به عنوان یک گروه در نظر گرفته شود شکل (۷ ب) در اینجا گره جدید و مصنوعی به نام N را معرفی می‌کردیم. که

A, C, F به آن متصل اند امکان رفتن از A به C در شبکه محلی ، با مسیر ANC مشخص شده است.

اندازه گیری هزینه خط

در الگوریتم مسیر حالت پیوند لازم است. هر مسیریاب اندازه تاخیر تا همسایه هایش را بداند. و یا حداقل ، اندازه تقریبی آن مشخص باشد مستقیم، ترین راه برای تعیین این تاخیر، ارسال بسته ECHO ویژه‌ای در خط است که طرف دیگر



زیر شبکه‌ای که بخش‌های شرقی و غربی آن با دو خط به هم متصل شده‌اند.



موضوع جالب این است که آیا هنگام اندازه گیری تاخیر، با را باید در نظر گرفت یا خیر برای در نظر گرفتن بار، تایمر رفت و برگشت باید از زمانی که ECHO در صف قرار می گیرد. شروع به کار کند برای صرف نظر از بار، تایمر رفت و برگشت باید از زمانی که ECHO به جلوی صف رسیده باشد.

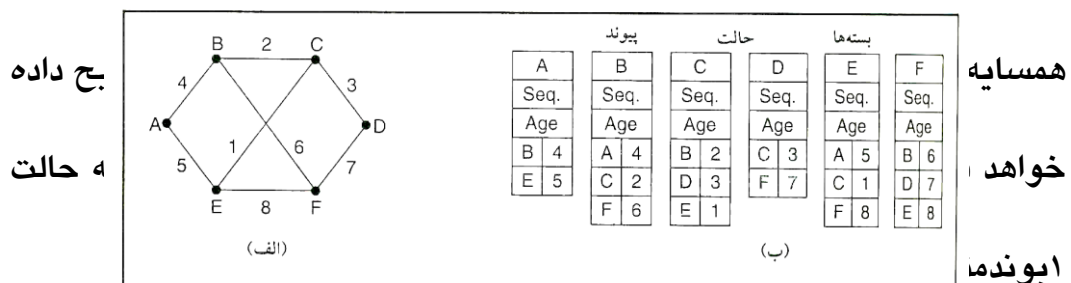
هر دو روش بحث هایی را می طلبد معنای به حساب آوردن تاخیرهای مربوط به ترافیک ، این است که وقتی مسیریاب دو خط با پهنای باند مساوی را در پیش روا داشته باشد، به طوری که یکی از آنها همواره تحت بار سنگین قرار دارد و دیگری این این طور نباشد مسیر مربوط به خط فاقد بار را به عنوان مسیر کوتاهتر در نظر می گیرد. این روش کارایی بهتری دارد متأسفانه با در نظر گرفتن بار در محاسبات تاخیر مخالفت هایی صورت گرفت زیر شبکه شکل ۸ را در نظر بگیرید که به دو بخش شرقی و غربی تقسیم شده است و توسط دو خط EI-, CF- به هم متصل شده اند فرض کنید بیشترین ترافیک بین شرق غرب از خط ترافیک شرقی - غربی از طریق EI منتقل می شود و بار ان افزون می گردد. در نتیجه در بازسازی بعدی، CF کوتاهترین مسیر خواهد بود. لذا امکان دارد جدول های مسیر یابی شدیداً تغییر می کنند و منجر به مسیر یابی غیر عادی و بسیاری از مشکلات

دیگر شوند. اگر از بار صرف نظر شود و فقط پنهان باند منظور گردد، این مشکل نمی‌آید از طرف دیگر بار می‌تواند در هر دو خط پخش شود. اما این راه حل، بهترین مسیر را مورد استفاده قرار نمی‌دهد با این وجود برای اجتناب از برخورد در انتخاب بهترین مسیر، معقول است که بار در چندین خط توزیع شود.

### ساخت بسته‌های حالت پیوند

وقتی اطلاعات مورد نیاز برای مبادله جمع آوری شد قدم بعدی هر مسیریاب این است که بسته‌ای حاوی تمام داده‌ها ایجاد کند. در ابتدای هر بسته، هویت

فرستنده



(الف) زیر شبکه. (ب) بسته‌های حالت پیوند مربوط به زیر شبکه (الف).

ساخت بسته‌های حالت پیوند ساده است. بخش مشکل آن تعیین زمان ساخت آن‌ها است یک راه حل این است که به طور دوره‌ای ساخته شوند. یعنی، در فواصل زمانی ایجاد گردند. روش دیگر این است که وقتی رویدادهای مهمی مثل از

کار افتادن خط یا همسایه و فعال شدن دوباره آنها یا تغییر خواص آن، اتفاق می افتد ایجاد گردد.

توزیع بسته‌های حالت پیوند.

جالب ترین بخش الگوریتم توزیع قابل اعتماد بسته‌های حالت پیوند است وقتی بسته‌ها توزیع شدند و در خط قرار گرفتند مسیر یاب‌ها اولین بسته‌هایی را که دریافت می‌کنند، تغییر می‌دهند. در نتیجه مسیر یاب‌های مختلف ممکن است نسخه‌هایی گوناگونی از توپولوژی را به کار گیرند. و این کار منجر به ناسازگاری حلقه‌های، ماشین‌های غیر قابل دستیابی و سایر مشکلات شوند.

ابتدا، الگوریتم توزیع اولیه را مورد بحث قرار می‌دهیم. سپس اصلاحاتی را انجام دهیم ایده اصلی، استفاده از الگوریتم غرق کردن برای توزیع بسته‌های حالت پیوند است برای کنترل غرق کردن هر بسته حاوی شماره ترتیبی است که با ارسال هر بسته، یک واحد افزایش می‌یابد. وقتی بسته حالت پیوند دیگری دریافت می‌شود، با لیستی از بسته‌ها که تاکنون دیده شده‌اند مقایسه می‌شود اگر بسته‌ای دریافت شود. به هر خطی به جز خطی که از آن آمده است، توزیع می‌گردد. و اگر تکراری باشد، صرف نظر می‌شود اگر بسته‌ای دریافت شود که

شماره ترتیب آن کوچک تر از بالاترین شماره‌ای باشد که تاکنون مشاهده شده است به دلیل کهنه بودن رد می‌شود. زیر مسیریاب‌ها جدیدی دارد. این الگوریتم دارای مشکلات خاصی است اما این مشکلات قابل کنترل‌اند یکی این که اگر شماره‌ها تمام شدند، بسته‌هایی بعدی از اول شماره گذاری شوند راه حل این مشکل، استفاده از شماره ترتیب ۳۲ بی‌تی ۳۲ بی‌تی است اگر در هر دقیقه یک بسته حالت پیوند ایجاد شود. ۱۳۷ سال طول می‌کشد تا چرخش صورت می‌گیرد. لذا از این حالت می‌توان صرف نظر کرد.

دوم اینکه اگر مسیریاب‌ها کار افتد و شماره ترتیب خود را از دست می‌دهد. اگر مجدداً از صفر شروع کند، بسته بعدی به عنوان بسته تکراری رد خواهد شد. سوم اینکه اگر شماره ترتیب خراب شود و ۵۴۰،۶۵ به جای ۴ (خطای یک بی‌تی) دریافت شود، بسته‌های ۵ تا ۵۴۰،۶۵ به علت کهنگی رد می‌شوند زیرا فرض می‌شود که شماره ترتیب باید ۵۴۰،۶۵ باشد.

راه حل این مشکلات این است که سن هر بسته بعد از شماره ترتیب قرار داده می‌شود و هر ثانیه یک واحد از آن کسر گردد. وقتی که سن به صفر رسید. از اطلاعات حاصل از آن مسیریاب صرف نظر می‌شود. فرض کنید در هر ۱۰ دقیقه

بسته جدیدی می‌رسد. لذا مهلت اطلاعات مسیریاب وقتی تمام می‌شود که مسیریاب غیر فعال شود یا شش بسته متوالی از بین رفته باشد، البته این حالت رویداد نامحتملی است هر مسیریاب در فرآیند غرق کردن اولیه، از فیلد سن یک واحد می‌کاهد لذا اطمینان حاصل می‌شود که هیچ بسته‌ای نمی‌تواند از بین برود و یا مدت زمان زیادی زنده بماند (بسته‌ای که سن آن به صفر باشد. نادیده گرفته می‌شود).

اصلاحاتی در این الگوریتم توانمندی آن را زیاد می‌کند وقتی بسته حالت پیوند به مسیریابی‌آید تا ارسال شود فوراً برای انتقال در صف قرار نمی‌گیرد. بلکه به ناحیه نگهدارنده‌ای می‌رود تا مدت کوتاهی را منتظر بماند. اگر قبل از انتقال آن، بسته دیگری از همان منبع برسد، شماره ترتیب آنها مقایسه می‌شود اگر باهم برابر باشند بسته تکراری نادیده گرفته می‌شود اگر مساوی نباشند قدیمی‌تر، نادیده گرفته می‌شود اگر مساوی نباشند. قدیمی‌تر نادیده گرفته نادیده گرفته خواهد شد. برای حفاظت در مقابل خطاها مسیریاب مسیریاب تمام بسته‌های حالت پیوند اعلام وصول می‌شوند وقتی خط آزاد می‌شود ناحیه نگهدارنده به طریق نوبتی پیمایش می‌شود. تا بسته با اعلام وصولی را برای ارسال انتخاب نماید.

ساختمان داده‌ای که مسیریاب B برای زیر شبکه شکل ۱۳-۵ الف استفاده می‌کند در شکل ۱۰ آمده است هر سطر ، متناظر با بسته حالت پیوندی است که از راه رسید و هنوز به طور کامل پردازش نشده است. جایی که بسته از آنجا ارسال شد و شماره ترتیب و سن ، داده‌های آن در جدول ذخیره می‌شود به علاوه نشانگرهای ارسالی و اعلام وصول برای هر سه خط B وجود دارند به ترتیب به A, C, F معنای نشانگرهای ارسالی این است که بسته باید به خط تعیین شده ارسال گردد و معنای نشانگرهای اعلام وصول این است که باید در آنجا اعلام وصول شوند.

در شکل ۱۰ بسته حالت پیوند مستقیماً از A رسیده است. لذا همانطور که با بیت‌های نشانگر نشان داده شده است باید به F, C ارسال شود. و به A اعلام وصول گردد. به طور مشابه بسته‌ای از F باید به A و C ارسال شود و به F اعلام وصول گردد.

اما، وضعیت در بسته سوم که از E می‌آید. این بسته دوباره می‌آید یک بار از طریق EAB و یک بار از طریق EFB در نتیجه فقط باید به C ارسال گردد، اما باید به A و F اعلام وصول شود (همانطور که با بیت‌ها مشخص شده است).

	۰	۰	۱	۱	۱	۰	۶۰	۲۱	A
	۱	۰	۱	۰	۱	۰	۵۹	۲۱	E
	۰	۱	۰	۱	۰	۱	۶۰	۲۰	C
	۱	۱	۰	۰	۰	۱	۵۹	۲۱	D

بافر بسته برای مسیریاب B در شکل ۱۳-۵.

اگر بسته‌ها اولیه هنوز در بافر باشد و بسته تکراری دریافت شود، بیتها باید تغییر کنند و به عنوان مثال اگر قبل از این که وارده چهارم موجود در جدول ارسال شود. یک کپی از حالت C برسد این شش بیت به ۱۰۰۰۱۱ تغییر می‌کند تا نشان دهد که بسته باید به f اعلام وصول شود، ولی نباید به آنجا ارسال گردد. محاسبه مسیرهای جدید

وقتی مسیریاب مجموعه کاملی از بسته‌های حالت پیوند را جمع اوری کرد، می‌تواند گراف کامل زیر شبکه را ایجاد نماید، زیرا همه پیوندها نمایش داده می‌شوند در واقع هر پیوند دوبار نمایش داده مس شود در هر جهت یکبار از میانگین دو مقدار یا از هر کدام به طور جداگانه می‌توان استفاده کرد.

اکنون الگوریتم دیکسترا را می‌توان اجرا کرد تا کوتاه ترین مسیر به همه مقصدها را بیاید نتایج این الگوریتم می‌تواند در جدول مسیر یابی قرار گیرد و عمل عادی از سر گرفته شود.

برای زیر شبکه‌ای با n مسیریاب که هر کدام k همسایه داشته باشد حافظه لازم را برای ذخیره داده ورودی متناسب با kn است این موضوع در شبکه‌های بزرگ

می‌تواند مشکل‌زا باشد زمان محاسبه نیز ممکن است زیاد باشد با این وجود مسیریابی حالت پیوند در بسیاری از حالت‌های عملی به خوبی کار می‌کند.

به هر حال مشکلات نرم‌افزاری و سخت‌افزاری این الگوریتم می‌تواند موجب بروز خساراتی شود در الگوریتم‌های دیگر نیز همین‌طور است به عنوان مثال اگر مسیریابی خطی را فاقد آن است تقاضا کند؛ یا خطی را که دارای آن است از دست بدهد، گراف زیر شبکه‌های درست نخواهد بود. اگر مسیریاب در ارسال بسته‌ها شکست بخورد یا در حین ارسال، آنها را خراب می‌کند. مشکلاتی پیش می‌آید. نرم‌افزاری و سخت‌افزاری این الگوریتم می‌تواند موجب بروز خساراتی شود در الگوریتم‌های دیگر در همین‌طور است به عنوان مثال اگر مسیریابی خطی را فاقد آن است تقاضا کند، یا خطی را که دارای آن است از دست بدهد گراف زیر شبکه درست نخواهد بود. اگر مسیریاب در ارسال بسته‌ها شکست بخورد یا در حین ارسال، آنها را خراب کند، مشکلات پیش می‌آید، سرانجام، اگر حافظه کافی وجود نداشته باشد یا محاسباتی مسیریابی را به درستی انجام ندهد، اتفاقات بدی خواهد افتاد وقتی شبکه دارای ده‌ها یا صدها هزار گره باشد شکست گاه به گاه



مسیریاب اهمیت می‌یابد در این خصوص باید سعی کرد خسارت را کاهش داد. پرلن (۱۹۸۸) این مشکلات و راه‌های آنها را به تفصیل بحث کرد. مسیر یابی حالت پیوند در شبکه‌های واقعی به طور گسترده به کار رفته است، لذا مثال هایی در رابطه با آن ارائه شده اند، قرار داد ospf به طور فزاینده‌ای در زیر شبکه‌ای به کار گرفته می‌شود، از الگوریتم‌های حالت پیوند استفاده می‌کند ospf که به طور فزاینده‌ای در زیر شبکه به کار گرفته می‌شود، از الگوریتم حالت پیوند استفاده می‌کند.

قرارداد حالت پیوند مهم دیگر is-is (سیستم میانی - سیستم میانی) نام دارد که برای شبکه dec طراحی شد و بعداً iso آنرا پذیرفت تا در قرارداد داد لایه شبکه بی اتصال خود یعنی clnp به کار گیرد از آن پس، اصلاح شد تا به سایر قرار دادها به خصوص ip خدمات ارائه کند is-is در بسیاری از ستونهای فقرات اینترنت به کار گرفته شد از جمله در ستون فقرات nsfnat قدیمی و در بعضی از سیستمهای سلولی دیجیتال مانند cdpd نیز به کار رفت شبکه novel از شکل تغییر یافته‌ای از is-is (nlsp) برای مسیر یابی بسته‌های ipx استفاده می‌کند.

اساساً is-is تصویری از توپولوژی مسیریاب را توزیع می‌کند که از آن کوتاهترین مسیرها محاسبه می‌شوند هر مسیر یاب، در اطلاعات حالت پیوند خود، اعلام می‌درد که به کدام آدرسهای لایه شبکه مستقیماً دسترسی دارد این آدرس ها، می‌توانند : از متد خود ایستایی مربوط به بازسازی‌های حالت پیوند غرق کردن، مفهوم مسیریاب تعیین شده در شبکه مستقیماً دسترسی دارد این آدرس‌ها می‌توانند ip-ipv6 talk یا بسیاری از آدرس‌های لایه شبکه را پشتیبانی کند.

ospf بسیاری از ابداعاتی را که is-is طراحی کرده پذیرفت ospe چند سال بعد از iS-IS طراحی شد این ابداعات عبارتند از متد خود ایستایی مربوط به بازسازی‌های حالت پیوند غرق کردن، مفهوم مسیریاب تعیین شده در شبکه محلی، و متد محاسبه و پشتیبانی تقسیم مسیر و مقیاس‌های چندگانه، در نتیجه تفاوت اندکی بین IS-IS و OSPF وجود دارد. مهمترین اختلاف این است که IS-IS طوری رمز گذاری شده است که می‌توان به طور همزمان اطلاعاتی راجع به قراردادهایی که لایه شبکه چندگانه را انتقال داد، این ویژگی در OSPF وجود ندارد این امتیاز در محیطهای قرارداد چندگانه بزرگ، ارزشمند است.

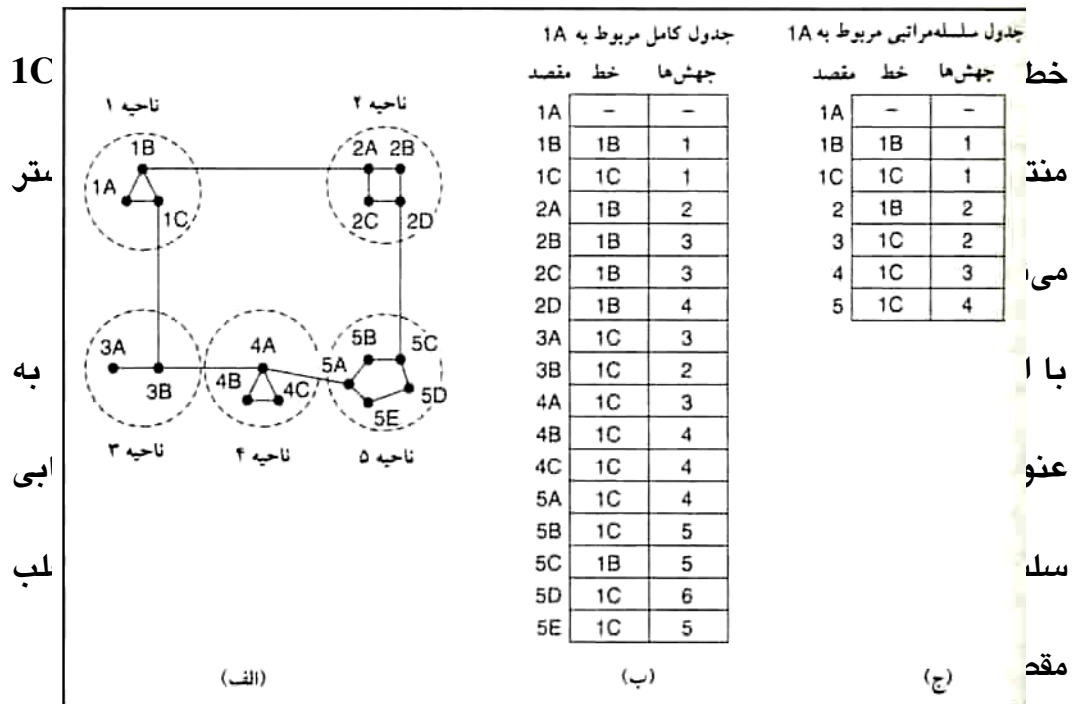
## مسیریابی سلسله مراتبی

با بزرگ شدن اندازه شبکه، جدول‌های مسیر یابی مسیریابنیز به تناسب آن رشد می‌کنند. با بزرگ شدن اندازه جدول‌های، نه تنها حافظه مصرف شده بیشتر می‌گردد، بلکه زمان لازم برای جست و جو در جدول بیشتر می‌شود. و برای گزارش وضعیت آنها به پنهان باند بیشتری نیاز است. ممکن است شبکه‌های به حدی رشد که دیگر امکان نداشته باشد. که هر مسیر باب برای هر مسیریاب دیگر دارای یک وارده باشد، لذا مسیر یابی به صورت سلسله مراتبی انجام می‌شود. مانند شبکه تلفن.)

وقتی مسیر یابی سلسله مراتبی به کار گرفته می‌شود، مسیر یاب‌ها به ناحیه‌هایی تقسیم می‌شوند به طوری که هر مسیریابدر ناحیه خودش تمام جزئیات مربوط به چگونگی ارسال بسته‌ها به مقصدها را می‌داند اما از ساختار داخلی سایر ناحیه خبر ندارد. وقتی شبکه‌های مختلفی به هم وصل می‌شوند. طبیعی است که باید به صورت ناحیه‌های جداگانه در نظر گرفته شوند تا نیاز نباشد مسیر یاب‌های موجود در یک شبکه، از ساختار توپولوژیکی مسیر یاب‌های دیگر اطلاع داشته باشند.

در شبکه‌های بزرگ، امکان دارد سلسله مراتب دو سطحی کافی باشد، امکان دارد لازم باشد که ناحیه‌ها به صورت خوشه‌ها دسته بندی شوند. خوشه‌ها به منطقه‌هایی تقسیم تقسیم شوند و غیره این روند آنقدر ادامه می‌یابد تا دیگر اسمی برای گروه بندی وجود نداشته باشند. به عنوان مثال از سلسله مراتب چند سطحی، فکر کنید که بسته چگونه می‌توانید ترافیک را به مسیر یابهای محلی دیگر هدایت کند، اما ترافیک‌ها خارج از ایالت را به مسیریاب‌لوس آنجلس می‌فرستد. مسیریاب لوس آنجلس می‌تواند ترافیک را به مسیریاب‌های محلی دیگر هدایت کند، اما ترافیک‌های ناحیه‌ای را به نیویورک می‌فرستد. مسیریاب نیویورک می‌تواند طوری برنامه نویسی شود که کل ترافیک را به مسیریابی در کشور مقصدی که مسئول کنترل ترافیک ناحیه‌ای است، مثل نایروبی، هدایت کند، سرانجام، بسته به سمت پایین درخت در کنیا حرکت می‌کند تا به مالیندی برسد.

شکل ۱۱ یک مثال کمی از مسیریابی در سلسله مراتب دو سطحی با پنج ناحیه را نشان می‌دهد. جدول مسیریابی کامل مربوط به مسیریاب 1A دارای ۱۷ وارده است (شکل ۱۱) (ب) وقتی مسیریابی‌های محلی، وارده‌های وجود دارد، اما ناحیه‌های دیگر در یک مسیر باب جمع شده‌اند لذا کل ترافیک ناحیه ۲ از طریق



مسیریابی سلسله مراتبی.

وقتی شبکه منفردی بسیار بزرگ می‌شود این سوال مطرح است: سلسله مراتب چند سطح باید داشته باشد؟ بعنوان مثال زیر شبکه‌ای با ۷۲۰ مسیریاب را در نظر بگیرید. اگر سلسله مراتبی وجود نداشته باشد، هر مسیریاب به ۷۲ وارده جدول نیاز دارد اگر زیر شبکه به ۲۴ ناحیه ۳۰ مسیریابی تقسیم شود هر مسیریاب نیازمند ۳۰ وارده محلی و ۲۳ وارده راه دور است که مجموع آن ۵۳ وارده است. اگر سلسله مراتب سه سطحی انتخاب شود، با هشت دسته که هر

کدام حاوی ۹ ناحیه از مسیر یاب‌ها باشند هر مسیر یاب برای مسیریابی محلی به ۱۰ وارده نیاز دارد و برای مسیریابی به سایر نواحی در دسته خود به ۸ وارده نیاز دارد و برای خوشه‌های راه دور به ۷ وارده نیاز دارد که در مجموع برابر با ۲۵ است. کامون و کلینراک (۱۹۷۹) کشف کردند که بهترین تعداد سطوح در زیر شبکه‌ای با  $\ln N$  است که به ازای هر مسیر یاب به  $\ln N$  وارده نیاز دارد. آنها همچنین نشان دادند که افزایش میانگین طول مسیر در اثر مسیریابی سلسله مراتبی اندک است و اغلب قابل قبول خواهد بود.

#### مسیریابی پخشی

در بعضی از کاربردها میزبانها می‌خواهند پیام‌هایی را به تمام یا بعضی از میزبانها ارسال کنند، بعنوان مثال خدمات توزیع گزارشات هواشناسی، بازسازی‌های بازار سهام، یا برنامه رادیویی روزمره، با عمل پخش به تمام ماشینها و خواندن اطلاعات توسط آن ماشینها بهتر کار می‌کنند ارسال همزمان بسته‌ای به تمام مقصدها، پخش کردن نام دارد. برای انجام آن راههای گوناگونی پیشنهاد شدند.

یک روش پخش که نیاز به ویژگی خاصی از زیر شبکه ندارد، این است که منبع، بسته متفاوتی را به تمام مقصدها بفرستد. ای روش نه تنها پهنای باند زیادی را مصرف می‌کند بلکه لازم است منبع لیست کاملی از تمام مقصدها را داشته باشد در عمل این راه حل ممکن است، تنها امکان باشد، اما روش مطلوبی نیست.

روش دیگر، غرق کردن است. گرچه غرق کردن برای ارتباطات نقطه به نقطه مناسب نیست، ولی برای پخش می‌تواند قابل قبول باشد به ویژه اگر هیچکدام از روشهای تشریح شده زیر، قابل استفاده نباشند. مشکل غرق کردن به عنوان تکنیک پخش این است که بسته‌های زیادی تولید می‌کند و پهنای باند بسیاری را مصرف می‌نماید. این مشکلات در به کار گیری آن بعنوان الگوریتم مسیریابی نقطه به نقطه نیز مطرح اند.

الگوریتم سوم مسیریابی مقصدهای چندگانه است. اگر این روش به کار گرفته شود، هر بسته یا حاوی لیستی از مقصدها است یا حاوی نگاشت بیتی است که نشان دهنده مقصد است. وقتی بسته‌ای به مسیریاب می‌رسد مسیریابها تمام مقصدها را کنترل می‌کند تا مجموعه‌ای از خطوط خروجی مورد نیاز را تعیین نماید (خط خروجی در صورتی مورد نیاز است که بهترین مسیر به حداقل یکی از

مقصدها باشد) مسیریاب نسخه جدیدی از بسته را برای تمام خطوط خروجی که مورد استفاده قرار گرفتند تولید می‌کند و در هر بسته فقط مقصدهایی را قرار می‌دهد که خط را به کار می‌گیرند. در نتیجه مجموعه مقصد بین خطوط خروجی تقسیم می‌شود. پس از تعداد کافی از جهش‌ها، هر بسته فقط یک مقصد را با خود می‌برد و می‌توان با آن مثل بسته معمولی برخورد کرد. مسیریابی مقصدهای چندگانه مانند بسته‌هایی است به طور جداگانه آدرس دهی شدند، مگر هنگامی که چند بسته از یک مسیر هدایت شوند که در این صورت یکی از آنها کل هزینه را می‌پردازد و بقیه مجانی عبور می‌کنند.

چهارمین الگوریتم پخش، برای مسیریاب آغازگر پخش، از درخت بایگانی استفاده می‌کند یا از هر درخت پوشای مناسب استفاده می‌نماید. درخت پوشا زیر مجموعه‌ای از زیرشبکه است که تمام مسیریابها را در بر می‌گیرد و فاقد حلقه است. اگر هر مسیریاب بداند کدامیک از خطوط متعلق به درخت پوشا است، می‌تواند بسته دریافتی را بر روی تمام خطوط درخت پوشا به جز خطی که بسته از آن رسیده است کپی نماید این روش از پهنای باند به خوبی استفاده می‌کند؛ و کمترین تعداد بسته‌های مورد نیاز برای انجام این کار را تولید می‌نماید. این روش



از پهنای باند به خوبی استفاده می‌کند و کمترین تعداد بسته‌های خمورد نیاز برای انجام کار را تولید می‌نماید. تنها مشکل این است که هر مسیریاب باید اطلاعاتی راجع به درخت پوشا داشته باشد. گاهی این اطلاعات وجود دارند (مثلاً، در مسیریابی حالت پیوند)، اما گاهی نیز وجود ندارد (مثلاً در مسیریابی بردار فاصله).

آخرین الگوریتم پخشی، حتی هنگامی که مسیریابها اطلاعاتی راجع به درختهای پوشا نداشته باشند، سعی می‌کند رفتار الگوریتم قبلی را تخمین بزند. این ایده، پیشروی مسیر معکوس نام دارد و بسیار ساده است. وقتی بسته پخشی به مسیریاب می‌رسد مسیریاب کنترل می‌کند آیا بسته دریافت شده از منبع از همان خطی آمد که بسته‌ها در حالت عادی برای آن منبع پخش ارسال می‌شوند یا خیر. اگر اینطور باشد، احتمال این که بسته پخشی خودش بهترین مسیر را از منبع طی کند بسیار زیاد است و اولین نسخه‌ای است که به مسیریاب می‌رسد به این ترتیب مسیریاب نسخه‌هایی از آن را به تمام خطوی به جز خطی که از آن آمده است می‌فرستد اما اگر بسته پخشی برای رسیدن به منبع از خطی غیر از خط بهینه وارد شود بسته بعنوان بسته تکراری نادیده گرفته می‌شود.

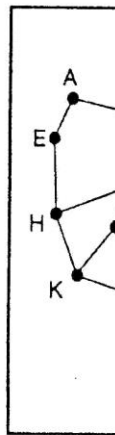
نمونه‌ای از الگوریتم پیشروی مسیر معکوس در شکل ۱۲ آمده است. قسمت (الف) زیر شبکه را نشان می‌دهد، قسمت (ب) درخت بایگانی مربوط به مسیر I آن زیر شبکه را نشان می‌دهد و قسمت (ج) چگونگی عملکرد الگوریتم مسیر معکوس را نشان می‌دهد در جهش اول U بسته‌هایی را به  $N, H, H, F$  می‌فرستد (که در سطر دوم درخت نشان داده شده است). هر کدام از این بسته‌ها از مسیر بهینه به I دریافت می‌شوند (با فرض اینکه مسیر بهینه در درخت بایگانی باشد) و دور حرف آن دایره‌ای کشیده شده است. در جهش دوم هشت بسته تولید می‌شوند؛ هر مسیریابی که در جهش اول بسته‌ای را دریافت کرد، دو بسته تولید می‌کند. ضمن تولید تمام این هشت بسته به مسیریاب‌های ملاقات نشده قبلی می‌رسند که پنج بسته از آنها در امتداد خط بهینه به مقصد می‌رسد. از شش بسته‌ای که در جهش سوم تولید می‌شود فقط سه تا از مسیر بهینه می‌رسند (در  $K, E, C$ ) و بقیه تکراری‌اند. پس از پنج جهش و ۲۴ بسته، پخش خاتمه می‌یابد در حالی که اگر درخت بایگانی دنبال می‌شد چهار جهش و ۱۴ بسته لازم بود.

امتیاز مهم پیشروی مسیر معکوس این است که کارایی خوبی دارد و پیاده‌سازی آن ساده است. لازم نیست مسیریابها اطلاعاتی راجع به درختهای پوشا داشته

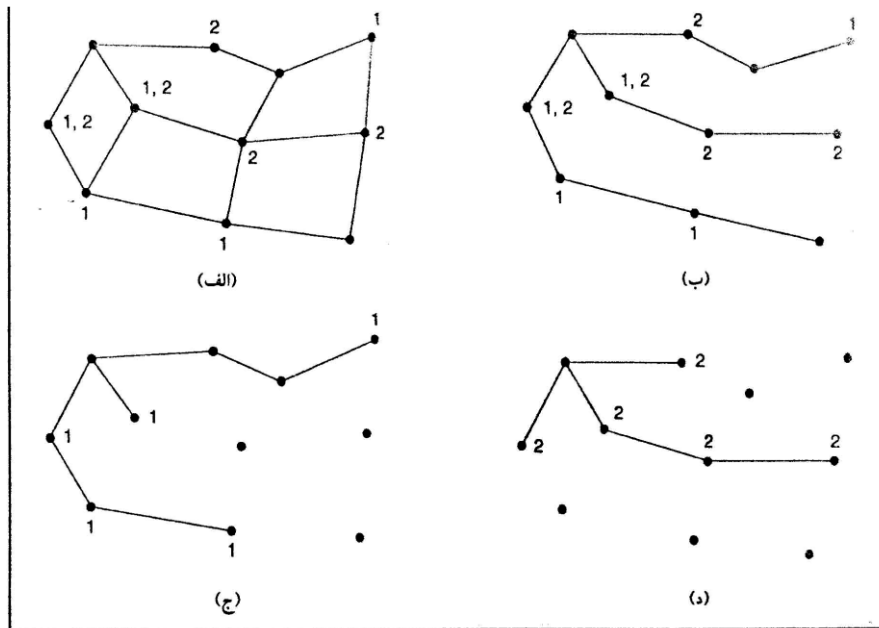
باشند، و در هر بسته نیازی به سربرار لیست مقصدها یا نگاشت خصی نیاز ندارد، در حالی که فرایند غرق کردن به این راهکار نیاز دارد (شمارنده جهش در هر بسته و اطلاع قبلی از قطر زیر شبکه، یا لیستی از بسته هایی که تا کنون از هر منبع دریافت شده اند).

#### مسیریابی چند پخش

در بعضی از کاربردها فرایندهای مستقل از هم به صورت گروهی کار می کنند مانند گورهی از فرایندهای سیستم بانک اطلاعاتی توزیعی را پیاده سازی می کنند. در مواد اغلب لازم است یکی از فرایندها پیامی را به سایر اعضای گروه ارسال نماید. اگر گروه کوچک باشد، می تواند پیام نقطه نقطه را به تمام اعضا صادر کند. اگر گروه بزرگ باشد، این راهبرد گران تمام می شود. گاهی می توان از پخش استفاده کرد، اما استفاده از پخش برای اطلاع دادن به ۱۰۰۰ ماشین در شبکه ای با میلیونها گره کارآمد نیست، زیرا اغلب گیرنده ها علاقه ای به پیام ندارند (حتی در حالت بدتر، علاقه دارند و تصور دیدن آن را ندارند). بنابراین باید بتوانیم پیامها را به گروهی بفرستیم که اندازه آن گروه از نظر عددی بزرگ باشد ولی در مقایسه با کل شبکه کوچک باشد.



بر معکوس



(الف) زیرشبکه. (ب) درخت پوشای مربوط به سمت چپ‌ترین مسیریاب. (ج) درخت چندبخشی

برای انجام چند بخشی نیاز به مدیریت گروه است روشهایی برای ایجاد و حذف گروه لازم است و نیاز به فرایندهایی برای اتصال به گروه و ترک آن است. انجام این کارها به عهده الگوریتم مسیریابی نیست. آنچه که به الگوریتم مربوط می‌شود این است که وقتی فرایندی به گروه ملحق می‌شود، آن را به میزبان خود خبر می‌دهد. توجه به این نکته مهم است که مسیریابها می‌دانند کدام میزبان آنها به کدام گروه تعلق دارند. یا میزبانها باید تغییر در گروه را به اطلاع مسیریابها برسانند، یا مسیریابها هر از چند گاهی از میزبانها درخواست کنند. در هر دو روش مسیریابها می‌فهمند که میزبانهای آنها در چه گروه‌هایی قرار دارند. مسیریابها به همسایه‌های خود خبر می‌دهند و بدین ترتیب اطلاعات از طریق زیرشبکه انتشار می‌یابد.

برای مسیریابی چند پخشی، هر مسیریاب، درخت پوشایی را ایجاد می‌کند که تمام مسیریابهای موجود در زیر شبکه را در بر گیرد. بعنوان مثال در شکل ۱۳ (الف) زیر شبکه‌ای با دو گروه ۱ و ۲ وجود دارد. بعضی از مسیریابها به میزبانهای دست یافتند که متعلق به یک یا هر دو گروه است (همانطور که در شکل آمده است). درخت پوشای مربوط به مسیریاب سمت چپ، در شکل ۱۳ (ب) آمده است. وقتی فرایندی بسته چند پخشی را به گروهی می‌فرستد، اولین مسیریاب، درخت پوشای خود را بررسی کرده آن را هرس می‌کند. برای این کار تمام خطوطی را که به میزبانهایی می‌روند که عضو این گروه نیستند حذف می‌کند. در مثال مورد نظر ما شکل ۱۳ (ج) درخت پوشای هرس شده مربوط به گروه ۱ را نشان می‌دهد. شکل ۱۳ (د) درخت پوشای هرس شده مربوط به گروه ۲ را نشان می‌دهد. بسته‌های چند بخشی فقط از طریق درخت پوشای مناسبی ارسال می‌گردند. راه‌های گوناگونی برای هرس کردن درخت پوشا وجود دارد. ساده ترین آنها وقتی مورد استفاده قرار می‌گیرد که از مسیریاب حالت پیوند استفاده گردد و هر مسیریاب از توپولوژی کامل زیر شبکه آگاهی داشته باشد از جمله بداند کدام مسیریاب به کدام گروه‌ها تعلق دارند. سپس درخت پوشا را می‌توان با شروع از

انتها هر مسیر و ادامه دادن به سمت ریشه هرس کرد برای این کار باید تمام مسیریابهایی را که متعلق به گروه مورد نظر نیستند حذف کرد.

در مسیریابی بردار فاصله باید از روش دیگری برای هرس کردن استفاده کرد الگوریتم اصلی پیشروی مسیر معکوس است اما هر گاه مسیریاب فاقد میزبانی به گروه خاصی متعلق باشد و به مسیریابهای دیگر متصل نباشد پیام چند بخشی برای آن گروه را دریافت می‌کند، آن گروه با پیام PRUNE پاسخ می‌دهد و به فرستنده می‌گوید که بسته‌های چند بخشی دیگری نفرستد. وقتی این پیامها به تمام ورودی‌های یک مسیریاب برسند که در بین میزبان‌هایش فاقد اعضای گروه است این مسیریاب نیز می‌تواند با PRUNE پاسخ می‌دهد. در این صورت زیر شبکه به طور بازگشتی هرس می‌شود.

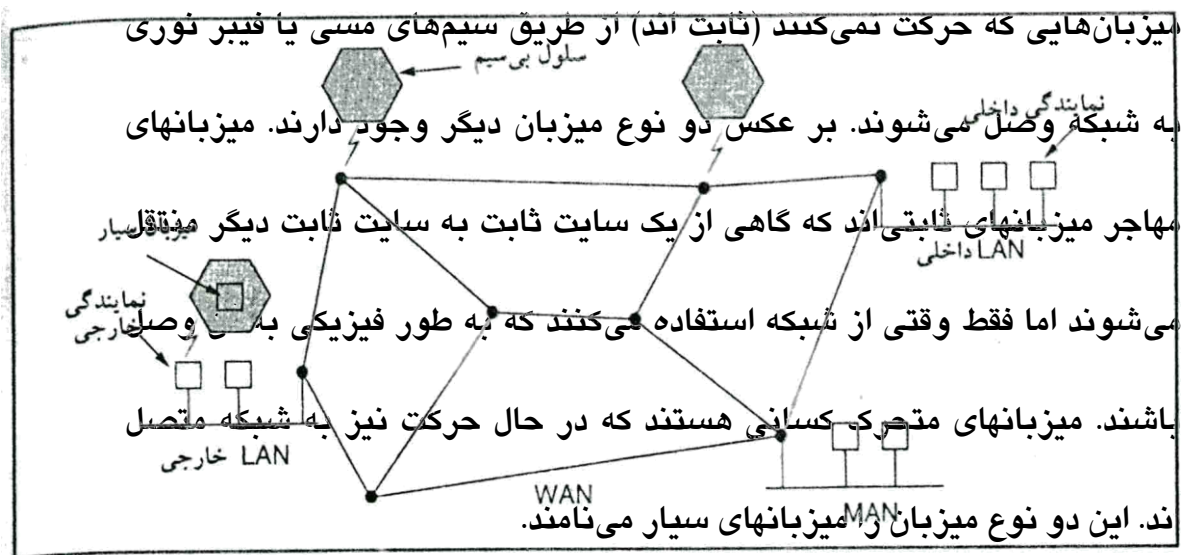
یکی از عیب‌های این الگوریتم این است که در شبکه‌های بزرگ به خوبی کار نمی‌کند فرض کنید شبکه‌ای دارای  $n$  گروه است و هر گروه به طور متوسط دارای  $m$  عضو است. برای هر گروه  $m$  درخت هرس شده پوشا باید ذخیره گردد و در نتیجه تعداد کل درختها  $mn$  است. وقتی گروه‌ها بزرگ باشند حافظه زیادی برای ذخیره همه درختها لازم است.

طراحی دیگر از درختهای هسته‌ای (بالاردای و همکاران ۱۹۹۳) استفاده کرده است. در اینجا در هر گروه یک درخت پوشا محاسبه می‌شود، به طوری که ریشه (هسته) در نزدیک به وسط گروه قرار دارد. برای ارسال پیام چند بخشی میزبان آن را به هسته می‌فرستد و چند پختی در سراسر درخت پوشا انجام می‌شود. گرچه این درخت برای تمام منابع بهینه نیست کاهش  $m$  درخت به یک درخت در هر گروه موجب صرفه جویی در حافظه می‌شود.

#### مسیریابی برای میزبانهای سیار

امروزه، میلیونها نفر کامپیوترهای قابل حمل دارند و علاقه مندند در هر جا که هستند پست الکترونیکی خود را بخوانند و به سیستم فایل معمولی خود نیز دسترسی داشته باشند. این میزبانهای سیار موجب پیچیدگی جدیدی می‌شوند: باری هدایت بسته‌ای به میزبان سیار، شبکه باید ابتدا آن را بیابد موضوع ملحق شدن میزبانهای سیار به شبکه خیلی جوان است اما در اینجا برخی از مشکلات را مطرح کرده راه حل‌های ممکن را ارائه می‌کنیم.

مدل میانی که طراحان از آن استفاده می‌کنند در شکل ۱۴ آمده است در اینجا یک شبکه گسترده وجود دارد که حاوی مسیریابها و میزبانها است. شبکه‌های محلی و شهری و سلول‌های بی سیم به این شبکه گسترده متصل‌اند.



شبکه گسترده‌ای که شبکه‌های محلی و شهری و سلول‌های بی‌سیم به آن متصل‌اند.

فرض می‌شود تمام میزبانها موقعیت داخلی ثابتی داشته باشند که هرگز تغییر نکند. میزبانها آدرس داخلی ثابتی نیز دارند که محل آنها را مشخص می‌کند این حالت را با شماره تلفن ۱-۲۱۲-۵۵۵۱۲۱۲ مقایسه کنید که نشان دهنده ایالات متحده (کد کشور ۱) و جزیره مان هاتان (۲۱۲) است. هدف مسیریابی در سیستمی



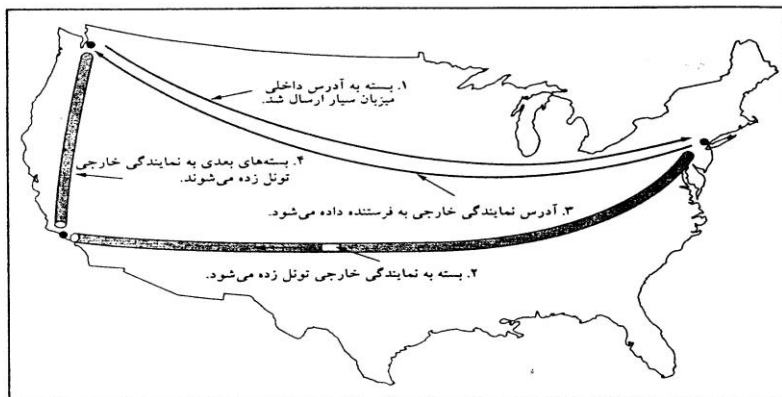
با میزبانهای سیار، عبارت است از: ارسال بسته‌ها به میزبانهای سیار به کمک آدرسهای داخلی آنها، و خواندن بسته‌ها توسط میزبانها در هر جایی که هستند. در مدل شکل ۱۴ جهان (از نظر جغرافیایی) به واحدهای کوچکی تقسیم شده است. این واحدها را ناحیه می‌نامیم به طوری که هر ناحیه یک شبکه محلی یا سلول بی سیم است هر ناحیه دارای یک یا چند نمایندگی خارجی است است که فرایندهایی هستند که تمام میزبانهای سیار ناحیه را نگهداری می‌کند بعلاوه هر ناحیه دارای نمایندگی داخلی است. این نمایندگی میزبانهایی را که خانه شان در این ناحیه قرار دارد ولی فعلا با ناحیه دیگری در حال ملاقات‌اند نگهداری می‌کند. وقتی میزبان جدیدی وارد ناحیه‌ای شود، چه از طریق اتصال به آن (مثل وصل شدن به شبکه محلی)، یا سرگردان بودن در سلول، کامپیوترش باید خودش را با نمایندگی خارجی ثبت نماید. روند ثبت به صورت زیر انجام می‌شود:

۱. هر نمایندگی خارجی به طور تناوبی بسته‌ای را پخش می‌کن تا وجود و آدرس خود را اعلام کند. میزبان سیار تازه وارد ممکن است منتظر یکی از این پیامها باشد اما اگر در مدت معدین چنین پیامی نرسد، میزبان سیار می‌تواند بسته‌ای را پخش کند و بگوید آیا هیچ نمایندگای خارجی وجود دارد؟

۲. میزبان سیار، با نمایندگی خارجی ثبت می‌شود برای این کار آدرس داخلی خود آدرس لایه پیونده داده فعلی، و اطلاعات سری دیگر را ارائه می‌کند.

۳. نمایندگی خارجی با نمایندگی داخلی میزبان سیار تماس برقرار می‌کند و

می‌گوید  
نمایندگی  
اطلاعات  
دارد.



مسیریابی بسته برای کاربران سیار.

۴. نمایندگی داخلی اطلاعات سری را که حاوی مهر زمان است، بررسی می‌کند تا ثابت کند در چند ثانیه قبل تولید شده است. اگر راضی باشد، به نمایندگی خارجی می‌گوید که پیشروی نماید.

۵. وقتی نمایندگی خارجی اعلام وصولی را از نمایندگی داخلی دریافت می‌کند یک وارده در جدول خود ایجاد می‌نماید و اطلاع می‌دهد که میزبان سیار ثبت شده است.

ایده آل آن است که وقتی کاربری ناحیه را ترک می‌کند، باید اطلاع دهد تا از ثبت خارج شود، اما اغلب کاربران به طور غیر منتظره، کامپیوترهای خودشان را خاموش می‌کنند.

وقتی بسته‌ای به میزبان ارسال می‌گردد به شبکه محلی داخلی کاربر هدایت می‌شود زیرا چیزی است که آدرس، انجام آن را می‌طلبد، مانند آنچه که در مرحله ۱ از شکل ۱۵ نشان داده شده است اینجا فرستند در شهر شمال غربی سیتل می‌خواهد بسته‌ای را از طریق ایالات متحده در نیویورک به یک میزبان بفرستد. بسته‌های ارسال شده به میزبان سیار در LAN داخلی خود در نیویورک، توسط نمایندگی داخلی متوقف می‌شود. سپس نمایندگی داخل، محل جدید (موقتی) میزبان سیار را جست و جو می‌کند و آدرس نمایندگی خارجی را می‌یابد که میزبان سیار را جست و جو می‌کند و آدرس نمایندگی خارجی را می‌یابد که میزبان سیار را اداره می‌کند.

نمایندگی داخلی دو کار را انجام می‌دهد اول اینکه بسته را در فیلد بار مفید بسته بیرونی تر بسته بندی می‌کن و آن را به نمایندگی خارجی می‌فرستد (مرحله دو در شکل ۱۵) این راهکار را تونل سازی می‌نامند؛ در ادامه آن را بیشتر مورد بحث

قرار می‌دهیم. پس از گرفتن بسته بسته بندی شده نمایندگی خارجی بسته اصلی را از فیلد بار مفید جدا می‌کند و آن را به صورت قاب پیوند داده‌ای به میزبان سیار می‌فرستد.

دوم اینکه نمایندگی داخلی به فرستنده می‌گوید از این پس بسته‌ها را با قراردادن آنها در فیلد بار مفید بسته‌هایی که صریحاً به نمایندگی خارجی آدرس دهی می‌شوند، به میزبان سیار ارسال نماید (به جای اینکه آنها را به آدرس داخل کاربر سیال ارسال کند (مرحله ۳) اکنون بسته‌های بعدی می‌توانند مستقیماً از طریق نمایندگی خارجی (مرحله ۴) به میزبان هدایت شوند (با نادیده گرفتن موقعیت داخلی).

الگوهای مختلفی که عرضه شدند تفاوت‌هایی با هم دارند اول اینکه چه مقدار از این قرارداد توسط مسیریابها و چقدر توسط میزبانها انجام می‌شوند و در حالت دوم توسط کدام لایه در میزبانها انجام می‌گیرد دوم اینکه در الگوهای اندکی مسیریابها در بین راه آدرسهای تطبیق شده را ثبت می‌کنند، لذا می‌توانند قبل از رسیدن ترافیک به موقعیت داخلی، از آن جلوگیری کرده مجدداً آن را هدایت نمایند سوم اینکه در بعضی از الگوها به هر مهمان آدرس موقت منحصر به فردی نسبت

داده می‌شود در بعضی دیگر آدرس موقت به نمایندگی اشاره می‌کند ه ترافیک مربوط به تمام مهمانها را کنترل می‌نماید.

چهارم اینکه الگوها در چگونگی ترتیب بسته هایی که به یک مقصد آدرس دهی شدند و باید به مقصد دیگری تحویل شوند با هم فرق می‌کنند یک روش تغییر آدرس مقصد و ارسال مجدد بسته اصلاح شده است. روش دیگر این است که کل بسته آدرس محلی و هر چیز دیگر می‌توانند در فیلد بار مفید بسته دیگری که به آدرس موقتی ارسال شده است، بسته بندی گردد. سرانجام الگوها از نظر حفاظت با یکدیگر متفاوت اند. به طور کلی وقتی میزبان یا مسیریابی پیامی به این شکل را دریافت کند: اکنون شروع کن، لطفا تمام نامه‌های کیلاس را به من بفرست، ممکن است با دو پرسش مواجه باشد: با چه کسی صحبت می‌کند و آیا این ایده ایده خوبی است یا خیر.

مسیریابی در شبکه‌های موقتی

تا کنون با چگونگی مسیریابی در حالتی که میزبانها سیار و مسیریابها ثابت باشند آشنا شدید حالت دیگر این است که خود مسیریابها سیار باشند. بعضی از این موارد عبارت‌اند از:

۱. وسایل نقلیه نظامی در میدان جنگ که هیچ ساختمانی وجود ندارد

۲. ناوگان کشتی‌ها در دریا

۳. کارکنان اضطراری در هنگام وقوع زلزله که ساختمانها را خراب کرده است

۴. گروه‌هایی افرادی با کامپیوترهای کیفی در منطقه‌ای که فاقد 802.11 است.

در همه این موارد و موارد مشابه، هر گروه متشکل از یک مسیر یاب و یک میزبان است که معمولاً در یک کامپیوتر قرار دارند. شبکه‌هایی از گره‌ها که در نزدیکی یکدیگر قرار می‌گیرند. شبکه‌های موقتی یا MANET (شبکه‌های موردی همراه) نام دارند. آنها را به طور مختصر شرح می‌دهیم.

تفاوت شبکه‌های موقتی با شبکه‌های سیمی این است که تمام قوانین مربوط به توپولوژی‌های ثابت همسایگان ثابت و مشخص رابطه ثابت بین آدرس IP و مکان، و غیره باید نادیده گرفته شوند. مسیر یابها از نقطه‌ای به نقطه دیگر جا به جا می‌شوند. در شبکه سیمی اگر مسیر یاب، مسیر معتبری به مقصد داشته باشند، این مسیر همواره معتبر خواهد بود (مگر اینکه سیستم خراب شود) در شبکه‌های موقتی توپولوژی ممکن است همواره تغییر کند. در نتیجه مطلوب بودن و اعتبار

مسیرها بدون هیچ اختطاری تغییر می‌کند. بدیهی است که این شرایط موجب می‌شود مسیریابی در شبکه‌های موقتی کاملاً متفاوت از شبکه‌های ثابت باشد. الگوریتم‌های مسیریابی گوناگونی برای شبکه‌های موقتی پیشنهاد شدند. یکی از جالب‌ترین الگوریتمها AODV (بردار فاصله موقتی بر حسب تقاضا) نام دارد. شکلی از الگوریتم بردار فاصله بلمن-فورد است که در محیط سیار کار می‌کند و محدودیت پهنای باند و طول باطری را در این محیط در نظر می‌گیرد. ویژگی غیر عادی دیگر این است که الگوریتم تقاضا است یعنی فقط در صورتی که کسی بخواهد بسته‌ای را به مقصدی بفرستد، مسیری به آن مقصد را می‌یابد.

#### کشف مسیر

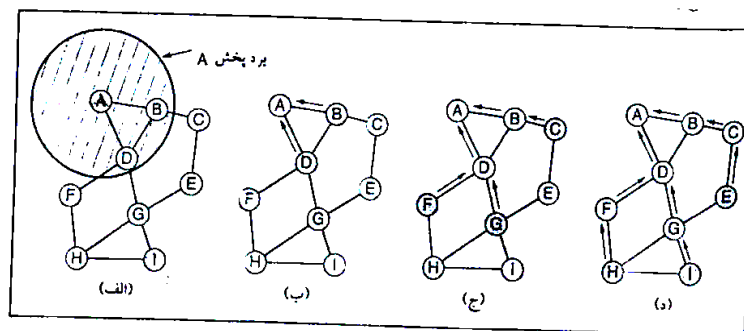
شبکه موقتی را در هر لحظه می‌توان به وسیله گرافی از گره‌ها (مسیریاب‌ها + میزبانها) توصیف کرد. دو گره در صورتی متصل به هم هستند که بتوانند مستقیماً با استفاده از رادیوهای خود با یکدیگر ارتباط برقرار کنند (در این صورت در گراف یالی بین آنها وجود دارد). چون ممکن است یکی از آنها خیلی قوی‌تر از دیگری باشد ممکن است A به B متصل باشد اما B به A متصل نباشد. اما برای سهولت فرض می‌کنیم تمام اتصالاتها متقارن هستند. توجه داشته باشید

که اگر دو گره در رادیوی یکدیگر باشند به معنای این نیست که به هم متصل هستند.

برای توصیف این الگوریتم شبکه موقتی شکل ۱۶ را در نظر بگیرید که در آن فرایندی در گره A می‌خواهد بسته‌ای را به گره I بفرستد. الگوریتم AODV در هر گره دارای جدولی است که کلید آن مقصد است و اطلاعاتی راجع به آن مقصد ارائه می‌کند از جمله کدام همسایه باید بسته را بفرستد تا به مقصد برسد و فرض کنید، A در جدول خود جست و جو می‌کند و وارده‌ای را برای I نمی‌یابد. اکنون باید مسیری را به I را کشف کند. چون این الگوریتم در صورت نیاز مسیرها را کشف می‌کند، الگوریتم تقاضا نام دارد.

برای یافتن I گره A بسته ROUTE REQUEST خاصی را می‌سازد و آن را

در D،  
کند. به  
A را



پخش ه  
گراف ب  
عنوان  
دریافت

(الف) پخش پخش A. (ب) پس از این که B و D پخش A را دریافت کردند. (ج) پس از این که C، F و G پخش A را دریافت کردند. گره‌های سایه دار گیرنده‌های جدید هستند. فلش‌ها، مسیرهای معکوس می‌کن را نشان می‌دهند.



شمارنده	شماره	شماره	آدرس	شناسه	آدرس
جهش	ترتیب	ترتیب	مقصد	تقاضا	منبع
	مقصد	منبع			

فرمت بسته **ROUTE REQUEST** در شکل ۱۹ آمده است. این فرمت حاوی آدرسهای منبع و مقصد (معمولا آدرس IP آنها) است که مشخص می کند چه کسی برای چه کسی جست و جو می کند. شناسه تقاضا یک شمارنده محلی است و توسط هر گره ای نگهداری می شود و هر وقت **ROUTE REQUEST** پخش می شود یک واحد به آن اضافه می شود فیلدهای آدرس منبع و شناسه تقاضا، یک بسته **ROUTE REQUEST** منحصر به فرد را مشخص می کنند که به گرهها اجازه می دهند بسته های تکراری را حذف کنند.

هر گره علاوه بر شمارنده شناسه تقاضا شمارنده دنباله ای دارد که هر وقت **ROUTE REQUEST** فرستاده شد (یا پاسخی به **ROUTE REQUEST** داده شد) یک واحد به آن اضافه می شود. تقریبا مثل ساعت عمل می کند و برای

تخشیص مسیر جدید از مسیر قدیم به کار می رود. فیلد چهارم در شکل ۱۹ شماره ترتیب A است و فیلد پنجم جدیدترین مقدار شماره ترتیب I است که A آن را دیده است. فیلد شمارنده جهش مشخص می کند که بسته تا کنون چند جهش انجام داد. مقدار اولیه آن صفر است.

وقتی بسته ROUTE REQUEST به گره ای می رسد به صورت زیر پردازش می شود:

۱. جفت (شناسه تقاضا و آدرس منبع) در یک جدول سابقه محلی جست و جو می شود تا مشخص شود آیا این درخواست قبلا دیده و پردازش شد یا خیر. اگر تکراری باشد، نادیده گرفته می شود و پردازش متوقف می گردد اگر تکراری نباشد این جفت در جدول سابقه قرار می گیرد و پردازش ادامه می یابد.

۲. گیرنده مقصد را در جدول مسیر خود جست و جو می کند اگر مسیر تازه ای به مقصد شناخته شود. بسته ROUTE REPLY به منبع ارسال می شود تا به آن بگوید چگونه به مقصد برسد. معنای مسیر تازه این است که شماره ترتیب مقصد که در جدول مسیریابی ذخیره شده است بزرگتر یا مساوی شماره ترتیب مقصد موجود در بسته ROUTE REQUEST است. اگر کمتر باشد مسیر ذخیره شده

قدیمی تر از مسیر قبلی است که منبع برای مقصد داشته است در نتیجه مرحله ۳ اجرا می شود.

۳. چون گیرنده مسیر تازه ای به مقصد نمی رساند به فیلد شمارنده جهش یک واحد اضافه می کند و بسته ROUTE REQUEST را پخش می کند داده ها را نیز از بسته استخراج و آن را بعنوان وارده جدید در جدول مسیر معکوس خود ذخیره می کند این اطلاعات برای ساختن مسیر معکوس به کار می روند لذا پاسخ می تواند بعدا به منبع برسد. فلشها در شکل ۱۶ برای ساخت مسر معکوس به کار می روند برای وارده مربوط به مسیر معکوس جدیدی که ساخته شد، تایمری شروع به کار می کند وقتی تایمر از کار افتاد وارده حذف می شود.

D , B نمی دانند I در کجا قرار دارد، لذا هر کدام از آنها وارده ای را برای مسیر معکوس ایجاد می کنند که به A اشاره می کند (مثل فلش های ۱۶) بسته را در حالی پخش می کنند که فیلد شماره جهش آن ۱ است. پخش حاصل از B به C , D می رسد. C وارده ای را در جدول مسیر معکوس خود ایجاد می کند و آن را پخش می کند در حالی که D آن را بعنوان پخش تکراری رد می کند. به طور مشابه پخش حاصل از D توسط B رد می شود. پخش D توسط F , G پذیرفته و ذخیره

می شود. (شکل ۱۶ ج) پس از اینکه E, H, I پخش را دریافت کردند، ROUTE REQUEST به مقصدی می رسد که می داند I در کجا قرار دارد (یعنی خود I)

شکل ۱۶ (۱۶) . . . . .

طول عمر	شمارنده جهش	شماره ترتیب مقصد	آدرس مقصد	آدرس منبع
---------	-------------	------------------	-----------	-----------

فرمت بسته ROUTE REQUEST .

در پاسخ به تقاضای ورودی I یک بسته ROUTE REPLY را می سازد (شکل ۱۸) آدرس منبع، آدرس مقصد، شمارنده جهش از تقاضای ورودی کپی می شوند، اما شماره ترتیب مقصد از شمارنده اش به حافظه منتقل می شود. فیلد شمارنده جهش برابر با صفر می شود. فیلد طول عمر مشخص می کند که مسیر چه مدت معتبر است. این بسته یک بسته تک پخش به گره ای است که بسته ROUTE REPLY از آن آمده است که در اینجا G است. سپس مسیر معکوس را به D طی می کند و به A می رسد. در هر گره شمارنده جهش یک واحد اضافه می شود لذا گره می تواند تشخیص دهد که چقدر از مقصد I فاصله دارد.

در هر گره میانی در مسیر معکوس بسته بررسی می‌شود اگر یک یا چند شرط زیر برقرار شود آن گره به عنوان مسیری به I در جدول مسیریابی محلی ذخیره می‌شود:

۱. هیچ مسیری به I شناخته نشده باشد

۲. شماره ترتیب مربوط به I در بسته ROUTE REPLY بزرگتر از مقدار

موجود در جدول مسیریابی است

۳. شماره‌های ترتیب برابرند ولی مسیر جدید کوتاه تر است

بدین ترتیب تمام گره‌های موجود در مسر معکوس مسیری به I به طور رایگان یاد می‌گیرند. گره هایی که بسته ROUTE REPLY را گرفتند ولی در مسیر معکوس نبودند (B, C, E, F, H در مثال ما)، با انقضای مدت تایمر مربوط، وارده‌ها را از جدول مسیر معکوس حذف می‌کنند.

لذا فرایند کشف مسیر به این صورت اصلاح می‌شود. برای یافتن مقصد، فرستنده یک بسته ROUTE REPLY را می‌فرستد که طول عمر آن ۱ است. اگر در مدت زمان معقولی پاسخ دریافت نشود، بسته ROUTE REPLY دیگری ارسال می‌شود. این بار طول عمر برابر با ۲ خواهد بود. دفعات بعد طول عمر برابر با ۳ و

۴ و ۵ و غیره خواهد شد به این ترتیب جست و جو به طور محلی انجام می‌شود  
به طور فزاینده فراگیرتر می‌شود.

### نگهداری مسیر

چون گره‌ها می‌توانند جا به جا شوند یا خاموش شوند توپولوژی خود به خود  
تغییر می‌کند. بعنوان مثال در شکل ۱۶ اگر G خاموش شود، A متوجه نمی‌شود  
که برای رسیدن به I استفاده می‌کرد. (ADGI) معتبر نیست. الگوریتم باید این  
موضوع را اداره کند. هر گره به طور تناوبی پیام Hello می‌فرستد. انتظار

یافت	مقصد	جهش بعدی	فاصله	همسایه‌های فعال	فیلد های دیگر
	A	A	1	F, G	
	B	B	1	F, G	
نر به	C	B	2	F	
	E	G	2		
	F	F	1	A, B	
	G	G	1	A, B	
	H	F	2	A, B	
ند که	I	G	2	A, B	

(الف)

(ب)

می‌را  
نشو  
آن ه  
پاس

(الف) جدول مسیریابی قبل از خاموش شدن G. (ب) گراف پس از خاموش شدن G.

این اطلاعات برای از بین بردن مسیرهایی به کار می‌روند که دیگر کار نمی‌کنند.  
برای هر مقصد ممکن، هر گره‌ای مثل N همسایه‌هایی را نگهداری می‌کند که برای  
آن بسته‌هایی را که در اثنای آخرین  $\Delta T$  ثانیه فرستادند تا به آن مقصد ارسال

شوند. این همسایه‌ها را همسایه‌های فعال  $N$  برای آن مقصد می‌نامند.  $N$  برای انجام این کار از جداول مسیریابی استفاده می‌کند که کلید آن مقصد است و حاوی گره خروجی برای رسیدن به مقصد، شمارنده جهش به مقصد، جدیدترین شماره تریب مقصد و لیست همسایه‌های فعال آن مقصد است. نمونه‌ای از جدول مسیریابی برای گره  $D$  در توپولوژی مثال ما در شکل ۱۹ - الف آمده است.

وقتی یکی از همسایه‌های  $N$  غیر قابل دسترسی می‌شود،  $N$  جدول مسیریابی خود را بررسی می‌کند تا مشخص کند کدام مقصدها مسیرهایی دارند که از همسایه ناپدید شده استفاده می‌کنند برای هر کدام از این مسیرها به همسایه‌های فعال اطلاع داده می‌شود که مسیر آن‌ها از طریق  $N$  نامعتبر است و باید از جداول مسیریابی آنها حذف شود. سپس همسایه‌های فعال به طور بازگشتی به همسایه‌های فعال خود خبر می‌دهند و غیره. این روند ادامه می‌یابد تا تمام مسیرهایی که به گره ناپدید شده بستگی دارند از تمام جدولهای مسیریابی حذف شوند.

بعنوان مثالی از نگهداری مسیر، مثال قبلی خود را در نظر می‌گیریم اما فرض می‌کنیم  $G$  خاموش می‌شود. توپولوژی تغییر یافته در شکل ۱۹ - ب آمده است

وقتی D می‌فهمد که G ناپدید شد، به جدول مسیریابی خود نگاه می‌کند و می‌بیند که G در مسیرهایی به E, G, I استفاده شده است. اجتماع همسایه‌های فعال مربوط به این مقصدها مجموعه {A,B} است. به عبارت دیگر، A و B در بعضی از مسیرهای خود به G وابسته‌اند و لذا باید به آنها اطلاع داده شود که این مسیرها دیگر معتبر نیستند. D با ارسال بسته‌هایی این خبر را به آنها می‌دهد که باعث می‌شود آنها جدول‌های مسیریابی خودشان را نوسازی کنند. علاوه بر این، D وارده‌های مربوط به E, G و I را از جدول مسیریابی خود حذف می‌کند.

تفاوت عمده بین الگوریتم AODV و بلَمَن-فورد این است که در AODV گره‌ها به طور تناوبی پخش‌هایی را که حاوی جدول‌های مسیریابی آنها باشد، انجام نمی‌دهند. این تفاوت منجر به صرفه جویی در پهنای باند و مصرف باتری می‌شود.

ADOV قادر است مسیریابی پخشی و چندپخشی را انجام دهد. مسیریابی در شبکه‌های موردی، موضوع پژوهش است.

جست و جوی گره در شبکه‌های نظیر به نظیر



یک پدیده نسبتاً جدید، شبکه نظیر به نظیر است که در آن تعداد زیادی از افراد منابع مشترکی استفاده می‌کنند. معمولاً این افراد اتصال دائمی سیمی با اینترنت دارند. اولین کاربرد گسترده فناوری نظیر به نظیر جرم گروهی بود: ۵۰ میلیون کاربر Napster آوازهایی با حق کپی رایت را بودن مجوز مالکین کپی رایت مبادله کردند و دادگاه با وجود اختلاف نظرهای زیاد، Napster را متوقف کرد. با این وجود، فناوری نظیر به نظیر، کاربردهای جالب و قانونی دارد. مشکلاتی مثل مسیریابی دارد، ولی این مشکلات با آن چه که تاکنون مطالعه کردیم متفاوت هستند. با این وجود، مروری سریع به آن‌ها خواهیم داشت.

جالب بودن سیستم‌های نظیر به نظیر به این دلیل است که کاملاً توزیعی اند. تمام گره‌ها متقارن‌اند و کنترل مرکزی یا سلسه مراتبی وجود ندارد. در سیستم نظیر به نظیر، هر کاربر اطلاعاتی دارد که ممکن است برای کاربران دیگر مفید باشد. این اطلاعات ممکن است برای کاربران دیگر مفید باشد. این اطلاعات ممکن است نرم افزار رایگان، موسیقی، عکس و غیره باشد. اگر تعداد کاربران زیاد باشد، یکدیگر را نمی‌شناسند و نمی‌دانند اطلاعات مورد نیازشان را از کجا تهیه کنند. یک راه حل استفاده از بانک اطلاعاتی مرکزی است، اما به دلایلی ممکن نیست (مثلاً هیچ

کس علاقه مند به میزبان و نگهداری آن نباشد). لذا، مسئله این است که وقتی بانک اطلاعاتی یا ایندکس مرکزی وجود ندارد، کاربر چگونه گره‌ای را پیدا کند که حاوی اطلاعات مورد نظرش باشد.

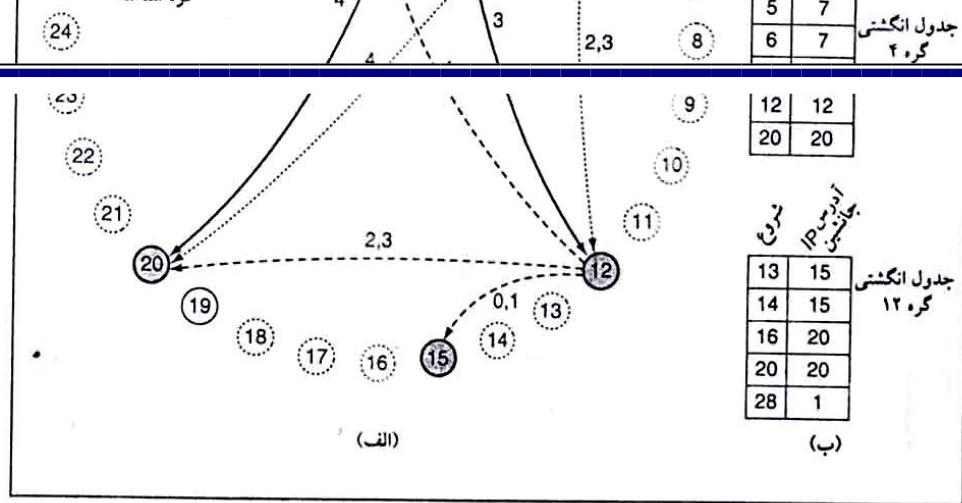
فرض می‌کنیم هر کاربر یک یا چند قلم داده مثل آواز، عکس، برنامه، فایل و غیره دارد و کاربران دیگر می‌خواهند از آن‌ها استفاده کنند. هر قلم دارای نام اسکی است. کاربر فقط رشته اسکی را می‌شناسد و می‌خواهد بداند آیا کسانی آن‌ها را کپی کردند یا خیر. چنانچه کپی کرده باشند، آدرس IP آن‌ها را بدانند.

به عنوان مثال، یک بانک اطلاعاتی توزیعی تبارشناسی را در نظر بگیرید. هر تبارشناس چند رکورد on-line برای اجداد و خویشاوندان دارد که ممکن است شامل عکس، صوت یا برش‌های ویدیوی فرد باشد. چند نفر ممکن است یک جد پدری داشته باشند، لذا رکوردهای مربوط به یک جد ممکن است در چندین گره وجود داشته باشد. نام رکورد، نام فرد به شکل کانونی است. در نقطه‌ای از این بانک اطلاعاتی، تبارشناس می‌بیند که جد پدری او در آرشیو قرار دارد و جد پدری تو ساعت جیبی طلایی خود را برای نوه اش به ارث گذاشت. اکنون تبارشناس نام نوه را می‌داند و می‌خواهد بداند که آیا تبارشناس دیگری رکوردی

برای آن دارد یا خیر. بدون وجود بانک اطلاعاتی متمرکز چگونه می‌توان این موضوع را تشخیص داد؟

الگوریتم‌های متعددی برای حل این مسئله ارائه شدند. الگوریتم کورد (دابک و همکاران، ۲۰۰۱) را بررسی می‌کنیم. سیستم کورد از چندین کاربر تشکیل شده است که هر کدام دارای چندین رکورد ذخیره شده هستند و آمادگی دارند که قطعاتی از ایندکس را برای استفاده دیگران ذخیره کنند. هر گره کاربر، دارای یک آدرس IP است که می‌تواند با استفاده از تابع درهم سازی، hash به یک عدد m بیتی درهم سازی شود. کورد از SHA-1 برای تابع hash استفاده کرد. SHA-1 در رمز نگاری استفاده می‌شود که در فصل ۸ بحث می‌شود. در حال حاضر می‌گوییم تابعی است که یک رشته بیتی طول متغیر را به عنوان آرگومان می‌گیرد و یک عدد تصادفی ۱۶۰ بیتی را تولید می‌کند. لذا می‌توانیم هر آدرس IP را به عدد ۱۶۰ بیتی تبدیل کنیم که نامش شناسه گره است.

از نظر مفهومی، کل  $2^{160}$  شناسه گره به ترتیب صعودی در یک دایره بزرگ چیده شدند. بعضی از آن‌ها متناظر با گره‌های کاربران هستند، ولی اغلب آن‌ها این طور



(الف) مجموعه‌ای از ۲۲ شناسه گره که در دایره قرار گرفتند. گره‌های سایه‌دار متناظر با ماشین‌های واقعی‌اند. کمان‌ها، انگشتانی از گره‌های ۱، ۴ و ۱۲ را نشان می‌دهد. برچسب کمان‌ها، اندیس‌های جدول انگشتان هستند. (ب) نمونه‌هایی از جدول انگشتان.

تابع  $successor(k)$  را به عنوان شناسه اولین گره واقعی بعد از  $k$  در جهت عقربه ساعت در دایره تعریف می‌کنیم. به عنوان مثال،  $successor(6)=7$  ،  $successor(8)=12$  و  $successor(22)=27$  است.

اسامی رکوردها ( اسامی آواز، اسامی اجداد و غیره) توسط تابع  $hash$  درهم سازی شدند تا یک عدد ۱۶۰ بیتی به نام کلید تولید شود. لذا برای تبدیل  $name$  (نام اسکی رکورد) به کلید، از عبارت  $key=hash(name)$  استفاده می‌کنیم. اگر شخصی، رکورد تبارشناسی برای  $name$  داشته باشد و بخواهد آن را برای دیگران مهیا کند، ابتدا یک دوتایی متشکل از (آدرس IP و  $name$ ) می‌سازد و سپس از  $successor(hash(name))$  می‌خواهد این دوتایی را ذخیره کند. اگر برای این نام، چندین رکورد (در گره‌های مختلف) وجود داشته باشند، دوتایی آن‌ها در گره یکسانی ذخیره می‌شود. به این ترتیب، ایندکس به طور تصادفی در

گره‌ها توزیع می‌شود. برای تحمل عیب می‌توان از p تابع درهم سازی مختلف استفاده کرد تا هر دوتایی را که در p گره ذخیره کند. اما این موضوع را در این جا بررسی نمی‌کنیم.

اگر بعداً کاربری بخواهد name را جست و جو کند، آن را درهم سازی می‌کند تا کلید را به دست آورد و سپس با استفاده از successor(key) آدرس IP گره‌ای را می‌یابد که دوتایی‌های ایندکس آن را ذخیره کرده است. مرحله اول ساده است ولی مرحله دوم ساده نیست. برای این که بتوان آدرس IP گره متناظر با کلید خاصی را پیدا کرد، هر گره باید ساختمان داده‌هایی را به منظور سرپرستی ذخیره کند. یکی از این‌ها آدرس IP گره جانشین آن در دایره شناسه گره است. به عنوان مثال، در شکل ۲۴-۵، گره جانشین ۴، و گره ۱۲ گره جانشین ۷ است.

اکنون جست و جو میتواند به این صورت ادامه یابد. گره متقاضی، بسته‌ای را به گره جانشینی که حاوی آدرس IP است می‌فرستد. علاوه بر این، کلید مورد جست و جو را نیز به آن ارسال می‌کند. اکنون بسته در حلقه پخش می‌شود تا جانشینی را پیدا کند که شناسه گره به دنبال آن است. آن گره بررسی می‌کند که آیا

اطلاعاتی دارد که با کلید تطبیق کند یا خیر. اگر داشته باشد مستقیماً آن را به گره متقاضی می‌فرستد که آدرس آن را دارد.

به عنوان اولین بهینه سازی، هر گره می‌توانست آدرس‌های IP گره‌های جانشین و پیشین را نگهداری کند و در نتیجه تقاضاها هم در جهت عقربه‌های ساعت و هم در جهت خلاف عقربه‌های ساعت (بسته به این که کدام مسیر کوتاه تر است) ارسال شوند. به عنوان مثال، گره ۷ در شکل ۲۴-۵ می‌توانست برای یافتن شناسه گره ۱۰ در جهت عقربه‌های ساعت و برای یافتن شناسه گره ۳ در خلاف جهت عقربه‌های ساعت حرکت کند.

حتی حرکت در دو جهت نیز با جست و جوی خطی در سیستم نظیر به نظیر کارآمد نیست، زیرا میانگین گره‌های مورد نیاز در هر جست و جو،  $\frac{n}{2}$  است. برای افزایش سرعت جست و جو، هر گروه جدولی به نام جدول انگشت دارد. جدول انگشت  $m$  وارده دارد که اندیس آن از ۰ تا  $m-1$  است و هر کدام به یک گره واقعی اشاره می‌کنند. هر وارده دارای دو فیلد است:  $start$  و آدرس IP مربوط به  $successor(start)$ . این جدول‌ها در شکل ۲۴-۵ (ب) آمده اند. مقادیر فیلدهای مربوط به وارده  $i$  در گره  $k$  به صورت زیر محاسبه می‌شود،

( به پیمانه  $2^m$  )  $start=k+2^i$

آدرس IP مربوط به  $successor(start[i])$

توجه کنید که هر گره، آدرس‌های IP تعداد نسبتاً کمی از گره‌ها را ذخیره می‌کند و اغلب این‌ها نیز از نظر شناسه گره به هم نزدیک هستند.

با استفاده از جدول انگشت، جست و جوی key در گره k به این صورت انجام می‌شود. اگر key بین k و  $successor(k)$  باشد، آنگاه گره  $successor(k)$  حاوی اطلاعاتی راجع به key خواهد بود و جست و جو خاتمه می‌یابد. وگرنه، جدول انگشت جست و جو می‌شود تا وارده‌ای را پیدا کند که فیلد start آن نزدیک‌ترین جانشین key است. سپس تقاضا مستقیماً به آدرس IP در آن وارده جدول انگشت ارسال می‌شود تا از آن بخواهد که جست و جو را ادامه دهد. چون نزدیک‌تر به key ولی کمتر از آن است، مزیتش این است که قادر پاسخی را با تعداد اندکی از تقاضاهای دیگر ارائه کند. در واقع، چون هر جست و جو، فاصله تا مقصد را نصف می‌کند، می‌توان نشان داد میانگین جست و جو  $\log_2^n$  است.

به عنوان اولین مثال، جست و جوی  $key=3$  را در گره ۱ در نظر می‌گیریم. چون گره ۱ می‌داند گره ۳ بین آن و جانشین آن یعنی ۴ قرار دارد، گره مطلوب ۴ است و

جست و جو خاتمه می‌یابد و آدرس IP گره ۴ برگردانده می‌شود. به عنوان دومین مثال، جست و جوی  $key=14$  را در گره ۱ در نظر می‌گیریم. چون ۱۴ بین ۱ و ۴ نیست، از جدول انگشت استفاده می‌شود. نزدیک ترین گره پیشین ۱۴، گره ۹ است. لذا تقاضا به سمت آدرس IP وارده ۹، یعنی گره ۱۲ پیش می‌رود. گره ۱۲ می‌بیند که ۱۴ بین آن جانشین آن، یعنی ۱۵ قرار دارد و در نتیجه آدرس IP گره ۱۵ برگردانده می‌شود.

به عنوان سومین مثال، جست و جوی  $key=16$  را در گره ۱ در نظر بگیرید. تقاضا به گره ۱۲ ارسال می‌شود. ولی گره ۱۲ پاسخ را نمی‌داند. نزدیک ترین گره قبل از ۱۶ جست و جو می‌کند و ۱۴ را می‌یابد که آدرس IP گره ۱۵ را ارائه می‌کند. سپس تقاضا به این جا ارسال می‌شود. گره ۱۵ می‌بیند که ۱۶ بین آن و جانشین یعنی ۲۰ قرار دارد و در نتیجه آدرس IP گره ۲۰ را برمی‌گرداند و در مسیرش به گره ۱ برمی‌گردد.

چون گره‌ها در هر زمان اضافه و کم می‌شوند الگوریتم کورد می‌باست این عملیات را اداره کند. فرض می‌کنیم وقتی سیستم شروع به کارکرد به اندازه کافی کوچک بود و گره‌ها می‌توانستند مستقیما اطلاعات را مبادله کنند و اولین دایره و



جدولهای انگشت را بسازند. از این پس نیاز به روبه خودکار است وقتی گره جدیدی مثل  $r$  می‌خواهد اضافه شود، باید با یک گره موجود تماس بگیرد و از او بخواهد آدرس IP مربوط به  $\text{successor}(r)$  را برایش جست و جو کند. سپس گره جدید از  $\text{successor}(r)$  می‌خواهد گره پیشین آن را بیابد. سپس گره جدید از هر دو می‌خواهد  $r$  را بین آنها در دایره اضافه کند. بعنوان مثال اگر گره ۲۴ در شکل ۲۰ بخواهد اضافه شود از هر گره‌ای می‌خواهد  $\text{successor}(24)$  را بیابد که گره ۲۷ است. پس از گره ۲۷ گره پیشین آن را می‌پرسد که ۲۰ است. سپس حضورش را به هر دو اعلان می‌کند. ۲۰ از ۲۴ به عنوان جانشین خود و ۲۷ از ۲۴ به عنوان پیشین خود استفاده می‌کند. علاوه بر این گره ۲۷ این کلیدها را در بازه ۲۱-۲۴ تحویل می‌دهد که اکنون متعلق به ۲۴ هستند. اکنون ۲۴ اضافه شده است. اکنون بسیاری از جدولهای انگشتی نادرست‌اند. برای اصلاح آنها، هر گره یک فرایند پس زمینه را اجرا می‌کند که با فراخوانی تابع  $\text{SUCCESSOR}$ ، هر انگشت را دوباره حساب می‌کند وقتی یکی از این تقاضاها به گره جدیدی می‌رسد وارده انگشت متناظر آن نوسازی می‌شود.

اگر گره‌ای به خوبی دایره را ترک کند، کلیدهایش را به جانشین خود تحویل می‌دهد و خروج خود را به گره پیشین خود خبر می‌دهد و گره پیشین می‌تواند به گره جانشین گره خارج شده وصل شود. وقتی گره‌ای خراب می‌شود مشکل پیش می‌آید. زیرا، گره پیشین آن دیگر جانشین معتبری ندارد. برای حل این مسئله هر گره نه تنها جانشین مستقیم خود را نگه می‌دارد، بلکه تعداد  $s$  جانشین معتبری ندارد. برای حل اینکه مسئله هر گره نه تنها جانشین مستقیم خود را نگه می‌دارد، بلکه تعداد  $s$  جانشین مستقیم خود را نگه می‌دارد تا در صورتی که  $s-1$  گره متوالی با شکست مواجه شود، بتواند به گره جانشین مناسبی وصل شود.

کورد خواست سیستم فایل توزیعی و سایر کاربردها را ایجاد کند و تحقیق در این زمینه ادامه دارد.

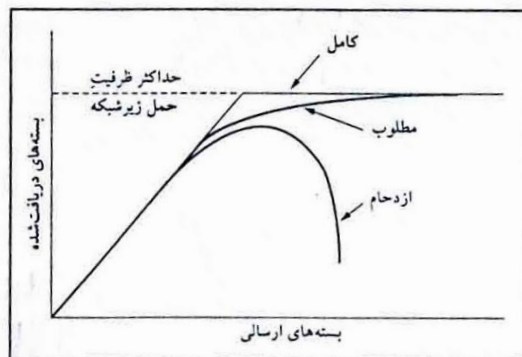
#### الگوریتم کنترل ازدحام

وقتی بسته‌های بسیار زیادی (در قسمتی از) زیر شبکه وجود داشته باشد کارایی کاهش می‌یابد این وضعیت ازدحام نام دارد. شکل ۲۱ این حالت را نشان می‌دهد. وقتی تعداد بسته‌هایی که توسط میزبانها به زیر شبکه ارسال می‌شوند، به اندازه ظرفیت حمل آن باشد، تمام آن (به جز آن‌هایی که تحت تاثیر خطای انتقال

قرار می‌گیرند) به مقصدهایشان تحویل داده می‌شوند و تعداد بسته‌های تحویل شده متناسب با تعداد ارسالی است با افزایش بی‌رویه ترافیک، مسیریابها نمی‌توانند از عهده آن برآیند، و بسته‌هایی از بین می‌روند و بر مشکل افزوده می‌شود. در ترافیک زیاد، کارایی بسیار پایین می‌آید، و تقریباً هیچ بسته‌ای تحویل داده نمی‌شود.

از سه یا چهار خط

بسی تشکیل خواهد



در ترافیک زیاد، ازدحام زیاد می‌شود و کارایی پایین می‌آید.

ازدحام به دلایل

ورودی بیابند

شد

اگر حافظه کافی برای ذخیره آنها وجود نداشته باشد، بسته‌ها از بین می‌روند

افزودن حافظه ممکن است کمک کند، اما ناگل (۱۹۸۷) دریافت که اگر مسیریابها

حافظه‌های زیادی داشته باشند، ازدحام بیشتر می‌شود، زیرا در این صورت بسته‌ها با رسیدن به مقصد را افزایش می‌دهند.

پردازنده‌های کند نیز می‌توانند ازدحام را وجود آورند. اگر پردازنده‌های مسیریابها در اجرای امور دفترداری (صف بندی بافرها، بازسازی جدولها و غیره) کند باشند، حتی اگر ظرفیت حمل خطوط بیشتر باشد، صف‌هایی تشکیل خواهد شد. خطوطی که پهنای باند آنها کم است، موجب ازدحام می‌شوند. بهبود خطوط و عدم تغییر پردازنده، یا برعکس اندکی کارساز است. اما اغلب فقط محل گلوگاه را جابه جا می‌کند. همچنین بخش بهبود یافته سیستم، فقط محل گلوگاه را به جای دیگری منتقل می‌نماید. عدم تطابق بخش‌های مختلف سیستم، مشکل اساسی را ایجاد می‌کند این مشکل تا عدم توازن قطعات مختلف سیستم ادامه می‌یابد.

بیان تفاوت کنترل ازدحام و کنترل جریان ارزشمند است همانطور که رابطه آنها نیز ظریف است. کنترل ازدحام باید برای حصول کنترل اطمینان از توانایی زیرشبکه در ترافیک عرضه شده انجام شود. این یک موضوع کلی است و رفتار

میزبانها، مسیریابها، فرایند ذخیره و ارسال در داخل مسیریابها، و تمام عواملی را که منجر به کاهش ظرفیت حمل زیر شبکه می‌شوند در بر می‌گیرد.

برعکس کنترل جریان به ترافیک نقطه به نقطه بین فرستنده و گیرنده خاصی مربوط می‌شود کارش این است که اطمینان حاصل کند که فرستنده سریع نمی‌تواند داده‌ها را سریع تر از آنچه که گیرنده می‌تواند دریافت نماید، ارسال کند. در کنترل جریان بازخوردهای مستقیمی از گیرنده به فرستنده وجود دارد تا به فرستنده بگوید کارها در طرف دیگر چگونه انجام می‌شوند.

برای پی بردن به تفاوت بین این دو مفهوم شبکه فیبر نوری با ظرفیت ۱۰۰۰ گیگابیت در ثانیه را در نظر بگیرید که در آن سوپر کامپیوتری سعی می‌کند فایلی را با سرعت 1 Gbps به کامپیوتر شخصی ارسال نماید. اگر هیچ ازدحامی وجود نداشته باشد (خود شبکه مشکلی نداشته باشد) نیاز به کنترل جریان است تا سوپر کامپیوتر را گاه‌گاهی متوقف کند تا کامپیوتر شخصی دچار مشکل نشود.

از طرف دیگر شبکه ذخیره و ارسال با خطوط 1Mbps و ۱۰۰۰ کامپیوتر بزرگ را در نظر بگیرید نیمی از آنها سعی می‌کنند فایل‌هایی را با سرعت 100kbps به

نیمی دیگر بفرستند. در اینجا مشکل این نیست که فرستنده سریع گیرنده کند را تحت فشار قرار می‌دهد بلکه کنترل ترافیک از توان شبکه بیشتر است.

علت این که کنترل ازدحام و کنترل جریان اغلب با هم اشتباه می‌شوند این است که بعضی از الگوریتم‌های کنترل ازدحام پیامهایی به منابع ارسال می‌دارند تا به آنها بگویند عمل ارسال را کندتر کنند تا شبکه از حالت ازدحام خارج شود. بنابراین میزبان در دو حالت می‌تواند پیام کندتر ارسال کن را دریافت دارد. یکی اینکه گیرنده قادر به تحمل بار نیست، دیگر اینکه شبکه نمی‌تواند این بار را تحمل نماید. در ادامه به این موضوع می‌پردازیم.

مطالعه کنترل ازدحام را با نگاهی کلی به آن شروع می‌کنیم. سپس به روش‌هایی می‌پردازیم که موجب می‌شوند، ترافیک اصلاً اتفاق نیفتد، سپس به الگوریتم‌های پویای گوناگونی می‌پردازیم تا بتوانیم پس از بروز ازدحام با آن کنار بیاییم.

#### اصول کلی کنترل ازدحام

بسیاری از مشکلات در سیستم پیچیده، مثل شبکه‌های کامپیوتری را می‌توان از دید نظیه کنترل نگرینست در این روش راه حلها به دو دسته تقسیم می‌شوند: حلقه باز و حلقه بسته، راه حل‌های حلقه باز سعی می‌کنند مشکلات را با طراحی خوب حل

کند، یعنی مطمئن می‌شوند که مشکلی بروز نخواهد کرد. وقتی سیستم راه اندازی شد و شروع به کار کرد، کامل (بی عیب) است.

ابزارهای کنترل انجام حلقه باز شامل: تصمیم‌گیری راجع به زمان پذیرش ترافیک جدید تصمیم‌گیری راجع به زمان دور انداختن بسته‌ها و دور انداختن کدام بسته‌ها و تصمیمات زمان بندی در نقاط مختلف شبکه است در تمام این موارد بدون توجه به حالت فعلی سیستم تصمیم‌گیری می‌شود.

بر عکس، راه‌های حلقه بسته بر مفهوم حلقه باز خوردی استوارند. وقتی این روش برای کنترل ازدحام به کار می‌رود دارای سه بخش است:

۱. نظارت بر سیستم جهت تشخیص زمان و مکان وقوع ازدحام.

۲. انتقال این اطلاعات به جایی که فعالیت باید انجام گیرد.

۳. تنظیم عمل سیستم برای برطرف کردن مشکل

برای نظارت بر ازدحام زیر شبکه مقیاس‌های گوناگونی به کار گرفته می‌شود از بین این‌ها، درصد بسته‌هایی که به دلیل عدم وجود فضای بافر از بین می‌روند، طول میانگین صف و انحراف معیار از تاخیر بسته از اهمیت زیادی برخوردارند. در تمام موارد افزایش نشان دهنده رشد ازدحام است.

مرحله دوم در حلقه بازخودی، انتقال اطلاعات مربوط به ازدحام به نقطه تشخیص آن به نقطه‌ای است که اعمالی راجع به آن انجام می‌پذیرد. روش بدیهی تشخیص ازدحام، ارسال بسته‌ای به منبع یا منابع ترافیک است تا آنها را از وجود مشکل آگاه کند. البته این بسته‌های اضافی، دقیقاً در موقعی که نیاز به بار بیشتری نیست (یعنی موقعی که در زیرشبکه ازدحام است) موجب افزایش بار می‌شود.

راه‌های دیگری نیز وجود دارد. بعنوان مثال در هر بسته می‌توان بیت یا فیلدی را برای مسیریاب‌ها رزرو کرد تا چنانچه ازدحام از یک حد معینی تجاوز کرد، مقدار بگیرند، وقتی مسیریابی این حالت ازدحام را تشخیص داد، فیلد تمام بسته‌های خروجی را مقدار دهی می‌کند تا این وضعیت را به آگاهی همسایه‌ها برساند.

روش دیگر این است که میزبان‌ها یا مسیریاب به طور تناوبی بسته‌های اکتشاف را ارسال کنند تا اطلاعاتی راجع به ازدحام کسب نمایند. با استفاده از این اطلاعات می‌توان ترافیک موجود در ناحیه‌هایی را که مشکل دارند برطرف کرد بعضی از ایستگاه‌های رادیوی هلیکوپترهایی دارند که بر فراز شهرها حرکت



می‌کنند تا گزارشی از زدحام را در جاده‌ها تهیه کنند، به امید اینکه شنونده‌های آنها، بسته (اتومبیل‌های) خود را در اطراف نقاط خطر هدایت کنند.

در تمام الگوهای باز خوردی، اگر میزبان‌ها اطلاعاتی راجع به ازدحام داشته باشند می‌توانند اعمالی انجام دهند که از زدحام را کاهش دهند برای صحت انجام کار، مقیاس زمان باید دقیقاً تنظیم گردد. اگر در هر زمان دو بسته در یک ردیف برسند، مسیریاب می‌گوید STOP، و هر وقت مسیریاب به مدت  $20\mu Sec$  بیکار باشد، می‌گوید GO. سیستم به شدت نوسان می‌کند و همگرا نمی‌شود. از طرف دیگر، اگر قبل از انجام هر کاری ۳۰ دقیقه منتظر بماند. عکس العمل راهکار کنترل ازدحام آنقدر کند است که عملاً به کار نمی‌آید. برای عملکرد خوب، تا حدی به تعدیل نیازمندیم، اما به دست آوردن ثابت زمانی، مسئله کوچکی نیست.

الگوریتم‌های کنترل ازدحام متعددی شناخته شده‌اند. برای سازماندهی معقول آنها، یانگ وردی ۱۹۹۵ الگوریتم‌های کنترل شده ازدحام را طبقه بندی کردند. آنها ابتدا الگوریتم‌ها را بر اساس حلقه بسته یا حلقه باز تقسیم بندی نمودند، سپس الگوریتم‌های حلقه باز را بر اساس این که در منبع عمل می‌کنند یا در مقصد تقسیم بندی، الگوریتم‌های حلقه بسته نیز به دو دسته تقسیم شدند. با خوردی صریح و

باز خوردی ضمنی در الگوریتمهای ضمنی، منبع با مشاهدات محلی، مثل زمان مورد نیاز برای برگشت اعلام وصول، به وجود ازدحام پی می‌برد.

وقتی میزان بار در بخشی از سیستم، به طور موقت بیش از ظرفیت خدمات دهی منابع باشد، ازدحام پیش می‌آید. دو راه حل به نظر می‌رسد: افزایش منابع یا کاهش بار. به عنوان مثال ممکن است زیر شبکه با استفاده از خطوط تلفن شماره گیری، پهنای باند بین بعضی از نقاط را موقتاً افزایش دهد. در سیستمهای ماهواره‌ای، افزایش قدرت انتقال، موجب افزایش پهنای باند می‌شود. اگر به جای استفاده از بهترین مسیر برای انتقال ترافیک آن را بین مسیرهای چند گانه تقسیم کنیم، پهنای باند افزایش می‌یابد. مسیرهای پشتیبان که برای برطرف کردن عیب سیستم به کار می‌روند. در صورت وجود ازدحام فعال می‌شوند تا ظرفیت بیشتری در اختیار شبکه قرار گیرد.

گاهی نمی‌توان ظرفیت را به اندازه کافی افزایش داد در این صورت تنها راه حل کاهش بار است. راههای متعددی برای کاهش بار وجود دارد از جمله عدم ارائه خدمات به بعضی از کاربران کاهش خدمات تمام یا بعضی از کاربران، و یا زمانبندی تقاضا کاربران به طور متناوب.

بعضی از این روشها که آنها را به طور مختصر تشریح می‌کنیم، در مدارهای مجازی بهتر عمل می‌کنند. در زیر شبکه‌هایی که در داخل خود از مدارهای مجازی استفاده می‌کنند این روشها را می‌توان در لایه شبکه به کار برد. در شبکه‌های داده گرام نمی‌توان آنها را در اتصالات لایه انتقال به کار برد. در این فصل به کاربرد آنها در لایه شبکه می‌پردازیم.

#### سیاست‌های جلوگیری از ازدحام

مطالعه روشهای کنترل ازدحام را با نگاهی به سیستم‌های حلقه باز شروع می‌کنیم. این سیستم‌ها طوری طراحی شدند که از بروز ازدحام جلوگیری کنند. برای این منظور از سیاست‌های مختلفی استفاده می‌شود. در شکل ۲۱ سیاست‌های مختلفی از لایه‌های پیونده داده‌ها، شبکه و انتقال را مشاهده

سیاست‌ها	لایه
<ul style="list-style-type: none"> <li>• سیاست انتقال مجدد</li> <li>• سیاست ذخیره خارج از ترتیب</li> <li>• سیاست اعلام وصول</li> <li>• سیاست کنترل جریان</li> <li>• تعیین مهلت زمانی</li> </ul>	انتقال
<ul style="list-style-type: none"> <li>• مدار مجازی در مقابل داده گرام در زیر شبکه</li> <li>• سیاست صفبندی بسته و خدمات</li> <li>• سیاست دورانداختن بسته</li> <li>• الگوریتم مسیریابی</li> <li>• مدیریت طول عمر بسته</li> </ul>	شبکه
<ul style="list-style-type: none"> <li>• سیاست انتقال دوباره</li> <li>• سیاست ذخیره خارج از ترتیب</li> <li>• سیاست اعلام وصل</li> <li>• سیاست کنترل جریان</li> </ul>	پیوند داده‌ها

می‌کنیم که بر ازدحام موثرند (جین ، ۱۹۹۰).

ابتدا از لایه پیوند داده‌ها شروع می‌کنیم، سیاست انتقال مجدد، به سرعت تمام مهلت زمانی فرستنده و آنچه که در این مدت انتقال دده است بستگی دارد فرستنده سریعی که مهلتش به زودی به اتمام مهلت زمانی فرستنده و آنچه که در این مدت انتقال داده است بستگی دارد فرستنده سریعی که مهلتش به زودی به اتمام می‌رسد و به کمک قرارداد عقب گرد نتایی تمام بسته‌ها را دوباره ارسال می‌نماید، نسبت به فرستنده کندی که از تکرار انتخابی استفاده می‌کند بار کمتری را به سیستم تحمیل می‌نماید. سیاستی مشابه با آن سیاست بافر کردن است. اگر گیرنده‌ها تمام بسته‌های خارج از ترتیب را دور می‌اندازند این بسته‌ها باید دوباره ارسال شوند و این کار بار اضافی را به شبکه تحمیل می‌کند. برای کنترل ازدحام تکرار انتخابی بهتر از عقب گرد نتایی است.

سیاست اعلام وصول نیز بر ازدحام موثر است. اگر هر بسته فوراً اعلام وصول شود، بسته‌های اعلام وصول ترافیک بیشتری را به وجود می‌آورند. اگر اعلام وصول‌ها به طور قاچاقی با ترافیک برگشتی ارسال شوند، ممکن است منجر به

انقضای مهلت و انتقال‌های مجدد شود. الگوی کنترل جریان فشرده (مثلا پنجره کوچک) از سرعت داده‌ها می‌کاهد و موجب افزایش ازدحام می‌شود.

در لایه شبکه، انتخاب بین مدارهای مجازی و داده گرام‌ها بر ازدحام تاثیر دارد. زیرا بسیاری از الگوریتم‌های کنترل ازدحام مربوط می‌شود که: آیا مسیریاب‌ها به ازای هر خط ورودی یک صف دارند، آیا به ازای هر خط خروجی یک صف دارند، یا هر دو حالت را دارند. به ترتیب پردازش بسته‌ها (مثل نوبتی یا اولویت) نیز مربوط می‌شود. در سیاست دور انداختن بسته‌ها، مشخص می‌شود که اگر حافظه کافی وجود نداشته باشد، چه بسته‌هایی باید دور انداخته شوند. سیاست خوب می‌تواند از ازدحام بکاهد و سیاست بد می‌تواند بر ازدحام بیفزاید.

الگوریتم مسیریابی می‌تواند با توزیع ترافیک در همه خطوط از ازدحام جلوگیری کند ولی اگر از الگوریتم بدی استفاده شود، ترافیک بیشتری را به خطی که اکنون دچار ازدحام است ارسال می‌کند. سرانجام، مدیریت دوران زندگی بسته، با مدت حضور بسته قبل از از بین رفتن سر و کار دارد. اگر این مدت طولانی باشد بسته‌های از بین رفته می‌توانند کار را مختل کنند اما اگر بسیار کوتاه باشند،

ممکن است مهلت زمانی بسته، قبل از رسیدن به مقصد به اتمام برسد و بسته دوباره ارسال شود.

در لایه انتقال مشکلی مانند لایه پیوند داده‌ها به وجود می‌آید، اما در مجموع تعیین مهلت مشکل‌تر است، زیرا قابلیت پیش بینی زمان انتقال از طریق شبکه نسبت به زمان انتقال از طریق سیم بن دو مسیریاب، کمتر است. اگر کاهش بسیار کوتاه باشد، بسته‌های زیادی به طور غیر ضروری ارسال می‌شوند. اگر بسیار طولانی باشد، ازدحام کاهش می‌یابد، اما وقتی بسته‌ای از بین می‌رود، زمان پاسخ آزار دهنده است.

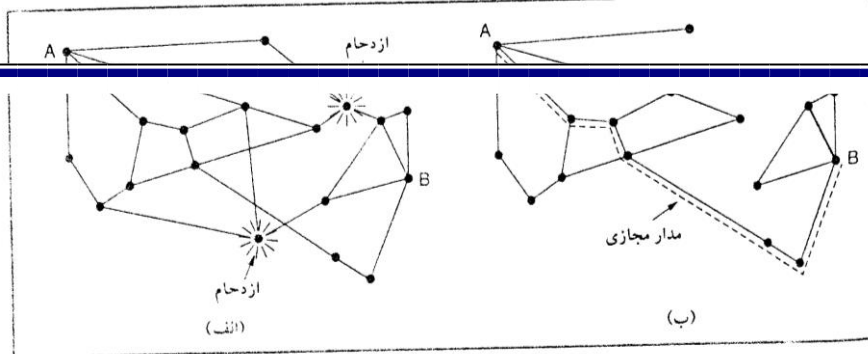
#### کنترل ازدحام در زیرشبکه‌های مدار مجازی

روشهایی که برای کنترل ازدحام مطرح شدند، از نو حلقه باز هستند، یعنی سعی می‌کنند از وقوع ازدحام جلوگیری کنند. در این بخش روشهایی را برای کنترل پویای ازدحام در زیرشبکه‌های مدار مجازی بررسی می‌کنیم. در دو بخش بعدی تکنیک‌هایی را که در زیر شبکه به کار می‌روند مورد بحث قرار می‌دهیم.

تکنیکی که به طور گسترده برای جلوگیری از افزایش ازدحام موجود به کار می‌رود، کنترل پذیرش نام دارد. ایده این تکنیک ساده است: وقتی ازدحام به وجود

آمد، مدارهای مجازی دیگری برقرار نمی‌شوند تا مشکل برطرف شود. بنابراین، تلاش‌های مربوط به برقراری اتصال لایه انتقال با شکست مواجه می‌شود و ورود افراد بیشتر وضع را بدتر می‌کند با این که این روش ظرافت خاصی ندارد، اجرای آن آسان است. در سیستم تلفن وقتی بار زیادی را به راه گیزن وارد می‌شود عمل کنترل پذیرش را انجام می‌دهد (با عدم پذیرش شماره تلفن).

روش دیگر این است که مدارهای مجازی جدیدی بتوانند فعال شوند، اما در جاهایی باید برقرار شوند که ازدحام موجود را رفع کنند. بعنوان مثال زیرشبکه شکل ۲۲ (الف) را در نظر بگیرید، در آن دو مسیریاب دچار ازدحام شده اند. فرض کنید میزبان متصل به مسیریاب A می‌خواهد با میزبان متصل به مسیریاب B اتصالی برقرار کند. این اتصال از طریق یکی از مسیریاب‌های دچار ازدحام عبور می‌کند. برای پرهیز از این وضعیت می‌توان زیر شبکه را مانند شکل ۲۲ (ب) رسم کرد. در این شکل مسیریاب‌های دچار ازدحام و کلیه خطوط آنها حذف شده اند. نقطه چین مسیری را برای مداری نشان می‌دهد که از مسیریاب‌های دچار ازدحام نمی‌گذرد.



(الف) زیرشبکه دچار ازدحام. (ب) رسم مجدد زیرشبکه که در آن ازدحام حذف شد و مدار مجازی از A به B برقرار شده است.

راهبرد دیگر مرتبط با مدارهای مجازی این است که وقتی مدار مجازی فعال شد مذاکراتی بین میزبان و زیرشبکه به عمل آید تا بر سر میزان و شکل ترافیک خدمات مورد نیاز و سایر پارامترها توافق به عمل آید بر اساس این توافق زیرشبکه منابع مربوط به مسر را به هنگام برقراری مدار رزور می‌کند. این منابع می‌تواند شامل جدول و فضای بافر در مسیریاب‌ها و پهنای باند در خطوط باشد. در این روش ازدحام در مدارهای جدید به وجود نمی‌آید، زیرا تضمین می‌شود که تمام منابع ضروری مهیا باشند.

این نوع رزور سازی یا می‌تواند همیشه بعنوان یک عملیات استاندارد انجام شود و یا در صورت وجود ازدحام انجام گیرد. اگر ای کار به طور همیشگی انجام شود منابع را به هدر می‌دهد. اگر شش مدار مجاز که می‌تواند از 10 Mbps استفاده کند. همگی از یک خط 6Mbps فیزیکی عبور کنند، باید خط را به عنوان خط پر علامت گذاری کرد. ولی احتمال اینکه هر شش مدار مجازی همزمان عمل انتقال را



انجام دهند کم است. در نتیجه هزینه کنترل ازدحام پهنای باند مصرف نشده است.

کنترل ازدحام در زیرشبکه‌های داده گرام

اکنون روشهایی را بررسی می‌کنیم که می‌توانند در زیرشبکه‌های داده گرام و مدار مجازی به کار روند. هر مسریاب می‌تواند بهره وری خطوط خروجی خود و سایر منابع را کنترل کند. بعنوان مثال می‌تواند متغیری مثل  $u$  را به هر خط اختصاص دهد که مقدارش بین  $0/0$  و  $1/0$  است و بهره وری فعلی خط را نشان می‌دهد برای برآورد دقیقی از  $u$  نمونه‌ای از بهره وری لحظه‌ای خط،  $f$  (یا  $0$  یا  $1$ ) را می‌توان به طور تناوبی ایجاد کرد و  $u$  به صورت زیر محاسبه گردد:

$$u_{new} = au_{old} + (1-a)f$$

که در آن،  $a$  سرعتی است که مسریاب وضعیت فعلی خود را از یاد می‌برد. وقتی  $u$  از حد آستانه‌ای تجاوز می‌کند خط خروجی حالت اخطار را اعلام می‌دارد تمام بسته‌های ورودی جدید چک می‌شوند تا مشخص شود که آیا خط خروجی آن

در حالت اخطار قرار دارد یا خیر. اگر باشد اعمال مختلفی ممکن است صورت گیرد که در ادامه بحث می‌شود.

### بیت اخطار

معماری قدیمی DECNET حالت اخطار را از طریق مقدار دادن به بیت خاصی در سرآیند بسته اعلان می‌کرد. وقتی بسته به مقصد رسید، نهاد انتقال، این بیت را در اعلام وصول بعدی کپی و به منبع ارسال می‌کرد سپس منبع ترافیک را کاهش می‌داد.

تا زمانی که مسیریاب در حالت اخطار بود به مقدار دادن بیت اخطار ادامه می‌داد معنایش‌ای بود که منبع آماده دریافت اعلام وصولها است. منبع کسری از اعلام وصولها را با این بیت نظارت می‌کرد و سرعت انتقال را بر اساس آن تنظیم می‌نمود. تا زمانی که جریان بیت‌های اخطار ادامه داشت، منبع در حال کاهش سرعت انتقال بود. وقتی که به حد معینی کاهش یافت، به سرعت انتقال می‌افزود. توجه کنید که چون هر مسیریاب موجود در مسیر می‌توانست بیت اخطار را مقدار دهد، ترافیک فقط وقتی اضافه می‌شود که هیچ مسیریابی دچار مشکش نباشد.

### بسته‌های چوک

الگوریتم قبلی برای کنترل ازدحام، ظریف است. به طور غیر مستقیم به منبع می گوید که سرعتش را کم کند. چرا مستقیماً به آن نگوید؟ در این روش، مسیریاب نیاز دارد بسته‌ای به نام بسته چوک را به میزبان منبع برگرداند و به او بگوید که مقصد در بسته پیدا شد. بسته اصلی برچسب دار می‌شود ( یک بیت از سرآیند مقدار می‌گیرد) و در نتیجه در طول مسیر، بسته‌های چوک دیگری را تولید نمی‌کند و به طور عادی به راهش ادامه می‌دهد.

وقتی میزبان منبع، بسته چوک را دریافت می‌کند لازم است ترافیک ارسالی به منبع را  $\times$  درصد کاهش دهد. از آنجا که ممکن است سایر بسته‌های ارسالی به این مقصد در راه باشند و بسته‌های چوک دیگری تولید کنند، میزبان باید در یک فاصله زمانی ثابت، از این بسته‌های چوک جلوگیری نماید. پس از سپری شدنای دوره زمانی میزبان در فاصله زمانی دیگر نیز مواظب بسته‌های چوک است. اگر یک بسته چوک برسد خط هنوز دچار ازدحام است، میزبان از جریان بسته‌ها می‌کاهد و مجدداً شروع به از بین بردن بسته‌های چوک می‌کند. اگر درای دوره هیچ بسته چوکی نرسد، امکان دارد میزبان جریان را بیشتر نماید. بازخورد ضمنی این

قرارداد می‌تواند تا بند نیامدن جریان، به جلوگیری از ازدحام کمک نماید، مگر اینکه مشکلی پیش بیاید.

میزبانها می‌توانند با تنظیم پارامترهای خود مثل اندازه پنجره، ترافیک را کاهش دهند، اولین بسته چوک موجب می‌شود تا سرعت به اندازه ۵۰٪ سرعت قبلی کاهش یابد و بسته چوک بعدی موجب ۲۵٪ کاهش می‌شود و غیره. افزایشها به تدریج انجام می‌شود تا از وقوع سریع ازدحام جلوگیری به عمل آید.

شکل‌های گوناگونی از این الگوریتم پیشنهاد شد. در یکی از آنها مسیریابها می‌توانند چندین حد آستانه‌ای را نگهداری کنند. بر اساس این که از کدام حد آستانه‌ای تجاوز شده باشد، بسته چوک می‌تواند حاوی اخطار ساده، اخطار جدی یا اولتیماتوم باشد.

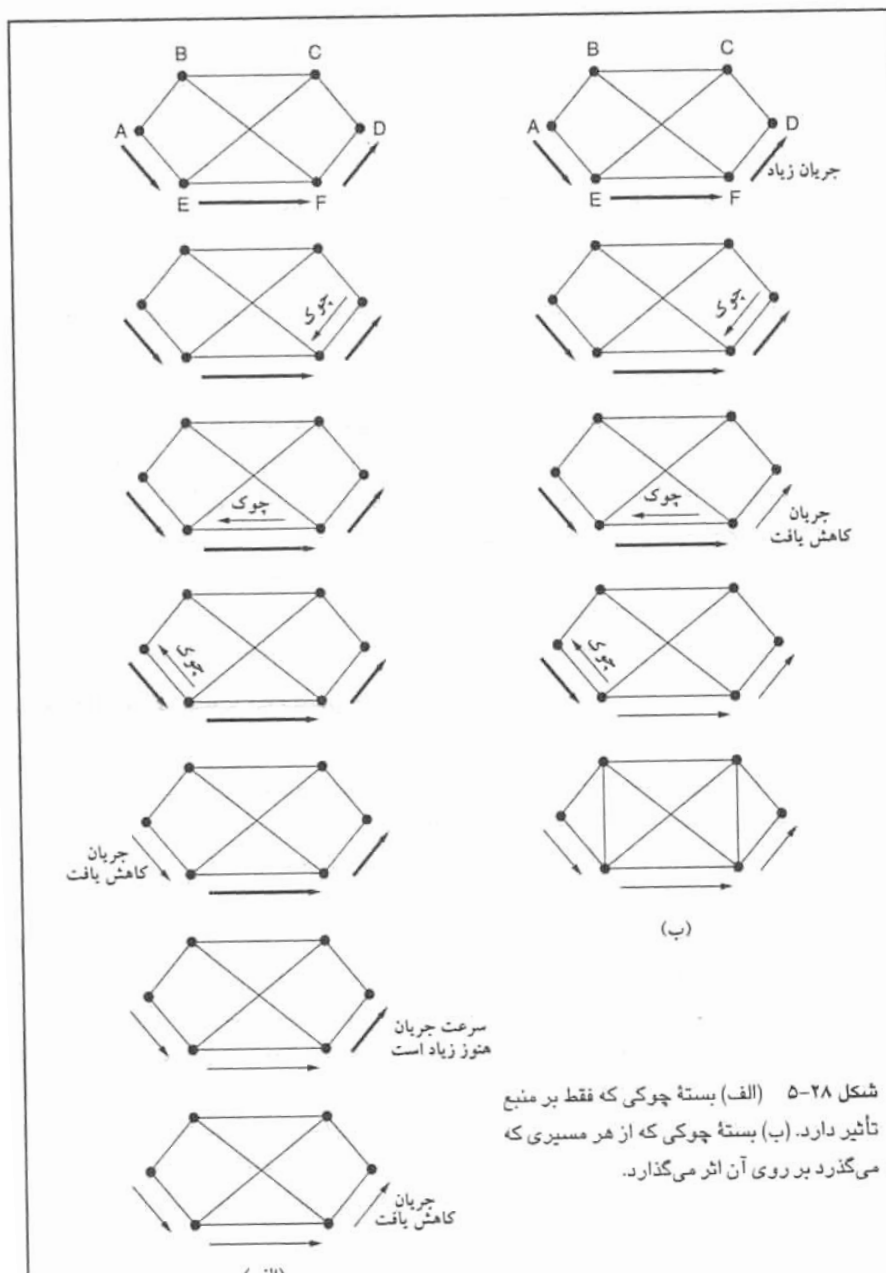
شکل دیگر، استفاده از طول صف یا بهره‌وری بافر به جای بهره‌وری خط، مانند سیگنال رها کننده است. البته، وزن دهی توای مشابه آنچه که با II به کار می‌رفت، می‌تواند با این مقیاس نیز به کار گرفته شود.

بسته‌های چوک مسیر به مسیر

در سرعت‌های زیاد و مسافت‌های طولانی، ارسال بسته چوک به میزبانهای منبع کارایی خوبی ندارد، زیرا واکنش کند است. بعنوان مثال میزبانی را در سان‌فرانسیسکو (مسیریاب A در شکل ۲۳) در نظر بگیرید که ترافیکی را به میزبانی در نیویورک (مسیریاب D در شکل ۲۳) با سرعت 155 Mbps می‌فرستد. اگر بافر میزبان نیویورک شروع به پر شدن کند، ۳۰ میلی ثانیه طول می‌کشد تا بسته چوک به سان‌فرانسیسکو برسد و به آن بگوید که کندتر عمل کند. انتشار بسته چوک در شکل ۲۳-الف به صورت مراحل دوم، سوم، چهارم نشان داده شده است. در آن ۳۰ میلی ثانیه‌ها، ۴/۶ مگابیت دیگر ارسال خواهند شد. حتی اگر میزبان در سان‌فرانسیسکو کاملاً از کار افتد، ۴/۶ مگابیت موجود در مجرا، به انتقال ادامه می‌دهد تا تمام شود. فقط در نمودار هفتم در شکل ۲۳-الف مسیریاب نیویورک جریان کندتری را مشاهده می‌کند.

روش دیگر این است که بسته چوک بر هر مسیر انتقالی که از آن عبور می‌کند، موثر باشد (مانند شکل ۳۲-ب) در اینجا به محض اینکه بسته چوک به F می‌رسد، F باید میزان جریان به D را کاهش دهد برای انجام این کار F باید فضای بافر بیشتری را به جریان اختصاص دهد. زیرا منبع با شدت تمام در حال ارسال است

اما مانند درمان دردسر در آگهی تجارتي تلویزیون، D را تسکین می‌دهد. در مرحله بعدی بسته چوک به E است ولی F را فوراً تسکین می‌دهد. سرانجام بسته چوک به A می‌رسد و جریان به تدریج کاهش می‌یابد.



حاصل کار الگوی مسیر به مسیر این است که فوراً در نقطه ازدحام تسکین ایجاد می‌شود و هزینه اش این است که فضای بافر بیشتری باید اختصاص داده شود در این روش ازدحام، بدون آسیب دیدن بسته ای، از بین می‌رود. این ایده با تفضیل بیشتر و نتایج شبیه سازی در (میشرا و کاناکیا، ۱۹۹۲) تشریح شده است.

### تخلیه بار

وقتی هیچکدام از روشهای فوق منجر به رفع ازدحام نشوند، مسیریابها می‌توانند به روش تخلیه بار از ازدحام خلاص شوند. تخلیه بار روش جالبی است و به این صورت که وقتی مسیریابها مورد هجوم بسته‌هایی قرار گرفتند که نمی‌توانند از شر آنها خلاص شوند، آنها را دور می‌اندازند. این اصطلاح از مقوله تولید انرژی نیروی الکتریسیته گرفته شد و به معنای خاموش کردن عمدی بعضی از مناطق است تا از کارافتادگی کل سیستم در روزهای داغ تابستان که نیاز به برق به مراتب از تولید بالاتر است جلوگیری نماید.

مسیریابی که در بسته‌های زیادی غرق شده است بسته‌ها را به طور تصادفی انتخاب و حذف می‌کند. اما بهتر از این می‌تواند کار کند. بسته‌هایی که باید حذف

شوند به برنامه کاربردی که در آن اجرا می‌شوند، بستگی دارد. برای انتقال فایل بسته قدیمی ارزشمندتر از بسته جدید است، زیرا حذف بسته ۶ و نگهداری بسته از بین ۱۰ بسته شکافی در گیرنده به وجود می‌آورد و موجب می‌شود بسته‌های ۶ تا ۱۰ دوباره ارسال شوند. این در صورتی است که گیرنده معمولاً بسته‌های خارج از ترتیب را به دور بریزد. در فایل ۱۲ بسته‌ای حذف بسته ۶ ممکن است منجر به ارسال مجدد ۷ تا ۱۲ شود، در حالی که حذف ۱۰ ممکن است نیاز به ارسال دوباره ۱۰ تا ۱۲ باشد. بر عکس برای چند رسانه‌ای بسته جدید مهمتر از بسته قدیمی است. سیاست قبلی (قدیمی بهتر از جدید است) معمولاً شراب نام دارد و سیاست بعدی (جدید بهتر از قدیم است) معمولاً شیر نام دارد.

اگر بخواهیم از نظر هوش یک گام بیشتر از این پیش برویم، نیازمند همکاری فرستنده‌ها هستیم، در بسیاری از کاربردها بعضی از بسته‌ها مهمتر از بسته‌های دیگر اند. بعنوان مثال بعضی از الگوریتم‌های مربوط به فشرده سازی تصویر، به طور دوره‌ای قاب کاملی را انتقال می‌دهند و سپس قاب‌های بعدی را بعنوان تفاوت‌هایی از آخرین قابل کامل ارسال می‌کنند و در این حالت حذف بسته‌ای که بخشی از تفاوت است، نسبت به حذف بسته‌ای که بخشی از قاب کامل است ارجح



است. بعنوان مثالی دیگر انتقال سدی حاوی متن اسکی و تصویر را در نظر بگیرید، خطر از بین رفتن خطی از پیکس در بعضی از تصاویر، کمتر از بین رفتن خطی از متن است

برای پیاده سازی سیاست حذف هوشمند، کاربردها باید رده ۴ اولویت بسته‌های خود را تعیین کنند تا میزان اهمیت آنها را نشان دهند. در این صورت مسیریابها برای حذف بسته‌ها ابتدا آنها را از رده پایین‌تر حذف می‌کنند و سپس به رده‌های بعدی می‌روند. البته مگر در حالتی که انگیزه‌های خاصی وجود داشته باشد که بسته‌ها را غیر از **EVER DISCARD , VERY IMPORTANT-NEVER** تعیین کند، که هیچکس این کار را انجام نمی‌دهد.

انگیزه باید اقتصادی باشد، به طوری که ارسال بسته‌هایی با اولویت پایین، ارزان‌تر از ارسال بسته‌هایی با اولویت بالا باشد. از طرف دیگر فرستنده ممکن است اجازه داشته باشد بسته‌هایی با اولویت بالا را در شرایط بار کم ارسال کند. اما وقتی بار افزایش می‌یابد بسته‌ها حذف می‌شوند و کاربران تشویق به توقف ارسال آنها می‌شوند.

روش دیگر این است که وقتی مداری مجاری فعال شد به میزبانها اجازه داده شوند یا را از محدودیت تعیین شده در توافق نامه فرا نهد (مثلا از پهنای باند بیشتر از حد مجاز استفاده کنند) اما مشروط به این که ترافیک اضافی بعنوان اولویت پایین مشخص شود. این راهبرد ایده بدی نیست. زیرا از منابع بیکار بهره استفاده می‌کند و علتش این است که میبانه می‌توانند تا زمانی که کسی از آن منابع استفاده نمی‌کند آنها را به کار گیرند در عین حال وقتی اوضاع سخت می‌شود، برای این کاربران حق ویژه‌ای ایجاد نمی‌کند.

#### تشخیص زودرس تصادفی

بدیهی است که اگر به محض کشف ازدحام به درمان آن بپردازیم، بهتر این است که اجازه دهیم اوضاع بدتر شود و سپس به آن بپردازیم این مشاهده منجر به این ایده می‌شود که قبل از پر شدن کل بافر، بسته‌ها از بین بروند. الگوریتم معروف برای انجام این کار RED (تشخیص زودرس تصادفی) نام دارد. در بعضی از قراردادهای انتقال (از جمله TCP)، پاسخ به بسته مفقود شده این است که منبع کندتر شود. منطق این عمل این است که TCP برای شبکه‌های سیمی طراحی شد و شبکه‌های سیمی قابل اعتماداند. لذا بسته‌ها اغلب به علت پر شدن

بافر از بین می‌روند تا در اثر خطاهای انتقال، این حقیقت می‌تواند به کاهش ازدحام کمک کند.

وقتی می‌گوییم بسته‌ها را تا بدتر نشدن اوضاع از ببین ببریم، علتش این است که زمان لازم برای بهتر کرد اوضاع را داریم. برای تعیین زمان شروع به از بین بردن بسته‌ها، مسیریاب‌ها میانگین طول صف‌های خود را نگهداری می‌کنند. وقتی میانگین طول صف در خطی از حد آستانه‌ای تجاوز کرد، خط دچار ازدحام می‌شود و عمل حذف بسته باید شروع شود.

چون مسیریاب نمی‌تواند بگوید کدام بسته بیشترین مشکل را ایجاد کرد، انتخاب تصادفی بسته‌ای از آن صف، مناسب به نظر می‌رسد.

مسیریاب چگونه باید حذف بسته را به اطلاع منبع برساند؟ یک روش ارسال بسته چوک به آن این است که شرح آن گذشت، مشکل آن روش این است که بار بیشتری را در شبکه‌ای که دچار ازدحام شده است ارسال می‌کند راهبرد دیگر این است که بسته انتخاب شده حذف شود و هیچ گزارشی به منبع ارسال نگردد. منبع سرانجام متوجه عدم اعلام وصول می‌شود و از عمل حذف بسته باخبر می‌گردد. چون می‌داند که بسته‌های حذف شده منجر به ازدحام شده اند، عمل

ارسال را کند میکند. این بازخورد ضمنی فقط وقتی کار می‌کند که منبع با از دست دادن بسته‌ها سرعت انتقال خود را کاهش دهد در شبکه‌های بی سیم که از دست دادن بسته‌ها بیشتر به خاطر اختلال در پیوند هوایی است نمی‌توان از این روش استفاده کرد.

### کنترل لرزش

برای کاربردهایی مثل صوت و ویدیو، تا زمانی که زمان انتقال ثابت است مهم نیست که بسته‌ای پس از ۲۰ میلی ثانیه برسد یا ۳۰ میلی ثانیه نوسان در زمانهای رسیدن بسته، لرزش نام دارد. لرزش زیاد مثلاً وقتی که بسته‌ای ۲۰ میلی

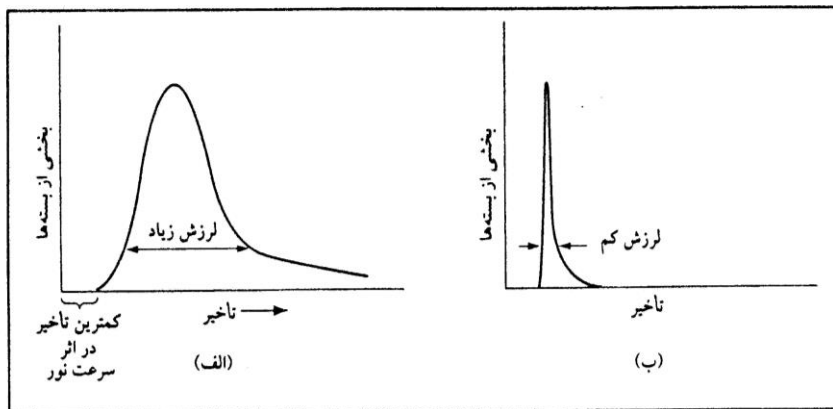
ثانیه

از

باید

باب

ت.



در

۴/۵

امک

در

(الف) لرزش زیاد. (ب) لرزش کم.

لرزش را می‌توان با محاسبه زمان انتقال برای هر جهش در طول مسیر، محدود کرد. وقتی بسته‌ای به مسیریاب می‌رسد، مسیریاب آن را بررسی می‌کند تا تشخیص دهد بسته چه مدت از زمان بندی عقب یا جلو است. این اطلاعات در بسته ذخیره می‌شود و در هر جهش اصلاح می‌گردد. اگر بسته نسبت به زمان بندی جلو باشد، نگه داشته می‌شود تا به زمانبندی برسد. اگر از زمان بندی عقب است، مسیریاب سعی می‌کند سریعاً آن را به خروجی بفرستد.

در واقع الگوریتمی که تعیین می‌کند کدام بسته‌ها برای یک خط خروجی رقابت می‌کنند، می‌تواند بسته‌ای را انتخاب کن که از بسته‌های دیگر نسبت به زمان بندی عقب تر است. در این روش بسته‌هایی که از زمان بندی جو هستند، کندتر می‌شود و بسته‌هایی که از زمان بندی عقب هستند، تسریع می‌شوند. در هر دو حالت از میزان لرزش کاسته می‌شود.

در بعضی از کاربردها مثل ویدیوی درخواستی می‌توان لرزش را حذف کرد برای این کار باید بسته‌ها در سمت گیرنده بافر شوند و داده‌ها برای نمایش در زمان بی‌درنگ از بافر برداشته شوند نه از شبکه اما برای کاربردهای دیگر به خصوص در کاربردهایی که نیاز به تعامل بی‌درنگ بین افراد است مثل تلفن اینترنت و

کنفرانس ویدیویی تاخیر ناشی از بافر کردن قابل قبول نیست. کنترل ازدحام موضوع تحقیق است.

### کیفیت خدمات

تکنیکهایی که قبلاً بحث شدند برای کاهش ازدحام و بهبود کارایی شبکه طراحی شدند اما با رشد شبکه‌های چند رسانه‌ای این سنجش‌های موردی کافی نیستند نیاز به تلاشهایی برای تضمین کیفیت خدمات و طراحی قراردادهایی برای این کار است.

### مسیر یابی منبع دینامیک (۱)

پروتکل DSR یک منبع مسیریاب روی پروتکل درخواستی است دو فاز اصلی برای پروتکل وجود دارد : کشف مسیر و نگهداری مسیر کلیدمفتاوتی بین DSR و دیگر پروتکل ها در اطلاعات مسیر یابی وجود دارد که در PACH HADER شامل می شود نظر به اینکه اطلاعات مسیر یابی شامل PACH HADER می شود پس NODE گرهما میانجی برای نگهداری اطلاعات مسیر یابی بی نیاز نیست یک گرهما میانجی ممکن است تمایل به ضبط اطلاعات مسیریابی در جداول خودش داشته باشد که به اصلاح کردن اجرا می پردازند اما

ان اجباری نیست دیگر ترکیب DSR وجود دارد که لینک های نامتقارن را حمایت می کنند همانند یک پاسخ مسیر که می تواند به درونیک بسته درخواستی مسیر بر پشت سوار شود. DSR برای شبکه های کوچک و متوسط متناسب است مانند OVERHEAD خود که می تواند تمام راههای پایین به صفر رسیده رسیده را قیاس کند OVERHEAD بطور معنی داری برای شبکه هایی با دیامترهای بزرگتر HOP افزایش خواهد یافت مثل اطلاعات مسیریابی بیشتر که شامل packet header ها خواهند شد.

مسیر یابی دینامیک و عملکرد موازنه در شبکه های ارتباط از راه دور

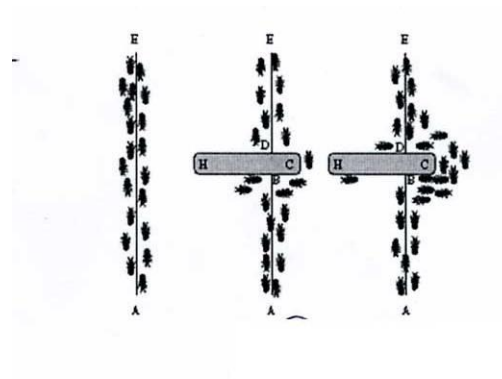
مشکل مسیر یابی

یافتن جداول مسیر یابی روی هر node شبکه جهت هایی را به مبنای پیامهای آمده روی مقصد مربوطه شان در دستور به بهینه سازی قیمت و توازن عملکرد شبکه می دهند.

یافتن انبوهی از کوتاهترین راهها

بوسیله استفاده مورچه های نشان دار pheromone مجموعا کوتاهترین راه پیدا می شود الگوریتمهای جدید مانند یک توسعه به الگوریتمهای مسیر clossicd

پیشرفته شده است آنها به ایده هایی بردار مسیر یابی مقصد درون خطی غیر  
 هنرمان با وضع مسیر یابی لینک انطباق ملحق می شوند تخمین هایی وضعیت  
 جریان ترافیک و ارزشهای لینک بوسیله ارسال نمایندگان مسیریابی در شبکه  
 اندازه گیری شده که با بسته های اطلاعاتی منظم ترکیب گشته و track ارزش  
 های قیمت های مواجه شده در طی سفرشان نگهداری می شوند.



جداول مسیریابی بنابراین به طور منظم جدید و بروز می شوند که اطلاعات  
 بدون برخی کنترل های مرکزی ، دانش توپولوژی شبکه کامل نمی شود . دو  
 الگوریتم جدید در اینجا پیشنهاد کرده می شود. اول اینکه براساس roud trip  
 نمایندگان مسیر قراردادده می شود که جداول مسیر یابی بوسیله رد گم کردن



مسیر شان بعد از جستجوی مقصد جدید به روز می شود جدید به روز می شود.  
دوم اینکه روی نمایندگان تکیه میشود که جداول مسیریابی را به طور مستقیم  
همچون آنها نسب به هدفشان جنبش دارند، جدید و به روزی می شوند و یک  
برنامه عملی موثر به سروکار داشتن با تماسهای نامتقارن، پیشنهاد کرده می  
شود.



مسیر یابی نیاز به مسیر یابی :

یک شبکه یک مجموعه به هم پیوسته از **node** (گرهها) است (با یک سیستم وسیع  
شبکه با عناوین منحصر به فرد باشد آنجا هیچ نیازی برای یک تماس مستقیم  
بین دو **node** وجود ندارد برخی **node** های داده شده یک تماس مستقیم به یک

یا تعداد بیشتری node خواهند داشت اگر یک node به یک addressed به طور مستقیم به یک node تماس رسیده به سادگی به آن در نرم افزار driver لینک اختصاصی ، عبور خواهد کرد. اگر یک node یک addressed به یک node برسد که از آن هیچ تماس مستقیمی ندارد و باید مشکل مسیر یابی تعیین نشده را حل کند که node ان را می فرستد.

### Forward در جستجوی الگوریتم

همچنین مشهور به الگوریتم Forward Dijkstra;s در جستجوی الگوریتم می تواند بنابراین بطور غیر رسمی توضیح داده شود.

$c(N,i)$  ، حداقل قیمت مسیر از I به n است

$(N,i)$  ۱ قیمت لینک از I به n است برای هر node است برای هر (گره)

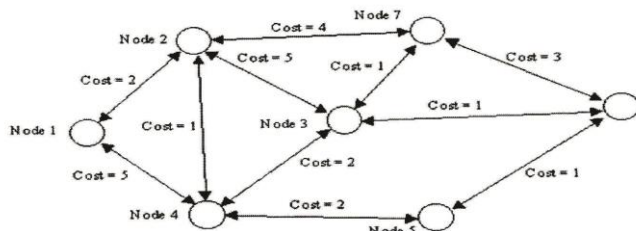
برای سایر node ها مجموع n است برای هر l ما دوباره تکرار می شود.

پیدا کردن یک node گره w که هنوز بوسیله الگوریتم ملاحظه نشده است ،

مانند  $c(w,i)$  حداقل برای تمام node های ملاحظه نشده است.

، پس (n)

بده اضافه

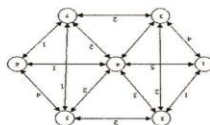


برای هر  $c$

$$c(w) = C(w)$$

Here is the initial link cost table. Note that links costs are the same in each direction.

می گردد .



الگوریتمهای مسیر یابی در کاربرد

در forward جستجوی الگوریتم ، عملکرد تمرکز یافته مناسب تری ادعا کرده می

شود در back ward جستجوی الگوریتم ها می توانست فقط ارزش منطقه یا

نیم منطقه اطلاعاتی پیروی شده را که بلافاصله را از node های مجاور است را

اداره کند.

ارزش پارامتر کاربردی در مسیر یابی الگوریتم ها ممکن است یک پارامترهای

جهانی گوناگونی را بازتاب کند که شامل مخابرات واقعی تاخیری و فضای

میانگیر مورد نیاز بوسیله لینک drivel می باشد همچنین آن ممکن است در

فرمول محاسبه ارزش کاربر ملین شده استفاده گردد و در برخی شبکه های

کاربردی در ارزش (قیمت) یک لینک یک کارکرد دینامیکی میزان و ماهیت ترافیک بر روی شبکه وجود دارد و بنابراین آن مطلوب در دوبر حساب کردن جداول مسیریابی در فواصل مناسب است. و ترافیک داده ها در گردآوری بالا در داده های مورد نیاز برای جدول محاسبه مجدد و انتقال نتایج به nodeها (گره ها) که می توانند به تراکم بیشتر منتج می شود وارد گردید آن بایستی همچنین شود که هر دو جدول مسیریابی الگوریتم یک پیچیدگی را دارند.

پروتوکل اینترنت :

در پروتوکل اینترنت (ip) یک پروتکل جهت دار داده بوسیله منبع و مقصد hot ها برای مکاتبه داده ای عبوری یک packet-switched inerntwork به کار برده می شود.

داده اه در یک ip intrenrtwork در قالبهای ارجاعی مثل بسته ها یا داتا گرام ها در دوره های بطور اساسی در ip مترداف هستند فرستاده می شوند بویژه در IP هیچ SETUP نیاز نمی شود. قبل از اینکه یک HOST مترداف هستند فرستاده می شوند بویژه در تلاش برای فرستادن بسته ها به یک HOST کنند آن قبلا کنند آن قبلا ابلاغ شده است. در پروتوکل اینترنت IP یک سرویس

داتاگرام تا مطمئن ایجاد شد (همچنین بهترین تلاش نامیده شد) آن تقریباً گارانتی در اطراف جعبه ایجاد می‌کند بسته ممکن است آسیب دیده برسد آن ممکن نداشت و در هم برهم گردد مقایسه شد با دیگر بسته های ارسالی در هر دو HOST مشابه آن ممکن است دو نسخه ای المثنی گردد و یا کاملاً رها شده و بیفتد اگر یک کاربرد نیاز به اعتبار داشته باشد ، آن توسط دیگر وسایل آماده گردیده می‌شود.

**packet switches** یا مسیر یابهای **internetwork** ، داتاگرام های **forward IP** از میان لایه شبکه های بهم متصل شدن در فقدان تحویل برخی گارانتی ها ، طرحی از **packet switches** در نظر گرفته می‌شود. که بسیار ساده تر ساخته شده است. (توضیح اینکه اگر شبکه سقوط ، نگارش دوباره یا در غیر اینصورت بسیاری از بسته ها آسیب ببینند در اجرا دیده شده بوسیله کاربر، سست خواهند شد . بنابراین اغلب عناصر شبکه به سختی تلاش می‌کنند این چیزها – از این پس در دوره بهترین تلاش انجام نشود.)

**ip** عنصر متعارف و معمول در اینترنت عمومی امروزه ، پیدا شد. پروتوکل رایج و عمومی ترین لایه شبکه در استفاده امروزه **ipv4** است این نسخه پروتوکل ،

نسخه ۴ را انتقال داده میکند و ipv6 جانشین ipv4 در نظر گرفته می شود در اینترنت تدریجا آدرسها را تمام می کند و ipv6، منبع ۱۲۸-bit و عنوان مقصدها را دارد، بیشتر از عناوین آدرس ipv4 یا منبع 32-bit عناوین فراهم میکند. نسخه ۵ برای یک جریان پروتکل های آزمایشی تعیین کرده شده اند دیگر شماره نسخه معمولا برای پروتکل های آزمایشی تعیین کرده شده اند اما بطور وسیعی استفاده نشده اند. IP Addressing و مسیر یابی : شاید بیشترین نمودهای مجموعه IP مسیر یابی و آدرس های هستند addressring به اینکه چگونه انتهای hot ها به صورت IP addresses تعیین می گردد و اینکه چگونه و اینکه چگونه زیر شبکه های addresses تقسیم کرده شوند و به یکدیگر طبقه بندی می کردند تخصص داده می شوند مسیر یابی ip بوسیله تمام host ها انجام گردیده می شود اما بطور مهمترین بوسیله مسیر یابل interetwork که به طور نمونه هم در مدخل درونی پروتکل ها ، IGP's و هم در مدخل خروجی پروتکل ها EGP's به کار می روند که کمک به ساختن تصمیمات Forwarding داتاگرام IP

از میان شبکه های اتصالی IP می کنند

IPV6 و سیستم نام گذاری حوزه domain name

IPV6 addressed در سیستم نام گذاری حوزه بوسیله گزارشات AAAA برای مراجعات Forwarding نشان داده می شوند بوسیله مقایسه با گزارشات A برای IPV4 مراجعات معکوس تحت ip6.arpa و گزارشات DNAME را داشته اند.

آن در RFC2874 آزمایشی و منابع خودش را تعیین کرده شد. مادامیکه طرح AAAA یک نتیجه و تعمیم ساده IPV4 DNS است، طرح A6 یک معاینه کامل DNS به عمومی تر شدن است و از اینرو پیچیده تر است.

#### مسیر یابی الگوریتم

تمام مسیریابی لبه، الگوریتمهای بوسیله مرحله طرح بندی اجرا Y فایل ها فراهم شدند. مرحله طرح بندی ان را آسان در بکار بردن یک مسیر لبه الگوریتم همانند یک مرحله پس پرداز به برخی طرح بندی اصلی الگوریتم می سازد . Y فایل ها انواع مختلف لبه مسیر یابی را تقویت می کنند مسیریابی اصلی ساختمانی و

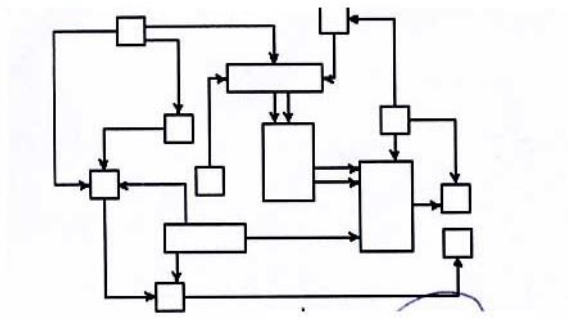
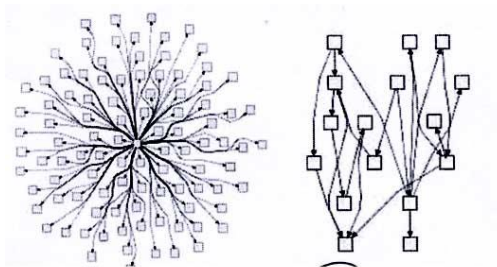
#### مسیر یابی قائم

مسیریابی ساختمانی : در بخش مسیر یابی لبه ساختمانی شرح داده می شود.

مسیر یابی قائم : در بخش مسیر یابی لبه قائم شرح داده می شود.

اختیارات مسیریابی : راه انتخاب شدن لبه ها: اگر این اختیار به فعالیت پرداخته شود فقط لبه های انتخاب شده برای مسیریابی مورد ملاحظه و رسیدگی قرار

خواهند گرفت



نمونه ای از مسیریابی قائم

حداقل فاصله این حداقل فاصله مجاز در بین **node** (گره ها) و لبه ها را مشخص

می کند.



کاربرد خمیدگی های موجود: این اختیار ، خواه خمیدگی های موجود را که بایستی مثل یک راه حل ابتدایی برای مسیر یابی جدید به کار برده می شود را، مشخص می کند.

تنها راه لازم : اگر این اختیار به فعالیت را داشته شود فقط لبه هایی که مختل گشته اند، در مقیاس حداقل فاصله دوباره تعیین مسیر خواهند شد.  
در مقیاس حداقل فاصله دوبار تعیین خواهند شد.

مسیر یابی لبه قائم : مسیریاب لبه قائم یک طرح بندی الگوریتمی گردان و نیرومندی ، برای مسیر یابی که دیاگرام لبه های کاربردی عمودی و افقی قطعات خطی است . وضعیت های دیاگرام گره ها یا برخی دیگر از لبه های روی هم افتاده بریده نخواهند شد.

امکانات عرضه شده بوسیله مسیریاب آن را ی طرح بندی کامل برای تاثیر بر یکدیگر یا توسعه جایی ایجاد می کنند. برخی لبه های بایستی دوباره طراحی می شوند. بعد از اینکه کار برخی گره ها node را جابجا می کند. متعاقبا اضافه کردن لبه ها بایستی به اندازه دیاگرام موجود طراحی می شوند. شکل ۵۴-۵ : یافتن یک

راه در میان یک مسیر پرپیچ و خم

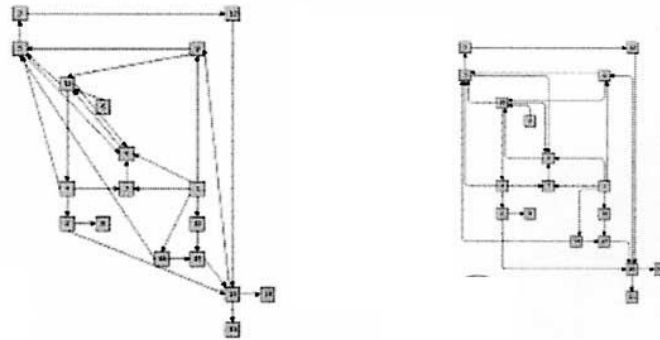
اختیارات مسیر یابی: مسیریاب لبه قائم یک مجموعه اختیارات را فراهم می کند که در رفتار مسیر یاب موثر می گردد. این بخش تشکیل چند شکل اختیارات موجود را روشن می کند.

کاربرد حداقل فاصله **custom** بر گرهای **Node** اگر پس از یک **custom value**

در این حداقل فاصله بین قطعه ای که گم شده گم استفاده خواهد شد. گزینه از جهت

بد شد.

وسیله

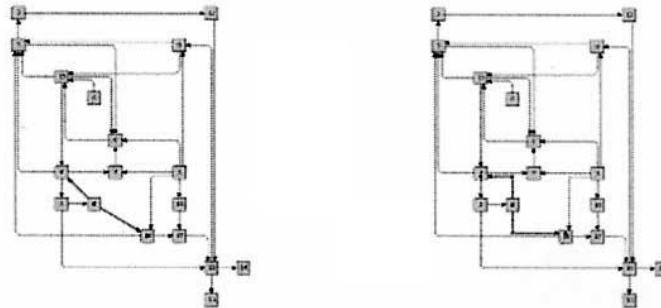


شود.

Graph with fixed node positions, before... ..and after initial orthogonal edge routing.

یج می

اینکه



Subsequently added edges...

...nicely integrated into existing layout.

مسیر روی دریچه مشبک **grid** : اگر مجموعه باشد بعد از همه مسیرهای لبه روی خطوط دریچه مشبک **grid** مسیریابی خواهد شد. اگر مجموعه نباشد ، بعد از مسیریابی آزاد به راههای فاصله فقط برای یک پروسه لبه بطور شایع در زمانیکه آنجا فاصله بسیار کمی در یک راه با **value** درست و صحیح است وجود دارد

فضای **arrive** شده در مقابل جستجوی مرکز **drive** شده: نسبت به دو استرانژهای سنگین تعریفی ، تعیین می شود ، زمانیکه به دنبال یک راه لبه ای می گردند ،

مسیریاب لبه تا حد امکان به **setvalue** می چسبد. اما در فاصله گذاری ارزش به طور انتخاب کاهش می یابد : فقط برای یک پروسه لبه به طور شایع در زمانیکه آنجا فاصله بسیار کمی در یک راه **value** درست و صحیح است وجود دارد

فضای **drive** شده در مقابل جستجوی مرکز **drive** شده : نسبت به دو استرانژی سنگین تعریفی ، تعیین می شود. زمانیکه به دنبال یک راه لبه ای می گردند یعنی مرکز **drive** شده و فضای **drive** شده سنگین و حجیم هستند و به نسبت با یک **value** ما بین صفر و یک ، صفر اظهار داشته می شود **values** به صفر، صفر

راههای پایه را راهنمایی می کند که بیشتر از فضای موجود پخش کرده می شوند  
values به یک صفر تاکید و اهمیت بیشتری به راههای نزدیک در **bary center**  
لبه داده شده، می دهند.

حداقل منطقه عبور: اگر مجموعه **set** نباشد، تعداد مناطق عبوری مشاهده نشده  
در یک بخش گره می تواند زیاد افزایش یابد. از زمانیکه این اختیار یک اثر مثبت  
روی دیاگرام با قابلیت خواندن را دارد. ان بوسیله کوتاهی و غفلت توانا گردیده  
شد.

بها عبور: یک جریحه برای محل های عبور لبه تعیین می گردد. اساسا یک نرخ  
گریحه به معنی آن است که یک لبه پرتو بیشتر مسیر  $n$  زمان را نسبت به عبور  
یک را لبه که قبلها که مسیریابی شده، تغییر می یابد. در مقابل به حداقل منطقه  
عبور بهینه سازی به طور سراسری روی یک راه لبه داخلی کار می کند **value**  
های خوب برای یک مجازات عبور از حدود یک صفر به سه، صفر قرار می گیرند با  
کوتاهی کردن این **value** به صفر، صفر **set** کرده می شود. که آنجا هیچ مجازات  
و جریحه ای وجود ندارد.

مسیر یابی دوباره لبه عبور اگر مجموعه باشد، پس لبه ها با برخی محل های عبوری دوباره مسیریابی خواهند شد این بهینه سازی فقط پی ترکیب با value بالاتر از صفر برای بهاء منطقه عبور برده و ایجاد می گردد. با کوتاهی کردن ،لبه های دوباره مسیریابی شده از کار افتاده می شوند

فواید و مقرارت مسیریاب - لیبنک شده : فایده اصلی و اولیه مسیریاب معین شده این است که ان به سیرعلت عکس العمل نشان می دهد ودریک مقدار کردن دار زمان به اتصال تغییر می کند همچنین بسته های معین لینک شده به بالای شبکه فرستاده کرده می شوند ، کوچکتر از بسته های کاربردی در مسیریاب بردار - مقصد هستند مسیر یاب بردار مقصد نیاز به یک جدول مسیریاب گره های درست و بی عیب دارند که انتقال داده می شوند. مادامیکه در مسیریاب معین لینک شده فقط اطلاعات درباره گره بلافاصله به همجواریها انتقال داده می شوند. بنابراین این بسته ها به حدی کوچک هستند که آنها در منابع شبکه به درجه مهمی استفاده نمی شوند. ضرر اصلی و عمده مسیریاب ملین لینک شده این است که نیاز به ذخیره سازی بیشتری نسبت به مسیر یاب بردار - فاصله در روان بودن و حرکت دارند

## مسیر یاب peer to peer

معرفی سیستم peer to (p2p) می تواند به چند شکل موثر واقع شود E-mail ،  
ایستگاه تقویت تمرکز یابی شده یابه طور استانیکی عددی می گردند و بنابراین  
بدون مشکل است .

دیگر دسته شبکه های p2p شبکه پوششی هست و شبکه های پوششی یک  
توپولوژی واقعی روی راس لینک های فیزیکی و طبیعی شبکه می سازد.

گره ها لازم شده وبه این شبکه به طور دینامیکی متصل می شوند و متوسط  
UPTIME گره های اختصاصی به طور نسبی پایین است و توپولوژی یک  
نسخه یک شبکه ممکن است در تمام مدت تغییر می کند به محض اینکه یکی از  
مسیرها تاسیس کرده شود در انجا هیچ گارانتی در طول در زمان وجود ندارد که  
ان معتبرو به ثوت خود باقی خواهد ماند.

مسیر یاب در این شبکه ها وجود دارد بنابراین بسیار مشکل ساز است و مرکز  
توجه گزارشات ما خواهد شد. برخی از طراحان مسائل تما الگوریتمهای مسیر یاب

**:P2P**

**SCALBILITY -۱**

۲- پیچیدگی

۳- گمنامی هستند

۴- SCALBILIT یک حد چگونگی عملکردهای سیستم های است زمانی

که شماری از گره ها ویشماری از پیغام ها روی ترقی و پیشرفت های شبکه

است . پیچیدگی مراحل ترتیب گامهای گامهای مورد نیاز برای یک بسته از یک

host به دیگری در بدترین حالت scenario سو می کند ، است .وگمنامی

anonymity نیاز بیشترین شبکه های p2p است اگر چه یک شبکه به طراحی

شدن به anonymity را فراهم می کنند و سپس این یک مشکل است که

بایستی در سطح مسیر یابی حل کرده شود ما به مثالهای کم الگوریتمهای

مسیریاب را از هرکدام از این مناظر اشاره خواهیم کرد.

مسیر یابی Guntella

Guntella قابل بحث در اولین مسیر اصلی شبکه پوششی که از کاربرد

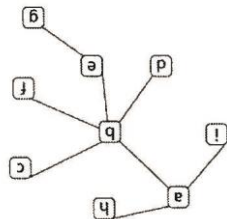
گسترده ای بهرمنند می شود.

عقیده قبلی ساده بود. در اتصال یک مشتری به شبکه ، می بایستی آدرس حداقل

یک گره موجود روی شبکه را شناسایی کرده باشد. بیشتر یک بیشتری یک

اتصال به این گره **node** داشت آن می توانست بعد از پراکنده کردن یک صدای  
 غر مانند آدرسهای دیگر گره ها **node** را بیاید ایده اصلی این است که هر گره

**node** یک اتصال به یک شماره از  $n$  را به طور نرمال تقریباً ۵ تا



برقرار می کند.

جستجو کردن در شبکه برای یک منبع تا حدی کاربرد یک **time-to-live**

**counter** را کنترل نمایند. این نوع مسیر یاب ساده ترین نوع ممکن برای یک

شبکه پوششی است به هر حال آن نیز خودش بدون مشکلات نیست.

مسیر یاب طرح **Gnutemlla** یا طیفان کننده ، خیلی خوب برای شبکه کوچک و

متوسط کار می کند. ان نشان داده است که ارزش جستجو روی یک شبکه طرح

**Gnutemlla** به طور **superlinear** مانند شماری از گره های **node** زیاد شده ،

افزایش می یابد. زمانی که اشباع گره **node** اشباع گره **node** رخ می تواند جز جز

و ناقص بشود **Gnutemlla** ، بنابراین در صورتی که یک حل ساده راه به خوبی

مطابق مقیاس قرار نمی دهد. جستجو شبکه **Gnutemlla** را به طور کلی

آمیختگی و پیچیدگی تعریفی است همانطور که هر جستجو درباره فواصل **n&d**



تفسیر خواهد شد در باره اینکه **time to live n** است و **d** در شماره ای از همتای های هر **node** گره است. **flooding** در بیشترین حل بهینه برای مسیر یابی آشکار بدهی نیست و ان مدتی نبود قبل از دیگر الگوریتمهای پدیدار مسیر یاب **p2p** که موثرند بودند.

ما درباره تعداد از این ها به انضمام مسیریاب معنایی و جدوال پخش **hash** را باعث و مطرح خواهیم کرد.

یک شبکه که بطور ویژه طراحی کرده شد به طور مستعارانه در ذهن طراحی کرده شد. مانظری درنتایج اتخاذ خواهیم کرد که **anonymity** بی نامی بر حسب **scalability** و پیچیدگی ارائه می گردد فایده اصلی **chord** ها در گارانتی است که شما یک پاسخ درون زمان **hog(n)** به دست خواهید آورد. همچنین یک فایده مهم در فقدان افزونه بالایی است. هر دو اینها به آن یک لبه وسیع روی برخی الگوریتمهای **fooding** می دهند اما درکل معلوم است که الگوریتمهای **DHT** منابع داده ها ایشان را دریک راه تشکیل شده اندوخته می کند آنها همیشه الگوریتمهای **flooding** را در این منطقه شکست خواهند داد

CHORD به خوبی مطابق مقیاس قرار داده می شود. معین است که جستجو

دسته LOG وجود دارد و همچنین به طور نسبی پیچیدگی کم دارد.

فواید:

۱- داده های عمومی در کل سیستم تکرار می شود.

۲- داده ها به طور بی نام و مستعارانه و به طور آزاد پخش کرده میشود،

اصول و عقاید کلی را در سیستم در هدف دارد.

۳- الگوریتم مسیریابی به وفق دادن و بهبودی سودمند به طور اضافه

طراحی کرده شد.

مضرات:

۱- مردم به بخشیدن و اهداء بخشی از hard-drive شان به سیستم

مخصوصاً دو دل و مردد هستند زمانی که آن می توانست در انداختن

اطلاعات استفاده شود. آنها پسند و موافقت نمی کنند.

۲- شبکه به سادگی قابل جستجو نیست، کاربر نیاز به شناخت کلید پیدا

کردن یک فایل دارد.

۳- شبکه کند و آهسته است. همانطور که داده ها باید در بین تمام گره ها (node) سریع و فوراً عبور کند. جستجوهای می توانند بطور جز جز کند و آهسته شوند زیرا آنها از multicasting استفاده نمی کنند.

رده بندی یک به یک الگوریتم های مسیریابی

تصمیمات مسیریابی: عملکرد معیار برای مسیریابی مکان و زمان عملکرد معین شده مسیریاب است.

مسیریابی پخش شده: عملکرد مسیریابی در مسیریاب یا گره ها همانند بسته سفر کرده عبوری در شبکه محاسبه کرده می شود. header فقط شامل آدرس مقصد، کاربرد بوسیله مسیریاب در انتخاب کردن بازده کانال یا کانال ها می باشد. هر مسیریاب فقط حوالی خودش را می شناسد، از زمانی که طراح تمام توپولوژی را به طور توزیعی درون مسیریاب اختصاصی به صورت رمزی درآورده است. مسیریابی توزیع شده مخصوصاً در توپولوژی های متقارن و منظم مطلوب و مساعد است، از زمانی که تمام مسیریاب ها الگوریتم مسیریابی یکسانی را استفاده می کنند.

مسیریابی منبع: گره های منبع به طور از بیش تعیین شده راه های مسیریابی را قبل از تزریق بسته ها به درون شبکه کامل می کنند، بطوریکه مسیریاب ها فقط header ها می خوانند ( و معمولاً subfield های مناسب و اختصاصی را جدا کرده یا مشخص می کنند) . و از اینرو مجموعه (set) به طور مکانیکی راه خودش را برمی گزیند. اگر میزان خروج یک مسیریاب  $k$  است. سپس header بسته پیروی از یک راه طویل  $d$  کرده که حداقل نیازمند bit های  $d \log k$  را برای رمزی کردن شماره های کارکرد کانال است. این برنامه در ماشین IBM SP-2 استفاده کرده می شود.

در شبکه ها اساس روی تولیدات Cartesian اندازه اطلاعات مسیریابی header که می تواند بوسیله استفاده مسیریابی street-sign (علامت خیابان) کاهش یابد.

کوتاهی و نقصان کارکرد کانال اندازه و جهت یکسان شبیه نیروی معروف شده کانال است، جز اینکه header صریحاً با شماره معلوم کارکرد جدید کانال به یک آدرس پیوسته گردیده که با آدرس های مسیریابی رایج جور می باشد.

مسیریابی پیوندی (هیبرید) چند فازی: گره منبع فقط آدرس های چند گره واسطه و میانه را از پیش محاسبه می کند و راههای دقیق بین آنها، در یک روش توزیع شده بوسیله مسیریاب ها تعیین گردیده می شود. این عملکرد برای مثال در ماشین NCUBE-2 انجام شد.

آن فرض می کند که مسیریاب ها بطور متوالی آدرس های گره واسطه و میانه را جدا می کنند.

مسیر یابی تمرکز یافته: در عملکرد مسیریابی بوسیله یک کنترل کننده تمرکز یافته تعیین کرده می شود. این به طور نمونه در ماشین های SIMD استفاده شده است.

اجرا الگوریتم مسیریابی: تصمیمات مسیریابی باید سریع باشند. در حالت مسیریابی توزیع شده، یک اجرا HW مطلوب است. آنجا دو معبر اصلی وجود دارد.

ماشین محدود-معین: الگوریتم SW یا HW اجرا و تکمیل چند رباط مکانیکی محدود-معین.

جدول مراجعه: منبع گره ها و یا مسیریاب ها، جداول مسیریابی را حفظ می کنند.

تعدادی از مدخل ها  $O(N)$  است. در جائیکه  $N$ ، # گره ها در شبکه است. جداول مسیریابی گره منبع شامل تمام راههای تشخیصی می باشند، با در نظر گرفتن اینکه در حالت مسیریابی توزیع شده در جداول مراجعه فقط می گویند که کارکرد کانال یا کانال ها مربوط به هر مقصد می شود.

کاهش یافتن اندازه جدول خطی، مسیریابی وقفه ای نامیده شده که می تواند کاربرد داشته باشد. جداول مسیریابی می توانند استاتیک یا به طور دینامیکی باقی بمانند. یک مثال که یک سیستم با جداول مراجعه ای به طور دینامیکی جدید و به روز می گردد. Myrinet است. که اجازه محاسبات مجدد اتوماتیک جداول مسیریابی را هنگامی که شبکه اتصالی داخلی تغییر می کند را دارد. برای مثال به واسطه حذف یک گره را می توان گفت.

**Adaptivity** تصمیمات مسیریابی می تواند نه فقط اساسی روی آدرس ها، بلکه همچنین روی دیگر اطلاعات باشد.

الگوریتم های مسیریابی قطعی: همیشه راه مسیریابی واحد یکسان برای جفت منبع معین و آدرس های مقصد، به طور نمونه یک shortest را تولید می کنند. زمانی که مسیریابی منبع به کار برده می شود، گره منبع برگشت عملکرد مسیریابی فرضی یک راه منحصر به فرد را بدون نظر به اطلاعات درباره عبور و مرور اجرا می کند. برای مسیریابی توزیع شده مسیریاب ها تصمیم های منحصر به فرد را در هر گره واسطه و میانه ایجاد می کنند. مسیریابی قطعی در بیشترین ماشین های موازی موجود به طور تجارتي به کار برده می شود.

این ساده، سریع و بسیار عملی تحت اتخاذ یکسان عبور و مرور است. در ماشین های مشبک- پایه ای آن مسیریابی اندازه- ترتیب است. (XY,XYZ,e-cube). آن شناسایی گردیده می شود به بی تکلیفی دچار وقفه شدن- آزاد بودن در شبکه و در اجرا انجام HW. آسان است. عبور و مرور ناهمسان نیاز به چند درجه adaptivity دارد.

الگوریتم های مسیریابی بی توجه: در توجه به برخی دیگر از اطلاعات به استثنای آدرس ها، به یک اندازه به مسیریابی قطعی، گرفته نمی شود. تصمیم

مسیریابی بی توجه به وضعیت عبور و مرور شبکه هستند. هر مسیریابی قطعی، بی توجه است. اما مسیریابی بی توجه حتماً قطعی نیست. برای مثال آنجا ممکن است تعدادی از کوتاهترین راهها در منبع و مقصد وجود داشته باشد و الگوریتم مسیریابی به طور چرخه ای یا به طور تصادفی یکی از آنها انتخاب می شود.

الگوریتم های مسیریابی انطباقی: اطلاعات درباره عبور و مرور شبکه و یا وضعیت کانال به اجتناب از مناطق انبوه شده یا معیوب به کار برده می شود. مسیریابی انطباقی گره- منبع فقط زمانی مفید است که در وضعیت عبور و مرور خیلی سریع تغییر ایجاد نکند، وگرنه در گره منبع ممکن است اطلاعات منسوخ داشته و یک وضعیت سراسری گزاف به monitor است. عملکرد مسیریابی: که یک مجموعه کانال های کارکردی ممکن، تحویل داده می شود.

عملکرد انتخاب کارکردی: که یکی از کانال های کارکردی آزاد در میان وضعیت اطلاعات منطقه کارکردی انتخاب می شود. (اگر این چنین موجود باشد).



تست در الگوریتم انطباقی: در تست الگوریتم انطباقی مجزا از دیگر وظیفه راهنمای مسیر منطبق است.

تست ترکیبی از ۲۰ وظیفه است که شامل سفرهایی بین اثرات متقابل در منطقه palo Alto می باشد. در جبران کردن برای فقدان تاثیر بر یکدیگر، ما ۴ مسیر برای هر وظیفه در عوض آن دو تولید کردیم. از زمانی که ما هیچ فرصتی در ساخت مدل های کاربر نداشتیم، ما به ترتیب بردارهای weight اکتشافی را با یک weight واحد برای یک صفت و صفر برای استراحت، ایجاد مسیرهای بهینه برای زمان، فاصله، تعداد پیچش ها را به کار می بردیم. ما ۴ مسیر از برجسب دار به طور تصادفی بین A,D روی یک نقشه palo Alto طرح ریزی کردیم. شکل ۵ مثالی از یکی از برنامه ها را نشان می دهد و خودش ۴ مسیر را انتخاب می کند.

شکل ۵: برنامه ساده برای موضوعات در نقطه آغاز در بسته در بالا سمت چپ است و نقطه انتهایی در بسته پایین سمت راست می باشد. A مسیری با کمترین پیچش ها می باشد. B سریعترین مسیر است. C مسیری با کمترین تقاطع ها است و D کوتاهترین مسیر است.

شکل ۶: میزان تبدلات برای ۳ صفت راجع به مسافت، محاسبه از تمام داده ها برای هر موضوع است. ارزشهای مثبت بالا برای یک مسافتی اشاره می کند که کوتاهترین مسافت است و کم اهمیت تر از کاهش است که نسبت داده می شود، ارزشهای نزدیک صفر اشاره می کند که فاصله کوتاهتر مهم تر است و ارزش های منفی بالا اشاره می کند که مسافت طولانی تر قابل ترجیح تر است.

مسیریابی دینامیک: مسیر برای به روز شدن و جدید شدن RIP روی شبکه پذیرفته خواهد شد و آنها را در ساختن یک جدول مسیریابی به کار می برند. RIP یک انتخاب مسیریابی خوب برای شبکه های خیلی بزرگ نیست اما آسان در اجرا کردن است و برای شبکه های کوچک به خوبی کار می کند در `etc/` ورودی های فایل، مسیرهای استاتیک مجاز به اضافه کردن به `daemon` مسیریابی را که به صورت دستی هستند را فراهم کنند. `Format` فایل به شرح زیر است:

**Startkey word** یکی از `net-1` - مسیر `A` به یک شبکه

`2-host` - مسیر `A` به یک `host` است.

آدرس مقصد مکان بسته را می گویند. اگر مقصد ..... است. پس آن در مسیر  
قصور default است.

ورودی در مدخل خارجی که در جستجوی مقصد می باشد، با gwaddress  
مشخصی در IP address ورودی تعریف می شود.

متریک یک نیاز keyword است و ارزش متریک در ارزش به مقصد است.

ارزش فعال / منفعل اشاره می کند اعم از اینکه یک مسیریاب به روز کردن  
updates مسیریابی را انجام می دهد.

مسیریابی adaptive از Biocrawler

مسیریابی adaptive قابلیت یک سیستم را شرح می دهد در میان اینکه  
مسیرها بوسیله مقصدشان مشخص کرده می شوند.

دگرگونی در مسیر که وظایف مسیر را در سوتالیرسیستم درواکنش به تغییر  
در وضعیت های است انطباق بر آن شد که به مانند برخی مسیرها امکان باقی  
ماندن معتبر را در پاسخ به تغییر مجاز کند استفاده مردمی یک سیستم حمل  
و نقل می تواند مسیر یابی adaptive را آشکار کند برای مثال اگر یک  
ایستگاه خط آهن بسته شود مردم می توانند از یک قطار در یک ایستگاه

متفاوت پایین آمده روش دیگری را به کار ببرند. همانند یک اتوبوس که به مقصدش می رسد . **term** به طور عادی در داده های شبکه های شرح دادن قابلیت یک شبکه در آسیب دیدن مسیر دور تا دور به کار برده می شود. مثل فقدان یک گره **node** با یک اتصال بین گره ها به شرطی که انتخاب های راه دیگر موجود باشند آنجا چندین پروتکل به رسیدن این ها به کار گرفته می شوند.

**RIP**

**OSPE**

**IS-IS**

**IGRP/EIGRP**

**IGRP/EIGRP**

سیستمهای که مسیریابی **adaptive** را اجرا نمی کنند همانند کاربرد مسیر یابی استاتیک شرح داده می شوند در جائیکه مسیرها در بین یک شبکه بوسیله راههای فیکس شده (بطور استاتیکی) شرح داده می شوند یک تغییر مثل فقدان یک گره یا فقدان یک اتصال بین گره ها جبران کرده نمی شود. این

به این معنی است که هیچ چیزی که تمایل گرفتن یک مسیر موثر را دارد،  
بایستی قبل از اینکه خودش را دوباره آغاز کند منتظر برای جبران کوتاهی  
و ناتوانی بماند تا امید در رسیدن به مقصد خودش گردد و از سفر دست  
بکشد.

متریک های متعدد

EIGRP با ۵ متریک مختلف با هر مسیر مربوط می باشد.

تاخیر کردن Dekay

پهنای باند Bandwidth

اعتبار و قابلیت اعتماد Reliability

MTV

Load

برای هدف های مسیرهای مقایسه ای اینها در فرمول **weigted** به ایجاد  
متریک واحد به یکدیگر می پیوندند.

$$\{(K4 + \text{قابلیت اعتماد } k_6) * \{K3 \text{ تاخیر کردن} + (256 - \text{Load}) * K2\} * \text{پهنای}\}$$

$$\{K1 * \text{پهنای باند} + \text{باند}\}$$

جایی که ثابت های مختلف ( $k_1$  تا  $k_5$ ) می توانند بوسیله کاربر در ایجاد حرکات مختلف set کرده شوند اگر  $k$  set به صفر است در term استفاده کرده نمی شود کسری برای  $k_1$  و  $k_3$  به  $k$  set می شود و به صفر استراحت می کند.

بطور موثر کاهش در فرمول بالا است.

تاخیر کردن +پهنای باند

سیستم میانه به سیستم میانه is-is یک پروتوکل کاربردی بوسیله تمهیدات شبکه مسیریاب در تعیین کردن بهتر خلاصه طبقه ای برای طرح پروتوکل شبکه کامپیوتری و ارتباطات ، توسعه بخشی از سیستمهای باز بهم پیوسته آغازی وجود دارد. ان همچنین اسلوب ۷لایه OSI نیز نامیده می شود .

Rip(۲)

Biocrwler

Rip که از اختصار ریا ترکیب حروف اول یکسری کلمات است ، چند معنی مختلف دارد: آن می تواند نماینده آسایش در صلح rest باشد یک اصلاحی که

اغلب روی سنگ قبرها نمایان است. در بیان بطور اصلی از اصلاح لاتین نماز میت در آرامش **reguiecat in peace** و در نسخه انگلیسی یک مثال تشکیلات **back** است. (اگر چه دو عبارت چیز یکسانی است).

۲- در **bbsing** و **rip** نماینده پروتوکل تصویر متحرک یا **ripscip** است. یک روش فرستادن شبیه سازی گرافیکی بر روی یک **bbs**

۳- در شبکه کامپیوتری، **rip** نماینده برای پروتوکل اطلاعات مسیر یابی **ip, ipx** و یا **idp** است.

۴- در صنعت **pre-press**، **rip** نماینده برای **raster image processor** است.

۵- در تولید فایل های **mp3** و **ripping** به فرآیند برگرداندن خواندن انالوگ **cd** به فایل های **mp3** دیجیتال رجوع می گردد.

در امور سیاسی انگلستان، **rip** نماینده فعالیت ۲۰۰۰ نیروهای تحقیقی تنظیم مباحثه ای حکومت انگلستان است، که قوانین وقواعدی برای ارتباطات قطع شده روی تمام شبکه های ارتباط از دور و پستی جدا می کند. که شامل دسترسی به ارتباط در اینترنت است.

۵- یک توزیع **linux**، استرداد و باز بافت ممکن است.

۷- roleplayiny در امکانات نامحدود، سیستم rpg یک کامپیوتر

Sri بین المللی : از biocrawler

Sri بین المللی یکی از موسسات تحقیقی بزرگترین قرار داد دنیا است. آن مثل موسسه تحقیقی استانفورد در سال ۱۹۴۶ بوسیله منافع یکی شده با دانشگاه استانفورد تاسیس گردیده شد. بعداً آن کاملاً مستقل شد و مثل یک سازمان بدون سود تحت قوانین ایالات متحده و کالیفرنیا تاسیس گردیده است.

در سال ۱۹۷۲ دکتر hal pauthoff یک محقق در sri ، یک سری پروتکل های مطالعه مکانیک های کوآنتوم را در فرآیندهای life به کاربرد این در برنامه ramote viewing مباحثه ای در حال حاضر نتیجه ای می دهد که به طور گزارشی قطع گردیده و غیر محرمانه و تنزل مرتبه پیدا کرد. DouglasC.Engelbart مشهورترین سازنده mouse کامپیوتری است و همانند یک pioneer اثر متقال کامپیوتر - انسان به طور قابل بحثی برجسته ترین دانش آموخته sri است. محققان بین المللی sri در جهان اول پیشرفت کردند و فقط تمام کامپیوترهای دیجیتال مغناطیسی ، براساس بسط و توسعه هایی به حافظه مغناطیسی قرار گرفت.



در سال ۱۹۷۷ موسسه تحقیق استانفورد مثل sri بین المللی ، مشهور شد و به طور رسمی از دانشگاه استانفورد جدا گردید این یک واکنش و پاسخ و دیرتر از موقع به دانشجویان معترض ضد جنگ بود کسانی که باور داشتند سرمایه کار sri darpa ضرورتا در قسمت استانفورد در ترکیب نظامی - صنعتی ساخته می شد.

در طی این کار، sri بیش از ۱۰/۰۰۰ حق و جواز ثبت شده انحصاری برای استفاده از اختراعی در مهندسی و تکنولوژی را اعطا کرده است. sri بین المللی تحقیق و توسعه در چند ناحیه ، انتقال داده و رهبری می کند، هر دو به طور مستقل و برای اجاره و فروش بر روی تحقیق مستقل ، گزارش می دهند. curtis carlos.ph.D رئیس SRI CEO بین المللی است.

پروتوکل داتا گرام کاربر UDF یکی از پروتکل های هسته در درخواست پروتوکل اینترنت است استفاده UDP یکی از پروتوکل های هسته در ، درخواست پروتوکل اینترنت است . استفاده UDP ، برنامه های روی کامپیوتر شبکه را می تواند پیامهای کوتاه مشهور مثل داتا گرام به یکدیگر را ارسال نماید. UDP در اعتبار و گارانتی های سفارشی که tcp انجام می دهد

فراهم نمی‌گردد، UDP در اعتبار و گارانتی‌های سفارشی‌های که tcp انجام می‌دهد، فراهم نمی‌گردد. داتا‌گرام‌ها ممکن است خارج از سفارش وارد شوند یا بدون توجه از دست بروند به هر حال، نتیجه می‌شود. که UDP سریعتر و موثرند برای اهداف شبکه یا زمان - حساس است.

کاربردهای شبکه‌های معمولی و عادی که UDP استفاده میکنند شامل سیستم Domain Name کاربردهای جریان رسانه‌ها ، voie osep و مسابقات Online می‌شوند.

مسیر یابی معنایی : مسیر یابی معنایی یک روش مسیر یابی است. که بیشتر از توپولوژی شبکه روی ماهیت تحقیق و پرسش متمرکز یک روش معنایی روی مسیر یابی سنتی را اصلاح می‌گردد بوسیله گره‌های priorisng که به طور ابتدایی در اطلاعات تهیه شده درباره انواع درباره انواع حجم ارجاع شده بوسیله تحقیق و پرسش ، معتبر شده اند.

در توانایی جستجو برای اطلاعات روی یک شبکه p2p به طور معنایی در داده‌های نیاز به تشریح معنایی در ارتباط با آن دارند، یک حل معمولی در ارتباط با آن دارند، یک حل عمومی ، کاربرد (1)rdfmata-data برای این

هدف است داده های اسناد ضمیمه شده با **rdf** یک وب معنایی وسیع را تهیه می کنند که می توانست در یک مدل **p2p** پی ریزی شود. یک شبکه طرح شده - اساسی **p2p** مثل این به طور بزرگی از مسیر یابی معنایی سود می برد - مسیر یابی معنایی اساسا از دیگر تکنیک های مسیر یابی متفاوت است زیرا گره های موثر در آینده بواسطه اطمینان دیگر گرهها در توانایی شان به پاسخ دادن به طور صحیح به یک پرسش معین صرفنظر از وضعیت نشان در شبکه انتخاب کرده می شوند.

آنجا چندین پروژه مختلف با یک نظر آزمایشی در قدرت مسیر یابی معنایی آغاز گردیده است نمونه ها **(2) tempidh siberli et wolpre, Nejdigrd** و **wranik,stab** به طور قابل بحثی جالب ترین اینها و سیستم **Remmindin** است که یک الگوریتم مسیریابی بی معنایی پیشرفته با مقصد شبکه های اجتماعی تقلیدی یکی می شود. به علاوه این تعداد تکنیک های پیشرفته برای فیلترینگ همکاری یافته به طور مساوی به **ranking** نظیر و همتهای درون یک شبکه مسیر یابی شده به طور معنایی قابل اجرا و عملی هستند.

## Vpn چیست؟

یک شبکه اختصاصی مجازی یا vpn یک شبکه تکمیلی کاربردی در یک بخش سازمانی شبکه اما برای تهیه کردن نگرهبانی و پوشیدگی یک شبکه خطی استجاری اختصاصی است.

مثالهای قدیمی تر **frame relay** و **ATM** می باشند اخیرا به رجوع بیشتری به تونل های **ipsee** بالا در اینترنت با شاید **pptp** با شماره گیری **l2tp** اتصال **vpn** از میان یک **internet network** بخش شده رسیده است.

برای اهدافمان در این مقاله ، **vpns** ، شبکه های **ip** خواهد شد که هسته **wan** یک شبکه یکی شده ه تهیه کننده سرویس ، **outsorce** کرده است در **IPVPN** اتصال در میان یک شبکه **IP** بخش شده متعلق به تهیه کننده سرویس ، آماده گردیده می شود.

ان در **BGP-MPLS** اساس **VPNS** ثابت و معلوم خواهد شد. ما درباره خواهد شد ما درباره قدرت کافی تهیه کردن اتصال ایمن وبی خطر (وترکیب نسبتا ساده) برای هر دو **in tranets extranets** گفتگو خواهیم کرد.

اصلاحات واژه شناسی

Vpn: اینترنت اتصالی داخلی با سایت یکی شود.

Extranet: vpn اتصالی سایت یا سایت های یکی شده به شرکای کاری

خارجی یا کارپردازان اینترنت vpn: Extranet باز پسین ناامن و نامحفوظ

است

مسیر یاب customer edge (ce) یک مسیر یاب در یک سایت مشتری که به

تهیه کنندگان سرویس متصل می شود. ( از طریق یک یا تعداد بیشتری

مسیریاب provierp.e edge

مسیر یاب هسته تهیه کننده: یک مسیریاب در اتصال داخلی شبکه تهیه کننده

سرویس به مسیر یاب pe ، اما عموماً خودش یک مسیریاب pe نیست.

مسیریاب های دخول و خروج pe : مسیریاب های pe بوسیله خروجی ها و

ورودی یک بسته در شبکه تهیه سرویس .

