

مسیریابی در شبکه های *Ad-hoc*

## مقدمه

یکی از مهمترین مسائلی که در هر نوع شبکه ای مطرح میشود مسیر یابی و پیدا کردن مسیرهای بهینه از هر مبدا به هر مقصدی میباشد، مسیریابی در شبکه های با سیم و شبکه های بی سیم دارای زیر ساخت که نقاط دسترسی در آنها ثابت هستند خود مسئله بسیار مهم و مشکلی است و نیازمند تدابیر و راه حل های خاص میباشد، حل این مسائل در شبکه های Ad-hoc که گره ها ثابت نیستند و دائم در حال تغییر مکان هستند بسیار سخت تر بوده و به تمهیدات بیشتری نیاز دارد چونکه این نوع شبکه ها بهیچوجه توپولوژی ثابتی ندارند و در هر لحظه نحوه چیدمان گره ها دچار تغییر میشود. ما در این فصل روش های مسیر یابی را مرور مینماییم. در ابتدا در مورد معیارها و انواع طبقه بندی ها صحبت مینماییم و سپس بصورت ریز تر یک طبقه بندی خاص و نمونه های مختلفی از انواع روش ها آورده شده است.

## معیارهای طبقه بندی روش ها

در حالت کلی میتوان گفت که هنوز طبقه بندی کامل و استاندارد برای مسیریابی در شبکه های adhoc بوجود نیامده و مقالات مختلف طبقه بندی های متفاوتی را برای این نوع شبکه ذکر نموده اند. شاید یکی از دلایل اینکه طبقه بندی استاندارد برای این شبکه ها وجود ندارد اینست که هنوز که مرزهای دانش در این زمینه مورد دستیابی کامل قرار نگرفته و هر روز روش های جدیدی که از تکنیک جدیدی استفاده میکنند مطرح میشود، در واقع در حال حاضر روش های مسیریابی را میتوان از دیدگاه های مختلفی طبقه بندی و بررسی نمود که از این گونه میتوان روش های Hybrid ، Proactive, Reactive، Geographic، Multicast، Hierarchical Geocast، و چند گروه دیگر اشاره نمود ولی این طبقه بندی ها هم خیلی صحیح نمیباشد چون مثلاً روش های جغرافیایی خود ممکن است از یک روش Proactive و یا Reactive استفاده نمایند، بنابراین ما ابتدا معیارهای دسته بندی روش های مسیریابی در شبکه های Adhoc را مرور میکنیم و سپس یک دسته بندی کلی را بررسی میکنیم. از دیدگاه های مختلفی میتوان این دسته بندی های مختلف را بوجود آورد که مهمترین آنها عبارتند از :

## • از دیدگاه نوع ذخیره کردن اطلاعات مسیریابی

از این لحاظ پروتکل های مسیریابی را میتوان به دو دسته تقسیم بندی نمود که روش مسیریابی حالت لینک<sup>۱</sup> و روش مسیریابی بردار مسافت<sup>۲</sup> نام دارند. در روش مسیریابی حالت لینک اطلاعات مسیریابی در فرم بسته های حالت لینک (LSP<sup>۳</sup>) رد و بدل می شود، هر تغییر لینکی باعث ایجاد یک LSP و ارسال سیل آسای آن به کل شبکه می شود. هر گره از روی LSP های دریافتی یک توپولوژی شبکه عمومی مجازی را ترسیم کرده و نگهداری کند و همه مسیرها را به گره های دیگر محاسبه می کند.

مشکل مسیریابی حالت لینک سر بار بیش از حد آن است، زیرا در شبکه های Ad-hoc توپولوژی بسرعت در حال تغییر میباشد بنابراین باید بسته های LSP بسیار زیادی در شبکه رد و بدل شود که سربار بالایی را ایجاد مینماید.

در مسیریابی بردار مسافت هر گره یک بردار مسافت را که شامل سه مؤلفه است برای هر مقصدی نگهداری می کند، این مؤلفه ها عبارتند از: Id همه مقصدهای موجود، فاصله بر حسب تعداد hop تا هر مقصد و Nexthop به هر مقصد. هر گره به صورت پریودیک بردارهای مسافت خود را بروزرسانی می کند. مسیر کامل به یک مقصد بایک روش توزیع شده با ترکیب Nexthop گره ها روی مسیر از مبدأ تا مقصد ایجاد می شود. مشکل مسیریابی بردار مسافت همگرایی کند و تمایل به ایجاد حلقه<sup>۴</sup> می باشد. بطور کلی این نوع روش ها بیشتر در شبکه های با سیم مطرح میباشند و کاربرد دارند مانند الگوریتم های مسیریابی حالت لینک OSPF و الگوریتم مسیریابی بردار مسافت RIP.

## • از لحاظ بروز رسانی

اطلاعات مسیریابی باید روی گره های شبکه به منظور اطلاع و آگاهی از وضعیت لینک ها و توپولوژی شبکه به روز بمانند. از این منظر پروتکل های مسیریابی Ad-hoc به دو گروه طبقه بندی میشوند، یا دارای بروز رسانی دوره ای<sup>۵</sup> هستند و یا بروز رسانی بر مبنای رخداد<sup>۶</sup> دارند.

<sup>1</sup> Link state routing  
<sup>2</sup> Distance Vector Routing  
<sup>3</sup> Link State packet  
<sup>4</sup> loop  
<sup>5</sup> Periodic Update

در پروتکل های دارای بروزرسانی دوره ای اطلاعات مسیر یابی بصورت پریودیک در بازه های زمانی مشخصی پخش می شوند، که این کار پروتکل ها را ساده کرده و پایایی شبکه را پشتیبانی می کند و مهمتر از همه این که گره ها را برای آگاهی از توپولوژی و حالت شبکه آگاه می کند.

در روش بروزرسانی بر مبنای رخداد زمانی بروزرسانی لازم میشود که یک رخداد مانند قطع شدن یک لینک و یا پدیدار شدن یک لینک رخ دهد، یعنی تا زمانی که تغییری در وضعیت لینک ها و توپولوژی شبکه بوجود نیامده هیچ گونه بسته بروزرسانی ارسال نمیشود، البته در شبکه هایی که پویایی<sup>۷</sup> گره ها خیلی بالا باشد چون توپولوژی سریع تغییر پیدا میکند بسته های بروز رسانی باید ارسال شود که پهنای باند موجود را اشغال می کنند.

می توان هر دو روش فوق را با هم استفاده کرد که این مکانیزم ترکیبی<sup>۸</sup> نامیده می شود. مثلا در الگوریتم DSDV هر گره بردار مسافتش را بصورت پریودیک پخش می کند و هر زمان هم که لینکی دچار تغییر شد یک پیغام فوری بروزرسانی در شبکه منتشر میشود.

### • از لحاظ کنترل

در هر پروتکل مسیریابی باید مشخص شود که یک مسیر در کجا و چگونه محاسبه می شود. دو نوع روش کنترل برای پروتکل های مسیریابی ارائه شده است: روش کنترل متمرکز<sup>۹</sup> که در این روش هر گره اطلاعات کامل و عمومی از توپولوژی کل شبکه را نگهداری می کند بطوریکه گره هر وقت خواست، می تواند خودش مسیر به مقصد را محاسبه کند. مثل روشهای مسیریابی حالت لینک.

روش دیگر کنترل توزیع شده<sup>۱۰</sup> که در آن هر گره فقط اطلاعات جزئی و محلی در مورد توپولوژی شبکه را نگهداری می کند و وقتی که یک مسیر باید محاسبه شود تعداد زیاد گره برای محاسبه مسیر با هم همکاری می کند. مانند روش های بردار مسافت و مکانیزم کشف مسیر در روش های On-demand.

<sup>6</sup> Event driven Update  
<sup>7</sup> mobility  
<sup>8</sup> Hybrid  
<sup>9</sup> centralized Computation  
<sup>10</sup> distributed computation

## • از لحاظ نوع محاسبه مسیر

اگر از این دیدگاه روش ها را بررسی نماییم دو نوع مسیریابی مرحله به مرحله<sup>۱۱</sup> و مسیریابی از مبدأ<sup>۱۲</sup> را می بینیم که در مسیریابی مرحله به مرحله مسیر به یک مقصد در next-hop گره ها در طول مسیر توزیع می شود، وقتی گرهی یک بسته را برای یک مقصد دریافت می کند آن را به next-hop بعدی در امتداد مسیر می فرستد یعنی اینکه مبدأ از مسیر کامل تا مقصد اطلاعی ندارد و فقط بسته را تحویل گره بعدی می دهد و این فرایند تا رسیدن بسته به مقصد ادامه پیدا میکند، مشکل آن اینست که همه گره ها باید اطلاعات مسیریابی را نگهداری کنند و نیز اینکه ممکن است حلقه ایجاد شود. در روش های مسیریابی از مبدأ، گره مبدأ تمام گره های در طول مسیر را که باید توسط بسته پیموده شوند در هدر بسته داده قرار می دهد و گره های میانی فقط این بسته را بر اساس مسیری که در سرایند هر بسته قید شده هدایت و ارسال می کنند، یعنی هر گره میانی بسته را به گرهی تحویل می دهد که در هدر بسته مشخص شده است. در این روش گره های میانی نیازی به اطلاعات مسیریابی به روز ندارند زیرا که خود بسته حاوی همه اطلاعات لازم می باشد، اگر این روش با روش On-demand همراه شود نیاز به اعلان مسیر را بصورت پریودیک را از بین می برد. مهمترین نمونه از این پروتکل ها DSR میباشد.

## • از لحاظ ساختار

از دید ساختار، پروتکل های مسیریابی یا ساختار مسطح<sup>۱۳</sup> دارند و یا دارای ساختار سلسه مراتبی<sup>۱۴</sup> هستند، در ساختار مسطح همه گره ها در یک سطح از نظر اطلاعات قرار دارند، این روش برای شبکه های کوچک خوب است، در شبکه های بزرگ حجم اطلاعات مسیریابی زیاد شده و زمان زیادی برای رسیدن به یک گره دور طول می کشد.

<sup>11</sup> hop-by-hop routing

<sup>12</sup> source routing

<sup>13</sup> Flat Structure

<sup>14</sup> hierarchical Structure

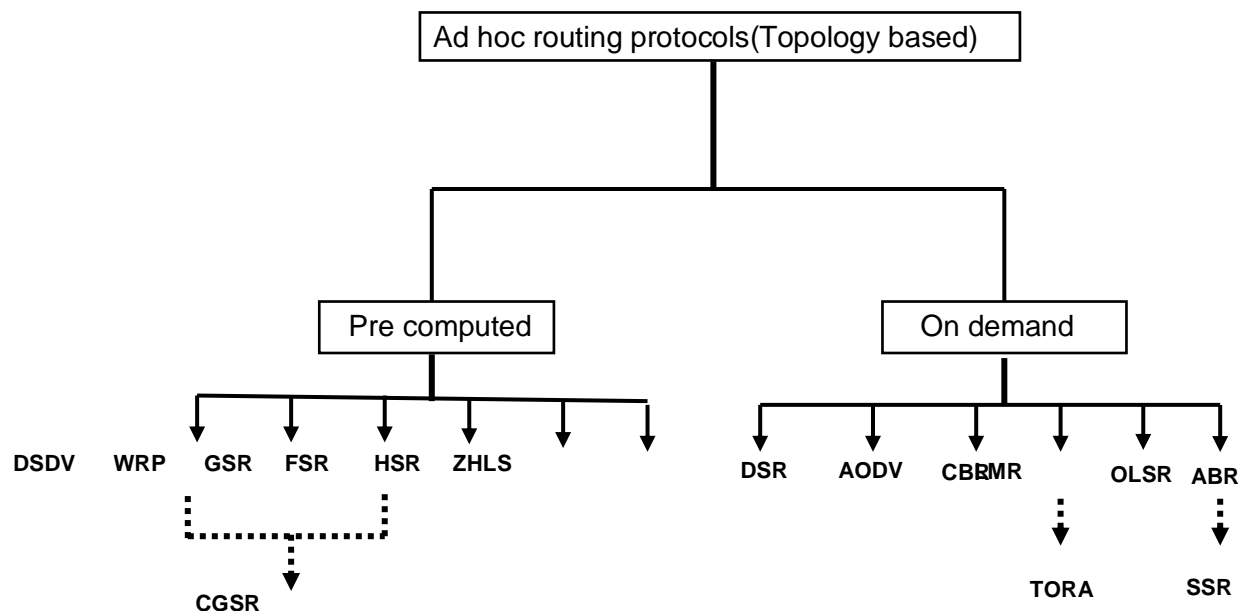
در ساختار سلسه مراتبی در شبکه های بزرگ جمعی از گره ها در شبکه بصورت پویا به قسمت هایی بنام کلاستر<sup>15</sup> افزای می شوند. از تجمع کلاسترها، کلاسترهای بزرگتر ایجاد می شوند و بهمین ترتیب سلسله مراتبی از cluster ها ایجاد میشود، سازماندهی شبکه بصورت cluster به پایایی توپولوژی شبکه کمک می کند. فقط اطلاعات سطح بالا و پایدار مثل سطح supercluster روی مسافت های دور منتشر می شود، بنابراین ترافیک کنترلی یا سربار مسیریابی به مقدار زیادی کاهش می یابد. داخل هر کلاستر گره ها اطلاعات کاملی از توپولوژی شبکه داخل کلاستر خودشان دارند و ممکن است مسیر یابی proactive استفاده شود، اگر مقصد در کلاستر دیگری باشد مسیر یابی بین کلاستر ها نیاز خواهد بود که عموماً reactive می باشد یا ترکیبی از proactive و reactive است.

## 1.1. انواع پروتکل های مسیریابی

در قسمت قبلی معیارهای مختلفی را برای طبقه بندی پروتکل های مسیریابی در شبکه های adhoc ارائه کردیم ولی در حالت کلی میتوان پروتکل های مسیریابی را به دو دسته topology based و location based تقسیم بندی نمود که ما در این فصل نوع اول را مورد بررسی قرار میدهم و در فصل سوم در مورد روش های location based صحبت خواهیم نمود. ما پروتکل های مسیریابی topology based در شبکه های Ad-hoc را از دیدگاه زمان محاسبه مسیر بررسی میکنیم، چون بیشتر مقالات هم این نوع طبقه بندی را پذیرفته اند ما هم خود را از آنها مستثنی نمیکنیم، ضمن اینکه روش های جغرافیایی که مبحث اصلی بحث ما هستند نیز از این دو نوع روش بسیار استفاده میکنند، بنابراین آنها را معرفی کرده و از هر کدام بهترین و متداولترین نمونه ها را شرح خواهیم داد. از این دیدگاه روش های مسیریابی یا Proactive هستند و یا On-demand. در شکل 1-2 یک نمودار درختی از این نوع طبقه بندی ارائه شده است، در این نمودار انواع مختلف پروتکل های مسیریابی Ad-hoc نشان داده شده است که برخی از آنها که از دیگری مشتق شده اند با خط چین مشخص شده اند. در ادامه هریک از این متدها را مورد بررسی قرار می دهیم.

## 1 + 1. پروتکل های مسیریابی Proactive

نام های دیگر این روش Precomputed و Table driven میباشد، در این روش مسیرها به تمام مقصد ها از پیش محاسبه می شوند، برای این کار گره ها باید تمامی اطلاعات جزئی در مورد وضعیت لینک ها و توپولوژی شبکه را ذخیره کنند، برای به روز ماندن اطلاعات گره ها باید بصورت پریودیک و یا در زمان تغییرات توپولوژی شبکه یا تغییر وضعیت لینک ها اطلاعات خود را بروز رسانی کنند. مزیت این روش این است که وقتی که یک مبدأ نیاز دارد که بسته ای را به یک مقصد ارسال کند، مسیر از قبل آماده و در دسترس است و دیگر نیاز به محاسبه مسیر نمیباشد که باعث بالا رفتن سرعت و کاهش تاخیر میشود ولی عیب آن اینست که برخی از مسیرها ممکن است هرگز استفاده نشوند درحالیکه از قبل مورد محاسبه قرار گرفته اند، مشکل دیگر آن این است که پخش اطلاعات مسیر یابی مقدار زیادی از پهنای باند محدود شبکه بی سیم را وقتیکه که وضعیت توپولوژی شبکه یا لینک به سرعت تغییر می کند مصرف می کند. روش های مسیریابی حالت لینک (LSR) و بردار مسافت (DVR) از نوع مسیر یابی proactive میباشند. از این نوع روش ها میتوان ZHLS[6], FSR[27], CGSR[36], WRP[39], DSDV[37] و HSR[5] را نام برد. در ادامه تعدادی از پروتکل های مسیریابی مهم Proactive شرح داده شده است.



شکل 1-4. تقسیم بندی پروتکل های مسیریابی topology based [4]

#### ▪ الگوریتم DSDV<sup>16</sup>

الگوریتم مسیریابی بردار مسافت بر مبنای شماره ترتیب مقصد یک پروتکل مسیریابی hop by hop است که از مکانیزم مسیریابی bellman-ford توسعه یافته استفاده میکند. در این روش هر گره بروزرسانی های مسیر را broadcast میکند و در آن آزادی حلقه ها تضمین میشود. هر گره یک جدول مسیریابی را نگهداری میکند که شامل همه مقصدها، تعداد hop ها تا رسیدن به هر مقصد و شماره ترتیب منطبق با گره مقصد میباشد. این شماره ترتیب برای مشخص کردن مسیر های قدیمی استفاده میشود، در واقع هر مسیری که شماره ترتیب مقصد آن

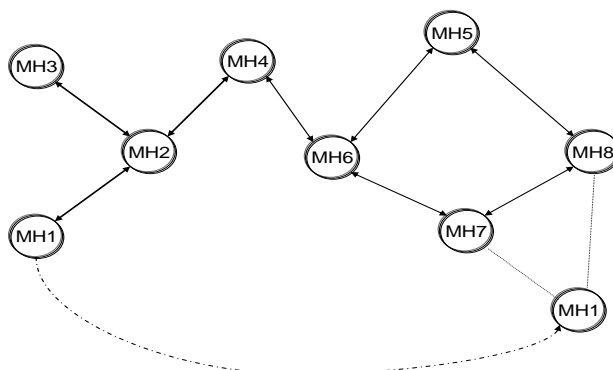


بزرگتر باشد مسیر جدیدتری است و این برای اجتناب از ایجاد حلقه استفاده میشود، به اینصورت که همیشه مسیرهایی استفاده میشود که شماره ترتیب بزرگتری دارد یعنی جدیدتر هستند. دلیل اینکه استفاده از مسیرهای قدیمی ممکن است باعث ایجاد حلقه شود اینست که چون گره ها در شبکه ad hoc همواره در حال تغییر مکان هستند یک گره ممکن طوری تغییر مکان دهد که در مسیر دارای حلقه قرار بگیرد. بازیابی های مسیر روی کل شبکه ارسال میشود، این بازیابی ها بصورت بصورت پریودیک هر بار بعد از یک بازه زمانی خاص و نیز هر زمان که تغییری در یکی از لینک های شبکه و یا توپولوژی شبکه اتفاق بیفتد انجام میگردد، یعنی هم time driven هستند و هم event driven. این ارسال بازیابی ها با افزایش مقدار شماره ترتیب همراه است یعنی هر گرهی که یک بازیابی را به کل شبکه ارسال میکند مقدار شماره ترتیب تمامی مسیرهای جدیدی را که اعلان میکند با مقدار ثابتی افزایش میدهد. وگره هایی که این بازیابی ها را دریافت میکنند بردار مسافت خود را بروز رسانی میکنند. در ضمن هر گره در جدول مسیر یابیش گره Nexthop را برای مقصدهای مختلف نگهداری میکند تا بداند که هر گره مقصد از چه طریقی قابل دسترس است. از آنجائیکه بازیابی ها باید روی کل شبکه ارسال شود امکان ایجاد ترافیک بالایی روی شبکه وجود دارد، برای کاهش دادن این ترافیک بازیابی ها دو نوع بسته را بنام بسته های fulldump و incremental بکار میبرند. در بسته های fulldump، بسته همه اطلاعات ممکن مسیر یابی را حمل میکند، یعنی این نوع بسته ها سایز بزرگی دارند ولی بسته های incremental تغییرات نسبت به آخرین باری که بسته بازیابی fulldump ارسال شده فرستاده میشود. از بسته های fulldump بندرت استفاده میشود چونکه شبکه آنقدر در یک لحظه تغییر نمیکند که نیاز به این نوع بسته باشد، مگر در شبکه هایی که پویایی در آنها بسیار بالا باشد. در شبکه های ad-hoc با پویایی معمولی بیشتر از بسته های incremental استفاده میشود و این بسته ها معمولاً در اندازه استاندارد NPDU ارسال میشوند.

وقتی گره چند مسیر را به یک مقصد دریافت میکند مسیری را انتخاب میکند که شماره ترتیب بزرگتری دارد، اگر دو مسیر شماره ترتیبشان یکسان بود مسیری انتخاب میشود که پارامتر کوچکتري داشته باشد، این پارامتر میتواند تعداد گره های روی مسیر تا آن مقصد باشد. وقتیکه یک گره مثل A مسیرش به گره همسایه ای مثل B شکسته شود، مسیر به B را با یک پارامتر نامتناهی و یک شماره ترتیب که از شماره ترتیب قبلی بزرگتر است را به کل شبکه اعلان میکند. از آنجایی که این شماره ترتیب به مقصد B بزرگترین شماره ترتیب است و با توجه به اینکه دارای پارامتر بینهایت است گره B برای سایر گره ها غیر قابل دسترس میباشد تا زمانیکه یک شماره ترتیب بزرگتر در بازیابی های بعدی اعلان شود. هر گره همچنین زمان مقیم شدن مسیر ها را نگهداری

میکند، گره ها میتوانند با به تاخیر انداختن ارسال بازیابی های مسیر ها به اندازه زمان مقیم شدن ترافیک را کاهش داده و مسیر ها را با از بین بردن ارسال هایی که قرار است اتفاق بیفتد در حالیکه مدت زمان خیلی کمی از کشف مسیر بهتر جدید گذشته است بهینه کنند.

در شکل 2-2 یک مثال از بازیابی در DSDV نشان داده شده است. در این شکل گره MH1 ابتدا در مجاورت گره MH2 قرار دارد، پس از مدت زمان کوتاهی بعثت پویایی شبکه این گره تغییر مکان داده و در مجاورت گره های MH7, MH8 قرار میگیرد و اتصالش با گره MH2 بطور مستقیم قطع میگردد. در شکل جدول مسیر یابی گره MH4 برای هر دو حالت قبل و بعد از حرکت نشان داده شده است. همانطور که در شکل مشاهده میشود گره MH4 به هریک از گره های دیگر مسیری دارد که با یک شماره ترتیب مشخص شده است و پارامتری هم برای هر مقصد در نظر گرفته شده است. پس از اینکه گره MH1 تغییر مکان داده و از مجاورت MH2 دور شده و به مجاورت MH7, MH8 در میاید مسیرها دیگر مسیر های سابق نیستند و بدرد نمیخورند، بنا بر این گره جدول مسیر یابیش را بازیابی کرده و برای اینکه مسیرهای جدید از مسیر های قدیمی قابل تشخیص باشند مقدار شماره ترتیب را نسبت به هر مقصد به اندازه 110 واحد افزایش میدهد، مثلاً مسیرش به گره MH8 قبل از تغییر مکان دارای شماره ترتیب 50 بوده و پس اینکه تغییر مکان میدهد مسیر جدیدش به این گره دارای شماره ترتیب 160 میباشد و به این ترتیب کلیه بازیابیهای دیگر هم انجام میشود.



Destination	Metric	Seq Number
MH1	2	S406-MH1
MH2	1	S128-MH2
MH3	2	S564-MH3
MH4	0	S710-MH4
MH5	2	S392-MH5
MH6	1	S076-MH6
MH7	2	S128-MH7
MH8	3	S050-MH8

جدول مسیریابی اعلان شده توسط گره MH4

Destination	Metric	Seq Number
MH4	0	S820-MH4
MH1	3	S516-MH1
MH2	1	S238-MH2
MH3	2	S674-MH3
MH5	2	S502-MH5
MH6	1	S186-MH6
MH7	2	S238-MH7
MH8	3	S160-MH8

جدول مسیریابی بروز شده توسط گره MH4

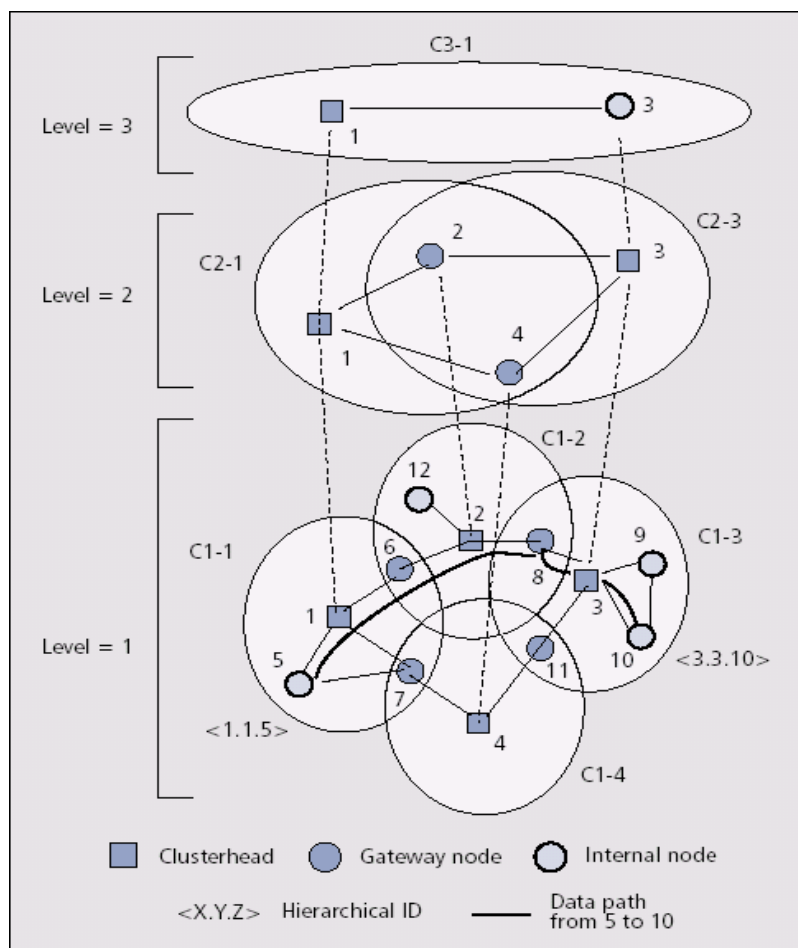
شکل 1 2. جدول مسیریابی یک گره در روش DSDV در دو زمان متوالی [37]

#### ▪ مسیریابی حالت سلسله مراتبی (HSR<sup>17</sup>)

ویژگی روش های HSR [5] دسته بندی چند سطحی<sup>18</sup> دسته بندی و تقسیم بندی منطقی گره های متحرک است، شبکه به دسته<sup>19</sup> ها تقسیم بندی شده و یک سردهسته با یک الگوریتم انتخاب سردهسته برگزیده میشود. در HSR این سردهسته ها از این پس خودشان را در دسته سازماندهی می کنند. گره های یک دسته اطلاعات لینک خود را به یکدیگر broadcast میکنند. سردهسته اطلاعات کلاستر خود را خلاصه کرده و به سردهسته های همسایه از طریق دروازه<sup>20</sup> ارسال می کند. همچنانکه در شکل 2-3 نشان داده شده این سردهسته ها عضو کلاستری در یک سطح بالاتر هستند و اطلاعات لینک های خود را مطابق اطلاعات خلاصه شده سطح پائین تر بین یکدیگر ردوبدل می کنند.

یک گره در هر سطح اطلاعاتی را که بعد از اجرای الگوریتم در آن سطح بدست آورده بصورت سیل آسا<sup>21</sup> به سطح پائین تر خود ارسال می کند. سطح پائین تر هم اطلاعات توپولوژی سلسله مراتبی دارد. هر گره یک آدرس سلسله مراتبی دارد، یک راه برای متناظر کردن آدرس سلسله مراتبی شماره دسته با شروع از ریشه به آن گره مطابق شکل 2-3 می باشد.

<sup>17</sup> Hierarchical State Routing  
<sup>18</sup> multi level clustering  
<sup>19</sup> cluster  
<sup>20</sup> Gateway  
<sup>21</sup> Flooding



شکل 1-3. یک نمونه سلسله مراتب در روش HSR

یک دروازه توسط ریشه از طریق یک، چند مسیر قابل دسترسی است، بطوریکه دروازه میتواند یک یا چند آدرس سلسله مراتبی داشته باشد. یک آدرس سلسله مراتبی برای تحویل از هر جای شبکه به میزبان کافی می باشد. بعلاوه گره ها همچنین به زیر شبکه های منطقی تقسیم بندی شده اند و هر گره به یک آدرس منطقی متناظر شده است. هر زیر شبکه یک سرور مدیر محلی<sup>۲۲</sup> دارد، همه گره های زیر شبکه آدرس منطقی خود را

با سرور مدیر محلی ثبت می کنند و او هم آدرس سلسله مراتبی آنها را به سطوح بالاتر اعلام کرده و اطلاعات همچنین برای تمام سرورهای مدیر محلی هم پایین فرستاده می شود.

لایه انتقال<sup>۲۳</sup> بسته ای را با آدرس منطقی مقصد به لایه شبکه می فرستد، لایه شبکه آدرس سلسله مراتبی سرور مدیر محلی مقصد را از سرور مدیر محلی خودش بدست می آورد و سپس بسته را به آن میفرستد و سرور مدیر محلی مقصد بسته را به مقصد ارسال می کند. چونکه از این پس مبدا و مقصد آدرس سلسله مراتبی همدیگر را می دانند می توانند سرور مدیر محلی را کنار گذاشته و خودشان مستقیماً برای هم ارسال کنند. از آنجائیکه آدرس منطقی / آدرس سلسله مراتبی برای مسیریابی استفاده می شود برای نقل انتقالات شبکه هم می تواند تطبیق پذیر باشد.

## 1 + 2. پروتکل های مسیریابی بر مبنای تقاضا

این روش ها که به روش های On demand معروف هستند یک مشی کند برای مسیریابی دارند، در این متدها مسیر به یک مقصد از قبل مشخص نمی باشد در تقابل با پروتکل های مسیریابی Proactive همه مسیرهای بروز، توسط همه گره ها نگهداری نمیشود. در هر زمان که برای ارسال داده به مسیری نیاز باشد، آن مسیر ایجاد میشود. زمانی که یک مبدا نیاز به ارسال به مقصدی دارد یک مکانیسم کشف مسیر را برای پیدا کردن یک یا چند مسیر به مقصد آغاز میکند، این پروسه کشف مسیر<sup>۲۴</sup> نامیده می شود. بعد از پیدا کردن مسیرها، مبدأ بسته ها را روی این مسیرها می فرستد، مسیر تا زمانی که مقصد قابل دسترسی باشد یا مسیر کوتاهتری پیدا شود معتبر خواهد بود.

در حین انتقال بسته ممکن است مسیر شکسته شود ( دچار قطعی شود )، چون که گره های روی مسیر ممکن است حرکت کنند و یا تغییر حالت دهند. مسیر شکسته شده باید دوباره ساخته شود، پروسه کشف شکستگی مسیر و بازسازی آن نگهداری مسیر<sup>۲۵</sup> نامیده می شود.

Transport layer<sup>23</sup>  
route discovery<sup>24</sup>  
Route Maintenance<sup>25</sup>

مزیت اصلی این گونه روش ها این است که پهنای باند با ارزش شبکه های بی سیم Ad-hoc به خوبی و در حد بالایی ذخیره می شود، زیرا پهنای باند مصرفی در تبادل اطلاعات مسیر یابی را با نگهداری مسیرها فقط برای مقصدهایی که گره نیاز به ارسال ترافیک به آنها را دارد محافظت و محدود میکند. روش ondemand همچنین نیاز به پخش پریودیک اطلاعات مسیر یابی بصورت broadcast و یا flooding را در زمان تغییرات حالت لینک از بین می برد. مشکل این روش این است که مقداری تاخیر در شروع ارسال بوسیله مکانیسم کشف مسیر (route discovery) ایجاد می شود. نام دیگر این روش مسیریابی reactive میباشد. از این نوع روش ها میتوان DSR,[1] OLSR,[15] LMR,[35] ABR و AODV[2] را نام برد، در ادامه نمونه های مختلفی از روش مسیریابی بر اساس تقاضا شرح داده میشود.

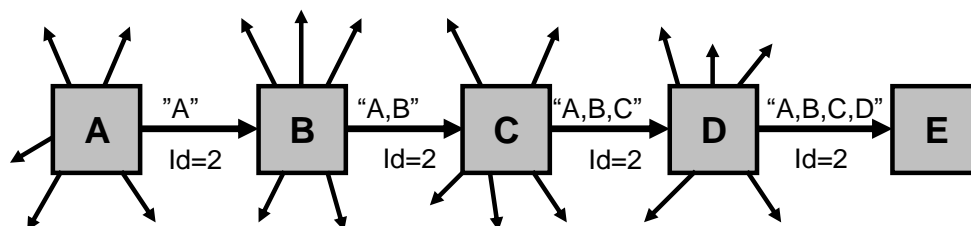
## 26 DSR

پروتکل مسیریابی از مبدا پویا (DSR) جزو روش های شناخته شده مسیریابی MANET میباشد، این روش از دو مکانیزم که با هم برای کشف و نگهداری مسیر های مبدا در شبکه Ad-hoc کار می کنند ترکیب شده است، کشف مسیر مکانیزمی است که توسط یک گره مبدا S که می خواهد برای ارسال یک بسته به یک مقصد D مسیری را به آن مقصد بدست آورد انجام می شود. کشف مسیر فقط زمانی انجام می شود که گره S می خواهد بسته را به گره D بفرستند و قبلا مسیری را به D در اختیار ندارد و به عبارت ساده تر مسیری را به D بلد نیست. نگهداری مسیر مکانیزمی است که برای نگهداری و بازسازی مسیرهای مورد استفاده بکار میرود. در حالیکه یک مسیر از مبدا S به مقصد D مورد استفاده است توپولوژی شبکه ممکن است تغییر کند (این حرکت گره ها و تغییر مداوم توپولوژی، ماهیت شبکه های Ad-hoc است) بطوریکه گره مبدا دیگر نتواند مسیر به مقصد را استفاده کند بدلیل اینکه یک لینک در طول مسیر به مقصدممکن است دیگر کار نکند. وقتی روال نگهداری مسیر نشان می دهد که یک مسیر مبدا شکسته شده است، S می تواند تلاش کند که مسیر دیگری را که بلد است یعنی در حافظه خود دارد برای ارسال به D استفاده کند و یا اینکه دوباره مکانیزم کشف مسیر را بکار ببرد تا یک مسیر جدید کشف کند. مکانیزم نگهداری مسیر فقط زمانی استفاده می شود که S واقعا در حال ارسال بسته ها به D باشد.

هر دو مکانیزم کشف مسیر و نگهداری مسیر کاملاً بصورت On\_demand کار می کنند، در واقع بر خلاف بسیاری از پروتکل های دیگر Ad-hoc در روش DSR هیچ نوع بسته ای در هیچ سطحی بصورت پریودیک در شبکه ارسال نمی شود و این رفتار On\_demand و کمبود فعالیت های پریودیک باعث کم شدن بسته های سرطپو می شود. در پاسخ به یک کشف مسیر یکتا یک گره چندین مسیر را به یک مقصد یافته و ذخیره می کند. این باعث می شود که عکس العمل نسبت به تغییرات مسیر یابی بسیار سریع شود، از آنجائیکه اگر مسیر مورد استفاده با خرابی مواجه شد مسیر دیگری را مورد استفاده قرار میدهد، در ضمن این ذخیره کردن چند مسیر باعث می شود که از سر بار مورد نیاز برای اجرای یک کشف مسیر جدید زمانی که مسیر مورد استفاده خراب می شود جلوگیری شود. عملیات های کشف و نگهداری مسیر در DSR طوری طراحی شده اند که توسط هر دو نوع لینک یک طرفه و لینک های متقارن پشتیبانی می شوند. DSR می تواند ارتباط بین گره های با شعاع های متفاوت را پشتیبانی نماید، بعلاوه مسیر یابی DSR بر مبنای مسیر یابی استاندارد اینترنت است که یک گره دروازه<sup>۲۷</sup> هم به اینترنت متصل است و هم عضوی از گره های در گیر در پروتکل مسیر یابی شبکه Ad-hoc میباشد. همچنین با مسیریابی IP متحرک هم سازگار است که یک گره دروازه نقش یک Agent خارجی IP متحرک را انجام می دهد.

کشف مسیر در DSR به اینصورت انجام میشود که مبدا S بسته ای را برای مشخص کردن مقصد D ایجاد می کند، در هر بسته یک مسیر شامل hop هایی که بسته باید از آنها برای رسیدن به مقصد D عبور کند را قرار می دهد. S یک مسیر مبدا مناسب را با جستجو در حافظه موقت مسیر خود (که شامل مسیر هایی است که از قبل میداند) بدست می آورد ولی اگر مسیری در حافظه اش به این مقصد پیدا نکرد پروتکل "کشف مسیر" را برای پیدا کردن پویای یک مسیر به مقصد D آغاز می کند، که S آغاز کننده و D هدف کشف مسیر است.

در شکل 2-4 یک مثال از کشف مسیر توضیح داده شده است، گره A تلاش می کند که یک مسیر به گره E بدست آورد، برای آغاز مکانیزم کشف مسیر گره A یک پیغام در خواست مسیر بصورت یک بسته Broadcast به همه گره های که در محدوده ارسالش هستند میفرستد، هر بسته در خواست مسیر گره آغاز کننده و گره هدف کشف مسیر را مشخص کرده و هم چنین حاوی رکوردی از لیست هر گره میانی است که یک کپی از پیغام در خواست مسیر به آن ارسال شده است این رکورد با یک لیست خالی توسط آغاز کننده کشف مسیر مقدار دهی اولیه می شود و هر گرهی که آنرا دریافت میکند نام خود را به آن اضافه میکند.



**شکل 1-4.** مکانیزم کشف مسیر در DSR : گره A سعی در بدست آوردن مسیری به E دارد

وقتی گره دیگری در خواست مسیر را دریافت می کند اگر خودش گره مقصد باشد یک پیغام پاسخ مسیر را به آغاز کننده کشف مسیر بر می گرداند، یک کپی از رکورد لیست گره های مسیر انباشته شده بر می دارد و این مسیر را در حافظه مسیرش برای ارسال متوالی بسته ها به این مقصد ذخیره می کند، در حالت دیگر یعنی اگر خودش مقصد نباشد اگر گرهی که در خواست مسیر را دریافت می کند قبلا بسته ای با ID یکسان با این بسته در یافت کرده و یا خودش در رکورد لیست مسیر های انباشته این بسته است آنرا دور می ریزد و گرنه آدرس خود را به آن اضافه کرده (در رکورد مسیر در پیغام درخواست مسیر) و آنرا به صورت یک بسته Broadcast محلی ارسال می کند (با ID در خواست یکسان).

در بازگرداندن پاسخ مسیر به آغاز کننده کشف مسیر، مثلا از E به A در شکل 2-4 گره E معمولا حافظه مسیر خود را بررسی خواهد کرد که ببیند آیا مسیری به A وجود دارد و اگر بود آنرا برای برگرداندن پاسخ مسیر به گره A مورد استفاده قرار می دهد، در غیر این صورت گره E ممکن است برای پیدا کردن گره A یک مکانیزم کشف مسیر را استفاده کند ولی برای اجتناب از بازگشت نامتناهی احتمالی کشف مسیر ها باید به صورت Piggybacking این پاسخ مسیر را روی درخواست مسیر A سوار کند و برگرداند، حتی می توان بسته های داده کوچک دیگر مثل بسته TCP-SYN را روی بسته درخواست مسیر با استفاده از یک مکانیزم مشابه سوار کرد. گره E می تواند به سادگی زنجیره hop های ثبت شده در لیست مسیر را معکوس کند و سعی کند پاسخ مسیر را روی آن بفرستد و این مسیر را برای حمل بسته پاسخ مسیر خود استفاده کند.

پس از آغاز یک کشف مسیر فرستنده بسته یک کپی از بسته اصلی را در یک بافر محلی به نام بافر ارسال ذخیره می کند، بافر ارسال حاوی یک کپی از هر بسته ای است که نتوانسته توسط این گره ارسال شود زیرا تا

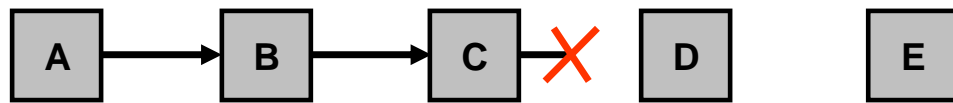


کنون مسیر از مبدای را برای ارسال به مقصد بسته ها نداشته است. هر بسته در بافر ارسال با زمان جاگذاری در بافر برچسب می خورد و بعد از ماندن در بافر به مدت دوره انقضای زمانی<sup>۲۸</sup> دور انداخته میشود. اگر لازم شود که از سرریز شدن بافر جلوگیری شود یک FIFO یا استراتژی دیگری می تواند برای خارج کردن بسته ها قبل از انقضای زمان آنها استفاده شود. در حالی که یک بسته در بافر ارسال قرار دارد گره باید گاه و بیگاه یک کشف مسیر جدید را برای آدرس مقصد بسته ها انجام دهد.

### اصول نگهداری مسیر در DSR

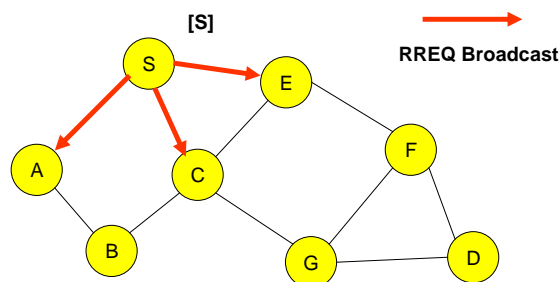
وقتی که یک بسته با استفاده از یک مسیر مبدای ارسال شد هر گره ای که بسته را انتقال می دهد باید مطمئن شود که بسته بوسیله Next\_hop بعدی در طول مسیر دریافت شده است یعنی تصدیق و تأیید رسیدن بسته را دریافت کند. به عنوان مثال در شکل 2-4 گره A یک بسته را برای گره E با استفاده از مسیر مبدای از طریق گره های میانی B, C, D ارسال کرده است در این حالت گره A مسئول است که رسید دریافت بسته را توسط B دریافت کند و B مسئول است که رسید دریافت را توسط C دریافت کند و بقیه هم تا مقصد به همین صورت عمل میکنند، این تأیید دریافت در بسیاری از موارد ممکن است بدون هیچ هزینه ای در DSR انجام شود چه به عنوان یک قسمت استاندارد از پروتکل MAC ای که استفاده می شود (مثل فریم های Ack سطح لینک که بوسیله IEEE... تعریف شده) و چه با یک Acknowledgement غیر فعال (مثلا B تأیید را در C با گوش فرا دادن به ارسال C، زمانی که بسته را برای فرستادن به D روی خط ارسال می کند دریافت کند). اگر هیچ کدام از این دو مکانیزم آماده نبود گره احتمالاً بسته را با Set کردن یک بیت در هر بسته برای درخواست یک ACK نرم افزاری ویژه DSR که برگشت داده شود تحویل گره بعدی<sup>۲۹</sup> می دهد. اگر لینک بین دو گره یک جهته باشد این ACK نرم افزاری ممکن است روی مسیر دیگری که حتی شاید شامل چندین hop باشد برگردانده شود. اگر گره به اندازه ما کزیمم تعداد بار ممکن بسته را ارسال کند ولی تأییدی دریافت نکند یک پیغام خطای مسیر برای فرستند اصلی ارسال می شود به نشانه اینکه یک لینک در روی مسیر نمی تواند عمل ارسال را انجام دهد. به عنوان مثال در شکل 2-5 اگر C نتواند به خاطر شکستگی لینک بسته را Next\_hop یعنی D تحویل دهد، C یک پیغام خطای مسیر به A بر می گرداند. گره ها این مسیر شکسته شده را از حافظه خود پاک می کنند،

اگر A در حافظه اش مسیر دیگری به E داشته باشد بسته را با استفاده از این مسیر جدید فوراً ارسال می کند و گرنه ممکن است یک کشف مسیر جدید برای این هدف انجام دهد.

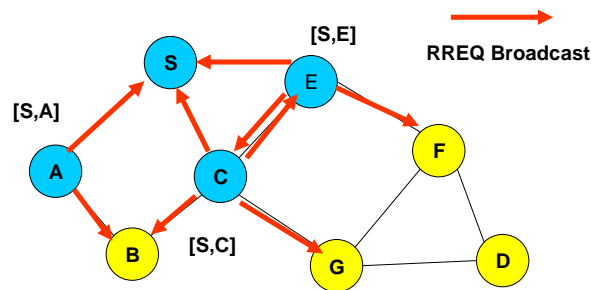


**شکل 1-5.** نمونه ای از نگهداری مسیر در DSR که گره C قادر به ارسال بسته A به E نیست

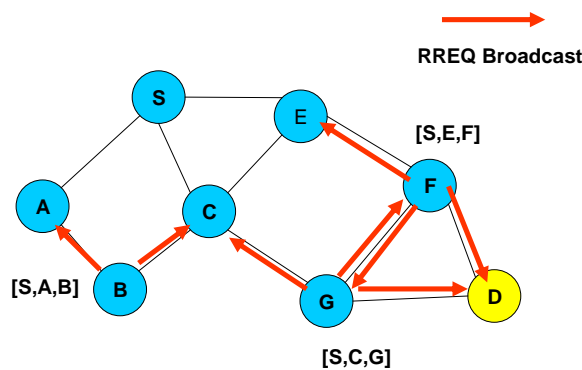
**مثال:** شکل 2-6 را در نظر بگیرید، فرض کنیم گره S می خواهد بسته ای را برای گره D ارسال نماید و در حافظه اش مسیری از قبل آماده به D ندارد، بنا براین مکانیزم کشف مسیر را آغاز میکند، همانطور که در شکل 2-6 قسمت الف مشاهده میشود، گره S یک بسته درخواست مسیر به همسایه های مستقیمش، یعنی A, C و E ارسال میکند که در هدر آن یک لیست خالی مسیر نیز قرار داده میشود، هر یک از این گره ها با دریافت بسته چونکه خودشان مقصد نیستند و نیز مسیر مستقیمی به مقصد هم ندارند نام خود را به بسته اضافه نموده و آنرا برای همسایه های مستقیمشان ارسال مینمایند که در شکل 2-6 قسمت ب نشان داده شده است، همانطور که در این شکل مشاهده میشود بعضی از گره ها مثل E و C ممکن است دو بار یا بیشتر یک بسته درخواست مسیر را دریافت کنند، که در اینصورت آنرا دور می اندازند.



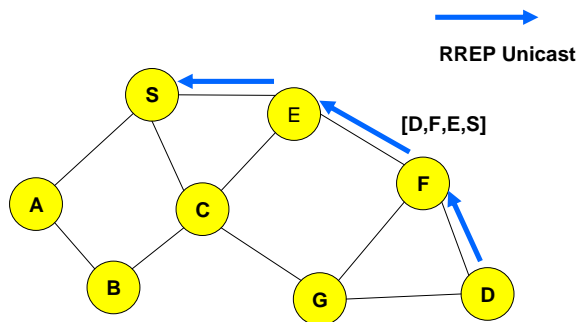
(الف)



(ب)



(ج)



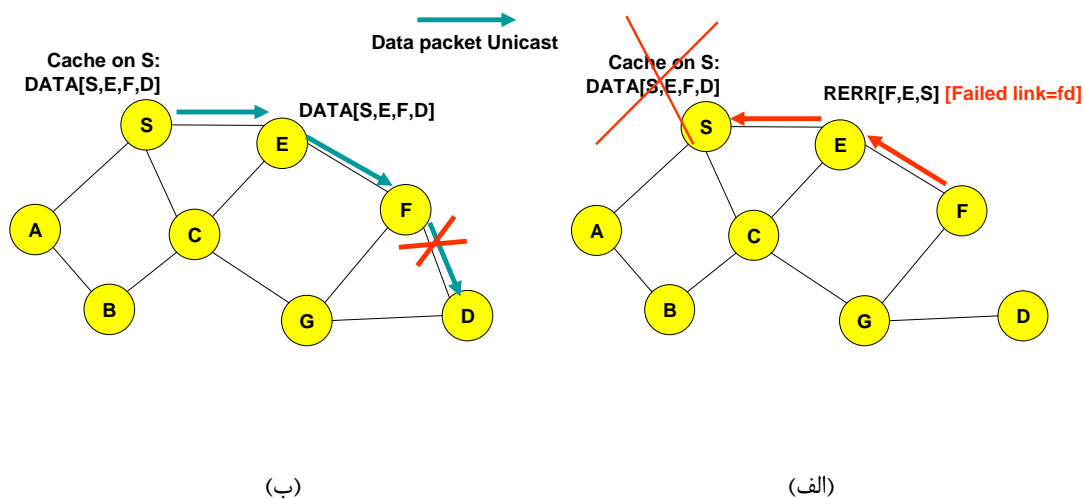
(د)

شکل 1 6. گونه ای از مکانیزم کشف مسیر DSR

عمل ارسال به همسایه ها تا زمانیکه یک گره یا خودش مقصد باشد و یا راهی به مقصد داشته باشد ادامه پیدا میکند و در مرحله سوم به B و سپس در مرحله چهارم به گره های F و G میرسد که هر دوی آنها مسیری به مقصد D دارند، بنابراین پیغام پاسخ مسیر روی همان مسیری که پیغام درخواست مسیر آمده برگردانده میشود،

مثلا از مسیری که در شکل 2-6 قسمت د نشان داده شده است، ذکر این نکته ضروری است که مبدا S ممکن است چندین پاسخ مسیر را دریافت کند در اینصورت از بین آنها بهترین مسیر را که میتواند کوتاهترین مسیر باشد انتخاب کرده و بقیه مسیرها را در جدول مسیر یابیش (در حافظه اش) ذخیره میکند تا اگر مسیر جاری دچار شکستگی لینک شد و خراب شد از آنها برای جایگزینی آن استفاده نماید.

حال فرض کنید که لینک گره F به گره D دچار شکستگی شده است و دیگر قابل استفاده نیست در اینصورت گره F این شکستگی میسر را فهمیده و برای اینکه مبدا متوجه این خرابی لینک شود یک پیغام خطا را به طرف مبدا از همان مسیر برعکس ارسال می فرستد و گره مبدا یعنی S با دریافت این پیغام خطا در حافظه اش تمام مسیرهایی را که شامل این لینک هستند حذف میکند و اگر به غیر از آنها مسیر دیگری وجود داشت که از طریق آنها ارسال میکند و گرنه دوباره یک مکانیزم کشف مسیر جدید را آغاز میکند تا مسیر تازه ای را به مقصد گره D پیدا کند و دوباره داده را به آن ارسال نماید. این فعل و انفعالات در شکل 2-7 الف و شکل 2-7 ب نشان داده شده است.



شکل 1.7. فاز نگهداری مسیر در DSR

## ▪ TORA:

TORA یک الگوریتم مسیر یابی بسیار قابل تطبیق، موثر، قابل توسعه و توزیع یافته بر مبنای مفهوم معکوس کردن لینک<sup>۳۰</sup> میباشد، TORA برای شبکه های بی سیم متحرک بسیار دینامیک Moltihop پیشنهاد شده است و یک پروتکل مسیریابی بر مبنای تقاضای با شروع از گره مبدا میباشد، این روش چندین مسیر را از گره مبدا به یک گره مقصد پیدا میکند. ویژگی اصلی TORA این است که پیغام های کنترلی برای یک مجموعه خیلی کوچک از گره های نزدیک به تغییرات توپولوژی متمرکز میشوند. برای بدست آوردن این هدف، گره ها اطلاعات مسیریابی در مورد گره های مجاور را نگهداری میکنند.

پروتکل سه تابع اصلی دارد: ایجاد مسیر<sup>۳۱</sup>، نگهداری مسیر<sup>۳۲</sup> و پاک کردن مسیر<sup>۳۳</sup>.

هر گره یک 5 تایی دارد که از اجزاء زیر تشکیل شده است:

1- زمان منطقی خراب شدن لینک

2-ID یکتای گرهی که در سطح ارجاعی جدید تعیین شده است.

3- یک بیت نشان دهنده انعکاس

4- یک پارامتر سفارش دهی انتشار (height)

5-ID جاری یکتای خود گره

سه المان اول جمعا سطح ارجاع را بیان میکنند یک سطح ارجاع جدید هر زمان که یک گره لینک ورودی خود را بسته به یک خرابی از دست بدهد رخ میدهد. دو مقدار آخر یک delta را با توجه به سطح ارجاع تعریف میکنند.

ایجاد مسیر با استفاده از بسته های QRY و UPD انجام میگردد. الگوریتم ایجاد مسیر با height مقصد که به مقدار صفر تنظیم شده و height سایر گره ها که مقدار NULL دارد آغاز میشود، مبدا یک بسته QRY را با استفاده از ID گره مقصد که در آن بسته قرار داده شده است برای همه همسایه ها پخش میکند، یک گره با

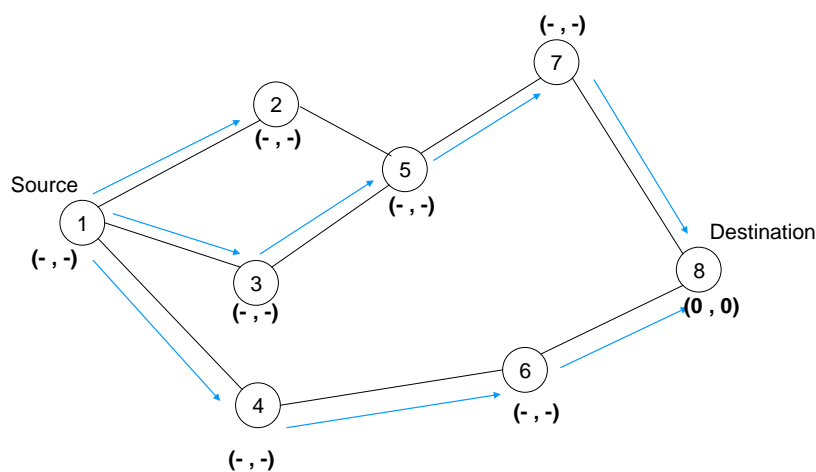
<sup>30</sup> Link Reversal

<sup>31</sup> Route Creation

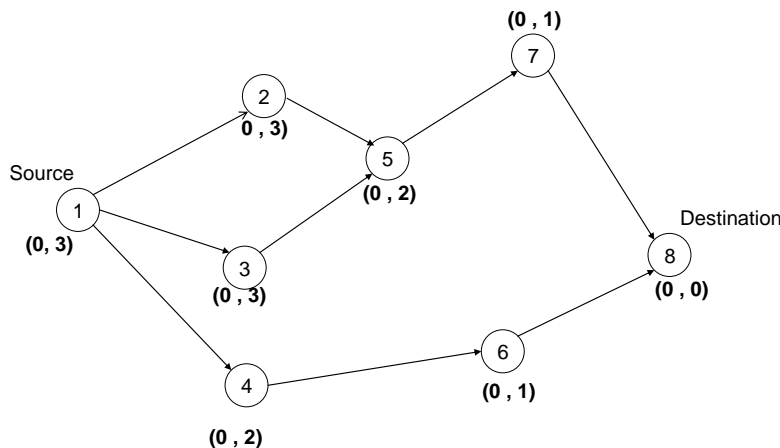
<sup>32</sup> Route Maintenance

<sup>33</sup> Route erasure

یک height غیر تهی با یک بسته UPD پاسخ میدهد که مقدار height در آن قرار دارد. گرهی که UPD رادریافت میکند مقدار height خود را با یک مقدار بیشتر از گرهی که UPD را تولید کرده تنظیم میکند. گره با height بزرگتر به عنوان Upstream در نظر گرفته میشود و گره با height کمتر بعنوان Downstream در نظر گرفته میشود. به این ترتیب یک گراف بدون حلقه مستقیم از مبدا تا مقصد ساخته میشود. شکل 2-8 یک رویه ساخت مسیر در TORA را نشان میدهد. در این شکل در قسمت الف گره 1 مبدا و گره 2 مقصد است، گره 1 پیام های Query را برای همسایه هایش میفرستد و این عمل تکرار میشود تا گرهی که مسیری به مقصد دارد با یک Height پاسخ دهد.



(الف)

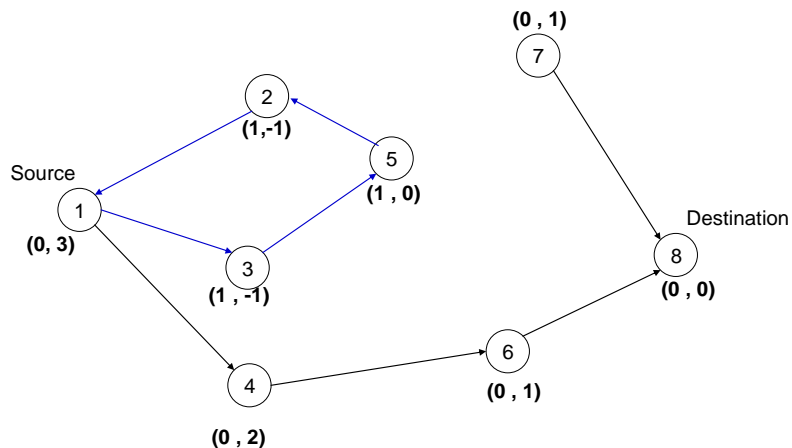


(ب)

شکل 1-8. نمونه ای از مکانیزم ایجاد مسیر در TORA [7]

همچنانکه در شکل 2-8 قسمت الف نشان داده شده گره پنج از گره سه هیچ گزارش منتشر شده ای را دریافت نمیکند چون قبلاً از گره دو دریافت کرده است. در شکل قسمت ب مبداء (مثلاً گره 1) ممکن است یک بسته UDP از گره دو یا گره سه دریافت کند، ولی از آنجایی که گره چهار مقدار height کمتری را ارائه میدهد آن را پذیرفته و height آنرا حفظ میکند. وقتی که گره حرکت میکند و تغییر مکان میدهد مسیر DAG شکسته میشود و به نگهداری مسیر برای ایجاد دوباره یک DAG برای همان مقصد نیاز است. وقتی که آخرین لینک Downstream یک گره خراب میشود یا سطح مرجع جدید را تولید میکند، این باعث انتشار سطح مرجع گره ها با همسایه ها مثل شکل 2-9 می شود.

لینک ها برای بازتاب تغییر در بازیابی ها برای سطح مرجع جدید برگشت داده میشود (معکوس میشوند). این ممکن است تاثیر مشابهی در معکوس کردن در جهت یک یا چند لینک، زمانی که یک گره لینک downstream ندارد داشته باشد. در فاز پاک کردن مسیر، TORA بسته های clear را روی شبکه برای از بین بردن مسیرهای نامعتبر پخش میکند.



شکل 1-9. نگهداری مسیر در TORA

در TORA پتانسیل رخ دادن نوسانات وجود دارد بخصوص وقتی چندین مجموعه از هماهنگ کردن گره ها بطور همزمان عملیات تشخیص تکه تکه شدن شبکه<sup>۳۴</sup>، پاک کردن مسیر و ساخت مسیرهای جدید بر مبنای یکدیگر ایجاد شود. چون در TORA هماهنگی بصورت میان گرهی انجام میشود مشکل بی ثباتی و نوسان آن شبیه مشکل حلقه بی نهایت<sup>۳۵</sup> در پروتکل های بردار مسافت است، با این تفاوت که برخی نوسانات زود گذر هستند و همگرایی در نهایت اتفاق می افتد.

detecting partitioning<sup>34</sup>  
count\_to\_infinity<sup>35</sup>



## خلاصه

الگوریتم های مسیریابی topology based در شبکه های Ad-hoc به دو دسته proactive و on demand دسته بندی میشوند، روش های on demand در شبکه های با پویایی بالا نرخ loss کمتری نسبت به روش های proactive دارند در حالیکه در پویایی پایین این موضوع بالعکس است، در مورد تاخیر در روش های proactive چون اغلب اطلاعات لازم مسیریابی از قبل ارسال شده است مسلماً تاخیر کمتری نسبت به روش های on demand دارند.

در زمینه سربار مسیریابی چون در روش های proactive ارسال های پریشودیک داریم، سربار بالاتری هم مشاهده میشود که البته ثابت است و اگر ملاک ارزیابی را تراکم گره های شبکه در نظر بگیریم میتوان گفت که با افزایش تراکم شبکه پروتکل های on demand کارایی تاخیر را بهتر نگه میدارند، چون ارسال های پریشودیک ندارند. در مجموع میتوان گفت که روش های on demand روش های بهتری بنظر میرسند.