

الگوریتم و پروتکل ساخت درختهای بهینه در multicast*

مظفر بگ محمدی[†]، سیاوش صمدیان برزکی[†] و ناصر یزدانی[×]

[†]{mozafarb, s.samadian}@ece.ut.ac.ir

[×]nasyaz@sofe.ece.ut.ac.ir

آزمایشگاه روتر، گروه برق و کامپیوتر دانشکده فنی

دانشگاه تهران

چکیده

در این مقاله ابتدا مشکل بهینه نبودن درخت (SPT(Shortest Path Tree))، از لحاظ پهنای باند مصرفی درخت تولید شده بررسی شده و یک پروتکل جدید به اسم NRBP(Near-Receiver Branching Point) معرفی و طراحی می گردد. درخت تشکیل شده توسط NRBP، ضمن حفظ مزیت اصلی درخت SPT (یعنی استفاده از کوتاهترین مسیر برای توزیع داده multicast)، پهنای باند کمتری مصرف می کند. همچنین درخت SPT ساخته شده در مسیر مستقیم می باشد. پروتکل NRBP قادر به ساخت درخت Steiner نیز می باشد. درخت Steiner تولید شده توسط NRBP ضمن بهینه بودن پهنای باند مصرفی دارای تأخیر کمتری است. می توان با تغییر پارامترهای پروتکل NRBP محدود و وسیعی از درختها را بین درخت SPT و درخت Steiner درست نمود. راه حلهای ارائه شده برای multicast هر کدام پارامتر یا پارامترهای خاصی را بهبود می دهند، که می توان به پارامترهای تأخیر، پهنای باند، امنیت گروه، منابع مصرفی مثل اندازه جدول multicast و یا تعداد حالت های مصرفی، پیچیدگی الگوریتم، سازگاری با روترهای قدیمیتر، کنترل دسترسی و... اشاره کرد. پروتکل NRBP ضمن اینکه پهنای باند مصرفی و تأخیر را بهینه می کند از لحاظ منابع مصرفی مثل اندازه جدول multicast و یا تعداد حالت های مصرفی و پیچیدگی الگوریتم نیز وضعیت مطلوبی دارد، توانایی در پشتیبانی از QoS نیز از دیگر مشخصات ارزشمند روش پیشنهادی می باشد.

۱. مقدمه

در بسیاری از کاربردها لازم است که داده ها از یک فرستنده به چندین گیرنده در شبکه ارسال گردد. multicast در واقع تکنیکی است که ما را قادر می سازد تا با استفاده از ارسال یک بسته multicast به شبکه، داده های مورد نظر را به صورت کارا به چندین مقصد که در واقع یک گروه multicast را تشکیل می دهند برسانیم. اگر بخواهیم همین کار را با استفاده از unicast انجام دهیم لازم است که فرستنده به هر یک از گیرنده ها، بصورت جداگانه از هر بسته یک کپی ارسال نماید. این کار علاوه بر اتلاف منابع شبکه، موجب تأخیر متفاوت در دریافت داده در گیرنده ها می گردد. بدیهی است که فواید multicast در این مورد

اتلاف کمتر پهنای باند و منابع، ایجاد موازی سازی در شبکه، کاهش بار فرستنده و کاهش ترافیک شبکه می باشد[5].

از جمله کاربردهای multicast می توان به کنفرانسهای صوتی و تصویری (مانند نشستهایی که در IETF برگزار می شوند)، پخش برنامه های تلویزیونی و رادیویی و آموزش از راه دور اشاره کرد. همچنین، چون نقشه های آب و هوایی در هواشناسی دارای حجم اطلاعاتی بالایی هستند و در یک روز ممکن است چند بار تغییر کنند، اگر از multicast استفاده نکنیم، رد و بدل کردن آنها بین پایگاههای هواشناسی هزینه بسیار بالایی را بر شبکه تحمیل خواهد کرد. علاوه بر این موارد می توان از به روز کردن تمام کپی های یک پایگاه داده ها یا جستجو کردن در تمام کپی های آن، توزیع نسخه جدید یک نرم افزار و سرویسهای اطلاعاتی بورس سهام نام برد[9,12].

پروتکل DVMRP یکی از پروتکل هایی است که از درخت SPT برای پخش داده multicast استفاده می کند و در اواسط دهه ۹۰ بطور وسیع در Mbone(Multicast Backbone) مورد استفاده قرار گرفت[13]. دو پروتکل دیگر CBT و PIM-SM، ایده استفاده از یک نقطه ملاقات^۱ را بطور تقریباً همزمان مطرح کردند. در این پروتکلها، گیرنده ها یک درخت اشتراکی^۲ می سازند که ریشه آن نقطه فوق الذکر می باشد و فرستنده ها داده خود را به سمت نقطه ملاقات می فرستند تا از آنجا در تمام درخت پخش شود. مشکلات اصلی این پروتکلها تمرکز ترافیک حول نقطه ملاقات و وجود single point of failure) همان نقطه ملاقات) می باشد.[6,10,11,12,14,15]

روش دیگری که در حل مسأله multicast استاتیک استفاده شده است، روش MST(Minimum Spanning Tree) می باشد. در multicast استاتیک وضعیت عضویت تمام اعضا و همچنین توپولوژی شبکه باید قبل از اجرای الگوریتم مشخص باشد و بدرد محیطهای دینامیک مثل اینترنت نمی خورد. درخت ساخته شده در این روش از لحاظ مجموع هزینه لینکهای درخت مینیمم می باشد. معیار هزینه می تواند پهنای باند مصرفی و یا تأخیر و یا هر پارامتر دیگر

¹ Rendezvous Point

² Shared tree

^۳ قسمتی از هزینه های این کار توسط مرکز تحقیقات مخابرات ایران تأمین شده است.

۲. انواع درختهای multicast

در multicast داده برای گروهی از کاربران ارسال می شود. و برای این منظور یک درخت multicast تشکیل شده و بسته به تمام اعضای این درخت ارسال می گردد. اساساً سه نوع از این درختها وجود دارد:

۱- درخت فرستنده: در این حالت به ازای هر فرستنده (S) و هر گروه multicast (G) یک درخت (S,G) در شبکه تشکیل می شود که ریشه این درخت، فرستنده می باشد. برای ساخت درخت از کوتاهترین مسیر بین فرستنده تا گیرنده ها (درخت SPT در مسیر مستقیم) یا از کوتاهترین مسیر از گیرنده ها تا فرستنده (درخت SPT معکوس) استفاده می شود. در حالت دوم چون مسیر حرکت داده multicast با جهت ساخته شدن درخت مخالف می باشد به درخت ساخته شده درخت SPT معکوس می گویند. درخت SPT از لحاظ تأخیر از فرستنده تا گیرنده ها، بهینه می باشد. قسمت a شکل شماره ۱، یک درخت SPT را نشان می دهد. در این شکل فرستنده با S، گیرنده ها با F_i ، بقیه نودها بدون اسم و نقطه ملاقات با RP نشان داده شده است. اعداد روی لینکها نشان دهنده هزینه لینکها می باشد. توجه کنید که در این شکل هزینه مسیرها در دو جهت مختلف مساوی فرض شده است. لذا درخت SPT در مسیر مستقیم و معکوس یکسان خواهند بود. [8]

۲- درخت اشتراکی: در این حالت به ازای هر گروه multicast و برای همه منابع یک درخت مشترک وجود دارد. ما درخت اشتراکی را با (G)* نشان می دهیم. که * بمعنای دلبخواه بودن فرستنده می باشد. نقطه ملاقات، RP، بعنوان ریشه درخت فعالیت می کند و گیرنده ها درخواست اشتراکشان را به این نقطه می فرستند تا عضو درخت اشتراکی گردند. فرستنده ها نیز بسته ها را به RP می فرستند و RP این داده ها را در درخت اشتراکی پخش می کند. مهمترین خاصیت این درخت صرفه جویی در میزان حافظه مصرفی در روترهای روی درخت می باشد. زیرا هر گروه به ازای تمام منابع از یک درخت استفاده می کند. در [16] ثابت شده است که تأخیر این درخت حداکثر دو برابر درخت SPT می باشد. انتخاب RP بهینه در این درختها کار مهمی می باشد و مقالاتی نیز در این زمینه چاپ شده است. قسمت b شکل شماره ۱، یک درخت اشتراکی را نشان می دهد. [6,10,11,12,14,15]

۳- درخت Steiner: این درخت دربر گیرنده تمام گیرنده ها و یک یا تمام فرستنده ها می باشد. یعنی هم می توانیم به ازای هر فرستنده یک درخت داشته باشیم و یا اینکه به ازای تمام فرستنده ها یک درخت داشته باشیم. به هر حال مهمترین خاصیت این درخت بهینه بودن مجموع معیار هزینه لینکهای درخت می باشد. کلمه مجموع در جمله قبل بدین معنا نیست که مجموع معیار هزینه لینکهای یک مسیر بهینه می باشد بلکه بدین معناست که مجموع معیار هزینه تمام لینکهای درخت مینیمم می باشد. تأخیر در این حالت معمولاً از تأخیر درخت SPT و تأخیر درخت اشتراکی بیشتر می باشد. مزیت اصلی این درخت صرفه جویی در مصرف منابع شبکه می باشد. ولی مشکل افزایش تأخیر و سختی ساخت آن باعث عدم

لینک باشد. به درخت درست شده در این حالت درخت Steiner میگویند. ساخت درخت Steiner جزء مسائل NP-Complete می باشد و تمام روشهای ساخت آن بصورت heuristic می باشد. [19] یک خلاصه مفید از روشهای مختلف ساخت درخت Steiner را ارائه می دهد.

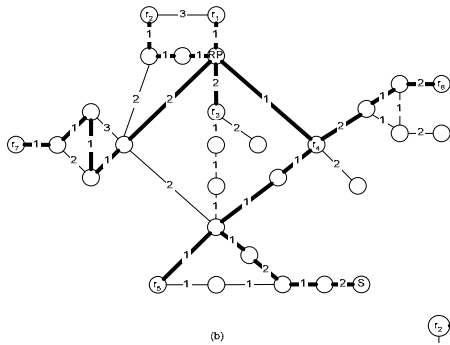
بدلیل مشکلات فوق و NP-Complete بودن ساخت درخت MST، ایده مینیمم کردن هزینه و ساخت درخت MST در پروتکل های multicast مورد استفاده قرار نگرفته است. ما در این مقاله پروتکل NRBP را عرضه می کنیم که می تواند درخت شبه Steiner را بصورت افزایشی بسازد ضمن این که درخت ساخته شده از لحاظ تأخیر یا طول مسیرها از فرستنده به گیرنده بهینه می باشد. در ضمن پروتکل ارائه شده می تواند درخت SPT را طوری بسازد که از لحاظ معیار هزینه از درختهای مشابه بهینه تر می باشد. همچنین می توان با تغییر پارامترهای پروتکل NRBP محدوده وسیعی از درختها را بین درخت SPT و درخت Steiner درست نمود. ما ایده خود را به حالتی که بخواهیم برای هر فرستنده یک درخت SPT درست کنیم اعمال کرده ایم ولی اعمال آن به درختهای اشتراکی نیز سرراست می باشد.

در این روش، تعدادی از روترهای روی درخت فعلی به عنوان کاندیدا انتخاب میشوند و به گیرنده جدید پیشنهاد میدهند. گیرنده بهترین پیشنهاد را انتخاب کرده و درخت را از آنجا درست میکند. پیغامهای پیشنهاد حاوی پارامترهای مسیر مثل مشخصات و وضعیت روترهای بین راه، پهنای باند و تأخیر لینکها و فاصله پیشنهاد دهنده از فرستنده داده می باشد. لذا گیرنده میتواند ضمن بهینه کردن هزینه درخت، بار فعلی شبکه را نیز در نظر بگیرد و از ازدحام شبکه جلوگیری کند. حتی پیغامهای پیشنهاد میتوانند حاوی اطلاعات QoS باشند تا گیرنده از وضعیت سرویسی که دریافت میکند مطمئن باشد. از دیگر مزایای پروتکل NRBP امکان صرفه جویی در مصرف حافظه روترها می باشد. بدین معنا که اگر روتری حافظه کافی برای اضافه کردن لینکهای جدید به درخت multicast نداشته باشد. می تواند از دادن پیشنهاد امتناع نماید

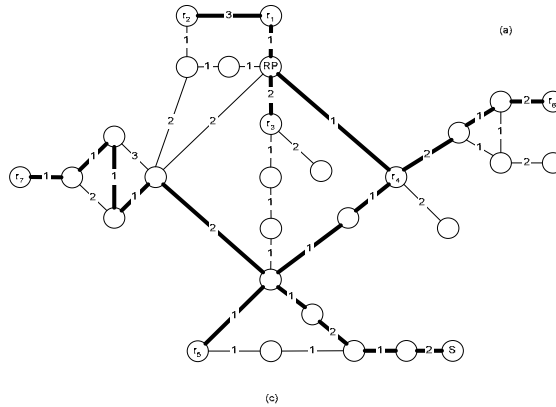
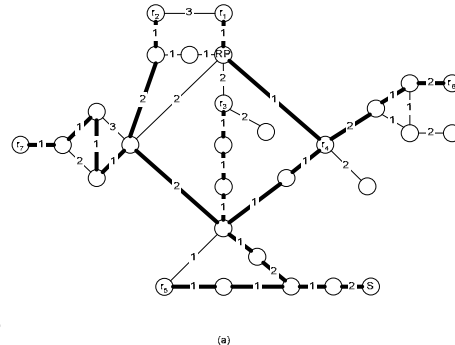
مزیت دیگر پروتکل NRBP ساخت درخت SPT در مسیر مستقیم می باشد. در واقع درخت SPT در اکثر پروتکل های موجود، بصورت معکوس درست میشود. یعنی بجای استفاده از کوتاهترین مسیر از فرستنده تا گیرنده ها (یا مسیر رفت)، از کوتاهترین مسیر از گیرنده ها تا فرستنده (یا مسیر برگشت) استفاده میکنند. این دو مسیر در شبکه های غیرمقارن¹ با هم تفاوت دارند. تحقیقات انجام شده در [18] نشان می دهد که 50% درصد از مسیرهای اینترنت نامقارن می باشد. در ادامه مقاله، در بخش دوم ما انواع درختهای multicast را معرفی خواهیم کرد و در بخش سوم به بحث و معرفی کارهای قبلی انجام شده خواهیم پرداخت. در بخش چهارم مشکلی را که تا کنون مورد توجه دقیق محققین قرار نگرفته است مطرح می کنیم و در بخش بعدی یک پروتکل برای حل آن طراحی میکنیم. بخش ششم نیز به نتیجه گیری و کارهای آینده اختصاص دارد.

¹ Asymmetric

نیز به ترتیب اضافه شده اند. [19]



استفاده از آن شده است. قسمت C شکل شماره ۱، درخت Steiner را نشان میدهد. در این شکل فرض شده است که ابتدا $r1$ به درخت اضافه شده است و بقیه



شکل ۱: سه نوع درخت مرسوم در multicast

این پیام در صورت برخورد به نودهای روی درخت متوقف می شود و آن نود به این پیام جواب می دهد. گیرنده پس از جمع آوری این پیغامها، الگوریتم پیدا کردن بهترین نود را جهت وصل شدن به درخت شروع میکند. گیرنده از بین جوابهای رسیده، بهترین نود از نوع trunk را پیدا می کند و درخت را از آنجا تشکیل میدهد. نودهای روی درخت بر حسب نوع وصل شدن، به دو نوع تقسیم می شوند. در صورتی که گیرنده از طریق کوتاهترین مسیر به فرستنده وصل شود نود گیرنده از نوع trunk می باشد. همچنین اگر گیرنده به یک نود trunk مثلاً $n1$ وصل شود دو حالت پیش می آید:

- ۱- اگر فاصله مسیر گذرنده از $n1$ حداکثر به اندازه C بیشتر از کوتاهترین مسیر باشد گیرنده از نوع trunk می باشد. در غیر این صورت:
- ۲- اگر فاصله مسیر گذرنده از $n1$ حداکثر به اندازه T بیشتر از کوتاهترین مسیر باشد گیرنده از نوع non-trunk می باشد.

همانطور که می بینیم گیرنده با broadcast کردن سربار زیادی روی شبکه میگذارد. همچنین برای اجرای الگوریتم لازم است که فاصله روترها از همدیگر مشخص باشد.

در [3] نیز ایده جستجوی محلی^۱ مطرح شده است که این کار نیز مورد استفاده اکثر پروتکلهای دیگر multicast که قصد پشتیبانی از QoS را داشته

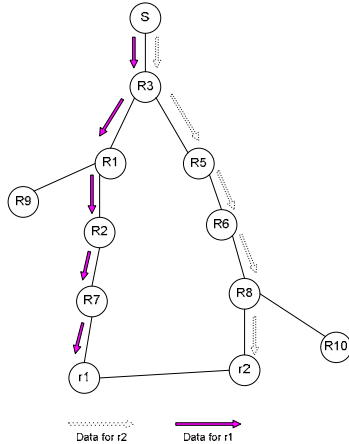
۳. کارهای انجام شده قبلی

آقای Fujinoki در [1] الگوریتمی ارائه داده است که درخت SPT را با کمترین هزینه و بصورت دینامیک درست میکند. در این روش هر نود که می خواهد به درخت وصل شود، ابتدا با استفاده از الگوریتم دیجسترا (Dijkstra) کوتاهترین مسیر بین خود و بقیه نودها را محاسبه می کند. سپس تمام مسیرهایی با طول برابر که دارای کمترین فاصله تا فرستنده هستند را با جستجو در میان همسایه های خود پیدا می کند و مسیری را که از نزدیکترین همسایه میگذرد انتخاب می کند. لذا هر نود یا روتر باید اطلاعات مربوط به توپولوژی شبکه مثل تمام کوتاهترین مسیرها و همچنین وضعیت عضویت تمام اعضا را بداند تا بتواند الگوریتم مربوطه را اجرا کند (مثل روشهای استاتیک). لذا الگوریتم مربوطه فقط در شبکه های که پروتکلهای link state (مثل OSPF) را اجرا می کنند قابل اجرا است. مشکل دیگر این است که در این روش بجای انتخاب مسیر با کمترین هزینه اضافی، نزدیکترین همسایه انتخاب میشود که الزاماً بهترین انتخاب نیست.

آقای Fujinoki در [4] روش دیگری ارائه کرده است که با وجود اینکه بهترین مسیر را انتخاب میکند، ولی همچنان به اطلاعات مربوط به توپولوژی شبکه نیاز دارد. در این روش نودی که میخواهد به درخت وصل شود پیغامی را با TTL محدود (برابر فاصله آن نود از فرستنده) در شبکه broadcast می کند.

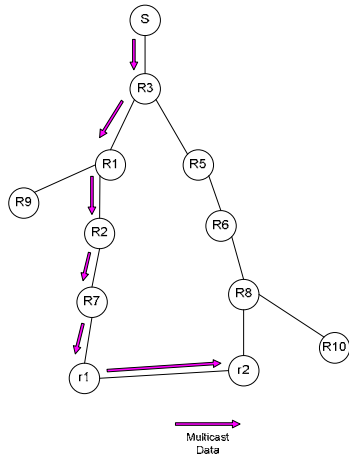
¹ Local Search

شکل (درخت SPT) حرکت نماید. پهنای باند مصرفی را برابر تابعی از مجموع تعداد دفعات عبور یک داده multicast مشخص از لینکهای شبکه در نظر می گیریم (که برابر تعداد لینکهای موجود در درخت ضربدر طول داده های multicast می باشد). در شکل شماره ۲ به ازای فرستادن هر داده multicast ۹ لینک از لینکهای شبکه مورد استفاده قرار می گیرد. لذا پهنای باند مصرفی برابر 9B می باشد که B برابر اندازه داده multicast می باشد. حال اگر ما به جای multicast از unicast استفاده کنیم باید یکبار داده را برای r1 و یک بار دیگر برای r2 بفرستیم و همانطور که در شکل شماره ۳ مشخص می باشد پهنای باند مصرفی برابر 10B خواهد بود.



شکل ۳: توزیع داده multicast بصورت unicast

می بینیم که استفاده از multicast تنها 10% وضعیت را بهبود داده است که با توجه به هزینه استفاده از multicast (مثل اجرای پروتکل SPT و تشکیل درخت و...) چندان مقرون به صرفه نیست. حال فرض کنید که از درخت MST استفاده کنیم در این صورت توزیع داده multicast به صورت شکل شماره ۴ خواهد بود.

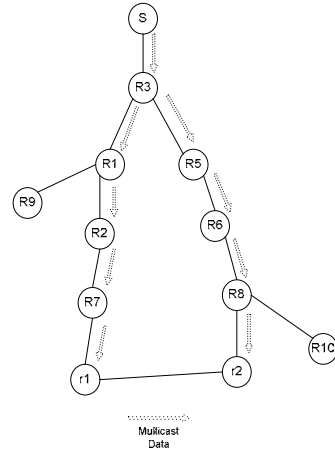


شکل ۴: توزیع داده multicast با استفاده از درخت MST

اند قرار گرفته است. در این روش هر گیرنده جدید یک پیام درخواست با TTL محدود را در شبکه broadcast میکند. این پیام به تعدادی از نودهای روی درخت برخورد می کند و همانجا متوقف می گردد. نودهای فوق الذکر به گیرنده جواب می دهند و گیرنده از بین جوابهای رسیده بهترین نود را جهت وصل شدن به درخت انتخاب می کند. ما به طور متفاوتی از این ایده برای پیدا کردن بهترین مسیر از لحاظ پهنای باند مصرفی و تأخیر استفاده می کنیم.

در روشی که اولین بار در [2] مطرح شد و بطور وسیعی در روشهای دیگر، مورد استفاده قرار گرفت، بدین صورت عمل می شود که پیام join هر کجا به درخت برخورد کرد درخت از آنجا تشکیل می گردد که ما نیز از این ایده استفاده کرده ایم. اما هیچکدام از کارهای فوق به حل مشکل مطرح شده در این مقاله نمی پردازند و ما تنها از ایده های موجود در آنها استفاده می کنیم.

در [7] خلاصه مفیدی از انواع روشهای متداول در multicast ارائه شده است. همچنین تاریخچه و مفاهیم اساسی multicast و نحوه پیاده سازی پروتکلهای multicast نیز توضیح داده شده است. مرجع [8] که یکی از مقالات اساسی در زمینه multicast است برای اولین بار multicast و اولین پروتکل آن یعنی DVMRP را توضیح می دهد. از دیگر مقالات اساسی multicast کار ارائه شده در [6] می باشد. این مقاله برای اولین بار ایده درختهای اشتراکی را در پروتکلی به اسم CBT معرفی کرد. پروتکلهای PIM-SM و PIM-DM نیز در [5] و [12] توضیح داده شده اند که از این میان پروتکل PIM-SM دارای مقبولیت گسترده ای می باشد و در اکثر روترها پشتیبانی می شود. دو مقاله [10, 11] نیز پروتکلهای مختلف multicast را از لحاظ پهنای باند مصرفی و تأخیر توزیع داده با هم مقایسه کرده اند.



شکل ۲: توزیع داده multicast در درخت SPT

۴. توضیح مسأله

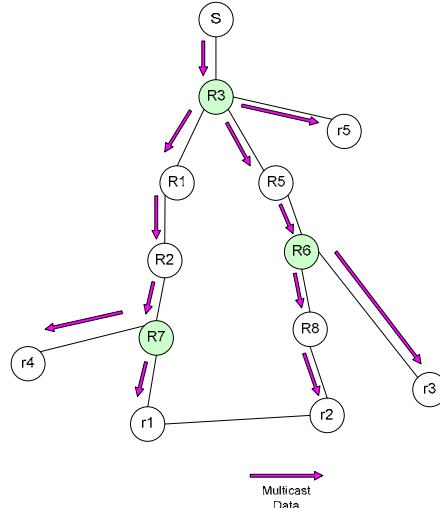
فرض کنید در شکل شماره ۲، از درخت SPT برای توزیع داده multicast فرستنده S به گیرندهای r1 و r2 استفاده کنیم و داده در مسیر نشان داده شده در

نکته مهم در شکل فوق و شکلهای قبلی این است که ما اگر n گیرنده داشته باشیم، باید جاهایی در شبکه، n کپی از داده را تولید کنیم. که این محلها می تواند فرستنده، روترهای میانی و یا حتی گیرنده ها (مثل $r1$ در شکل شماره ۴) باشند. مثلاً در شکل فوق $R3$ دو کپی از داده و $R6$ و $R7$ هر کدام یک کپی از داده را تولید می کنند. هر چه ما عمل تولید کپی را در نزدیکی فرستنده انجام دهیم پهنای باند بیشتری مصرف میکنیم و هر چه ما این کار را در نزدیکی گیرنده ها انجام دهیم در مصرف پهنای باند صرفه جویی خواهیم کرد.

در شکل شماره ۶ یک گیرنده جدید r_{new} می خواهد به گروه multicast فرستنده S بپیوندد. در این شکل درخت فعلی برای گیرنده های $r1$ و $r2$ با خطوط پر رنگ نشان داده شده اند. الگوریتم SPT مسیر شماره ۱ را انتخاب خواهد کرد و درخت را از طریق $R3$ تشکیل خواهد داد.

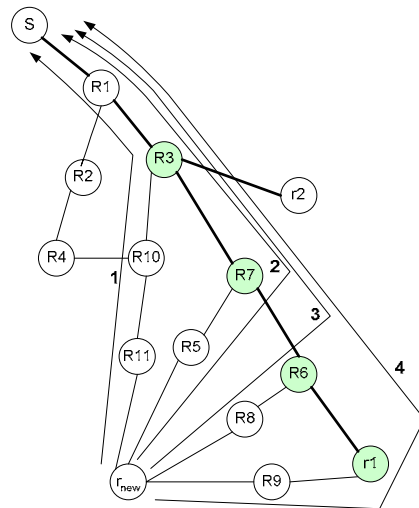
الگوریتم MST نیز درخت را از طریق $R7$ یا $R6$ یا $r1$ و با استفاده از مسیرهای شماره دو تا چهار تشکیل خواهد داد. اما همانطور که در شکل دیده می شود مسیر شماره ۲ هم از لحاظ تأخیر (معیار SPT) و هم از لحاظ پهنای باند مصرفی (معیار MST) نسبت به بقیه بهینه می باشد. اما هیچکدام از پروتکلها و الگوریتمهای موجود انتخاب این مسیر را تضمین نمی کنند. حال مسأله این است که چگونه عمل کپی داده و یا نقاط انشعاب را به گیرنده ها نزدیک کنیم بطوری که تأخیر افزایش پیدا نکند یا تغییر چندانی نکند. ما در بخش بعدی یک پروتکل برای این منظور طراحی می نمایم.

در این حالت پهنای باند مصرفی برابر $6B$ خواهد بود یعنی پهنای باند مصرفی 40% کاهش یافته است. توجه شود که ما در این حالت $5B$ را در هر صورت برای یک گیرنده نیاز داریم و با $1B$ اضافی به گیرنده دیگر نیز جواب داده ایم. در حالی که برای گیرنده $r2$ ، تأخیر فقط یک واحد اضافه شده است.



شکل ۵: توزیع داده multicast با استفاده از درخت SPT

حال فرض کنید که بخواهیم برای پنج گیرنده موجود در شکل شماره ۵، داده multicast مربوط به S را از طریق درخت SPT بفرستیم. ما نقاط انشعاب درخت را با رنگ تیره تر نشان داده ایم. این روترها، داده multicast را کپی می کنند و به واسطهای خروجی می فرستند.



شکل ۶: نحوه وصل شدن یک گیرنده جدید

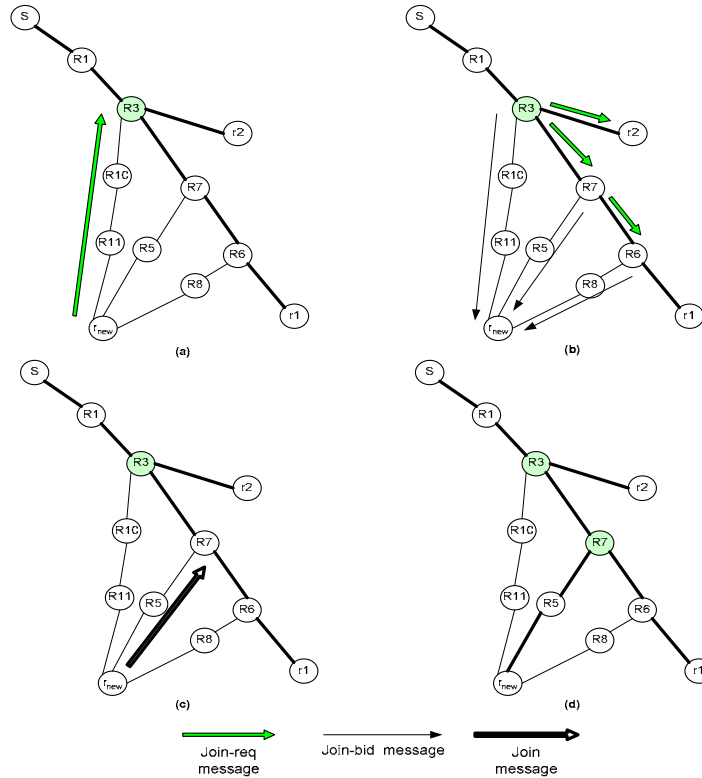
۵. پروتکل NRBP

فرض کنید که r_{new} می خواهد به گروه multicast فرستنده S بپیوندد. لذا پیغام $Join-req(r_{new}, S, G, C)$ را به سمت S خواهد فرستاد (شکل شماره ۷ قسمت a). پارامترهای این پیغام به ترتیب آدرس فرستنده پیغام (r_{new}), آدرس گیرنده پیغام (S), آدرس گروه (G) و یک شمارنده (C) می باشد که r_{new} مقدار C را برابر صفر قرار می دهد. فرض کنید که این پیغام از مسیر شماره ۱ (شکل ۶) به سمت فرستنده می رود. لذا این پیغام در نود $R3$ به درخت برخورد خواهد کرد و در این نقطه متوقف می شود. (مثل روش [2])

حال روتر $R3$ پیغام $Join-bid(R3, r_{new}, S, G, d, C, P^*)$ را به سمت r_{new} می فرستد که پارامترهای این پیغام به ترتیب آدرس فرستنده پیغام ($R3$), آدرس گیرنده پیغام (r_{new}), آدرس فرستنده گروه (S), فاصله فرستنده پیغام از S یعنی d و یک پارامتر متغیر به اسم P^* و یک شمارنده (C) می باشد. مقدار پارامتر C موجود در این پیغام کپی مقدار پارامتر C موجود در پیغام $Join-req$ رسیده میباشد. تمام روترهای موجود در مسیر آدرس خود را به پیغام اضافه می کنند لذا گیرنده می تواند از روی تعداد آدرسهای موجود در پیغام، فاصله خود را تا پیشنهاد دهنده پیدا کند و از جمع این مقدار با مقدار d دریافتی میتواند فاصله خودش تا S را محاسبه کند. گیرنده از آدرسهای موجود در P^* جهت هدایت

پیغام Join به سمت پیشنهاد دهنده استفاده می کند تا درخت در مسیر مستقیم

درست شود.



شکل ۷: مراحل ساخته شدن درخت در پروتکل NRBP

کند و از جمع این مقدار با مقدار d دریافتی می‌تواند فاصله خودش تا S را محاسبه کند.

پارامتر دیگری به اسم K در تمام گیرنده ها وجود که اگر فرض کنیم در پیشنهادات رسیده، کمترین فاصله تا S برابر d_{min} باشد، گیرنده تمام پیشنهادات با فاصله کمتر یا مساوی $K + d_{min}$ تا S را پیدا میکند و پیغام با کمترین فاصله نسبت به پیشنهاد دهنده را از بین آنها انتخاب میکند و پیغام $Join(r_{new}, P, S, d_s)$ را به سمت آن می فرستد.

که اگر مقدار K صفر باشد بدین معناست که هیچگونه آزادی در انتخاب مسیر وجود ندارد و باید کوتاهترین مسیر تا S انتخاب گردد. لذا گیرنده تمام پیشنهادات با کمترین فاصله مساوی تا S را پیدا میکند و پیغام با کمترین فاصله نسبت به پیشنهاد دهنده را از بین آنها انتخاب میکند و پیغام $Join(r_{new}, P, S, G, d_s)$ را به سمت آن می فرستد (شکل شماره ۷ قسمت c). پارامتر d_s در این پیغام نشان دهنده فاصله r_{new} تا فرستنده میباشد. روترهای بین راه یک واحد از این مقدار کم میکنند و فاصله خود تا فرستنده را برابر این مقدار جدید میگذارند. همچنین بعد از اضافه کردن حالت مربوط به گروه در جدول multicast خود، این پیغام را به روتر بعدی می فرستند. (شکل شماره قسمت d). بدین ترتیب درخت SPT با هزینه بهینه درست میشود.

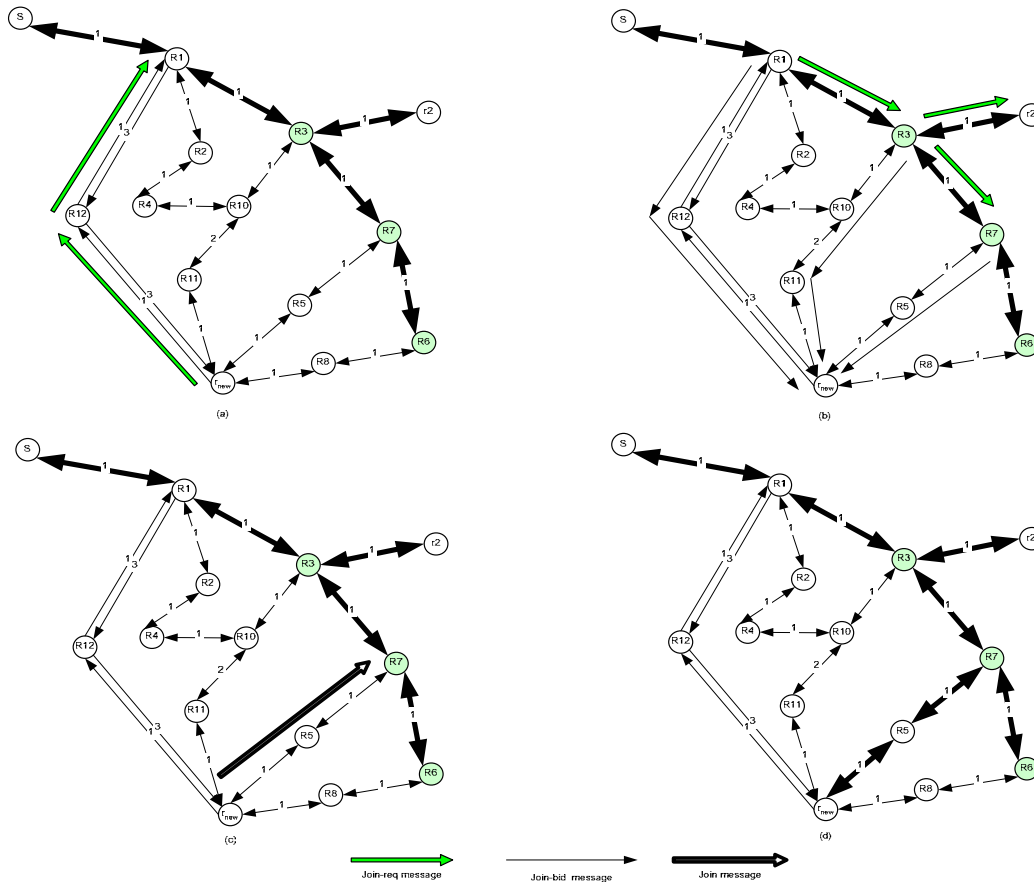
روتر $R3$ نیز همچنین مقدار C را یک واحد افزایش میدهد و پیغام $Join-req(r_{new}, S, G, C)$ را در جهت پخش می کند (شکل شماره ۷ قسمت b). دقت کنید که $R3$ این پیغام را به تمام شاخه های درخت به جز شاخه ای که به سمت S می باشد می فرستد زیرا در این مسیر هیچ روتری وجود ندارد که شرایطش از $R3$ بهتر باشد ضمن اینکه ما از تشکیل نقاط انشعاب در نزدیکی فرستنده جلوگیری می نمایم. در روش ما بر خلاف تمام روشهای فعلی به جای اینکه پیغام join متوقف شود یا به سمت بالاتر فرستاده شود به سمت پایین فرستاده می شود.

هر روتری که روی درخت باشد در صورت دریافت پیغام $Join-req(r_{new}, S, G, C)$ ابتدا C را چک میکند در صورتی که مقدار آن از حد مجاز (C_{max}) نگذشته باشد آنرا یک واحد افزایش می دهد و به صورتی که در بالا توضیح دادیم در درخت پخش می کند و همچنین یک پیغام $join-bid$ مناسب به سمت r_{new} می فرستد. این مشابه همان جستجوی محلی می باشد که در بخش سوم ذکر کردیم.

گیرنده جدید (r_{new}) با دریافت پیغام $Join-bid(R3, r_{new}, S, G, d, C, P^*)$ که C آن برابر C_{max} باشد می تواند بفهمد که تمام پیشنهادات را دریافت کرده است و از روی P^* دریافتی می تواند فاصله خود را تا پیشنهاد دهنده پیدا

گیرنده باید یک زمان اضافی صرف کند چون ممکن است چند پیغام با C برابر دو برسد. مقدار پارامتر C_{max} در تأخیر عضو شدن گیرنده ها تأثیر مستقیمی دارد و هر چه مقدار آن بیشتر باشد زمان بیشتری طول میکشد تا درخت مربوط به گیرنده جدید درست شود ولی درخت ساخته شده دارای کیفیت بهتری میباشد.

حال ما نحوه ساخت درخت را در شبکه های نامتقارن نشان میدهیم. در شکل شماره ۸ مثالی از یک شبکه نامتقارن نشان داده شده است. پیکانهای دوطرفه نشان دهنده لینکهای متقارن می باشند و برای لینکهای نامتقارن دو مسیر رفت و برگشت بصورت متمایز نشان داده شده اند. اعداد روی لینکها نیز نشان دهنده هزینه لینک در جهت مربوطه می باشد. همانطور که میبینید روش ما بهترین مسیر را از فرستنده به گیرنده انتخاب می کند.



شکل ۸: نحوه ساخت درخت در شبکه های نامتقارن

در این پیغام یک می کند. گیرنده از روی بیت Immediate-join پیغام دریافتی می فهمد که اولین گیرنده میباشد و لذا پیغام Join را به سمت فرستنده میفرستد و در مسیر خود به سمت فرستنده درخت را درست میکند. روترهای بین راه از روی پارامتر d_s موجود در پیغام فاصله خود تا فرستنده را تشخیص می دهند.

اگر مقدار پارامتر K برابر بینهایت باشد، مقدار $K+d_{min}$ برابر بینهایت خواهد بود. لذا گیرنده تمام پیشنهادات را بررسی میکند و پیغام با کمترین فاصله نسبت به خود را پیدا میکند و پیغام $Join(r_{new}, P, S, G, d_s)$ را به سمت آن می فرستد. یعنی درخت Steiner را با تأخیر بهینه می سازد.

اگر مقدار پارامتر K بزرگتر از صفر باشد یک درخت بهینه که از لحاظ رفتار، بین SPT و Steiner می باشد درست میشود.

می توان از پارامتر C_{max} برای کنترل حوزه جستجوی محلی استفاده کرد. مثلاً اگر مقدار آن برابر دو باشد، پیغام join-req بعد از برخورد به درخت حداکثر تا عمق دو در درخت پیش می رود و گیرنده نیز منتظر دریافت پیغامی با C برابر دو می ماند تا نسبت به پیشنهادات رسیده تصمیم بگیرد. دقت شود که

۶. جزئیات طراحی پروتکل NRBP

فرض کنید که اولین گیرنده می خواهد به گروه وصل شود چون هیچ درختی برای گروه وجود ندارد پیغام Join-req تا فرستنده می رود و فرستنده نیز پیغام Join-bid را به سمت آن میفرستد. فرستنده بیتی به نام Immediate-join را

امکان دارد که پیغام Join-bid مربوط به روتر با C_{max} در شبکه گم شود و یا روتر مربوطه مایل به ارائه پیشنهاد نباشد. در این صورت باید راهی وجود داشته باشد که از انتظار بیهوده گیرنده جلوگیری کنیم. راه حل پیشنهادی بدین صورت است که گیرنده بعد از ارسال پیغام Join-req تا رسیدن اولین پیشنهاد صبر می کند و سپس یک timer با مقدار $T_p \cdot C_{max} + \frac{T}{2}$ را روشن میکند و پس از سر آمدن آن نسبت به بررسی پیشنهادات رسیده اقدام می کند. مقدار T برابر مدت زمان طول کشیده تا رسیدن اولین پیشنهاد می باشد و T_p نیز زمان لازم برای پردازش پیغام Join-req در یک روتر می باشد.

پیغام Join-bid ممکن است در مسیر خود به سمت گیرنده به یک نود دیگر روی درخت برخورد کند در این صورت، آن روتر پیغام دریافتی را در نظر نمی گیرد و یک پیغام Join-bid جدید تولید می کند و به سمت گیرنده میفرستد. این کار باعث جلوگیری از تشکیل حلقه در درخت می شود. وقتی گیرنده ای بخواهد از گروه multicast خارج شود پیغام Leave(r, P*, S, G) را به سمت همان نقطه ای که پیغام Join را فرستاده بود میفرستد. لذا هر گیرنده باید اطلاعات موجود در پیغام Join-bid پذیرفته شده را تا هنگام خارج شدن از گروه نگهداری کند. نکته دیگر این است که هر گیرنده باید در فواصل زمانی مشخص (مثلاً هر ۳۰ دقیقه یکبار) دوباره پیغام Join بفرستد تا اطلاعات موجود در جداول multicast روترهای بین راه را تازه کند.

Node A is joining group G with sender S

1. **Send** Join-req message toward S;
2. **Set** timer T_1 ;
3. **Wait until** first join-bid message received or T_1 expires;
4. **Set** timer $T_2 \leftarrow (T_p \cdot C_{max} + T_1/2)$;
5. **Let** $D_1 \leftarrow \{\}$, $D_{min} \leftarrow +\infty$;
6. **For** each of received Join-bid messages before T_2 expires
 - a. **Let** $D_b \leftarrow D_{br} + D_{bs}$;
 - b. $D_1 \leftarrow D_1 \cup \{(D_b, D_{br}, P)\}$;
 - c. $D_{min} \leftarrow \min(D_{min}, D_b)$;
7. **For** each $(D_i, D_j, P) \in D_1$
 - a. **If** $D_i \leq D_{min} + K$ then
 - i. $D_{opt} \leftarrow \min(D_{opt}, D_i)$;
 - ii. $P_{opt} \leftarrow P$;
8. **Send** Join message Using P_{opt} ;

(a)

Node A received a Join-bid message for group G with sender S

1. **if** $A \in \text{tree}(S,G)$ then
 - a. **Discard** Join-bid message
 - b. **Send** a new Join-bid message from A toward receiver of Join-bid message;
2. **else**
 - a. **Update** P^* value of the Join-bid message;
 - b. **Forward** updated Join-bid message;

(b)

Node A received a Join-req message for group G with sender S

2. **if** $A \in \text{tree}(S,G)$ then
 - a. **if** resource exists **then**
 - i. **Send** Join-bid message toward sender of Join-req message;
 - b. **if** $(C < C_{max})$ **then**
 - i. **Increase** C;
 - ii. **Send** Join-req to All_Multicast_Links(S, G) - iff_{Join-req} - iff_S ;
3. **else**
 - a. **Forward** Join-req message toward it's next-hop ;

(c)

Node A received a Join message for group G with sender S

1. **if** $A \in \text{tree}(S,G)$ then
 - a. **Discard** Join message
 - b. **Update** the multicast routing table;
2. **else**
 - a. **Decrease** D_S value;
 - b. **Create** multicast routing table for (S,G);
 - c. **Forward** Join message toward it's next-hop based on P;

(d)

شکل ۹: pseudo code های مربوط به پروتکل NRBP

همانطور که گفتیم می توان با انتخاب مقدار K بزرگتر از ۱ محدوده وسیعی از درختها را ساخت. با افزایش مقدار K هزینه ساخت درخت یا پهنای باند مصرفی درخت کاهش می یابد. اما این باعث می شود که امکان انتخاب مسیرهای طولانی تر جهت ساخت درخت بالاتر رود. همچنین مقدار بینهایت را که برای ساخت درخت Steiner در بخش قبل از آن استفاده کردیم را در عمل

شکل شماره ۹ pseudo code های مربوط به پروتکل NRBP را نشان می دهد. در قسمتهای a, b, c, d این شکل به ترتیب pseudo code مربوط به وصل شدن یک گیرنده جدید به درخت، دریافت پیغام Join در نود A، دریافت پیغام Join-bid در نود A و دریافت پیغام Join-req در نود A نشان داده شده است.

منابع

- [1] H. Fujinoki and K. Christensen, "The New Shortest Best Path Tree (SBPT) Algorithm for Dynamic Multicast Trees," *Proceedings of the IEEE 24th Conference on Local Computer Networks*, pp. 204-211, October 1999, Boston, Massachusetts
- [2] M. Doar, I. Leslie, "How bad is naïve multicast routing", In *Proceeding of the IEEE Infocom 1993*.
- [3] K. Carlberg, J. Crowcroft, "Building shared trees using a one-to-many joining mechanism", *ACM SIGCOMM Computer Communication Review*, v.27 n.1, p.5-11, Jan. 1997
- [4] H. Fujinoki and K. Christensen, "A Routing Algorithm for Dynamic Multicast Trees with End-to-End Path Length Control," *Computer Communications*, Vol. 23, No. 2, pp. 101-114, January 2000.
- [5] S. Deering, D. Estrin, D. Faranacci, V. Jacobson, C. G. Liu, L. Wei, "The PIM Architecture for Wide-Area Multicast Routing", *IEEE/ACM Transactions on Networking*, Vol.4, No.2, April 1996
- [6] T. Ballardie, P. Francis, J. Crowcroft, "Core Based Trees (CBT): An Architecture for Scalable Inter-Domain Multicast Routing", *ACM SIGCOMM'93*
- [7] K. C. Almeroth, "The Evolution of Multicast: From the Mbone to Inter-Domain Multicast to Internet2 Deployment", *IEEE Network*, Jan./Feb. 2000
- [8] S. E. Deering, D. R. Cheriton, "Multicast Routing in Datagram Internetworks and Extended LANs", *ACM Transactions on Computer Systems*, Vol.8, No.2, May.1990
- [9] T. Billhartz, J. B. Cain, E. Farrey-Goudreau, D. Fieg, S. Batsell, "Performance and Resource Cost Comparisons for the CBT and PIM Multicast Routing Protocols", *IEEE Journal on Selected Areas in Communications*, April 1997
- [10] L. Wei, D. Estrin, "Multicast Routing in Dense and Sparse Modes: Simulation study of Tradeoffs and Dynamics", University of Southern California, Computer Science, Technical Report, 1995
- [11] L. Wei, D. Estrin, "The Trade-offs of Multicast Trees and Algorithms", in *Proc. International Conference on Computer Communications and Networks*, Sep. 1994
- [12] S. Deering, D. Estrin, D. Faranacci, V. Jacobson, C. G. Liu, L. Wei, "An Architecture for Wide-Area Multicast Routing", *Proc. SIGCOMM'94, Computer Communication Review*, Vol.24, No.4, Oct.1994
- [13] D. Waitzman, C. Partridge, S. Deering, "Distance Vector Multicast Routing Protocol", RFC 1075, Nov.1988
- [14] A. Ballardie, "Core Based Trees (CBT Version 2) Multicast Routing - Protocol Specification", RFC 2189, Sep.1997
- [15] A. Ballardie, "Core Based Trees (CBT) Multicast Routing Architecture", RFC 2201, Sep.1997
- [16] Y. K. Dalal, R. M. Metcalfe, "Reverse path forwarding of broadcast Packets" *Communication of the ACM*, 21(12), page 1040-1048, 1978.
- [17] UCB/LBNL/VINT, "Network simulator, ns (version 2)" <http://www.mash.cs.berkeley.edu/ns>, 2002.
- [18] V. Paxson, "End-to-End Routing Behavior in the Internet", In *Proceedings of SIGCOMM '96 (Stanford, CA, August 1996)*.
- [19] P. Winter, "Steiner problem in networks: a survey", *Networks*, v.17 n.2, p.129-167, Summer 1987

برابر قطر شبکه قرار می دهیم. در این صورت تمام نودهای واجد شرایط روی درخت به گیرنده جدید پیشنهاد عضویت می دهند.

وقتی روتری پیغام Join-req را دریافت کرد، در صورت کمبود حافظه یا حالت میتواند به پیغام فوق الذکر جواب ندهد و فقط آنرا روی درخت پخش کند. روترهای دیگر با پیشنهادات خود درخت را از نقطه دیگری درست خواهند کرد. لذا هم میتوان در مصرف حافظه صرفه جویی کرد و هم این امکان به روترهای پرازدحام داده میشود که در ساخت درختهای بیشتری مشارکت کنند. پیغامهای Join-bid می توانند در طی طریق به سمت گیرنده مشخصات QoS مسیر را جمع آوری کنند. این مشخصات می تواند شامل وضعیت ازدحام روترهای بین راه، توان پردازشی موجود در آنها، وضعیت بافرها و صفهای آنها و تعداد جریانهای عبوری از آنها باشد. همچنین می توانند وضعیت لینکهای مسیر مثل تأخیر آنها و پهنای باند باقیمانده آنها باشد. گیرنده می تواند با استفاده از این اطلاعات بهترین مسیر را انتخاب کند و ضمن مشارکت فعال در پروسه QoS، از مسیرهای پرازدحام اجتناب نماید.

۷. نتیجه گیری

در این مقاله ساخت درخت SPT مورد بررسی قرار گرفت و یک پروتکل جدید به نام NRBP طراحی گردید. درخت SPT ساخته شده توسط NRBP در عین حفظ خاصیت استفاده از کوتاهترین مسیره، پهنای باند کمتری مصرف می نماید. همچنین درخت SPT ساخته شده با استفاده از شیوه پیشنهادی، واقعی می باشد. امکان ساخت درخت Steiner بهینه نیز وجود دارد و می توان با تغییر پارامتر K رده وسیعی از درختها را ایجاد نمود و یک trade-off بین بهینه کردن تأخیر و هزینه کلی درخت ایجاد نمود. می توان با تغییر پارامتر C حوزه جستجوی محلی را کنترل نمود که تأثیر مستقیم بر تعداد پیغامهای رد و بدل شده در پروتکل دارد. امکان لحاظ کردن QoS و بالانس کردن ترافیک شبکه توسط گیرنده ها نیز از دیگر مزایای پروتکل پیشنهادی میباشد. قدم بعدی ما استفاده از شبیه ساز شبکه NS [17]، جهت شبیه سازی و ارزیابی پروتکل NRBP می باشد. با تولید کننده گراف شبکه موجود در NS گرافهای متعددی تولید کرده ایم و می خواهیم در مراحل بعدی پیاده سازی پروتکل، با استفاده از این گرافها درختهای تولید شده را از لحاظ پهنای باند مصرفی و تأخیر با درختهای SPT و Steiner مقایسه کنیم. همچنین تأثیر تغییر پارامترهای K و C_{min} را روی درختهای تولید شده مطالعه خواهیم کرد. پیاده سازی و ارزیابی پروتکل پیشنهادی در یک شبکه واقعی قدم نهایی ما در ارزیابی پروتکل NRBP می باشد.