نشریه علمی- ترویجی محاسبات نرم
شمارهٔ ششم/ پاییز و زمستان ۹۳/ صفحهٔ ۵۴-۶۷

ارسال:۹۳/۱۰/۱۳

پذیرش:۹٤/۵/۱٦

درخت تجمیع داده براساس الگوریتم پویای شکل‌گیری رودخانه درشبکه حسگر بی سیم

سعید مهرجو*، فرشاد خون جوش[۲]، سامان دهقانیان[۳]

* مربی، دانشگاه آزاد اسلامی، واحد داریون، گروه مهندسی کامپیوتر، داریون، ایران
mehrjoo@iaudariun.ac.ir
[۲] استادیار، دانشگاه شیراز، دانشکده مهندسی برق و کامپیوتر، شیراز، ایران
khunjush@cse.shirazu.ac.ir
[۳] دانشگاه آزاد اسلامی، واحد داریون، گروه مهندسی کامپیوتر، داریون، ایران
saman_dehghanian@yahoo.com

چکیده: از اساسی‌ترین چالش‌های موجود در شبکه حسگر بی‌سیم مسئله انرژی محدود گره‌ها می‌باشد. انتقالات بین گره‌ها بیشترین مصرفی انرژی درون گره‌ها را شامل می‌شود لذا به حداقل رساندن انتقالات غیر ضروری باعث به حداقل رساندن مصرف انرژی خواهد شد. افزونگی در داده‌های خام می‌تواند در نقاط تجمیع حذف گردد و تنها اطلاعات کاربردی بازیابی و به ایستگاه اصلی ارسال شوند. به منظور حل این موارد و بهینه‌سازی مصرف انرژی در ارتباطات، جمع‌آوری اطلاعات به عنوان یک تکنیک مؤثر درنظر گرفته می‌شود. در این مقاله گره‌های حسگر در یک درخت سازماندهی می‌شوند و جمع‌آوری داده‌ها در گره‌های میانی و در محل اتصال شاخه های درخت انجام می گیرد. بسته های داده جمع‌آوری شده بعداً به گره ریشه که همان ایستگاه اصلی است، فرستاده می‌شوند. یکی از خصوصیات اصلی پروتکل‌های درختی، بهینه‌سازی ساختار درخت تجمیعی داده از نظر مصرف انرژی است که ما این کار را با استفاده از یک الگوریتم هوش جمعی به نام الگوریتم شکل‌گیری پویای رودخانه انجام خواهیم داد. نتایج شبیه‌سازی‌ها حاکی از آن است که الگوریتم پیشنهادی ما در مقایسه با الگوریتم مورچگان نتایج بهتری را از نظر طول عمر شبکه ارائه خواهد داد.

واژه‌های کلیدی: شبکه حسگر بی‌سیم، طول عمر، درخت تجمیع داده، الگوریتم شکل‌گیری رودخانه.

۱. مقدمه

شبکه حسگر بی‌سیم نوع خاصی ازشبکه‌های موردی است و شامل مجموعه‌ای ازگره‌های کوچک می‌باشد که توانایی حس محیط اطراف باهدف معین، پردازش اطلاعات، ذخیره‌سازی، تبادل اطاعات با سایرگره‌ها را دارد. معمولاً تمامی گره‌ها همسان می‌باشند و عملاً با همکاری یکدیگر، هدف کلی شبکه را برآورده می‌سازند. هدف اصلی در شبکه‌های حسگر بی‌سیم نظارت و کنترل شرایط و تغییرات جوی، فیزیکی و یا شیمیائی در محیطی با محدوده معین، می‌باشد[۱, ۲].

گره‌های حسگر معمولاً پس از به کار گرفته شدن درمحیط موردنظر، بدون مراقبت رها می‌شوند. منابع محدود انرژی مانند باتری که به منظور تأمین انرژی این گره‌ها استفاده می‌گردد، به سختی جایگذاری یا شارژ می‌شوند. ضمناً نیاز است تا طول عمر مفید چنین شبکه‌هایی به منظور حفظ عملیات آنها، طولانی گردد. بنابراین استراتژی‌های مسیریابی داده‌ها براساس مصرف بهینه انرژی برای حفظ انرژی این شبکه‌ها نیاز می‌باشد.

زمانی که گره‌های حسگر به صورت متراکم در محیط قرار می‌گیرند، اطلاعات جمع‌آوری شده توسط گره‌های مجاور ممکن است شبیه به هم و بنابراین زائد باشند. اتلاف انرژی و ترافیک بالا هزینه‌هایی است که باید برای انتقال داده‌های تکراری از گره‌های منبع[۱] به ایستگاه پایه[۲] پرداخت. به منظور حل این موارد و بهینه‌سازی مصرف انرژی در ارتباطات، جمع‌آوری اطلاعات به عنوان یک تکنیک مؤثر درنظر گرفته می‌شود. افزونگی در داده‌های خام می‌تواند در نقاط تجمیع حذف گردد و تنها اطلاعات کاربردی و مفید بازیابی و به ایستگاه پایه ارسال شوند. بنابراین تعداد بسته‌های داده ای که در ایستگاه پایه دریافت می‌شود، کاهش پیدا خواهد کرد و درنتیجه در مصرف انرژی و پهنای باند صرفه جویی می‌شود.

پروتکل‌های معمول مسیریابی که از تکنیک‌های جمع‌آوری داده استفاده می‌کنند، بر اساس معماری شبکه به دو دسته تقسیم می‌شوند: پروتکل‌های خوشه بندی[۳] و پروتکل‌های درختی[۴]. پروتکل‌های خوشه بندی مانند LEACH ، HEED و FCM و غیره، گره‌های حسگر را در خوشه هایی دسته بندی می‌کنند. این پروتکل‌ها تکنیک‌های جمع‌آوری داده را در خوشه مرحله اول (سرخوشه) به کار می برند تا داده‌ها را از گره‌های عضو جمع‌آوری کرده و سپس داده فشرده شده را به ایستگاه پایه بفرستند. در این پروتکل‌ها فرض بر آن است که هر گره می‌تواند در یک گام بسته‌هایی را به ایستگاه پایه ارسال کند. این فرض اندازه شبکه را کاهش می دهد چون میزان ارتباطات گره‌های حسگر محدود می‌شود.

در این تحقیق روی دسته دوم معماری شبکه، یعنی پروتکل درختی تمرکز می‌شود. این پروتکل‌ها گره‌های حسگر را در یک درخت سازماندهی می‌کنند و تجمیع داده‌ها در گره-های میانی و در محل اتصال شاخه‌های درخت انجام می گیرد. بسته‌های داده جمع‌آوری شده بعداً به گره ریشه که همان ایستگاه پایه است، فرستاده می‌شوند. پروتکل‌های درختی برای کاربردهایی مناسب هستند که با تجمیع داده‌های درون شبکه ای سروکار داریم. تشخیص آتش سوزی در جنگل، نظارت بر ایمنی در طرح‌های صنعتی و مواردی که بیشترین اطلاعات مفید درباره وضعیت های ایمن اندازه گیری می‌شود را می‌توان به عنوان مثال‌هایی از این نوع کاربرد نام برد. یکی از خصوصیات اصلی پروتکل‌های درختی، بهینه‌سازی ساختار درخت تجمیعی داده از نظر مصرف انرژی است.

از آنجایی که گره‌های حسگری که از ایستگاه پایه دورتر هستند انرژی بیشتری را نسبت به آنهایی که به ایستگاه پایه نزدیکترند مصرف خواهند کرد، انتقال مستقیم از هیچ گرهی به یک ایستگاه پایه راه دور انجام نمی‌شود. در طرح مسیریابی چند گامی[۵] اکثر گره‌ها مشغول انتقال تعداد زیادی پیام داده (از طرف گره‌های دیگر) به ایستگاه پایه هستند، لذا به سرعت

1. Source nodes
2. Base Station
3. Cluster based
4. Tree based
5. Multi-hop

منابع انرژی آنها تخلیه خواهد شد. بنابراین یک راه حل، استفاده از ارتباطات چند گامی و همزمان ترکیب و تجمیع داده‌های همبسته درون شبکه می‌باشد[۱].

در این تحقیق ما برای اولین بار با بهره بردن از الگوریتم شکل‌گیری رودخانه درخت تجمیع داده‌ی بهینه‌ای را در شبکه حسگر بی‌سیم خواهیم ساخت که بکارگیری آن منجر به افزایش طول عمر شبکه حسگر خواهد شد. مکانیسم پیشنهادی گره‌های منبع را به عنوان قطره‌های مصنوعی درنظر گرفته تا مسیرهایی انرژی کارآمد بین گره‌های منبع و ایستگاه پایه ایجاد کند. مسیرهایی از گره‌های منبع مختلف به یک ایستگاه پایه، با یک درخت تجمیعی نمایش داده می‌شوند که ایستگاه پایه در ریشه قرار دارد.

مسئله درخت تجمیعی بهینه از سری مسائل NP-hard می‌باشد [۳] که معادل درخت Steiner [٤] و مسئله پوشش مجموعه وزنها می‌باشد. بسیاری از محققین در زمینه تجمیع داده‌ها در شبکه‌های حسگر بی سیم تحقیق کرده اند[۱, ۲].

الگوریتم حریصانه[۳] یک درخت چند هزینه[1] را با اضافه کردن گره‌ها به صورت تکراری ایجاد می کندکه دراین درخت ابتدا فقط ایستگاه پایه وجود دارد. در هر مرحله الگوریتم از بین گره‌های منبع باقیمانده گره منبعی را که به درخت فعلی نزدیکتر است، انتخاب می کند. این الگوریتم کوتاهترین مسیر بین آن گره منبع و درخت فعلی را نیز به درخت اضافه می کند. این عملیات ادامه پیدا می کند تا همه گره‌های منبع در درخت درج شوند.

در[٤] نویسندگان انرژی کارآمدی روش تجمیع حریصانه را بررسی کردند که متفاوت با روشهای انتشار قبلی برای تجمیع فرصت طلبانه روی درخت با کمترین تاخیر می‌باشد. روش تجمیع حریصانه، درخت افزایشی حریصانه (GIT) را به این ترتیب ایجاد می‌کند: کوتاهترین مسیر به ایستگاه پایه تنها برای اولین گره منبع در نظر گرفته می‌شود؛ درحالی که بقیه گره‌های منبع به صورت افزایشی به نزدیکترین گره در درخت فعلی متصل می‌شوند. شبیه‌سازی نشان داد که جمع‌آوری حریصانه بدون هیچ تاثیر منفی در معیار تأخیر یا قدرت در مقایسه با جمع‌آوری فرصت طلبانه، به طور قابل ملاحظه ای در مصرف انرژی کارآمدتر می‌باشد.

در [٦] الگوریتم کلونی مورچگان (ACO) را برای مسئله جمع‌آوری داده‌ها به کار بردند و آنرا الگوریتم تجمیع-مورچه[2] نامیدند. در یک شبکه حسگر بی‌سیم با منابع چندگانه و یک مقصد مشخص، مورچگان مصنوعی همان گره‌های منبع هستند که مسیرهایی را برای انتقال بسته های داده به ایستگاه پایه ایجاد می‌کنند. مسیرکوتاهتر بین یک گره منبع و یک گره تجمیعی آن مسیری است که فرومون مورچه بیشتری روی آن باقیمانده است. این الگوریتم از فرمول فاصله اقلیدسی برای محاسبه فاصله یک گره منبع از گره تجمیعی و همچنین محاسبه فاصله یک گره تجمیعی تا ایستگاه پایه، استفاده می کند. این درحالی است که فاصله اقلیدسی محاسبه شده ممکن است در شبکه حسگر بی سیم کاربردی نباشد چرا که میزان ارتباطات یک گره محدود می‌باشد. از آنجایی که گره‌ها فقط می‌توانند با همسایه یک گام[3] خود ارتباط داشته باشند، فاصله اقلیدسی بین یک گره منبع و ایستگاه پایه قابل اطمینان نمی‌باشد. درحقیقت منابع باید به طور غیر مستقیم و از طریق یک گره واسط با ایستگاه پایه ارتباط داشته باشند .

لیاو[4] و همکارانش [۵] با استفاده از ACO و با گسترش ناحیه جستجو به اطراف مسیرهای مشخص شده، روش بهبودیافته ای را برای دستیابی به درخت تجمیعی داده بهینه مطرح کردند و بنابراین احتمال یافتن گره‌های جمع‌آوری را افزایش دادند. این در حالی است که الگوریتم آنها از شمارش گام به جای فاصله اقلیدسی استفاده می‌کند. روش پیشنهادی آنها نشان داده است که از نظر مصرف انرژی نسبت به روشهای معمول بهتر عمل می کند.

از آنجایی که الگوریتم پیشنهادی ما به صورت توزیع شده می‌باشد و الگوریتم مورچگان نیز یک الگوریتم توزیع شده با نتایج بسیار بهتری نسبت به روش‌های تجمیع داده مشابه می‌باشد [۵] و الگوریتم‌های جدید نیز با این روش مقایسه

1. multicast
2. Ant aggregation
3. One hop
4. Liao

می‌شوند [٦]، از این رو ما نیز الگوریتم پیشنهادی خود را با این الگوریتم مقایسه خواهیم کرد. لازم به ذکر است که از آنجاییکه جزییات نحوه ساخت درخت تجمیع داده توسط الگوریتم مورچگان بسیار مفصل بوده و در این مقال نمیگنجد لذا خواننده را به [۵] ارجاع میدهیم.

در ادامه و در بخش ۲ مفاهیم الگوریتم شکل‌گیری رودخانه و نیز ایده‌ی اصلی الگوریتم را بیان می‌نماییم. در بخش ۳ الگوریتم اصلی شکل‌گیری پویای رودخانه و نحوه اعمال آن در ساخت درخت تجمیع داده در شبکه حسگر بی-سیم را توضیح می‌دهیم و در بخش آخر نیز به نتایج شبیه-سازی‌های مقاله می‌پردازیم.

۲. الگوریتم شکل‌گیری رودخانه

مسائل NP-Complete برای حل شدن در بدترین حالت به زمان نمایی احتیاج دارند.خوشبختانه روشهای مکاشفه‌ای می‌توانند به منظور حل تقریبا بهینه چنین مسائلی استفاده شوند که این کار را در زمان قابل قبولی انجام می‌دهند.

طبیعت سرچشمه بسیاری از الگوریتم‌های مکاشفه‌ای جذاب و مفید بوده است[۷-۹] که از بین آنها الگوریتم مورچگان حائز اهمیت بیشتری می‌باشد. این روش بر اساس چگونگی حرکت مورچگان برای پیدا کردن کوتاه‌ترین مسیر از کولونی تا منبع غذایی شکل گرفته است و نیز یک روش بسیار کارا و مشهور می‌باشد. مورچه‌ها هنگام حرکت از خود ماده‌ای به نام فرمون ترشح می‌کنند و تمایل دارند تا مسیر فرمون را دنبال کنند بدین ترتیب مسیرهای بهتر پررنگ تر میشوند و در طول زمان بلندتری، مسیرهای کوتاهتر بیشتر مورد توجه قرار می‌گیرند و نتیجتاً نسبت به مسیرهای بلند پررنگ‌تر می‌شوند. نهایتاً تنها یک مسیر کوتاه خوب غالب می‌شود و بقیه مسیرها ناپدید می‌شوند. یک مقدار فرمون به هر مسیر انتساب داده می‌شود و مورچگان به صورت احتمالی تمایل دارند که آن مسیرهایی را انتخاب کنند که نسبت میزان فرمون مقصد به هزینه مسیر بیشترین باشد.

متشابهاً، فرض می‌کنیم که تصمیم‌گیری مورچگان بر اساس شیب مقدار فرمون مسیر، بجای خود میزان فرمون مسیر باشد. در عمل، فرض کنیم که مورچگان به صورت احتمالی تمایل به انتخاب حرکت به سمت مسیری با بیشترین نسبت به صورت "تفاضل میزان فرمون مقصد بین مکان جدید و مکان کنونی به هزینه مقصد" باشد.

چه تفاوت‌هایی بین روش استاندارد و روش شیب‌دار می‌باشد؟ اولاً در روش استاندارد، مورچگان به شکلی توسط میزان رد پای فرمون هدایت می‌شوند که بعد از چند حرکت، غیر ممکن می‌باشد که یک گره را تکرار نکنند که برای مثال یک دور محلی به وجود می‌آید.اجازه دهید بیان کنیم که دنبال کردن یک دور از مسیرهای فرمونی این را بیان نمی‌کند که قبلاً نیز یک مورچه واقعاً این مسیر و دور را پیمایش کرده است. در عمل، هر قسمت از دور می‌تواند توسط مورچه‌های جداگانه تقویت شده باشد. در حالی که دنبال کردن یک دور در روش شیب‌دار غیر ممکن می‌باشد چرا که احتیاج به یک دور با شیب کاهش یافته می‌باشد که چنین چیزی امکان‌پذیر نمی‌باشد.

دوماً، در روش استاندارد زمانی که یک مورچه مسیر کوتاه‌تر را پیدا می‌کند، احتیاج به تعداد زیادی حرکت می‌باشد تا بقیه مورچه‌هایی که در حال دنبال کردن مسیر قبلی می‌باشند را راضی کند که مسیر جدید را دنبال کنند. از طرف دیگر، اگر تفاضل مسیر رد پاها در نظر گرفته شود، زمانی که یک مسیر کوتاه‌تر کشف می‌شود از همان لحظه به بعد این مسیر به صورت میانگین مورد علاقه بیشتری می‌باشد و نسبت به مسیرهای قدیمی‌تر ترجیح داده می‌شود و این بدین خاطر است که تفاضل میزان فرمون مسیرها بین مقصد نهایی و مبدأ در چنین مسیرهایی یکسان می‌باشد. (در واقع مقصد و مبدأ یکی می‌باشد) اما هزینه در مسیر کوتاهتر کمتر خواهد بود. بنابراین نسبت "تفاضل کلی مسیرها به هزینه کل" در مسیرهای کوتاهتر بیشتر می‌شود.

در نظر بگیرید که یک توده آب در یک ارتفاع بلند رها شود. جاذبه باعث می‌شود که این آب به سمت پایین حرکت کند تا جایی که دیگر نتواند پایین‌تر بروند. هنگامی که در کوهستان باران می‌بارد، آب سعی می‌کند مسیر خود را به سمت دریا پیدا کند. در طول مسیر، آب زمین را می‌ساید و

خاک و رسوبات را انتقال می‌دهد، که نهایتاً مسیر رودخانه را تشکیل می‌دهد. هنگامی که یک شیب شدید به سمت پایین توسط آب پیمایش می‌شود، تمام خاک‌های مسیر را جدا کرده و با خود می‌برد، سپس این رسوبات هنگامی که شیب کاهش می‌یابد رسوب کرده و ته‌نشین می‌شوند. رودها بر روی محیط پیرامون خود بر کاهش یا افزایش سطح ارتفاع زمین تأثیر می‌گذارند. در نظر بگیرید که اگر آب بر روی تمام نقاط سطح ریخته شود (مانند باران) پس شکل‌گیری رودخانه تمایل به بهینه‌سازی جمع‌آوری تمام آب‌ها و جمع‌آوری آنها در دریا دارد که این عملیات لزوماً به معنای ساخت کوتاه‌ترین مسیر از یک نقاط مبدأ به مقصد که دریا می‌باشد، نیست.

یک الگوریتم بر اساس این ایده بنام الگوریتم شکل‌گیری پویای رودخانه[1] (RFD) در [۱۰] ارائه شده است. به منظور اعمال در طرح پیشین، مورچه‌ها جای خود را به قطره‌ها و فرمون به ارتفاع می‌دهد. قطرات تمایل دارند که به سمت سطح شیب‌دار حرکت کرده و ارتفاع را تغییر دهند. RFD در چندین مسئله مانند مسئله فروشنده دوره‌گرد به کار برده شده است.[۱۰-۱۳].

پیدا کردن درخت پوشای مینیمم بهینه در یک گراف وزن‌دار یک مسئله‌ی NP نمی‌باشد و راه حل‌هایی همچون کروسکال و پرایم برای یافتن چنین درخت‌هایی با زمان چند جمله‌ای وجود دارد. اما درخت تجمیع داده نوع خاصی از درخت پوشا می‌باشد که در آن برخی از گره‌ها بایستی حتماً در درخت موجود باشند اما برخی دیگر لزومی ندارند که در درخت موجود باشند. به چنین درختی، درخت Steiner گفته می‌شود که مسئله‌ی NP-hard می‌باشد و ما در این تحقیق از الگوریتم RFD به منظور ساخت چنین درختی در شبکه حسگر بی‌سیم استفاده خواهیم کرد و با الگوریتم مورچگان مقایسه خواهیم نمود.

از آنجا که مسئله‌ی یافتن درخت تجمیع داده بهینه در شبکه‌ی حسگر بی‌سیم شامل یافتن نوعی کوتاه‌ترین مسیر می‌باشد، خاصیت‌های RFD که در قبل توضیح داده شد (عدم وجود دور محلی و تقویت و ساخت سریع‌تر مسیرهای کوتاه‌تر) باعث می‌شود که یک گزینه‌ی مناسب باشد.

۱,۲. ایده‌ی اصلی الگوریتم RFD

در این بخش، ساختار اساسی الگوریتم شکل‌گیری پویای رودخانه که در[۱۲] ارائه شد را بیان خواهیم کرد. روش بدین شکل کار می‌کند: بجای انتساب فرمون به هر یال در این روش یک ارتفاع به هر گره داده می‌شود. قطره‌ها هنگام حرکت کردن سطح را می‌سایند (کاهش ارتفاع) یا رسوبات را بر جا می‌نهند (افزایش ارتفاع). احتمال انتخاب یک یال از بین چندین یال به نسبت شیب پایین رونده سطح در آن یال می‌باشد که بستگی به تفاضل سطح ارتفاع دو گره روی یال و نیز فاصله (هزینه‌ی یال) آن یال دارد.

در ابتدا، یک سطح کاملاً مسطح را داریم که در واقع تمام گره‌ها یک سطح ارتفاع را دارند. تنها استثناء گره مقصد است که در عمل به عنوان دریا می‌باشد. قطره‌ها بر روی نقاط مبدأ قرار گرفته می‌شوند و بر روی سطح مسطح پخش می‌شوند تا زمانی که تعدادی از آنها به مقصد (دریا) برسند. این پروسه باعث می‌شود که گره‌های همسایه سایش یابند و باعث تولید سطح شیب‌دار پایین‌رونده جدید شود و بدین ترتیب پروسه سایش منتشر می‌شود. قطره‌های جدید بر روی مبدأها دوباره قرار می‌گیرند تا مسیرها انتقال یابند و مسیرهای ساخته شده توسط سایش تقویت یابند. بعد از چند مرحله، مسیرهای خوب از مبدأ به مقصد ساخته می‌شوند. این مسیرها به شکل دنباله‌ای از یال‌های کاهش یافته ارتفاعی از مبدأ به مقصد می‌باشد.

۳. الگوریتم اصلی

طرح اصلی الگوریتم RFD به صورت شبه کد زیر می‌باشد:

```
(initializeDrops);
(initializeNodes);
while (not allDropsFollowTheSamepath) and
(not otherEndingCondition)
```

1. River Formation Dynamics

(moveDrops);
(erodePathes);
(depositSediments);
(analyzePathes);
end while

این طرح ایده‌ی اصلی الگوریتم را نشان می‌دهد. در ابتدا، قطره‌ها مقدار اولیه می‌گیرند - (*initializedrops*)- برای مثال تمام قطره‌ها در گره‌های منبع واقع می‌شوند. مرحله دوم -(*initializenodos*)- شامل دو عملیات می‌شود، از یک طرف، ارتفاع گره مقصد صفر در نظر گرفته می‌شود که از دید الگوریتم شکل‌گیری رودخانه این گره همان دریا می‌باشد که هدف تمام قطره‌ها می‌باشد. از طرف دیگر، ارتفاع تمام گره-های باقیمانده به یک مقدار برابر مقداردهی می‌شود. حلقه *while* الگوریتم تا زمانی که تمام قطره‌ها یک راه حل را دنبال کنند که در واقع تمام قطره‌هایی که از یک گره رها می‌شوند از یک مسیر و یک سری یکسان گره عبور می‌کنند و به مقصد می‌رسند، یا یک شرط پایانی دیگر، که این شرط می‌تواند تعداد معینی تکرار باشد یا زمان اجرای الگوریتم باشد یا می‌تواند بدین شکل باشد که جواب بهینه برای تعداد معینی تکرار تغییر نکند.

گام اول درون حلقه - *moveDrops* - شامل حرکت دادن قطره‌ها روی گراف است، که به صورت شبه‌تصادفی می‌باشد. قانون انتقال، یک احتمال است که به شکل زیر می-باشد. احتمال اینکه قطره K از گره i به گره j حرکت کند به شکل زیر می‌باشد.

$$P_k(i,j) = \begin{cases} \frac{decreasingGradient(i,j)}{\sum_{l \epsilon V_k(i)} decreasingGradient(i,l)} & if \quad j \epsilon V_k(i) \\ 0 & if \quad j \notin V_k(i) \end{cases} \qquad (۱)$$

در رابطه فوق *VK*(i) مجموعه گره‌هایی می‌باشد که در همسایگی گره i است و می‌تواند توسط قطره k مشاهده و پیمایش شود. *decreasingGradient*(i,j) شیب بین گره i و j را نشان می‌دهد که به صورت زیر بیان می‌شود.

$$decreasingGradient(i,j) = \frac{altitude(j) - altitude(i)}{distance(i,j)} \qquad (۲)$$

که در آن *altitude*(x) ارتفاع گره x و *distance*(*i,j*) فاصله بین گره i و گره j می‌باشد که گره i و j را به هم وصل می‌کند. لازم به ذکر است که بدانیم در ابتدای الگوریتم، ارتفاع تمام گره‌ها یکسان است پس $\sum_{l \epsilon V_k(i)} decreasingGradient(i,l)$ برای تمام یال‌ها صفر می‌باشد. لذا به منظور برخورد با چنین مشکلی که شیب مسطح داریم این الگو را به شکل زیر تغییر می‌دهیم. فرض می کنیم احتمال اینکه یک قطره از یک یال با شیب صفر حرکت کند یک مقدار غیر صفر باشد. این باعث می‌شود که قطره‌ها بر روی یک سطح پخش شوند که در عمل برای شروع الگوریتم لازمه کار می‌باشد.

در حقیقت اگر یک مرحله جلوتر برویم این پیشرفت را نیز می‌توانیم داشته باشیم. ما می‌توانیم قطره‌هایی را داشته باشیم که از شیب بالا بروند که البته احتمال آن کم می‌باشد. این احتمال متناسب با شیب می‌باشد و در طول اجرای الگوریتم کاهش می‌یابد. چیزی شبیه آنچه در الگوریتم *Simulated Annealing* می‌باشد. این خاصیت جدید باعث یافتن راه حل‌های مناسبی می‌شود. بدین ترتیب راه حل‌های اولیه که یافته می‌شوند گراف را به صورت سراسری جستجو[1] کرده‌اند.

مکانیسم بالا رفتن قطرات از شیب به شکل زیر کار می-کند. قطره d که در گره k واقع شده است را درنظربگیرید. الگوریتم به صورت تصادفی تصمیم می‌گیرد که آیا قطره d با احتمال زیر به سمت بالا حرکت می‌کند یا خیر.

$$P(d) = \frac{1}{notClimbingFactor} \qquad (۳)$$

notClimbingFactor یک متغیر می‌باشد که در ابتدا مقدار 1 را دارد و به صورت تدریجی و در هر گام الگوریتم افزایش می‌یابد اما هر از N تکرار، این متغیر افزایش نمی‌یابد بلکه کاهش می‌یابد. زمانی که قطره اجازه یافت تا به بالا برود

1. exploration

و حرکت بعدی خود را انتخاب کرد می‌تواند به سمت بالا حرکت خود را ادامه دهد. در عمل قانون انتقال یک قطره بالا رونده به شکل زیر می‌باشد.

$$P_k(i,j) = \begin{cases} \frac{decreasingGradient(i,j)}{total} & if \quad j \epsilon V_k(i) \\ \frac{\omega/|decreasingGradient(i,j)|}{total} & if \quad j \epsilon U_k(i) \\ \frac{\delta}{total} & if \quad j \epsilon F_k(i) \end{cases} \quad (٤)$$

که در آن

$$total = \left(\sum_{l \epsilon V_k(i)} decreasingGradient(i,l)\right) + \sum_{l \epsilon U_k(i)} \left(\frac{\omega}{|decreasingGradient(i,l)|}\right) + \sum_{l \epsilon F_k(i)} \delta \quad (٥)$$

و $FK(i)$, $UK(i)$, $Vk(i)$ مجموعه گره‌هایی می‌باشند که با گره i همسایه می‌باشند و به ترتیب یک شیب پایین رونده، بالا رونده و مسطح دارند. در ضمن δ و ω پارامترهای الگوریتم می‌باشند. از طرف دیگر اگر گرهی به عنوان گره بالارونده در نظر گرفته نشد تنها می‌تواند شیب پایین رونده و مسطح را انتخاب کند.

در فاز بعد–(*erodepaths*) – مسیرها سایش می‌یابند که این عمل با توجه به حرکت قطره‌ها در فاز قبل صورت می‌پذیرد. در عمل اگر یک قطره از گره A به گره B حرکت کند باعث سایش گره A می‌شود. کاهش ارتفاع این گره بسته به شیب در حال حاضر بین گره A و B می‌باشد. در عمل مقدار سایش بیشتراست، اگر شیب پایین رونده بین A و B بیشتر باشد. ارتفاع گره A با توجه به فرمول زیر تغییر می‌یابد:

$$altitude(A) := altitude(A) - erosion(A,B) \quad (٦)$$

$$erosion(A,B) = \frac{paramErosion}{(numNodes-1).numDrops} . decreasingGradient(A,B) \quad (٧)$$

که در آن *paramErosion* یک پارامتر سایش، *numNodes* تعداد گره‌های گراف و *numDrops* تعداد قطره‌های الگوریتم در گراف می‌باشد. برعکس زمانی که یال مسطح یا بالا رونده باشد آنگاه یک مقدار کم سایش صورت می‌گیرد. به هر حال ارتفاع گره نهایی (دریا) هرگز تغییر نخواهد کرد و همیشه مقدار ثابت صفر را دارد.

زمانی که پروسه سایش به اتمام رسید، ارتفاع گره‌های گراف به تدریج افزایش می‌یابد –(*depositSediments*) – وهدف اصلی این می‌باشد که اجتناب کنیم که بعد از چند مرحله، پروسه سایش منجر به این شود که اکثر گره‌ها ارتفاع نزدیک صفر داشته باشند. در عمل ارتفاع یک گره N با توجه به فرمول زیر افزایش می‌یابد:

$$altitude(N) := altitude(N) + \left(\frac{erosionProduced}{numNodes-1}\right) \quad (٨)$$

که در آن *erosionProduced* مجموع تمام سایش‌های مربوط به فاز قبل می‌باشد که در تمام گره‌ها به وجود آمده است و *numNodes* تعداد گره‌های گراف می‌باشد.

این الگوریتم همچنین باعث می‌شود که گره‌ها مقداری رسوب را بر روی گره دیگر ته‌نشین کنند. این پروسه زمانی اتفاق می‌افتد که تمام حرکت‌های موجود برای یک گره از نوع بالابرنده باشد و قطره نمی‌تواند از هیچ یالی بالا رود (با توجه به احتمال که به آن انتساب داده شده است). در این حالت قطره بلوکه می‌شود و مقداری از رسوباتی که حمل می‌کرده در آن گره ته نشین می‌شود و باعث می‌شود که ارتفاع گره به نسبت مقدار رسوبی که حمل می‌کرده افزایش یابد و این مقدار متناسب با مقدار سایشی می‌باشد که در مرحله قبل انجام داده است. اگر یک قطره در یک گره N بلوکه شود آنگاه ارتفاع گره N با توجه به رابطه زیر افزایش می‌یابد:

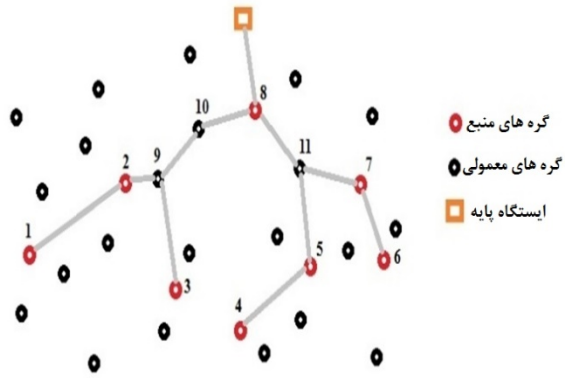
$$altitude(N) := altitude(N) + paramBlockedDrop.cumulatedSediment \quad (٩)$$

در رابطه فوق *paramBlockedDrop* یک پارامتر و *cumulatedSediment* مقدار رسوبی است که توسط قطره حمل شده است.

آخرین فاز -*AnalyzePaths* – تمام راه حل‌هایی که تا به حال پیدا شده را آنالیز کرده و بهترین را ذخیره می‌کند.

۱،۳. اعمال RFD در ساخت درخت تجمیع داده در شبکه حس گر بی‌سیم

- درخت تجمیع داده در شبکه حسگر بی سیم یک درخت می‌باشد که ریشه آن ایستگاه پایه می‌باشد و گره‌های منبع- گره‌هایی که قرار است داده‌هایی را ارسال نمایند - یا برگ‌ها یا گره‌های میانی درون درخت می باشند. مابقی گره‌های درون شبکه یا هیچ نقشی در این درخت نداشته یا این که به عنوان گره-های میانی در درخت می باشند. برخی از گره‌های میانی تنها وظیفه ارسال داده را دارند که به آنها گره بازپخش[1] می گوییم - مانند گره ۱۰- و برخی دیگر وظیفه تجمیع[2] داده را بر عهده دارند- مانند گره ۱۱-. شکل (۱) یک درخت تجمیع داده را نشان می دهد. مسئله، ساخت بهینه ترین درخت می‌باشد که البته معیار بهینه بودن می‌تواند پارامترهای متفاوتی باشد.



شکل (۱): درخت تجمیع داده

در این بخش چگونگی اعمال الگوریتم RFD به منظور ساخت درخت تجمیع داده‌ی بهینه را توضیح خواهیم داد. لازم به ذکر است که تغییراتی در الگوریتم اصلی RFD اعمال گردیده است که منجر به افزایش کارایی الگوریتم در ساخت درخت تجمیع داده درون شبکه حسگر خواهد شد. بخش های اصلی الگوریتم را در بخش قبل توضیح دادیم، در این قسمت ما جزییات ریزتر الگوریتم و قسمت هایی که تغییراتی را در آن اعمال نموده ایم و مربوط به ساخت درخت تجمیع داده می‌باشد را بیان خواهیم کرد.

در فاز اول یعنی (*initializeDrops*) تنها گره‌های منبع تعدادی قطره را میسازند، تعداد این قطره‌ها در شبیه‌سازی ما برابر 100 در نظرگرفته شده است.

در فاز (*initializeNodes*) ارتفاع گره‌ها را برابر با نسبتی از تعداد گام‌هایش تا ایستگاه پایه قرار می دهیم. برای این منظور، در ابتدا ایستگاه پایه به صورت سیل‌آسا پیغامی حاوی ID خود را در شبکه پخش می‌کند. هر گره در شبکه که این پیغام را دریافت می‌کند پیغام را به همسایگانش باز پخش می-نماید. تعداد گام‌های گره به ایستگاه پایه زمانی به‌روزرسانی می‌شود که پیغامی را از همسایه دریافت کند و تعداد گام‌های این پیغام بعلاوه یک، از تعداد گام‌های فعلی گره کمتر باشد. حلقه *while* به مدت زمان مشخصی تکرار را انجام می‌دهد که این تکرار برابر با تعداد قطره‌ها می‌باشد.

در فاز (*moveDrops*) ما تغییراتی را بر روی عبارت *decreasingGradient* اعمال نموده ایم. در زیر عبارت اصلاح شده را مشاهده می نمایید.

$$decreasingGradient(i,j) = \begin{cases} 90-\gamma & if\ \ gradient(i,j) < 0 \\ 90-\eta & if\ \ gradient(i,j) = 0 \\ 180-\gamma & if\ \ gradient(i,j) > 0 \end{cases} \quad (۱۰)$$

$$\gamma = |ArcTg(|gradient(i,j)|)|.\left({}^{180}/_{\pi}\right) \quad (۱۱)$$

$$Gradient(i,j) = (altitude(i) - altitude(j))/((W_1 * distance(i,j)) * (W_2 * Energy(j)) \quad (۱۲)$$

$$\eta = edgeCost/\mu \quad (۱۳)$$

که در آن Υ زاویه (به درجه) را نمایش می‌دهد. η به منظور برخورد با مشکل مسطح بودن دوگره می‌باشد که پیش از این نیز درباره آن صحبت نمودیم (پارامتر μ برابر ۵ در نظر گرفته شد). *Gradiant*(i,j) علاوه بر معیار فاصله بین دو گره،

1. Relay nodes

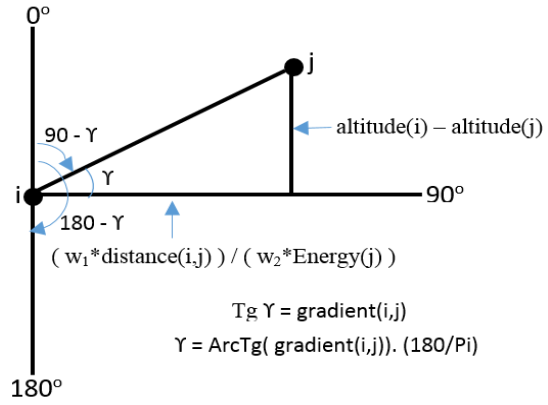
2. Aggregation

پارامتر انرژی را نیز در خود جای داده است. چرا که در شبکه حسگر بیسیم تنها معیار فاصله برای انتخاب گره بعدی برای ارسال داده نمی‌تواند معیار مناسبی باشد. اگر تنها معیار فاصله را در نظر بگیریم می‌تواند منجر به ساخت درخت‌های مشابهی در هر مرحله شود و متعاقباً انرژی گره‌های درون درخت به سرعت کاهش می یابد و این در حالی است که دیگر گره‌ها انرژی خود را حفظ نموده‌اند. پس به سرعت برخی از گره‌ها انرژی خود را از دست داده و منجر به مرگ آن گره و متعاقباً کل شبکه خواهد شد. پس علاوه بر در نظر گرفتن معیار فاصله برای انتخاب گره بعدی بایستی انرژی باقیمانده گره همسایه را نیز در نظر گرفت. در ضمن به منظور متوازن کردن[1] عبارت، دو پارامتر w1 و w2 را به عبارت اضافه نموده‌ایم. مقادیر این دو پارامتر با توجه به نوع شبکه که می‌تواند فشرده یا پراکنده باشد و همچنین طول عمر شبکه متفاوت می‌باشد. برای مثال اگر در ابتدای عمر شبکه باشیم که تمام گره‌ها انرژی اولیه خود را دارند معیار فاصله بسیار مهمتر می‌باشد پس پارامتر w1 بسیار بزرگ تر از w2 است. اما در انتهای عمر شبکه معیار انرژی باقیمانده بیشتر بوده و لذا پارامتر w2 بزرگ‌تر از w1 می‌باشد. برای مثال می‌توان از روابط زیر بهره برد.

$$w1 = \frac{\text{all nodes remaining energy}}{\text{all nodes initial energy}} \qquad (۱۴)$$

$$w2 = \frac{\text{all nodes consumed energy}}{\text{all nodes initial energy}} \qquad (۱۵)$$

شایان ذکر است که ما فرض کرده‌ایم که $0 \le decreasingGradient \le 180$ بنابراین اگر مقدار *decreasingGradient* برابر با صفر باشد یعنی اینکه قطره از یک دیوار عمودی در حال بالا رفتن می‌باشد که همانطور که پیش از این نیز صحبت شد احتمال این کار بسیار ضعیف است. اگر *decreasingGradient* برابر با ۹۰ باشد بدین معنی است که قطره روی یک سطح هم ارتفاع در حال حرکت می‌باشد. به همین ترتیب اگر برابر با ۱۸۰ باشد یعنی اینکه به صورت عمودی در حال پایین آمدن می‌باشد. در شکل (۲) یک نمایش گرافیکی از این پارامترها نمایش داده شده است.



شکل (۲): پارامترها در محاسبه Gradient decreasing

۲،۳. جزییات پیاده‌سازی الگوریتم

در این بخش جزییات پیاده سازی و اعمال الگوریتم RFD در شبکه حسگر بی‌سیم و ساختار پاکت های ارسالی را توضیح خواهیم داد.

قطره‌ها در واقع پیغام های کنترل می‌باشند و هیچ داده‌ای را با خود حمل نمی‌کنند. قطره‌ها صرفا از منابع به سمت ایستگاه پایه حرکت می‌کنند تا درخت تجمیع داده بهینه را بسازند. پاکت قطره به شکل (۳) می‌باشد.

pkt-typ	src-ID	snd-ID	rcv-ID	hop

شکل (۳): قالب پاکت کنترلی قطره

pkt-typ مشخص کننده پاکت قطره می‌باشد. *Src-ID* مشخص کننده *ID* مربوط به گره منبعی است که قطره از آن به حرکت درآمده است. *Snd-ID* بیانگر گره‌ی می‌باشد که پاکت را به گره بعد ارسال می‌نماید. *rcv-ID* نشان‌دهنده‌ی گره بعدی می‌باشد که قطره در فاز *moveDrops* به عنوان گام بعد انتخاب کرده است و *hop* مشخص‌کننده‌ی تعداد گام-ها از منبع تا آن گره می‌باشد.

هر گره یک جدول واکنش[2] را در خود ذخیره می‌کند. این جدول در شکل (٤) نشان داده شده است.

src-ID	prv-ID	rcv-ID

شکل (٤): جدول واکنش گره‌ها

1. tuning

2. Reaction table

هنگامی که گرهی پاکت قطره را دریافت نمود اطلاعات پاکت جدید را به درون جدول اضافه می نماید. بدین ترتیب که مقدار *src-ID* و *snd-ID* مربوط به پاکت را به ترتیب در فیلدهای *src-ID* و *prv-ID* جدول قرار می‌دهد و مقدار *rcv-ID* را برابر با *ID* گره بعدی قرار می‌دهد. این جدول به منظور ساخت درخت استفاده می‌شود.

هنگامی که گرهی پاکت قطره جدیدی را دریافت می‌کند ابتدا گره بعدی را مشخص می‌کند، اگر قبلا قطره‌ای را به آن گره ارسال کرده مقدار گره بعدی در فیلد *rcv-ID* جدول موجود باشد پاکت قطره را با مقدار hop قبلی به گره بعد ارسال می‌نماید اما اگر تابه حال به گره بعدی قطره‌ای را ارسال ننموده است مقدار hop+1 را در فیلد hop قرار می‌دهد. دلیل این امر به دست آوردن تعداد یال‌های درون درخت تجمیع داده توسط ایستگاه پایه می‌باشد.

در فاز (*analyzePaths*) مجموع تمام hop های پاکت-های قطره دریافت شده توسط ایستگاه پایه که بیانگر هزینه[1] درخت تجمیع داده می‌باشد، محاسبه می‌شود. دو حالت ممکن است اتفاق بیافتد:

۱) در صورتی که هزینه درخت فعلی از درخت بهینه تاکنون بهتر –کمتر- باشد بایستی راه حل جدید – درخت تجمیع داده جدید- را به اطلاع گره‌ها برساند و مقدار هزینه درخت به عنوان درخت بهینه جدید در ایستگاه پایه ذخیره شود. برای این منظور یک پاکت کنترلی جدید را به گره‌هایی که در درخت شرکت داشته‌اند ارسال می‌نماید. برای انجام این کار از جدول واکنش درون گره‌ها استفاده می‌کند. پاکت کنترلی به شکل (۵) می‌باشد.

pkt-typ	rcv-ID	altitude	costSolution

شکل (۵): قالب پاکت کنترلی درخت بهینه

pkt-typ بیانگر پاکت درخت بهینه و *rcv-ID* نشان-دهنده‌ی *ID* گره دریافت کننده‌ی پاکت می‌باشد. *Altitude* بیانگر ارتفاع گره فرستنده پیغام و *costSolution* نیز هزینه درخت می‌باشد که برای مرحله سایش مورد استفاده قرار خواهند گرفت. هنگامی که گرهی این نوع پیغام را دریافت نمود جدول بهترین خود را بایستی به‌روزرسانی کند. جدول بهترین به‌صورت شکل (٦) می‌باشد.

src-ID	nxt-node

شکل(٦): جدول بهترین گام بعدی

که در آن *src-ID* مشخص کننده گره منبع و *nxt-node* مشخص کننده بهترین گره بعدی از این گره می‌باشد. مقادیر *src-ID* و *nxt-node* برابر مقادیر *src-ID* و *rec-ID* از جدول واکنش می‌گردد.

۲) در صورتی که هزینه درخت فعلی از درخت بهینه تاکنون، بهتر نباشد بایستی تنها مقدار هزینه درخت جدید را به اطلاع گره‌های به‌کار گرفته شده در درخت رسانید تا ارتفاع خود را تغییر دهند. برای این منظور ایستگاه پایه، پاکت کنترلی قبلی را به گره‌هایی که در درخت شرکت داشته‌اند ارسال می-نماید. *pkt-typ* این پیغام بیانگر پاکت درخت غیربهینه می-باشد. *rcv-ID* نشان‌دهنده *ID* گره دریافت کننده پاکت می-باشد.*Altitude* بیانگر ارتفاع گره فرستنده پیغام و *costSolution* نیز هزینه درخت می‌باشد که برای مرحله سایش مورد استفاده قرار خواهند گرفت. هنگامی که گرهی این نوع پیغام را دریافت نمود جدول بهترین خود را به-روزرسانی نخواهدکرد.

پس از اینکه تمام قطره‌ها به حرکت در آمدند و هر بار درخت تجمیع داده‌ی جدید را ساختند آنگاه بهترین درخت به عنوان درخت تجمیع داده بهینه، برای انتقال پیام حاوی داده از طریق جدول بهترین، در نظر گرفته می‌شود.

٤. شبیه‌سازی

در این تحقیق گم شدن و از بین رفتن بسته های داده حین انتقال، درنظر گرفته نمی‌شود. طول عمر شبکه مجزا[2] است و

1. Cost Solution

2. discrete

با تعداد دفعات اجراها درنظر گرفته می‌شود. در طول هر اجرا، یک درخت تجمیعی داده ساخته می‌شود و همه منابع داده خود را از طریق این درخت به ایستگاه پایه ارسال می‌کنند. برای محاسبه بهره‌وری انرژی الگوریتم پیشنهادی، از مدل انرژی مرتبه اول[1] که در [۱٤] ارائه شده، برای محاسبه مجموع انرژی مصرف شده در شبکه، استفاده می‌شود. در این مدل میزان انرژی E_{Tx} مصرف شده توسط یک گره برای انتقال k بیت پیام از طریق مسیری به طول d ، با استفاده از روابط زیر محاسبه می‌شود. در ضمن برای دریافت این k بیت پیام، انرژی مصرف شده E_{Rx} توسط یک گره، با استفاده از [۱٤] محاسبه می‌گردد.

$$E_{Tx}=E_{elec}\times k+\varepsilon_{amp}\times k\times d^2 \qquad E_{Rx}=E_{elec}\times k \qquad (۱۷)$$

که در آن E_{elec} انرژی مورد نیاز برای کار کردن مدار الکتریکی انتقال دهنده یا دریافت کننده و ϵ_{amp} انرژی مصرف شده توسط آمپلی فایر انتقال دهنده می‌باشد.

نتایج شبیه‌سازی‌های ما بر اساس شبیه‌سازی یک شبکه حسگر بی‌سیم با تعداد گره‌های N = 300 و شعاع R=12m برد ارتباطی برای هر گره که به‌صورت تصادفی در یک مربع 100*100m2 پخش شده‌اند، می‌باشد. پارامترهای شبیه‌سازی در جدول (۱) نمایش داده شده است.

جدول (۱): پارامترهای شبیه‌سازی

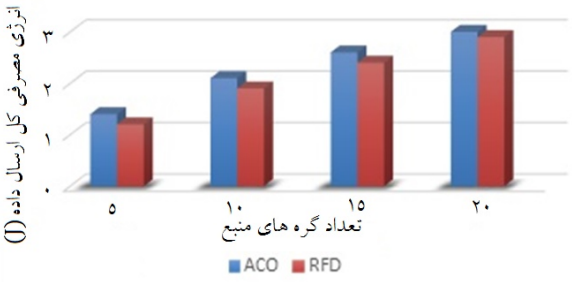
توضیحات	مقدار	پارامتر
تعداد کل گره‌ها	300	N
تعداد گره‌های منبع	5-20	S
برد ارتباطی گره‌ها	12 m	R
فرستنده-گیرنده	50 nJ/bit	E_{elec}
آمپلی‌فایر	0.1 nJ/bit/m^2	ϵ_{amp}
اندازه پاکت کنترلی	8 Byte	*ctr-pkt*
اندازه پاکت داده	500 Byte	*Data-pkt*
انرژی اولیه گره‌ها	0.5 J	E_{int}

جدول (۲) مقادیر اولیه و پارامترهای الگوریتم RFD را بیان می‌نماید.تمامی گره‌ها یکسان بوده و دارای انرژی اولیه یکسان می‌باشند. گره‌های منبع داده‌ها را جمع‌آوری کرده و سپس از طریق درخت تجمیع بدست آمده به ایستگاه پایه ارسال می‌کند. ایستگاه پایه در مرکز مربع یعنی نقطه (50,50) واقع شده است.

جدول (۲) : پارامترهای RFD

توضیحات	مقدار	پارامتر
تعداد قطره‌ها در منابع	100	*Initialdrops*
ضریب سایش	0.1	$\emptyset$
ضریب شیب	0.02	θ

اولین آزمایش بر روی انرژی کل مصرفی برای ارسال اطلاعات درون شبکه حسگر بی‌سیم بعد از تعداد معینی اجرا می‌باشد. نتایج بیان شده در شکل‌ها، حاصل ۱۰۰ بار اجرای الگوریتم‌ها بر روی شبکه حسگر بی‌سیم می‌باشد. این نتایج با تعداد گره‌های منبع متفاوت در شکل (۷) نشان داده شده است.



شکل (۷): مقایسه انرژی مصرفی برای ارسال داده‌ها توسط الگوریتم‌های مختلف

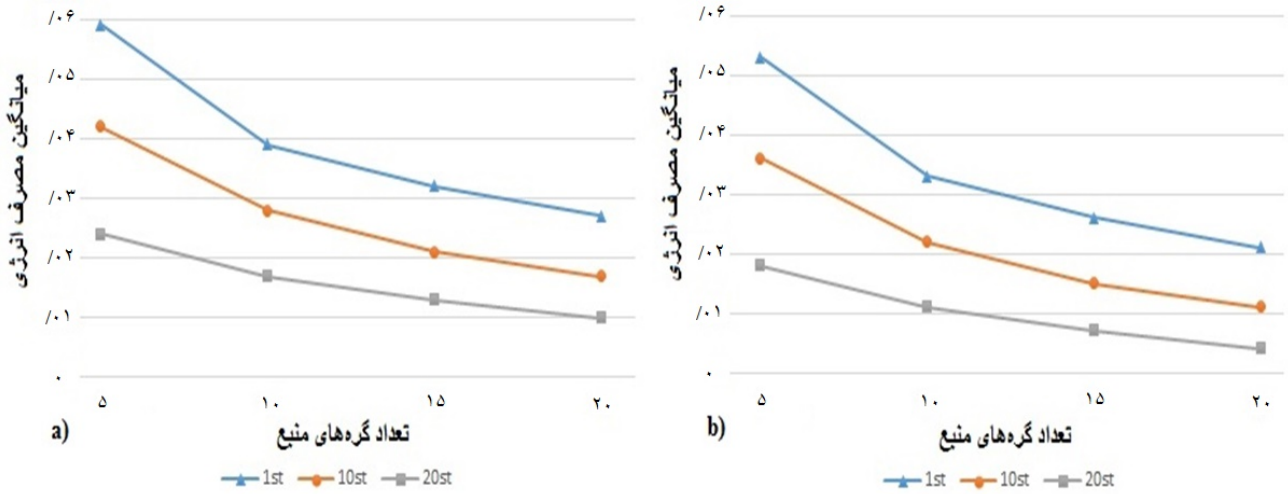
شکل (۷) مصرف انرژی شبکه حسگر برای ارسال داده هنگام استفاده از الگوریتم RFD و ACO را نشان می‌دهد. همانطور که در شکل مشاهده می‌شود شبکه حسگر که از الگوریتم RFD برای ارسال اطلاعات استفاده می‌کند انرژی مصرفی کمتری را نسبت به ACO مصرف می‌کند. دلیل این امر این واقعیت می‌باشد که مسیرهای ساخته شده در روش RFD بهینه‌تر از مسیرهای ACO است.

شکل (۸) بیانگر میانگین انرژی مصرفی به ازای هر گره منبع می‌باشد که قرار است داده‌ها را به ایستگاه پایه ارسال

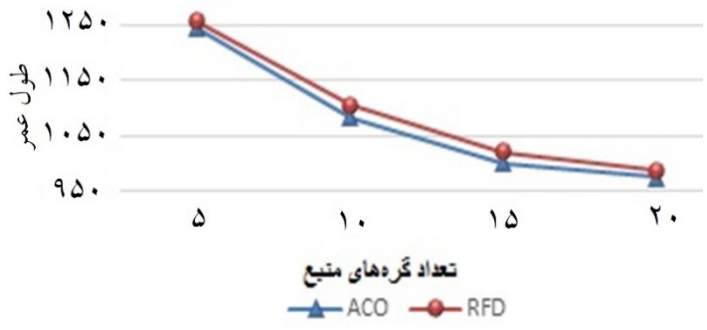
1. First order energy

نماید. بعد از اولین اجرا میانگین مصرف انرژی برای تمام گره‌های منبع متفاوت تقریباً یکسان می‌باشد. دلیل این امر این واقعیت است که درخت تجمیع داده به صورت تقریبا تصادفی ساخته می‌شود. بنابراین میانگین انرژی مصرف شده توسط هر گره منبع برای ارسال داده برابر با شبکه‌هایی با تعدادی گره-های منبع می‌باشد. بعد از ۱۰مین، ۲۰مین و ۳۰مین و ... اجرا درخت تجمیع داده به تدریج بهینه می‌شود. هنگامی که تعداد گره‌های منبع بیشتر می‌باشند، داده‌های تجمیع شده بیشتری در درخت بهینه به وجود می‌آید. به هرحال مقدار کل داده ارسال شده درون شبکه کاهش می‌یابد که در نتیجه منجر به کاهش مصرف متوسط به ازای هر گره منبع می‌شود.

در آزمایش بعدی طول عمر شبکه مورد بررسی قرار گرفته است. طول عمر شبکه به تعداد دفعات اجرای الگوریتم مسیریابی بر روی شبکه که منجر به مرگ اولین گره در شبکه می‌گردد، گفته می‌شود. شکل (۹) طول عمر شبکه را به واسطه استفاده از الگوریتم RFD و ACO نمایش می‌دهد. همانگونه که از شکل نیز پیدا می‌باشد، طول عمر شبکه به واسطه استفاده از الگوریتم RFD بهتر از الگوریتم ACO می‌باشد. دلیل این اتفاق، یافتن درخت تجمیع داده بهینه‌تر است که در آن گره-های تجمیع نزدیک‌تر به گره‌های منبع و همچنین کاهش طول کل درخت می‌باشد. در ضمن یکی از دلایل عمده افزایش طول عمر شبکه- ای که از الگوریتم RFD استفاده کرده است، به کار بردن پارامتر مقدار انرژی باقیمانده در گره‌ها هنگام ساخت درخت تجمیع داده می‌باشد که پیش از این در بخش‌های پیشین مفصلا درباره آن صحبت شد.



شکل (۸) میانگین انرژی مصرفی توسط الگوریتم‌های a) ACO b) RFD



شکل (۹): مقایسه طول عمر شبکه برای الگوریتم‌های مختلف

۵. نتیجه‌گیری

در این تحقیق ما چگونگی اعمال الگوریتم RFD به منظور ساخت درخت تجمیع داده‌ی بهینه را توضیح دادیم و دریافتیم که الگوریتم RFD مناسب برای ساخت چنین درخت‌هایی در شبکه حسگر بی‌سیم می‌باشد. جزییات پیاده سازی الگوریتم را بر روی شبکه حسگر بیان نمودیم و سپس با اعمال تغییراتی بر روی الگوریتم اصلی منجر به بهبود الگوریتم و نتیجتاً افزایش طول عمر شبکه حسگر شدیم.

نتایج شبیه‌سازی‌ها که در مقایسه با الگوریتم بسیار مشهور کولونی مورچگان صورت پذیرفت، نشان دهنده کیفیت الگوریتم پیشنهادی بود که دلیل این امر برتری نسبی الگوریتم شکل‌گیری رودخانه در حل چنین مسئله‌ای می‌باشد. این روش چندین مزیت را نسبت به الگوریتم مورچگان فراهم می‌کند. از یک طرف، دور محلی ساخته نمی‌شود. در ضمن با توجه به اینکه قطره‌ها نگران دور محلی نیستند، در حالت کلی قطره‌ها لازم ندارند که حرکات پیشین خود را در حافظه نگهداری کنند که این باعث می‌شود محاسبات حافظه‌ای کاهش یابد و نیز منجر به کاهش زمان اجرا می‌شود. از طرف دیگر زمانی که یک مسیر کوتاهتر در RFD پیدا می‌شود، پر رنگ کردن و تقویت مسیر جدید بسیار سریع‌تر می‌باشد.

در کل می‌توان نتیجه گرفت که استفاده از الگوریتم شکل‌گیری رودخانه که ذاتا بسیار شبیه به ساخت درخت تجمیع داده می‌باشد، از نظر مصرف انرژی بسیار کارآمد بوده و لذا منجر به ساخت درخت‌های تجمیع داده‌ای می‌گردد که مصرف انرژی در شبکه را پخش نموده و منجر به افزایش طول عمر کلی شبکه می‌شود.

سپاسگزاری

سپاس فراوان از دانشگاه آزاد اسلامی واحد داریون که کلیه اعتبارات مالی این مقاله توسط معاونت پژوهش و فناوری این واحد تأمین گردید.

مراجع

[1] B. Krishnamachari, D. Estrin, and S. Wicker, "The impact of data aggregation in wireless sensor networks", *22nd International Conference on Distributed Computing Systems*, pp. 575-578, 2002.

[2] J. N. Al-Karaki, R. Ul-Mustafa, and A. E. Kamal, "Data aggregation in wireless sensor networks - exact and approximate algorithms", *Workshop on High Performance Switching and Routing, HPSR*, pp. 241-245, 2004.

[3] L. Deying, C. Jiannong, L. Ming, and Z. Yuan, "Construction of Optimal Data Aggregation Trees for Wireless Sensor Networks", *15th International Conference on Computer Communications and Networks, ICCCN*, pp. 475-480, 2006.

[4] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of network density on data aggregation in wireless sensor networks", *22nd International Conference on Distributed Computing Systems*, pp. 457-458 *, 2002.*

[5] W.-H. Liao, Y. Kao, and C.-M. Fan, "Data aggregation in wireless sensor networks using ant colony algorithm", *Journal of Network and Computer Applications,* vol. 31, pp. 387-401, 2008.

[6] D. C. Hoang, R. Kumar, and S. K. Panda, "Optimal data aggregation tree in wireless sensor networks based on intelligent water drops algorithm", *Wireless Sensor Systems, IET,* vol. 2, pp. 282-292, 2012.

[7] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by Simulated Annealing", *Science,* vol. 220, pp. 671-680, 1983.

[8] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization", *IEEE Computational Intelligence Magazine,* vol. 1, pp. 28-39, 2006.

[9] R. Poli, J. Kennedy, and T. Blackwell, "Particle swarm optimization", *Swarm Intelligence,* vol. 1, pp. 33-57, 2007.

[10] P. Rabanal, I. Rodríguez, and F. Rubio, "Using River Formation Dynamics to Design Heuristic Algorithms", *Unconventional Computation,* vol. 4618, Springer Berlin Heidelberg, pp. 163-177, 2007.

[11] P .Rabanal, I. Rodriguez, and F. Rubio, "Solving Dynamic TSP by Using River Formation Dynamics", *4th International Conference on Natural Computation, ICNC*, pp. 246-250, 2008.

[12] P. Rabanal, I. Rodríguez, and F. Rubio, "Applying River Formation Dynamics to Solve NP-Complete Problems", *Nature-Inspired Algorithms for Optimisation*, vol. 193, Springer Berlin Heidelberg, pp. 333-368, 2009.

[13] P. Rabanal, Rodri, x, I. guez, and F. Rubio, "Applying River Formation Dynamics to the Steiner Tree Problem", *9th IEEE International Conference on Cognitive Informatics, ICCI,* pp. 704-711, *2010.*

[14] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks", *33rd Annual Hawaii International Conference on System Sciences,* vol. 2, *2000.*