

**ذخیره سازی داده ها و بازیابی با مسیریابی RPL**

**چکیده**

در حالاتی مانند نظارت بر مناطق جدا شده، زمانی که گره مرز یک شبکه، یک ارتباط دائمی با اینترنت نداشته باشد، شبکه ‌ های حسگر بی‌سیم (WSN ها) خواستار تکنیک ‌ های ذخیره‌سازی داده انعطاف ‌پذیر در شبکه ‌ها برای به‌حداقل رساندن خطر از دست دادن داده‌ها هستند. بهره‌وری این تکنیک‌ها می‌تواند تا حد زیادی بهره‌برداری از اطلاعات شبکه را بهبود دهد، مانند مسیر‌یابی که توسط پروتکل استفاده می‌شود. به ‌طورخاص، یکی از پروتکل‌ های مورد استفاده در اینترنت اشیا (IoT) پروتکل مسیریابی IPv6 برای مصرف برق کم و شبکه ‌ های پراتلاف است (RPL). در این مقاله، یک ذخیره ‌سازی توزیع شده‌ی داده ‌ها و مکانیسم بازیابی برای افزایش انعطاف‌پذیری و ظرفیت ذخیره ‌سازی از یک WSN بر اساس RPL در برابر کمبود حافظه محلی پیشنهاد می‌کنیم. این رویکرد در سیستم‌ عامل Contiki و از طریق تجزیه ‌و تحلیل گسترده با شبیه ‌ساز Cooja ارزیابی می‌شود.

**کلید واژه‌ها:** ذخیره‌سازی توزیع‌ شده‌ی داده‌ها؛ RPL؛ تکرار داده‌ها؛ بازیابی داده‌ها؛ Contiki وCooja.

**1. مقدمه**

در مقابل شبکه معمولی ذخیره ‌سازی داده‌ها، ذخیره ‌سازی داده‌ ها در شبکه‌ های حسگر بی‌سیم (WSN) یک چالش به‌ دلیل محدودیت قدرت، حافظه و ارتباطات پهنای باند شبکه گیرنده بی‌سیم را نشان می‌دهد. به‌تازگی، سنسورها به قابلیت‌ های بالاتری از نظر سرعت پردازش و ذخیره‌سازی محلی، نسبت به سال گذشته رسیده‌اند [1]، که آنها را برای ذخیره‌سازی در شبکه جذاب‌تر می‌کند.

به‌ طور معمول WSN ها از مجموع های از گره‌ های بدون مراقبت برای پویش محیط زیست اطراف آن تشکیل شده است و گره sink مسئول جمع‌آوری داده‌ های اندازه‌گیری و بازپخش کردن به یک نهاد مدیریتی است. دلایل متعددی وجود دارد که ممکن است یک گره حسگر از انتقال داده‌ها به گره sink بعد از به‌دست آوردن آن جلوگیری کند. به‌عنوان مثال، ممکن است گره‌ های حسگر همیشه قادر به رسیدن گره با توجه به قطع شدن موقعیت لینک و یا عملیات چرخه‎ی وظیفه در گره نباشد. علاوه براین، وقتی برنامه‌ های کاربردی به مجموعه زمان واقعی نیاز ندارند، ذخیره‌سازی واحد داده‌ها و ارسال اطلاعات جمع‌آوری شده می‌تواند به‌کمک میزان انتقال رادیویی، که موجب افزایش طول عمر عملیات WSN می‌شود کاهش یابد. برنامه‌ های کاربردی شامل نظارت زیستگاه، مانند ردیابی مهاجرت حیوانات در مناطق دور افتاده [2]، مطالعه شرایط آب و هوایی در پارک‌ های ملی [3] و غیره هستند. چنین سناریو هایی نیاز به جمع‌آوری و ذخیره‌ی داده‌ها تا جایی که ممکن است بین دو بازیابی متوالی داده توسط یک عامل خارجی انجام می‌شود. با این‌حال، ذخیره‌سازی داده در گره حسگر منجر به سرریز حافظه محلی می‌شود در صورت که بازیابی داده‌ها به موقع توسط sink انجام نشده باشد. برای جلوگیری از حذف داده و یا دوباره نوشتن، گره‌ های حسگر می‌توانند با یکدیگر توسط داده‌ های به اشتراک گذاشته شده همکاری کنند.

خرابی گره یک مسئله حیاتی در WSN است. عدم فعالیت تناوبی (به ‌عنوان مثال، صرفه‌جویی در تولید انرژی)، تخریب فیزیکی، و اشکالات (نرم‌افزار) به احتمال زیاد در شبکه‌ های حسگر بی‌سیم رخ می‌دهد، که منجر به از دست دادن داده‌ها می‌شود. بنابراین، افزونگی با استفاده از تکرار داده‌ها (مثلا، با ذخیره‌سازی نسخه مشابه از داده‌ها بر روی گره‌ های مختلف) به افزایش انعطاف‌پذیری WSN کمک می‌کند.

در این مطالعه، کار ارائه شده در [4] را گسترش می‌دهیم، که در آن تنها ذخیره‌سازی توزیع شده (و بدون بازیابی داده‌ها) در نظر گرفته شده است. توزیع ذخیره‌سازی و بازیابی اطلاعات مکانیسم برای افزایش انعطاف‌پذیری و ظرفیت ذخیره‌سازی یک WSN بر اساس RPLدر برابر کمبود حافظه محلی را پیشنهاد می‌کنیم. RPL، یک استاندارد IETF برای مسیریابی در شبکه‌ های حسگر بی‌سیم از IPv6، یک پروتکل مسیریابی بردار فاصله است که یک گراف مقصد بدون دور (DODAG) را که ریشه در یک sink قرار دارد ایجاد می‌کند. ما رویکرد را در سیستم عامل Contiki و با استفاده از شبیه‌ساز Cooja.ارزیابی می‌کنیم ]6[. به‌طورخاص، ما نشان می‌دهیم که چگونه RPL می‌تواند برای ذخیره‌سازی داده‌ها و بازیابی توزیع شده با صرفه‌جویی در انرژی استفاده شود.

ساختار مقاله به شرح زیر است. بخش دوم به کار های گذشته اختصاص یافته است. یک نمای کلی از RPL در بخش سوم ارائه شده است. بخش چهارم به طراحی توزیع داده‌ها و مکانیزم بازیابی اختصاص داده شده است. بخش پنجم نتایج عملکرد را ارائه می‌دهد. در ن هایت، بخش ششم نتیجه‌گیری مقاله است.

**2. کار های گذشته**

طرح‌ های مختلف به‌شکلی موثر در ذخیره و پردازش داده‌ های حسگر در WSN در سال گذشته ارائه شده است [7]. در رویکرد ذخیره‌سازی توزیع شده‌ی داده‌ها، تمام گره‌ها در سنجش و ذخیره‌سازی به‌ همان شیوه شرکت می‌کنند. تمام گره‌ها، برای اولین‌بار، قرائت سنسور خود به صورت محلی نگهداری می‌کنند و یک بار که حافظه‌ی محلی آنها پر شد، آنها ذخیره‌سازی را به گره‌ های در‌دسترس دیگر واگذار می‌کنند. برای اولین‌بار سهم قابل توجهی در این جهت با مزارع داده بدست آمده است [8]. نویسندگان یک مکانیزم ذخیره‌سازی داده‌ی توزیع شده با بازیابی دور های داده‌ها پیشنهاد می‌کنند. آنها یک مدل هزینه برای اندازه‌گیری مصرف انرژی ایجاد کرده‌اند که نشان می‌دهد که چگونه یک انتخاب دقیق از گره ذخیره‌سازی، به نام اهداکننده گره‌ها، ظرفیت سیستم را در هزینه‌ های انتقال بالا بهینه می‌کند. آنها فرض می‌کنند که شبکه دارای یک درخت توپولوژی است و هر گره حسگر مسیر بازگشت به گره را جهت بازیابی دور های اطلاعات می‌داند.

تکرار داده‌ها شامل افزونگی به سیستم با کپی کردن داده‌ها در چندین گره دهنده (در WSN) برای کاهش خطر ابتلا به نارسایی گره است. یک تابع برای انتخاب مناسب گره تکرارکننده در [9] ارائه شده است. تابع توسط پارامتر های مهمی مانند تعداد کپی مورد نظر، انرژی باقی‌مانده از دو گره تکرارکننده و انرژی از همسایه دو گره تکرارکننده است. نویسنده در [10] PROFLEX، را برای پروتکل ذخیره‌سازی داده‌ های توزیع شده برای تکرار اندازه‌گیری داده‌ها از گره محدود به گره‌ های قوی‌تر پیشنهاد می‌کند.

بازیابی داده‌ها شامل حمل‌ و نقل داده‌ های حسگر جمع‌آوری شده از WSN به یک پایگاه مرکزی برای پردازش بیشتر است. پروتکل درخت مجموعه (CTP) احتمالا مکانیزم مسیریابی استفاده شده برای چند هاپ ثابت بازیابی داده‌ها در شبکه‌ های حسگر بی‌سیم است [11]. نقاط قوت CTP (1) توانایی برای کشف سریع و تعمیر تناقضات مسیر است (2) تطبیق ​​آن، باعث کاهش سربار پروتکل با استفاده از چرخه وظیفه رادیویی است. Dozer [12] یک پروتکل بازیابی داده‎ها با هدف دستیابی به انرژی مصرفی بسیار پایین است. که یک ساختار درختی برای انتقال داده‌ها به sink می‌سازد، که با طرح دسترسی چندگانه تقسیم زمانی (TDMA) در لایه MAC برای همزمان‌سازی گره غنی شده است.

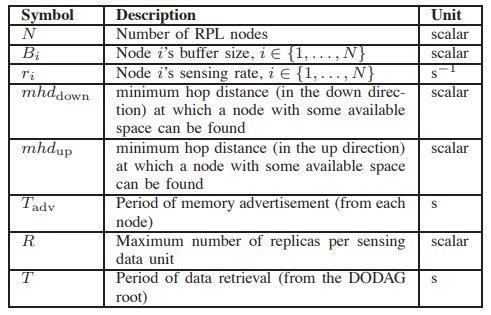
با توجه به آثار مرتبط، مطالعه ما فراتر می‌رود. ابتدا، یک مکانیزم کاملا توزیع شده، برای هر دو تکرار داده و ذخیره‌سازی توزیع شده دایر می‌کنیم. دوم، نشان می‌دهیم که چگونه RPL می‌تواند طراحی انعطاف‌پذیر قرار دادن داده را به عنوان یک طرح کارآمد بازیابی داده‎‌ها بیان کند. بنا به دانش ما، این اولین کاری است که به ذخیره‌سازی داده‌ها و مکانیزم بازیابی توسط RPL پرداخته است.

**3. نمای کلی RPL**

RPL ]5[ به‌تازگی به‌عنوان استانداردی برای مسیریابی در IPv6 در شبکه گیرنده بی‌سیم ظهور کرده است. که بر اساس DODAG در یک یا چند گره (ریشه DAG) است. هر گره رتبه‌اش را در درخت RPL محاسبه می‌کند.که عمق گره را در DODAG توصیف می‌کن. برای ساخت و حفظ توپولوژی، گره‌ های RPL به‌صورت دور های پیام‌ های اطلاعات شی DODAG (DIO) را به‌منظور انتشار اطلاعات مسیریابی رو به پایین در درخت تبادل می‌کنند. این نوع از ساختار برای ترافیک چند نقطه-به-نقطه مفید است، که در آن ریشه DAG مقصد از همه بسته‌ های داده است.

در حمایت از ترافیک چند نقطه به نقطه و نقطه به نقطه ، RPL پیام‌ های کنترلی اضافی را تعریف می‌کند، که با عنوان پیام مقصد شیء (DAO) تعریف می‌شود و به‌عنوان جمعیت جداول مسیریابی گره والدین در (DAG) به‌منظور روت بسته‌ های اطلاعاتی در جهت پایین استفاده می‌شود، (به‌عنوان مثال، گره‌ها با رتبه پایین‌تر). مسیرها با توجه به تابع هدف (OF) و با توجه به مجموع های از معیارها و محدودیت‌ های مورد علاقه محاسبه می‌شوند.

**4. ذخیره‌سازی داده‌ های تکراری و بازیابی**

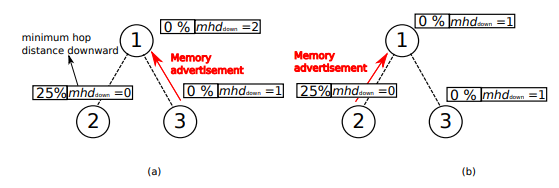


جدول 1: پارامتر های اصلی سیستم

در سناریوی ما، گره‌ های WSN، پس از پیوستن به یک RPL DODAG ، داده‌ های جمع‌آوری شده را نگه می‌دارند (با توجه به نرخ سنجش به‌دست می‌آیند). به منظور جلوگیری از دست دادن داده‌ها، داده‌ها در چند گره تکرار شده است (احتمالا گره مولد). که شامل کپی کردن و کپی توزیع ‌شده از داده‌ های مشابه به گره‌ های دیگر از فضای حافظه در دسترس است. اطلاعات در مورد در دسترس بودن حافظه، به‌ صورت دور های توسط هر یک از گره‌‎ها، به همه گره‌ های مجاور به ‌صورت مستقیم پخش می‌شود.

بازیابی داده‌ها توسط یک عامل خارجی که به صورت دور های به ریشه DAG متصل می‌شود و جمع‌آوری تمام داده‌ها از WSN اجرا می‌شود. پارامتر های اصلی در جدول 1 بدون از دست دادن کلیت ذکر شده است، از نظر ما یک WSN با N گره ثابت RPL و گره‌ های اضافی است، که به‌عنوان ریشه DAG عمل می‌کنند اما در سنجش و ذخیره‌سازی شرکت نمی‌کنند. بنابراین، تعداد کل گره‌ها در شبکه گیرنده بی‌سیم + N 1 است. گره i ام دارای یک بافر محلی محدود از اندازه Bi (ابعاد: [واحد اطلاعات]) و نرخ سنجش ri (ابعاد: [واحد داده‌ها بر ثانیه[ است.

هر گره، بدون دانش و هر Tadv (ابعاد: [S])، وضعیت حافظه خود را به تمام گره‌ها در محدوده‌ی انتقال مستقیم پخش می‌کند. هر تبلیغات حافظه متشکل از 6 زمینه نسبت به ارسال گره است: (1) رتبه RPL گره؛ (2) ارزش نرخ سنجش ؛ (3) به روز رسانی فضای حافظه در دسترس (4) دانه‌ها که نشان دهنده‌ی وضعیت گره‌ها در جهت پایین در DODAG است. (5) یک مقدار مشابه برای جهت بالا ؛ و (6) یک شماره توالی. هر گره یک جدول سوابق وضعیت حافظه‌ی دریافت شده از گره همسایه را حفظ می‌کند. به‌محض دریافت یک تبلیغ حافظه از یک همسایه، یک گره، جدول حافظه خود را با استفاده از شماره توالی برای به دور انداختن دریافت متعدد یا تبلیغات تاریخ گذشته به‌روز می‌کند. مجموع وضعیت گره در جهت پایین در DODAG توسط حداقل فاصله هاپ (پارامتر mhddown) داده شده است که در آن یک گره با برخی از فضا های در دسترس می‌توان یافت. این فاصله به شرح زیر محاسبه می‌شود: اگر یک گره تشخیص دهد که حداقل یکی از فرزندانش (به‌عنوان مثال، همسایگان با رتبه RPL بالاتر) فضایی در سطح محلی دارد، فاصله آن را مقدار 1 قرار می‌دهد. درغیراین‌صورت، والدین 1 واحد از ارزش حداقل فاصله‌ی داده شده توسط فرزاندانشان افزایش می‌یابند. زمانی که فاصله به مقدار بیشینه می‌رسد، یک گره فرض می‌کند که هیچ حافظه ممکنی در جهت پایین DODAG وجود ندارد. به‌طور مشابه، وضعیت گره‌ها در جهت بالا به همان روش اما در جهت معکوس DODAG محاسبه می‌شود، در این مورد، توسط حداقل فاصله پارامتر هاپ به سمت بالا (mhdup) داده می‌شود.



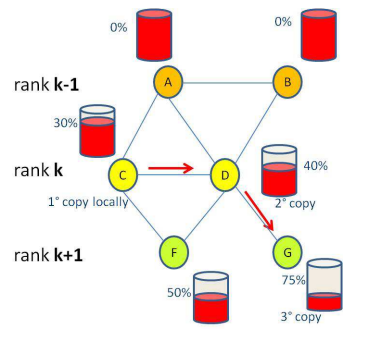
شکل 1: پیام برای تبلیغات حافظه مبادله می‌شود. (الف) حافظه‌ی در دسترس توسط گره 3 در 1 هاپ در جهت پایین آن اعلام می‌گردد. (ب) گره 1 mhddown خود را به 1 به عنوان یک گره نزدیک به حافظه موجود به‌روز می‌کند.

یک سناریوی گویا در شکل 1 نشان داده شده است. در شکل 1 (a)، گره 3 تبلیغات حافظه را به گره 1 انتقال می‌دهد، گفتنی است که هیچ فضایی در دسترس نیست، اما یکی از فرزندان 1-هاپ آن با فضای موجود وجود دارد و به عنوان mhddown اعلام می‌گردد.

گره 1، که دارای هیچ حافظه‌ی در دسترسی نیست، پس mhddown را مقدار 2 می‌دهد، به‌دلیل این‌که اطلاعات از نزدیک‌ترین گره با دریافت حافظه در دسترس در فاصله 2 هاپ رسیده است. پس از آن، همانطور که در شکل 1(ب) نشان داده شده است، گره اعلام می‌کند که فضای کافی در دسترس دارد. در نتیجه، گره 1 آگاه می‌شود که یک گره نزدیک‌تر با سطح دسترسی بالا از فضا پس از به‌روزرسانی mhddown آن به 1وجود دارد.

مکانیسم ذخیره‌سازی توزیع شده‌ی ارائه شده‌، به‌طورکامل غیر متمرکز است، به این معنا که تمام گره‌ها همان نقش را بازی می‌کنند. آن در ایجاد نسخه R در هر واحد داده‌ی تولید شده توسط یک گره و توزیع آنها در سراسر شبکه، ذخیره‌سازی حداکثر یک کپی در هر گره سهم اساسی دارد. کپی‌ها باید تا حد ممکن نزدیک به ریشه DAG به منظور کاهش انرژی مصرفی فاز بازیابی ذخیره شوند. هر نسخه به عنوان replica اشاره شده است.

گره مشخص شده است. در زمان t، گره برحسب سنجش یک واحد داده تولید شده است. جدول حافظه گره i شامل یک ورودی برای هر همسایه مستقیم است. گره i از حافظه جدول آن گره‌ های همسایه را، با نام اهدا کننده، با بزرگترین فضای حافظه در دسترس و جدیدترین اطلاعات انتخاب می‌کند. علاوه بر این، اولویت به آن اهداکنندگانی که والد گره i در درخت هستند داده می‌شود، به‌عنوان مثال، گره با رتبه پایین تر. اگر هیچ والدی نتواند انتخاب شود، گره i به دنبال فرزندی برای خود در درخت می‌گردد، به‌عنوان مثال، گره با رتبه RPL بالاتر، ارائه چنین گر های نیاز به فضای در دسترس دارد. اگر تمام همسایگان هیچ فضایی در سطح محلی نداشته باشند، گره i ، یک همسایه که حداقل همسای های (در 2 هاپ از گره i) با فضای در دسترس دارد در جهت بالا و/یا در جهت پایین‌تر از DODAG بررسی می‌کند. در این مورد، دوباره، اولویت به گره در جهت بالا داده می‌شود. اگر هیچ همسایه‌ی مناسبی در جدول حافظه وجود نداشته باشد، امکان توزیع کپی از واحد داده در سراسر شبکه وجود ندارد. در این مورد، تنها یک نسخه را می‌توان درحافظه محلی گره i به شرط داشتن فضای محلی ذخیره کرد.



شکل 2: تکرار هاپ توسط هاپ در مورد3R = به کپی نیاز دارد. داده‌ها در جهت پایین از RPL DODAG به‌عنوان خاطرات والد بین گره‌ها توزیع می‌شود.

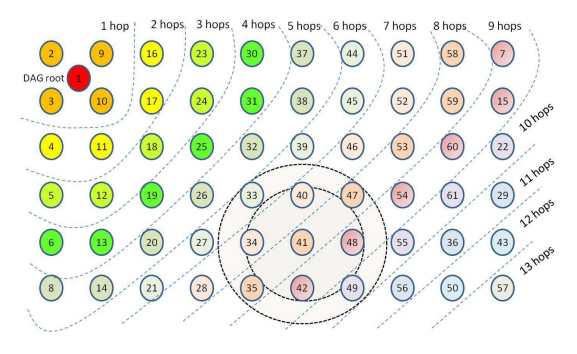
اگر یک گره اهداکننده می‌تواند انتخاب شود، گره i به آن یک کپی از واحد داده می‌فرستد و مشخص می‌کند که چه تعدادی از نسخه‌ های دیگر هنوز هم در WSN توزیع شده هستند. با توجه به اصل مشخص فوق، تعدادی از کپی‌ های مورد نیاز در 1R- (در صورتی گره i بتواند داده‌ های اصلی را به‌صورت محلی نگه‌داری کند) و یا R (اگر حافظه محلی گره i پر باشد) ذخیره شوند. پس از دریافت کپی، گره اهداکننده‌ کپی را در حافظه خود ذخیره می‌کند، اگر فضای محلی داشته باشد، و گره‌ های اهداکننده‌ی بعدی را در میان همسایگان خود، دور از ارسال گره و گره منبع از گره نامزد انتخاب می‌کند. اهداکننده‌ی بعدی با همین قانون انتخاب می‌شود، اولویت‌بندی گره‌ها به ریشه DAG نزدیک‌تر است.

این اجازه می‌دهد تا کپی به‌خوبی در سراسر DODAGگسترش یابد. در این مرحله، گره اهدا کننده کپی به گره‌ های اهداکننده‌ی انتخاب شده‌ی بعدی با کاهش تعداد کپی مورد نیاز توسط 1 ارسال می‌کند. فرآیند تکرار به‌صورت بازگشتی تا زمانی‌که آخرین (R امین) کپی ذخیره شود و یا متوقف شود ادامه می‌یابد وقتی که یکی از گره‌ های اهداکننده می‌تواند گره اهداکننده‌ی مناسب دیگر را پیدا کند. در مورد دوم، تعداد ن هایی کپی‌ های ذخیره شده در WSN کمتر از R است. توجه داشته باشید که، اگر یک اهداکننده نمی‌تواند یک همسایه با رتبه RPL پایین‌تر (به‌عنوان مثال، نزدیک‌تر به ریشه‌ی DAG) پیدا کند، replica ممکن است یک مسیر برعکس و متفاوت را در امتداد DODAG، و برای پس گرفتن جهت اصلی بعدی دنبال کند.

در شکل 2، یک مثال با 3R= کپی مورد نظر نشان داده شده است. گره همیشه سعی می‌کند کپی را به والد در جهت بالای DODAG، به‌عنوان مثال، گره‌ها با یک رتبه پایین‌تر، توزیع کند. برای اشباع شبکه، داده‌ها به سمت گره با رتبه بالاتر، به‌عنوان مثال، در جهت پایین‌تر از DODAG، توزیع می‌شوند.

طبق بازیابی داده‌ها، درخواست بازیابی دور های توسط ریشه DAG فرستاده می‌شود و تبلیغ به تمام گره‌ های DODAG انتقال می‌شود. بازیابی داده‌ها پس از دریافت درخواست از هر گره به سمت ریشه DAG، به دنبال مسیر پیش فرض RPL به سمت ریشه اتفاق می‌افتد. چنین الگوی ترافیک many-to-many، اگر به دقت به کار گرفته‌شود، می‎‌تواند باعث (1) بسیاری از برخوردها شود (2) مصرف بالای انرژی نامتعادل و ناکارآمد در کل شبکه. برای کاهش این خطرات، replica اطلاعات داده شده، فقط اگر در میان تمام کپی‌ های ذخیره شده در مورد داده‌ها نزدیک به ریشه DAG باشد، فرستاده می‌شود.

**5. نتایج عملکرد ارائه شده**



شکل 3: سناریو در Cooja مورد بررسی قرار گرفت. 60 گره ذخیره‌سازی در یک شبکه به طور منظم مستقر شده‌اند. هر گره دارای 4 همسایه مستقیم است. گره 1 ریشه DAG است. شبکه دارای یک اندازه کل برابر با 13 هاپ است.

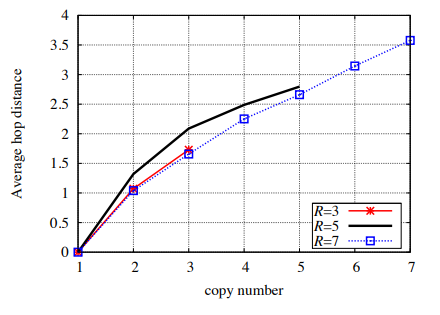
ذخیره‌سازی توزیع‌شده‌ی پیشنهادی و مکانیزم بازیابی در 2.5 Contiki و در Cooja، یک شبیه‌ساز شبکه گیرنده بی‌سیم مبتنی بر جاوا [6]، پیشنهاد شده است. سناریوی شبیه‌سازی شده، در شکل 3نشان داده شده است، که از 60 N = گره ذخیره‌سازی، که در یک شبکه مستطیل شکل و یک گره اضافی که به عنوان ریشه DAGعمل می‌کند، تشکیل شده است. هر گره ذخیره‌سازی در داخل شبکه با همسایگان خود در 4 جهت مستقیم ارتباط برقرار می‌کند. علاوه‌براین، برای شبیه‌سازی شرایط واقعی، گره با برخی از گره‌ های اضافی تداخل می‌کند: یک برخورد رخ می‌دهد اگر یک گره و حداقل یکی از همسایگانش و یا تداخل آن گره، یک بسته در زمان یکسان انتقال دهد. به‌عنوان مثال، گره 41 دارای 4 همسایه است: به ترتیب گره 34، 40، 48 و 42. گره مداخله، بین دو دایره نشان داده شده است، گره‌ های: 33، 35، 47 و 49.

توجه داشته باشید که گره‌‎ های در امتداد مرزها، همسایه‌ی کمتری از آن هایی که در داخل هستند دارند.

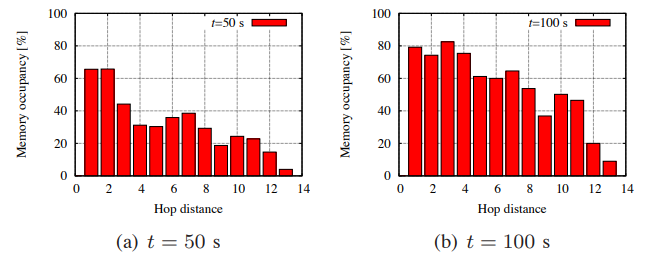
فاصله بین دو بازیابی داده‌ی متوالی 10T = دقیقه است. دوره سنجش گره‌ها یک عدد صحیح است که به‌طور تصادفی و به‌طور مستقل در محدوده‌ی [1،9] انتخاب شده است. همه گره‌ها اندازه بافر یکسانی برابر با 100 B = واحد داده دارند. دوره تبلیغات حافظه Tadv مقدار 30 است. ما تعداد مورد انتظار انتقال (ETX) را به‌عنوان متریک RPL انتخاب می‌کنیم. به‌طورخاص، متریک ETX تعداد انتقال مورد انتظار برای رسیدن به ریشه DAG را به حداقل می‌رساند.

**تاثیر تعداد کپی:** چهار مقدار برای R کپی در نظر گرفته شده است: 1، 3، 5، و 7. توجه داشته باشید که مورد 1 R = بدون تکرار است، به‌عنوان مثال، تنها نسخه اصلی ذخیره شده است.

با توجه به توزیع فضایی کپی، شکل 4 فاصله هاپ متوسط ​​رسیده توسط کپی‌ های اضافی از صاحب اصلی آن، برای مقادیر مختلف R را نشان می‌دهد. نتایج ما نشان می‌دهد که کپی‌ های متوالی تمایل به حرکت دور از گره منبع در امتداد DODAG را دارند. در حالی‌که مکانیسم پیشنهادی در [4] از نظر گسترش ذخیره‌سازی ناکارآمد بود (در واقع، فاصله هاپ در حالت متوسط ​​حدودا 2 بود)، اما مکانیسم فعلی، به لطف RPL، می‌تواند منجر به حفظ اطلاعات انعطاف‌پذیر در حضور یک شکست در یک منطقه شامل چند گره همسایه شود.



شکل 4: میانگین فاصله هاپ رسیده توسط k امین replica، به‌عنوان یک تابع از k است. مقدار k بین 1 و 3 (3R=) و 1 و 5 (5R=) و 1 و 7 (7R =) است. فاصله از موقعیت اولیه‌ی replica محاسبه می‌شود.



شکل 5: درصد اشغال حافظه‌ی مختلف فاصله هاپ از ریشه DAG، در فواصل زمانی مختلف. خاطرات گره‌ های نزدیک به ریشه DAG سریعتر اشباع شده‌اند. تعداد کپی R مقدار 7 است.

ارزیابی قرار دادن داده‌ها در سراسر WSN در طول زمان از موضوعات جالب است. همان‌گونه که قبلا مورد بحث قرار گرفت، مکانیسم توزیع اولویت ‌بندی گره اهدا‌کننده‌ی نزدیک به ریشه DAG را کپی می‌کند.

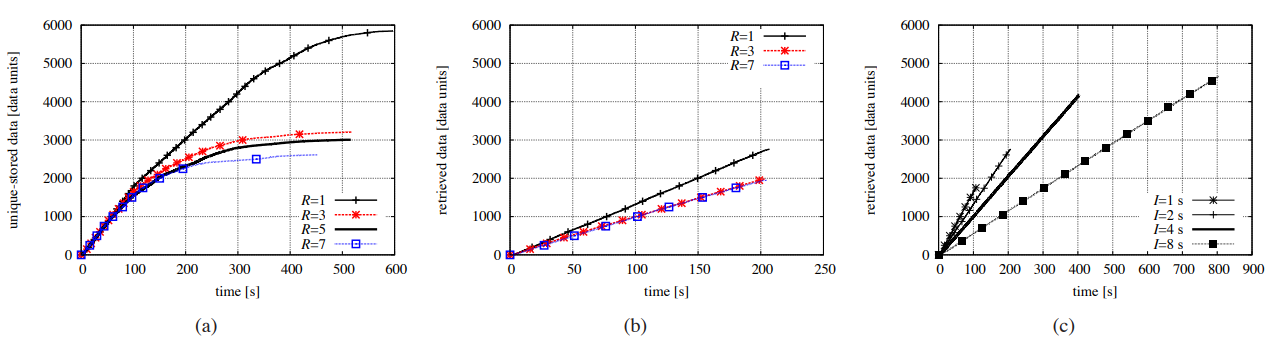
شکل. 5 سطح اشباع متوسط خاطرات گره در فواصل مختلف هاپ از ریشه DAG، برای چند لحظه مشاهده را نشان می‌دهد. می‌توان آن را متوجه شد که بخشی از DODAG نزدیک به ریشه خاطرات را سریع‌تر پر می‌کند. این می‌تواند در مرحله بازیابی داده‌ها، به‌عنوان نیاز به داده‌ها به دنبال یک مسیر کوتاه‌تر برای رسیدن به ریشه DAG باشد.

در شکل (6)، مقدار داده‌ های منحصر به فرد ذخیره شده (به‌عنوان مثال، کپی حساب نمی‌شود) به‌عنوان تابعی از زمان نشان داده شده است. بدون افزونگی (1R =)، مقدار منحصر به فرد داده‌ های ذخیره شده به ظرفیت ذخیره‌سازی شبکه 6000 C = N × B = واحد داده می‌رسد. در حضور افزونگی (1R>)، چند کپی از هر واحد داده در WSN توزیع شده است، بنابراین، مقدار کمتری از داده‌ های منحصر به فرد ذخیره شده است.

همان‌طور که برای بازیابی داده‌ها، در مورد 1R> ، داده‌ی کمتری به ریشه DAG تحویل داده شده است، بنابراین تنها نزدیک‌ترین replica ارسال می‌شود. از سوی دیگر، با 1R = ، تمام 6000 = N × B واحد داده به sink تحویل داده می‌شود. این باعث افزایش مقدار ترافیک many-to-one می‌شود و در نتیجه، خطر برخورد وجود دارد. برای کاهش اثرات تصادم، واحد داده‌ها از هر گره به سمت ریشه DAG، یک‌در‌یک در زمان منتقل می‌شوند.. ‌فاصله‌ی بین دو انتقال متوالی به عنوان I نشان داده شده است (ابعاد: [S]). به ‌طور مستقیم، فاصله کوتاه‌تر ممکن است فرایند بازیابی را سرعت بخشد، اما ممکن است احتمال برخورد را افزایش دهد و در نتیجه، درصدی از داده‌ها در ریشه‌ی DAG از دست برود. از سوی دیگر، ارزش I ممکن است مقدار داده‌ های بازیابی در sink را افزایش دهد و تاخیر فاز بازیابی را جریمه کند.

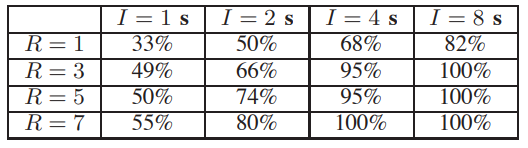
در شکل 6 (ب)، مقدار داده‌ های بازیابی در ریشه DAG به‌عنوان یک تابع از زمان، برای مقادیر مختلف R نشان داده شده است. ورودی I مقدار 2 S می‌گیرد. همان‌گونه که انتظار می‌رود، میزان داده‌ های بازیابی با R کاهش می‌یابد، زیرا داده‌ های منحصربه فرد بیشتری وجود دارد که در سیستم زمانی بدون افزونگی مورد نیاز است. اگر ما بتوانیم آن را با 1R = مشاهده کنیم، تنها بخش کوچکی، به‌عنوان مثال، حدود 50 درصد از داده‌ های ذخیره شده C = N × B، با موفقیت بازیابی می‌شود؛ بقیه به‌دلیل برخورد از دست می‌رود. برای مقادیر بالاتر R، درصد مربوطه از داده‌ های بازیابی به‌طور قابل توجهی افزایش می‌یابد. به‌عنوان مثال، در مورد 7R=، حدود 2000 واحد داده بازیابی می‌شود، که مربوط به حدود 80 درصد از مقدار کل داده‌ های منحصربه فرد است.

برخورد نه تنها توسط R، بلکه توسط I نیز تحت تاثیر قرار می‌گیرد. شکل. 6 (ج) نشان می‌دهد که، با هیچ تکراری (1R =)، مقدار داده‌ های بازیابی برای مقادیر بالاتر از I افزایش قابل توجهی نمی‌یابد.



شکل 6: داده‌ های بازیابی و ذخیره شده با توجه به اندازه‌ی حافظه برابر با 100 واحد داده، گره‌ های ذخیره‌سازی 60 N = ، و یک دوره تبلیغات حافظه 30 ثانی های؛ (الف) اطلاعات منحصربه فرد ذخیره شده است. (ب) داده‌ های بازیابی شده برای چندین مقدار R با 2i=؛ (ج) داده‌ های بازیابی شده برای چندین مقدار I، با 1R =.

به‌طورخلاصه، درصد داده‌ های بازیابی در میان مجموع داده‌ های ذخیره شده در سیستم‌ های منحصر به فرد است در جدول 2 برای ترکیبات مختلف R و I نشان داده شده است.



جدول 2: درصد بازیابی داده‌ها.

**6. نتیجه‌گیری**

این مقاله با مشکل داده‌ های تکراری توزیع شده و بازیابی برای نظارت مبتنی بر WSNمواجه است. مکانیزم ذخیره‌سازی داده‌ های توزیع شده در بالای RPL، به‌منظور افزایش انعطاف‌پذیری رویکرد پیشنهادی و ظرفیت ذخیره‌سازی WSN در برابر خرابی گره و کمبود حافظه محلی ساخته شده است. عملکرد از طریق شبیه‌سازی مورد ارزیابی قرار گرفت. مکانیسم مستقیما به‌اجرای یک طرح بازیابی داده‌ها منجر شد، تا عملکرد آن به‌خوبی انجام شود. نتایج ما به وضوح یک افزونگی بین ذخیره‌سازی (که باعث آلوده شدن کل حافظه ذخیره‌سازی در دسترس شده است) و بازیابی بهره‌وری (از نظر درصد داده‌ های بازیابی) را نشان می‌دهد.

فعالیت‌ های تحقیقاتی آینده شامل استفاده از سایر پارامترها در استراتژی تکرار، مانند مصرف انرژی گره‌ها یا قابلیت دسترسی گره است، به‌ویژه اگر آنها در لایه MAC با وظیفه چرخه (به‌عنوان مثال، ContikiMAC و یا X-MAC) عمل کنند. ما همچنین مطالعه‌ی پیکربندی دوباره‌ی پویا از رفتار گره (به‌عنوان مثال، نمونه‌برداری) و لایه‌ های ارتباطی (به‌عنوان مثال، قدرت انتقال) را برای مواجهه با تکرار خواسته‌ها با حداقل هزینه انرژی را پیگیری خواهیم کرد. در ن هایت، ما می‌خواهم عملکرد مکانیسم در دنیای واقعی را ارزیابی کنیم.

**سپاس‌گزاری**

این کار توسط بودجه چارچوب برنامه هفتم جامعه اروپا، منطقه " Internetconnected Objects"، تحت اعطای شماره‌ی 288879، پروژه CALIPSO با اتصال تمام اشیاء هوشمند انجام شده است. کار تنها نشان‌دهنده‌ی دیدگاه نویسندگان است؛ جامعه اروپا در قبال استفاده از اطلاعات موجود که در اینجا ساخته شده است مسئول نیست.

**REFERENCES**

[1] G. Mathur, P. Desnoyers, D. Ganesan, and P. Shenoy, “Ultra-low power data storage for sensor networks,” in The Fifth International Conference on Information Processing in Sensor Networks (IPSN’06), Nashville, TN, USA, 2006, pp. 374–381.

[2] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. S. Peh, and D. Rubenstein, “Energy-efficient computing for wildlife tracking: design tradeoffs and early experiences with zebranet,” in Proceedings of the 10th international conference on Architectural support for programming languages and operating systems (ASPLOS-X), San Jose, CA, USA, October 2002, pp. 96–107.

[3] A. Beaufour, M. Leopold, and P. Bonnet, “Smart-tag based data dissemination,” in Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications (WSNA’02), Atlanta, GA, USA, September 2002, pp. 68–77.

[4] P. Gonizzi, G. Ferrari, V. Gay, and J. Leguay, “Redundant distributed data storage: experimentation with the SensLab testbed,” in Proc. Int. Conf. Sensor Networks (SENSORNETS), Rome, Italy, Feb. 2012.

[5] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, “RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks,” RFC 6550 (Proposed Standard), Mar. 2012. [Online]. Available: <http://www.ietf.org/rfc/rfc6550.txt>

[6] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, and T. Voigt, “Crosslevel sensor network simulation with cooja,” in Proceedings of 31st IEEE Conference on Local Computer Networks, nov. 2006, pp. 641 –648.

[7] F. Hongping and F. Kangling, “Overview of data dissemination strategy in wireless sensor networks,” in International Conference on E-Health Networking, Digital Ecosystems and Technologies (EDT), Shenzhen, China, April 2010, pp. 260–263.

[8] A. Omotayo, M. Hammad, and K. Barker, “A cost model for storing and retrieving data in wireless sensor networks,” in IEEE 23rd International Conference on Data Engineering Workshop (ICDE), Istanbul, Turkey, April 2007, pp. 29–38.

[9] J. Neumann, C. Reinke, N. Hoeller, and V. Linnemann, “Adaptive quality-aware replication in wireless sensor networks,” in International Workshop on Wireless Ad Hoc, Mesh and Sensor Networks (WAMSNET’09), Jeju Island, Korea, December 2009, pp. 413–420.

[10] G. Maia, D. L. Guidoni, A. Carneiro Viana, A. L. L. Aquino, R. A. F. Mini, and A. A. F. Loureiro, “Proflex: A probabilistic and flexible data storage protocol for heterogeneous wireless sensor networks,” INRIA, Research Report, July 2011.

[11] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, “Collection tree protocol,” in Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems (SenSys), Berkeley, CA, USA, november 2009, pp. 1–14.

[12] N. Burri, P. von Rickenbach, and R. Wattenhofer, “Dozer: Ultra-low power data gathering in sensor networks,” in 6th International Symposium on Information Processing in Sensor Networks (IPSN), Cambridge, MA, USA, april 2007.