

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 بی سیم (WSN ها) خواستار تکنیک های ذخیره سازی داده انعطاف پذیر در شبکه ها برای به حداقل رساندن خطر از دست دادن داده ها هستند. بهره‌وری این تکنیک‌ها می‌تواند تا حد زیادی بهره‌برداری از اطلاعات شبکه را بهبود دهد، مانند مسیریابی که توسط پروتکل استفاده می‌شود. به طور خاص، 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 شده ی داده ها و مکانیسم بازیابی برای افزایش انعطاف پذیری 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 شده ی داده‌ها؛ RPL؛ تکرار داده‌ها؛ بازیابی داده‌ها؛ Contiki و Cooja.

1. مقدمه

در مقابل شبکه معمولی ذخیره سازی داده‌ها، ذخیره سازی داده ها در شبکه های حسگر بی سیم (WSN) یک چالش به دلیل محدودیت قدرت، حافظه و ارتباطات پهنای باند شبکه گیرنده بی سیم را نشان می‌دهد. به تازگی، سنسورها به

قابلیت های بالاتری از نظر سرعت پردازش و ذخیره سازی محلی، نسبت به سال گذشته رسیده اند [1]، که آنها را برای ذخیره سازی در شبکه جذاب تر می کند.

به طور معمول WSN ها از مجموع های از گره های بدون مراقبت برای پوشش محیط زیست اطراف آن تشکیل شده است و گره sink مسئول جمع آوری داده های اندازه گیری و بازپخش کردن به یک نهاد مدیریتی است. دلایل متعددی وجود دارد که ممکن است یک گره حسگر از انتقال داده ها به گره sink بعد از به دست آوردن آن جلوگیری کند. به عنوان مثال، ممکن است گره های حسگر همیشه قادر به رسیدن گره با توجه به قطع شدن موقعیت لینک و یا عملیات چرخه ای وظیفه در گره نباشد. علاوه بر این، وقتی برنامه های کاربردی به مجموعه زمان واقعی نیاز ندارند، ذخیره سازی واحد داده ها و ارسال اطلاعات جمع آوری شده می تواند به کمک میزان انتقال رادیویی، که موجب افزایش طول عمر عملیات WSN می شود 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 جمع آوری و ذخیره ای داده ها تا جایی که ممکن است بین دو بازیابی متوالی داده توسط یک عامل خارجی انجام می شود. با این حال، ذخیره سازی داده در گره حسگر منجر به سرریز حافظه محلی می شود در صورتی که بازیابی داده ها به موقع توسط sink انجام نشده باشد. برای جلوگیری از حذف داده و یا دوباره نوشتن، گره های حسگر می توانند 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، صرفه جویی در تولید انرژی)، تخریب فیزیکی، و اشکالات (نرم افزار) به احتمال زیاد در شبکه های حسگر بی سیم رخ می دهد، که منجر به از دست دادن داده ها می شود. بنابراین، افزونگی با استفاده از تکرار داده ها (مثلا، با ذخیره سازی نسخه مشابه از داده ها بر روی گره های مختلف) به افزایش انعطاف پذیری WSN کمک می کند.

در این مطالعه، کار ارائه شده در [4] را گسترش می دهیم، که در آن تنها ذخیره سازی توزیع شده (و بدون بازیابی داده ها) در نظر گرفته شده است. توزیع ذخیره سازی و بازیابی اطلاعات مکانیسم برای افزایش انعطاف پذیری و ظرفیت ذخیره سازی یک WSN بر اساس RPL در برابر کمبود حافظه محلی را پیشنهاد می کنیم. RPL، یک استاندارد IETF

برای مسیریابی در شبکه های حسگر بی سیم از IPv6، یک پروتکل مسیریابی بردار فاصله است که یک گراف مقصد بدون دور (DODAG) را که ریشه در یک sink قرار دارد ایجاد می کند. ما رویکرد را در سیستم عامل Contiki و با استفاده از شبیه ساز Cooja ارزیابی می کنیم [6]. به طور خاص، ما نشان می دهیم که چگونه RPL می تواند برای ذخیره سازی داده ها و بازیابی توزیع شده با صرفه جویی در انرژی استفاده شود.

ساختار مقاله به شرح زیر است. بخش دوم به کارهای گذشته اختصاص یافته است. یک نمای کلی از RPL در بخش سوم ارائه شده است. بخش چهارم به طراحی توزیع داده ها و مکانیزم بازیابی اختصاص داده شده است. بخش پنجم نتایج عملکرد را ارائه می دهد. در نهایت، بخش ششم نتیجه گیری مقاله است.

2. کارهای گذشته

طرح های مختلف به شکلی موثر در ذخیره و پردازش داده های حسگر در WSN در سال گذشته ارائه شده است [7]. در رویکرد ذخیره سازی توزیع شده داده ها، تمام گره ها در سنجش و ذخیره سازی به همان شیوه شرکت می کنند. تمام گره ها، برای اولین بار، قرائت سنسور خود به صورت محلی نگهداری می کنند و یک بار که حافظه محلی آنها پر شد، آنها ذخیره سازی را به گره های در دسترس دیگر واگذار می کنند. برای اولین بار سهم قابل توجهی در این جهت با مزارع داده بدست آمده است [8]. نویسندگان یک مکانیزم ذخیره سازی داده ی توزیع شده با بازیابی دور های داده ها پیشنهاد می کنند. آنها یک مدل هزینه برای اندازه گیری مصرف انرژی ایجاد کرده اند که نشان می دهد که چگونه یک انتخاب دقیق از گره ذخیره سازی، به نام اهداکننده گره ها، ظرفیت سیستم را در هزینه های انتقال بالا بهینه می کند. آنها فرض می کنند که شبکه دارای یک درخت توپولوژی است و هر گره حسگر مسیر بازگشت به گره را جهت بازیابی دور های اطلاعات می داند.

تکرار داده ها شامل افزونگی به سیستم با کپی کردن داده ها 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 پارامترهای مهمی مانند تعداد کپی مورد نظر، انرژی باقی مانده از دو گره تکرارکننده و انرژی از همسایه دو گره تکرارکننده است.

نویسنده در [10] PROFLEX، را برای پروتکل ذخیره‌سازی داده‌های توزیع شده برای تکرار اندازه‌گیری داده‌ها از گره محدود به گره‌های قوی‌تر پیشنهاد می‌کند.

بازیابی داده‌ها شامل حمل و نقل داده‌های حسگر جمع‌آوری شده از WSN به یک پایگاه مرکزی برای پردازش بیشتر است. پروتکل درخت مجموعه (CTP) احتمالاً مکانیزم مسیریابی استفاده شده برای چند هاپ ثابت بازیابی داده‌ها در شبکه‌های 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 کاملاً 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 استاندارد برای مسیریابی در IPv6 در شبکه‌گیرنده 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، گره‌های RPL به‌صورت دور‌های پیام‌های 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 بسته‌های داده است.

در حمایت از ترافیک چند نقطه به نقطه و نقطه به نقطه، RPL پیام های کنترلی اضافی را تعریف می کند، که با عنوان پیام مقصد شیء (DAO) تعریف می شود و به عنوان جمعیت جداول مسیریابی گره والدین در (DAG) به منظور روت بسته های اطلاعاتی در جهت پایین استفاده می شود، (به عنوان مثال، گره ها با رتبه پایین تر). مسیرها با توجه به تابع هدف (OF) و با توجه به مجموع های از معیارها و محدودیت های مورد علاقه محاسبه می شوند.

4. ذخیره سازی داده های تکراری و بازیابی

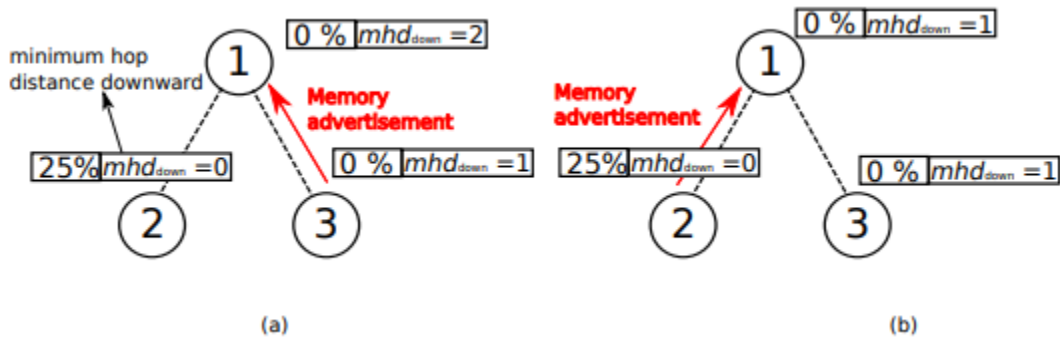
Symbol	Description	Unit
N	Number of RPL nodes	scalar
B_i	Node i 's buffer size, $i \in \{1, \dots, N\}$	scalar
r_i	Node i 's sensing rate, $i \in \{1, \dots, N\}$	s^{-1}
mhd_{down}	minimum hop distance (in the down direction) at which a node with some available space can be found	scalar
mhd_{up}	minimum hop distance (in the up direction) at which a node with some available space can be found	scalar
T_{adv}	Period of memory advertisement (from each node)	s
R	Maximum number of replicas per sensing data unit	scalar
T	Period of data retrieval (from the DODAG root)	s

جدول 1: پارامتر های اصلی سیستم

در سناریوی ما، گره های WSN، پس از پیوستن به یک RPL DODAG، داده های جمع آوری شده را نگه می دارند (با توجه به نرخ سنجش به دست می آیند). به منظور جلوگیری از دست دادن داده ها، داده ها 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 گره ها، به همه گره های مجاور به صورت مستقیم پخش می شود.

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 $1 + N$ است. گره i ام دارای یک بافر محلی محدود از اندازه B_i (ابعاد: [واحد اطلاعات]) و نرخ سنجش r_i (ابعاد: [واحد داده‌ها بر ثانیه]) است.

هر گره، بدون دانش و هر T_{adv} (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 mhd_{down}) 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. در غیر این صورت، والدین 1 واحد از ارزش حداقل فاصله‌ی داده شده توسط فرزندانشان افزایش می‌یابند. زمانی که فاصله به مقدار بیشینه می‌رسد، یک گره فرض می‌کند که هیچ حافظه ممکن 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 (mhd_{up}) 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 mhd_{down} خود را به 1 به عنوان یک گره نزدیک به حافظه موجود به روز می‌کند.

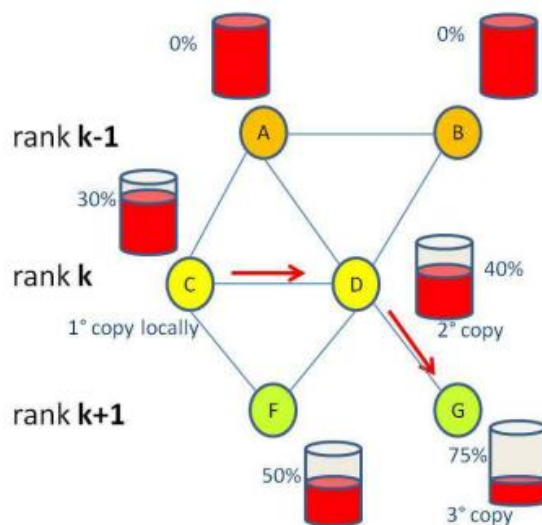
یک سناریوی گویا در شکل 1 نشان داده شده است. در شکل 1 (a)، گره 3 تبلیغات حافظه را به گره 1 انتقال می‌دهد، گفتنی است که هیچ فضایی در دسترس نیست، اما یکی از فرزندان 1-هاپ آن با فضای موجود وجود دارد و به عنوان mhd_{down} اعلام می‌گردد.

گره 1، که دارای هیچ حافظه‌ی در دسترس نیست، پس mhd_{down} را مقدار 2 می‌دهد، به دلیل این که 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 mhd_{down} آن به 1 وجود دارد.

مکانیسم ذخیره‌سازی توزیع شده‌ی ارائه شده، به‌طور کامل 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 بازبازی ذخیره شوند. هر نسخه به عنوان replica اشاره شده است.

گره $i \in \{1,000..N\}$ 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 گره‌های همسایه را، با نام

اهدای 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 گره‌های نیاز به فضای در دسترس دارد. اگر تمام همسایگان هیچ فضایی در سطح محلی نداشته باشند، گره i ، یک همسایه که حداقل همسایه‌های (در 2 هاپ از گره i) با فضای در دسترس دارد در جهت بالا و/یا در جهت پایین تر 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 در حافظه محلی گره i به شرط داشتن فضای محلی ذخیره کرد.



شکل 2: تکرار هاپ توسط هاپ در مورد $R = 3$ به کپی نیاز دارد. داده‌ها در جهت پایین از RPL DODAG به‌عنوان خاشرات 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 نسخه‌های دیگر هنوز هم در WSN توزیع شده هستند. با توجه به اصل مشخص فوق، تعدادی از کپی‌های مورد نیاز در $R-1$ (در صورتی گره i بتواند داده‌های 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

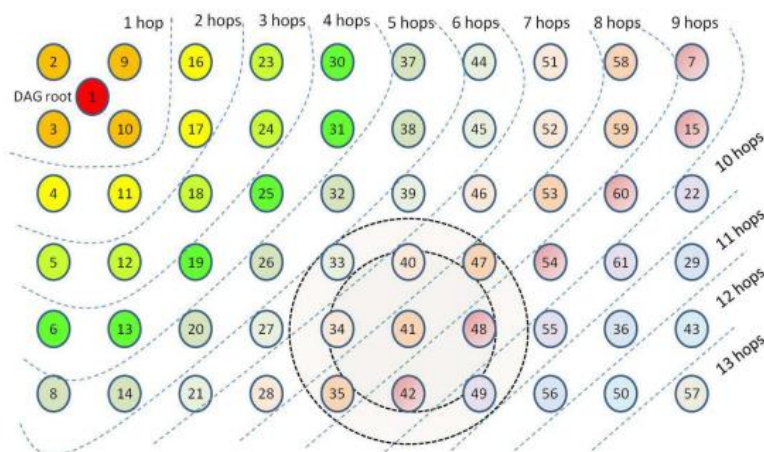
محلی داشته باشد، و گره های اهداکننده ی بعدی را در میان همسایگان خود، دور از ارسال گره و گره منبع از گره نامزد انتخاب می کند. اهداکننده ی بعدی با همین قانون انتخاب می شود، اولویت بندی گره ها به ریشه DAG نزدیک تر است.

این اجازه می دهد تا کپی به خوبی در سراسر DODAG گسترش یابد. در این مرحله، گره اهدا کننده کپی به گره های اهداکننده ی انتخاب شده ی بعدی با کاهش تعداد کپی مورد نیاز توسط 1 ارسال می کند. فرآیند تکرار به صورت بازگشتی تا زمانی که آخرین (R امین) کپی ذخیره شود و یا متوقف شود ادامه می یابد وقتی که یکی از گره های اهداکننده می تواند گره اهداکننده ی 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 نمی تواند یک همسایه با رتبه RPL پایین تر (به عنوان مثال، نزدیک تر به ریشه ی 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 $R=3$ کپی مورد نظر نشان داده شده است. گره همیشه سعی می کند کپی را به والد در جهت بالای DODAG، به عنوان مثال، گره ها با یک رتبه پایین تر، توزیع کند. برای اشباع شبکه، داده ها به سمت گره با رتبه بالاتر، به عنوان مثال، در جهت پایین تر از DODAG، توزیع می شوند.

طبق بازیابی داده ها، درخواست بازیابی دور های توسط ریشه DAG فرستاده می شود و تبلیغ به تمام گره های DODAG انتقال می شود. بازیابی داده ها پس از دریافت درخواست از هر گره به سمت ریشه DAG، به دنبال مسیر پیش فرض RPL به سمت ریشه اتفاق می افتد. چنین الگوی ترافیک many-to-many، اگر به دقت به کار گرفته شود، می تواند 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 داده ها نزدیک به ریشه DAG باشد، فرستاده می شود.

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.

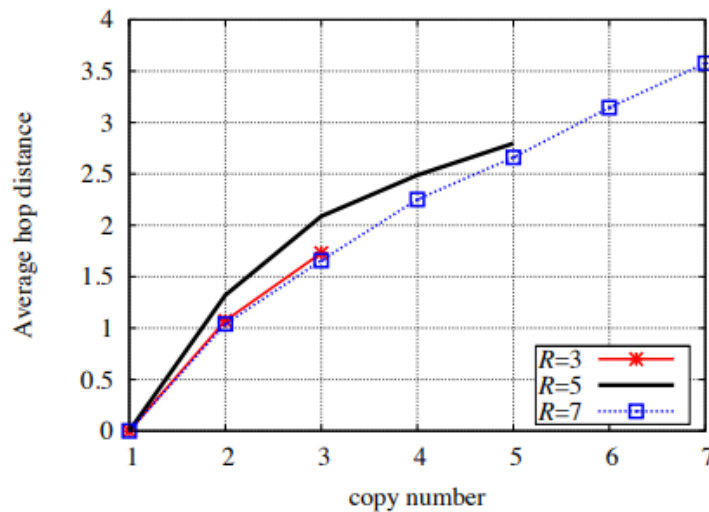
ذخیره‌سازی توزیع‌شده‌ی پیشنهادی و مکانیزم بازیابی در Contiki 2.5 و در Cooja، یک شبیه‌ساز شبکه گیرنده بی‌سیم مبتنی بر جاوا [6]، پیشنهاد شده است. سناریوی شبیه‌سازی شده، در شکل 3 نشان داده شده است، که از 60 = N گره ذخیره‌سازی، که در یک شبکه مستطیل شکل و یک گره اضافی که به عنوان ریشه DAG عمل می‌کند، تشکیل 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 گره‌های 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، گره‌های: 33، 35، 47 و 49.

توجه داشته باشید که گره‌های در امتداد مرزها، همسایه‌ی کمتری از آن‌هایی که در داخل هستند دارند.

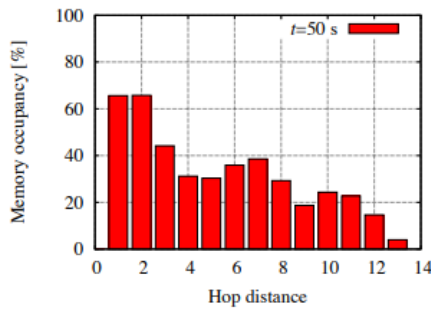
فاصله بین دو بازیابی داده‌ی متوالی $T = 10$ دقیقه است. دوره سنجش گره‌ها یک عدد صحیح است که به‌طور تصادفی و به‌طور مستقل در محدوده‌ی [1,9] انتخاب شده است. همه گره‌ها اندازه بافر یکسانی برابر با $B = 100$ واحد داده دارند. دوره تبلیغات حافظه T_{adv} مقدار 30 است. ما تعداد مورد انتظار انتقال (ETX) را به‌عنوان متریک RPL انتخاب می‌کنیم. به‌طور خاص، 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 R 1 = بدون تکرار است، به‌عنوان مثال، تنها نسخه اصلی ذخیره شده است.

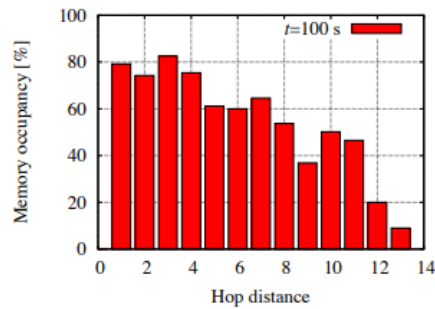
با توجه به توزیع فضایی کپی، شکل 4 فاصله هاپ متوسط رسیده توسط کپی های اضافی از صاحب اصلی آن، برای مقادیر مختلف R را نشان می‌دهد. نتایج ما نشان می‌دهد که کپی های متوالی تمایل به حرکت دور از گره منبع در امتداد DODAG را دارند. در حالی که مکانیسم پیشنهادی در [4] از نظر گسترش ذخیره‌سازی ناکارآمد بود (در واقع، فاصله هاپ در حالت متوسط حدوداً 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 و (R=3) 1 و 5 و (R=5) 1 و 7 و (R=7) است. فاصله از موقعیت اولیه‌ی replica محاسبه می‌شود.



(a) $t = 50$ s



(b) $t = 100$ s

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 قبلاً 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 ($R=1$)، مقدار منحصر به فرد داده های ذخیره شده به ظرفیت ذخیره‌سازی شبکه

$C = N \times B = 6000$ واحد داده می‌رسد. در حضور افزونگی ($R>1$)، 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 $R > 1$ ، داده‌ی کمتری به ریشه DAG تحویل داده شده است، بنابراین تنها

نزدیک‌ترین replica ارسال می‌شود. از سوی دیگر، با $R = 1$ ، تمام $N \times B = 6000$ واحد داده به sink تحویل داده

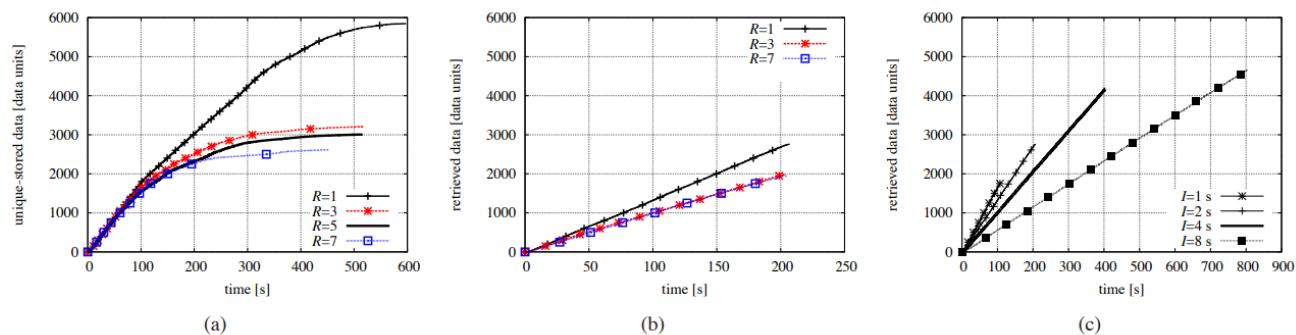
می‌شود. این باعث افزایش مقدار ترافیک many-to-one می‌شود و در نتیجه، خطر برخورد وجود دارد. برای کاهش

اثرات تصادم، واحد داده‌ها از هر گره به سمت ریشه DAG، یک‌دریک در زمان منتقل می‌شوند. فاصله‌ی بین دو انتقال

متوالی به عنوان 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 λ ممکن است مقدار داده‌های بازبازی در sink را افزایش دهد و تاخیر فاز بازبازی را جریمه کند. در شکل 6 (ب)، مقدار داده‌های بازبازی در ریشه DAG به‌عنوان یک تابع از زمان، برای مقادیر مختلف R نشان داده شده است. ورودی λ مقدار S_2 می‌گیرد. همان‌گونه که انتظار می‌رود، میزان داده‌های بازبازی با R کاهش می‌یابد، زیرا داده‌های منحصربه‌فرد بیشتری وجود دارد که در سیستم زمانی بدون افزونگی مورد نیاز است. اگر ما بتوانیم آن را با $R = 1$ مشاهده کنیم، تنها بخش کوچکی، به‌عنوان مثال، حدود 50 درصد از داده‌های ذخیره شده $C = N \times B$ ، با موفقیت بازبازی می‌شود؛ بقیه به دلیل برخورد از دست می‌رود. برای مقادیر بالاتر R ، درصد مربوطه از داده‌های بازبازی به‌طور قابل توجهی افزایش می‌یابد. به‌عنوان مثال، در مورد $R=7$ ، حدود 2000 واحد داده بازبازی می‌شود، که مربوط به حدود 80 درصد از مقدار کل داده‌های منحصربه‌فرد است.

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



شکل 6: داده‌های بازبازی و ذخیره شده با توجه به اندازه‌ی حافظه برابر با 100 واحد داده، گره‌های ذخیره‌سازی $N = 60$ ، و یک دوره تبلیغات حافظه 30 ثانی‌های؛ (الف) اطلاعات منحصربه‌فرد ذخیره شده است. (ب) داده‌های بازبازی شده برای چندین مقدار R با $\lambda=2$ ؛ (ج) داده‌های بازبازی شده برای چندین مقدار λ ، با $R = 1$.

به طور خلاصه، 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.

	$I = 1 s$	$I = 2 s$	$I = 4 s$	$I = 8 s$
$R = 1$	33%	50%	68%	82%
$R = 3$	49%	66%	95%	100%
$R = 5$	50%	74%	95%	100%
$R = 7$	55%	80%	100%	100%

جدول 2: درصد بازیابی داده ها.

6. نتیجه گیری

این مقاله با مشکل داده های تکراری توزیع شده و بازیابی برای نظارت مبتنی بر WSN مواجه است. مکانیزم ذخیره سازی داده های توزیع شده در بالای RPL، به منظور افزایش انعطاف پذیری رویکرد پیشنهادی و ظرفیت ذخیره سازی WSN در برابر خرابی گره و کمبود حافظه محلی ساخته شده است. عملکرد از طریق شبیه سازی مورد ارزیابی قرار گرفت. مکانیسم مستقیماً به اجرای یک طرح بازیابی داده ها منجر شد، تا عملکرد آن به خوبی انجام شود. نتایج ما به وضوح یک افزونگی بین ذخیره سازی (که باعث آلوده شدن کل حافظه ذخیره سازی در دسترس شده است) و بازیابی بهره وری (از نظر درصد داده های بازیابی) را نشان می دهد.

فعالیت های تحقیقاتی آینده شامل استفاده از سایر پارامترها در استراتژی تکرار، مانند مصرف انرژی گره ها یا قابلیت دسترسی گره است، به ویژه اگر آنها در لایه MAC با وظیفه چرخه (به عنوان مثال، ContikiMAC و یا X-MAC) عمل کنند. ما همچنین مطالعه ی پیکربندی دوباره ی پویا از رفتار گره (به عنوان مثال، نمونه برداری) و لایه های ارتباطی (به عنوان مثال، قدرت انتقال) را برای مواجهه با تکرار خواسته ها با حداقل هزینه انرژی را پیگیری خواهیم کرد. در نهایت، ما می خواهیم عملکرد مکانیسم در دنیای واقعی را ارزیابی کنیم.

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 zebrantet," 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.