

الگوریتم مسیریابی تحمل‌پذیر خطا مبتنی بر اتوماتای یادگیر برای خطاهای ناسازگاری داده در شبکه‌های حسگر بی‌سیم

سید محمد ابوالحسنی	مهدی اثنی‌عشری	محمد رضا میبدی
آزمایشگاه محاسبات نرم	آزمایشگاه محاسبات نرم	آزمایشگاه محاسبات نرم
دانشکده مهندسی کامپیوتر	دانشکده مهندسی کامپیوتر	دانشکده مهندسی کامپیوتر
دانشگاه صنعتی امیرکبیر	دانشگاه صنعتی امیرکبیر	دانشگاه صنعتی امیرکبیر
تهران ایران	تهران ایران	تهران ایران
sm_abolhasani@aut.ac.ir	esnaashari@aut.ac.ir	mmeybodi@aut.ac.ir

حسگرهای همسایه ارسال دارند [1]. شبکه‌های حسگر می‌توانند در کاربردهایی مانند نظارت هوشمند بر بزرگراه‌ها، امداد رسانی در حوادث غیرمترقبه، دیده‌بانی محیط و پیگیری هدف به کار گرفته شوند [2,3]. با توجه به نوع کاربرد این شبکه‌ها، تحمل‌پذیری خطا امری مهم در آنها محسوب می‌شود. این اهمیت خصوصاً در کاربردهائی نظیر محیط‌های نظامی و محیط‌های آزمایشات هسته‌ای به طور مشخص‌تری دیده می‌شود. از آنجا که در چنین محیط‌هایی اطلاعات بسیار حیاتی است، لازم است که اولاً اطلاعات به مقصد برسد و ثانياً اطلاعات رسیده اطلاعات درستی باشد. زیرا تصمیم‌گیری بر اساس اطلاعات اشتباه بسیار بدتر از عدم تصمیم‌گیری است.

اغلب فعالیت‌های صورت پذیرفته تاکنون در زمینه تحمل‌پذیری خطا در شبکه‌های حسگر بر پایه افزونگی در تعداد گره‌ها، مسیرها و یا تعداد گره‌های مرکزی بنا شده‌اند. توجه اصلی این تحقیقات برچگونگی ترکیب و انتشار داده در پروتکل‌های مسیریابی به منظور افزایش میزان تحمل‌پذیری خطا می‌باشد [5-8]. استفاده از تکنیک‌های مسیریابی چند مسیری (استفاده از چند مسیر بین مبدأ و مقصد برای اطمینان از رسیدن داده) به منظور افزایش قابلیت اطمینان از مهمترین راه‌کارها در این زمینه به شمار می‌رود [9-15].

توجه اصلی اکثر تحقیقات در این زمینه معطوف به تحمل خطاهایی بوده که در آنها گره به طور کامل خراب شده است و تلاش‌های کمی برای خطاهای ناسازگاری داده صورت پذیرفته است. خطاهای ناسازگاری داده، به علت تغییر محتوای باینری بسته داده در حین پردازش به وجود می‌آید. این خطا که به عنوان خطای نرم در [16] معرفی شده است، هنگامی رخ می‌دهد که محتوای بسته d_{pkt} دریافت شده به وسیله گره n مشابه محتوای بسته d_{pkt} فرستاده شده توسط گره n نباشد. خطای ناسازگاری داده می‌تواند در یک گره به صورت پایدار یا گذرا رخ

چکیده: از جمله مهم‌ترین چالش‌های مطرح در شبکه‌های حسگر بی‌سیم خطا و خرابی گره می‌باشد. اکثر تحقیقات صورت پذیرفته در این زمینه، معطوف به تحمل خطاهایی بوده که در آنها گره به طور کامل خراب می‌شود، اما تلاش‌های کمی برای خطاهای ناسازگاری داده^۱ صورت پذیرفته است. خطای ناسازگاری داده خطائی است که بیانگر اختلاف مابین بسته داده ارسال شده و بسته داده دریافت شده در گره مقصد می‌باشد. این خطا به علت تغییر محتوای باینری بسته داده در حین پردازش به وجود می‌آید. در این مقاله پروتکل مسیریابی تحمل‌پذیر خطا برای خطاهای ناسازگاری داده به نام LADIT^۲ پیشنهاد می‌گردد. در این پروتکل، محیط شبکه به سلول‌هایی مجازی تقسیم شده و مسیریابی میان سلول‌های مجاور صورت می‌پذیرد. سه گره نماینده در هر سلول مسؤول انتقال اطلاعات سلول به سلول‌های مجاور می‌باشند. هر گره نماینده در هر سلول بسته داده را از سه گره نماینده سلول مجاور دریافت می‌کند و با استفاده از رأی‌گیری اکثریت^۳ بین بسته‌های دریافتی، خطای ناسازگاری داده را یافته و بر آن غلبه می‌نماید. پروتکل LADIT از اتوماتای یادگیر به منظور انتخاب نماینده‌های سلول استفاده می‌کند. به منظور ارزیابی، پروتکل LADIT با استفاده از نرم-افزار شبیه‌ساز ns2 شبیه‌سازی شده است. نتایج شبیه‌سازی عملکرد مطلوب پروتکل پیشنهادی را نشان می‌دهد.

واژه‌های کلیدی: شبکه‌های حسگر، مسیریابی، تحمل‌پذیری خطا، اتوماتای یادگیر، خطای ناسازگاری داده

مقدمه

شبکه‌های حسگر که در سال‌های اخیر مورد توجه بسیار قرار گرفته‌اند از تعداد زیادی (که ممکن است به هزاران مورد برسد) گره حسگر کوچک، ارزان قیمت با توانایی بالا و ذخیره انرژی پایین تشکیل شده‌اند. این حسگرها می‌توانند اطلاعاتی (نظیر دما، رطوبت، فشار و ...) را از محیط اطراف خود دریافت کرده و برای

دهد. ایرادهای داخلی در مؤلفه‌های سخت‌افزاری همچون پردازنده و یا واحدهای حافظه می‌توانند باعث بروز خطاهای ناسازگاری داده شوند. برخلاف خرابی‌های توان، یک گره که دچار خطای ناسازگاری داده می‌باشد، برخی از سرویس‌های خود را به درستی انجام می‌دهد، اما در انجام برخی دیگر از سرویس‌ها دچار خطا می‌گردد. از محدود فعالیت‌های صورت پذیرفته در زمینه تحمل‌پذیری خطای ناسازگاری داده، پروتکلی است که توسط سو و همکارانش^۴ ارائه شده است [4]. روال کاری این پروتکل که ما آن را FDDA^۵ می‌نامیم به این صورت است که ابتدا دو مسیر کاملاً مجزا بین منبع و گره مرکزی ایجاد می‌شود و سپس داده در دو نسخه و از هر دوی این مسیرها ارسال می‌شود. در گره مرکزی داده‌ها با یکدیگر مقایسه می‌شوند و در صورت یکسان بودن مورد تأیید قرار می‌گیرند. در غیر این صورت، یک مسیر جدید (بدون هر گونه اشتراک با مسیرهای قبلی) بین منبع و گره مرکزی ایجاد شده و سپس داده در سه نسخه و از دو مسیر قبلی و مسیر جدید ارسال می‌شود. گره مرکزی از رای‌گیری اکثریت بین بسته‌های داده دریافتی برای تشخیص بسته داده صحیح استفاده می‌کند. در پروتکل FDDA به منظور تشخیص گره خراب، گره مرکزی برای هر مسیر نادرست پیغام خطا را ارسال می‌کند (با توجه به نگهداری مسیر بازگشتی توسط هر گره) که با این پیغام، مقدار خطای هر یک از گره‌های مسیر دارای خطا افزایش می‌یابد و هرگاه برای گره‌ای این میزان خطا از حد آستانه‌ای بیشتر شد آن گره به عنوان یک گره کاملاً خراب تشخیص داده شده و غیرفعال می‌شود. گره مرکزی برای هر مسیر درست نیز پیغام درستی مسیر را ارسال می‌کند و میزان خطای گره‌های مسیر صحیح برابر صفر می‌شود. مکانیزم تشخیص گره خراب به این طریق ممکن است با اشتباه همراه باشد و گره سالم، خراب تشخیص داده شود. چرا که یک گره سالم ممکن است به خاطر بودن در مسیرهای دارای خطا، دارای پارامتر مقدار خطای بیش از حد آستانه مورد نظر شود و گره خراب تشخیص داده شود، در حالی که خطا را گره‌های دیگری انجام داده‌اند.

در این مقاله پروتکل مسیریابی جدیدی به نام LADIT پیشنهاد می‌گردد. در این پروتکل، محیط شبکه به سلول‌هایی مجازی تقسیم می‌شود و مسیریابی میان این سلول‌ها صورت می‌پذیرد. در هر سلول سه گره به عنوان نماینده (برای ارسال داده) انتخاب می‌شوند. به این طریق هر گره در سلول C_i بسته داده را از سه گره سلول C_{i-1} دریافت می‌کند. تأیید بسته در سلول C_i بر اساس مکانیزم رأی‌گیری اکثریت انجام می‌شود [17,18]. ارتباط سلول‌های مجاور تا رسیدن بسته داده به گره مرکزی ادامه می‌یابد. در پروتکل LADIT از اتوماتای یادگیر به منظور انتخاب گره‌های نماینده استفاده می‌شود. اتوماتای یادگیر گره‌های

نماینده را به نحوی انتخاب می‌کند که احتمال انتخاب گره‌های سالم بیشتر شده و در نتیجه نرخ داده‌های مورد تأیید دریافتی در گره مرکزی افزایش یابد. به منظور ارزیابی، پروتکل‌های LADIT و FDDA با استفاده از نرم‌افزار شبیه‌ساز ns2 [19] شبیه‌سازی و نتایج به دست آمده با یکدیگر مقایسه شده‌اند. نتایج مقایسه برتری پروتکل پیشنهادی را نشان می‌دهد.

سازماندهی ادامه مقاله به این صورت است. در بخش ۲ اتوماتاهای یادگیر به اختصار شرح داده می‌شود. در بخش ۳ پروتکل پیشنهادی توضیح داده می‌شود. محاسبه قابلیت اطمینان و نتایج شبیه‌سازی‌ها در بخش‌های ۴ و ۵ آورده شده است. بخش ۶ نتیجه‌گیری می‌باشد.

اتوماتاهای یادگیر

اتوماتای یادگیر [20,21]، ماشینی است که می‌تواند تعدادی متناهی عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی می‌شود و نتیجه ارزیابی در قالب سیگنالی مثبت یا منفی به اتوماتا داده می‌شود و اتوماتا از این پاسخ در انتخاب عمل بعدی تأثیر می‌گیرد. هدف نهایی آن است که اتوماتا یاد بگیرد که از بین اعمال خود بهترین عمل را انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداکثر برساند. کارکرد اتوماتای یادگیر در تعامل با محیط، در شکل ۱ مشاهده می‌شود.



محیط را می‌توان توسط سه تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودی‌ها، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه خروجی‌ها و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمال‌های جریمه می‌باشد. c_i احتمال اینکه عمل α_i نتیجه نامطلوب داشته باشد می‌باشد. در محیط ایستا مقادیر c_i بدون تغییر می‌مانند، حال آنکه در محیط غیر ایستا این مقادیر در طی زمان تغییر می‌کنند. اتوماتاهای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم‌بندی می‌گردند. در ادامه اتوماتای یادگیر با ساختار متغیر معرفی می‌شود.

اتوماتای یادگیر با ساختار متغیر را می‌توان توسط چهار تایی $\{\alpha, \beta, p, T\}$ نشان داد که $\alpha = \{\alpha_1, \dots, \alpha_r\}$ مجموعه عمل‌های اتوماتا، $\beta = \{\beta_1, \dots, \beta_m\}$ مجموعه ورودی-

های اتوماتا، $p = \{p_1, \dots, p_r\}$ بردار احتمال انتخاب هریک از عمل‌ها و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می‌باشد. الگوریتم زیر یک نمونه از الگوریتم‌های یادگیری خطی است. فرض کنید عمل α_i در مرحله n ام انتخاب شود.

پاسخ مطلوب

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1-a)p_j(n) \quad \forall j \neq i \end{aligned} \quad (1)$$

پاسخ نامطلوب

$$\begin{aligned} p_i(n+1) &= (1-b)p_i(n) \\ p_j(n+1) &= (b/r-1) + (1-b)p_j(n) \quad \forall j \neq i \end{aligned} \quad (2)$$

در روابط (۱) و (۲)، a پارامتر پاداش و b پارامتر جریمه می‌باشند. با توجه به مقادیر a و b سه حالت زیر را می‌توان در نظر گرفت. زمانی که a و b با هم برابر باشند، الگوریتم را L_{RP} می‌نامند، زمانی که b از a خیلی کوچکتر باشد، الگوریتم را L_{REP} می‌نامند و زمانی که b مساوی صفر باشد الگوریتم را L_{RI} می‌نامند [22].

در [23] نسخه‌ای از اتوماتای یادگیر L_{RI} معرفی شده است که در آن در هر گام s تنها زیرمجموعه $\alpha(s) = \{\alpha_1, \dots, \alpha_{r_s}\}$ ($r_s \leq r$ و $\alpha(s) \subseteq \alpha$) از عمل‌های اتوماتا قابل اعمال می‌باشند. این نوع اتوماتا، اتوماتای یادگیر با مجموعه عمل‌های متغیر نام دارد. به منظور آنکه مجموع احتمالات عمل‌های قابل اعمال این نوع از اتوماتای یادگیر در هر گام برابر ۱ باشد، لازم است که تغییری در بردار احتمال عمل‌ها صورت پذیرد. اگر \hat{p} بردار احتمال حاصل از این تغییر موقت باشد، رابطه (۳) نحوه محاسبه آن را نشان می‌دهد.

$$M = \sum_{j=1}^{r_s} p_j(s) \quad \forall j \text{ such that } \alpha_j \in \alpha(s) \quad (3)$$

$$\hat{p}_i(s) = \frac{p_i(s)}{M} \quad \forall i \text{ such that } \alpha_i \in \alpha(s)$$

الگوریتم زیر الگوریتم یادگیری خطی برای این نوع اتوماتا است. فرض کنید عمل α_i در مرحله s از مجموعه عمل‌های $\alpha(s)$ انتخاب شود.

پاسخ مطلوب

$$\begin{aligned} \hat{p}_i(s+1) &= \hat{p}_i(s) + a[1 - \hat{p}_i(s)] \\ \hat{p}_j(s+1) &= (1-a)\hat{p}_j(s) \quad \forall j \neq i \end{aligned} \quad (4)$$

پاسخ نامطلوب

$$\begin{aligned} \hat{p}_i(s+1) &= (1-b)\hat{p}_i(s) \\ \hat{p}_j(s+1) &= (b/r_s-1) + (1-b)\hat{p}_j(s) \quad \forall j \neq i \end{aligned} \quad (5)$$

در روابط (۴) و (۵)، a پارامتر پاداش و b پارامتر جریمه می‌باشند. به منظور برگشت اتوماتا به شرایط عادی و شروع مرحله $s+1$ بردار احتمال اتوماتا با توجه به رابطه (۶) به روز می‌شود. همان گونه که دیده می‌شود احتمال عمل‌هایی که در مرحله s قابل اعمال نیستند تغییری نمی‌کند.

$$p_i(s+1) = \hat{p}_i(s+1) \times M \quad \forall i \text{ such that } \alpha_i \in \alpha(s) \quad (6)$$

$$p_i(s+1) = p_i(s) \quad \forall i \text{ such that } \alpha_i \notin \alpha(s)$$

پروتکل مسیریابی ارائه شده در این مقاله به نوع خاصی از اتوماتا نیاز دارد که در آن در هر مرحله بیش از یک عمل توسط اتوماتا انتخاب می‌شود. این اتوماتا که ما آن را اتوماتای یادگیر با k انتخاب عمل (KSALA)^۶ می‌نامیم بسیار مشابه اتوماتای یادگیر با ساختار متغیر است، با این تفاوت که در اتوماتای یادگیر با k انتخاب عمل، در هر مرحله، k عمل (متفاوت) از اتوماتا انتخاب می‌شود و سپس نتیجه هر یک از عمل‌های انتخابی مشخص شده و احتمالات عمل‌ها به‌روز می‌شود. اتوماتای یادگیر با k انتخاب عمل را می‌توان توسط چهارتایی $\{\alpha, \beta, p, T\}$ نشان داد که $\alpha = \{\alpha_1, \dots, \alpha_r\}$ مجموعه عمل‌های اتوماتا، $\beta = \{\beta_1, \dots, \beta_m\}$ مجموعه ورودی‌های اتوماتا، $p = \{p_1, \dots, p_r\}$ بردار احتمال انتخاب هریک از عمل‌ها و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می‌باشد. در این اتوماتا در هر مرحله k عمل انتخاب شده و در محیط اعمال می‌شوند. $\alpha^h(s) = \{\alpha_1, \dots, \alpha_{r_s^h}\}$ ($r_s^h \leq r$) و $\alpha^h(s) \subseteq \alpha$ در این اتوماتا مجموعه عمل‌های قابل اعمال اتوماتا برای انتخاب عمل h ام در مرحله s است که $\alpha^1(s) = \alpha$ است. در این نوع اتوماتا در مرحله s عمل انتخاب شده h ام از مجموعه عمل‌های قابل اعمال اتوماتا برای انتخاب عمل‌های بعدی حذف می‌شود. سه نوع الگوریتم یادگیری خطی برای این اتوماتا پیشنهاد می‌شود. الگوریتم اول که آن را الگوریتم یادگیری خطی ساده می‌نامیم، توسط روابط (۷) و (۸) بیان می‌شود. فرض کنید عمل انتخاب شده h ام در مرحله h ام عمل α_i باشد.

پاسخ مطلوب

$$\begin{aligned} p_i^{h+1}(n) &= p_i^h(n) + a[1 - p_i^h(n)] \\ p_j^{h+1}(n) &= (1-a)p_j^h(n) \quad \forall j \neq i \end{aligned} \quad (7)$$

پاسخ نامطلوب

$$\begin{aligned} p_i^{h+1}(n) &= (1-b)p_i^h(n) \\ p_j^{h+1}(n) &= (b/r-1) + (1-b)p_j^h(n) \quad \forall j \neq i \end{aligned} \quad (8)$$

ارتباط روابط (۷) و (۸) با بردار احتمال اصلی توسط رابطه (۹) بیان می‌شود.

$$M = \sum_{j=1}^{r_i^h} p_j^h(s) \quad \forall j \text{ such that } \alpha_j \in \{\alpha - \alpha_k(s) + \alpha_i\} \quad (۱۶)$$

$$\hat{p}_i^h(s) = \frac{p_i^h(s)}{M} \quad \forall i \text{ such that } \alpha_i \in \{\alpha - \alpha_k(s) + \alpha_i\}$$

در روابط (۱۴) و (۱۵)، پارامتر پاداش و b پارامتر جریمه می‌باشند. به منظور برگشت اتوماتا به شرایط عادی و انتخاب عمل $h+1$ بردار احتمال اتوماتا با توجه به رابطه (۱۷) به‌روز می‌شود.

$$p_i^{h+1}(s) = \hat{p}_i^{h+1}(s) \times M \quad \forall i \text{ such that } \alpha_i \in \{\alpha - \alpha_k(s) + \alpha_i\} \quad (۱۷)$$

$$p_i^{h+1}(s) = p_i^h(s) \quad \forall i \text{ such that } \alpha_i \notin \{\alpha - \alpha_k(s) + \alpha_i\}$$

ارتباط روابط (۱۴) و (۱۵) با بردار احتمال اصلی توسط رابطه (۹) بیان می‌شود.

پروتکل پیشنهادی (LADIT)

پروتکل مسیریابی تحمل‌پذیر خطای ناسازگاری داده با استفاده از اتوماتای یادگیر پروتکلی است که به جای مسیریابی بین گره‌ها، مسیریابی را بین سلول‌ها انجام می‌دهد. در این پروتکل محیط شبکه به سلول‌هایی مجازی تقسیم می‌شود. سه گره از گره‌های داخل هر سلول به عنوان نماینده آن سلول انتخاب می‌شوند و به ترتیب نماینده‌های اول تا سوم آن سلول نامیده می‌شوند. نماینده‌ها وظیفه مسیریابی و ارسال اطلاعات به سمت گره مرکزی را بر عهده دارند. در صورتی که یک بسته اطلاعاتی از طرف سلول i در سلول z دریافت شود، نماینده اول سلول z وظیفه ارسال Ack را به سلول i بر عهده خواهد داشت. از پیغام Ack در سلول i به منظور انتخاب مسیر بهتر و نماینده‌هایی شایسته‌تر استفاده می‌شود. پروتکل LADIT به منظور انتخاب نماینده‌های سلول‌ها از اتوماتای یادگیر استفاده می‌کند.

جزئیات پروتکل به قرار زیر است:

یک توری منظم فرضی در سطح شبکه در نظر گرفته می‌شود. شکل‌های مختلفی برای سلول‌های توری فرضی می‌توان در نظر گرفت. توری منظم متشکل از سلول‌های مربع یا عضلانی در شکل ۲ دیده می‌شود. ابعاد این توری به گونه‌ای انتخاب می‌شود که گره‌های موجود در سلول‌های همسایه در محدوده انتقال یکدیگر قرار گیرند. همسایگی‌های مختلفی را می‌توان برای سلول‌ها در نظر گرفت. در توری $SC4^v$ سلول‌های مجاور به صورت افقی و عمودی به عنوان همسایه در نظر گرفته می‌شوند (شکل ۳.۳). در توری $SC8^u$ سلول‌های مجاور به صورت افقی و عمودی و قطری به عنوان همسایه در نظر گرفته می‌شوند (شکل ۳.۳). در توری HC^9 سلول‌های مجاور به عنوان همسایه در نظر گرفته می‌شوند (شکل ۳.۳). اطلاعات و بسته‌ها تنها می‌توانند از یک سلول به سلول همسایه منتقل شوند، بنابراین بسته‌های دریافتی در یک سلول که از سلول همسایه نباشند مورد پردازش قرار نمی‌گیرند.

$$p_i^1(n) = p_i(n) \quad \forall i \text{ such that } p_i \in p(n) \quad (۹)$$

$$P_i(n+1) = p_i^{k+1}(n) \quad \forall i \text{ such that } p_i \in p(n+1)$$

الگوریتم یادگیری دوم که آن را الگوریتم یادگیری خطی ترتیب‌دار می‌نامیم، توسط روابط (۱۰) و (۱۱) بیان می‌شود. فرض کنید عمل انتخاب شده h ام در مرحله n ام عمل α_i باشد. \hat{p} در روابط (۱۰) و (۱۱) از رابطه (۱۲) به دست می‌آید.

پاسخ مطلوب

$$\hat{p}_i^{h+1}(n) = \hat{p}_i^h(n) + a[1 - \hat{p}_i^h(n)] \quad (۱۰)$$

$$\hat{p}_j^{h+1}(n) = (1-a)\hat{p}_j^h(n) \quad \forall j \quad j \neq i$$

پاسخ نامطلوب

$$\hat{p}_i^{h+1}(n) = (1-b)\hat{p}_i^h(n) \quad (۱۱)$$

$$\hat{p}_j^{h+1}(n) = (b/r_s^h - 1) + (1-b)\hat{p}_j^h(n) \quad \forall j \quad j \neq i$$

$$M = \sum_{j=1}^{r_i^h} p_j^h(s) \quad \forall j \text{ such that } \alpha_j \in \alpha^h(s) \quad (۱۲)$$

$$\hat{p}_i^h(s) = \frac{p_i^h(s)}{M} \quad \forall i \text{ such that } \alpha_i \in \alpha^h(s)$$

در روابط (۱۰) و (۱۱)، پارامتر پاداش و b پارامتر جریمه می‌باشند. به منظور برگشت اتوماتا به شرایط عادی و انتخاب عمل $h+1$ بردار احتمال اتوماتا با توجه به رابطه (۱۳) به‌روز می‌شود. ارتباط روابط (۱۰) و (۱۱) با بردار احتمال اصلی توسط رابطه (۹) بیان می‌شود.

$$p_i^{h+1}(s) = \hat{p}_i^{h+1}(s) \times M \quad \forall i \text{ such that } \alpha_i \in \alpha^h(s) \quad (۱۳)$$

$$p_i^{h+1}(s) = p_i^h(s) \quad \forall i \text{ such that } \alpha_i \notin \alpha^h(s)$$

الگوریتم یادگیری سوم که آن را الگوریتم یادگیری خطی بدون ترتیب می‌نامیم، توسط روابط (۱۴) و (۱۵) بیان می‌شود. فرض کنید عمل انتخاب شده h ام در مرحله n ام عمل α_i باشد. \hat{p} در روابط (۱۴) و (۱۵) از رابطه (۱۶) به دست می‌آید. در این رابطه $\alpha_k(s)$ نشاندهنده k عمل انتخاب شده در مرحله s است.

پاسخ مطلوب

$$\hat{p}_i^{h+1}(n) = \hat{p}_i^h(n) + a[1 - \hat{p}_i^h(n)] \quad (۱۴)$$

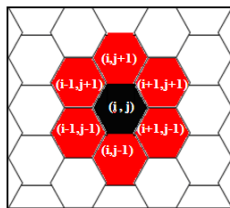
$$\hat{p}_j^{h+1}(n) = (1-a)\hat{p}_j^h(n) \quad \forall j \quad j \neq i$$

پاسخ نامطلوب

$$\hat{p}_i^{h+1}(n) = (1-b)\hat{p}_i^h(n) \quad (۱۵)$$

$$\hat{p}_j^{h+1}(n) = (b/r_s^h - 1) + (1-b)\hat{p}_j^h(n) \quad \forall j \quad j \neq i$$

$$R^2 = \left(\frac{5}{2}a\right)^2 + \left(\frac{3\sqrt{3}}{2}a\right)^2 \Rightarrow a_{HC} = \frac{R}{\sqrt{13}} \quad (20)$$



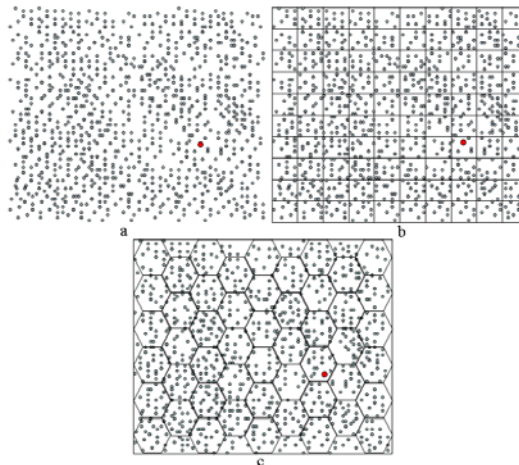
شکل(۵): شماره گذاری سلول‌ها در توری با سلول‌های عضلانی

همانگونه که بیان شد، در این الگوریتم، مسیریابی بین سلول‌های ایجاد شده صورت می‌پذیرد. جدول مسیریابی دارای فیلدهای شماره سلول و میزان خطای سلول است و هر رکورد آن مربوط به اطلاعات یکی از سلول‌های بعدی برای مسیریابی می‌باشد. پروتکل دارای دو مرحله "احوالپرسی اولیه" و ایجاد جداول مسیریابی و "انتشار رخدادها" است.

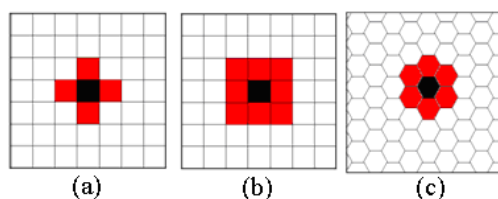
۱-۱- مرحله احوالپرسی اولیه و ایجاد جداول

در این مرحله، گره‌ها با تبادل اطلاعات، سلول‌ها و نماینده‌ها را مشخص می‌کنند و با انتشار کلی که گره مرکزی انجام می‌دهد جداول مسیریابی را تشکیل می‌دهند. در ابتدا هر گره شماره شناسائی، مکان فیزیکی و میزان انرژی خود را در همسایگی خود منتشر می‌کند. بعد از فرستادن این پیغام توسط همه گره‌ها، هر گره با توجه به توری و ابعاد در نظر گرفته شده برای هر سلول (فرض می‌کنیم که این اطلاعات به صورت کد شده در حافظه گره قرار دارد)، متوجه می‌شود که در چه سلولی قرار دارد و با چه گره‌هایی هم سلول است. گره دارای بیشترین انرژی در هر سلول به عنوان نماینده اول آن سلول در نظر گرفته می‌شود. به منظور کاهش مصرف انرژی، تنها نماینده‌های اول هر سلول در مرحله ایجاد جداول مسیریابی شرکت می‌کنند.

در ادامه، گره مرکزی پیغام Flood را تولید می‌کند. این پیغام دارای دو فیلد شماره سلول و تعداد گام تا گره مرکزی می‌باشد. گره مرکزی فیلدهای پیغام Flood را مقداردهی کرده و آن را در سطح شبکه منتشر می‌کند. هر گره (نماینده اول سلول‌ها) با دریافت اولین بسته Flood، اطلاعات آن را به عنوان یک رکورد جدید در جدول مسیریابی خود وارد می‌کند. گره مقادیر فیلدهای بسته Flood را به‌روز می‌کند و سپس آن را در شبکه منتشر می‌کند. به‌روز کردن فیلدهای بسته Flood با قرار دادن شماره سلول گره در فیلد شماره سلول و اضافه کردن یک واحد به مقدار قبلی فیلد تعداد گام تا گره مرکزی انجام می‌شود. گره‌ها در مقابل دریافت بسته‌های Flood بعدی بسته به مورد، به یکی از روش‌های زیر عمل می‌کنند:

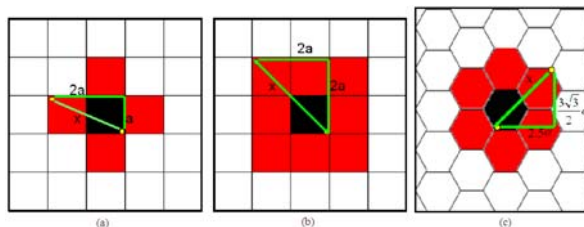


شکل(۲): توری فرضی منظم با سلول‌های مربع و عضلانی بر روی محیط شبکه



شکل(۳): همسایگی برای توری با سلول‌های مربع و عضلانی (a) SC4 (b) SC8 (c) HC

از آن جا که تمامی گره‌های دو سلول همسایه باید در محدوده انتقال یکدیگر قرار داشته باشند، طول ضلع سلول‌ها باید به گونه‌ای انتخاب شود که این مهم قابل دستیابی باشد. به عبارت دیگر، طول ضلع سلول‌ها باید به گونه‌ای انتخاب شود که حداکثر فاصله‌ی موجود مابین دو گره در دو سلول همسایه برابر R (محدوده انتقال هر گره) شود. شکل ۴ نشان دهنده حداکثر فاصله دو گره در دو سلول همسایه برای توری‌های مختلف است. با توجه به این شکل، طول ضلع سلول (a) با توجه به محدوده انتقال هر گره، توسط رابطه‌های (۱۸) تا (۲۰) محاسبه می‌شود. رابطه (۱۸) مقدار a را برای توری SC4، رابطه (۱۹) مقدار a را برای توری SC8 و رابطه (۲۰) مقدار a را برای توری HC محاسبه می‌کند. شماره‌گذاری هر سلول در توری HC در شکل ۵ دیده می‌شود.



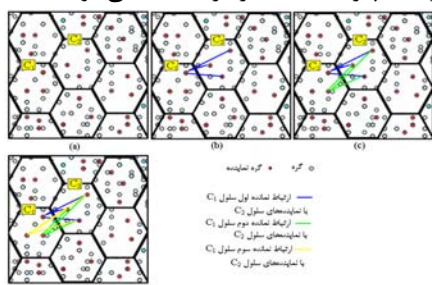
شکل(۴): حداکثر فاصله دو گره در همسایگی مربع و عضلانی

$$R^2 = (2a)^2 + a^2 \Rightarrow a_{SC4} = \frac{R}{\sqrt{5}} \quad (18)$$

$$R^2 = (2a)^2 + (2a)^2 \Rightarrow a_{SC8} = \frac{R}{\sqrt{8}} \quad (19)$$

۲-۱- انتشار رخدادها

هرگاه رخدادی که مورد علاقه گره مرکزی می‌باشد در سلول C_1 حس شد، نماینده‌های آن سلول اقدام به ایجاد بسته داده می‌کنند. دو نماینده اصلی با توجه به جدول مسیریابی، سلول همسایه C_2 با خطای کمتر را انتخاب کرده (به منظور اولین انتخاب که تعداد خطاهای سلول‌ها برابر صفر است، سلول با شماره کوچکتر انتخاب می‌شود)، بسته داده را به آن می‌فرستند. گره‌های نماینده سلول C_2 ، دو بسته را دریافت می‌کنند. اگر دو بسته یکسان بود، بسته داده تأیید شده و پیغام Ack توسط نماینده اول این سلول به سلول C_1 ارسال می‌شود. در غیر این صورت، این نماینده‌ها منتظر دریافت بسته سوم از سلول C_1 می‌مانند. نماینده جایگزین در سلول C_1 به اندازه مدت زمان tw_1 منتظر دریافت بسته Ack می‌ماند. اگر در این مدت بسته Ack دریافت نشد، این نماینده نیز بسته داده خود را به سلول C_2 می‌فرستد. با دریافت بسته سوم در سلول C_2 ، اگر دو بسته از سه بسته ارسالی یکسان بودند بسته داده تأیید می‌شود. بسته Ack شامل چهار فیلد MN ، S_1 ، S_2 و S_3 است. MN میزان ارسال اشتباه سلول و S_i ‌ها نشان‌دهنده عملکرد نماینده نام سلول هستند. S_1 در صورت ارسال اشتباه نماینده نام مقدار منفی یک، ارسال درست نماینده نام مقدار یک و در صورتی که نماینده نام در ارسال بسته داده شرکتی نداشته باشد مقدار صفر می‌گیرد. نماینده اول سلول C_2 فیلدهای بسته Ack را با توجه به عملکرد هر یک از نماینده‌های سلول C_1 مقارن می‌کند و آن را برای سلول C_1 ارسال می‌کند. در صورت تأیید بسته، این روند تا رسیدن بسته داده به گره مرکزی ادامه می‌یابد. در مواقعی که دو بسته ارسالی اول یکسان باشند (دو نماینده اول اطلاعات را بدون وقوع خطا انتقال داده باشند)، فیلد چهارم بسته Ack (S_3) مقدار ۰ می‌گیرد. جریان ارسال بسته‌های داده بین دو سلول C_1 و C_2 در شکل ۶ دیده می‌شود. لازم به ذکر است که به خاطر کم بودن اندازه بسته Ack، فرض می‌شود برای ارسال و دریافت این نوع بسته، گره‌ها دچار خطای ناسازگاری داده نمی‌شوند.



شکل (۶): روند ارسال بسته‌های داده از سلول C_1 به سلول C_2

گره‌هایی که بسته Ack را دریافت می‌کنند با توجه به اطلاعات همراه بسته Ack جدول مسیریابی و اتوماتای یادگیر خود را به‌روز می‌کنند. هر گره احتمال عمل متناظر با انتخاب نماینده‌ای را که

• اگر فیلد تعداد گام بسته Flood دریافتی از فیلد تعداد گام رکوردی از جدول مسیریابی کمتر باشد، تمامی رکوردهای جدول پاک می‌شوند و یک رکورد جدید برای بسته Flood دریافتی جایگزین آنها می‌شود. گره مقادیر فیلدهای بسته Flood را به‌روز می‌کند و سپس آن را در شبکه منتشر می‌کند.

• اگر فیلد تعداد گام بسته Flood با فیلد تعداد گام رکوردی از جدول مسیریابی برابر باشد، اطلاعات بسته Flood به عنوان یک رکورد جدید در جدول افزوده می‌شود.

• در غیر این صورت بسته دور انداخته می‌شود. در ادامه هر گره (نماینده اول سلول‌ها) فیلد تعداد خطای سلول هر رکورد از جدول مسیریابی‌اش را برابر صفر قرار می‌دهد.

هر گره مجهز به یک اتوماتای یادگیر با تعداد عمل‌های برابر با تعداد گره‌های داخل سلول خود است. انتخاب یک عمل توسط اتوماتای یادگیر به معنای انتخاب یک گره از گره‌های سلول به عنوان نماینده سلول است. بردار احتمال انتخاب عمل‌های اتوماتای یادگیر هر گره طبق رابطه (۲۱) مقداردهی می‌شود. در این رابطه n تعداد گره‌های سلول و $EnergyLevel_i$ سطح انرژی گره نام سلول می‌باشد. احتمال متناسب شده به گره‌ها یک احتمال اولیه است که در طول حیات شبکه تغییر می‌کند و به سمتی می‌رود که گره‌های با قابلیت اطمینان بالاتر، با احتمال بیشتری به عنوان نماینده سلول انتخاب شوند. در واقع با استفاده از اتوماتای یادگیر، سلول یاد می‌گیرد که گره‌هایی را به عنوان نماینده انتخاب کند که قابلیت اطمینان بالاتری داشته باشند تا قابلیت اطمینان الگوریتم مسیریابی افزایش یابد. با توجه به لزوم داشتن جدول مسیریابی و اتوماتای یادگیر یکسان برای تمامی گره‌های یک سلول، نماینده اول هر سلول پس از پایان مرحله ساخت جدول مسیریابی، جدول مسیریابی و اتوماتای یادگیر خود را به گره‌های دیگر سلول ارسال می‌کند.

$$\forall i \quad i \leq n \quad P_i = \frac{EnergyLevel_i}{\sum_{i=1}^n EnergyLevel_i} \quad (21)$$

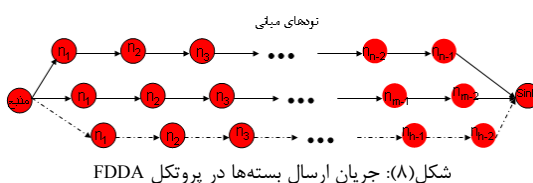
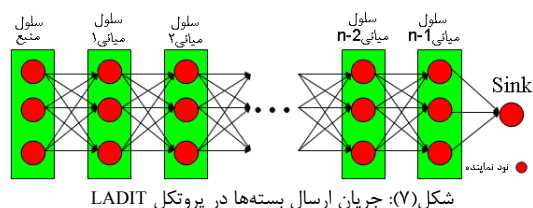
در پایان این مرحله هر گره دارای یک جدول مسیریابی و اتوماتای یادگیر برای انتخاب نماینده سلول است. با توجه به نحوه ارسال بسته Flood و ارسال جدول مسیریابی و اتوماتای یادگیر توسط نماینده اول هر سلول، گره‌های هر سلول دارای جدول مسیریابی و اتوماتای یادگیر یکسان هستند. پس از این مرحله، دو گره دیگر (به جز گره نماینده فعلی) که دارای بیشترین انرژی در سطح سلول هستند به عنوان نماینده‌های دوم (نماینده اصلی) و سوم (نماینده جایگزین) انتخاب می‌شوند. به این ترتیب هر سلول دارای دو نماینده اصلی و یک نماینده جایگزین می‌شود.

عنوان نماینده سلول می‌شناسد. راه‌حل دوم از الگوریتم یادگیری خطی ساده، راه‌حل سوم از الگوریتم یادگیری خطی ترتیب‌دار و راه‌حل چهارم از الگوریتم یادگیری خطی بدون ترتیب استفاده می‌کند. در ادامه مقاله، پروتکلی که از راه‌حل دوم، سوم و یا چهارم استفاده می‌کند به ترتیب با $LADIT_{S3SALA}$ ، $LADIT_{A3SALA}$ و $LADIT_{UA3SALA}$ نشان داده شده است.

به منظور مشخص‌تر شدن برتری‌های استفاده از چهار راه‌حل فوق، راهکار دیگری نیز برای انتخاب نماینده‌های سلول در نظر گرفته شده است که در آن تمامی نماینده‌ها به صورت کاملاً تصادفی انتخاب می‌شوند. پروتکلی که از این راه‌حل استفاده می‌کند در ادامه مقاله با $LADIT_{RAN}$ نشان داده شده است.

قابلیت اطمینان

در این قسمت قابلیت اطمینان مکانیزم‌های (و نه خود پروتکل‌ها) مورد استفاده در پروتکل‌های $LADIT$ و $FDDA$ مورد بررسی قرار می‌گیرد. با توجه به نحوه عملکرد هر یک از پروتکل‌ها، جریان ارسال اطلاعات برای دو پروتکل متفاوت است. جریان ارسال بسته‌های داده برای پروتکل‌های $LADIT$ و $FDDA$ به ترتیب در شکل‌های ۷ و ۸ دیده می‌شود.



احتمال درست رسیدن بسته داده به مقصد (قابلیت اطمینان بسته داده) برای پروتکل $LADIT$ طبق رابطه (۲۲) محاسبه می‌شود. در این رابطه، R_{PKT} قابلیت اطمینان بسته داده و احتمال رسیدن بسته داده درست به گره مرکزی، R احتمال کارکرد درست گره نماینده و n تعداد گام بین منبع و گره مرکزی می‌باشد.

$$R_{PKT} = (3R^2 - 2R^3)^n \quad (22)$$

قابلیت اطمینان مسیر و احتمال رسیدن بسته داده درست به مقصد برای پروتکل $FDDA$ طبق رابطه (۲۳) محاسبه می‌شود (با فرض برابر بودن طول مسیرهای مختلف). در این رابطه پارامتر n نشان دهنده طول مسیرهای مختلفی است که بسته داده از

داده درست ارسال کرده ($S_i=1$) با توجه به الگوریتم یادگیری افزایش و احتمال عمل متناظر با انتخاب نماینده‌ای را که عملکرد درستی نداشته ($S_i=-1$) کاهش می‌دهد. نوع الگوریتم یادگیری به کار رفته در ادامه در بخش انتخاب نماینده‌ها توضیح داده می‌شود. تمامی گره‌های موجود در هر سلول، نمایندگان سلول را با استفاده از اتوماتای خود انتخاب می‌کنند. با توجه به آنکه تمامی گره‌ها از یک منطق تولید اعداد تصادفی برای انتخاب عمل اتوماتا استفاده می‌کنند و با فرض مقدار راه‌انداز یکسان برای الگوریتم تولید اعداد تصادفی در تمامی گره‌های یک سلول، نمایندگان انتخابی توسط تمامی گره‌ها یکسان خواهند بود. هرگاه تعداد ارسال‌های اشتباه یک گره بیشتر از حد آستانه مشخص ($MaxMistake$) شد، فرض می‌شود که آن گره دارای خطای دائم است. گره مورد نظر غیر فعال شده و عمل انتخاب آن از اتوماتاهای یادگیر تمامی گره‌های آن سلول حذف می‌شود. پارامتر $MaxMistake$ یک پارامتر تنظیم‌پذیر است و نشان دهنده حداکثر تعداد خطایی است که یک گره می‌تواند انجام دهد.

۳-۲-۱- انتخاب نماینده‌ها

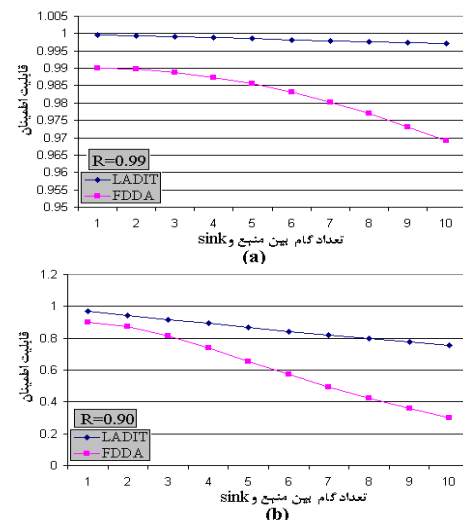
به منظور انتخاب نماینده‌های هر سلول به کمک اتوماتای یادگیر، راهکارهای گوناگونی وجود دارد. راه‌حل اول استفاده از اتوماتای یادگیر با مجموعه عمل‌های متغیر است. در این راه‌حل هر گره سه اتوماتا برای انتخاب نماینده‌های سلول در نظر می‌گیرد که هر اتوماتا مربوط به انتخاب یک نماینده است. در هر گره هنگام انتخاب نماینده سلول، در اتوماتای اول یک عمل (گره) انتخاب می‌شود. در این حالت، مجموعه عمل‌های قابل اعمال توسط اتوماتای دوم همه عمل‌ها به جز عمل انتخابی اتوماتای اول است. اتوماتای دوم از بین مجموعه عمل‌های قابل اعمال خود یک عمل (گره) را برمی‌گزیند. مجموعه عمل‌های قابل اعمال اتوماتای سوم همه عمل‌ها به جز دو عمل انتخاب شده توسط اتوماتاهای اول و دوم می‌باشد. اتوماتای سوم نیز از بین مجموعه عمل‌های قابل اعمال خود یک عمل (گره) را انتخاب می‌کند. اگر گره مورد نظر برابر یکی از عمل‌های (گره‌های) انتخابی بود گره خود را به عنوان نماینده سلول می‌شناسد. در ادامه مقاله، پروتکلی که از این راهکار استفاده می‌کند با $LADIT_{3LA}$ نشان داده شده است.

راه‌حل‌های دوم تا چهارم از اتوماتای یادگیر $KSALA$ استفاده می‌کنند. پارامتر k در این اتوماتای یادگیر برابر ۳ می‌باشد. این بدان مفهوم است که اتوماتای یادگیر ابتدا ۳ عمل را انتخاب کرده و در محیط اعمال می‌کند و سپس نتایج حاصل از این سه عمل را از محیط دریافت می‌دارد. در هر گره هنگام انتخاب نماینده سلول، اتوماتای یادگیر سه عمل را انتخاب می‌کند. اگر گره مورد نظر برابر یکی از عمل‌های (گره‌های) انتخابی بود گره خود را به

آنها به گره مرکزی می‌رسد، R_{PKT} قابلیت اطمینان بسته داده و احتمال رسیدن بسته داده درست به گره مرکزی و R احتمال کارکرد درست گره است.

$$R_{PKT} = R(3R^{2(n-1)} - 2R^{3(n-1)}) \quad (23)$$

به منظور ارزیابی قابلیت اطمینان بسته داده برای دو پروتکل LADIT و FDFA، قابلیت اطمینان دو پروتکل با استفاده از روابط (۲۲) و (۲۳) محاسبه گردید دو مقدار ۰/۹۹ و ۰/۹۰ برای احتمال کارکرد درست هر گره در نظر گرفته شده است. برای هر یک از این احتمالات مقدار n (تعداد گام بین منبع و گره مرکزی) از یک تا ده در نظر گرفته شده است. نتایج محاسبات در شکل ۹ دیده می‌شود. همانگونه که این شکل نشان می‌دهد، قابلیت اطمینان پروتکل LADIT بسیار بیشتر از پروتکل FDFA است و هر چه قابلیت اطمینان یک گره (R) کمتر باشد اختلاف قابلیت اطمینان دو پروتکل بیشتر می‌شود.



شکل (۹): قابلیت اطمینان با احتمال کارکرد درست گره به میزان ۰/۹۹ و ۰/۹۰ برای دو پروتکل LADIT و FDFA.

نتایج شبیه‌سازی‌ها

به منظور ارزیابی، پروتکل LADIT و پروتکل FDFA در نرم‌افزار شبیه‌ساز ns2 شبیه‌سازی و نتایج به دست آمده با یکدیگر مقایسه شده‌اند [19]. در شبیه‌سازی، از IEEE 802.11 در مد DCF برای لایه کنترل دستیابی رسانه استفاده شده است. تعداد ۱۰۰۰ گره در منطقه‌ای به مساحت ۵۰۰ * ۵۰۰ مترمربع توزیع شده‌اند. مکان گره مرکزی به طور تصادفی در این محیط تعیین می‌شود و گره‌ها همه با هم یکسان و دارای محدوده انتقال ثابت ۱۰۰m هستند. مصرف انرژی برای حالت ارسال و دریافت بسته داده به ترتیب برابر ۶۶۰mJ و ۳۹۵mJ است [24]. زمان شبیه‌سازی ۱۰۰ ثانیه در نظر گرفته شده است. یک توری مجازی با سلول‌های عضلانی در سطح شبکه در نظر گرفته می‌شود که طول هر ضلع

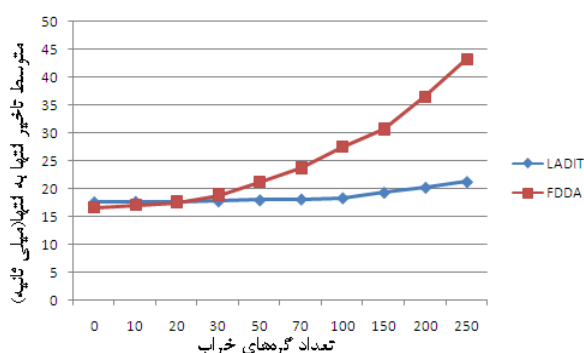
سلول با توجه به رابطه (۲۰) برابر $2Y/Ym$ می‌باشد. MaxMistake نیز برابر ۵ در نظر گرفته شده است. انواع خطاهای ناسازگاری داده‌ای که به شبکه تزریق می‌شود، خطاهای ناسازگاری پایدار و خطای ناسازگاری گذرا می‌باشند. در حالت خطای پایدار، گره خراب همچنان خراب می‌ماند، اما در حالت خطای گذرا، گره خراب تنها برای لحظاتی (مدت زمان ارسال حداکثر ۵ بسته) خراب است. تعداد گره‌های خراب برابر با ۰، ۱۰، ۲۰، ۳۰، ۵۰، ۷۰، ۱۰۰، ۱۵۰، ۲۰۰ و ۲۵۰ در نظر گرفته شده است. نیمی از گره‌های خراب دارای خطای پایدار و نیمی دیگر دارای خطای گذرا هستند. با توجه به محیط شبیه‌سازی، دو پروتکل LADIT و FDFA از سه جنبه متوسط نسبت تحویل داده درست به داده ارسالی، متوسط تأخیر انتها به انتها و متوسط مصرف انرژی مورد بررسی قرار گرفته‌اند. به منظور ارزیابی راه‌حل‌های مختلف انتخاب نماینده و شکل توری در پروتکل LADIT، پروتکل LADIT با راه‌حل‌های مختلف در نرم‌افزار شبیه‌ساز ns2 شبیه‌سازی و نتایج به دست آمده از لحاظ متوسط نسبت تحویل داده درست به داده ارسالی، متوسط تأخیر انتها به انتها و متوسط مصرف انرژی با یکدیگر مقایسه شده‌اند. در این شبیه‌سازی تعداد گره‌های خراب از ۰ تا ۵۰۰ گره در نظر گرفته شده است تا تفاوت راه‌حل‌های مختلف انتخاب نماینده بهتر دیده شود.

در آزمایش‌های اول تا سوم، از نسخه LADIT_{3LA} به منظور مقایسه با پروتکل FDFA استفاده شده است. نتایج آزمایش‌ها برتری این نوع پروتکل را از جهات مختلف نسبت به پروتکل FDFA نشان می‌دهد. در آزمایش‌های چهارم تا ششم، این نسخه از LADIT با نسخه‌های دیگر آن مقایسه شده است. این مقایسه‌ها نشان می‌دهد که LADIT_{3LA} در مقایسه با نسخه‌های دیگر LADIT عملکرد ضعیفتری دارد. بنابراین، نتایج آزمایش‌ها نشان می‌دهد که حتی ضعیف‌ترین نسخه از پروتکل LADIT نیز نسبت به پروتکل FDFA عملکرد بهتری را از خود نشان می‌دهد. در این آزمایش‌ها، از توری با سلول‌های عضلانی در پروتکل‌ها استفاده شده است. در آزمایش‌های هفتم تا نهم، تاثیر شکل سلول بر روی کارایی شبکه مورد بررسی قرار گرفته است. در این آزمایش‌ها، در پروتکل‌های LADIT استفاده کننده از توری HC (LADIT_{HC})، توری SC4 (LADIT_{SC4}) و توری SC8 (LADIT_{SC8}) از نسخه LADIT_{A3SALA} استفاده شده است.

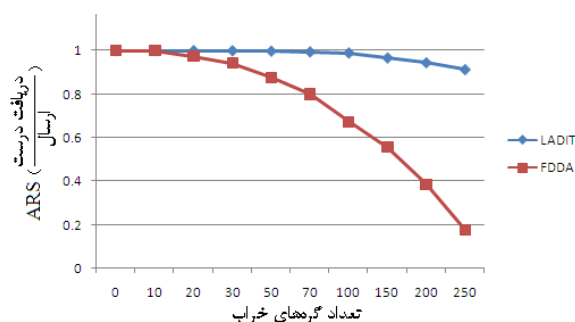
آزمایش اول: در این آزمایش متوسط نسبت تحویل داده درست به داده ارسالی (ARS^1) برای دو پروتکل LADIT و FDFA برای تعداد مختلف گره‌های خراب مورد بررسی قرار گرفته است. مقدار ARS با استفاده از رابطه (۲۴) محاسبه می‌شود. در این رابطه R_{td} متوسط تعداد دریافت بسته‌های داده درست و S_d متوسط تعداد بسته‌های داده ارسالی است.

$$ARS = \frac{Rtd}{Sd}$$

آمده به صورت محلی مرتفع می‌شود و لزومی به ارسال بسته در کل مسیر نمی‌باشد. این کار تأثیر کمی روی تأخیر انتها به انتها بسته داده خواهد داشت. با افزایش تعداد گره‌های خراب، متوسط تأخیر انتها به انتها بسته‌های داده در هر دو پروتکل مورد بررسی افزایش می‌یابد. علت این امر افزایش لزوم ایجاد مسیر سوم می‌باشد. برای داده‌هایی که به دلیل بروز خطا به گره مرکزی نمی‌رسند، تأخیر حداکثری برابر با ۵۰ میلی ثانیه در نظر گرفته شده است. در نظر گرفتن یک مقدار حداکثر برای تأخیر به منظور قابل مقایسه کردن پروتکل‌های شبیه‌سازی شده است، زیرا پروتکل FDDA در صورت بروز خطاهای فراوان، توانایی ارسال بسیاری از بسته‌ها را به گره مرکزی از دست می‌دهد و لذا نمی‌توان میزان تأخیر بسته‌ها را در آن با پروتکل ارائه شده در این مقاله مقایسه نمود. با توجه به این مطلب، از دیگر علت‌های افزایش تأخیر انتها به انتها با افزایش تعداد گره‌های خراب، کاهش نرخ تأیید بسته‌های داده و در نتیجه در نظر گرفتن بالاترین تأخیر (۵۰ میلی ثانیه) برای تعداد بیشتری از بسته‌هاست.



شکل (۱۱): متوسط تأخیر انتها به انتهای داده‌های ارسالی برای دو پروتکل FDDA و LADIT



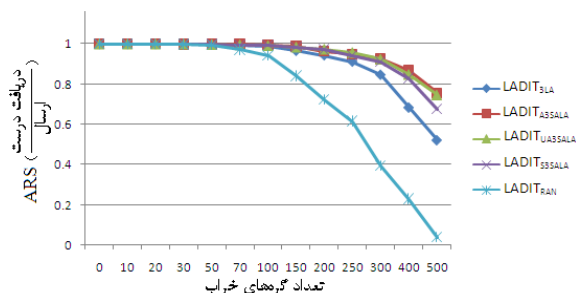
شکل (۱۰): ARS برای دو پروتکل LADIT و FDDA

آزمایش سوم: هدف از این آزمایش بررسی متوسط تأخیر انتها به انتها بسته‌های داده برای دو پروتکل LADIT و FDDA می‌باشد. این پارامتر نشان دهنده اختلاف زمانی مابین زمان تولید بسته و زمان دریافت آن توسط گره مرکزی است. در شکل ۱۱ میانگین تأخیر انتها به انتها بسته‌های ارسالی برای پروتکل‌های LADIT و FDDA برای تعداد مختلف گره خراب دیده می‌شود. همانگونه که دیده می‌شود، در پروتکل FDDA، بسته‌های داده، متوسط تأخیر انتها به انتها بیشتری را نسبت به پروتکل LADIT دارند. علت این امر آن است که در پروتکل FDDA وقتی که دو داده دریافت شده در گره مرکزی یکسان نباشند، مسیر سوم ایجاد می‌شود و داده مورد نظر مجدداً از سه مسیر جدا برای گره مرکزی ارسال می‌شود. این روال باعث افزایش چشم‌گیر تأخیر انتها به انتها برای بسته داده خواهد شد. این در حالی است که در پروتکل LADIT هنگام دریافت داده‌های ناهمسان توسط گره‌های نماینده، تنها لازم است که گره نماینده سوم سلول فرستنده داده را مجدداً ارسال کند. در حقیقت مشکل به وجود

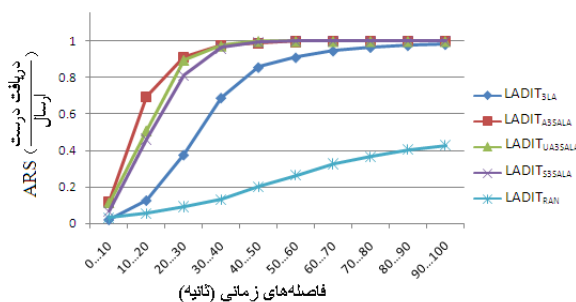
$$e_{id} = \frac{c_e}{n_{id}} \quad (25)$$

نتایج شبیه‌سازی عملکرد بهتر پروتکل LADIT را از لحاظ متوسط مصرف انرژی نشان می‌دهد. در شکل ۱۲ نشان داده شده است که متوسط مصرف انرژی به ازای دریافت هر بسته درست برای دو پروتکل مورد بررسی به چه میزان است. همان گونه که دیده می‌شود، برای پروتکل LADIT متوسط مصرف انرژی به ازای دریافت هر بسته درست همواره کمتر از پروتکل FDDA

آزمایش دوم: هدف از این آزمایش بررسی متوسط تأخیر انتها به انتها بسته‌های داده برای دو پروتکل LADIT و FDDA می‌باشد. این پارامتر نشان دهنده اختلاف زمانی مابین زمان تولید بسته و زمان دریافت آن توسط گره مرکزی است. در شکل ۱۱ میانگین تأخیر انتها به انتها بسته‌های ارسالی برای پروتکل‌های LADIT و FDDA برای تعداد مختلف گره خراب دیده می‌شود. همانگونه که دیده می‌شود، در پروتکل FDDA، بسته‌های داده، متوسط تأخیر انتها به انتها بیشتری را نسبت به پروتکل LADIT دارند. علت این امر آن است که در پروتکل FDDA وقتی که دو داده دریافت شده در گره مرکزی یکسان نباشند، مسیر سوم ایجاد می‌شود و داده مورد نظر مجدداً از سه مسیر جدا برای گره مرکزی ارسال می‌شود. این روال باعث افزایش چشم‌گیر تأخیر انتها به انتها برای بسته داده خواهد شد. این در حالی است که در پروتکل LADIT هنگام دریافت داده‌های ناهمسان توسط گره‌های نماینده، تنها لازم است که گره نماینده سوم سلول فرستنده داده را مجدداً ارسال کند. در حقیقت مشکل به وجود

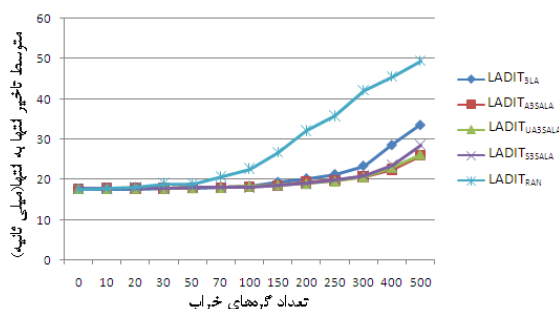


شکل (۱۳): ARS برای مکانیزم‌های مختلف انتخاب نماینده در پروتکل LADIT



شکل (۱۴): ARS برای بازه‌های زمانی مختلف برای مکانیزم‌های مختلف انتخاب نماینده در پروتکل LADIT با ۴۰۰ گره خراب

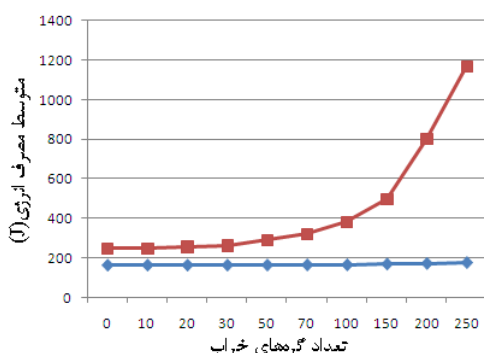
آزمایش پنجم: هدف از این آزمایش که نتایج آن در شکل ۱۵ دیده می‌شود، بررسی متوسط تأخیر انتها به انتهای بسته‌های داده برای مکانیزم‌های مختلف انتخاب نماینده در پروتکل LADIT می‌باشد. دیده می‌شود که هر پروتکلی که متوسط نسبت تحویل داده درست به داده‌ی ارسالی بیشتری دارد، دارای متوسط تأخیر انتها به انتهای کمتری است. علت این امر آن است برای پروتکل‌هایی با متوسط نسبت تحویل داده درست بالاتر، تعداد داده‌هایی که نمی‌توانند به گره مرکزی برسند (داده‌هایی که تأخیر ۵۰ میلی‌ثانیه برای آنها در نظر گرفته می‌شود) کمتر است. علاوه بر این، لزوم استفاده از مسیر سوم در بین دو سلول در ارتباطات آنها نیز کمتر است.



شکل (۱۵): تأخیر انتها به انتهای داده‌های ارسالی برای مکانیزم‌های مختلف انتخاب نماینده پروتکل LADIT

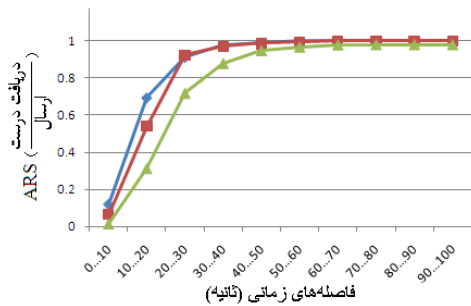
آزمایش ششم: هدف از این آزمایش بررسی متوسط مصرف انرژی برای مکانیزم‌های مختلف انتخاب نماینده در پروتکل LADIT می‌باشد. در شکل ۱۶ نشان داده شده است که متوسط

است. در پروتکل FDFA با افزایش تعداد گره‌های خراب، متوسط مصرف انرژی به ازاء دریافت هر بسته درست بیشتر می‌شود که به خاطر این است که گره مرکزی داده‌های درست کمتری را دریافت می‌کند. در پروتکل LADIT هرچند باید با افزایش تعداد گره‌های خراب میزان مصرف انرژی هر گره به خاطر استفاده بیشتر از مسیرهای سوم افزایش یابد، اما به خاطر اینکه بسته‌های داده بیشتری مورد تأیید قرار نمی‌گیرند، میزان مصرف انرژی تقریباً یکسان است. در پروتکل LADIT هرگاه نماینده‌های سلولی داده‌ای را تأیید نکردند، دیگر آن داده را به سمت گره مرکزی ارسال نمی‌کنند و در نتیجه مصرف انرژی کاهش می‌یابد.



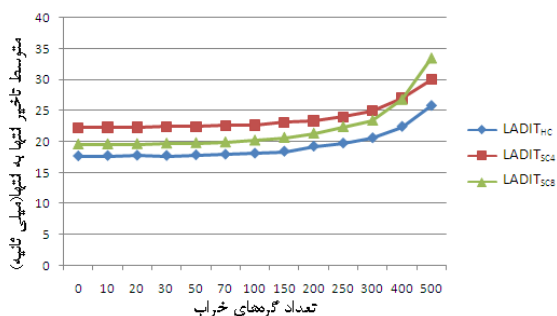
شکل (۱۶): متوسط مصرف انرژی به ازاء دریافت هر داده درست برای دو پروتکل FDFA و LADIT

آزمایش چهارم: در این آزمایش که نتایج آن در شکل ۱۳ دیده می‌شود، ARS برای مکانیزم‌های مختلف انتخاب نماینده در پروتکل LADIT مورد بررسی قرار گرفته است. همان گونه که دیده می‌شود، پروتکل LADIT_{A3SALA} همواره ARS بالاتری نسبت به پروتکل‌های دیگر دارد. علت این امر آن است که در پروتکل LADIT_{A3SALA} هر گره تنها از یک اتوماتای یادگیر استفاده می‌کند و همچنین الگوریتم یادگیری به کار رفته در اتوماتای یادگیر به گونه‌ای است که عملکرد نماینده دوم بر روی نماینده اول و عملکرد نماینده سوم بر روی نماینده اول و دوم تأثیری ندارد و در نتیجه زمان یادگیری کوتاه‌تر می‌شود. البته ARS برای پروتکل LADIT_{UA3SALA} بسیار نزدیک به پروتکل LADIT_{A3SALA} است که به خاطر عدم تأثیر عملکردهای نماینده‌ها بر یکدیگر است. در پروتکل LADIT_{3LA} به خاطر استفاده از ۳ اتوماتای یادگیر، زمان طولانی‌تری برای یادگیری مصرف می‌شود. شکل ۱۴ نشان دهنده ARS برای مکانیزم‌های مختلف انتخاب نماینده در پروتکل LADIT با تعداد گره خراب ۴۰۰ برای بازه‌های زمانی مختلف است. دیده می‌شود که با افزایش زمان، اتوماتاها یاد می‌گیرند که گره سالم را برای نماینده شدن انتخاب کنند.



شکل (۱۸): ARS برای بازه‌های زمانی مختلف برای شکل‌های مختلف سلول در پروتکل LADIT با ۴۰۰ گره خراب

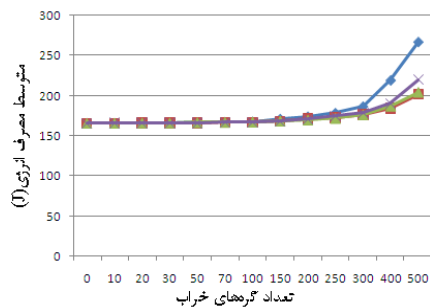
آزمایش هشتم: هدف از این آزمایش بررسی متوسط تأخیر انتها به انتهای بسته‌های داده برای شکل‌های مختلف سلول در پروتکل LADIT می‌باشد. در شکل ۱۹ متوسط تأخیر انتها به انتهای بسته‌های ارسالی برای شکل‌های مختلف سلول در پروتکل LADIT با تعداد گره‌های خراب مختلف دیده می‌شود. همانگونه که دیده می‌شود، پروتکل LADIT_{HC} کمترین مقدار میانگین تأخیر انتها به انتهای داده را داراست که به خاطر کم بودن تعداد گام بین منبع و گره مرکزی است. علت کم بودن تعداد گام بین منبع و گره مرکزی در پروتکل LADIT_{HC} استفاده از ارتباطات مورب و پوشش محدوده توزیع گره‌ها با سلول‌های کمتر است.



شکل (۱۹): تأخیر انتها به انتهای داده‌های ارسالی برای شکل‌های مختلف سلول در پروتکل LADIT

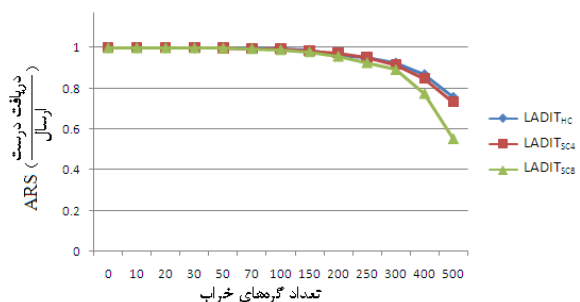
آزمایش نهم: هدف از این آزمایش بررسی متوسط مصرف انرژی برای شکل‌های مختلف سلول در پروتکل LADIT می‌باشد. متوسط مصرف انرژی به ازای دریافت هر بسته درست برای شکل‌های مختلف سلول در پروتکل LADIT در شکل ۲۰ نشان داده شده است. همانگونه که دیده می‌شود، پروتکل LADIT_{SC8} دارای کمترین مقدار مصرف انرژی است که به علت استفاده از ارتباطات قطری و در نتیجه تعداد گام کمتر بین منبع و گره مرکزی است. به علاوه با توجه به تعداد زیادتر سلول در این نوع پروتکل نسبت به دو نوع دیگر، متوسط تعداد گره در هر سلول شبکه نیز کمتر است و از آنجا که در ارتباط دو سلول تمامی گره‌های دو سلول داده‌ها را دریافت می‌کنند، کم بودن تعداد گره در سلول باعث مصرف انرژی کمتری برای شبکه می‌شود. کم

مصرف انرژی به ازای دریافت هر بسته درست برای مکانیزم‌های مختلف انتخاب نماینده در پروتکل LADIT به چه میزان است. همانگونه که دیده می‌شود، هر پروتکلی که ARS بیشتری دارد، متوسط مصرف انرژی آن به ازای دریافت هر بسته درست کمتر می‌باشد. به منظور نمایش بهتر مکانیزم‌های مختلف انتخابی در پروتکل LADIT، و اختلاف زیاد پروتکل LADIT_{RAN} در مصرف انرژی با دیگر پروتکل‌ها، نتایج پروتکل LADIT_{RAN} در شکل ۱۶ در نظر گرفته نشده است.



شکل (۱۶): متوسط مصرف انرژی به ازای دریافت هر داده درست برای مکانیزم‌های مختلف انتخاب نماینده در پروتکل LADIT (بدون LADIT_{RAN})

آزمایش هفتم: در این آزمایش، ARS برای شکل‌های مختلف سلول توری در پروتکل LADIT مورد بررسی قرار گرفته است. با توجه به آزمایش‌های قبل می‌توان نتیجه گرفت که نسخه LADIT_{A3SALA} بهتر از نسخه‌های دیگر پروتکل LADIT است. به همین منظور در آزمایش‌های هفتم تا نهم، تنها این نسخه از پروتکل مورد بررسی قرار گرفته است. ARS برای شکل‌های مختلف سلول در پروتکل LADIT در شکل ۱۷ دیده می‌شود. همانگونه که دیده می‌شود، پروتکل LADIT_{HC} همواره ARS بالاتری نسبت به پروتکل‌های دیگر دارد. شکل ۱۸ نشان دهنده ARS برای شکل‌های مختلف سلول در پروتکل LADIT با تعداد گره خراب ۴۰۰ برای بازه‌های زمانی مختلف است. دیده می‌شود که با افزایش زمان، اتوماتاها یاد می‌گیرند که گره سالم را برای نماینده شدن انتخاب کنند.

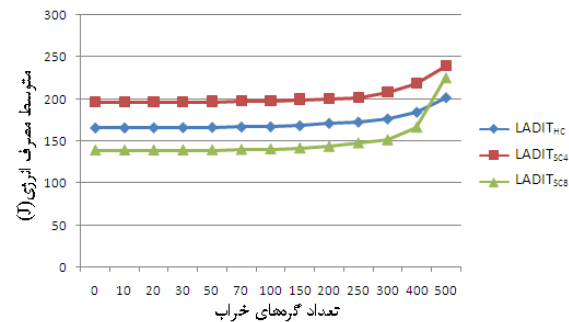


شکل (۱۷): ARS برای شکل‌های مختلف سلول در پروتکل LADIT

Networks, Volume 50, Issue 9, Pages 1247-1260, 20 June 2006.

- [5] M. Ding, D. Chen, A. Thaeler and X. Cheng, "Fault-Tolerant Target Detection in Sensor Networks", IEEE WCNC 2005, Vol. 4, pp. 2362-2368, 13-17 March 2005.
- [6] M. Ding, D. Chen, K. Xing and X. Cheng, "Localized Fault-Tolerant Event Boundary Detection in Sensor Networks", Proceedings of IEEE INFOCOM 2005, Miami, March 2005.
- [7] X. Luo, M. Dong and Y. Huang, "On Distributed Fault-Tolerant Detection in Wireless Sensor Networks", IEEE Transactions on Computers, vol. 55, no. 1, pp. 58-70, January 2006.
- [8] X.M. Huang, J. Deng and J. Ma, "Fault Tolerant Routing For Wireless Sensor Grid Networks", In Proceedings of IEEE Sensors Applications Symposium, Houston, TX, February 2006.
- [9] T. K. K. Ishida and Y. Kakuda, "A routing protocol for finding two node-disjoint paths in computer networks", in: International Conference on Network Protocols, 1992, pp. 340-347.
- [10] K. F. Ssu, C. H. Chou and C. S. Chiu, "Dynamic route switching protocol in mobile ad hoc networks", International Journal of Ad Hoc and Ubiquitous Computing, in press.
- [11] P. Djukic and S. Valaee, "Maximum network lifetime in fault tolerant sensor networks", in IEEE Globecom, 2005.
- [12] P. Djukic and S. Valaee, "Minimum energy fault tolerant sensor networks", in IEEE Globecom Workshops, pp. 22-26, 2004.
- [13] K. Sha, J. Du and W. Shi, "WEAR: a balanced, fault-tolerant, energy-aware routing protocol in WSNs", International Journal of Sensor Networks 2006 - Vol. 1, No.3/4 pp. 156 - 168.
- [14] G. Khanna, S. Bagchi and Y. S. Wu, "Fault tolerant energy aware data dissemination protocol in sensor networks", Dependable Systems and Networks (DSN'04), 2004 International Conference on, pp.795-804. 1 July 2004.
- [15] A. Boukerche, R. W. N. Pazzi and R. B. d. Araujo, "A fast and reliable protocol for wireless sensor networks in critical conditions monitoring applications", MSWiM: 157-164, 2004.
- [16] S. Chessa and P. Santi, "Comparison-based system-level fault diagnosis in ad hoc networks", in: Proceedings of 20th IEEE Symposium on Reliable Distributed Systems, 2001, pp. 257-266.
- [17] J. Seguin, G. Sergeant and P. Wilms, "A majority consensus algorithm for the consistency of duplicated and distributed information", in: Proceedings of the 1st International Conference on Distributed Computing Systems, 1979, pp. 617-624.
- [18] D. K. Gifford, "Weighted voting for replicated data", in: Proceedings of the Seventh ACM Symposium on Operating Systems Principles, 1979, pp. 150-161.
- [19] The Network Simulator - ns-2. <http://www.isi.edu/nsnam/ns/>.
- [20] K. S. Narendra and M. A. L. Thathachar, "Learning Automata: An Introduction", Prentice Hall, 1989.
- [21] H. Beigy and M. R. Meybodi, "A Mathematical Framework for Cellular Learning Automata", Advanced in Complex Systems, to appear.

بودن میانگین مصرف انرژی در پروتکل $LADIT_{HC}$ نسبت به پروتکل $LADIT_{SC4}$ به خاطر کم بودن تعداد گام بین منبع و گره مرکزی است. علت کم بودن میانگین مصرف انرژی در پروتکل $LADIT_{HC}$ نسبت به پروتکل $LADIT_{SC8}$ در تعداد خطای ۵۰۰ کاهش شدید ARS برای پروتکل $LADIT_{SC8}$ است.



شکل (۲۰): متوسط مصرف انرژی به ازاء دریافت هر داده درست برای شکل‌های مختلف سلول در پروتکل $LADIT$

نتیجه‌گیری

در این مقاله یک پروتکل مسیریابی با نام $LADIT$ به منظور تحمل‌پذیر کردن شبکه در مقابل خطاهای ناسازگاری داده پیشنهاد گردید. این پروتکل از ارتباطات سلول به سلول به جای گره به گره استفاده می‌کند. با استفاده از این نوع ارتباط، در شرایط خطاهای زیاد گره، شبکه تحمل‌پذیری خطای مناسبی را خواهد داشت. در هر سلول گره‌ها با استفاده از اتوماتای یادگیر گره‌هایی با قابلیت اطمینان بیشتر را به عنوان نماینده سلول انتخاب می‌کنند تا ارتباط سلول به سلول را فراهم کنند. پروتکل $LADIT$ در شبیه‌ساز ns2 شبیه‌سازی شده و با پروتکل FDDA که تنها پروتکل موجود در زمینه تحمل‌پذیری خطای ناسازگاری داده می‌باشد مقایسه گردید. نتایج این مقایسه‌ها عملکرد مطلوب پروتکل پیشنهادی را نشان می‌دهد.

مراجع

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "A Survey on Sensor Network", IEEE Communication Magazine, Vol. 40, pp. 102-114, August 2002.
- [2] D. Janakiram, R. Venkateswarlu and S. Nitin, "A survey on programming languages, middleware and applications in wireless sensor networks", in IITM-CSE-DOS-2005-04, 2005.
- [3] D. Estrin, "Embedded Everywhere: A Research Agenda for Network Systems of Embedded Computers", National Academy Press, 2001, Computer Science and Telecommunication Board (CSTB) Report.
- [4] K. F. Ssu, C. H. Chou, H. C. Jiau and W. T. Hu, "Detection and diagnosis of data inconsistency failures in wireless sensor networks", Computer

SMC-17, No. 6, PP. 1095-1100, November/December 1987.

- [24] Intanagonwiwat C., Govindan R., Estrin D., Heidemann J. and Silva F., **“Directed diffusion for wireless sensor networking”**, in: Proceedings of the IEEE/ACM Transactions on Networking (TON), pp. 2-16, 2003.
- [22] M. A. L. Thathachar and P. S. Sastry, **“Varieties of Learning Automata: An Overview”**, IEEE Transaction on Systems, Man, and Cybernetics-Part B: Cybernetics, Vol. 32, No. 6, PP. 711-722, 2002.
- [23] M. A. L. Thathachar and B. R. Harita, **“Learning Automata with Changing Number of Action”**, IEEE Transaction on Systems, Man, and Cybernetics, Vol.
-

¹ Data Inconsistency

² Learning Automata-Based Data Inconsistency Fault Tolerant

³ Majority Voting

⁴ Ssu et al

⁵ Fault Detection and Diagnosis Algorithm

⁶ K Select Action Learning Automata

⁷ Square Cell with 4 neighbor

⁸ Square Cell with 8 neighbor

⁹ Hexagonal Cell

¹⁰ Average Receive true data per Send data