مجله علمی-پژوهشی رایانش نرم و فن‌آوری اطلاعات

جلد ۵، شماره ۲، تابستان سال ۱۳۹۵ شاپا: ۱۰۰۶-۲۳۸۳

# طراحی و بهبود یک الگوریتم یادگیرنده برای جستجوی منابع در شبکه نظیربه‌نظیر بر روی شبکه ادهاک سیار

سمیه خالقی[*1]، کریم فائز[2]

*۱- نویسنده مسئول: دانشجوی ارشد، دانشگاه آزاد اسلامی واحد قزوین، دانشکده مهندسی کامپیوتر و فناوری اطلاعات، قزوین، ایران، SomayeKhaleghi@gmail.com

۲- استاد، دانشکده مهندسی برق دانشگاه صنعتی امیر کبیر، تهران، ایران، kfaez@aut.ac.ir

**چکیده- شبکه نظیربه‌نظیر[1] مجموعه‌ای از گره‌های خودسازمان‌دهنده[2] و یکسان است که با هم همکاری می‌نمایند. این نوع از شبکه‌ها نیاز به هماهنگ‌کننده مرکزی[3] برای ارتباط باهم ندارند. از پیاده سازی شبکه نظیربه‌نظیر بر بستر شبکه ادهاک سیار شبکه نظیربه‌نظیر سیار[4] ایجاد می‌گردد. ساختار توزیع‌شده شبکه‌های نظیربه‌نظیر سیار جستجوی منبع[5] را در آن‌ها دشوار می‌سازد. از سوی دیگر درخواست در این شبکه‌ها معمولا به صورت سیل‌آسا[6] و تصادفی پخش می‌گردد که سبب هدر رفت انرژی گره‌های سیار خواهد شد. لذا طراحی مکانیزمی خودتطبیق که درخواست را آگاهانه در شبکه ارسال نماید، حائز اهمیت می‌باشد، تا بتوان منبع درخواستی را با نرخ موفقیت بیشتر، صرف کمترین انرژی، تحمیل بار کمتر به شبکه کشف کرد. الگوریتم پیشنهادی با هدف بهبود جستجو در شبکه نظیربه‌نظیرسیار با استفاده از یادگیری گره‌ها، ارائه گردیده است. از اتوماتای یادگیر به‌عنوان یک عامل یادگیرنده تقویتی که از طریق تعامل با محیط عملکرد خود را بهبود می‌بخشد، استفاده می‌شود. در این الگوریتم هر گره‌ای که درخواست داشته باشد، پیام جستجو را به تعدادی از همسایگانی که از بقیه مفیدتر هستند، ارسال می‌کند. همچنین منبع درخواستی از طریق بهترین مسیر به گره درخواست‌دهنده انتقال می‌یابد. علاوه‌براین بدلیل ذخیره نتایج به‌دست آمده از هر گره‌ی همسایه در حافظه نهان[7]، سابقه گره های همسایه در جلورانی‌های آتی تاثیرگذار است. نتایج شبیه‌سازی الگوریتم پیشنهادی و مقایسه ان با الگوریتم‌های جستجوی منابع شایعه‌پراکنی-توازن‌بار[8] و جستجوی احتمال تطبیقی بهبودیافته[9]، نشان می‌دهد که اجتناب از انتخاب‌های تصادفی همسایگان برای جلورانی درخواست و درنظر گرفتن سابقه‌ی هر گره با استفاده از نتایج ذخیره شده در جداول حافظه نهان، سبب کاهش مصرف انرژی، تعداد بسته‌های ارسال‌شده و حذف‌شده و بار تحمیل شده به شبکه می‌گردد. از طرفی با در نظرگرفتن سابقه گره‌های همسایه در جلورانی، پیام درخواست به همسایه‌هایی ارسال می‌گردد که در جستجوهای قبلی موفقتر بوده‌اند، نرخ موفقیت شبکه نیز افزایش می‌یابد.**

**واژه‌های کلیدی:** شبکه نظیربه‌نظیر ادهاک سیار، AIPS، Gossiping_LB، اتوماتای یادگیر، کشف منبع.

[1] Peer To Peer Network
[2] Self Adaptive
[3] Central Managment
[4] Mobile Ad-hoc Network (MANET)
[5] Resource Searching
[6] Broadcast
[7] Cache
[8] Gossiping Load Balancing(Gossping-LB)
[9] Imporoved Adaptive Probablitistic Search (IAPS)

**۱- مقدمه**

شبکه‌های نظیربه‌نظیر در سال‌های اخیر، به سرعت درحال رشد هستند. بسیاری از کاربردهای شبکه‌های نظیربه‌نظیر مانند اشتراک‌گذاری فایل یا منابع چندرسانه‌ای، از ساختار توزیع‌شده‌ی این شبکه‌ها ناشی می‌گردد. شبکه‌های نظیربه‌نظیر، معمولا بدون هیچ کنترل‌کننده مرکزی و کاملا خودسازمان‌دهنده به منظور تامین برخی امکانات مانند انعطاف‌پذیری، مقیاس‌پذیری و قابلیت اطمینان پیاده‌سازی می‌شوند. از سوی دیگر امروزه دستگاه‌های سیار با افزایش پهنای باند، ذخیره‌سازی شبکه و ظرفیت محاسباتی قدرتمندتر شده‌اند و در همه بخش‌های زندگی روزمره حضور دارند. بنابراین تقاضا برای ایجاد شبکه نظیربه‌نظیری که سیار باشد و بتواند منبع را بین دستگاه‌ها سیار بر روی شبکه بی‌سیم به اشتراک بگذارد افزایش یافته است [۱،۲]. تحقیقات بسیاری در مورد پیاده‌سازی پروتکل‌های شبکه‌های نظیربه‌نظیر برروی شبکه سیار صورت گرفته‌است [۶-۳]. در محیط نظیربه‌نظیر سیار مجموعه‌ای از گره‌های متحرک با استفاده از تکنولوژی‌های بی‌سیم باهم ارتباط برقرار می‌کنند [۸،۷]. این نوع از شبکه‌ها برای موقعیت‌هایی که امکان ایجاد زیرساخت وجود ندارد و یا ساختار شبکه بسیار پویا و برقراری ارتباط بین گره‌ها دشوار است مانند جنگ و عملیات نجات، مناسب می‌باشند [۹،۱۰]. در شبکه نظیربه‌نظیرسیار هر گره، یک ناحیه تحت پوشش دارد. گره‌های این محدوده، همسایه محسوب می‌گردند و می‌توان با انها در ارتباط بود درغیر این‌صورت، از گره‌های میانی، برای جلورانی پیام استفاده می‌کنند.

ساختار شبکه‌های نظیربه‌نظیر معمولا به دو گروه ساخت‌یافته و ساخت‌نایافته تقسیم می‌شوند. در شبکه نظیربه‌نظیر ساخت‌یافته، مکان داده‌ها در شبکه از طریق جدول درهم‌سازی توزیع‌شده[1] کنترل می‌شود. در شبکه‌های نظیربه‌نظیر ساخت‌نایافته[2]، گره‌ها به‌سادگی می‌توانند به شبکه وارد یا خارج شوند. بنابراین به راحتی نمی‌توان مکان منبع درخواستی را پیدا کرد. امروزه اکثر کاربردهای سیستم‌های نظیربه‌نظیر، معطوف به شبکه‌های ساخت نایافته نظیربه‌نظیر است و طراحی یک الگوریتم جستجوی کارآمد در این زمینه، یکی از چالش‌های مهم محسوب می‌شود.

در شبکه‌های نظیربه‌نظیر سیار عدم وجود زیرساخت و ساخت-نایافته بودن شبکه سبب می‌شود تا کاربران برای کشف منبع پیام پرس‌وجو را به همه همسایه‌های خود پخش سیل‌آسا کنند، این روند تا زمانی‌که منبع درخواستی یافت شود، ادامه خواهد داشت که منجر به مصرف انرژی بیشتر و افزایش تاخیر شبکه می‌گردد. درحالی‌که به علت محدودیت باتری دستگاه‌های سیار، مصرف انرژی همواره نوع شبکه‌ها مهم است [۱۱].

از جمله الگوریتم‌های ساخت‌نایافته که از پخش‌سیل‌آسا برای ارسال درخواست استفاده می‌کنند می‌توان به الگوریتم‌هایی مانند گنوتلا [۱۳]، گام‌تصادفی [۱۴،۱۵] و شایعه‌پراکنی اشاره کرد. در گنوتلا در هر بار جستجو فقط یک درخواست بررسی می‌شود.

در الگوریتم گام تصادفی[3] پیام‌های پرس‌وجو بطور تصادفی به همسایه‌های گره درخواست‌دهنده ارسال می‌شود. روش گام تصادفی استاندارد، که تنها از یک واکر استفاده می‌کند، تعداد پیام‌ها را با گسترش پیام در سراسر توپولوژی شبکه کاهش می-دهد و همزمان می‌تواند چند درخواست را جستجو نماید.

گنوتلا در مقایسه با روش گام تصادفی استاندارد، سریع‌تر محتوا را می‌یابد و نرخ موفقیت بیشتری نیز دارد اما بار شبکه را خیلی زیاد می‌کند. برای بهبود الگوریتم گام تصادفی تعداد واکرها را افزایش می‌دهیم، یعنی به‌جای ارسال یک پیام پرس‌وجو، گره درخواست دهنده، k تا پیام پرس‌وجو به همسایه‌های خود که به‌طور تصادفی انتخاب گردیده‌اند، ارسال می‌کند. هر یک از این پیام‌ها مسیر خود را دنبال کرده، گره‌های میانی، این پیام‌ها را به‌طور تصادفی به k همسایه‌ی خود، جلورانی[4] می‌کنند. انتخاب همسایه‌ها به‌صورت تصادفی و در نظر نگرفتن انرژی گره‌های سیار، در این الگوریتم، یکی از چالش‌های مهم شبکه نظیربه نظیر سیار است.

الگوریتم شایعه‌پراکنی[5] درخواست را به درصد مشخصی از همسایه‌های خود، جلورانی می‌کند که نسبت به گنوتلا، ترافیک شبکه را کاهش می‌دهد.

برای بهبود شایعه‌پراکنی الگوریتم دیگری به نام شایعه‌پراکنی-توازن بار [۱۵] ارائه گردید که به جای ارسال درخواست به درصد مشخصی از همسایه‌ها، این درصد، بر حسب بار کنونی شبکه محاسبه و اعمال می‌شود. احتمال جلورانی برای هر همسایه طبق فرمول P×(1-u) محاسبه می‌شود. P یک عدد ثابت است و u یک مقدار بین صفر و بر حسب بسته‌های موجود در آن گره می‌باشد. اگر مقدار فرمول مذکور از یک عدد تصادفی بیشتر باشد، جستجو از گره جاری، به این گره همسایه ارسال خواهد شد.

کائو و همکارانش در [۱۶] الگوریتم MOBI-DIC را پیشنهاد داده-اند. این الگوریتم امکان جستجوی‌داده سیار پویا و ناپایدار[6] از طریق همه‌پخشی فراهم می‌کند که سبب افزایش سربار شبکه و زمان

[1] Distributed Hash Table (DHT)
[2] Unstructured

[3] Random Walk
[4] Forward
[5] Gossiping
[6] Unstable

پاسخ می‌شود. در [۱۷] استفاده از مکانیزم حافظه نهان، برای کاهش طول جستجو و ترافیک، ارائه شده است. در [۱۸] پروتکل جستجوی احتمالی ارائه شده است که هر گره k واکر برای کشف هدف در شبکه پخش می‌کند، اما فرآیند جلورانی بجای تصادفی بودن احتمالی است. [۱۹] همچنین براساس همین مقاله APS از پهنای باند بهینه استفاده نمی‌کند. با توجه به نتایج ارائه شده در [۲۰] این الگوریتم کارایی ضغیفی در شبکه پویا دارد. [۲۱] الگوریتم جستجوی تطبیقی[1] را بهبود داده است. این الگوریتم از بهینه‌سازی کلونی-زنبور استفاده کرده است و با توجه به نوع فایل و سابقه جستجو برای این نوع فایل در جستجوهای قبلی، فضای جستجو و سربار تحمیل شده به شبکه را کاهش می‌دهد. اما این الگوریتم برای محیط‌های پویا که انرژی برای گره‌ها مهم است، مناسب نمی‌باشد. IAPS یک رتبه[2] برای هر فایل براساس نتایج جستجوهای قبلی فایل، نگهداری می‌کند. اگر گره‌ای منبع درخواسته‌شده را داشته باشد، رتبه‌ی این فرمت فایل خود را دو واحد افزایش می‌دهد و گره‌های میانی یک واحد افزایش می‌دهند.

اخیرا الگوریتم‌های مبتنی بر شایعه پراکنی به عنوان ابزاری کارآمد برای کاربردهای متفاوت اجرا و مطرح شده‌اند مانند: نگهداری و پاسخ‌گوئی به پایگاه داده[۲۲]، انتشار محتوا [۲۳]، ساخت توپولوژی [۲۴]، مدیریت عضویت نظیرها [۲۵]، جمع‌آوری داده [۲۶]، کشف خطا [۲۷]، در بسیاری از برنامه‌های کاربردی واقعی مانند Tribler [۲۸]، جهت به‌روزرسانی و نگهداری اطلاعات نظیرها [۲۹]، انتقال داده در شبکه نظیربه‌نظیر سیار [۳۴-۳۰]، ذخیره‌سازی فایل [۳۵] و ذخیره‌سازی آگاه از علایق[3] [۳۶].

این مقاله ابتدا به‌دنبال ارائه راه حلی خودتطبیق و یادگیرنده است تا برای جلورانی پیام در شبکه، همسایه‌های مفیدتر انتخاب شوند و گره درخواست‌دهنده، جستجو را با همسایه‌هایی آغاز کند که با احتمال بیشتری ممکن است منبع درخواستی را پیدا کنند. علاوه‌براین ویژگی‌های گره‌های شبکه را در انتخاب همسایه، برای جلورانی پیام تاثیر دهد. با دخیل نمودن ویژگی‌هایی مانند انرژی، سرعت حرکت، تعداد دفعات ورود و خروج از شبکه و مدت زمان حضور گره‌ها در شبکه، می‌توان نتیجه مطلوب از جستجو به‌دست اورد. هرچه گره‌ها سرعت حرکت و تعداد دفعات خروج از شبکه کمتری داشته باشند، پایدارتر هستند و از اطلاعات این گره‌ها در جستجوهای بعدی، می‌توان کمک گرفت. همچنین بهترین مسیر باتوجه به این عوامل و کوتاهترین مسیر برگزیده می‌شود. استفاده از اتوماتای یادگیر، بعنوان یک عامل یادگیرنده که از طریق تعامل با محیط عمل خود را بهبود بخشیده و درخواست‌ها را در شبکه باتوجه به یادگیری جلورانی می‌کند، یک راه حل مناسب می‌تواند باشد و نتایج هر مرحله از جستجو در جداول Cache ذخیره شده و در جستجوهای آتی از اطلاعات آن استفاده می‌شود.

در ادامه سازماندهی بدین صورت است: در بخش ۲ اتوماتای یادگیر که به‌عنوان استراتژی اصلی یادگیری و بهبود در الگوریتم پیشنهادی استفاده شده است به اختصار توضیح داده می‌شود، در بخش ۳ به تشریح و بیان جایگاه مسئله می‌پردازیم، الگوریتم پیشنهادی در بخش ۴ و نتایج شبیه‌سازی برای ارزیابی الگوریتم پیشنهادی در بخش ۵ ارائه می‌شود. در بخش پایانی نتایج حاصل از این گزارش بیان می‌گردد.

## ۲- اتوماتای یادگیر

اتوماتای یادگیر یک واحد تصمیم‌ساز تطبیقی برای اجرا در محیط‌های تصادفی است که عمل بهینه را از طریق تکرار تعامل با محیط خود یادمی‌گیرد. اتوماتای یادگیر [۳۷،۳۸] کارایی را با استفاده از یادگیری اینکه چطور عمل بهینه را از مجموعه عمل-های مجاز از طریق تکرار تعامل با محیط تصادفی انتخاب کند، بهبود می‌بخشد. در هر تعامل، عمل انتخاب‌شده بعنوان ورودی به محیط تصادفی اعمال می‌شود. محیط نیز با تقویت سیگنال پاسخ می‌دهد. بردار احتمال عمل با بازخورد تقویتی از محیط بروز می-شود.

از دیدگاه دیگر اتوماتای یادگیر می‌تواند در دسته دیگر نیز قرار گیرد: اتوماتای یادگیر با ساختار ثابت و اتوماتای یادگیر با ساختار متغیر. اتوماتای یادگیر با ساختار متغیر توسط $<\alpha,\beta,L>$ معرفی می‌شود که $\beta$ مجموعه ورودی است، $\alpha$ مجموعه عمل و L اتوماتای یادگیر است. اگر $\alpha\in\alpha i(k)$ و P(k) معرف عمل انتخابی توسط اتوماتای یادگیر باشد و بردار احتمال برروی مجموعه عمل در لحظه k تعریف گردد. a و b نشان‌دهنده پارامترهای جایزه و جریمه باشند و میزان افزایش و کاهش مجموعه احتمال را بیان می‌کنند. r تعداد عمل‌هایی است که توسط اتوماتای یادگیر به-کارگرفته شده است. در هر لحظه k، بردار احتمال عمل P(k) توسط الگوریتم یادگیری خطی موجود در رابطه (۱) به‌روز می-شود، اگر عمل انتخابی $\alpha i(k)$ توسط محیط تصادفی جایزه داده شود و اگر جریمه دریافت کند بروزرسانی آن در معادله (۲) است.

پاسخ مطلوب:

$$Pi(n+1) = pi(n) + a[1 - pi(n)]$$
$$Pj(n+1) = (1-a)pj(n)\ ,\forall\ j, j \neq i \qquad (۱)$$

پاسخ نامطلوب:

$$Pi(n+1) = (1-b)pi(n)$$

[1] Improve APS (IAPS)
[2] Score
[3] Interersted-aware

$$Pj(n+1) = (b/r-1) + (1-b)\,pj(n), \forall\, j, j \neq i \qquad (۲)$$

اگر b=a روابط (۱) و (۲) الگوریتم (LRP) جایزه جریمه نامیده می‌شود.

## ۳- شرح مسئله

شبکه نظیربه‌نظیرسیار متشکل از ساختار روگستر و زیرگستر است. ساختار زیرگستر بصورت یک گراف اتصال که رئوس آن گره‌های بی‌سیم هستند می‌باشد. در این گراف دو گره درصورتی همسایه هستند که بین آنها لینکی وجود داشته باشد. ساختار روگسستر نیز یک گراف است که رئوس آن نظیرهای شرکت‌کننده در برنامه کاربردی و یال‌ها همسایه‌های مجازی در ساختار روگستر را معرفی می‌کنند [۳۷].

الگوریتم شایعه‌پراکنی یکی از الگوریتم‌های جستجوی منبع است. در این الگوریتم بدلیل جلورانی کورکورانه و تصادفی درخواست به همسایگان، نرخ موفقیت کم است. از طرف دیگر در شبکه نظیربه-نظیر سیار انرژی برای گره‌ها همواره حائز اهمیت بوده است. انتخاب تصادفی گره‌ها با یک احتمال یکسان ممکن است سبب برگزیدن مسیر طولانی‌تر برای رسیدن به هدف و یا انتخاب مسیری با سطح انرژی گره‌های نامناسب منجر گردد. از این‌رو هو و همکارانش در [۱۵] الگوریتم شایعه‌پراکنی-توازن‌بار را ارائه کردند. این الگوریتم تا حدودی از ارسال ناآگاهانه پیام به گره‌های همسایه اجتناب می‌کند اما بدلیل درنظر گرفتن یک ضریب ثابت P در محاسبه احتمال، جلورانی پیام پرس‌وجو به همسایگان هنوز نرخ موفقیت کمی دارد.

برای جستجوی منابع، شجاع‌فر و دوستان در [۲۱] الگوریتم جستجوی احتمالی تطبیقی را بهبود داده و الگوریتم IAPS را ارائه کرده‌اند. این الگوریتم فرآیند جلورانی به جای تصادفی بودن احتمالی است و تا حدودی از ارسال ناآگاهانه به همسایه‌ها اجتناب کرده است. یکی از ویژگی‌های بارز این الگوریتم، یادگیرنده بودن آن است اما به دلیل استفاده از k واکر در جلورانی درخواست‌ها در شبکه، تعداد بسته‌های ارسالی و انرژی مصرفی و زمان پاسخ زیادی دارد.

در این مقاله، ما به دنبال بهبود الگوریتم‌های شایعه‌پراکنی-توازن بار و جستجوی بهبودیافته‌ی احتمال تطبیقی هستیم. الگوریتم پیشنهادی با توجه به بازخورد دریافتی از شبکه، به‌طور آگاهانه، همسایه‌های مفیدتر را برای جلورانی پیام پرس‌وجو، انتخاب می-کند. به همین دلیل نرخ موفقیت، مصرف انرژی، بار شبکه، تعداد بسته‌های ارسالی و زمان پاسخ، بهبود می‌یابد.

## ۴- الگوریتم پیشنهادی

تمرکز اصلی الگوریتم پیشنهادی بر روی جستجوی منبع در شبکه نظیربه‌نظیر سیار با بکارگیری محیط مدل S اتوماتای یادگیر در هر گره است. در این رویکرد هر گره‌ی درخواست‌دهنده مجهز به یک اتوماتای یادگیر است که N تعداد همسایه‌های گره است. اتوماتای یادگیر مرتبط با همسایه jام را با LAj نمایش می‌دهیم. LAj دارای دو عمل است؛ "شرکت‌کردن گره Nj در جستجو" و "شرکت‌نکردن گره Nj در جستجو" که مجموع احتمال این دو عمل همواره ثابت و برابر یک است. در آغاز، احتمال عمل‌های آن باهم برابر و مساوی ۰,۵ می‌باشد.

به‌طور کلی هر گره‌ای که درخواستی داشته باشد ابتدا منبع درخواستی را درخود جستجو می‌کند اگر کشف شود جستجو خاتمه می‌یابد، درغیراینصورت یک بسته با نام DREQ ایجاد و به تعدادی از همسایه‌های خود می‌فرستد. این روند تازمانیکه منبع درخواستی در شبکه پیدا شود یا بعبارت دیگر HIT رخ دهد و یا طول عمر بسته[1] خاتمه یابد، ادامه خواهدیافت و وضعیت بسته از DREQ به DREP تغییر می‌یابد. اما اگر منبع درخواستی توسط یکی از همسایه‌ها کشف شود ولی طول عمر بسته هنوز صفر نشده باشد، گره‌ای که شامل منبع درخواستی است مدتی منتظر بازگشت نتیجه جستجوی سایر گره‌های می‌ماند، یعنی وضعیت بسته از DREQ به DWAIT تبدیل می‌شود و بعد از اتمام این زمان انتظار، وضعیت بسته DWAIT به DREP تبدیل می‌شود. به-طور کلی وضعیت بسته‌های شبکه، یکی از سه موارد ذیل می‌باشد:

**DREQ**: این بسته در حال جستجوی منبع درخواستی می‌باشد. هنگامیکه گره‌ای، درخواست برای منبعی داشته باشد، بسته‌ای با این وضعیت به تعدادی از همسایه‌های خود ارسال می‌کند.

**DWAIT**: بسته‌ای که در این وضعیت است در واقع منبع درخواستی را کشف نموده اما چون TTL> 0 است، در گره‌ی مقصد، منتظر می‌ماند تا نتیجه جستجوی سایر مسیرهای این درخواست در را ه بازگشت به گره درخواست‌دهنده برسند.

**DREP**: بسته‌ای است که نتیجه جستجو را به گره آغازین، بر می گرداند. هنگامیکه منبع درخواستی کشف شد و یا TTL=0 شد، این بسته توسط گره مقصد ایجاد و به گره‌های ماقبلش ارسال می-گردد. در حقیقت بسته‌های HIT که درحالت DREP هستند در فاز انتقال منبع درخواستی، به گره درخواست‌دهنده می‌باشند.

هنگام بازگشت بسته‌ها به گره درخواست‌دهنده، هر گره باتوجه به بسته‌هایی که از همسایه‌های خود دریافت کرده، بهترین مسیر را انتخاب می‌کند. علاوه‌براین عمل‌های اتومای یادگیر نیز بروز می-

[1] Time-To-Live (TTL)

شود و نتیجه‌ی جستجوی در گره‌ها در جدولی بنام جدول Cache ثبت می‌شود. این روند تازمان رسیدن به گره آغازین ادامه می‌یابد. در نهایت گره آغازین اتوماتای یادگیر همسایه‌هایی که عمل شرکت نکردن در جستجو را انتخاب کرده بودند را بروز می‌کند.
در جدول ۱ ساختار بسته‌های ارسال شده در شبکه امده است که جزییات فیلدهای ان به قرار ذیل است:

**جدول ۱: ساختار بسته‌های ارسالی در شبکه**

| Bit offset | 0-7 | 8-15 | 16-27 | 28-31 |
|---|---|---|---|---|
| 0 | MessageID | | | |
| 32 | Data ID | | | |
| 64 | Sender Node IP Address | | | |
| 69 | TTL | | Packet State | Success State |
| | WaitingTime | | Route Length | Data Length |
| ... | Route (variable) | | | |
| ... | Data (variable) | | | |

**Message ID:** شناسه‌ی عددی

**Data ID:** شناسه‌ی عددی منبع مورد جستجو که هر منبع‌ای در شبکه دارا دادای یک شناسه عددی است.

Sender Node IP Address: ادرس IP گره‌ی شروع‌کننده‌ی جستجوی منبع درخواستی

**TTL:** طول عمر بسته

**Packet State:** وضعیت پیام ارسالی ( DREQ=0X00, DWAIT=0X02, DWAIT=0X01)

**Success State:** وضعیت پیدا شدن یا نشدن منبع درخواستی در گره جاری (HIT=0X01, MIS=0X02)

Waiting Time: میزان زمان توقف بسته با وضعیت DWAIT در گره ی جاری

**Route Length:** تعداد گره‌های پیمایش شده از گره درخواست-دهنده داده در حین جستجو

**Data Length:** اندازه منبع یافت‌شده

**Route:** ادرس IP گره‌های پیمایش‌شده از درخواست‌دهنده‌ی منبع در حین جستجو

**Data:** منبع درخواستی یافته شده

نحوه دقیق عملکرد الگوریتم پیشنهادی در زیر به تفصیل بیان شده است؛

در شروع هر گره‌ای که درخواستی داشته باشد، ابتدا درخود جستجو می‌کند. اگر منبع درخواستی را دارا باشد، جستجو خاتمه می‌یابد، درغیراینصورت یک بسته با وضعیت DREQ و TTL مشخص، ایجاد کرده و شناسه خود را در مکان معلومی، در سرایند بسته درج می‌کند. در اولین جستجوی منبع برای هر گره به صورت تصادفی، برای تعدادی از همسایه‌ها عمل شرکت‌کردن و برای بقیه همسایه‌ها عمل‌شرکت نکردن در جستجو انتخاب می-شود. همسایه‌هایی که عمل شرکت‌نکردن در جستجو را انتخاب کرده‌اند، اطلاعاتشان در جدول Non-Assistant موجود در آن گره ذخیره می‌شود و همسایه‌هایی که عمل شرکت‌کردن در جستجو را انتخاب کرده‌اند، پیام پرس‌وجو به آن‌ها ارسال می‌شود. هر همسایه‌ای که پیام پرس‌وجو را دریافت کرد، شناسه‌ی درخواست را در لیست بسته‌های درخواستی خود قرار می‌دهد و همچنین شناسه خود را در سرایند بسته درج می‌کند تا مشخص‌کننده مسیری باشد که بسته تا آن گره پیموده، تا در هنگام بازگشت بتواند از همین مسیر باز گردد. علاوه‌براین یک واحد از TTL کم کرده سپس بررسی می‌کند که آیا منبعی درخواستی را در لیست داده‌های ذخیره شده در خود، دارد یا خیر که دو حالت زیر ممکن است رخ دهد:

۱. منبع درخواستی در گره یافت می شود، پس HIT رخ داده است و منبع درخواستی در بسته درخواست گذاشته می شود حال اگر:

۱،۱. اگر TTL برابر با صفر شده بود، وضعیت بسته درخواست از حالت DREQ به حالت DREP تبدیل می‌شود و جستجو خاتمه می‌یابد.

۱،۲. اگر TTL صفر نشده باشد وضعیت بسته درخواست از حالت DREQ به حالت DWAIT تبدیل شده و مدت زمانی، در این گره، منتظر برگشت بسته‌های جستجو این درخواست، باقی می‌ماند. این زمان انتظار با استفاده از رابطه (۳) محاسبه می-گردد:

$$WatingTime = 2 * (initTTLValue - HopCount) \quad (۳)$$

در این فرمول $initTTLValue$ طول عمر بسته و $HopCount$ تعداد گره‌هایی است که درخواست از گره آغازین تا این گره پیموده است. در هر واحد زمانی، یک واحد از مقدار $WatingTime$ کم می‌شود. اگر مقدار زمان انتظار بسته برابر با صفر شود وضعیت بسته از حالت DWAIT به حالت DREP تبدیل شده و نتیجه HIT به گره اغازین برمی‌گردد.

۲. اگر منبع درخواستی در گره همسایه یافت نشود:

۲-۱ اگر TTL بسته صفر شده باشد وضعیت بسته درخواست از حالت DREQ به حالت DREP تبدیل می‌شود و جستجو

خاتمه می‌یابد و نتیجه شکست (MIS) به گره‌ی آغازین ارسال می‌گردد در مسیر احتمال عمل‌های اتوماتا را به‌روز می‌کند.

۲-۲ اگر TTL صفر نشده باشد و منبع درخواستی نیز یافت نشود، روند جستجو و ارسال بسته به همسایه، ادامه می یابد که ابتدا در جدول Cache که در هر گره ذخیره شده است، جستجو آغاز می‌شود.

جدول Cache موجود در هر گره شامل دو ستون شناسه گره همسایه و تگ منبع کشف شده است. اگر تگ منبع درخواستی در این جدول پیدا شد، بررسی می‌کند که آیا گره درج شده در جدول Cache هم اکنون نیز همسایه گره جاری است یا از ناحیه تحت پوشش گره خارج شده است. اگر آن همسایه هنوز معتبر باشد، بسته‌ی درخواست، از گره جاری خارج و به آن ارسال می‌گردد. ولی اگر آن همسایه معتبر نباشد، گره جاری، برای ادامه جستجو، باید همسایه‌هایش را شناسایی کند تا درخواست را به گره‌هایی که با احتمال بیشتری ممکن است منبع درخواستی را کشف کنند، ارسال نماید.

هر گره برای شناسایی همسایه‌های خود از پیام‌های PONG-PING استفاده می‌کند. یک پیام بنام PING به گره‌هایی که در ناحیه تحت پوشش هستند ارسال می‌کند. از هر گره‌ای که بسته PONG دریافت شود، گره همسایه محسوب خواهد شد. پیام PONG که توسط هر همسایه ارسال می‌شود شامل سه فیلد En، Pa و Pnon است. Pnon احتمال عمل شرکت‌نکردن اتوماتای یادگیر، Pa احتمال عمل شرکت‌کردن اتوماتای یادگیر و En سطح انرژی گره همسایه است. حال گره جاری که بسته‌های PONG را از همسایه-های خود دریافت می‌کند، سطح انرژی هر گره همسایه را با سطح آستانه‌ی انرژی مقایسه می‌کند. اگر انرژی گره همسایه بزرگتر از سطح آستانه‌ی انرژی بود، حال اتوماتای یادگیری که در هر گره با دو عمل Pn و Pnon ذخیره شده است بررسی می‌گردد. هر عملی که احتمال بیشتری داشته باشد آن عمل انتخاب خواهد شد، لذا هر کدام از همسایه‌هایی که Pa آن بزرگتر از Pnaش باشد، درخواست به آن ارسال می‌گردد.

اگر گره‌ای، درخواستی را همزمان از چند مسیر متفاوت دریافت کرد، طول مسیرهای طی‌شده[1] (تعداد گام پیمایش شده) را بررسی می‌کند و هر کدام کم‌تر باشد نگه داشته و بقیه را حذف می‌کند.

جستجو زمانی خاتمه می‌یابد که یا منبع درخواستی کشف گردد (HIT) و یا طول عمر بسته تمام شده باشد ولی منبع درخواستی هنوز کشف نشده باشد که در چنین حالتی شکست (MIS) رخ داده است. در نهایت نتیجه جستجو در بسته‌ای به نام DREP قرار داده می‌شود به سمت گره‌ی اغازین بسته ارسال می‌شود و اتوماتای تمامی گره‌های مسیر بروز شوند. بسته‌هایی که در وضعیت DREP هستند، نتیجه جستجو را از مسیری که در سرایندشان درج شده، یعنی بهترین مسیر، به گره اغازین برمی-گردانند. بهترین مسیر، مسیری است که سطح انرژی گره‌های ان مناسب و کمترین تعداد گره بین مبدا و مقصد وجود داشته باشد، گره‌هایش پایدارتر وکم سرعت باشند.

اگر نتیجه جستجو HIT بود، در مسیر بازگشت، اطلاعات این بهترین مسیر، یعنی شناسه گره همسایه و تگ منبعی درخواستی را در جدول Cache خودش ثبت می‌کند تا در جستجوهای بعدی از ان‌ها کمک بگیرید. هر ایتمی که به جدول Cache اضافه می-شود طول عمری دارد. در هر واحد زمانی، هر گره جدول Cache خود را بررسی می‌کند، اگر عضوی داشته باشد که طول عمرش منقضی شده باشد، آن را حذف می‌نماید. اما اگر ظرفیت جدول Cache در هنگام درج ایتم جدید، تکمیل باشد، درایه‌های قدیمی-تر را حذف نموده تا بتوانند درایه‌ی جدید را درج نماید. سپس احتمالات عمل‌های اتوماتای یادگیر را به‌روز می‌کند. هدف از به‌روز رسانی عمل‌های اتوماتا این است که گره‌هایی که در مسیری قرار داشتند که منجر به HIT شده‌اند را پاداش دهیم تا در جستجوهای بعدی احتمال عمل شرکت کردن آن‌ها افزایش یابد زیرا گره‌های مفیدی در جستجو بوده‌اند. همچنین گره‌هایی که در مسیری قرار داشتند که منجر به MIS شده‌اند را تنبیه کنیم تا در جستجوهای بعدی احتمال عمل شرکت‌کردن آن‌ها کاهش یابد، زیرا گره‌های غیر مفیدی در جستجو بوده‌اند. برای هر گره‌ی میانی برای هر درخواستی که هم در مسیری که منجر به HIT و هم در مسیر دیگری که منجر به MIS شده، حضور داشته باشد، عمل شرکت‌کردن اتوماتای گره اعمال می گردد و گره پاداش دریافت می‌کند. بروزرسانی عمل‌های اتوماتا در هر گره به این صورت انجام می‌شود: در زمان بازگشت بسته‌ها به گره آغازین، هر گره در مسیر بازگشت، تمام بسته‌هایی که در وضعیت DREP و مربوط به هر درخواست هستند را بررسی می‌کند و یکی از دو مورد ذیل لحاظ می شود:

۱- اگر از بین بسته‌های این درخواست، بسته‌هایی که موفق به یافتن منبع درخواستی شده‌اند (HIT)، موجود باشند، آن‌ها را جدا کرده و با استفاده از رابطه (۳) محاسبات انجام می‌شود و بسته‌ای که بیشترین مقدار را دارد به‌عنوان بهترین بسته، انتخاب خواهد شد تا منبع را به گره درخواست‌دهنده برساند. این مسیر به‌عنوان بهترین مسیر شناسایی می‌شود. بقیه این بسته‌های HIT از شبکه حذف می‌گردند. علاوه‌براین عمل‌های اتوماتای یادگیر گره جاری پاداش داده می‌شود. یعنی Pa ان‌ها را افزایش وPnon را کاهش

[1] Hop Count

می‌دهیم تا در جستجوهای بعدی نیز احتمال عمل شرکت‌کردن آن‌ها افزایش یابد. جدول Cache نیز به‌روز می‌شود. سپس تمام بسته‌های MIS مربوط به این درخواست در گره جاری، از شبکه حذف خواهند شد. درنهایت این بهترین بسته از گره جاری خارج و به گره قبلی موجود در مسیر جستجو ارسال می‌شود.

۲- اگر بسته HIT ای وجود نداشت باشد، بسته‌های MIS مربوط به این درخواست را جدا می‌کند. با استفاده از رابطه (۳)، محاسبات بر روی انها انجام می‌شود. بسته‌ای که بیشترین مقدار را داشته باشد انتخاب می‌شود و بقیه ان‌ها را از شبکه حذف می‌کند. سپس این گره با جاری با محاسبه β(i) طبق رابطه (۴) و استفاده از فرمول (اتوماتای یادگیر) عمل‌های اتوماتای یادگیر خودش را تنبیه می‌کند یعنی Pnon آن‌ها را افزایش بدهیم و Pa آن‌ها را کاهش تا در جستجوهای بعدی شرکت نکنند. در نهایت این بسته از گره جاری خارج می‌شود و به گره قبلی در مسیر جستجو، که در سرایند این بسته درج شده، ارسال می‌شود. این روند تا رسیدن به گره آغازین ادامه می‌یابد و احتمال عمل‌های اتوماتای یادگیر را به‌روز می‌کند. اگر بسته‌ی HIT ای به گره درخواست دهنده منبع مورد جستجو رسید، این گره، منبع درخواستی را برداشته و تمام بسته‌های مربوط به این درخواست را حذف می‌کند.

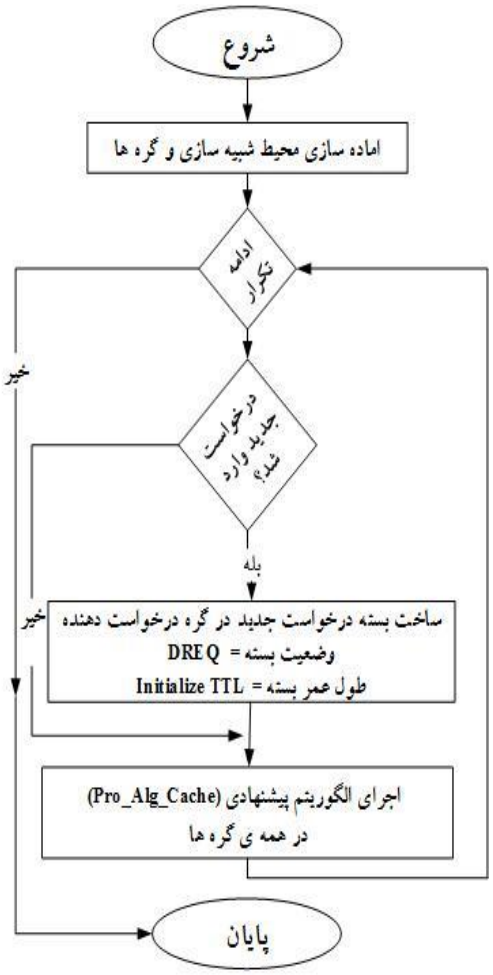
$$\gamma(i) = \lambda_1\left(\frac{energy[i]}{\left(\frac{\sum enrgy}{hopCount}\right)}\right) + \lambda_2\left(\frac{time[i]}{n[i]*StartTime}\right) + \lambda_3\left(\frac{Vmin}{Vi}\right),\ \lambda_1 = \lambda_2 = \lambda_3 = (1/3) \qquad (۳)$$

$$\beta\,(i) = ۱ - \gamma\,(i) \qquad (۴)$$

در پایان هر جستجو باید احتمال عمل شرکت‌کردن و عمل شرکت‌نکردن اتوماتای یادگیر همه گره‌های شرکت‌کننده در جستجو، به‌روز شود و علاوه‌براین اتوماتای یادگیر آن گروه از گره‌های همسایه‌ی گره‌ی اغازین، که عمل شرکت‌نکردن در این جستجو را برگزیده بودند، نیز به‌روز شود و اطلاعات آن‌ها از جدول Non-Assistant گره اغازین حذف شود. زیرا اگر در این جستجو، منبع درخواستی یافت شد، همسایه‌هایی که در جستجو شرکت نکرده‌اند باید پاداش دریافت کنند تا احتمال عمل شرکت‌نکردن آنها افزایش یافته و در جستجوهای آتی نیز شرکت نکنند ولی اگر در نتیجه این جستجو، منبع یافت نشد، آن‌ها را تنبیه می‌کنیم تا احتمال عمل شرکت‌نکردن آن‌ها کاهش یافته و بتوانند در جستجوهای آینده شرکت نمایند.

## ۵- جایگاه و شبه کد الگوریتم پیشنهادی

در شکل(۱) جایگاه الگوریتم پیشنهادی آمده است:




**شکل ۱:جایگاه الگوریتم پیشنهادی**

که در ذیل شبه کد الگوریتم پیشنهادی آمده است:

**الگوریتم ۱: شبه کد الگوریتم پیشنهادی با مگانیزم** Cache

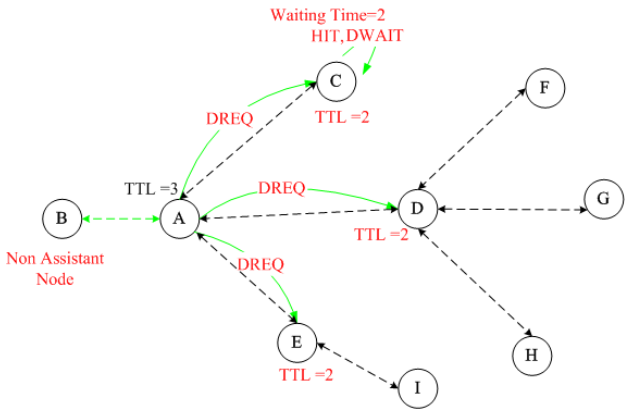
```
      n: Current Node in Network
1:    procedure Pro_Alg_Cache(n)
2:        ManageDREQ(n);//manage DREQ packets in n
3:        ManageDWAIT(n);//manage DWAIT packets in n
4:        ManageDREP(n); //manage DREP packets in n
5:        UpdateCacheTable(n);//remove expired items in n
6:    end procedure
7:    procedure ManageDREQ(n)
8:      foreach  p in n's DREQ Pcakets do
9:        if have_find(p.DataID) then
10:           p.SuccessState=HIT;
11:       end if
12:       if  p.TTL==0 then
13:           p.PacketState = DREP;
14:       else if (p.SuccessState==HIT) then
15:           p.PacketState = DWAIT;
16:       Else
17:           if n.ID==p.SenderID then
```

```
18:                 FillNonAsistantNodeList();
19:               end if
20:             ForwardQueryToNeighboures(n);/*select
      neighbours by automata and forward p to them*/
21:           end if
22:       Done
23:   end procedure
24:   procedure ManageDWAIT(n)
25:       foreach p in n's DREQ Pcakets do
26:           if (n.waitingTime == 0) then
27:               n.State = DREP
28:           else
29:               n.waiting time --;
30:           end if
31:       Done
32:   end procedure
33:   procedure ManageDREP(n)
34:       foreach p in n's DREP packets do
35:           UpdateLearningAutomata(n);
36:           if  p.SuccessState == HIT then
37:               AddDataToCacheTable(p.Data);
38:           end if
39:           if p.SenderNodeIPAddress == p.IPAddress then
40:               UpdateNoneAssitstantTable(n);
41:           Else
42:               ForwardBackInRoutePath(p);
43:           end if
44:       Done
45:   end procedure
```

## ۶- یک نمونه مثال کاربردی از اجرای الگوریتم پیشنهادی

در این قسمت سناریویی از اولین بار اجرای الگوریتم پیشنهادی در گره‌ی A، امده است، مقدار طول عمر بسته‌ها در اغاز ۳ در نظر گرفته‌شده است:

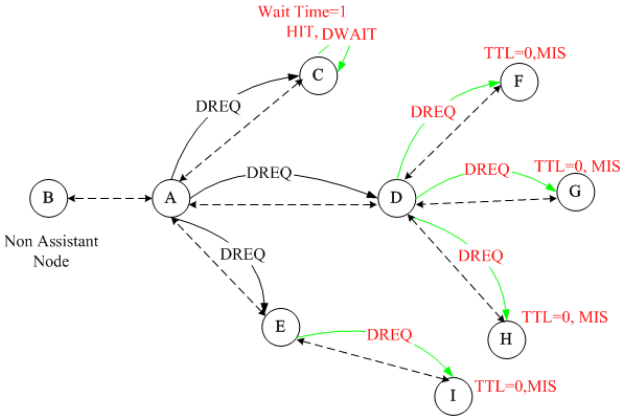
گره A یک درخواستی برای منبعی دارا می باشد. مطابق شکل(۲)، ابتدا یک درخواست DREQ، به تعداد تصادفی از همسایگانش (گره‌های C,D,E) ارسال می‌کند و بقیه همسایگانش را (گره B) به‌عنوان گره‌های None Assitant ذخیره می‌کند. در حین جستجو، در گره C، منبع مورد جستجو یافت می‌شود ولی چون، TTL بسته ۲ می‌باشد، به اندازه ۲ واحد زمانی منتظر بازگشت نتایج جستجو باقی می‌ماند.




**شکل ۲: ارسال درخواست های DREQ از گره آغازین**

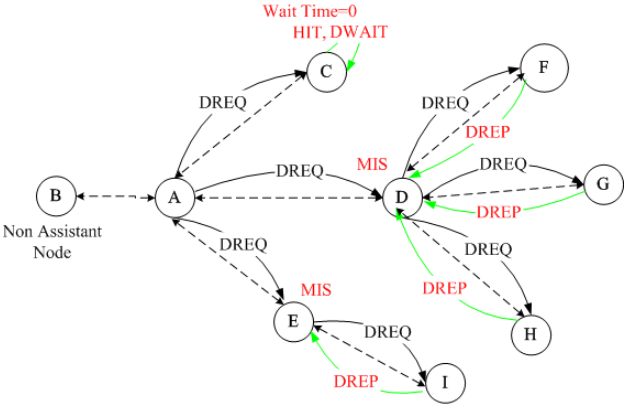
ولی چون گره‌های D,E منبع مورد جستجو را ندارند، درخواست DREQها را به همسایه‌های خود ارسال می‌نمایند.

یعنی مطابق شکل(۳) بسته از گره D به گره‌های F, G, H و از گره E به گره I ارسال می‌شود. در این گره‌ها، TTL بسته‌ها صفر می‌شود و جستجو به پایان می‌رسد ولی چون هیچ کدام حاوی محتوای درخواستی نیستند در نتیجه، جستجو در این گره‌ها MIS می‌شود و این گره‌ها تنبیه می‌شوند. اتوماتای یادگیر ان‌ها به‌روز رسانی می‌شوند.




**شکل ۳: ارسال درخواست‌های DREQ از گره D, E**

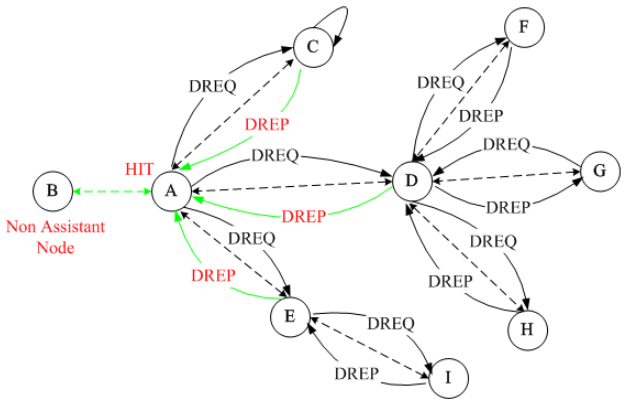
نتیجه جستجو مطابق شکل (۴) به گره‌های ماقبل (D, E) باز گردانده می‌شود. چون گره‌های D, E نیز در جربان شکست جستجو (MIS) حضور داشتند، این دو گره نیز، تنبیه می‌شوند. در این هنگام، زمان انتظار بسته موجود در گره C به پایان رسیده و چون منجر به HIT شده است، به این گره پاداش داده می‌شود و اتوماتای یادگیر آن به‌رروز رسانی می‌شود.




**شکل ۴: ارسال درخواست‌های DREP از انتها به سمت مبدا**

در نهایت مطابق شکل (۵) از بین بسته‌ی HIT شده‌ی گره C و بسته‌های MIS گره‌های D, E بسته HIT گره C، به‌عنوان نتیجه جستجو به گره آغارین، یعنی A می‌رسد و اتوماتای یادگیر آن به‌روز رسانی می‌شود. چون گره A، موفق به یافت منبع شده است، تشویق می‌شود. چون گره A بدون حضور گره B توانست منبع مورد نظر خود را بیابد، عمل شرکت نکردن B نیز تشویق می‌شود

تا در جستجوهای اینده نیز شرکت نکند. گره A در جستجوهای آینده، به همسایه‌هایی که Pa ان‌ها بیشتر از Pna باشد درخواست خود را جلورانی می‌کند.




**شکل ۵: رسیدن درخواست‌های DREP به گره آغازین**

## ۷- شبیه‌سازی و ارزیابی

برای شبیه‌سازی الگوریتم پشنهادی از نرم افزار اورسیم مطابق جدول ۲ استفاده شده است. گره‌های الگوریتم‌های پیشنهادی با مدل حرکتی Random way point [۳۹] در شبکه حرکت می‌کنند. ۲۵۰ منبع مختلف با حداکثر سایز ۵۱۲ بایت در شبکه، بین گره‌ها به صورت یکنواخت پخش شده است. همواره ۷۵٪ گره‌ها در شبکه آنلاین هستند و بقیه گره‌ها به شبکه وارد و خارج می‌شوند. جستجو برای منبع زمانی در شبکه خاتمه می‌یابد که یا منبع یافت شود و یا TTL صفر گردد. برای محاسبه انرژی مصرف‌شده در شبکه انرژی هر گره از مدل رادیویی IEEE 802.11b که بر روی کارت Cisco Aironet 350 [۴۰] با توان انتقال ۱۰dBm استفاده شده است. هر گره برای انتقال داده ۱,۶۸۸۷وات، برای دریافت‌داده ۱,۰۷۹۱ و در حالت بیکاری ۰,۶۶۹۹ وات انرژی صرف می‌کند. تعداد درخواست‌ها ۱۲۵۰ درخواست می‌باشد که با پراکندگی زمانی یکنواخت به گره‌های شبکه، لحاظ می‌شوند.

الگوریتم پیشنهادی با الگوریتم‌های شایعه‌پراکنی-توازن‌بار و جستجوی بهبودیافته احتمال تطبیقی، تحت سناریوی افزایش تعداد گره‌های شبکه، مقایسه شده‌اند. تعداد درخواست‌ها و منابع-های موجود در شبکه، همواره ثابت و مطابق جدول (۲) می‌باشند و صرفا تعداد گره‌های شبکه از ۵۰ تا۱۰۰، افزایش می‌یابند. از هر بار افزایش گره‌ها، ۵۰ بار اجرا گرفته شده، سپس حاصل میانگین آن-ها، در نمودارها لحاظ شده است.

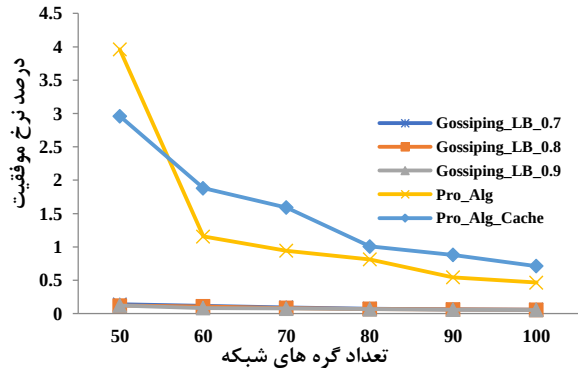
**جدول ۲: پارامترهای شبیه‌سازی**

| پارامترها | مقادیر |
|---|---|
| ابعاد محیط شبیه سازی | ۱۰۰۰ متر × ۱۰۰۰ متر |
| حداکثر طول عمر بسته‌ها | ۶ واحد زمانی |
| شعاع ناحیه تحت پوشش هر گره | ۲۵۰متر |
| تعداد منابع موجود در شبکه | ۲۵۰ |
| تعداد منابع موجود در هر گره | ۵ |
| تعداد درخواست‌ها | ۱۲۵۰ |
| سرعت حرکت گره‌ها | ۲ تا ۱۶ متر بر ثانیه |
| نرخ خطای کانال | ٪ |
| حداکثر تعداد بسته‌ها در NIC | ۳۰ |
| مشخصات سیستم اجرایی | GHZ 2.7 GHz CPU: Core i5- 2.6 RAM: 4 G 12.0.4) OS: Linux (Ubuntu |

## ۸- مقایسه الگوریتم پیشنهادی با Gossiping-LB

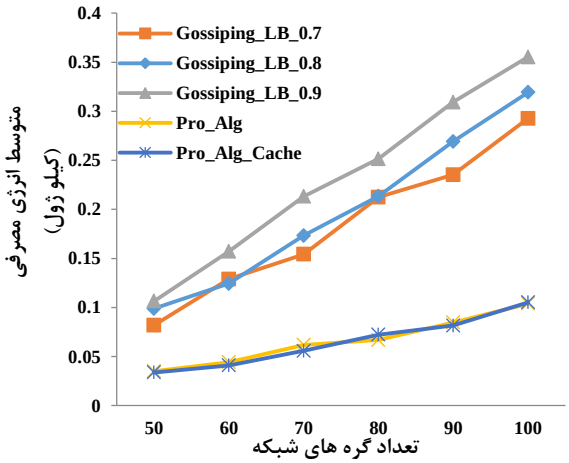
در این آزمایش الگوریتم پیشنهادی بدون مکانیزم Cache (Pro_Alg) و نیز همراه با مکانیزم Cache(Pro_Alg_Cache) با الگوریتم شایعه‌پراکنی-توازن‌بار با pهای ۰,۹ ، ۰,۸ و ۰,۷ , Gossiping_LB_0.9) Gossiping_LB_0.8, Gossiping_LB_0.7) مقایسه شده‌اند. در تمامی نمودارهای حاصل از این ازمایش، Gossiping LBها، به دلیل رفتار تصادفی، دارای فراز و نشیب‌های محلی می‌باشند ولی نمودارهای الگوریتم پیشنهادی به‌دلیل یادگیرنده بودن، روند ثابتی دارند و تقریبا خطی هستند.

شکل (۶) درصد نرخ موفقیت برمبنای افزایش گره‌های شبکه را نشان می‌دهد. همان‌طورکه ملاحظه می‌شود، با افزایش تعداد گره-های شبکه، نرخ موفقیت الگوریتم پیشنهادی در ابتدا زیاد و رفته رفته از میزان موفقیت کاسته می‌شود اما همواره عملکرد الگوریتم-های پیشنهادی بدلیل انتخاب هوشمندانه همسایگان در جلورانی پیام نسبت به الگوریتم Gossiping_LB، بهتر است. درحالیکه در Gossiping_LB همواره عدد تصادفی آستانه موجود در فرمول برای ارسال درخواست به همسایه‌ها لحاظ می‌شود، به‌عبارت دیگر درخواست‌ها تا حدی کورکورانه در شبکه جلورانی می‌شود. الگوریتم Pro-Alg-Cache بهبودیافته الگوریتم Pro-Alg می‌باشد که شامل مکانیزم Cache می‌باشد. نرخ موفقیت الگوریتم Pro-Alg-Cache نسبت به الگوریتم Pro-Alg به دلیل استفاده از جداول Cache که نتایج جستجوها را ذخیره می‌کند، بیشتر است.



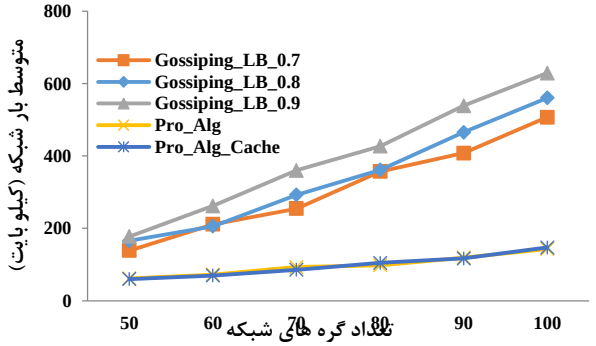

**شکل ۶: نرخ موفقیت به ازای افزایش تعداد گره‌های شبکه**

در نمودار شکل(۷) میزان مصرف انرژی گره‌های شرکت‌کننده در جستجو بر مبنای افزایش تعداد گره‌های شبکه مورد بررسی قرار گرفته شده است. ملاحظه می‌شود که الگوریتم پیشنهادی بدلیل اینکه به مسیرهای با انرژی بهینه که منبع درخواستی را کشف می‌کنند پاداش می‌هد، بنابراین در جستجو های آتی، درخواست‌ها به گره‌هایی ارسال می‌شوند که با احتمال بیشتری موفق به یافت منبع درخواستی می‌باشند. در نتیجه انرژی گره‌ها بیهوده صرف نمی‌گردد و فقط گره‌هایی که با احتمال بیشتر شامل منبع هستند در جستجو شرکت می‌کنند. درحالیکه Gossiping_LBها تا حدی تصادفی در جستجو شرکت می‌کنند که سبب برگزیدن مسیر طولانی‌تر برای رسیدن به هدف و یا انتخاب مسیری با سطح انرژی گره‌های نامناسب منجر می‌گردد. در بین آن‌ها نیز Gossiping_LB_0.9 چون درخواست‌های خود به تعداد گره-های بیشتری می‌فرستد، بیشترین مقدار انرژی را مصرف می‌کند. چون Pro_Alg_Cache در جستجوهای خود از Cache استفاده می‌کند، به نسبت Pro_Alg انرژی کم‌تری مصرف می‌کند.



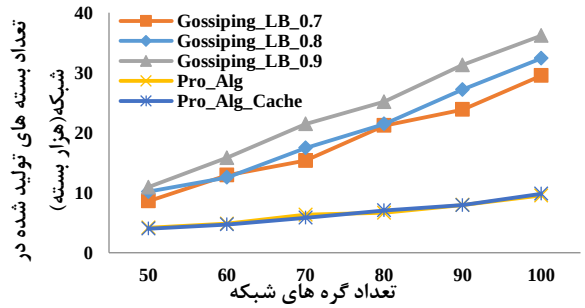

**شکل ۷: متوسط مصرف انرژی به ازای افزایش تعداد گره‌ها**

شکل (۸) مقایسه‌ای بر مبنای متوسط بار تحمیل شده به شبکه ناشی از افزایش گره‌های شبکه را نشان می‌دهد. از آنجایی که حداکثر طول عمر بسته‌ها ۶ می‌باشد، تعداد پیام‌های تولیدشده در شبکه به نحوی کنترل می‌گردد. باتوجه به شکل بار تحمیل‌شده به شبکه توسط الگوریتم پیشنهادی کمتر از تمام Gossiping_LB می‌باشد، چون Gossiping_LB تا حدی تصادفی عمل می‌کند. بار شبکه در Gossiping_LB_09 از همه‌ی نمودارها بیشتر می‌باشد زیرا به تعداد همسایه‌های بیشتری درخواست‌های خود را ارسال می‌نماید.

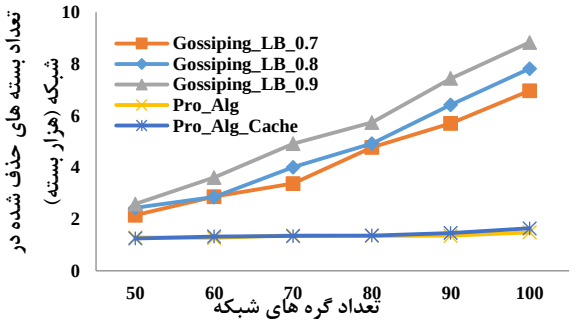



**شکل ۸: متوسط بار شبکه به ازای افزایش تعداد گره‌ها**

شکل‌های (۹) و (۱۰) به ترتیب تعداد بسته‌های تولیدشده و تعداد بسته‌های حذف‌شده در شبکه را برمبنای افزایش تعداد گره‌های شبکه، را نشان می‌دهند. چون الگوریتم پیشنهادی مبتنی بر یادگیری است بسته‌های کم‌تری در شبکه تولید و حذف می‌کند ولی Gossiping_LBها چون تا حدودی تصادفی عمل می‌کنند، بسته‌های نامناسب بیشتری تولید و حذف می‌کنند. Gossiping-LB_0.9 چون همسایه‌های بیشتری را انتخاب می‌کند، بیشترین نرخ تولید و حذف بسته را دارا می‌باشد.




**شکل ۹: متوسط تعداد بسته‌های تولید شده در شبکه**

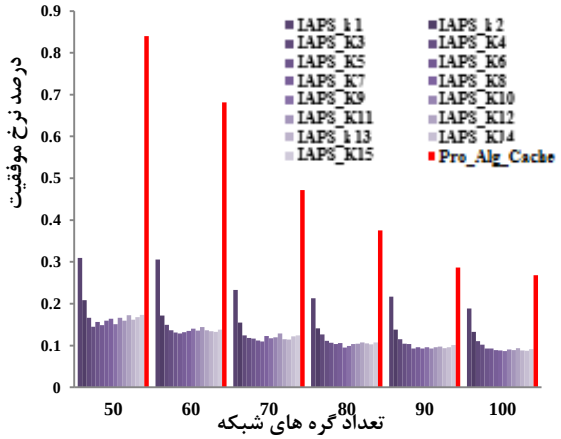



**شکل ۱۰: متوسط تعداد بسته‌های حذف شده در شبکه به ازای افزایش تعداد گره‌های شبکه**

## ۹- مقایسه الگوریتم پیشنهادی با IAPS

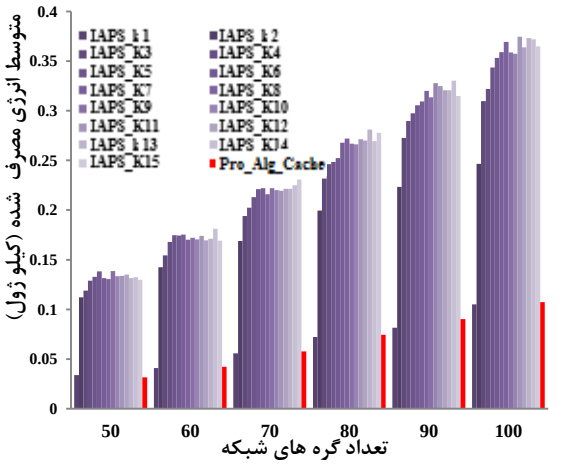
همان‌طورکه در نتایج قسمت ۵-۱ ملاحظه گردید الگوریتم پیشنهادی به همراه مکانیزم Cache (Pro_Alg_Cache) کارایی نسبتا بهتری نسبت به الگوریتم پیشنهادی بدون مکانیزم Cache (Pro_Alg) دارد، لذا در این قسمت ازمایش‌هایی جهت مقایسه الگوریتم پیشنهادی همراه با مکانیزم Cache (Pro_Alg_Cache) با الگوریتم جستجوی بهبودیافته احتمال‌تطبیقی با $k$ واکر از واکر ۱ تا واکر ۱۵( IAPS_K1تا IAPS_K15)، اورده شده است. همواره عملکرد الگوریتم پیشنهادی، بهتر از الگوریتم‌های IAPS می‌باشد زیرا، در الگوریتم پیشنهادی، انتخاب همسایه‌ها برای جلورانی پیام-ها، هوشمندانه است. درحالی‌که در IAPSها، همواره از $k$ واکر، برای ارسال درخواست به همسایه‌ها استفاده می‌شود. هر چند در IAPS، در جلورانی درخواست به همسایه ها $k$ عدد از بهترین همسایه‌ها، انتخاب می‌شود اما درخواست‌ها تا حدی کورکورانه به $k$ بهترین همسایه‌ی مجاور گره درخواست دهنده، در شبکه جلورانی می‌شود اما در الگوریتم پیشنهادی، درخواست‌ها، صرفا به همسایه-هایی جلورانی می‌شوند که با احتمال بیشتری با توجه به تجربه-های پیشین و شرایط کنونی شبکه، ممکن است منجر به یافتن منبع درخواستی شوند.

شکل (۱۱) درصد نرخ موفقیت را با افزایش تعداد گره‌های شبکه نشان می‌دهد. نرخ موفقیت الگوریتم پیشنهادی در ابتدا زیاد و سپس کم می‌شود ولی همیشه نسبت به هر ۱۵ نمودار IAPS بهتر می‌باشد زیرا با توجه به شرایط موجود در شبکه مانند سطح انرژی و میزان پایداری و سرعت حرکت و ورود و خروج گره‌ها در شبکه و از طریق یادگیری، در زمان اجرای آزمایش، همسایه‌های مناسبی که به یافت منبع در خواستی منجر می‌شوند، برای جلورانی درخواست، انتخاب می‌کند و علاوه بر این استفاده از جداول Cache نیز در موفقیت تاثیرگزار است. ولی IAPS همواره درخواست‌های خود را به k همسایه بهتر خود ارسال می‌کند. لذا الگوریتم پیشنهادی هوشمندانه‌تر رفتار می‌کند و موفق تر نیز می-باشد.



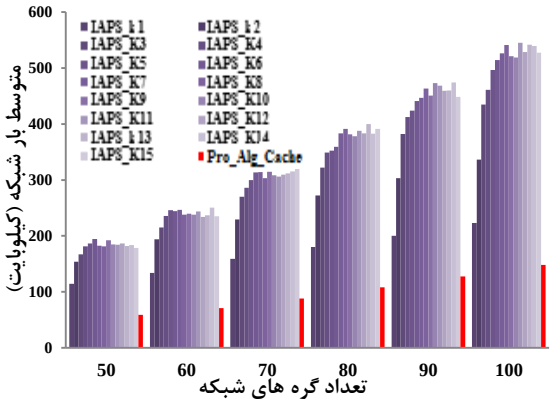

**شکل ۱۱: نرخ موفقیت به ازای افزایش تعداد گره‌های شبکه**

در نمودار شکل (۱۲) میزان انرژی مصرف شده با افزایش گره‌های شبکه نشان داده شده است. ملاحظه می‌شود که به‌دلیل اینکه الگوریتم پیشنهادی به مسیرهایی که انرژی بهینه دارند و منبع درخواستی را کشف می‌کنند پاداش می‌هد بنابراین انرژی گره‌ها بیهوده مصرف نمی‌گردد و فقط گره‌هایی که با احتمال بیشتری شامل منبع هستند در جستجو شرکت می‌کنند. درحالیکه در الگوریتم IAPS صرفا درخواست همواره به $k$ همسایه‌ای جلورانی می‌شود که قبلا در جستجوها موفق بوده‌اند و شرایط جاری گرههای شبکه را در نظر نمی‌گیرد، لذا انرژی بیشتری در حین جستجو مصرف می‌کند و هرچه $k$ بزرگتر باشد به‌علت شرکت همسایه‌های بیشتر انرژی بیشتری مصرف می‌گردد. از طرفی با استفاده از جداول Cache چون سریعتر منبع درخواستی یافت می‌شود، کمتر انرژی مصرف می‌شود.

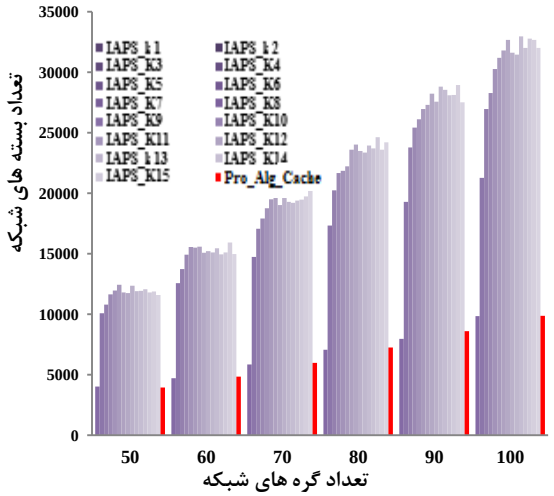



**شکل ۱۲: متوسط انرژی مصرفی به ازای افزایش تعداد گره‌های شبکه**

شکل (۱۳) مقایسه‌ای بر مبنای بار تحمیل‌شده به شبکه و شکل (۱۴) مقایسه‌ای بر مبنای تعداد بسته‌های تولیدشده در شبکه در حین جستجوی منبع درخواستی، ناشی از افزایش گره‌های شبکه را نشان می‌دهد. چون Pro_Alg_Cache مبتنی بر یادگیری است و صرفا درخواست‌های خود را به گره‌هایی که در جستجوهای قبلی موفق‌تر بوده‌اند ارسال می‌کند لذا نسبت به IAPSها بسته‌های کم-تری تولید می‌کند و در شبکه ارسال می نماید در نتیجه بار کم-تری به شبکه تحمیل می‌کند و از طرفی استفاده از جداول Cache نیز دراین امر موثر بوده است. در صورتی که الگوریتم IAPS همواره به $k$ همسایه خود درخواست را ارسال می‌نماید. همان-طورکه ملاحظه می‌شود هرچه $k$ در IAPSها بزرگتر باشد چون درخواست به همسایه‌های بیشتری جلورانی می‌شود، تعداد بسته-های بیشتری تولیدشده و بار بیشتری به شبکه تحمیل می‌شود.




**شکل ۱۳: متوسط بار شبکه به ازای افزایش تعداد گره‌های شبکه**




**شکل ۱۴: متوسط تعداد بسته ای ارسال شده به شبکه ازای افزایش تعداد گره‌های شبکه**

## ۱۰- نتیجه‌گیری

در الگوریتم ارائه شده در این مقاله یک روش جستجوی منبع خودتطبیق و یادگیرنده در شبکه نظیربه‌نظیر سیار معرفی گردید. از اتوماتای یادگیر برای انتخاب هوشمندانه همسایه‌ها برای جلورانی پیام پرس‌وجو استفاده شده است که سبب می‌شود تا همسایگان مفیدتر پیام پرس‌وجو را دریافت کنند. علاوه براین منبع از بهترین مسیر به گره‌ی درخواست‌دهنده انتقال یابد. نتایج شبیه‌سازی نشان می‌دهد که در الگوریتم پیشنهادی به‌علت پاداش دادن به همسایه‌هایی که در مراحل قبلی جستجو، موفق‌تر بوده‌اند، درخواست را به نواحی از شبکه که احتمال وجود منبع در آنجا بیشتر است، جلورانی می‌کند. در مقایسه‌ای که با الگوریتم‌های شایعه‌پراکنی-توازن‌بار (Gossiping_LB) و جستجوی بهبودیافته احتمال تطبیقی (IAPS)، تحت سناریوی افزایش تعداد گره‌های شبکه صورت گرفت، مشخص شد که الگوریتم پیشنهادی کارایی بهتری دارد. همچنین بدلیل یادگیری گره‌ها و با دخیل نمودن ویژگی‌های شبکه مانند انرژی، سرعت حرکت، تعداد دفعات ورود و خروج از شبکه و مدت زمان حضور گره‌ها در شبکه، تعداد گره‌های کمتری در فرآیند جستجو درگیر می‌شوند که سبب کاهش مصرف انرژی شبکه و افزایش نرخ موفقیت و کاهش بار تحمیل‌شده به شبکه می‌گردد.

## مراجع


[1] Huang,W., Nahrstedt, K., Wu, B., Message propagation in adhoc based proximity mobile social networks, In Proceeding 8th IEEE International Conference on Pervasive Computing and Communications Workshops(PERCOM Workshops), 2010, 114-146.

[2] Tsai,F., Han, W., Xu, J., Chua, H., Design and development of a mobile peer-to-peer social networking application, Journal Elsevier of Expert System and Appllication, vol. 36, 2009, 11077–11087.

[3] Borg J., A, comparative study of ad hoc & peer to peer networks, Master's thesis, University College London, 2003.

[4] Ding G., Bhargava B., Peer-to-peer file-sharing over mobile ad hoc networks, In Proceeding 2th IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004, 104-108.

[5] Franciscani F.P., Vasconcelos M .A, et al, Peer-to-peer over ad-hoc networks configuration algorithms, In proceedings 17th IEEE International Parallel and Distributed Processing Symposium, 2003.

[6] Hu Y .C, Das S .M, Pucha H., Exploiting the Synergy between Peer-to-Peer and Mobile Ad Hoc Networks, In Proceeding HotOS-IX :Ninth Workshop on Hot Topics in Operating Systems, Vol. 9, 2003, 7-7.

[7] Xu, B., Wolfson, O., Data Management in Mobile Peer-to-Peer Networks, In Proceeding of Databases, Information Systems and Peer-to-Peer Computing (DBISP2P), 2004, 1-15.

[8] Huang, Z., Jensen,C., Lu, H., Ooi,B., Skyline Queries Against Mobile Lightweight Devices in MANETs, In Proceeding of

H.J., (2008), Tribler: a social-based peer-to-peer system, Concurrency and Computation, Vol. 20, No. 2, 120-127.

[29] Zhang, X., Liu, J., Li, B., Yum, T.S.P., (2005), Cool Streaming/DONet: a data-driven overlay network for efficient live media streaming, In Proceedings of IEEE Infocom, No. 3, 13–17.

[30] Haas, Z.J., Halpern, J.Y., Li, L., (2006), Gossip-based ad hoc routing, In Proceeding IEEE/ACM Transactions on Networking (TON), Vol. 14, No. 3, 485-491.

[31] Hayashi, H., Hara, T., Nishio., S., (2003), Cache invalidation for updated data in ad hoc net-works, In Proceeding International Conference on Cooperative Information Systems (CoopIS'03), Vol. 2888, 516-535.

[32] Verma, S., Ooi, W., (2005), Controlling Gossip Protocol Infection Pattern Using Adaptive Fanout, In Proceeding 25th IEEE International Conference on Distributed Computing Systems, 665 – 674.

[33] Wolfson, O., Xu, B., Yin, H., Cao, H, (2006), Search-and-Discover in Mobile P2P Network Databases, In Proceedings of the 26th IEEE International Conference on Distributed Computing Systems (ICDCS'06), 65-70.

[34] Papadopouli, M., Schulzrinne, H., (2001), Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices, In Proceeding MobiHoc '01 of the 2nd ACM international symposium on Mobile ad hoc networking & computing, 117-127.

[35] Gao, G., Li, R., He, H., Xu, Z., Distributed cashing in structured peer-to-peer file sharing networks, Computer and Electrical Engineering, Elsevier Journal, Vol. 40, 2013, 688-703.

[36] Luo X, Qin Z, Geng J, Luo J. IAC: Interest-aware caching for unstructured P2P, in SKG, IEEE Computer Society, 2006, 58.

[37] Najim, K., Poznyak, A.S., Learning automata :theory and application, in proceeding of the Tarrytown, New York, Elsevier Science Publishing Ltd., 1994.

[38] M.Kumpati, Thathachar and Narendra, Learning Automata-A Survey, In Proceeding of IEEE Transactions on systems, Man and Cybernetics, 1974.

[39] Babaei, H., Fathy, M., Berangi, R., The impact of mobility models on the performance of P2P content discovery protocols over mobile ad hoc networks, Journal of Springer, vol 6, 2012, 344-348.

[40] Cisco aironet 350 series client adapters. Available from: http//:www.cisco.com/en/US/prod/collateral/wireless/ps6442/ps4555/ps581product_data_sheet09186a00801ebc29.html, 2008.

International Conference on Data Engineering (ICDE), 2006, 66-74.

[9] Shirky,C., What Is P۲P And What Isn't," OpenP۲P.com, 2000.

[10] Chu, J., Labonte, K., Levine, B., Availability and locality measurements of peer-to-peer file systems. In Proceeding of ITCom: Scalability and Traffic Control in IP Networks SPIE, 2002.

[11] Tung, Y., Lin, K., Location-Assisted Energy-Efficient Content Search for Mobile Peer-to-Peer Networks, In Procedding 7th International Workshop on Mobile Peer-to-Peer Computing IEEE, 2011, 477-482.

[12] Kalogeraki, V., Gunopulos, D., Zeinalipour-Yazti, D., A Local Search Mechanism for Peer-to-Peer Networks, In Proceeding of 11th international conference on Information and knowledge management CIKM, 2002, 300-307.

[13] http://www.gnutella.com, Gnutella website.

[14] Gkantsidis, C., Mihail, M., Saberi, A., Random Walks in Peer-to-Peer Networks, In Proceeding INFOCOM Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies , Vol 1, 2004, 1-12.

[15] Hora, D., Macedo, D.F., Oliveira, L. B., and et al. Enhancing peer-to-peer content discovery techniques over mobile ad hoc networks, Journal of Elsevier Computer communication, Vol.32, 2009, pp 1445-1459.

[16] Cao, H., Wolfson, O., Xu, B., Yin, H., (2005), MOBI-DIC: MOBIle Discovery of local Resource in Peer-to-Peer Wireless Network, In Proceeding IEEE Computer Society Technical Committee Data Engineering, 232-240.

[17] Lv, Q., Cao, P., Cohen, E., Li, K., Shenker, S., Search and replication in unstructured peer-to-peer network, In Proceeding of the 16th ACM International Conference on Supercomputing (ICS'02), ACM, 2002, 329–350.

[18] Tsoumakos, D., Roussopoulos, N., Adaptive Probabilistic (APS) for Peer-to-Peer Networks, In Proceesings of Peer-to-Peer computing, 2003, 102-109.

[19] Tsoumakos, D., Roussopoulos,N., Evaluation of ad hoc routing protocols under a peer-to-peer application, In Proceeding of the 3rd International Conference on Peer-to-Peer Computing (P2P), 2003, 102–109.

[20] Tsoumakos, D., Roussopoulos, N., Adaptive Probabilistic (APS) for Peer-to-Peer Networks, In Proceedings of the 3rd international conference on peer-to-peer computing, 2003, 102-109.

[21] Shojafar, M., Abawajy, J., Delkhah, Z., Pooranian, Z., Abraham, A., An efficient and distributed file search in unstructured peer-to-peer networks, Journal of Peer-to-peer Network Application, Springer Journal, pp. 356-362, 2013.

[22] Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H., Swinehart, D., Terry, D., (1987), Epidemic algorithms for replicated database maintenance, In Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing (PODC), 1–12.

[23] Eugster, P.T., Guerraoui, R., Kermarrec, A.M., Massoulié, L., (2004),From epidemics to distributed computing, In Proceeding IEEE Computer, Vol. 37, No. 5, 60–67.

[24] Jelasity, M., Babaoglu, O., Man, T., (2009), Gossip-based fast overlay topology construction, Computer Networks, Vol. 53, No. 13, 2321–2339.

[25] Kermarrec, A.M., Massoulié, L., Ganesh, A.J., (2003), Probabilistic reliable dissemination in large-scale systems, In Proceeding IEEE Transactions on Parallel and Distributed systems, 248–258.

[26] Jelasity, M., Montresor, A., Babaoglu, O., (2005), Gossip-based aggregation in large dynamic networks, ACM Transactions on Computer Systems (TOCS), Vol. 23, No. 3, 245 -252.

[27] Renesse, R. V., Minsky, Y., Hayden, M., (1998), A gossip-style failure detection service, In Proceedings of Middleware'98, IFIP International Conference on Distributed Systems Platforms and Open Distributed Processing, 55–70.

[28] Pouwelse, J.A., Garbacki, P., Wang, J., Bakker, A., Yang, J., Iosup, A., Epema, D.H., Reinders, M., Van Steen, M.R., Sips,

# A Novel Learning based Algorithm for Resource Searching in Peer-to-Peer over Mobile Ad-hoc Networks

Somaye Khaleghi[1*], Karim Faez[2]
1*- Corresponding Author: Department of Computer Engineering and Information Technology, Qazvin Islamic Azad University, Barajin, Qazvin, Qazvin, Iran.
2- Department of Electrical Engineering, Amirkabir University, Valiasr, Tehran, Tehran, Iran.
[1*]SomayeKhaleghi@gmail.com, [2]kfaez@aut.ac.ir

**Abstract-** Peer to peer network is a set of self-adaptive and identic nodes that cooperate together without the central coordinator. Mobile peer to peer network is caused from the implementation of peer to peer network over mobile Ad-hoc networks' platform. The distributed structure of these networks reinforces the resource searching. The requests in these networks are often propagated randomly and broadcast. Thus, designing a self- adavptive mechanism that sends the request intentionally to the network is crucial, to be able to discover the demanded source of the request with the maximum amount of success rate and minimum amount of energy and traffic load.
In this research is presented solution with the aim of improving serach on the mobile peer to peer network.
Learning automata are adaptive decision making units which run in random environments that learn the optimal action through repeated interaction with its environment.In the proposed algorithm, each node having a message sends a search request to some more useful neighbors. Correspondingly the best route from the source node will transmit the requested resource. Moreover Because of saving result in neighbourse cache table, Neighbourse experince are effective in future forwards. The simulation results of the proposed algorithm and comparison with the "Gossiping Load Balancing alghorithm" algorithm and "Imporoved Adaptive Probablitistic Search", show that avoiding selecting random neighbors for forward request and considering the history of each node using the results stored in cache tables, reduce energy consumption, the number of sended and droped network packets , and the networkload. Since, due to the influence of neighboring nodes in proceeding, the request message is sent to the neighbors who have been more successful in previous searches, the success rate of the network also increases.

**Keywords**- Mobile Ad-hoc peer to peer Networks, Gossiping –LB , AIPS, Learning Automata, Resource Search .