

# SID



سرویس های ویژه



سرویس ترجمه تخصصی



کارگاه های آموزشی



بلاگ مرکز اطلاعات علمی



سامانه ویراستاری STES



فیلم های آموزشی

## کارگاه های آموزشی مرکز اطلاعات علمی جهاد دانشگاهی

کارگاه آنلاین  
بررسی مقابله ای متون (مقدماتی)

کارگاه آنلاین  
پروپوزال نویسی و پایان نامه نویسی

کارگاه آنلاین آشنایی با پایگاه های اطلاعات علمی بین المللی و ترند های جستجو



# ارائه یک شیوه مسیریابی برای کاهش زمان و فضای مورد نیاز در شبیه‌سازی

## شبکه‌های حسگر بی‌سیم

احسان خراطی

دانشگاه آزاد اسلامی واحد اراک

[e-kharati@iau-arak.ac.ir](mailto:e-kharati@iau-arak.ac.ir)

شبیه‌سازی شده، مدل‌های جریان و بسته‌های ورودی را در هر مسیریاب، با هم ترکیب کرده و وارد صف کرد. [۵] از این رو، یک محاسبه مسیریابی صحیح و شیوه ذخیره و جستجوی مناسب می‌تواند مقیاس شبیه‌سازی را ارتقا بخشد و کارایی را افزایش دهد. لذا دو پارامتر مهم برای ارزیابی یک روش مسیریابی، حافظه مورد نیاز برای جداول مسیریابی و زمان پردازشگر برای جستجوی حالات مسیریابی مورد استفاده است. دستیابی به تعادل بهتر میان این دو پارامتر، برای هر شیوه مسیریابی بسیار با اهمیت است.

روش پیشنهادی، مزایای بعضی از شیوه‌های معروف و کارآمد موجود مسیریابی را دارد و تعداد درختان پوشا را به‌عنوان جدول پایه مسیریابی برای اکثر کوچکترین مسیرها حفظ می‌کند و برای مسیرهایی که در محدوده هیچ یک از درختان پوشا قرار نمی‌گیرند، از مسیریابی Nix\_Vector برای محاسبه آنها برحسب تقاضا استفاده می‌کند. با ارزیابی فضای ذخیره‌سازی و زمان مورد نیاز پردازشگر برای جستجوی حالات ذخیره، به توازن بهتری میان آنها دست می‌یابیم.

ادامه این مقاله به این شکل سازماندهی شده است که بخش دوم شامل یک مرور کلی از بعضی از شیوه‌های مسیریابی موجود بوده و در بخش سوم، طراحی و پیاده‌سازی روش پیشنهادی بیان می‌شود. در بخش چهارم، نتایج تئوری و آزمایشات ارائه شده و در بخش پنجم، نتیجه‌گیری‌ها و تحقیقات آتی مطرح خواهد شد.

### ۲- تحقیقات و کارهای صورت گرفته

مسیریابی Flat [6]، کوتاه‌ترین مسیر را محاسبه می‌کند و یک جدول مسیریابی تولید می‌کند که هر گره، اطلاعات گره بعدی به سایر گره‌ها را دارد. لذا این روش مسیریابی، به فضای بزرگی برای حفظ جدول مسیریابی نیاز دارد. اگر  $N$  تعداد گره‌ها در شبکه باشد، پیچیدگی فضای مصرفی این روش،  $O(N^2)$  است. بنابراین یافتن یک گره همسایه و ایجاد یک جفت گره، پیچیدگی زمانی بسیار اندک  $O(1)$  را دارد. لذا کاربرد این مسیریابی هنگامی است که منابع حافظه نامحدود هستند.

الگوریتم مسیریابی در [7]، اندازه جداول مسیریابی برای مسیر بین دو گره در یک درخت پوشا را کاهش می‌دهد و شبیه‌ساز، به‌جای یک جدول مسیریابی کلی، یک درخت پوشا را به‌عنوان یک جدول مسیریابی که اندازه آن فقط  $O(N)$  است ذخیره کرده و در نتیجه پیچیدگی عمل جستجو  $O(\log N)$  خواهد بود. این مسیریابی توازن بهتری میان فضای ذخیره‌سازی و زمان پردازشگر برای جستجوی مسیریابی ارائه می‌دهد؛

**چکیده:** شبیه‌سازی ابزاری بسیار مهم برای ارزیابی شبکه‌های بزرگ است. مسیریابی یا مسیرگزینی یکی از مهمترین فاکتورهای موثر بر مقیاس یا اندازه و کارایی شبیه‌سازی است. در این مقاله یک روش جدید برای محاسبات مسیریابی، ذخیره‌سازی و جستجو ارائه می‌شود. این روش تعداد درختان پوشا را به‌عنوان جدول مسیریابی حفظ کرده و از مسیریابی Nix\_Vector برای محاسبه حالت‌هایی از مسیریابی برحسب تقاضا که در محدوده درختان پوشا نیست، استفاده می‌کند. هدف ما در مسیریابی پیشنهادی، ایجاد توازن بهینه بین فضای ذخیره‌سازی جداول مسیریابی و زمان پردازشگر برای جستجو و مراجعه مسیریابی است. نتایج شبیه‌سازی نشان می‌دهند که با اجرای مسیریابی پیشنهادی، فضای ذخیره‌سازی فقط به میزان یک درصد بیشتر از فضای ذخیره‌سازی مسیریابی Nix\_Vector بوده و زمان شبیه‌سازی را می‌تواند حدود 85% در مقایسه با شیوه مسیریابی Nix\_Vector کاهش دهد.

**کلمات کلیدی:** شبکه‌های حسگر بی‌سیم، مسیریابی، زمان جستجو، فضای ذخیره‌سازی، درخت پوشا و جدول پیچیدگی

### ۱- مقدمه

در دهه گذشته با رشد سریع اینترنت و پیچیدگی و تغییرات شبکه‌های حسگر بی‌سیم، شبیه‌سازی نقش حیاتی در شناخت رفتار شبکه ایفا می‌کند. مقیاس و کارایی، دو فاکتور مهم عملکرد شبیه‌سازی است که محققان سعی دارند مقیاس را افزایش داده و کارایی شبیه‌سازی را با محدود کردن منابع موجود بر روی کامپیوترها ارتقا دهند. به‌عنوان مثال، برای کاهش از بین رفتن بسته‌ها از صف‌بندی در گره‌های میانی استفاده می‌شود [۱]. مسیریابی مهمترین فاکتور موثر روی اندازه و کارایی شبیه‌سازی است. مهمترین محدودیت در شبیه‌سازی شبکه‌های حسگر بی‌سیم بزرگ، حافظه مورد نیاز برای حالت‌های مسیریابی است. عامل دیگر محدودیت، جستجو حالت‌های مسیریابی است که زمان زیادی از زمان پردازشگر را در هنگام شبیه‌سازی تلف می‌کند و به شدت بر کارایی شبیه‌سازی موثر است. به‌عنوان مثال، می‌توان بسته‌های مسیریابی را فقط به گره‌های همسایه‌ای که در طی مدت خاصی به درخواست مسیریابی پاسخ دهند، ارسال کرد. [۲] و یا منابع محاسباتی در رویدادگسسته را توزیع و موازی کرد تا باعث افزایش نرخ پردازش مسیریابی و اشکال‌زدای سریع‌تر و با مقیاس گسترده‌تر در یک شبیه‌ساز شبکه حسگر گردد. [۳و۴] همچنین می‌توان در هر قسمت از شبکه

### ۳- روش پیشنهادی

در شبیه‌سازی شبکه‌های حسگر بی‌سیم، دو فاکتور اصلی در هر روش مسیریابی، فضای مورد نیاز برای جداول مسیریابی و زمان پردازشگر مورد نیاز برای جستجو است که با یکدیگر در تناقض هستند. زیرا برای جستجوی سریع‌تر، نیاز به فضای بزرگتر برای ذخیره اطلاعات بیشتر است. لذا لازم است به توازن بیشتری بین این دو فاکتور رسید.

بنابراین در این مقاله ما می‌خواهیم فضای ذخیره شده، حاوی بیشترین اطلاعات مورد نیاز باشد تا زمان جستجو سریع‌تر گردد و کارایی روش مسیریابی افزایش یابد. برای پیاده‌سازی این ایده، از ساختمان داده درخت پوشا استفاده می‌شود؛ زیرا برای ذخیره اطلاعات مسیریابی، پیچیدگی فضایی آن کم و برابر  $O(N)$  بوده و لذا فضای مورد نیاز برای جداول مسیریابی را کاهش می‌دهد. همچنین زمان جستجوی حالات مسیریابی میان هر دو جفت گره در درخت پوشا اندک است. از آنجا که شبکه‌هایی نظیر اینترنت، عموماً متراکم نیستند، یک درخت پوشا درصد نسبتاً بیشتری از کوتاه‌ترین مسیرهای بین بردارها را دربر خواهد گرفت. هر درخت پوشا حاوی یک زیرمجموعه مختلف از کوتاه‌ترین مسیرهای میان بردارهاست. لذا با افزایش تعداد درخت‌ها، تعداد مسیرهای کوتاه نیز افزایش می‌یابد. بنابراین برای پوشش تمام مسیرهای کوتاه، نیاز به  $N$  درخت خواهد بود که فضای ذخیره‌سازی آنها  $O(N^2)$  خواهد شد.

(مانند جدول مسیریابی عمومی در روش Flat)

همچنین برای ذخیره‌سازی اطلاعات مسیریابی کوتاه‌ترین مسیرهایی که در هیچ درخت پوشایی قرار نمی‌گیرند، به یک حافظه نیاز است. لذا از روش مسیریابی Nix\_Vector برای محاسبه آنها می‌توان استفاده کرد. در این حافظه، اطلاعات آدرس‌های منبع و مقصد ذخیره شده تا با انتخاب فهرست‌های هم‌جوار گراف، بتوان شاخص گره مبدأ را در زمان  $O(1)$  یافت. همچنین اگر تعداد همسایه‌های گره مبدأ  $m$  باشد، زمان لازم برای یافتن سریع‌ترین گره مقصد،  $O(\lg m)$  است؛ زیرا لیست‌های هم‌جوار هر گره به وسیله شاخص‌های گره‌ها مرتب شده‌اند و با جستجوی دودویی می‌توان شاخص گره مقصد را یافت. همچنین چون گراف‌های شبکه، غیرجهت دار هستند، اطلاعات یک جفت گره، فقط یکبار در این حافظه ذخیره می‌شود، که در نتیجه فضای مورد نیاز ذخیره‌سازی را کاهش می‌دهد.

براساس این مطالب، زمان جستجوی برای شیوه مسیریابی پیشنهادی برابر مجموع زمان جستجوی این حافظه و زمان جستجوی هر درخت چندگانه و یا آرایه Nix\_Vector است. لذا اگر جستجو  $T_{\text{جستجو}}$ ، متوسط زمان جستجوی مسیریابی روش پیشنهادی و  $T_{\text{cache}}$ ، زمان جستجو برای حافظه و  $T_{\text{MTree}}$ ، زمان جستجو برای درخت پوشا و  $T_{\text{Nix\_Vector}}$  زمان جستجو برای بردار Nix\_Vector و پوشش  $P$ ، درصد پوشش‌دهی درخت پوشا باشد؛ خواهیم داشت:

$$T_{\text{جستجو}} = T_{\text{cache}} + P \times T_{\text{MTree}} + (1-P) \times T_{\text{Nix\_Vector}} \quad (1)$$

که  $T_{\text{Nix\_Vector}}$  خیلی از  $T_{\text{MTree}}$  بزرگتر است و هرچه تعداد درختان پوشا بیشتر شود، درصد پوشش‌دهی نیز افزایش یافته و در نتیجه زمان جستجو کاهش می‌یابد. همچنین با افزایش تعداد درختان پوشا، زمان جستجو افزایش می‌یابد.

اما چون تمام مسیرهای کوتاه، در محدوده درخت پوشا قرار نمی‌گیرد، نیاز به محاسبه جزئی کوتاه‌ترین مسیرها دارد.

الگوریتم مسیریابی در [8] با نام STree\_Flat، همه مسیرهای کلی را شامل شده و جدول مسیریابی آن به صورت ترکیبی از یک جدول مسیریابی درخت پوشا و جدول مسیریابی جزئی است. در این روش، اگر یک ورودی جدید در جدول مسیریابی کلی را بتوان هنگام جستجوی یک حالت مسیریابی، به وسیله درخت پوشا نمایش داد، آنگاه یک ورودی جدید به وسیله جدول مسیریابی درخت پوشا دوباره ارائه می‌شود و در غیر این صورت، یک ورودی جدید در جدول مسیریابی کلی ارائه می‌شود. پیچیدگی فضا و زمان در این مسیریابی، به تعداد ورودی در محدوده جدول مسیریابی درخت پوشا بستگی دارد که به آن درصد پوشش‌دهی درخت پوشا یا  $p$  می‌گویند. پیچیدگی فضای مورد نیاز در این روش مسیریابی،  $O(N + (1-p) \times N^2)$  است و بدترین زمان جستجوی آن  $O(\lg N + (1-p) \times N^2)$  است. لذا این شیوه مسیریابی، فضا را به مقدار زیادی کاهش داده اما زمان شبیه‌سازی را بسیار افزایش می‌دهد و بنابراین توازن در این روش مسیریابی، حتی بدتر از مسیریابی Flat است.

الگوریتم مسیریابی در [9] با نام MTree\_Flat، مسیریابی STree\_Flat را از طریق ذخیره‌سازی جدول مسیریابی جزئی و درختان پوشای چندگانه، برای ایجاد جدول مسیریابی عمومی توسعه داده است. تفاوت این مسیریابی با STree\_Flat، هنگام جستجوی یک حالت مسیریابی است. در این مسیریابی، اگر این حالت در محدوده یکی از درختان پوشا باشد، باید تمام آنها جستجو شوند و یکی از آنها که کوتاه‌ترین مسیر را دارد، انتخاب شود. لذا پیچیدگی زمان و فضای این مسیریابی، به درصد پوشش‌دهی بستگی دارد. اگر  $p$  درصد پوشش‌دهی و  $k$  تعداد درختان پوشا باشد؛ فضای مورد نیاز برابر است با  $O(k \times N + (1-p) \times N^2)$  و بدترین زمان محاسبه برای یک جستجو  $O(k + \lg N + (1-p) \times N^2)$  است. لذا درصد پوشش‌دهی این روش، بیش از مسیریابی STree\_Flat است و پیچیدگی زمان و فضای کمتر و با توازن بهتری دارد. اما در این روش مسیریابی، اگر اندازه شبکه افزایش یابد، تعداد حالات جستجویی که در محدوده درخت پوشا قرار نمی‌گیرند، به شدت افزایش یافته و در نتیجه به فضا و زمان بیشتری نیاز خواهد بود.

الگوریتم مسیریابی در [10, 11] با نام Nix\_Vector، به جداول مسیریابی نیاز نداشته و به جای آن، هر زمان که بسته جدیدی تولید شود، حالات مسیریابی را براساس تقاضا محاسبه می‌کند. لذا این روش مسیریابی برای جستجو و آغاز شبیه‌سازی نیاز به فضایی برای ذخیره‌سازی جداول ندارد و کمترین پیچیدگی فضایی را دارد. اما هر زمانی که بسته به گره شبیه‌سازی وارد شود، باید از طریق الگوریتم جستجوی سطحی، توپولوژی را پیمایش کند تا اطلاعات کوتاه‌ترین مسیر را محاسبه نماید. اگر  $N$  تعداد گره‌ها و  $E$  تعداد لینک‌ها باشد، پیچیدگی زمانی آن  $O(N+E)$  است. بنابراین پیچیدگی زمانی این روش مسیریابی طولانی بوده و برای شبیه‌سازی شبکه‌های بزرگ و کاربردهای پیچیده نظیر انتشار مناسب نیست.

مکانیزم‌ها	فضای ذخیره‌سازی	زمان جستجو
Flat	$O(N^2)$	$O(1)$
STree_Flat	$O(N+(1-p)\times N^2)$	$O(\lg N+(1-p)\times N^2)$
MTree_Flat	$O(k\times N+(1-p)\times N^2)$	$O(k\times \lg N+(1-p)\times N^2)$
Nix_Vector	0	$O(N+E)$
روش پیشنهادی	$O(k\times N+(1-p)\times N^2/4)$	$O(\lg N+p\times k\times \lg N+(1-p)\times (N+E))$

#### جدول ۱: فضا و زمان مورد نیاز مکانیسم‌های مسیریابی.

برای بررسی کارایی، روش مسیریابی پیشنهادی را در شبیه‌ساز NS-2 شبیه‌سازی کرده و نتایج را با سایر روش‌های مسیریابی مقایسه می‌کنیم. این آزمایشات در یک کامپیوتر پنتیوم ۴ با پردازشگر ۱ گیگا هرتز و حافظه اصلی، ۲GB و سیستم عامل Xp اجرا شده است. توپولوژی‌های شبکه با استفاده از [12] NEM که یک تولیدکننده توپولوژی مصنوعی است ایجاد می‌شود که قابلیت تنظیم پارامترهای اندازه شبکه و متوسط درجه گره را دارد. در هر توپولوژی شبیه‌سازی شده، ۱۰۰۰۰ اتصال TCP بین جفت گره‌های تصادفی ایجاد شده و ۱۰۴۰ بیت در هر اتصال TCP ارسال می‌شود. هر آزمایش ۵ بار تکرار می‌شود تا میانگین آنها به‌عنوان نتیجه آزمایش در نظر گرفته شود.

کارایی شیوه مسیریابی پیشنهادی به درصدی از حالات مسیریابی که تحت پوشش درخت پوشا است، بستگی دارد. ما درصد پوشش دهی و تعداد درختان پوشا را تحت شرایط معادله (۴) و در توپولوژی‌هایی با اندازه‌های مختلف و متوسط درجه گره متفاوت، محاسبه می‌کنیم. نمودار ۱، درصد پوشش دهی اندازه توپولوژی‌های مختلف را نشان داده و نمودار ۲، تعداد درختان پوشا برای دستیابی به درصد پوشش دهی را نشان می‌دهد. در این آزمایشات، اندازه توپولوژی‌ها از ۲۰۰ تا ۳۰۰۰ گره با متوسط درجه گره ۲/۶ در حال تغییر است. مشاهده می‌شود که با افزایش اندازه توپولوژی، درصد پوشش دهی کاهش می‌یابد و تعداد درختان پوشا افزایش می‌یابد.

نمودار ۳ تغییرات درصد پوشش دهی با متوسط درجه‌های مختلف گره در یک توپولوژی ثابت با ۱۰۰۰ گره را نشان داده و نمودار ۴، تعداد درختان پوشا برای رسیدن به متوسط درجه گره در نمودار ۳ را نشان می‌دهد. متوسط درجه گره در این دو نمودار از ۴ تا ۱۴ در حال تغییر است. مشاهده می‌شود که با افزایش متوسط درجه گره، درصد پوشش دهی کاهش یافته و تعداد درختان پوشا برای رسیدن به درصد پوشش دهی افزایش می‌یابند. در نتیجه توپولوژی‌هایی با اندازه بزرگتر و متوسط درجه گره بالاتر، به درختان پوشای بیشتری نیاز دارند و درصد پوشش دهی کمتری دارند. همچنین با افزایش متوسط درجه گره، تعداد کوتاه‌ترین مسیرهایی را که لبه‌های کمتری را با هم مشترک هستند، افزایش می‌یابد. لذا مجموعه درختان پوشا کمتری تحت پوشش حالت‌های مسیریابی در جدول مسیریابی عمومی به‌عنوان تعداد لبه‌ها در هر دخت پوشا قرار می‌گیرند. کارایی یک شیوه مسیریابی در شبیه‌سازی را می‌توان با ۳ پارامتر زمان قبل از محاسبه، مصرف حافظه و زمان شبیه‌سازی نشان داد. در این بخش، نتایج بررسی این پارامترها را با اندازه توپولوژی‌های مختلف و متوسط درجه گره متفاوت ارائه می‌شود. نمودار ۵، زمان قبل از محاسبه را با توپولوژی‌های مختلف نشان می‌دهد. درصد پوشش دهی و تعداد درختان پوشا مانند شکل‌های

حافظه مورد نیاز در روش مسیریابی پیشنهادی برابر مجموع فضای ذخیره برای حافظه و درخت پوشا است. یعنی اگر  $M_{Cache}$  و  $M_{MTree}$  به ترتیب حافظه مورد نیاز برای درخت پوشا و حافظه باشند، کل فضای مورد نیاز روش پیشنهادی به‌صورت زیر خواهد بود.

$$M_{روش\ پیشنهادی} = M_{MTree} + M_{Cache} \quad (2)$$

ارتباط میان  $M_{MTree}$  و  $M_{Cache}$  آن است که وقتی یک درخت پوشای بیشتر ایجاد می‌شود،  $M_{MTree}$  با تعداد مشخصی از حافظه افزایش می‌یابد اما  $M_{Cache}$  با تعداد متغیری از حافظه کاهش می‌یابد. هنگام ایجاد یک درخت جدید، درصد پوشش دهی و فضای  $M_{MTree}$  نیز افزایش یافته و  $M_{Cache}$  کاهش می‌یابد. لذا برای اینکه فضای مورد نیاز روش مسیریابی پیشنهادی را تا حد ممکن کاهش دهیم، لازم است که وقتی که درخت پوشای جدیدی تولید می‌شود، افزایش فضای  $M_{MTree}$  بیشتر از کاهش  $M_{Cache}$  باشد. یعنی:

$$\Delta M_{MTree} > \Delta M_{Cache} \quad (3)$$

یعنی با افزایش یک درخت پوشای جدید، سبب افزایش درصد پوشش دهی و پر شدن  $N$  واحد حافظه می‌شود و همچنین سبب کاهش  $M_{Cache}$  به اندازه  $\Delta P_{Coverage} \times N^2/2$  واحد خواهد شد. لذا معادله (۳) به‌صورت زیر تبدیل می‌گردد.

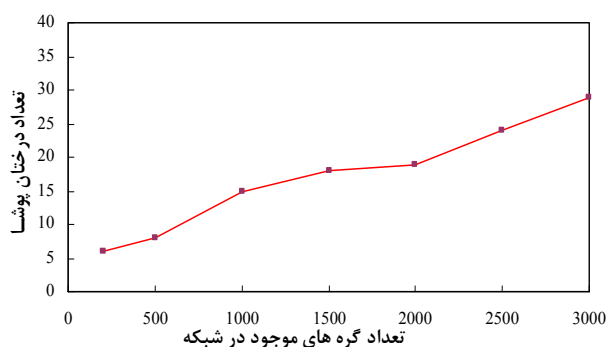
$$N > \Delta P_{Coverage} \times N^2/2 \Rightarrow \Delta P_{Coverage} < 2/N \quad (4)$$

لذا تحت این شرایط تعداد درختان پوشا به کمترین فضای لازم نیاز خواهند داشت. برای شبیه‌سازی شبکه‌های بی‌سیم، می‌توان ارتباط بین تعداد درختان پوشا و درصد پوشش دهی را به‌وسیله یک الگوریتم بازگشتی محاسبه و تعیین نمود.

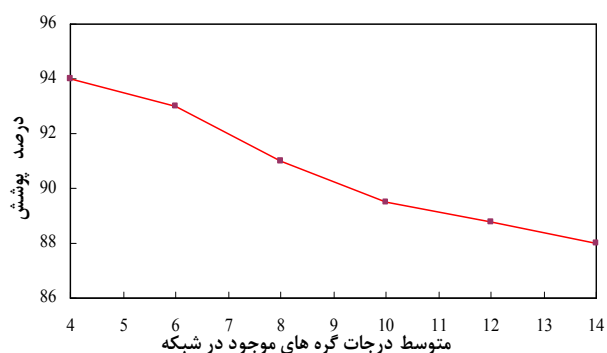
در روش مسیریابی پیشنهادی، شیوه یافتن یک حالت مسیریابی، براساس درختان پوشا یا Nix\_Vector است، لذا نمی‌توان هنگام ایجاد یک بسته جدید، بردار شاخص مجاور را محاسبه کرد و برای این منظور باید هنگامی که بسته در اولین گره بعدی بود، حافظه cache بررسی گردد. چون از شاخص درخت پوشا استفاده می‌شود، شبیه‌ساز باید شاخص را مانند بردار شاخص همسایه در مسیریابی Nix\_Vector، در بسته آغازین ذخیره کند. این بسته حاوی تمام اطلاعات گره‌های مبدا و مقصد است. لذا پس از آن، هر زمان که یک بسته بخواهد مسیریابی شود، شبیه‌ساز نیاز به محاسبه و بررسی cache و انتخاب کوتاه‌ترین مسیر در درخت پوشا را ندارد و در نتیجه زمان کمتری صرف می‌شود. برای تولید درخت پوشا، در روش مسیریابی پیشنهادی، گره‌ها را برحسب درجاتشان مرتب کرده و گره بعدی را با بالاترین درجه به‌عنوان ریشه درخت جدید انتخاب می‌کنیم. همچنین برای کاهش ارتفاع درخت پوشا، از الگوریتم BFS استفاده می‌شود. زیرا آزمایشات در [9] ثابت کرده که درختان پوشای چاق و کوتاه درصد بیشتری از کوتاه‌ترین مسیرها را پوشش می‌دهد.

#### ۴- ارزیابی کارایی و پیچیدگی

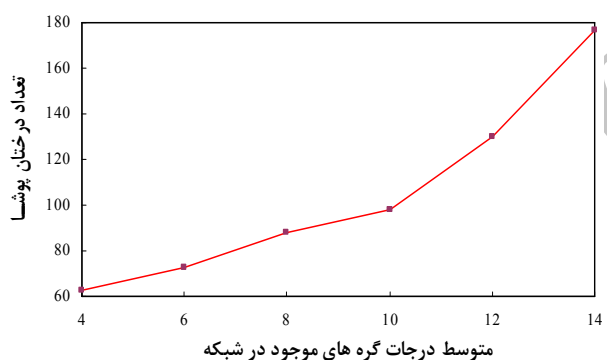
در جدول ۱ پیچیدگی فضا و زمان شیوه‌های مختلف مسیریابی که در بخش قبلی بیان شد، آمده است. از این جدول می‌توان نتیجه گرفت که پیچیدگی زمان و فضای روش پیشنهادی بهتر از MTree\_Flat و STree\_Flat و خیلی بهتر از پیچیدگی فضای Flat است و پیچیدگی زمانی کمتری از Nix\_Vector دارد.



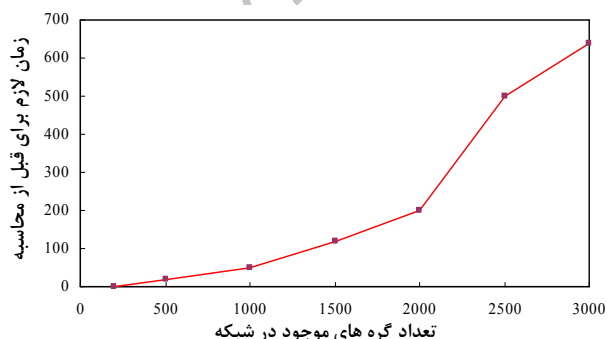
نمودار ۲: تعداد درختان پوشا برای دستیابی به درصد پوشش دهی در نمودار ۱



نمودار ۳: میزان تغییر درصد پوشش دهی با متوسط درجه گره متفاوت



نمودار ۴: تعداد درختان پوشا برای دستیابی به درصد پوشش دهی در نمودار ۳



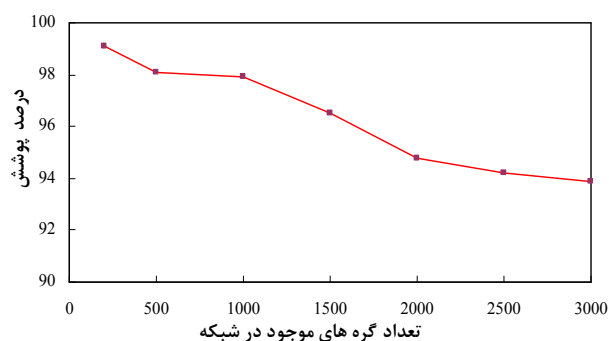
نمودار ۵: میزان تغییرات زمان پیش از محاسبه با تعداد گره های متفاوت

۱ و ۲ است. با افزایش اندازه توپولوژی، زمان قبل از محاسبه نیز افزایش می یابد. اما نسبت به زمان شبیه سازی، سریع تر است.

نتیجه اندازه گیری حافظه مصرفی شیوه های مختلف مسیریابی در نمودار ۶ آمده است. از آنجا که ذخیره سازی اندازه و لبه های توپولوژی های شبیه سازی، در تمام شیوه های مسیریابی یکسان است، تفاوت این شیوه ها در ذخیره سازی جدول مسیریابی است. مشاهده می شود که مسیریابی Nix\_Vector به کمترین حافظه نیاز دارد زیرا هیچ گونه جدول مسیریابی نیاز ندارد. روش مسیریابی پیشنهادی فقط یک درصد بیشتر از مسیریابی Nix\_Vector بوده و مکانیزم مسیریابی MTree\_Flat به کمی حافظه بیشتر از روش پیشنهادی نیاز دارد. استفاده از حافظه در روش های مسیریابی STree\_Flat و Flat از سایر روش ها بسیار بیشتر است، زیرا جدول مسیریابی آنها بزرگتر است.

شکل ۷ زمان شبیه سازی با اندازه توپولوژی های مختلف را نشان می دهد. تفاوت این شیوه ها در عمل جستجوی مسیر مناسب است. مشاهده می شود که مسیریابی STree\_Flat به ده برابر زمان شبیه سازی سایر روش ها نیاز دارد. زیرا با افزایش اندازه توپولوژی، تعداد حالات مسیریابی که یک درخت پوشا می تواند پوشش دهد، بسیار کوچک می شود و در هر جستجو، شبیه ساز باید حدود  $O(N^2)$  خانه جدول را جستجو کند. روش مسیریابی Flat کمترین زمان شبیه سازی را دارد. زمان روش مسیریابی پیشنهادی، حدود ۲۰ درصد بیشتر از Flat و ۸۵ درصد کمتر از مسیریابی Nix\_Vector است.

شکل ۸، زمان شبیه سازی شیوه های مختلف مسیریابی با متوسط درجات مختلف گره را نشان می دهد. چون زمان مسیریابی STree\_Flat بسیار بیشتر از سایر روش ها است، نتایج آن را نشان نمی دهیم. مشاهده می شود که زمان شبیه سازی مسیریابی پیشنهادی نزدیک به روش مسیریابی Flat است و از روش های مسیریابی MTree\_Flat و Nix\_Vector بسیار کمتر است. براساس نتایج آزمایش و ارزیابی مصرف حافظه و زمان شبیه سازی، روش مسیریابی پیشنهادی به مقادیر بهینه نزدیک است و توازن بهتری نسبت به سایر روش ها دارد.



نمودار ۱: میزان تغییر درصد پوشش دهی با تعداد گره های مختلف

## ۵- نتیجه‌گیری و تحقیقات آتی

مسیریابی یکی از مهمترین فاکتورهای موثر بر عملکرد شبیه‌سازی است. این مقاله یک روش مسیریابی، پیشنهاد می‌کند که تعدادی درخت پوشا را به‌عنوان جدول مسیریابی ذخیره می‌کند و از روش مسیریابی Nix\_Vector برای محاسبه حالات مسیریابی که در محدوده هیچ درخت پوشایی نیستند، استفاده می‌کند. روش مسیریابی پیشنهادی، مزایای مسیریابی ایستا و برحسب تقاضا را با هم ترکیب کرده و توازن بهتری روی زمان و فضای مصرفی نسبت به سایر روش‌های مسیریابی دارد. براساس نتایج آزمایش شبیه‌سازی در NS-2، روش پیشنهادی نسبت به روش مسیریابی Nix\_Vector، فقط یک درصد به حافظه بیشتری احتیاج دارد و زمان شبیه‌سازی آن حدود ۸۵ درصد از زمان شبیه‌سازی Nix\_Vector کمتر است. در آینده می‌توان مسیریابی پیشنهادی را با الگوریتم‌های بهتر یا انتخاب ریشه‌ها و درخت بهبود داد و در کنار آن، شبکه‌های بزرگ و توزیع‌شده و موازی نظیر PDNS را شبیه‌سازی کرد.

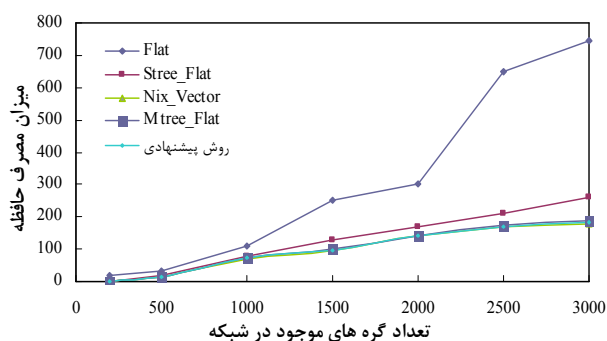
## منابع

- [۱] خراطی، احسان، بررسی و بهینه‌سازی پروتکل‌های مسیریابی در شبکه‌های بی‌سیم AD HOC، پایان‌نامه برای درجه کارشناسی ارشد، دانشگاه آزاد اسلامی واحد اراک، ۹۸-۱۱۹، سال ۸۳
- [۲] خراطی، احسان و موقر، علی، "یک ساختار مسیریابی شی‌گرا در محیط J-Sim، برای شبکه‌های حسگر بی‌سیم"، سیزدهمین کنفرانس سالانه انجمن کامپیوتر ایران، دانشگاه شریف، کیش، ۱۳۸۶
- [۳] خراطی، احسان و موقر، علی، "بهینه‌سازی پروتکل مسیریابی S-MAC برای کاهش مصرف انرژی"، سیزدهمین کنفرانس سالانه انجمن کامپیوتر ایران، دانشگاه شریف، کیش، ۱۳۸۶
- [۴] خراطی، احسان، "ایجاد یک اشکال‌زدا در شبیه‌ساز شبکه‌های حسگر DiSenS"، اولین همایش سالانه مهندسی کامپیوتر، برق و فناوری اطلاعات، دانشگاه آزاد اسلامی واحد همدان، همدان، ۱۳۸۶
- [۵] خراطی، احسان، "بهینه‌سازی پروتکل مسیریابی AODV از طریق تبدیل مدل بسته‌ای به مدل جریان"، کنفرانس مهندسی برق ایران، دانشگاه آزاد اسلامی واحد تبریز، تبریز، ۱۳۸۸

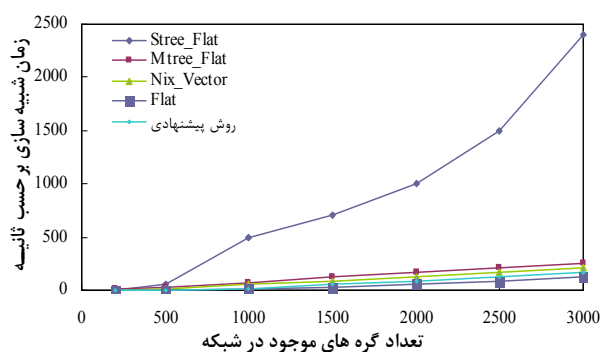
[6] P. Huang, D. Estrin, and J. Heidemann, "Enabling Large scale simulations: selective abstraction approach to the study of multicast protocols," in Proceedings of the International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, Montreal, Canada, July 1998, pp. 241-248.

[7] P. Huang and J. Heidemann, "Minimizing routing state for light-weight network simulation," in Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, August 2001.

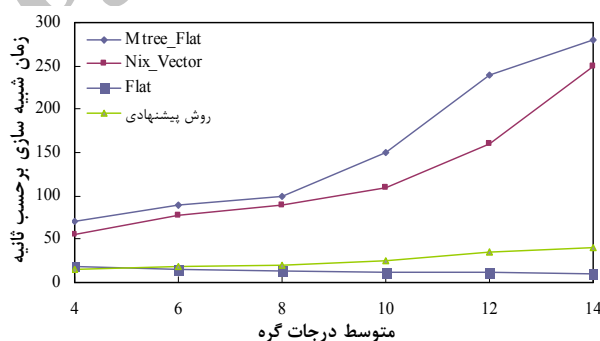
[8] A. Hiromori, H. Yamaguchi, K. Yasumoto, T. Higashino and K. Taniguchi, "Reducing the Size of Routing Tables for Large-scale Network Simulation," in Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, October 2003.



شکل ۶: میزان مصرف حافظه با شیوه‌های مختلف مسیریابی با تعداد گره‌های متفاوت



شکل ۷: زمان شبیه‌سازی شیوه‌های مختلف مسیریابی با تعداد گره‌های متفاوت



شکل ۸: زمان شبیه‌سازی شیوه‌های مختلف مسیریابی با متوسط درجات مختلف گره

- [9] J. Chen, D. Gupta, K.V. Vishwanath, A. C. Snoeren and A. Vahdat, "Routing in an Internet-Scale Network Emulator," in Proceedings of the IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, October 2004.
- [10] G. F. Riley, M. H. Ammar, and E.W. Zegura, "Efficient routing with Nix-Vectors," in Proceedings of IEEE Workshop on High Performance Switching and Routing HPSR 2001, 2001.
- [11] G. F. Riley, R. Fujimoto, and M. H. Ammar, "Stateless routing in network simulations," in Proceedings of the 8th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, May 2000.
- [12] D. Magoni and J. Pansiot, "Internet Topology Modeler Based on Map Sampling," in Proceedings of the 7th IEEE Symposium on Computers and Communications, July 2002, pp. 1021-1027.

Archive of SID

# SID



سرویس های ویژه



سرویس ترجمه تخصصی



کارگاه های آموزشی



بلاگ مرکز اطلاعات علمی



سامانه ویراستاری STES



فیلم های آموزشی

## کارگاه های آموزشی مرکز اطلاعات علمی جهاد دانشگاهی

نوبت آشنایی  
بررسی مقاله ای متون (مقدماتی)

کارگاه آنلاین  
بررسی مقابله ای متون (مقدماتی)

نوبت آشنایی  
پروپوزال نویسی و پایان نامه نویسی

کارگاه آنلاین  
پروپوزال نویسی و پایان نامه نویسی

نوبت آشنایی  
آشنایی با پایگاه های اطلاعات علمی بین المللی و ترند های جستجو

کارگاه آنلاین آشنایی با پایگاه های اطلاعات علمی بین المللی و ترند های جستجو