



مسیریابی توزیع شده با استفاده از کلونی مورچه‌ها در شبکه‌های رایانه‌ای

سعید صفاری امان^۱، محمود نقیب زاده^۲، محمدرضا اکبرزاده توتونچی^۳

^۱ دانشجوی ارشد هوش مصنوعی، شرکت مخابرات خراسان رضوی، sa_saf_amn@yahoo.com

^۲ استاد گروه کامپیوتر دانشگاه فردوسی مشهد، naghbzadeh@um.ac.ir

^۳ دانشیار گروه برق دانشگاه فردوسی مشهد، akbarzadeh@ieee.org

چکیده

شکلهای هوش جمعی بعنوان اساس سری جدیدی از الگوریتم‌هاست، که از رفتار اجتماعی حشراتی که بصورت جمعی زندگی می‌کنند، الهام گرفته‌اند. در این مقاله، شکل جدیدی از الگوریتم AntNet برای مسیریابی در شبکه‌های کامپیوتری معرفی شده است. این الگوریتم با استفاده از کلاستر بندی شبکه و بکارگیری مسیریابی به دو صورت محلی و سراسری، علاوه بر پایین آوردن تأخیر، مشکل عدم مقیاس پذیری الگوریتم AntNet را بصورت توزیع شده حل کرده است. شبیه سازیهای انجام شده نشان می‌دهد، این الگوریتم در بیشتر مواقع از سایر الگوریتمهای مسیریابی نظیر DV (Distance-Vector) و LS (Link-State) نیز بهتر عمل می‌کند.

کلمات کلیدی

AntNet، ANTRAL، مسیریابی داخل کلاستر، مسیریابی بین کلاستری.

۱- مقدمه

مورچه به هر مقصد بفرستد، نه تنها سربار، برای یک شبکه بزرگ زیاد می‌شود، بلکه برای مقاصدی که فواصل زیاد دارند، تعداد زیادی از مورچه‌ها نابود می‌شوند. علاوه بر زمانهای سفر طولانی، باعث می‌شود تا مورچه‌های حامل اطلاعات شبکه، اطلاعاتشان کهنه شود.

در این تحقیق برای برطرف سازی مشکلات فوق، شبکه را کلاستر بندی کردیم، و با بکارگیری عاملهای هوشمند بصورت محلی و سراسری عملیات مسیریابی را بهبود دادیم. نتایج شبیه سازیها نشان می‌دهند، این الگوریتم در بیشتر مواقع از سایر الگوریتمهای مسیریابی نظیر DV و LS نیز بهتر و سریعتر عمل می‌کند.

باقیمانده این مقاله بصورت ذیل خواهد بود؛ در بخش ۲ مسیریابی و هوش جمعی، شامل مروری بر فن آوری مسیریابی در ANTRAL ها بیان شده. در بخش ۳ به توضیح روش پیشنهادی می‌پردازیم. در بخش ۴ مدل شبکه و در بخش ۵ نتایج شبیه‌سازیها را نشان داده‌ایم و بالاخره در بخش ۶ نتیجه گیری بیان شده است.

۲- فن آوری مسیریابی در ANTRALها

هوش جمعی اغلب برای بهینه‌سازی در سیستمهای توزیع شده بکار می‌رود [3]. در این زمینه بیشترین کار روی مورچه‌ها انجام شده است. که مهمترین آنها عبارتند از: AntNet [4,5]، کنترل Ant Based

تنوع زیادی در پروتکلها و الگوریتمهای مسیریابی برای ارتباطات شبکه‌ای وجود دارد. در مسیریابی سنتی جداول مسیریابی به واسطه تبادل اطلاعات مسیریابی بین مسیریابها به روز می‌شوند. بعنوان مثال در DV بر اساس انتقال جداول مسیریابی، و در LS اطلاعات وضعیت لینک بصورت سیل گونه، در کل شبکه ارسال می‌شود.

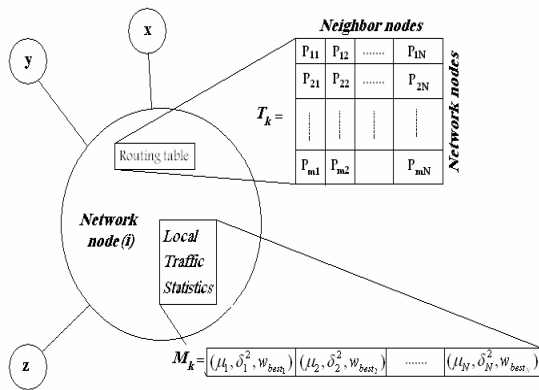
اخیراً، با الهام از مسیریابی مورچه‌ها، الگوریتمهایی ارائه شده، که از عاملهای موبایل برای مسیریابی شبکه‌های دینا استفاده می‌شود. در این مقاله، از آنها بنام ANTRAL (Ant Routing Algorithm) یاد می‌کنیم. ANTRAL یک الگوریتم مسیریابی hop-by-hop است که بر اساس خاصیت Stigmergy در کلونی مورچه‌ها کار می‌کند. Stigmergy شکلی از ارتباط غیر مستقیم است که بوسیله تأثیر روی محیط انجام می‌شود. نشان داده شده که مورچه‌ها در زندگی طبیعی خود قادرند، تا کوتاهترین مسیر را از روی آثار به جای مانده از یک ماده شیمیایی به نام فرمون که از سایر مورچه‌ها ترشح می‌شود، پیدا کنند [1] و [2].

AntNet بعنوان یکی از ANTRAL های رایج در مسیریابی شبکه، دارای مشکل مقیاس پذیری است. چون هر گره لازمست مورچه‌هایی به تمام گره‌های شبکه بفرستد. اگر هر گره فقط یک

شامل یک پنجره دید متحرک W_d ، با اندازه یک رشته W_{max} شامل آخرین زمانهای سفر مورچه پیشرو که از گره k به مقصد d حرکت می کند، را آشکار می کند. پنجره دید متحرک W_d به منظور محاسبه بهترین زمان سفر t_{best_d} بکار می رود. بهترین زمان سفر بوسیله یک مورچه پیشرو که از گره k به مقصد d می رود، از بین W_{max} های قبلی مورچه های پیشروی که از گره k به مقصد d می روند، محاسبه می شود. متوسط μ_d و واریانس σ_d^2 متوسط زمانهای سفری را که بوسیله مورچه های پیشرو در سفر از گره k به مقصد d تجربه می شود، را نشان می دهند.

$$\mu_d \leftarrow \mu_d + \eta(t_{k \rightarrow d} - \mu_d) \quad (1)$$

$$\sigma_d^2 \leftarrow \sigma_d^2 + \eta((t_{k \rightarrow d} - \mu_d)^2 - \sigma_d^2) \quad (2)$$



x, y, z : ()

(M_k) (T_k) : N

در (۱) و (۲) زمان سفر مورچه های پیشروی، را نشان می دهد که به تازگی از گره k به مقصد d حرکت کرده است. و $\eta \in (0, 1]$ فاکتوری است که وزنه های نمونه های اخیر که در میانگین μ_d و واریانس σ_d^2 تاثیر خواهند داشت، را نشان می دهد. Di Caro و Dorigo در [4] ارتباط بین η با سایز ماکزیموم اندازه پنجره دید W_{max} را بصورت زیر بیان کردند.

$$W_{max} = \frac{5c}{\eta} \text{ where } c < 1 \quad (3)$$

η نمونه های موثر در Di Caro و Dorigo نشان دادند، که η میانگین است، (همچنانکه در (۳) بکار رفته است) و متوسط و بهترین زمان سفر روی پنجره دید متحرک، یکسان محاسبه می شود. ما اندازه ماکزیموم پنجره دید W_{max} را به اندازه کافی بزرگ انتخاب می کنیم

$$(W_{max} > \frac{5}{\eta})$$

[2]، همچنین خلاصه ای از الگوریتم های swarm-based در [6] ارائه شده است.

دو جزء اساسی در اجرای یک ANTRAL وجود دارد. ابتدا، چگونگی جستجوی عامل های موبایل برای پیدا کردن کوتاه ترین مسیر، که از آن به Exploration یاد می شود. دوم اینکه عامل های موبایل لازمست جداول مسیریابی را بر اساس مسیری که جستجو کرده اند، به روز کنند. «جستجو» و «تغییر جداول مسیریابی» در ANTRAL ها به هم گره خورده اند، چرا که عامل های موبایل از همان مقادیری برای جستجو استفاده می کنند، که خود، آنرا به روز کرده اند. بنابراین بهبود در چگونگی جستجوی عاملها و به روز رسانی جداول مسیریابی می تواند در بالا بردن سرعت همگرایی الگوریتم موثر باشد.

در تمام الگوریتمهایی که در بالا ذکر شد [1-5] هر گره یک جدول از احتمال hop بعدی به سمت هر مقصد را داراست، که نمونه ای از آن در جدول (۱) آمده است.

جدول (۱): جدول مسیریابی در ANTRAL

گره جاری	Hop بعدی		
	0	3	3
گره مقصد	0	0.9	0.1
	2	0.15	0.85
	3	0.1	0.9

هر ردیف در این جدول متناظر با یک مقصد و هر ستون متناظر با یک همسایه است. اعداد هر ردیف در جدول، احتمال انتخاب هر همسایه را بعنوان hop بعدی برای رفتن به گره مقصد مشخص می کنند. در الگوریتم های swarm-based این احتمالات، در واقع راهنمای مورچه هایی هستند که اجازه پیدا می کنند، در شبکه به جستجو بپردازند، و امکان پیدا کردن مسیرهای جدید و بهتر را داشته باشند. با کشف هر مسیر جدید و بهینه، مقادیر احتمال جدول مسیریابی به روز می شود. لذا بسته های داده ای که از آن مسیر عبور می کنند، با انتخاب بیشترین احتمال به hop بعدی هدایت می شوند.

۲-۱- ساختار داده ها در هر گره شبکه

عامل های موبایل بشکل غیرمستقیم، از طریق اطلاعاتی که همزمان در دو ساختار داده ای که در گره k ام هر شبکه وجود دارند و خواننده و نوشته می شوند، با هم ارتباط دارند.

در جدول مسیریابی T_k احتمال routing policy در گره k به این صورت مشخص می شود: برای هر مقصد ممکن d و هر گره همسایه n مقدار p_{nd} که احتمال انتخاب n بعنوان گره بعدی است، ذخیره می شود.

$$\sum_{n \in N_k} p_{nd} = 1 \quad N_k = \{neighbors(k)\} \quad d \in [1, N]$$

آنچنانکه از یک گره محلی k دیده می شود، جدول $M_k(\mu_d, \sigma_d^2, W_d)$ شامل احتمالات توپولوژی شبکه و توزیع ترافیک روی شبکه است. برای هر مقصد d در شبکه، جدول M_k

۲-۲- توصیف الگوریتم AntNet

الگوریتم AntNet می تواند بصورت زیر توصیف شود :

۱- در بازه های مشخص زمانی، از هر گره s شبکه یک مورچه پیشرو $F_{s \rightarrow d}$ بصورت تصادفی به گره مقصد d فرستاده می شود. مقاصد طوری انتخاب می شوند، تا با الگوی ترافیکی شبکه همخوانی داشته باشند. به بیان دیگر اگر f_{sd} یک معیار (در بیتها یا در تعداد بسته ها) از جریان دیتا $d \rightarrow s$ باشد، و احتمال y_d ایجاد یک مورچه پیشرو در گره s برای رفتن به گره مقصد d باشد، داریم؛

$$y_d = \frac{f_{sd}}{\sum_{d'=1}^N f_{sd'}} \quad (۴)$$

۲- در حالیکه مورچه های پیشرو به سمت گره های مقصدشان در حال حرکت هستند، مسیرها و شرایط ترافیکی را ذخیره می کنند. شناسه های هر گره ویزیت شده k و زمان رسیدن مورچه پیشرو از هنگام ایجاد تا رسیدن به k امین گره در پشته حافظه $S_{s \rightarrow d}$ موجود در مورچه پیشرو ذخیره (Push) می شود.

۳- در هر گره k ، هر مورچه پیشرو گره بعدی را بصورت ذیل انتخاب می کند؛

• اگر همه گره های همسایه ویزیت نشده باشند، پس همسایه بعدی از بین گره هایی که ویزیت نشده اند انتخاب می شود؛

$$P'_{nd} = \frac{P_{nd} + \alpha l_n}{1 + \alpha(|N_k| - 1)} \quad (۵)$$

در (۵)، N_k مجموعه ای از همسایه های گره جاری k را نشان می دهد. و l_n که به مقادیر $[0,1]$ نرمالایز می شود، از رابطه (۶) بدست می آید. که $1 - l_n$ متناسب است با طول صف q_n مربوط به لینک بین گره k و همسایه اش n ؛

$$l_n = 1 - \frac{q_n}{\sum_{n'=1}^{|N_k|} q_{n'}} \quad (۶)$$

مقدار α در (۵) تقاضای وضعیت صف گره را با مراجعه به مقادیر احتمال ذخیره شده در جدول احتمال می سنجد.

• اگر همه همسایه ها قبلاً ویزیت شده باشند، گره بعدی از بین همه گره های همسایه، با احتمال برابر انتخاب می شود. بنابراین، با توجه به اینکه همه همسایه ها قبلاً ویزیت شده اند، مورچه پیشرو مجبور است به گره ای بازگردد، که قبلاً ویزیت شده. بنابراین، قطع نظر از اینکه کدام همسایه بعنوان گره بعدی انتخاب شود، مورچه پیشرو در یک حلقه قرار می گیرد.

• با یک احتمال کوچک ϵ ، گره بعدی ممکن است بطور یکسان از بین همه گره های همسایه انتخاب شود. پارامتر ϵ عمداً در ANTRAL وارد می شود، تا بر این مسئله که یکی از مسیرها

تقریباً همیشه انتخاب شود، و دیگران تضعیف شوند، غلبه کند. در چنین موقعیتی مورچه های پیشرو، همیشه همان لینک را انتخاب می کنند، و بنابراین جستجوی شبکه برای دیگر مسیرها متوقف می شود. پارامتر ϵ در شبکه باعث می شود، عملیات جستجو بطور دائم انجام گیرد، گرچه که شاید پاسخ بی اثری را نیز تولید کند. عبارت دیگر پارامتر ϵ بعنوان یک $\epsilon - greedy$ policy در مبحث یادگیری تقویتی نقش بازی می کند [7] و حالت تصادفی را در الگوریتم ایجاد می کند. در اجرای AntNet [8] مقدار پارامتر ϵ صفر انتخاب شد.

۴- اگر یک حلقه کشف شود، مورچه مجبور می شود، به گره قبلی بازگردد، و گره های موجود در حلقه، لازمست از پشته مورچه حذف شوند. اگر حلقه بیشتر از زمان زندگی مورچه پیشرو طول بکشد، مورچه نابود می شود (شکل (۲)).

۵- هنگامیکه مورچه پیشرو $F_{s \rightarrow d}$ به گره مقصد d رسید، یک مورچه عقبگرد $B_{d \rightarrow s}$ تولید می کند. مورچه پیشرو همه محتویات حافظه پشته $S_{s \rightarrow d}$ را به مورچه عقبگرد منتقل می کند و از بین می رود.

۶- مورچه عقبگرد دقیقاً همان مسیر مورچه پیشرو را در جهت عکس باز می گردد. در هر گره k پشته اش را واکنشی می کند تا به گره بعدی برسد.

۷- هنگامیکه مورچه های عقبگرد از همسایه h به گره k می رسند، دو ساختار داده ای مهم را برای همه مداخل متناظر با مقصد d به روز می کنند، یکی مدل محلی ترافیک M_k و دیگری جدول مسیریابی T_k . به روز رسانی جداول مسیریابی در هر گره در طول مسیری که مورچه عقبگرد از آن می گذرد، بعنوان یک روش به روز رسانی sub-path شناخته می شود. (اگر حلقه های موجود در مسیر مورچه های پیشرو حذف نشوند، اینکار در به روز رسانی sub-path به صورت آماری انجام می شود). بنابراین به روز رسانی sub-path فقط برای حذف حلقه ها از مسیر مورچه های پیشرو بکار گرفته می شود.

• میانگین μ_d و واریانس σ_d^2 مداخل در مدل محلی ترافیک M_k با استفاده از (۱) و (۲) تغییر می کند. بهترین مقدار t_{best_d} (برای زمان سفر مورچه های پیشرو از گره k به مقصد d) که در پنجره دید متحرک W_d ذخیره شده، بوسیله مورچه های عقبگرد به روز می شود. اگر زمان سفر جدید مورچه پیشرو $t_{k \rightarrow d}$ کمتر از t_{best_d} باشد، بنابراین t_{best_d} بوسیله $t_{k \rightarrow d}$ جایگزین می شود.

• جدول مسیریابی T_k بوسیله افزایش احتمال $p_{hd'}$ (احتمال انتخاب همسایه h وقتی مقصد d' باشد) تغییر می کند. احتمال $p_{hd'}$ بوسیله افزایش مقدار تقویتی r بصورت ذیل، افزایش می یابد:

$$p_{hd'} \leftarrow p_{hd'} + r(1 - p_{hd'}) \quad (۷)$$

t_{best_d} اشاره می‌کند. مقادیر C_1 و C_2 وابستگی مهمی به هر قسمت دارند. بخش اول از بخش دوم مهم‌تر است. بنابراین، مقدار C_1 لازمست بزرگتر از مقدار C_2 انتخاب شود. و بالاخره مقدار r در فرمول (۹) توسط میانگین تابع squash $s(x)$ محاسبه و بصورت زیر تعریف می‌شود:

$$s(x) = \frac{1}{1 + \exp\left(\frac{a}{|x|}\right)}, \text{ where } x \in (0, 1], a \in R^+ \quad (11)$$

$$r \leftarrow \frac{s(r)}{s(1)} \quad (12)$$

تابع squash $s(x)$ در الگوریتم AntNet تعریف می‌شود، چرا که مقادیر کوچک r تاثیر کمی در به روز رسانی جداول مسیریابی دارند. لذا لازمست تا تابع squash $s(x)$ مقادیر پایین r را بیشتر کاهش دهد، تا کمک کمتری در به روز رسانی جداول مسیریابی بکند (مقادیر پایین r به زیر مسیرهای پهنه اشاره می‌کنند).

این نکته را نیز نباید فراموش کرد که بسته‌های داده یک زمان ثابت زندگی (TTL) دارند، که اگر در مدت TTL به مقصد نرسند، از بین می‌روند.

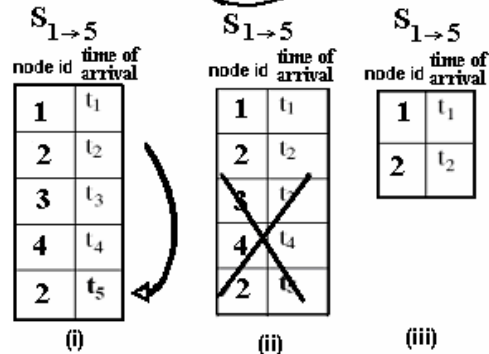
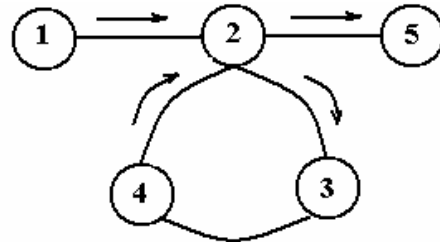
۳- الگوریتم پیشنهادی

اگر کمی دقیق‌تر به مسئله تبادل اطلاعات در شبکه‌ها بپردازیم، متوجه می‌شویم که معمولاً شبکه‌ها برای دسترسی و تبادل داده با یکسری اهداف خاص بوجود آمده‌اند. طبیعی است ارایه یک مدل مناسب در مسیریابی می‌تواند بسیار کارساز و مفید واقع شود.

بعنوان مثال یک شبکه داخل سازمان برای نقل و انتقال داده‌ها در آن، و انجام امور اداری طراحی شده است. بنابراین ارتباطات داخل شبکه مذکور زیاد است. از طرفی نیاز ارتباط با دنیای خارج را نیز نباید فراموش کرد. در چنین مواردی با مشخص کردن مجموعه‌هایی بصورت محلی، عملیات مسیریابی داخل مجموعه احتیاج به اطلاعات محلی دارد و نیاز به تبادل حجم زیاد اطلاعات سربرار در کل شبکه از بین می‌رود.

در حالت کلی وقتی در یک شبکه بزرگ، تغییری ایجاد می‌شود، لازمست به اطلاع همه گره‌های شبکه برسد. در حالیکه اهمیت این تغییرات، برای گره‌هایی که با آن سروکار بیشتری دارند (ما آنها را در یک مجموعه قرار می‌دهیم) بیشتر است. بدین ترتیب تقسیم‌بندی شبکه به مجموعه‌های کوچکتر باعث می‌شود:

گره‌های هر مجموعه، فقط اطلاعات مسیریابی آن مجموعه را نگهداری کنند (مسیریابی توزیع شده می‌شود)، و از دانستن جزئیات تغییرات در کل شبکه خلاص شوند. حتی در صورت ایجاد تغییرات در



$$F_{1 \rightarrow 5} \quad (i) : () \quad (ii)$$

$$F_{1 \rightarrow 5} \quad (t_5 - t_2) \quad (iii)$$

$$F_{1 \rightarrow 5}$$

مقدار احتمال $p_{nd'}$ مربوط به همسایه‌های دیگر گره n به مقصد d' با مقدار تقویتی منفی بصورت ذیل کاهش می‌یابد:

$$p_{nd'} \leftarrow p_{nd'} - r p_{nd'}, \forall n \neq h, n \in N_k \quad (8)$$

بنابراین، در AntNet، هر مسیر بوسیله مورچه‌های پیشرویی پیدا می‌شود، که مقدار تقویتی مثبتی را دریافت می‌کنند.

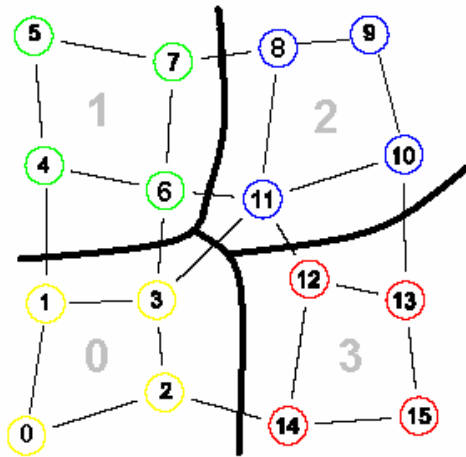
مقدار تقویتی r که در (γ) و (λ) آمده در بازه $(0, 1]$ قرار دارد و بصورت ذیل محاسبه می‌شود:

$$r = C_1 \frac{t_{best_d}}{t_{k \rightarrow d}} + C_2 \frac{t_{sup} - t_{best_d}}{(t_{sup} - t_{best_d}) + (t_{k \rightarrow d} - t_{best_d})} \quad (9)$$

در (9) $t_{k \rightarrow d}$ زمان سفر جدید مورچه پیشرو از گره k به مقصد d است و t_{best_d} بهترین زمان سفری است که بوسیله مورچه‌های پیشرو در هنگام حرکت به سمت مقصد d روی پنجره دید W_d تجربه شده است. مقدار t_{sup} نیز بصورت زیر محاسبه می‌شود:

$$t_{sup} = \mu_d + z \frac{\sigma_d}{\sqrt{W_d}}, \quad z = const \quad (10)$$

بخش اول فرمول (9) نسبت بین زمان سفر فعلی و بهترین زمان سفر طی شده روی پنجره دید متحرک را ارزیابی می‌کند. بخش دوم نیز یک فاکتور اصلاحی است و به فاصله بین $t_{k \rightarrow d}$ و



(:)

توجه شود که جدول مسیریابی کلاستر به تعداد کلاسترهای شبکه، ردیف (مقصد) و به تعداد همسایه‌های گره جاری ستون خواهد داشت.

جدول (۲): جدول مسیریابی بین کلاستری

گره جاری = 1	Hop بعدی			
	0	3	4	
کلاستر	1	0.05	0.15	0.8
مقصد	2	0.05	0.9	0.05
	3	0.2	0.75	0.05

جدول (۳): جدول مسیریابی محلی (داخل کلاستر)

گره جاری = 1	Hop بعدی		
	0	3	
گره مقصد	0	0.9	0.1
	2	0.15	0.85
	3	0.1	0.9

در جدول کلاستر همه همسایه‌های گره‌ها، شامل لیستی از hop های بعدی هستند. این در حالی است که در جدول مسیریابی محلی فقط گره‌هایی شامل hop بعدی می‌شوند که متعلق به همان کلاستر باشند. بعنوان مثال در جدول کلاستر فوق چون در کلاستر صفر هستیم، به ازای تمام کلاسترهای 1,2,3 گره بعدی مشخص می‌شود. احتمالات این جداول همانند روش AntNet با افزودن تغییرات زیر تغییر می‌کند:

- 1- در دوره‌های مشخص و منظم هر گره، یک مورچه محلی به هر کدام از گره‌های درون کلاستر و یک مورچه کلاستر برای هر کلاستر در خارج می‌فرستد.
- 2- مورچه‌های محلی گره بعدی را از روی جدول مسیریابی محلی پیدا می‌کنند. در حالیکه مورچه‌های کلاستری گره بعدی را بر اساس جدول مسیریابی کلاستری می‌یابند.
- 3- هر دو نوع مورچه اطلاعات مربوط به گره‌هایی که به آن رسیده‌اند و زمان ورود را در پشته خود ثبت می‌کنند.

شبکه، بدلیل همین تقسیم‌بندی، این تغییرات مربوط به بخش کوچکی از شبکه شده و دیگران کمتر از آن متأثر می‌شوند. بنابراین از یکسو می‌توان بدون توجه به تغییرات شبکه در خارج مجموعه عملیات مسیریابی را انجام داد. از سوی دیگر به علت تبادل حجم کمتری از داده‌های سربر با خارج مجموعه‌ها، و انجام مسیریابی در داخل آنها سرعت مسیریابی بیشتر می‌شود.

در این تحقیق با بکارگیری عملهای هوشمند بصورت محلی و سراسری سعی کرده‌ایم، عملیات مسیریابی را بهبود بخشیم.

این الگوریتم شامل سه مرحله می‌باشد. فاز اول کلاستربندی گره‌های شبکه به کلونی‌های کوچکتر. این فاز مطابق با نیاز طراحی می‌شود. مثلاً قرار دادن گره‌هایی که به لحاظ هم زبان بودن مرادوات بیشتری با هم دارند (ارتباطات داخل کشور اسپانیا با سایت‌های اسپانیولی بیشتر است). فاز دوم یافتن مسیرهای شبکه توسط عاملها (مورچه‌ها) و فاز سوم فرستادن ترافیک شبکه به مقصد، از طریق مسیرهای یافت شده توسط مورچه‌هاست. فاز اول معمولاً خیلی تکرار نمی‌شود. چرا که فقط در ابتدای هر مرحله از اجرای الگوریتم و موقعی که توپولوژی شبکه تغییر کرد، عملیات کلاستر بندی مجدد انجام می‌گیرد. فازهای دوم و سوم بعنوان عملیات معمول شبکه بطور دائم انجام می‌شوند.

۳-۱- کلاستربندی شبکه به کلونی‌های کوچکتر

AntNet دارای مشکل مقیاس پذیری است. چون هر گره لازمست مورچه‌هایی به تمام گره‌های شبکه بفرستد. اگر هر گره فقط یک مورچه به هر مقصد بفرستد و تعداد گره‌های شبکه N باشد، تعداد کل مورچه‌های فرستاده شده $N(N-1)$ است. برای یک شبکه بزرگ ترافیک ایجاد شده، زیاد است. همچنین برای مقاصدی که فواصل زیاد دارند، تعداد زیادی از مورچه‌ها نابود می‌شوند. بعلاوه زمانهای سفر طولانی باعث می‌شود، تا مورچه‌های حامل اطلاعات شبکه، اطلاعاتشان کهنه شود. یک روش برای جلوگیری از این مشکلات کلاستربندی شبکه به شبکه‌های کوچکتر است.

۳-۲- یافتن مسیرها

بعد از شکل‌گیری کلاسترها دو نوع عامل (مورچه) به شبکه معرفی می‌گردند:

نوع اول، مورچه‌های کلاستر و نوع دوم مورچه‌های محلی نامیده می‌شوند. وظیفه مورچه‌های کلاستر یافتن مسیرها، بین یک کلاستر و کلاسترهای دیگر است. در حالیکه مورچه‌های محلی به درون یک کلاستر محدود می‌شوند، و مسؤل یافتن مسیرها در یک کلاستر می‌باشند. بنابراین در هر گره دو جدول مسیریابی نیاز داریم. نمونه‌ای از جدول کلاستر و جدول محلی، به ترتیب در جداول (۲) و (۳) نشان داده شده است (فرض بر اینست که، در شبکه موجود در شکل (۳) و در گره‌ای مثل ۱ قرار دارید).

را up کردیم، و مشاهده کردیم، الگوریتم دوباره مسیر کوتاه ۹ به ۸ را برای ارسال داده، انتخاب می‌کند.

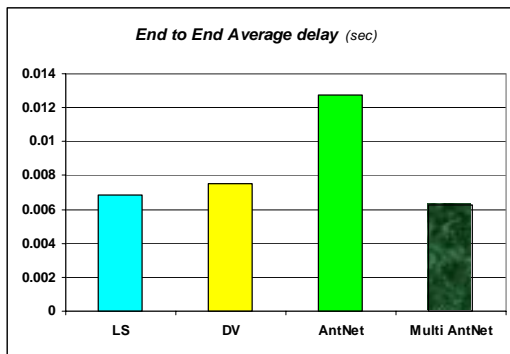
برای بررسی چگونگی مسیریابی بین کلاستری، با راه اندازی دو جریان CBR با بسته‌هایی به اندازه ۲۵۶ بایت، اولی از گره ۱۴ در کلاستر ۳ و دومی از گره ۵ در کلاستر ۱ به مقصد گره ۱۱ در کلاستر ۲ عملکرد الگوریتم در مسیریابی بین کلاسترها تست شد.

۵- نتایج آزمایشگاهی

برای بررسی چگونگی کار الگوریتم پیشنهادی و مقایسه آن با سایر الگوریتمها، مدل شبکه علاوه بر AntNet سنتی [10] با الگوریتمهای LS و DV [9] به مدت ۱۰ ثانیه تست و نتایج در فایل‌های Trace ذخیره شد.

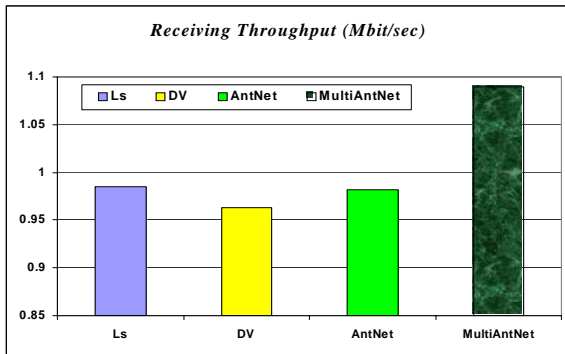
گفتنی است شبیه‌سازی انجام شده در شبیه‌ساز NS2 [11] تحت سیستم عامل Linux انجام شد.

دقت شود که ترافیک CBR شبکه در هر بار آزمایش، از ثانیه ۳ شروع و تا پایان ثانیه ۹ ادامه پیدا کرد. همانطور که در شکل (۴) مشاهده می‌کنید، میزان تاخیر الگوریتم پیشنهادی (آنرا MultiAntNet نامیده‌ایم) نسبت به سایر الگوریتمها کمتر است.



شکل (۴): مقایسه میزان میانگین تاخیر End-to-End اجرای الگوریتمهای مختلف روی شبکه هدف

مطابق شکل (۵) بالا بودن میزان Throughput دریافتی الگوریتم پیشنهادی در مقایسه با سایرین، کارایی الگوریتم را نشان می‌دهد.



شکل (۵): مقایسه میزان Receiving Throughput

۴- مورچه‌ها مراقب ایجاد حلقه هستند. اگر به گره ملاقات شده‌ای برسند، تمام حافظه مربوط به آن حلقه پاک شده و گره بعدی به صورت تصادفی انتخاب می‌شود. اگر مورچه‌ها تعداد مشخصی جهش در راه رسیدن به مقصد داشتند (برای مورچه‌های محلی برابر تعداد گره‌های داخل کلاستر و برای مورچه‌های بین کلاستری برابر تعداد کل گره‌های شبکه) متوقف می‌شوند.

۵- هرگاه مورچه‌ای به مقصد رسید، مسیر طی شده را به صورت بر عکس تا مبدأ طی می‌کند، و به هر گره که می‌رسد، جدول مسیریابی آن گره را تغییر می‌دهد.

دقت شود که در بخش (۲) چگونگی عملکرد الگوریتم AntNet بطور کامل و دقیق بحث شد، تا در این قسمت مجبور به تکرار بحثها و فرمولهای آن برای عملکرد الگوریتم در حوزه داخل کلاستر و بین کلاستری نباشیم.

۳-۳- ارسال ترافیک

جداول مسیریابی از حرکت تصادفی مورچه‌ها بدست می‌آیند. در مدل‌های قبلی نظیر AntNet بعد از اینکه مورچه‌ها مسیری را پیدا می‌کنند، جداول مسیریابی مقدار دهی می‌شود، و بسته‌های داده، گره‌ای که دارای بیشترین احتمال است، را برای رسیدن به مقصد خود انتخاب می‌کنند.

در شبیه سازی انجام شده، بسته‌های داده برای رسیدن به مقصد از رویه زیر استفاده می‌کنند:

۱- وقتی بسته اطلاعاتی تولید می‌شود، از کلاستر و گره مقصدش مطلع است.

۲- اگر گره فرستنده و گیرنده در یک کلاستر نباشند، بسته داده hop بعدی را از روی بیشترین احتمال، از جدول مسیریابی کلاستری انتخاب می‌کند.

۳- در غیر اینصورت از جدول مسیریابی محلی استفاده خواهد شد.

۴- مدل شبکه

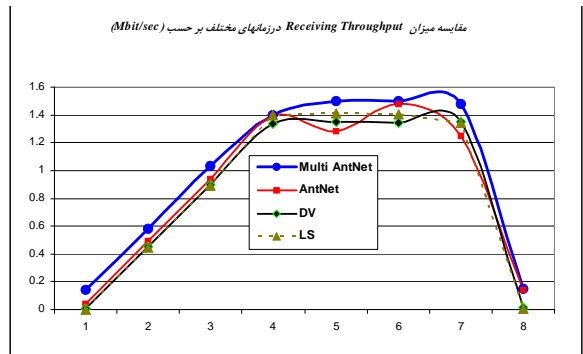
برای نشان دادن مفاهیم گفته شده در بالا، الگوریتممان را روی یک شبکه ۱۶ گره‌ای آزمایش کردیم، که در شکل (۳) نشان داده شده است. شبکه مورد نظر را به ۴ کلاستر تقسیم کردیم. همه لینکها بصورت دوطرفه (duplex) هستند. در این شبکه ۳ منبع با نرخ بیت ثابت (CBR) وجود دارد.

ابتدا دینامیک بودن الگوریتم را با راه اندازی جریان CBR با بسته‌هایی به اندازه ۱۰۰۰ بایت، در کلاستر ۲ از گره ۹ به گره ۱۱ تست کردیم. به اینصورت که از لحظه $t=5.5$ تا $t=6.8$ لینک بین دو گره ۹ و ۸ که حامل ترافیک به مقصد ۱۱ بودند، را down کردیم. الگوریتم مسیر خودش را از طریق لینک ۹ به ۱۰ به مقصد ۱۱ راهیابی کرد، که نشان داد عملکرد الگوریتم دینامیک است. سپس برای تست عملکرد بهینه الگوریتم در پیدا کردن کوتاهترین مسیر، لینک ۹ به ۸

این الگوریتم که MultiAntNet نامگذاری شده است، مطابق آنچه گفته شد، در اکثر موارد، از جمله میزان تأخیر، Throughput و Packet Loss از نسخه اصلی AntNet بهتر عمل می‌کند و می‌تواند کاربردهای فراوانی در شبکه‌های WAN داشته باشد.

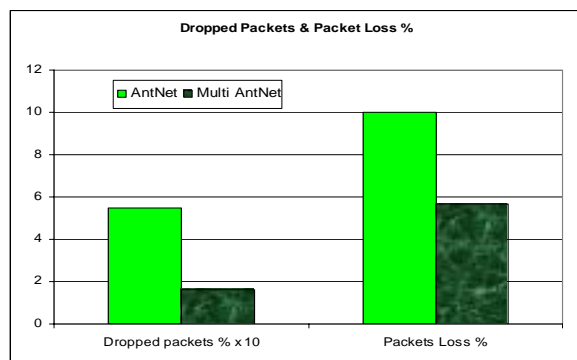
مراجع

- [1] S.S. Dhillon, P. Van Mieghem, "Performance analysis of the AntNet algorithm, *Computer Network*," (2006), doi:10.1016/j.comnet.2006.11.002.
- [2] K.M. Sim, W. H. Sun, "Ant Colony Optimization for Routing and Load-Balancing", IEEE transactions on systems, man, and Cybernetics part A: systems and humans, vol. 33.No.5. september 2003.
- [3] E. Bonabeau, M. Dorigo, and G. Théraulaz, "Swarm intelligence: from natural to artificial systems," Oxford University Press, 1999.
- [4] G. Di Caro and M. Dorigo, "AntNet: distributed stigmergetic control for communications networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317-365, 1998.
- [5] G. Di Caro and M. Dorigo, "AntNet: a mobile agents approach to adaptive routing," Tech. Rep. IRIDIA/97-12, Université Libre de Bruxelles, Belgium.
- [6] I. Kassabalidis, M.A. El-Sharkawi, R.J. Marks II, P. Arabshahi, and A.A. Gray, "Swarm intelligence for routing in communication networks," *IEEE Globecom* 2001, Nov 25-29, 2001, San Antonio, Texas.
- [7] Richard S. Sutton, Andrew G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, Cambridge, MA, 1998.
- [8] G. Di Caro, M. Dorigo, *Two ant colony algorithms for best-effort routing in datagram networks*, in: Proceedings of the 10th International Conference on Parallel and Distributed Computing and Systems, 1998.
- [9] D. Bertsekas and R. Gallager, *Data Networks*, Prentice-Hall, Inc, Upper Saddle River, New Jersey, 1992.
- [10] Azadeh Soltani, M.-R. Akbarzadeh-T and M. Naghibzadeh, "Helping ants for adaptive network routing," *Journal of the Franklin Institute* Volume 343, Issues 4-5, July-August 2006, Pages 389-403.
- [11] Network Simulator, V.2., <http://www.isi.edu/nsnam/ns>.



شکل (۶): مقایسه میزان Throughput دریافتی در زمانهای مختلف

در مقایسه الگوریتم پیشنهادی با نسخه اصلی AntNet پایین بودن میزان Packet Dropped و Packet Loss و تأخیر الگوریتم پیشنهادی، نشان می‌دهد، نسبت به AntNet، از کارایی خوب، و سرعت بالاتری در مسیریابی برخوردار است. (شکل (۷))



شکل (۷): مقایسه میزان Packet Dropped و Packet Loss

جدول (۴): نتایج عددی اجرای الگوریتمها

پارامترهای شبیه‌سازی	LS	DV	AntNet	MultiAntNet
End to End Average delay	0.00688	0.007488	0.012718	0.006283
Receiving bits Throughput	985197.7	962532.5	981846.8	1089556.5

۶- نتیجه گیری

در این مقاله با ارایه یک مدل مناسب در مسیریابی و بکارگیری عملیات هوشمند بصورت محلی و سراسری سعی کردیم، عملیات مسیریابی را بهبود بخشیم.

در شبکه جدید با عملیات کلاستر بندی و استفاده از دو جدول مسیریابی توانستیم، مشکل مقیاس پذیری AntNet را حل کنیم. علاوه بر آن با این روش، بدون توجه به تغییرات شبکه در خارج کلاستر عملیات مسیریابی را انجام دادیم. از سوی دیگر به علت تبادل حجم کمتری از داده‌های سر بار با خارج مجموعه‌ها، و انجام مسیریابی در داخل آنها سرعت مسیریابی نیز بیشتر شد.