

بهبود سرویس جستجوی مبتنی بر جدول درهم‌سازی توزیع‌شده در شبکه‌های هم‌تا به هم‌تای آلوده

گلشن قربانیان^۱، دانشجوی کارشناسی ارشد؛ عبدالرسول قاسمی^۲، استادیار

۱- دانشکده مهندسی کامپیوتر - دانشگاه صنعتی خواجه‌نصیرالدین طوسی - تهران - ایران - golshan.ghorbanian@ee.kntu.ac.ir

۲- دانشکده مهندسی کامپیوتر - دانشگاه صنعتی خواجه‌نصیرالدین طوسی - تهران - ایران - arghasemi@kntu.ac.ir

چکیده: از جدول‌های درهم‌سازی توزیع‌شده، به‌منظور مکان‌یابی مقیاس‌پذیر و توزیع‌شده منابع در شبکه‌های هم‌تا به هم‌تا استفاده می‌شود. هم‌تاهای آلوده می‌توانند با دست‌کاری اطلاعات مسیریابی، بر روی جستجوها تأثیر بگذارند. در نبود یک گره مرکزی ناظر، آلودگی یک مسئله جدی در این شبکه‌ها قلمداد می‌شود. در این مقاله، با استفاده از نظرات هم‌تاها در مورد یکدیگر، یک سیستم شهرت تشکیل شده است. الگوریتم اجماع پیشنهادی به هم‌تاهای سیستم، امتیاز شهرت تخصیص می‌دهد. با استفاده از این الگوریتم، هم‌تاها می‌توانند در مورد امتیاز شهرت هم‌تای مورد نظر به توافق رسیده و برای ارتباط با آن هم‌تا تصمیم‌گیری نمایند. نرخ موفقیت در فرآیند مکان‌یابی راهکار پیشنهادی بیش از ۱۵ درصد بیشتر از الگوریتم ReDS است که به‌عنوان یکی از بهترین سیستم‌های شهرت مبتنی بر جدول‌های درهم‌سازی توزیع‌شده شناخته می‌شود که هدف آن مبارزه با آلودگی در سیستم‌های هم‌تا به هم‌تا است. در ادامه برای بهتر کردن راهکار اجماع پیشنهادی، الگوریتم اجماع وزن‌دار مطرح شده است. به این‌صورت که با توجه به امتیاز شهرت هم‌تا، نظر ارائه‌شده توسط آن هم‌تا وزن‌دار می‌شود. نرخ موفقیت مکان‌یابی در این روش حدوداً ۴۰ درصد نسبت به اجماع ساده بهتر شده در حالی که ۱۴ درصد حافظه بیشتری اشغال می‌شود.

واژه‌های کلیدی: هم‌تا به هم‌تا، آلودگی، هم‌تای آلوده‌کننده، جدول درهم‌سازی توزیع‌شده، امتیاز شهرت، جستجوی افزونه، اجماع.

Improving Distributed Hash Table-based Lookup Services in Polluted Peer-to-Peer Networks

G. Ghorbanian¹, MSc Student; A. Ghasemi², Assistant Professor

1- Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran, Email: golshan.ghorbanian@ee.kntu.ac.ir

2- Faculty of Computer Engineering, K. N. Toosi University of Technology, Tehran, Iran, Email: arghasemi@kntu.ac.ir

Abstract: Distributed Hash Tables (DHTs) are used in the peer-to-peer networks for scalable and distributed localization of resources. However, malicious peers may affect the lookups by manipulating the routing information. Without any central monitoring nodes, pollution will be a real concern in these networks. In this paper, using the peer feedbacks about each other, a reputation system is formed. The proposed consensus algorithm assigns reputation scores to the peers of the system. By using consensus algorithm, the peers can aggregate on the reputation score of the desired peer and make a trust decision about it. The success rate of searches in the proposed consensus algorithm is 15% higher than the ReDS algorithm which is one of the best reputation systems based on DHTs that is designed against pollution in the peer to peer systems. In the following, the weighted consensus algorithm has been proposed. In the weighted consensus algorithm, the reputation scores suggested by other peers are weighted by their own reputation scores. In the weighted consensus algorithm, the success rate of the searches is approximately 40% increased while the memory usage is increased about 14% compared to the consensus algorithm.

Keywords: Peer-to-peer, pollution, malicious peer, dht, reputation score, redundant search, consensus.

تاریخ ارسال مقاله: ۱۳۹۵/۰۵/۳۱

تاریخ اصلاح مقاله: ۱۳۹۵/۰۷/۲۲

تاریخ پذیرش مقاله: ۱۳۹۵/۰۹/۰۸

نام نویسنده مسئول: عبدالرسول قاسمی

نشانی نویسنده مسئول: تهران - دانشگاه صنعتی خواجه‌نصیرالدین طوسی - دانشکده مهندسی کامپیوتر.

۱- مقدمه

محاسبات توزیع شده: در محاسبات توزیع شده، از منابع پردازشی موجود اما بدون استفاده کاربران استفاده می شود. بارزترین نمونه از این سیستم ها SETI@home است که در صدد یافتن نشانه هایی از موجودات هوشمند در کرات دیگر به وسیله سیگنال های رادیویی دریافت شده از تلسکوپ های رادیویی است. این پروژه محاسبات پیچیده و سنگین خود را در میان حدود ۲/۵ میلیون رایانه داوطلب پخش کرده است.

پیام رسانی فوری: برنامه های پیام رسانی فوری، یکی از پرطرفدارترین کاربردهای شبکه های همتا به همتا هستند و تحولی عظیم در ارتباطات شخصی و کاری به وجود آورده اند. در این نرم افزارها بیش از هر جای دیگری خاصیت اضافه شدن و از بین رفتن همتاها قابل مشاهده است [۱].

با وجود مزایای فراوان و محبوبیت سیستم های همتا به همتا، این سیستم ها همواره در معرض آلودگی قرار دارند. برخلاف ساختار سیستم های مشتری/سرویس دهنده، در سیستم های همتا به همتا کنترل کننده مرکزی قابل اعتماد وجود ندارد. به دلیل استقلال و خودمختاری همتاها، امکان آلودگی در این سیستم ها وجود دارد. همتاها می توانند شبکه را به صورت عمدی و غیر عمدی در روند سیستم اختلال ایجاد کنند. آلودگی در سیستم های همتا به همتا انواع مختلفی دارد که به عنوان نمونه می توان به موارد زیر اشاره کرد:

آلودگی محتوایی: در این آلودگی میزان دسترسی پذیری داده، با تغییر دادن محتوای واقعی کاهش پیدا می کند. همتاها ممکن است بدون اطلاع از اعتبار محتوای دریافتی، داده آلوده را در سیستم به اشتراک گذارند که این فرآیند باعث توزیع آلودگی در شبکه خواهد شد. آلودگی به سرعت در شبکه همتا به همتا پخش شده و تأثیر چشمگیری در هدر رفتن منابع و به خصوص پهنای باند و کیفیت سرویس خواهد داشت.

آلودگی در فرایند مکان یابی: نوع دیگر آلودگی در سیستم های همتا به همتا می تواند در ارائه خدماتی به غیر از اشتراک گذاری داده باشد. مسیریابی و مکان یابی بهینه همتاها و داده های مورد جستجو، مهم ترین هدف سیستم های همتا به همتا است. هر همتایی که به صورت عمدی یا غیرعمدی در روند مسیریابی اختلال ایجاد کند، به طور مستقیم بر کارایی سیستم اثر گذاشته و ارائه خدمت را بی کیفیت می کند.

درخواست خدمت رایگان: همتا، بدون مشارکت در ساختار سیستم، تنها به استفاده از خدمات می پردازد. به عنوان مثال در سیستم های اشتراک گذاری فایل، همتای درخواست کننده خدمت رایگان، بدون اینکه در فرآیند به اشتراک گذاری فایل شرکت کند و از پهنای باند خود برای بارگذاری فایل استفاده کند تنها به دریافت منابع و داده از بقیه همتاها می پردازد. عدم همکاری همتا در فرآیند توزیع فایل، باعث کاهش کارایی سیستم خواهد شد.

همتا به همتا گونه ای از معماری شبکه های رایانه ای است که در آن سرویس دهنده مرکزی وجود نداشته، همه اعضا در یک سطح بوده و هر همتا نقش سرویس دهنده و سرویس گیرنده را ایفا می کند. ظهور این فناوری توانسته است با حل بسیاری از معضلات سیستم های سنتی مشتری/سرویس دهنده، محبوبیت فراوانی به خصوص در میان کاربران اینترنتی پیدا کند. مهم ترین علت استفاده از شبکه های همتا به همتا قابلیت توسعه یافتن و مقیاس پذیری آن است. از دیگر مزایای استفاده از شبکه های همتا به همتا می توان به موارد زیر اشاره کرد:

استفاده کارآمد از منابع: در یک معماری مشتری/سرویس دهنده، هر چه سرویس گیرندگان بیشتری به سیستم ملحق شوند، منابع کمتری برای سرویس دهی در اختیار هر سرویس گیرنده وجود خواهد داشت؛ اما در شبکه های همتا به همتا، با افزایش همتاها ملحق شده و درخواست های سیستم، ظرفیت کلی سیستم نیز افزایش می یابد؛ زیرا سرویس گیرندگان، علاوه بر مصرف منابع، فراهم کننده منابع نیز هستند که این منابع شامل پهنای باند، فضای ذخیره سازی و قدرت محاسباتی می باشند.

تقسیم و کاهش هزینه: راه اندازی یک سیستم متمرکز که بتواند از سرویس گیرنده های زیادی پشتیبانی کند، هزینه زیادی را به سرویس دهنده تحمیل خواهد کرد. معماری همتا به همتا می تواند کمک کند تا این هزینه بین تمام همتاها تقسیم شود.

قابلیت اطمینان: در یک شبکه همتا به همتا به دلیل ماهیت غیرمتمرکز آن، نقطه شکست واحد وجود ندارد و بد عمل کردن یک همتا در مقیاس بزرگ شبکه، تأثیر چشمگیری بر روی کل سیستم نخواهد داشت.

پویایی: سیستم های همتا به همتا دارای محیطی کاملاً پویا هستند، به طوری که همتاها و منابع آن ها می توانند آزادانه به سیستم وارد و از آن خارج شوند.

افزایش خودمختاری: در بسیاری از موارد، کاربران یک شبکه توزیع شده مایل نیستند که متکی به یک سرویس دهنده متمرکز باشند؛ زیرا متکی بودن به یک سرویس دهنده متمرکز باعث محدود شدن آن ها می شود. خودمختاری به این معنا است که همتاها خودشان برای پهنای باند و سایر منابعشان تصمیم گیری می کنند. همچنین همکاری یا عدم همکاری در فعالیت های شبکه، پیوستن به شبکه و ترک کردن آن بر عهده خود همتاهاست.

بعضی از کاربردهای شبکه های همتا به همتا عبارتند از:

اشتراک گذاری فایل: یک شبکه همتا به همتای اشتراک گذاری فایل، پروتکل های جستجو و انتقال داده ها را بر روی پروتکل اینترنت پیاده سازی می کند. برای دسترسی به چنین شبکه ای، کاربران یک برنامه سرویس گیرنده همتا به همتا را دریافت و نصب می کنند. از انواع شبکه های همتا به همتای اشتراک گذاری فایل می توان به بیت تورنت اشاره کرد.

کننده نیاز دارد تا در مورد همتایی تصمیم‌گیری نماید، با استفاده از نظر دست اول خود در مورد آن همتا و با جمع‌آوری نظرات همسایه‌ها و با کمک رابطه اجماع ارائه‌شده، امتیاز دقیق‌تری محاسبه می‌نماید.

همچنین برای بهبود این الگوریتم، الگوریتم اجماع وزن‌دار ارائه شده است که در آن نظرات همسایه‌ها، با توجه به امتیاز شهرت آن‌ها اهمیت پیدا می‌کند. در نتیجه نظر ارائه‌شده توسط همسایه‌ها با امتیاز شهرت بهتر، وزن بیشتری گرفته و در اجماع نظرات تأثیر بیشتری می‌گذارد. همچنین بر اساس درستی یا نادرستی نظر ارائه‌شده توسط همسایه‌ها، به آن‌ها امتیاز داده می‌شود که بر تصمیمات آینده در مورد آن‌ها و نظرات آن‌ها تأثیر می‌گذارد.

۲- مکان‌یابی منابع در سیستم‌های همتا به همتا

۲-۱- الگوریتم Chord

Chord، الگوریتمی بهینه، ساده و مقیاس‌پذیر برای مکان‌یابی ارائه کرده است که در بسیاری از سیستم‌های همتا به همتا مورد استفاده قرار می‌گیرد [۳]. هدف Chord بسیار ساده است و با داشتن یک کلید (به‌عنوان مثال شناسه یک فایل)، همتای صاحب آن را پیدا می‌کند. مزایای زیاد Chord از قبیل توازن بار، غیرمتمرکز بودن و مقیاس‌پذیری، باعث شده است به‌عنوان پروتکل اصلی مکان‌یابی در بسیاری از سیستم‌های همتا به همتا و در این مقاله مورد استفاده قرار گیرد.

الگوریتم Chord با استفاده از توابع درهم‌سازی پایدار به همتاها و داده‌ها یک شناسه m بیتی منحصر به فرد در بازه $[0 \dots 2^m - 1]$ اختصاص می‌دهد. در Chord شناسه هر همتا، درهم شده آدرس IP و درگاه آن است در حالی که شناسه هر داده درهم شده نام آن است. همتاها به ترتیب شناسه روی یک دایره قرار می‌گیرند. جهت جریان داده همیشه در جهت عقربه‌های ساعت است. هر همتا در این دایره، همتای مابعد و ماقبل خود را می‌شناسد. هر داده به اولین همتایی در دایره که شناسه‌اش بزرگ‌تر یا مساوی آن است تعلق می‌گیرد. اگر اندازه شبکه N همتا باشد، هر همتا اطلاعات مسیریابی $O(\log N)$ همتا را که با فواصل لگاریتمی افزایشی در دایره قرار دارند، در یک ساختمان داده با نام جدول انگشتی^۲ نگهداری می‌کند.

زمانی که همتایی بخواهد داده با شناسه k را پیدا کند، پرس‌وجو برای یافتن k را به بزرگ‌ترین همتا در جدول انگشتی که شناسه آن از k کوچک‌تر است، می‌فرستد. این پرس‌وجو تا زمانی ادامه دارد که یک همتا کلید را در همتای مابعد خود بیابد. اثبات می‌شود که با داشتن جدول‌های انگشتی، برای شبکه‌ای با N همتا با $O(\log N)$ پرس‌وجو می‌توان به مقصد رسید [۳].

برای درک بهتر الگوریتم Chord، مثالی در شکل ۱ ارائه شده است. در این مثال ده همتا با شناسه‌های نمایش داده‌شده در دایره همبندی شبکه قرار گرفته‌اند. نحوه شکل‌گیری جدول‌های انگشتی دو همتا با شناسه‌های ۵ و ۳۵ نمایش داده شده است. فرض کنید همتا با شناسه ۳۵ به دنبال کلید ۱۸ می‌گردد. در این حالت، طبق جدول انگشتی

• عدم ارائه سرویس: حالتی است که برخی همتاها با مصرف بیش از حد منابع، مانع از خدمت‌رسانی به سایر همتاها می‌شوند. این رفتار خودخواهانه، علاوه بر کاهش کارایی سیستم، باعث نارضایتی سایر همتاها شده که ممکن است منجر به ترک آن‌ها گردد [۲].

به‌دلیل اهمیت مکان‌یابی در سیستم‌های همتا به همتا آلودگی در نظر گرفته‌شده در این مقاله، آلودگی در فرایند مکان‌یابی است.

عموماً در سیستم‌های همتا به همتا برای مکان‌یابی منابع از جدول‌های درهم‌سازی استفاده می‌شود. جدول درهم‌سازی نوعی ساختمان داده است که مقدرهایی که باید ذخیره شوند را به‌وسیله تابع درهم‌سازی با کلیدهای ویژه‌ای مرتبط می‌سازد. مهم‌ترین هدف این جدول‌ها، تسهیل عملیات مکان‌یابی است. به این معنی که کاربر می‌تواند با سرعتی کارآمد داده مورد نظر خود را در کل ساختمان داده بیابد. همچنین در جدول‌های درهم‌سازی افزودن داده‌های جدید در زمان کم، امکان‌پذیر است. در بسیاری از اوقات، جدول‌های درهم‌سازی بسیار کارآمدتر از درخت‌های جستجو یا هر الگوریتم جستجوی دیگری عمل می‌کند. به همین دلیل، این جدول‌ها به‌طور گسترده در برنامه‌های توزیع‌شده، از قبیل سیستم‌های همتا به همتا مورد استفاده قرار می‌گیرند.

هر منبع (به‌عنوان مثال یک فایل) در سیستم‌های همتا به همتا، به یک همتا تخصیص داده می‌شود که مسئول نگهداری آن است. هدف اصلی اکثر سیستم‌های همتا به همتا مکان‌یابی بهینه این منابع است. جدول‌های درهم‌سازی توزیع‌شده (DHT)، یک ساختمان داده است که با دادن شناسه منبع، مکان‌یابی لازم برای شناسایی همتای مسئول انجام شده و شناسه آن (که معمولاً آدرس IP است) برگردانده می‌شود. در صورتی که در شبکه‌های همتا به همتا همتاها آلوده‌کننده^۴ وجود داشته باشند ممکن است فرایند مکان‌یابی تحت تأثیر آلودگی قرار گیرد و در نتیجه نمی‌توان به نتایج مکان‌یابی اطمینان کرد.

برای ارزیابی آلوده بودن یا نبودن یک همتا می‌توان از مکانیزم‌های مبتنی بر شهرت استفاده کرد. سیستم شهرت^۵، سیستمی است که در یک مجموعه، بر اساس نظرات اعضای مجموعه و تاریخچه نحوه عملکرد هر عضو، به هرکدام یک امتیاز شهرت نسبت می‌دهد. از امتیاز محاسبه‌شده برای هر عضو برای رده‌بندی آن استفاده شده و میزان اعتماد به آن را مشخص می‌کند. سایر اعضا می‌توانند بر اساس این امتیاز، در مورد آن عضو تصمیم‌های مختلفی اتخاذ کنند. در سیستم‌های همتا به همتا، امتیاز منتسب شده به هر همتا نشان‌دهنده آلوده یا سالم بودن آن است [۲].

در این مقاله، راهکاری برای تجمیع نظرات بر اساس قاعده اجماع^۶ ارائه شده است که با کمک آن و با جمع‌آوری امتیاز شهرت محاسبه‌شده توسط همسایه‌ها، همتای محاسبه‌کننده امتیاز می‌تواند تصمیم اعتماد با قابلیت اطمینان بالاتر اتخاذ نماید. الگوریتم اجماع یک قاعده ساده است که به‌صورت محلی به ایجاد توافق بین همسایه‌ها در مورد امتیاز شهرت یک همتا منتهی می‌شود. به این‌صورت که وقتی همتای محاسبه

می‌شود این جستجوها ناکارآمد شود. به بیان ساده‌تر، هر جستجوی افزونه، لزوماً قابلیت اطمینان را بالا نمی‌برد. HALo^۸ الگوریتمی است که بدون تغییرات اساسی در بنیان جدول‌های درهم‌سازی توزیع‌شده، با جستجوهای افزونه مجزا کارایی مکان‌یابی را بهبود می‌بخشد [۴]. ایده اصلی این الگوریتم ساده است: مقصد یک جستجو، در چندین جدول مسیریابی در بین همتهای موجود در جدول‌های درهم‌سازی توزیع‌شده وجود دارد. به این همتها مفصل گفته می‌شود. در واقع، همتهایی که v انگشت آن‌ها است را مفصل‌های همتهای v می‌گویند.

در الگوریتم HALo، به‌جای جستجوی مقصد اصلی در چند مسیر، به دنبال مفصل‌های مقصد اصلی هستیم. در واقع HALo از این واقعیت که «هر همتهای v در جدول مسیریابی $O(\log N)$ همتهای دیگر وجود دارد» استفاده می‌کند. بعد از انجام جستجوهای افزونه، در صورتی که چند کاندیدا برای همتهای مالک کلید یافت شود، نزدیک‌ترین همتهای مالک کلید جستجو انتخاب می‌شود. جستجوهای افزونه در حالت ایده‌آل از مسیرهای متفاوت و مستقلی حرکت کرده و در نتیجه احتمال موفقیت را افزایش می‌دهند. در اینجا l جستجوی متفاوت و مستقل انجام می‌شود که l یک عدد حقیقی کوچک است که به‌عنوان پارامتر افزونگی در این الگوریتم مورد استفاده قرار می‌گیرد. اهمیت مفصل‌ها در این است که آن‌ها، اطلاعات دقیق انگشتان خود را می‌دانند. در واقع اگر ما دنبال همتهای مقصد v باشیم، آدرس IP آن در جدول مسیریابی مفصل‌هایش وجود دارد. به‌طور میانگین هر همتهای در Chord دارای $O(\log N)$ مفصل است که همه در فواصل نمایی متفاوت از آن قرار گرفته‌اند. در حالت ایده‌آل اگر مفصل‌های v پیدا شود و در شبکه همتهای آلوده‌کننده وجود نداشته باشد، همه به v اشاره می‌کنند؛ اما در واقعیت برخی از مفصل‌ها آلوده‌کننده هستند و اطلاعات مسیریابی همراه‌کننده‌ای را برمی‌گردانند. با این حال با داشتن حداقل یک مفصل صادق، می‌توان کوچک‌ترین نتیجه در جهت عقربه‌های ساعت را به‌عنوان مقصد v شناسایی کرد.

از عکس روش پیدا کردن انگشت‌ها برای یافتن مفصل‌ها استفاده می‌شود. فرض کنید به دنبال داده r هستیم و این داده به کلید k نگاشت شده است. P_k مجموعه‌ای از $m = \log N$ تا مکان ماقبل k است. این مجموعه در رابطه (۱) نشان داده شده است.

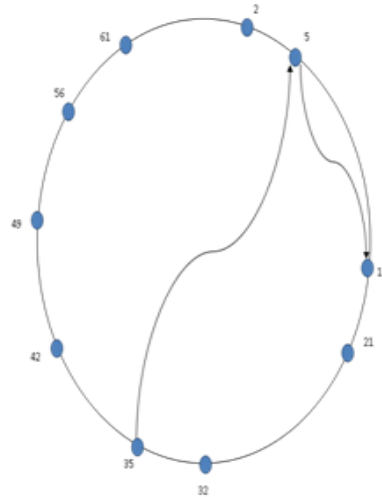
$$P_k = \{k - 2^0, k - 2^1, \dots, k - 2^{m-1}\} \quad (1)$$

وقتی هدف جستجو کلید k است، HALo، l جایگاه بزرگ‌تر P_k که موارد نمایش داده‌شده در (۲) هستند را به‌عنوان کلیدهای جستجو برای مفصل‌های k انتخاب می‌کند. سپس i امین مفصل همتهای v توسط همتهای ماقبل یا مابعد $k - 2^{m-i}$ تخمین زده می‌شود. این l تا مفصل انتخابی را می‌توان با تابع مکان‌یابی در الگوریتم Chord پیدا کرد [۴].

$$\{k - 2^{m-1}, k - 2^{m-2}, \dots, k - 2^{m-l}\} \quad (2)$$

به‌عنوان مثال، شبکه نشان داده‌شده در شکل ۲ را در نظر بگیرید. در دایره هم‌بندی، همتها به ترتیب شناسه خود قرار گرفته‌اند. فرض کنید همتهای 8 با شناسه 8 به دنبال کلید با شناسه 54 باشد. جدول انگشتی همتهای 8 در جدول ۱ نمایش داده شده است. طبق الگوریتم Chord،

جستجو به بزرگ‌ترین انگشت که شناسه آن از کلید کوچک‌تر است داده می‌شود که همتهای 5 است. مجدداً همتهای 5 در جدول انگشتی خود جستجو می‌کند و جستجو را به بزرگ‌ترین انگشت که شناسه آن از 18 کوچک‌تر است محول می‌کند که همتهای 17 است. از آنجایی که کلید 18 بین شناسه همتهای 17 و همتهای مابعد آن با شناسه 21 قرار دارد، همتهای 17 همتهای 21 را به‌عنوان مالک کلید 18 معرفی می‌کند.



جدول انگشتی همتهای ۵			جدول انگشتی همتهای ۱۷		
سطر l	شناسه انگشت $id + 2^l$	شناسه انگشت	سطر l	شناسه انگشت $id + 2^l$	شناسه انگشت
0	5+1	17	0	35+1	42
1	5+2	17	1	35+2	42
2	5+4	17	2	35+4	42
3	5+8	17	3	35+8	49
4	5+16	21	4	35+16	56
5	5+32	42	5	35+32	5

شکل ۱: مثال مکان‌یابی در الگوریتم Chord

با اینکه الگوریتم Chord در برابر پویایی شبکه، اضافه و کم شدن همتها و خراب شدن آن‌ها بسیار مقاوم است، اما به‌راحتی می‌توان اطلاعات مکان‌یابی را در آن تغییر داد. به‌عنوان مثال در یک شبکه با 10000 همتهای 10 درصد همتهای خرابکار، 50 درصد جستجوها ناموفق است [۴].

۲-۲- الگوریتم HALo

همان‌طور که اشاره شد وجود همتهای آلوده‌کننده می‌تواند فرآیند مکان‌یابی را تحت تأثیر قرار دهد. این همتها ممکن است پرس‌وجوها را از مسیر درست به بیراهه هدایت کنند تا کارایی سیستم کم شود. همچنین ممکن است یک همتهای نادرست را به‌عنوان مالک یک داده معرفی کنند. یکی از راهکارهای ارائه‌شده برای بالا بردن قابلیت اطمینان، جستجوهای افزونه برای مسیریابی مقصد است. چالش اصلی جستجوهای افزونه این است که به‌دلیل ماهیت مسیریابی در جدول‌های درهم‌سازی توزیع‌شده، نتیجه چند جستجو با کلید یکسان، به یک مجموعه همتهای نزدیک به همتهای مقصد همگرا می‌شوند که باعث

این، طول مسیرهای جستجو به صورت لگاریتمی با افزایش اندازه شبکه زیاد می‌شود.

۲-۳- الگوریتم ReDS

الگوریتم ReDS^۱ چارچوبی برای استفاده از سیستم‌های شهرت ارائه کرده است به نحوی که قابلیت اطمینان جدول‌های درهم‌سازی توزیع شده را افزایش داده اما هزینه محاسباتی کمتری نسبت به روش‌های مبتنی بر جستجوی افزونه دارد [۶]. به طور خلاصه ReDS دو هدف عمده را دنبال می‌کند:

- ۱- آغازکننده یک جستجو، بهترین همتا را برای نقطه آغاز جستجوهای افزونه انتخاب کند که منجر به بهبود ناحیه‌ای می‌شود.
- ۲- هر همتای مشارکت‌کننده در جستجو بهترین انگشت را انتخاب کرده و در نتیجه در هر قدم از انتخاب همتای آلوده جلوگیری شود. این امر بهبود مشارکتی نامیده می‌شود.
- برای پیاده‌سازی ReDS بر روی جدول‌های درهم‌سازی توزیع شده چهار شرط زیر مورد نیاز است:
 - ۱- امکان جستجوی افزونه در مسیرهای متمایز وجود داشته باشد.
 - ۲- هر همتای جستجوکننده باید قابلیت انتخاب انگشت‌های جدول مسیریابی‌اش را داشته باشد.

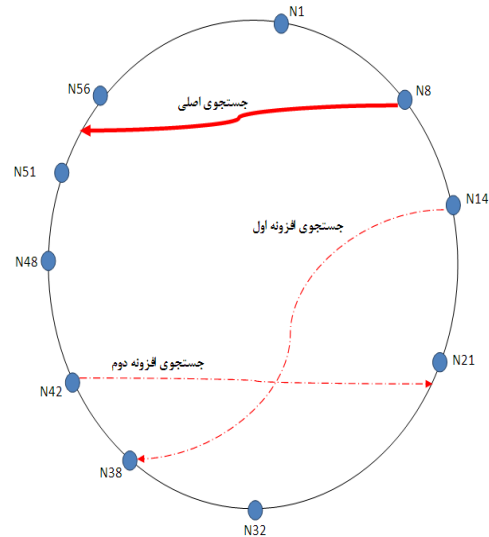
- ۳- با در دست بودن مجموعه‌ای از مقصدها بر مبنای جستجوی افزونه، انتخاب مقصد اصلی از بین آن‌ها ممکن باشد.
- ۴- موفقیت یا شکست یک جستجو، قابل پیگیری باشد تا انگشت مسئول آن جستجو شناسایی شود.

شرط اول اساس سیستم‌هایی از قبیل HALo را نشان می‌دهد چرا که بدون جستجوهای افزونه نرخ موفقیت در حضور همتهای آلوده بسیار کاهش می‌یابد. در مورد شرط دوم، توجه به این نکته ضروری است که انتخاب از بین هر کدام از انگشتان، نباید قوانین جدول‌های درهم‌سازی توزیع شده را نقض کند. علی‌الخصوص این شرط که هر سطر جدول مسیریابی، مسیر رسیدن به مقصد را نصف کند. شرط سوم باعث می‌شود پس از انجام جستجوهای افزونه، از بین تمامی مقصدهای پیداشده مقصد حقیقی انتخاب شود. لازم به ذکر است در Chord، نزدیک‌ترین مقصد به کلید جستجو، مقصد حقیقی است. شرط چهارم نیز باعث می‌شود که موفقیت یا شکست یک جستجو به یک انگشت خاص نسبت داده شود.

جدول ۱: جدول انگشتی همتا با شناسه ۸ در مثال ارائه شده

سطر i	id + 2 ⁱ	شناسه انگشت
0	9	14
1	10	14
2	12	14
3	16	21
4	32	32
5	40	42

جستجو به بزرگ‌ترین شناسه جدول انگشتی که مقدار آن از شناسه مقصد کوچک‌تر است داده می‌شود. طبق جدول انگشتی همتا با شناسه ۸، جستجو به همتا با شناسه ۴۲ محول می‌شود. به طور مشابه، جستجو از همتای ۴۲ به همتای ۵۱ و سپس به همتای ۵۶ که مالک کلید ۵۴ است محول می‌شود. در نهایت همتا با شناسه ۵۶ به عنوان مقصد برگردانده خواهد شد.



شکل ۲: جستجوی افزونه در HALo

در الگوریتم HALo، برای جلوگیری از تأثیر آلودگی از جستجوهای افزونه استفاده می‌شود. فرض کنید پارامتر افزونگی برابر با ۲ باشد. در نتیجه دو مفصل کلید ۵۴ به عنوان دو مقصد جستجوی افزونه انتخاب می‌شود. این مقصدها به صورت نمایش داده شده در رابطه (۳) انتخاب می‌شوند.

$$\{54 - 2^{6-1}, 54 - 2^{6-2}\} \quad (3)$$

حال به جای جستجوی کلید ۵۴، به دنبال مفصل‌های آن با شناسه‌های ۲۲ و ۳۸ هستیم. این دو جستجوی افزونه، توسط دو تا از انگشت‌های همتای ۸، مطابق با الگوریتم جستجوی Chord صورت می‌پذیرد.

HALo از الگوریتم‌های مقاوم مکان‌یابی در سیستم‌های همتا به همتا است که همگام با Salsa [۵] اولین قدم را در جهت ایمن نمودن سرویس‌های مکان‌یابی برداشته است. این الگوریتم فرآیند جستجوی منبع را به صورت چندباره استفاده کرده است تا با احتمال بالا، اطلاعات درست‌تری را پیدا کند. گرچه جدول‌های درهم‌سازی توزیع شده با قابلیت جستجوی افزونه، تأثیر آلودگی‌های همتهای خرابکار را کم کرده است اما با افزایش تعداد جستجوها، هزینه ارتباطی تحمیل شده بر شبکه را افزایش داده است. در واقع هزینه ارتباطی در HALo، $O(\log N)$ برابر بیشتر از مکان‌یابی بدون جستجوی افزونه است و همچنین هزینه محاسباتی ناشی از یافتن مفصل را به شبکه تحمیل کرده است. علاوه بر

معینی است که همتهای عضو سیستم در مورد آن اعلام نظر کرده‌اند. این توافق با تبادل این نظرات به صورت توزیع شده به دست می‌آید. الگوریتم اجماع، به شکل تکراری با استفاده از اطلاعات تبادل شده بین یک عضو و همسایه‌هایش، سعی در همگن کردن نظرات دارد. حل مسئله اجماع، یک وظیفه مشارکتی است که جز با همکاری اعضا ممکن نخواهد شد. از الگوریتم اجماع در علوم مختلفی استفاده می‌شود که هدف همه آن‌ها ایجاد توافق بین اعضا در یک مجموعه پویا است. بر اساس پارامتر مورد توافق، تصمیمات متعددی در این مجموعه‌ها اتخاذ می‌گردد.

فرض کنید همتهای i نیازمند بررسی پارامتر x است. برای این کار نیاز به مشارکت بقیه اعضای شبکه دارد. برای حل مسئله اجماع، شبکه و ارتباطات بین اعضا با گراف $G = (V, E)$ نمایش داده می‌شود. در این گراف، اعضا در مجموعه $V = \{1, 2, \dots, n\}$ و ارتباطات با $E \subset V \times V$ نمایش داده می‌شود. در هم‌بندی شبکه، همسایه‌های کاربر i مطابق با رابطه (۵) تعیین می‌شوند.

$$N_i = \{j \in V, j \neq i: (i, j) \in E\} \quad (5)$$

الگوریتم اجماع تکراری در بازه‌های زمانی گسسته t طبق رابطه (۶) تعریف می‌شود.

$$x_i(t+1) = x_i(t) + \alpha \sum_{j \in N_i} (x_j(t) - x_i(t)) \quad (6)$$

در رابطه (۶)، $x_i(t)$ مقدار محاسبه شده پارامتر x توسط عضو i در زمان t و N_i فهرست همسایه‌های i است. در این رابطه α اندازه گام است که بازه اندازه آن در رابطه (۷) نشان داده شده است.

$$0 < \alpha < \frac{1}{\max |N_i|} = \frac{1}{\Delta} \quad (7)$$

Δ حداکثر یال اعضا در شبکه است که به آن حداکثر درجه شبکه می‌گویند. در زمان‌های گسسته $t = 0, 1, 2, \dots$ و در هر تکرار الگوریتم اجماع، اعضا مقادیر وضعیت خود را با همسایه‌ها مبادله می‌کنند. بر اساس رابطه (۶) پارامتر مورد بحث به روزرسانی شده تا هنگامی که توافق حاصل شود [۷].

۳-۲- استفاده از الگوریتم اجماع برای مبارزه با آلودگی در سیستم‌های همتا به همتا

هدف این مقاله استفاده از الگوریتم اجماع در سیستم‌های همتا به همتا به منظور به کارگیری نظرات سایر همتاها در مورد هر همتا برای محاسبه امتیاز شهرت آن همتا است. در مقاطع زمانی مختلف همتهای i به عنوان تصمیم‌گیرنده نیاز دارد تا برای همتهای m امتیاز شهرت را محاسبه کرده و در مورد آن تصمیم بگیرد. از امتیازات شهرت برای انتخاب بهترین همتاها در فرآیند مکان‌یابی استفاده می‌شود به نحوی که همتهای آلوده‌کننده، به مرور زمان در سیستم امتیاز منفی گرفته و در سیستم شهرت بدنام می‌شوند. در نتیجه، همتهای آلوده‌کننده توسط سایر همتاها مورد استفاده قرار نرفته و منزوی می‌شوند. در سیستم شهرت مبتنی بر الگوریتم اجماع، همتهای تصمیم‌گیرنده i به نظر شخصی خود در مورد همتهای m اکتفا نکرده و از سایر همتاها در مورد آن نظرسنجی

در بسیاری از سیستم‌های مکان‌یابی از قبیل HALO این انتساب مشخص است زیرا هر جستجو به صورت مستقل صورت می‌گیرد [۶].

در HALO جستجوها اغلب بازگشتی هستند بدین معنا که همتهای جستجوکننده از انگشت‌های خود درباره مفصل‌های مقصد پرس‌وجو می‌کند. در نتیجه می‌توان موفقیت و یا شکست یک جستجو را به آن انگشت نسبت داد. در نتیجه شرط یک و سه در HALO وجود دارد. چالش اصلی این است که انگشت‌ها ثابت و غیر قابل انتخاب هستند که برخلاف شرط دوم است. برای این منظور یک سید b تایی برای هر همتا ایجاد می‌شود. این سید شامل کلید انگشت $(i + 2^i)$ شامل صاحب آن کلید (انگشت اصلی) و $b - 1$ همتهای ماقبل است. وقتی نزدیک‌ترین انگشت به کلید مقصد k توسط همتهای x جستجو می‌شود، x پس از انتخاب سطر درست در جدول انگشتی، امتیاز شهرت b انگشت موجود در آن سطر را بررسی کرده و بهترین انگشت را انتخاب می‌نماید. با این بهبود در HALO می‌توان ReDS را روی آن پیاده کرد.

در سیستم شهرت تشکیل شده در ReDS، از روشی با نام Drop-off استفاده می‌شود که عملکرد آن در ادامه تشریح می‌گردد: امتیازات محاسبه شده توسط سایر همتاها در صورتی که به امتیاز محاسبه شده توسط همتهای اولیه شباهت بیشتری داشته باشند، ارزش بیشتری خواهند داشت. در این روش فرض شده است که اطلاعات دست اول در مورد امتیازات شهرت، اعتبار بیشتری داشته در نتیجه هر چه امتیاز ارائه شده توسط سایر همتاها به این امتیازات نزدیک‌تر باشد، احتمال درست بودن این امتیاز بیشتر است. فرض کنید $r_k(f)$ امتیاز شهرت دست اول محاسبه شده توسط همتهای k برای انگشت f باشد. برای محاسبه امتیاز شهرت همتهای f ، همتهای محاسبه کننده k از مفصل مشترک z پرس‌وجو می‌کند تا نظرش را در مورد همتهای f بداند. امتیاز محاسبه شده توسط همتهای z برای همتهای f با $r_j(f)$ نمایش داده می‌شود. همتهای k احتمال درست بودن نظر همتهای z را که با w نشان داده می‌شود به همراه امتیاز ارائه شده در انبار امتیازات نگهداری می‌نماید. احتمال درست بودن نظر همتهای z طبق رابطه (۴) محاسبه می‌شود.

$$w = 1 - |r_j(f) - r_k(f)| \quad (4)$$

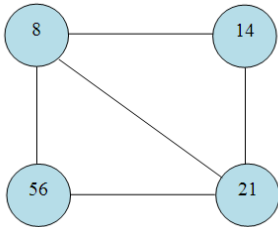
امتیاز نهایی محاسبه شده برای همتهای f توسط همتهای k ، میانگین امتیازات موجود در انبار امتیاز می‌باشد. نکات منفی روش Drop-off این است که برای تجمع آراء، روش میانگین‌گیری روش دقیقی محسوب نشده و همچنین در این روش هیچ تاریخچه‌ای از عملکرد همتاها نگهداری نمی‌شود.

۳- محاسبه امتیاز شهرت با الگوریتم اجماع

۳-۱- معرفی الگوریتم اجماع

الگوریتم اجماع، یک قاعده هماهنگ‌سازی موضعی بسیار ساده است که به ایجاد توافق در سطح گروهی می‌انجامد و وجود نقطه تصمیم‌گیری متمرکز و یا اطلاعات سراسری و جامع در این الگوریتم ضروری نیست. موضوع اصلی در مسئله اجماع، دستیابی به توافق در خصوص پارامتر

(۹) یک رابطه خطی ساده است که با ایجاد سربار محاسباتی قابل چشم‌پوشی، به همتهای i اجازه می‌دهد که در زمان تصمیم‌گیری برای انتخاب همتا، با محاسبه امتیاز شهرت بهترین تصمیم را اخذ نماید. به اشتراک‌گذاری امتیاز شهرت، بین همتهایی که انگشت مشترک دارند انجام می‌شود. در واقع هر همتا برای هر انگشت خود، لیست مفصل‌های مشترک را نگهداری می‌کند. از امتیاز ارائه‌شده توسط مفصل‌های مشترک استفاده می‌شود تا امتیاز همتهای مورد نظر محاسبه شود.



شکل ۳: گراف همسایگی برای یافتن امتیاز همتهای ۳۲ در مثال مورد بررسی

محاسبه امتیاز با الگوریتم اجماع ارائه‌شده، توسط همتهای i در دو لحظه صورت می‌گیرد:

۱- زمانی که همتهای i به‌عنوان آغازگر جستجو به دنبال کلید k می‌گردد، بایستی با توجه به پارامتر l که تعداد جستجوهای افزونه را نشان می‌دهد، در جدول انگشتی خود l تا از بهترین همتهایش را به‌عنوان آغازگر جستجوهای افزونه انتخاب نماید. در این لحظه برای انتخاب بهترین همتاها، i نیازمند محاسبه امتیاز شهرت انگشت‌هایش است. اگر تعداد همتهای شبکه N باشد، همتهای i برای تمام $O(\log N)$ همتهای موجود در جدول انگشتی، امتیاز شهرت را طبق رابطه (۹) محاسبه کرده و با توجه به امتیازات، l همتا با امتیاز بیشتر را به‌عنوان آغازگر جستجوهای افزونه برمی‌گزیند.

۲- در هر گام الگوریتم مکان‌یابی برای رسیدن به کلید جستجوهای افزونه، همتهای i نیازمند انتخاب بهترین همتهای موجود در سبد است تا جستجو در گام بعد را به آن محول کند. استفاده از بهترین همتهای موجود در سبد در هر گام الگوریتم، احتمال استفاده از همتهای آلوده را در هر مرحله کاهش داده و در نهایت باعث افزایش نرخ موفقیت جستجوها می‌شود. برای این کار، همتهای i با توجه به انگشتی که باید جستجو را به آن محول کند، امتیاز همه b همتهای موجود در سبد را محاسبه کرده و سپس همتا با بیشترین امتیاز شهرت را برمی‌گزیند. گام بعدی الگوریتم مکان‌یابی برای رسیدن به کلید مورد نظر، توسط همتهای انتخاب‌شده از سبد صورت می‌گیرد.

اگر همتهایی تازه به سیستم پیوسته باشد و تاریخچه‌ای از نحوه عملکرد آن و امتیازات شهرتش موجود نباشد، امتیاز اولیه صفر برای آن در نظر گرفته می‌شود. امتیاز دست اول هر همتا بدون در نظر گرفتن

می‌کند. با استفاده از این نظرات و مشارکت سایر همتاها، همتهای i با همسایه‌هایش به توافق رسیده و در مورد همتهای m تصمیم‌گیری می‌نماید. برای محاسبه امتیاز شهرت همتهای m توسط همتهای i از الگوریتم اجماع تکراری طبق رابطه (۹) استفاده می‌شود. در رابطه (۹)، N_{im} فهرست همه همسایه‌های همتهای i برای محاسبه امتیاز همتهای m در هم‌بندی شبکه بوده و α پارامتر اندازه گام است. اگر تعداد همتهای موجود در شبکه N باشد، بازه اندازه α مطابق با رابطه (۸) است:

$$0 < \alpha < \frac{1}{O(\log N)} \quad (8)$$

هر همتهای i که نیازمند تصمیم‌گیری در مورد همتهای m در لحظه t باشد، با جمع‌آوری نظرات همسایه‌هایش در مورد آن همتا و استفاده از رابطه (۹) قادر به محاسبه امتیاز شهرت در یک لحظه بعد یعنی در زمان $t + 1$ خواهد بود. در این رابطه $x_{im}(t)$ امتیاز همتهای محاسبه‌کننده i در مورد همتهای m در لحظه t و $x_{jm}(t)$ امتیاز ارائه‌شده توسط همتهای همسایه j در مورد همتهای m در لحظه t می‌باشد.

$$x_{im}(t + 1) = x_{im}(t) + \alpha \sum_{j \in N_{im}} (x_{jm}(t) - x_{im}(t)) \quad (9)$$

لازم به ذکر است در سیستم‌های همتا به همتهای بزرگ در مقیاس اینترنت، امکان توافق بین تمامی اعضای سیستم وجود ندارد. اصولاً توافق سراسری در این سیستم‌ها معنایی نداشته و تنها توافق محلی مورد نظر است. با توجه به ساختار توزیع‌شده سیستم‌های همتا به همتا، محاسبه امتیازها توسط هر همتا به‌صورت مستقل انجام می‌شود.

برای محاسبه امتیاز شهرت همتهای m ، گراف همسایگی $NG_m = (V, E)$ شکل می‌گیرد که V مجموعه گره‌های گراف و E مجموعه یال‌های آن است. گراف همسایگی NG_m به‌صورت زیر تشکیل می‌شود:

- مفصل‌های مشترک همتهای m که با JK_m نمایش داده می‌شود، گره‌های گراف را تشکیل می‌دهند.
- بین گره i_1 و i_2 یال قرار دارد اگر i_1 در جدول انگشتی i_2 باشد و یا بالعکس.

برای شرح نحوه ساخته شدن گراف همسایگی در الگوریتم اجماع، فرض کنید در مثال مطرح‌شده در شکل ۲، به دنبال یافتن امتیاز همتا با شناسه ۳۲ هستیم. در جهت رسیدن به اجماع، برای پیدا کردن امتیاز همتهای ۳۲، مفصل‌های مشترک آن باید به تبادل نظر بپردازند. مفصل‌های مشترک همتهای ۳۲ همتاها می‌باشند که همتهای ۳۲ انگشت آن‌ها می‌باشد. همتا با شناسه‌های ۸، ۱۴، ۲۱ و ۵۶ مفصل‌های مشترک همتهای ۳۲ می‌باشند. گراف همسایگی محاسبه امتیاز همتهای ۳۲، به‌صورت زیر ساخته می‌شود:

- مفصل‌های مشترک همتهای ۳۲ گره‌های گراف هستند.
- در صورتی بین گره i_1 و i_2 یال قرار دارد که یا i_1 در جدول انگشتی i_2 باشد و یا بالعکس.

گراف همسایگی ساخته‌شده برای یافتن امتیاز همتهای ۳۲ در شکل ۳ نشان داده شده است.

همتای ۲۱ الگوریتم اجماع را برای همتای ۵۶ و همتای ۳۸ اجرا می‌نماید و نهایتاً بهترین تصمیم اعتماد را بر اساس امتیازات شهرت محاسبه شده، اخذ می‌کند. فرض کنید همتای ۵۶ امتیاز بهتری می‌گیرد در نتیجه این همتا به‌عنوان ادامه‌دهنده الگوریتم جستجو توسط همتای ۲۱ انتخاب می‌شود. در تمامی قدم‌های رسیدن به مقصد، در صورتی که همتای آلوده‌ای درگیر نشود، این روال تکرار خواهد شد و با محاسبه امتیاز همتاهای موجود در سبد، بهترین انتخاب برای گام بعدی انجام می‌شود. در انتهای جستجو، با توجه به موفقیت یا عدم موفقیت در یافتن کلید مورد نظر، همتای ۲۱ که آغازگر این جستجو است، به تمامی همتاهای مشارکت‌کننده در این جستجو در کل مسیر، مطابق با رابطه (۱۰) امتیاز می‌دهد.

۴- ارزیابی راهکار مبتنی بر اجماع

برای ارزیابی راهکار اجماع ارائه شده در این مقاله، نتایج حاصل از شبیه‌سازی آن در این بخش ارائه شده و معیارهای ارزیابی مختلفی برای آن مورد بررسی قرار می‌گیرد.

برای شبیه‌سازی الگوریتم‌ها از شبیه‌ساز معرفی شده در [۸] استفاده شده است. در این شبیه‌ساز نسخه ابتدایی Chord با زبان برنامه‌نویسی جاوا پیاده‌سازی شده است. در این مقاله، با توسعه در این شبیه‌سازی بستر لازم جهت جستجوی مقیاس‌پذیر در الگوریتم Chord و پیاده‌سازی HALo و ReDS فراهم شده است. با شروع شبیه‌سازی، با توجه به تعداد همتاهای موجود در شبکه که به‌عنوان پارامتر ورودی در نظر گرفته شده است، N همتا با IP تصادفی تولید می‌شود. با درهم‌سازی IP طبق الگوریتم SHA-1 ۱۶۰ بیتی، شناسه‌های همتاها شکل می‌گیرد. با توجه به این شناسه‌ها، همتاها در دایره به‌صورت صعودی در یک مسیر ساعت‌گرد قرار می‌گیرند. با توجه به نرخ آلودگی همتاها که پارامتر ورودی است، از N همتای شبکه، p درصد آن‌ها آلوده در نظر گرفته می‌شود. برای اینکه در طول آزمایش یک شبکه با نرخ شکست ثابت، نتایج تحت تأثیر تغییر تعداد همتاهای آلوده قرار نگیرد، نزدیک‌ترین تعداد از همتاها به درصد آلودگی در نظر گرفته می‌شود. انتخاب همتاهای آلوده از بین اعضای شبکه به‌صورت اتفاقی صورت می‌گیرد.

تعداد جستجوها به‌عنوان پارامتر ورودی شبیه‌سازی است. مبدأ هر جستجو از همتاهای شبکه و مقصد در فضای شناسه شبکه به‌صورت اتفاقی انتخاب می‌شوند. ثابت بودن یا نبودن شبکه، به‌عنوان پارامتر ورودی شبیه‌سازی در نظر گرفته شده است. در صورتی که شبکه ثابت نباشد، طبق فرض الگوریتم ReDS بین هر دو جستجو، تعدادی همتا به شبکه اضافه شده و به همان تعداد، شبکه را ترک می‌کنند.

۴-۱- بررسی نرخ شکست جستجوها

در شکل ۴ نرخ شکست جستجوهای چهار الگوریتم Chord، HALo، ReDS و راهکار پیشنهادی مبتنی بر اجماع، بر اساس نرخ آلودگی همتاها مقایسه شده است. نتایج ارائه شده متوسط انجام ۳۰ بار شبیه‌سازی است که به همراه بازه اطمینان ۹۵ درصد ارائه شده است. اندازه شبکه برابر با

نظرات سایر همتاها، از تقسیم تعداد جستجوهای موفق بر تعداد کل جستجوهای که همتای m در آن مشارکت کرده است، محاسبه می‌شود که در رابطه (۱۰) نشان داده شده است.

$$reputation_score = \frac{number_of_successful_lookups + 1}{number_of_lookups + 1} \quad (10)$$

۳-۳- مثال نحوه مکان‌یابی در راهکار مبتنی بر اجماع

فرض کنید در شبکه شکل ۲، همتا با شناسه ۸ به دنبال کلید با شناسه ۴۹ می‌گردد. با توجه به اندازه شبکه، تعداد جستجوی افزونه و اندازه سبد هر دو برابر با دو در نظر گرفته می‌شود. همتای ۸ برای شروع جستجوهای افزونه، دو انگشتی که بیشترین امتیاز شهرت را کسب کرده‌اند به‌عنوان آغازگر جستجوهای افزونه انتخاب می‌کند. برای این کار، همتای ۸ برای تمام همتاهای موجود در جدول انگشتی، الگوریتم اجماع را محاسبه کرده تا با در نظر گرفتن تجارب قبلی خود از عملکرد انگشت‌ها و نظرات سایر همتاها، امتیاز تمامی انگشت‌ها را محاسبه نماید. با توجه به فضای شناسه در این مثال، اندازه جدول انگشتی برابر با ۶ است. در نتیجه محاسبه امتیاز همتاها ۶ بار اجرا می‌شود.

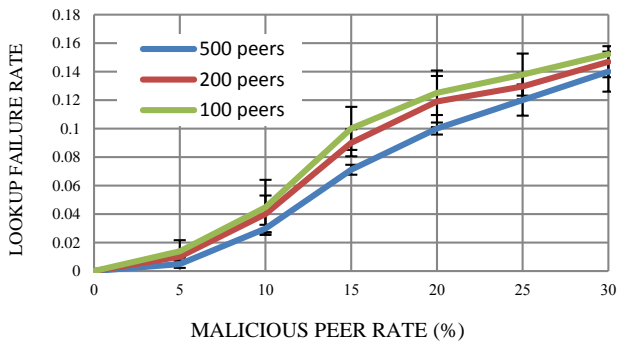
فرض کنید بیشترین امتیاز محاسبه شده مربوط به همتاها با ۱۴ و ۲۱ باشد. این دو همتا به‌عنوان قابل اعتمادترین انگشت‌های همتای ۸ انتخاب شده و آغازگر جستجوهای افزونه هستند. در جستجوهای افزونه، به‌جای گشتن به دنبال کلید ۴۹، به دنبال دو مفصل آن می‌گردیم. مفصل‌های کلید ۴۹ به‌صورت رابطه (۱۱) به‌عنوان مقصدهای جستجوهای افزونه انتخاب می‌شوند.

$$\begin{aligned} k_{49}(1) &= 49 - 2^5 = 17 \\ k_{49}(2) &= 49 - 2^4 = 33 \end{aligned} \quad (11)$$

در نتیجه، به‌جای اینکه همتای ۸ دنبال کلید ۴۹ بگردد، همتای ۱۴ دنبال کلید ۳۳ و همتای ۲۱ دنبال کلید ۱۷ می‌گردد. فرض شده است در شکل ۲ دو همتای ۱۴ و ۴۲ آلوده هستند. در اولین جستجوی افزونه، به‌دلیل آلوده بودن همتای آغازگر، همتای ۱۴ اولین همتای آلوده بعد از خود را به‌عنوان مالک کلید برمی‌گرداند. لازم به ذکر است همتای آلوده هیچ محاسبه امتیاز شهرتی انجام نمی‌دهد. به این‌صورت همتای آلوده ۱۴ در سیستم شهرت اختلال ایجاد می‌کند.

روند جستجوی افزونه دوم به‌صورت زیر است:

همتای ۲۱ به دنبال کلید ۱۷ می‌گردد. در صورتی که الگوریتم مکان‌یابی استفاده شده، مطابق با الگوریتم Chord باشد، همتای ۲۱ طبق جدول انگشتی خود، همتای ۵۶ را به‌عنوان قدم بعدی الگوریتم جستجو انتخاب می‌کند؛ اما در اینجا، برای اینکه در هر قدم از الگوریتم، همتا بتواند بهترین انتخاب ممکن را برای گام بعدی داشته باشد، سبدهای آن، اندازه دو تشکیل می‌شود. در این سبد همتای ۵۶ و همتای قبلی آن، ۳۸ قرار دارند. حال همتای ۲۱ باید بین دو همتای موجود در سبد، تصمیم اعتماد گرفته و قابل اعتمادترین همتا را به‌عنوان قدم بعدی الگوریتم جستجو انتخاب نماید. برای محاسبه امتیاز شهرت این دو همتا،



شکل ۵: مقایسه عملکرد الگوریتم اجماع در شبکه با ابعاد مختلف

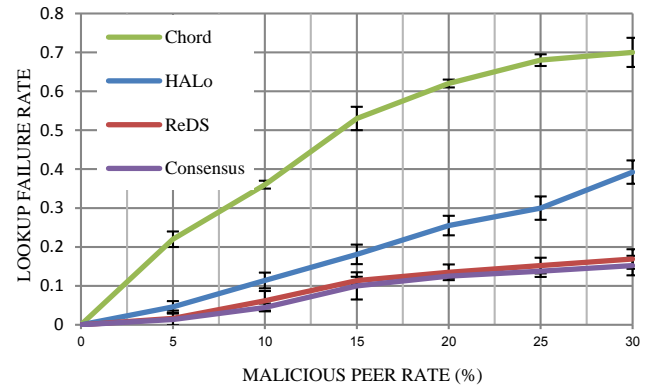
همان‌طور که در شکل ۵ مشاهده می‌شود با بزرگ شدن شبکه، عملکرد الگوریتم اجماع نه تنها بدتر نشده بلکه با توزیع بهتر کلیدها در شبکه و همکاری همتهای سالم، نرخ موفقیت جستجوها بهبود می‌یابد. با توجه به این عملکرد، ادعای مقیاس‌پذیری برای این الگوریتم درست بوده و استفاده از آن در شبکه‌های بزرگ، ممکن است.

۴-۳- بررسی تعداد گام‌های الگوریتم جستجو

از دیگر پارامترهای ارزیابی راهکار ارائه‌شده، تعداد گام‌های الگوریتم جستجو برای یافتن مقصد است. برای بررسی این پارامتر در الگوریتم اجماع، شبکه‌ای با ۱۰۰ همتا و ۱۰۰۰ جستجو شبیه‌سازی شده است. نتایج حاصل از مقایسه میانگین تعداد گام‌های الگوریتم جستجو در دو روش Drop-off استفاده‌شده در ReDS و الگوریتم اجماع در شکل ۶ نمایش داده شده است.

همان‌طور که مشاهده می‌شود میانگین تعداد گام‌های الگوریتم جستجو در راهکار مبتنی بر اجماع کمی بیشتر از تعداد گام‌های الگوریتم جستجو در ReDS است. علت این افزایش، تلاش برای یافتن مقصد جستجو، حتی در حضور همتهای آلوده می‌باشد. به بیان دیگر، الگوریتم اجماع با تشکیل سیستم شهرت و امتیازدهی به همتها، باعث می‌شود بهترین تصمیم‌های اعتماد توسط همتها اخذ گردد که طی آن ممکن است برای رسیدن به مقصد در حضور همتهای آلوده، تعداد گام‌های الگوریتم جستجو افزایش یابد. لازم به ذکر است، ساختار Chord تضمین می‌کند که در شبکه‌ای عاری از همتهای آلوده، تمامی جستجوها در $O(\log N)$ گام به مقصد برسد. همان‌طور که در شکل ۶ قابل مشاهده است، در الگوریتم اجماع پیشنهادی، حتی با حضور ۳۰ درصد همتهای آلوده‌کننده همچنان این ویژگی اصلی در الگوریتم پایه، حفظ شده است. در شبکه‌ای با ۱۰۰ همتا در حضور ۳۰ درصد همتهای آلوده‌کننده، به‌طور میانگین الگوریتم جستجو در کمتر از هفت گام به مقصد رسیده است.

۱۰۰ همتا و تعداد جستجوها برابر با ۱۰۰۰ است. نرخ همتهای آلوده از صفر تا سی درصد افزایش می‌یابد. اندازه سبد برابر با ۲ و تعداد جستجوهای افزونه برابر با ۳ است. همچنین اندازه گام در الگوریتم اجماع برابر با ۰/۱ در نظر گرفته شده است.



شکل ۴: مقایسه نرخ شکست جستجوها در چهار الگوریتم

همان‌طور که در شکل ۴ مشاهده می‌شود، الگوریتم Chord در شبکه آلوده بسیار بد عمل کرده و در حضور ۳۰ درصد همتهای آلوده‌کننده، حدود ۷۰ درصد جستجوها ناموفق هستند. الگوریتم HALo با انجام سه جستجوی افزونه توانسته است نرخ شکست جستجوها را تا حد زیادی بهبود دهد. این نرخ، با تشکیل سیستم شهرت در ReDS باز هم بهبود پیدا کرده است به‌صورتی که در حضور ۳۰ درصد همتهای آلوده‌کننده کمتر از ۲۰ درصد جستجوها ناموفق هستند. الگوریتم اجماع، با استفاده از نظرات شخصی و همسایه‌های همتها توانسته است نرخ شکست جستجوها را نسبت به ReDS، بیش از ۱۵ درصد بهبود ببخشد.

۴-۲- بررسی عملکرد الگوریتم اجماع در ابعاد مختلف شبکه

به دلیل عدم نیاز به محاسبه‌کننده مرکزی و ارتباط محلی همتها با یکدیگر، الگوریتم اجماع ارائه‌شده کاملاً توزیع‌شده و مقیاس‌پذیر است و تمامی محاسبات توسط خود همتها صورت می‌گیرد. برای بررسی تأثیر اندازه شبکه بر عملکرد الگوریتم اجماع، در شکل ۵، سه شبکه با ابعاد ۱۰۰، ۲۰۰ و ۵۰۰ با یکدیگر مقایسه شده‌اند. لازم به ذکر است، مقادیر پارامترهای جستجوی افزونه و اندازه سبد با توجه به اندازه شبکه، متغیر است. برای شبکه با ۲۰۰ همتا پارامتر تعداد جستجوهای افزونه برابر با پنج و اندازه سبد برابر با سه و برای شبکه با ۵۰۰ همتا، پارامتر تعداد جستجوهای افزونه برابر با هفت و اندازه سبد برابر با چهار در نظر گرفته شده است. تعداد جستجوها برابر با ۱۰۰۰، اندازه گام برابر با ۰/۱ و تعداد تکرار شبیه‌سازی ۱۰ مرتبه است.

می‌شود. به این منظور بعد از پایان یافتن جستجو و مشخص شدن موفق بودن یا نبودن آن، تمامی همتهای محاسبه کننده امتیازات شهرت، به بررسی نظرات ارائه شده توسط همسایه‌های خود می‌پردازند. بر اساس این بررسی‌ها، طبق نظر ارائه شده توسط همسایه و درستی آن، به وی امتیاز مثبت یا منفی داده می‌شود. تخصیص امتیاز به همسایه‌های ارائه‌دهنده نظر به صورت زیر است:

- در صورتی که جستجو موفق باشد و همسایه مورد نظر در مورد همتای مورد بحث، امتیاز مثبت ارائه کرده باشد، همتای محاسبه‌کننده امتیاز، به این همسایه یک امتیاز مثبت می‌دهد. مشابه حالتی که این همتا، در فرآیند جستجو دخیل بوده است.
- در صورتی که جستجو موفق باشد و همسایه مورد نظر در مورد همتای مورد بحث، امتیاز منفی ارائه کرده باشد، همتای محاسبه‌کننده امتیاز، به این همسایه یک امتیاز منفی می‌دهد.
- در صورتی که جستجو ناموفق باشد و همسایه مورد نظر در مورد همتای مورد بحث، امتیاز مثبت ارائه کرده باشد، همتای محاسبه‌کننده امتیاز، به این همسایه یک امتیاز منفی می‌دهد.
- در صورتی که جستجو ناموفق باشد و همسایه مورد نظر در مورد همتای مورد بحث، امتیاز منفی ارائه کرده باشد، همتای محاسبه‌کننده امتیاز، به این همسایه یک امتیاز مثبت می‌دهد.
- در صورتی که همسایه مورد نظر در مورد همتای مورد بحث، امتیاز صفر ارائه کرده باشد، همتای محاسبه‌کننده امتیاز، به این همسایه امتیازی تخصیص نمی‌دهد.

لازم به ذکر است طبق رابطه (۱۰) امتیاز شهرت با تعداد جستجوهای موفق تقسیم بر تعداد کل جستجوها برابر است. حال برای استفاده از امتیازات تخصیص داده شده به همتاها، الگوریتم اجماع وزن دار شده به صورت رابطه (۱۲) تغییر پیدا می‌کند.

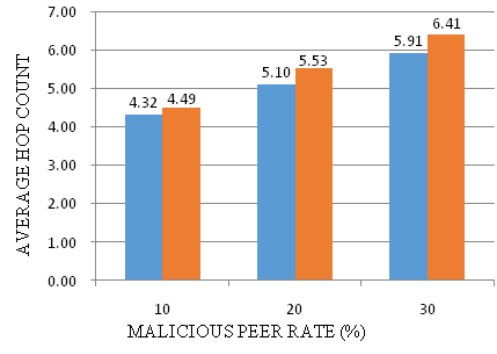
$$x_{im}(t+1) = x_{im}(t) + \alpha \sum_{j \in N_{im}} (x_{ij}(t))(x_{jm}(t) - x_{im}(t)) \quad (12)$$

در واقع نظر همتای i در مورد همسایه j در لحظه t که با $x_{ij}(t)$ نمایش داده شده است، به عنوان ضریب وزن به نظر همتای j اختصاص داده می‌شود. در نتیجه همتایی که امتیاز بهتری داشته باشد، نظرش ارزش بیشتری پیدا خواهد کرد. طبق رابطه (۱۲) در صورتی که همتای i در مورد همسایه j امتیازی نداشته باشد، نباید به نظر او اعتماد کند. در نتیجه ضریب برابر با صفر در نظر گرفته می‌شود.

۶- ارزیابی راهکار مبتنی بر اجماع وزن دار

۶-۱- بررسی نرخ شکست جستجوها

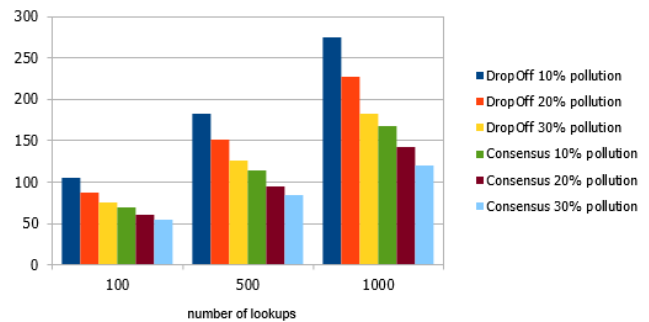
به عنوان مهم‌ترین معیار ارزیابی در این مقاله، نرخ شکست جستجوهای راهکارهای پیشنهادی، در شکل ۸ مورد بررسی قرار گرفته است. این شکل، نرخ شکست جستجوهای دو راهکار مبتنی بر اجماع و اجماع



شکل ۶: مقایسه تعداد گام‌های جستجو در دو الگوریتم ReDS و اجماع

۴-۴- بررسی تعداد پیام‌های منتقل شده

برای بررسی سربار ارتباطی الگوریتم اجماع پیشنهادی، تعداد پیام‌های منتقل شده در شکل ۷ نمایش داده شده است. در این شکل، راهکار مبتنی بر اجماع و سیستم شهرت ارائه شده در ReDS با نام Drop-off، در سه شبیه‌سازی با تعداد جستجوهای ۱۰۰، ۵۰۰ و ۱۰۰۰ مورد مقایسه قرار گرفته است. سه نرخ آلودگی ۱۰، ۲۰ و ۳۰ درصد برای این مقایسه لحاظ شده است. تعداد جستجوهای افزونه و اندازه سبد برای هر دو راهکار مشابه است. همان‌طور که مشاهده می‌شود تعداد پیام‌های منتقل شده در الگوریتم اجماع کمتر است. علت این است که در الگوریتم اجماع، تنها نظر همتاهایی پرسیده می‌شود که در گراف همسایگی وجود دارند. این موضوع علاوه بر محلی‌سازی این الگوریتم، بار ارتباطی آن را در شبکه‌های بزرگ کاهش می‌دهد. همچنین، با توجه به تعداد پیام‌های منتقل شده کمتر، زمان تجمیع آرا و اخذ تصمیم اعتماد کاهش می‌یابد.

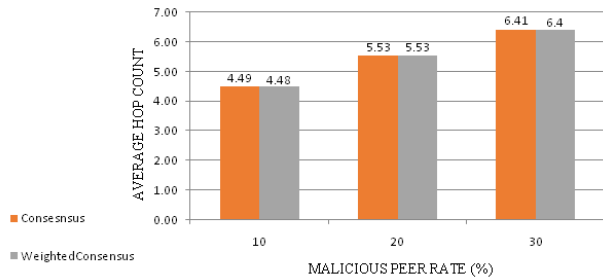


شکل ۷: مقایسه تعداد پیام‌های تبادل شده در دو الگوریتم ReDS و اجماع

همان‌طور که در شکل ۷ نشان داده شده است، با افزایش نرخ آلودگی، تعداد پیام‌های منتقل شده در سیستم کمتر شده است. علت این است که همتاهای آلوده کننده در سیستم شهرت مشارکت نمی‌کنند. در نتیجه هرچه تعداد همتاهای آلوده کننده بیشتر باشد، تعداد پیام‌های رد و بدل شده برای به اشتراک‌گذاری امتیازات و تجمیع آرا کاهش می‌یابد.

۵- محاسبه امتیاز شهرت با الگوریتم اجماع وزن دار

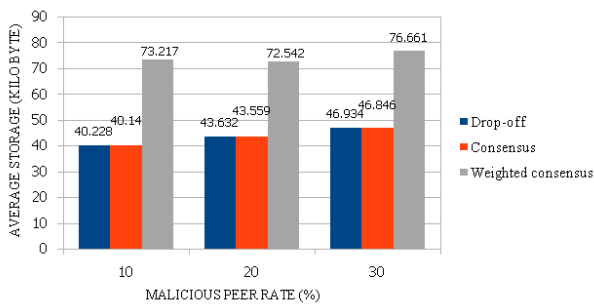
در این بخش با وزن دار کردن الگوریتم اجماع و ایجاد بهبود در سیستم شهرت، نرخ جستجوهای موفق در سیستم همتا به همتا افزایش داده



شکل ۹: مقایسه تعداد گام‌های جستجو در دو الگوریتم اجماع و اجماع وزن‌دار شده

۳-۶- بررسی حافظه مصرفی

در ازای کاهش نرخ شکست جستجوها در راهکار اجماع وزن‌دار شده، با توجه به محاسبه بیشتر امتیازات در مورد ارائه‌دهنده‌ها و نگهداری امتیازات آن‌ها، فضای ذخیره‌سازی بیشتری توسط هر همتا مورد استفاده قرار می‌گیرد. برای بررسی حافظه استفاده‌شده در راهکارهای پیشنهادی، حافظه اشغال‌شده در سه الگوریتم Drop-off، اجماع و اجماع وزن‌دار بررسی شده است. شبیه‌سازی در شبکه‌ای با اندازه ۱۰۰ همتا و با ۱۰۰۰ جستجو انجام شده است. تعداد جستجوهای افزونه برابر با سه و اندازه سبد برابر با دو می‌باشد. نرخ همتای آلوده از صفر تا ۳۰ درصد در حال تغییر است و تمامی شبیه‌سازی‌ها ۳۰ مرتبه تکرار شده است.

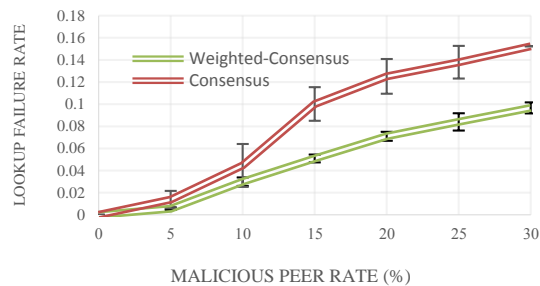


شکل ۱۰: مقایسه حافظه مصرفی در Drop-off و راهکار مبتنی بر اجماع و اجماع وزن‌دار

همان‌طور که در شکل ۱۰ مشاهده می‌شود، حافظه مصرف‌شده در راهکار Drop-off تقریباً برابر با حافظه مصرف‌شده در راهکار اجماع است. علت اینکه حافظه مصرف‌شده در الگوریتم اجماع اندکی کمتر است، کوچک بودن اندازه گراف همسایگی می‌باشد؛ اما در راهکار اجماع وزن‌دار شده به دلیل محاسبه و نگهداری امتیازات همسایگان و ارزیابی امتیازات ارائه‌شده توسط آن‌ها، حافظه بیشتری مصرف می‌شود. حافظه مصرفی در راهکار اجماع وزن‌دار حدود ۱۴ درصد بیشتر از حافظه مصرف‌شده در راهکار اجماع و راهکار Drop-off است.

وزن‌دار شده را مقایسه می‌کند. شبیه‌سازی در شبکه‌ای با اندازه ۱۰۰ همتا و با ۱۰۰۰ جستجو انجام شده است. تعداد جستجوهای افزونه برابر با سه و اندازه سبد برابر با دو می‌باشد. نرخ همتای آلوده از صفر تا ۳۰ درصد در حال تغییر است و تمامی شبیه‌سازی‌ها ۱۰ بار تکرار شده است. اندازه گام برابر با ۰/۱ بوده و متوسط نتایج با بازه اطمینان ۹۵ درصد نشان داده شده است. همان‌طور که مشاهده می‌شود با در نظر گرفتن امتیاز شهرت همتای ارائه‌دهنده نظر و مطابق آن، وزن‌دار کردن الگوریتم اجماع، نرخ شکست جستجوها کاهش یافته است.

همان‌طور که در شکل ۸ مشاهده می‌شود، با در نظر گرفتن امتیاز شهرت همتای ارائه‌دهنده نظر و وزن‌دار کردن الگوریتم اجماع، نرخ شکست جستجوها نسبت به الگوریتم اجماع ساده، به میزان ۴۰ درصد کاهش پیدا کرده است. در الگوریتم اجماع وزن‌دار شده، در حضور ۳۰ درصد همتای آلوده‌کننده، بیش از ۹۰ درصد جستجوها با موفقیت انجام می‌شود.



شکل ۸: مقایسه نرخ شکست جستجو در اجماع و اجماع وزن‌دار

۲-۶- بررسی تعداد گام‌های الگوریتم جستجو

برای بررسی پارامتر تعداد گام‌های الگوریتم جستجو در الگوریتم اجماع وزن‌دار شده، شبکه‌ای با ۱۰۰ همتا و ۱۰۰۰ جستجو شبیه‌سازی شده است. تعداد جستجوهای افزونه برابر با سه، اندازه سبد برابر با دو و اندازه گام ۰/۱ است. نتایج حاصل از مقایسه میانگین تعداد گام‌های الگوریتم جستجو در شکل ۹ نمایش داده شده است. همان‌طور که مشاهده می‌شود میانگین تعداد گام‌های الگوریتم جستجو در الگوریتم اجماع وزن‌دار شده تقریباً تغییری نکرده است. این موضوع قابل پیش‌بینی بود چرا که در منطق الگوریتم اجماع تغییری ایجاد نشده است بلکه تنها نظرات ارائه‌شده توسط همسایه‌ها، ارزش‌یابی شده و مطابق با امتیاز شهرت هر همتا، نظر ارائه‌شده توسط وی، وزن‌دار می‌گردد. به همین ترتیب تعداد پیام‌های رد و بدل شده نیز، تغییری نخواهد کرد. علت عدم تغییر تعداد پیام‌های رد و بدل شده این است که برای محاسبه امتیاز همسایه‌ها، همتا از سایرین نظرسنجی نمی‌کند و تنها به مشاهدات خود اکتفا می‌کند. در نتیجه پیام بیشتری نسبت به راهکار مبتنی بر اجماع رد و بدل نخواهد شد.

۷- نتیجه گیری

مراجع

- [1] J. Kurose and K. Ross, *Computer networking: a top-down approach*, 4th edition, Addison-Wesley, 2007.
- [2] E. Koutrouli and A. Tsalgaidou, "Taxonomy of attacks and defense mechanisms in P2P reputation systems—Lessons for reputation system designers," *Computer Science Review*, vol. 6, no. 2, pp. 47-70, May 2012.
- [3] I. Stoica, R. Morris, D. Karger, M. Kaashoek and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *ACM SIGCOMM Computer Communication conference*, vol. 31, no. 4, pp. 149-160, 2001.
- [4] A. Kapadia and N. Triandopoulos, "HALo: high-assurance locate for distributed hash tables," in *Proceedings of the 15th Ann. Network and Distributed System Security Symp. (NDSS)*, Vol. 8, pp. 61-79, 2008.
- [5] A. Nambiar and M. Wright, "Salsa: a structured approach to large-scale anonymity," in *Proceedings of the 13th ACM conference on Computer and Communications Security*, pp. 17-26, 2006.
- [6] R. Akavipat, M. Al-Ameen, A. Kapadia, Z. Rahman, R. Schlegel and M. Wright, "ReDS: a framework for reputation-enhanced DHTs," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 2, pp. 321-331, 2014.
- [7] R. Olfati-Saber, J. Fax and R. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215 - 233, 2007.
- [8] (2016, Jan.) Google Code, [Online], Available at: <https://code.google.com/archive/p/joonion-jchord/>

در این مقاله، مسئله آلودگی در فرآیند مکان‌یابی منابع در سیستم‌های هم‌تا به هم‌تا مورد بررسی قرار گرفته و راهکاری مبتنی بر شهرت با استفاده از الگوریتم اجماع برای مقابله با آن ارائه شده است. در این آلودگی در مسیر رسیدن به مقصد مورد نظر، در صورتی که جستجو به هم‌تای آلوده‌کننده برسد، به‌جای طی فرآیند نرمال و پیدا کردن هم‌تای بعدی، اولین هم‌تای آلوده‌کننده به‌عنوان گام بعدی جستجو معرفی شده و در نتیجه جستجو از مسیر صحیح خود منحرف می‌شود. تمرکز این مقاله بر سیستم‌های هم‌تا به هم‌تای مبتنی بر جدول‌های درهم‌سازی توزیع‌شده، مانند Chord است. این مقاله، راهکاری برای تجمیع نظرات بر اساس قاعده اجماع ارائه کرده است که با کمک آن و با جمع‌آوری امتیاز شهرت محاسبه‌شده توسط همسایه‌ها، هم‌تای محاسبه‌کننده امتیاز می‌تواند تصمیم با قابلیت اطمینان بالاتر اتخاذ نماید. راهکار ارائه‌شده در این مقاله بار محاسباتی قابل چشم‌پوشی دارد و مقیاس‌پذیر است. در ادامه برای بهبود الگوریتم ارائه‌شده، الگوریتم اجماع وزن‌دار معرفی شده است. در این الگوریتم نظر ارائه‌شده توسط همسایه‌ها، با توجه به امتیاز شهرت آن‌ها اهمیت پیدا می‌کند. با شبیه‌سازی سیستم شهرت ارائه‌شده بر مبنای الگوریتم اجماع وزن‌دار شده، مشاهده می‌شود که نرخ موفقیت جستجوها افزایش یافته است.

زیرنویس‌ها

⁶Consensus⁷Finger Table⁸High Assurance Locate⁹Reputation for Directory Services¹Peer-to-peer²Scalability³Distributed Hash Table⁴Malicious Peers⁵Reputation System