تاریخ دریافت مقاله: ۹۸/۱۰/۳۰
تاریخ پذیرش مقاله: ۹۹/۰۴/۲۰

مروری بر روش‌های برخط ذخیره داده در شبکه‌های مبتنی برحافظه نهان

سید مصطفی سید رضازاد دلالی*
مرکز تحقیقات دانش‌های بنیادی، پژوهشکده علوم کامپیوتر، تهران
پست الکترونیکی: mostafa@ipm.ir

پگاه فرهادی
دانشگاه صنعتی شریف- پردیس بین الملل کیش، جزیره کیش
پست الکترونیکی: Pegah.farhadi@kish.sharif.edu

چکیده:

استفاده از حافظه‌نهان جهت افزایش کارآیی شبکه اینترنت همواره از دو جهت مورد توجه و ارزیابی بوده است: (۱) افزایش بهره‌وری کانال‌های ارتباطی و کاهش بار کارسازها، و (۲) از جهت کوتاه شدن فاصله دسترسی به داده‌ها برای کاربران. با پیشنهاد و روی کار آمدن شبکه‌های محتوا محور، امکان ذخیره داده‌ها در داخل شبکه، به‌وسیلهٔ تجهیزات شبکه از قبیل مسیریاب ها به‌وجود آمده است. با این حال، مسئله پیش‌رو، ارزیابی کارایی چنین حافظه‌هایی می‌باشد. با توجه به این‌که اندازه حافظه مورد استفاده در یک مسیریاب به نسبت حجم ترافیک عبوری از آن محدود بوده و نیز ترافیک عبوری از یک مسیریاب به مسیریاب‌های دیگر وابسته است، این امکان وجود دارد که واکاوی این حافظه‌ها نه تنها بازخورد مثبتی نداشته باشد بلکه باعث افزایش تاخیر عبور از یک مسیریاب بشود. در این مقاله به بررسی و مقایسه چند روش کلی برای ذخیره داده‌ها در مسیریاب‌ها می‌پردازیم.

واژه‌های کلیدی: شبکه مبتنی بر محتوا، حافظه نهان داخل شبکه، مقایسه کارآیی، الگوریتم‌های پذیرش داده در حافظه‌نهان، Named Data Networking

* نویسندهٔ مسئول

۱- مقدمه

به‌کارگیری حافظهٔ‌نهان یکی از تکنیک‌های بسیار مهم جهت افزایش کارآیی انواع سیستم‌های رایانه‌ای می‌باشد[۱]. استفاده از این نوع حافظه‌ها در شبکه‌های کامپیوتری نیز بسیار مرسوم است. از کاربردهای حافظهٔ‌نهان در شبکه‌های رایانه‌ای می‌توان به نزدیک کردن داده‌ها (داده‌های محبوب) به کاربران و همچنین کاهش بار کارسازها اشاره کرد. تاکنون این تکنیک در سطح لایه کاربردی[1] مورد استفاده قرار گرفته است که باعث می‌شود حافظهٔ‌نهان تنها به یک کاربرد خاص سرویس دهد. همان‌طور که می‌دانیم در شبکه‌های TCP/IP دسترسی به محتوا براساس نشانی مکان آن محتوا صورت می‌گیرد. اما در شبکه‌های محتوامحور برخلاف شبکه‌های TCP/IP، این نشانی‌دهی براساس نام محتوا انجام می‌شود نه براساس محل آن. بنابراین شبکه‌های محتوا محور با اختصاص نام (نشانی) به محتوا این امکان را فراهم می‌نمایند که داده‌ها بتوانند در داخل شبکه بازیابی شوند. تاکنون معماری‌های متنوعی برای شبکه‌های محتوامحور پیشنهاد شده است که از جمله مهم‌ترین آن‌ها، NDN [2]

1-Application Layer
2-Named Data Networking

[۲,۳] می‌باشد. در این معماری شبکه، با دادن نام به داده‌ها می‌توان آن‌ها را به‌طور مستقیم از شبکه درخواست نمود بدون آن‌که نیاز به یافتن مکان استقرار داده باشد. این داده‌ها هنگامی که از شبکه عبور می‌کنند می‌توانند به‌وسیله دستگاه‌های داخل شبکه ذخیره شوند تا مورد استفاده درخواست‌های دیگر قرار گیرند. بنابراین هنگامی که دادهٔ (محتوای) درخواستی در یکی از حافظه‌های موجود در شبکه یافت شود، مطمئنا با تاخیر کمتری (نسبت به دریافت داده از کارساز) به کاربر خواهد رسید. ولی چنانچه داده در حافظه داخل شبکه یافت نشود مقدار تاخیر افزوده شده به هر بسته قابل توجه خواهد بود. از این‌رو، ارزیابی کارایی حافظه‌های نهان داخل شبکه از اهمیت زیادی برخوردار است.

از سوی دیگر، با توجه به محدود بودن اندازه حافظه مورد استفاده در یک مسیریاب نسبت به حجم ترافیک عبوری از آن، و وابسته بودن ترافیک عبوری از هر مسیریاب به سایر مسیریاب‌های دیگر، این امکان وجود دارد که واکاوی این حافظه‌ها نه تنها بازخورد مثبتی نداشته باشد بلکه باعث افزایش تاخیر عبور از یک مسیریاب بشود. برای کاهش این تاخیر، تاکنون الگوریتم‌های پیچیده گوناگونی برای جایگذاری داده در حافظهٔنهان موجود درشبکه پیشنهاد شده است که مشخص می‌نمایند که چه داده‌ای در چه مکانی ذخیره شود [۲,٤,٥,٦,۷,۸,۹,۱۰]. این الگوریتم‌ها به‌دسته‌های مختلفی تقسیم می‌شوند مانند طبقه‌بندی براساس الگوریتم مسیریابی، براساس نوع تصمیم‌گیری و غیره. از لحاظ الگوریتم مسیریابی، این الگوریتم‌ها به دو دسته کلی جایگذاری برخط و جایگذاری غیربرخط تقسیم می‌شوند. در جایگذاری برخط، داده‌ها فقط می‌توانند در مسیر بین گره تولیدکننده داده تا درخواست کننده ذخیره شوند. درحالی‌که در الگوریتم‌های غیربرخط، داده‌ها می‌توانند در هر گرهی در شبکه ذخیره شوند و مسیریاب باید تصمیم بگیرد که بستهٔ مورد درخواست[۳] را به سمتی ارسال کند که الگوریتم مسیریابی مشخص می‌نماید و یا به سمت گرهی ارسال کند که قبلا ممکن است آن را در خود ذخیره کرده باشد. به دلایل زیر ما به بررسی الگوریتم‌های غیر برخط نمی‌پردازیم:

۱- افزایش بارترافیکی به علت تبادل اطلاعات شبکه‌ای بین گره‌ها جهت یافتن گرهی که داده موردنظر را ذخیره کرده است.

۲- احتمال ارسال به گرهی که داده در آن دیگر وجود ندارد.

۳- عدم تطابق با الگوریتم‌های مسیریابی که هم‌اکنون بر روی شبکه‌اینترنت مورد استفاده قرار می‌گیرند.

٤- افزایش مقدار پردازشی که باید به‌وسیله هر گره انجام شود.

دسته‌بندی دیگری که می‌توان برای الگوریتم‌های ذخیره داده درنظر گرفت، دسته‌بندی بر اساس نوع تصمیم‌گیری در انتخاب داده جهت ذخیره در یک گره می‌باشد. این نوع تصمیم‌گیری می‌تواند بر اساس محبوبیت داده، استفاده از یک تابع تصادفی و یا هر روش پیشنهادی دیگر باشد. در این مقاله به مقایسه چند روش کلی در ذخیره داده‌ها در مسیریاب‌ها می پردازیم.

۲- تحقیقات پیشین مرتبط

نسخه‌ای از شبکه محتوامحور[٤] که در اینجا مورد بحث قرار می‌دهیم، ابتدا در [۳] به‌عنوان نمونه‌ای از شبکه‌های اطلاعات محور[٥] ارائه شده است. نسخه بعدی CCN در [۲] به نام شبکه NDN ارائه شده است که هم اکنون یک پروژه فعال می‌باشد. در کاربردهایی مانند خانه‌های هوشمند [۱۱]، شبکه‌های Ad-hoc [۲۱]، برنامه‌های کاربردی وسیله نقلیه [۱۳]، شبکه‌های حسگر بی‌سیم [۱٤]، اینترنت اشیاء [۱٥] و بسیاری از برنامه‌های کاربردی دیگر رو به رو رشد است. با این حال، اصلی‌ترین مزیت NDN در سازگاری آن با زیرساخت‌های فعلی است. یکی از مزایای NDN نسبت به رقبای آن، قابلیت ذخیره محتوا است. قابلیت ذخیره ترافیک

3-Request Packet

4- CCN
5-ICN

اینترنت توسط Ager و همکاران در [۱۶] تأیید شده است. با این حال، کارآیی ذخیره‌سازی بستگی به محبوبیت محتوا، سیاست ذخیره‌سازی و دیگر پارامترهای شبکه دارد [۱۷]. سیاست نادرست ذخیره‌سازی حتی می‌تواند به کارآیی آسیب برساند. قدسی و همکاران، معتقد بودند که حافظه نهان داخلی شبکهٔ ICN بی‌نتیجه است زیرا بیشترین بازدیدها فقط در گره‌های لبه شبکه اتفاق می‌افتد، جایی که فایل‌های محبوب در آن ذخیره شده‌اند[۳]. همچنین محاسبات Perino و Varvello نشان می‌دهد که فناوری حافظه (به‌عنوان مثال اندازه و سرعت) نمی‌تواند با هزینه‌ای مناسب سرعت خط را برای NDN مطابقت دهد [۱۸]. در مقاله تازه منتشر شده نشان داده شده است که افزایش حجم حافظه باعث افزایش کارآیی حافظه‌نهان (به‌دلیل ذخیره محتواهای محبوب‌تر) و نیز افزایش هزینه استقرار می‌شود. براساس این مشاهدات، یک طرح ذخیره‌سازی برای افزایش کارآیی حافظه‌های‌نهان کوچک [۱۹] برای کاهش هزینه‌های استقرار معرفی شده است. با این وجود، راه‌حل ریاضی آن‌ها برای یافتن مدل بهینه، در عمل نمی‌تواند مورد استفاده قرار گیرد. حافظه نهان داخلی شبکه شبیه به حافظه نهان مشارکتی وب است. مقالات متعددی پروتکل‌هایی را برای چنین ممکاری‌هایی ارائه داده‌اند [۲۰,۲۱,۲۲]. اما تحلیل Wol-man و همکارانش نشان می‌دهد که آن‌ها برای جمعیت‌های بزرگ کاربرد زیادی ندارند [۲۳]. با این حال، این تجزیه و تحلیل، از داده‌های ترافیک مربوط به حدود ۱۹ سال پیش است که در آن هنگام وب بسیار متفاوت از امروز بوده است. به‌عنوان مثال، در مورد تلویزیون‌های مبتنی بر تقاضا، Li و Simon نشان داده‌اند که راهبرد حافظه نهان مشارکتی آن‌ها می‌تواند ترافیک متقابل را به میزان قابل توجهی کاهش دهد[۲۴]. به‌طور کلی، اگرچه این مشکل با تکنیک‌های بهینه‌سازی برای کاربردهای مختلف از جمله شبکه‌های تحویل محتوا[6]، IPTV، و ذخیره‌سازی وب، مورد استفاده گسترده قرار گرفته است، اما این‌ها مبتنی بر کاربردها و توپولوژی‌های خاص (به‌عنوان مثال، ساختارهای سلسله مراتبی) است. محدودیت برای NDN بسیار زیاد است زیرا حافظه‌های‌نهان در زیرساخت‌های شبکه قرار می‌گیرند (لایه –۳). بنابراین، مشکل قراردادن حافظه نهان در شبکه اطلاعات کاملا متفاوت است با برنامه قبلی مبتنی بر ذخیره‌سازی [۲۵]. در [۲۵] یک جایگذاری حافظه نهان مطلوب ارائه شده است. آن‌ها برای حل معادله خود از روش دو مرحله‌ای استفاده کردند که مبتنی بر ذخیره‌سازی در هر گره است. ذخیره‌سازی در گره‌های دورتر باعث کاهش ذخیره محتوای کمتر محبوب و همچنین کاهش مزیت آن می‌شود. از آنجایی که جایگذاری بهینه در حافظه نهان، عملی نیست، آن‌ها یک روش ابتکاری نیمه بهینه ارائه دادند. البته، رویکرد اکتشافی آن‌ها نیاز به محاسبه نمودار زیادی دارد. بسیاری از مطالعات، از CCN برای حل برخی از جنبه‌های مشکل‌ساز (مانند اتصال) شبکه‌های تلفن همراه استفاده می‌کنند [۲۷,۲۶,۱۳]. به‌طور کلی، اکثر راه‌حل‌هایی که حافظه نهان در آن‌ها برخط نیست، از نوعی هماهنگی استفاده می‌کنند. حافظه‌های نهان موجود در لبهٔ شبکه، بدون همکاری و مشارکت، بیشترین بازدید را به‌دست می‌آورند و به‌عنوان فیلتر عمل می‌کنند. این اثر برای سیستم‌های سلسله مراتبی مشهود بوده، و همچنین برای حافظهٔ‌نهان وب [۱] شناخته شده می‌باشد، و برای CCN نیز مشاهده شده است[۲۸,۲۹]. حافظه‌های نهان لبه حاوی فایل‌های محبوب هستند، در حالی که آن‌هایی که در هسته هستند با محتوای غیر محبوب انباشته می‌شوند که تعداد کمی بازدید می‌گیرند. به همین دلیل است که فیاض‌بخش و همکارانش باور داشتند که بیشترین مزیت حافظهٔ‌نهان درون شبکه‌ای برای ICN را می‌توان با، TCP/IP به‌وسیله قرار دادن کل حافظهٔ‌نهان در لبه به‌دست آورد [۳۰]. از طرف دیگر، مشاهدات Tyson و همکارانش با BitTorrent نشان می‌دهد که حافظهٔ‌نهان قابل توجهی در سیستم‌های خودمختار Tier-2 .(AS) وجود دارد[۳۱]. Aubry و همکارانش، [۳۲] چنین استدلال کردند که در همه گره‌ها نیازی به حافظهٔ‌نهان نیست و می‌توان با اجازه

6-Content Delivery Networks

دادن بـــه نیمی از مسیریاب‌ها بـــرای ذخیره‌سازی محتوا، همـــان سطح کارآیی را به‌دست آورد. اگرچه اعتقاد بر این است که باید محتوای محبوب در لبه شبکه و محتوای کمتر محبوب در مسیریاب‌های مرکزی ذخیره شود، اما معرفی یک روش تخصیص بهینه حافظهٔنهان در [۲۵] این باور را برطرف می‌کند. Wang و همکارانش مسئله را به‌عنوان یک برنامه خطی برای به حداکثر رساندن مزیت ذخیره‌سازی، کـــه معادل کمینه‌سازی فاصله است فرموله کردند. یکی از دلایـــل راهبرد ذخیره‌سازی درمسیر، که پیش فرض NDN نیز می‌باشد این است که هسته شامل نسخه‌هایی از مطالب در لبـــه است. یکی از راه‌های کاهش این افزونگی این است که حافظه‌های‌نهان پروتکل هماهنگی را اجرا کنند. هماهنگی ممکن است نیـــاز به اندازه‌گیری محبوبیـــت محتوا داشته باشـــد[٤,٦,٩]. چنین طرح‌هایی باعـــث افزایش سربارهای ترافیکی، و اضافه کردن پیچیدگی به مسیریاب‌ها می‌شوند و ممکن است به سرعت کافی با تغییرات محبوبیت سازگار نشوند. اندازه‌گیری محبوبیت عمومی در [۸] ارائه شده است. اگرچه آن‌ها ایده خوبی را برای ذخیره کردن محبوب‌ترین محتوا در مسیریاب مرکـــزی دارند که می‌توانند بازخورد (hit) بیشتـــری از درخواســـت کنندگـــان بگیـــرد، در همه مسیریاب‌هـــا از توزیع Zipf برای ارزیابی محبوبیت محتوا استفـــاده می‌کنند که لزوما یـــک رویکرد درست نمی‌باشد. در یک رویکرد مبتنی بـــر محبوبیت برای کاهش افزونگی کـــه از حافظهٔنهان وب الهام گرفتـــه شده است، یک برنامه «ذخیـــره تعاونی مبتنی بر سن» در [۷] ارائه شده است. در طراحی آن‌ها، محتوای محبوب که در مسیریاب‌های نزدیک به درخواســـت کنندگان ذخیـــره می شـــود، مقادیر سنی بیشتری را دریافت می‌کننـــد. در این روش، آن‌ها محتوای محبـــوب را بـــه مشتری نزدیک‌تـــر می‌کننـــد. اندازه‌گیری محبوبیـــت مانند سایـــر پیشنهادهای ذخیره‌ســـازی، کار ســـاده‌ای نیست. عـــلاوه بر این، ذخیـــره محتوای محبوب به‌طور کامل در مسیریاب‌های لبه، همیشه بهترین انتخاب نیست.

۳ - معماری شبکه NDN

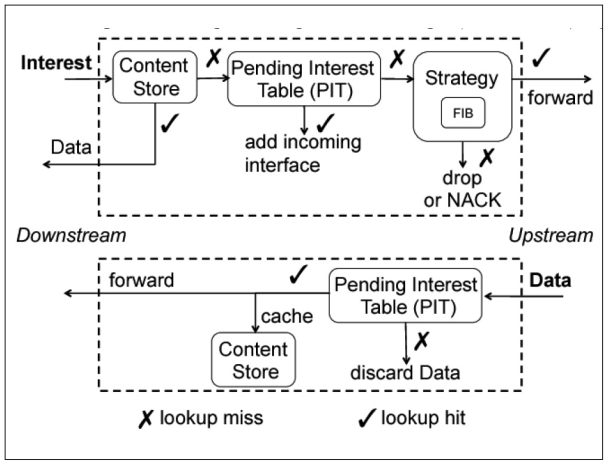
قبل از مقایسه الگوریتم‌های ذخیره داده، ابتدا به معرفی معماری [۳] می‌پردازیم. مشخصه اصلی این معماری، اختصاص نشانی به داده‌ها به‌جـــای اختصاص نشانی به مکان داده است. در این معماری، دستگاه‌های مسیریاب براساس نام داده، اقدام به یافتن بهترین مسیر برای هر درخواست می‌نمایند. یکی از نقاط قوت این معماری تطابق کامل الگوریتم مسیریابی آن با الگوریتم‌های مسیریابی فعلی می‌باشد که باعث شده این شبکه بتواند به تدریج به شبکه IP اضافه شود. در این معماری، بسته‌های شبکه به دو دسته: بسته درخواست و بسته داده تقسیم می‌شوند. کاربر جهت دریافت داده و یا فایل، نام آن داده یا فایل را به اولین دستگاه شبکه‌ای که به آن متصل است ارسال می‌کند و شبکه مسئول یافتن و فراهم کردن داده یا فایل است. این داده‌ها در قالب بسته‌های داده به کاربر برگشت داده می‌شوند. کوچک‌ترین واحد داده که می‌تواند بر روی خط ارسال شود اندازه بسته داده را مشخص می‌کند. بســـته‌های داده هریک به‌طـــور مستقل یک نام منحصر به فرد دریافت می‌نمایند. جهت جلوگیری از افزایش اندازه جداول مسیریابی، نام‌ها به‌صـــورت سلسله مراتبی انتخاب می‌شوند. هر مسیریاب مجهز به سه جدول اصلی برای انتقال داده از ورودی به خروجی خود می‌باشد. این جداول عبارتند از:

۱- CS(Content Store)

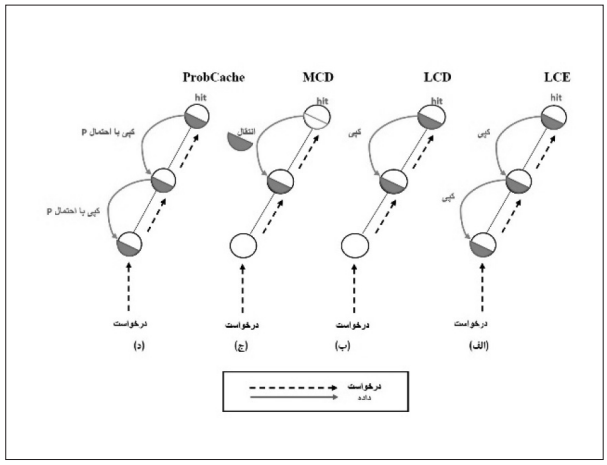
۲- Pending Interest Table(PIT)

۳- Forwarding Information Base (FIB)

جدول CS همان حافظه نهان داخل یک مسیریاب می‌باشد که چنانچه داده قبلا در این جدول ذخیره شده باشد، می‌تواند آن را به کاربر برگرداند بدون آن‌کـــه نیاز باشد که بستـــهٔ درخواست، بقیه مسیر تا کارساز را طی نماید. جدول PIT جهت ذخیره‌سازی مسیر برگشت برای بسته‌های داده تعبیه شده است. همان‌طـــور که ذکر شد در شبکه NDN گره‌ها دارای نشانی نیستند. بنابراین باید تمهیدات لازم جهت



شکل ۱: معماری یک مسیریاب NDN



شکل۲: مقایسه الگوریتم‌هایProbCace MCD ،.LCD،LCE

برگشت بسته داده در هر مسیریاب وجود داشته باشد که همان جدول PIT می‌باشد. علاوه بر آن، جدول PIT از ارسال مجدد یک درخواست با یک نام تکراری جلوگیری می‌کند. جدول PIT با ذخیره کردن تمام درگاه‌هایی که مسیریاب از آن‌ها درخواستی برای یک نام خاص دریافت نموده، می‌تواند داده ورودی برای آن درخواست را به‌طور همزمان به تمام آن درگاه ها ارسال نماید. آخرین جدول، جدول FIB می‌باشد که درآن اطلاعات مسیریابی مربوط به ارسال درخواست‌ها به خارج از مسیریاب جهت رسیدن به کارساز، ذخیره شده است. شکل ۱ به اختصار فرآیند ارسال بسته‌های درخواست و داده در یک مسیریاب را نشان می‌دهد.

علاوه بر این جداول، دو الگوریتم دیگر نیز جهت ذخیره داده در حافظه نهان یک مسیریاب، لحاظ شده است که عبارتند از ۱- الگوریتم جایگزینی داده در حافظه و ۲- الگوریتم تصمیم‌گیری جهت ذخیره داده[۷]. با استفاده از الگوریتم جایگزینی، مسیریاب تصمیم می‌گیرد که کدام داده موجود در حافظه باید از آن خارج شده و داده جدید جایگزین آن شود. متداول‌ترین الگوریتم جایگزینی، الگوریتم جایگـــذاری با داده[۸] اخیرا کمترین دسترسی یافته[۹] می‌باشد که در این پژوهش ما نیز از آن بهره برده‌ایم. الگوریتم دوم، الگوریتم تصمیم‌گیری است که مشخص می‌کند کدام داده در کدام مسیریاب و یا حافظه نهان باید ذخیره گردد.

7- Admission Policy
8-Replacement Policy
9-(LRU) Least Recently Used

سهم اصلی این مقاله بررسی و مقایسه چند روش کلی از الگوریتم‌های تصمیم‌گیری جهت ذخیره داده در حافظه نهان شبکه‌های مبتنی بر داده می‌باشد.

۴- الگوریتم‌های تصمیم گیری داده

در این بخش ابتدا به معرفی الگوریتم‌های تصمیم‌گیری جهت مشخص کردن مکان ذخیره داده‌ها در مسیریاب‌ها می‌پردازیم. این الگوریتم‌ها نماینده دسته‌ای از الگوریتم‌ها می‌باشند که رفتار مشابه با آن دارند. البته در انتخاب این الگوریتم‌ها دقت شده تا بهترین نماینده از هر دسته مورد انتخاب قرار گیرد. به‌طـــور کلی، در الگوریتم‌های تصمیم‌گیری سعی بر این است که داده‌های محبوب به کاربران نزدیک‌تر شوند. البته اندازه‌گیری محبوبیت بدون سربار نبوده و هزینه زیادی دارد. علاوه بر آن، ممکن است که الگوریتم اندازه‌گیری محبوبیت، یک داده را به‌عنوان داده محبوب تلقی کند درحالی‌که محبوبیت آن رو به افول است. بنابراین در این مقاله، از روش‌هایی که به‌طور مستقیم اقدام به اندازه‌گیری محبوبیت داده می‌نمایند استفاده نشده است.

٤-١ الگوریتم Leave Copy Every Where (LCE)

LCE ابتدایی‌تریـــن الگوریتم تصمیم گیری ذخیره داده بـــرای شبکـــه NDN می‌باشد که در مقاله اولیه NDN [۳] ازآن استفاده شده است. در این روش سادهٔ جایگذاری، تمام داده‌ها در تمام گره‌های عبوری ذخیره می‌شوند.

شکل ۲(الف) فرآیند LCE را نشان می‌دهد. تنها مزیت این الگوریتم در سادگی پیاده‌سازی آن بوده و فقط در مقالاتی که ذخیره‌سازی داده در حافظه نهان از اهمیت برخوردار نبوده استفاده شده است. اما از آنجا که این الگوریتم پیش‌فرض NDN می‌باشد، در این مقاله مورد ارزیابی و مقایسه قرار گرفته است.

٤-٢ الگوریتم Leave Copy Down (LCD)

در این الگوریتم [۳۳]، همان‌طـــور که در شکل ۲(ب) نشان داده شده است داده‌های یافت شده (چه از کارساز و چه از حافظه نهان میانی) در طول مسیرِ برگشت به سمت درخواست کننده، تنها در مسیریاب همسایـــهٔ یک مرحله پایین‌تر ذخیره می‌شوند. اگر داده‌ای از محبوبیت بیشتری برخوردار باشد، مجددا برای آن درخواست خواهد آمد و بدین ترتیب با هر درخواست، داده به کاربران نزدیک‌تر خواهد شد. بنابراین با استفاده از این الگوریتم، گره‌های لبه که نزدیک‌تـــرین گره‌ها به کاربران هستند، تنها با داده‌های بسیار محبوب انباشته می‌شوند. این الگوریتم یکی از موثرترین الگوریتم‌ها برای جداسازی داده‌های محبوب از داده‌های غیرمحبوب بدون استفاده از امکانات اضافه (مانند جداول فهرست) می‌باشد.

٤-٣ الگوریتم Move Copy Down (MCD)

عملکرد این الگوریتم مشابه الگوریتم LCD می‌باشد با این تفاوت که داده به‌طـــور کامل از یک گره به گره دیگر منتقل می‌شود و از روی حافظه گره حاضر برداشته می‌شود [۳٤]. شکل ۲(ج) روند الگوریتم MCD را نشان می‌دهد. با این روش، سعی شده تا تعداد نسخـــه‌های موجود از هر داده در شبکه به کمترین مقدار برسد. البته این احتمال وجود خواهد داشت که داده حذف شده از یک گره، می‌توانست جوابگوی درخواست‌های صادره از یک مسیریاب دیگر باشد.

٤-٤ الگوریتم ProbCache

این الگوریتم نماینده دسته‌ای از الگوریتم‌هایی است که از یک تابع احتمال برای تصمیم‌گیری جهت ذخیره داده در یک گره استفاده می‌کنند. این الگوریتم در مقایسه‌ای که در مقاله [۳۵] آمده، یکی از موثرترین الگوریتم‌های مبتنی بر احتمالات می‌باشد. در این الگوریتم، احتمال ذخیره داده بر اساس دو پارامتر: (۱) فاصله گره تا درخواست کننده و (۲) مقدار حافظه باقی مانده از گره کنونی تا کاربر محاسبه می‌شود. هرچه داده به کاربر نزدیک‌تر شود احتمال افزایش می یابد، و در عین حال، برای نزدیک شدن به کاربر باید از گره‌های بیشتری عبور کرد و جمع حافظه‌های موجود تا درخواست کننده کم می‌شود، که این امر باعث کاهش احتمال می‌گردد. بنابراین بنا به شرایط شبکه، این دو پارامتر سعی در ایجاد تعادل در حافظه‌های نهان موجود در طول مسیر بین درخواست کننده تا کارســـاز می‌نمایند. شکل ۲(د) فرآیند Prob-Cache را نشان می‌دهد.

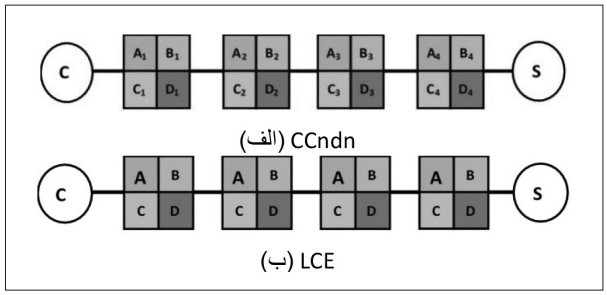
٤-٥ الگوریتم CCndn

الگوریتم CCndn و الگوریتم مشابه آن CCndnS [۳٦] سعی در توزیع داده‌ها در طول مسیر بین درخواست کننده و کارســـاز می‌نمایند. در این دسته از الگوریتم‌ها، فایل‌ها در مبدا (کارســـاز) به چندین قسمت تقسیم می‌شوند که به آن‌هـــا تکّه می‌گوییم. در طول مسیر در هر مسیریاب یک یا چند تکّـــه ذخیره می‌شود، به گونه‌ای که تکّه‌های ابتدایی در مسیریاب لبه (نزدیک به کاربر) و تکّه‌های انتهایی در انتهای مسیر (نزدیک به کارســـاز) ذخیره خواهند شد. از مزایای استفاده از این روش می‌توان به موارد زیر اشاره کرد:

۱- نوع داده‌ها در یک مسیریاب افزایش می‌یابد. بدین ترتیب در صورتی‌کـــه داده از محبوبیت زیادی برخوردار نباشد تاثیر منفی آن بسیار کم خواهد شد. درضمن با افزایش تعداد داده‌ها، شانس داشتن داده‌های محبوب بیشتر می‌شود.

۲- مسیریاب‌های لبه که از اهمیت زیادی برخوردار هستند، می‌توانند تکّـــه‌های ابتدایی فایل‌های بیشتری را در خود نگهداری کنند.

۳- خاصیت پالایـــش گره‌های لبه با این روش کاهش



شکل ۳: مقایسه چیدمان‌ها در (الف) CCndn با (ب) LCE

می‌یابد. گره‌های لبه معمولا تمام درخواست‌ها برای‌فایل‌های محبوب را برآورده می‌کنند و باعث می‌شوند که درخواست مناسبی به گره‌های بعدی در داخل شبکه ارسال نشود. با تقسیم کردن یک فایل محبوب بین گره‌های مختلف، هرکدام از آن‌ها می‌توانند مقداری از درخواست‌های فایل‌های محبوب را اجابت کنند و بدین‌ترتیب کارآیی آن‌ها افزایش می‌یابد.

٤- از آنجا که اغلب داده‌های حجیم و محبوب، از نوع داده‌های ویدئو می‌باشند، این روش، از ایده streaming پیـــروی می‌کند و تأخیر دریافت داده‌های مختلف یک فایل می‌توانند با همدیگر همپوشانی داشته باشند. در حالـــی که بسته‌های اولیه به سرعت از گره لبه دریافت می‌شوند، زمان دسترسی به داده‌های بعدی می‌تواند با زمان دسترسی و استفاده این داده‌ها همپوشانی شوند.

شکل ۳ به مقایسه الگوریتم CCndn و الگوریتم LCE می‌پردازد. درحالی‌کـــه با LCE فقط یک فایل در تمام گره‌ها ذخیره می‌شود، در CC-ndn چهار فایل می‌تواند در ٤ گره ذخیره شود.

۵- محیط شبیه‌سازی

جهت بررسی و مقایسه الگوریتم‌های تصمیم‌گیری برای جایگذاری داده نیاز به یک محیط شبیه‌ساز می‌باشد که ما این محیط را با استفاده از زبان برنامه‌سازی جاوا پیاده‌سازی نموده‌ایم. این شبیه‌ساز، شبکه با معماری NDN را کاملا پیاده‌سازی نموده و می‌تواند انواع پارامترهای شبکه را جهت مقایسه اندازه‌گیری نماید. در جدول ۱ تمام پارامترهای استفاده شده در محیط شبیه‌ساز آورده شده است.

جدول ۱: پارامترهای استفاده شده در شبیه ساز

پارامتر	مقدار
توپولوژی	Geant
الگوریتم جایگزینی	LRU
توزیع محبوبیت α	۱
تعداد کاربران	۳۰۰۰
تعداد کارسازها	۴۰۰۰
طول فایل	۶۰۰ بسته
طول تکّه	۱۰۰ بسته
تعداد کل داده‌ها	۲۴۰۰۰۰۰ بسته
اندازه حافظه نهان	۱۰۰ تا ۳۰۰ بسته

شبکه استفاده شده در این پژوهش شبکه Geant[10] می‌باشد که شکل آن در شکل ٤ نمایش داده شده است. این شبکه یک شبکه واقعی در اروپا است که شامل ٤٠ مسیریاب است. تعدادی از این مسیریاب‌ها به کاربر و تعدادی به کارساز متصل شده‌اند که این مسیریاب‌ها در شکل مشخص شده‌اند. ما به دو صورت تصادفی و ثابت، مسیریاب‌ها را به کاربر و کارساز متصل کرده و شبیه ساز را اجرا کردیم.

در شکل ٤ کاربران و کارسازها به‌صورت ثابت درنظر گرفته شده‌اند. مسیریاب‌هایی که به کاربر متصل هستند با رنگ طوسی و مسیریاب‌هایی که به کارساز متصل هستند با دو دایره مشخص شده‌اند. پارامترهای گزارش شده در این مقاله عبارتند از:

۱- احتمال یافتن داده از یکی از گره‌های شبکه[11]

۲- متوسط تعداد کپی داده‌ها در شبکه[12]

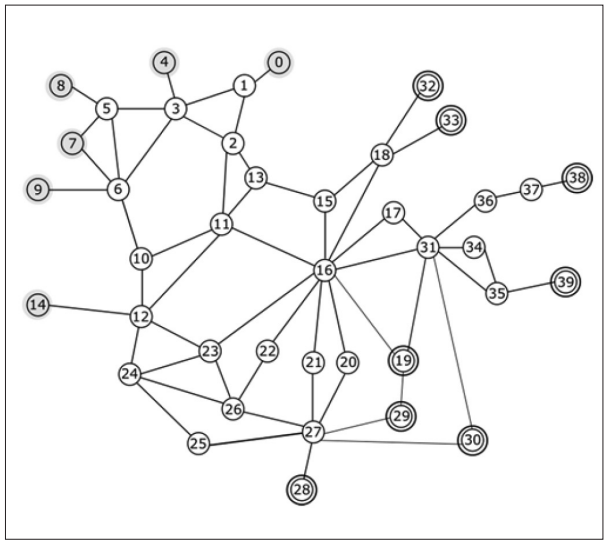
۳- متوسط فاصله کاربران تا داده‌های مورد درخواست آن‌ها[13]

٤- متوسط فاصله کاربران تا اولین بسته داده‌های درخواستی که در حافظه نهان قرار دارد[14]

۶ - مقایسه الگوریتم‌ها

داده‌های حاصل از اجرای برنامه شبیه ساز با به‌کارگیری

10- https://www.geant.org/
11-Probability of Network Hit
12-Average Data Redundancy
13-Average Distance to Content
14-Average Distance to the First Packet



شکل ۴: توپولوژی Geant با ۴۰ مسیریاب

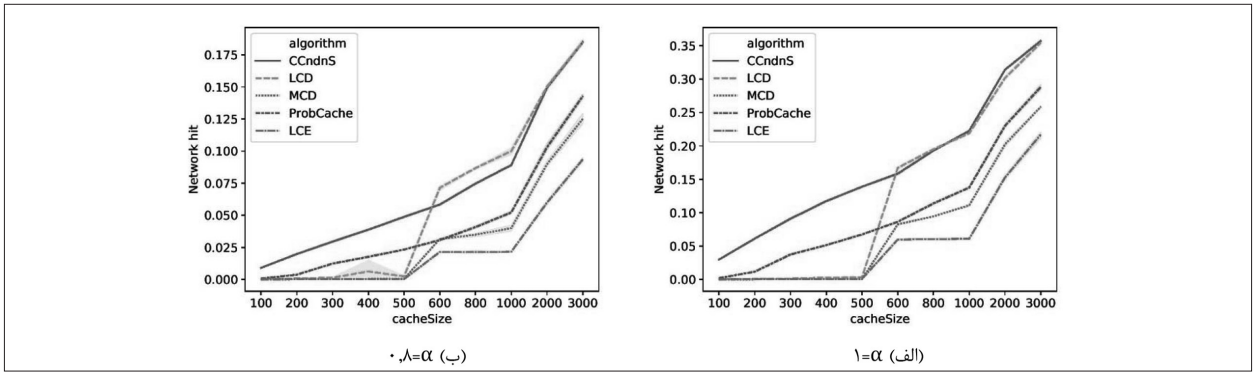
پنج الگوریتم تصمیم‌گیری حافظه نهان با هم مقایسه شده‌اند. جهت بررسی این داده‌ها، می‌توان آن‌هـــا را به دو دسته کلی تقسیم کرد: ۱- نتایجی که برای حافظه‌های با اندازه کمتر از اندازه یک فایل هستند ۲- نتایج حاصل از اجرای برنامه برای حافظه‌های بزرگ‌تر از اندازه یک فایل. در تمام اشکال ۸،۷،٦،۵ و ۹ رفتار الگوریتم‌ها، با عبور اندازه حافظه از اندازه یک فایل تغییر محسوسی داشته است. در قسمت‌های ذیل به بررسی هر یک از این نتایج می‌پردازیم.

۶-۱ احتمال یافتن داده در شبکه

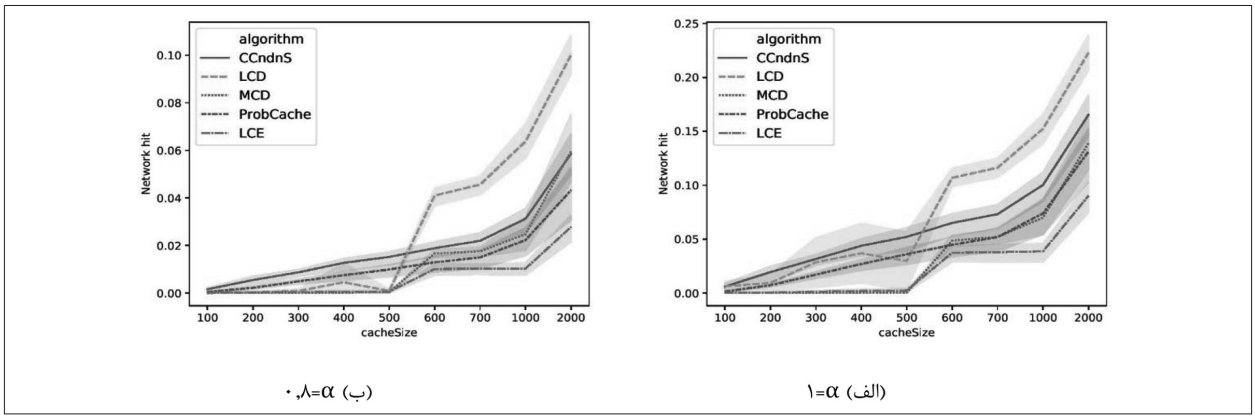
شکل‌هـــای ۵ و ٦ گزارشی از احتمال یافتن داده در یکی از گره‌های شبکه (قبل از رسیدن به کارســـاز) را ارائه می‌دهند. در شکل ۵ کاربران و کارسازها به‌صورت ثابت/ دستی و در شکل ٦ به‌صـــورت تصادفی به مسیریاب‌های شبکه Geant اختصاص داده شده‌اند. این نتایج برای دو مقدار ضریب محبوبیت α = 1 و α = 0.8 در شکل‌های به ترتیب (۵ (الـــف) و ٦ (الف)) و (۵ (ب)و ٦ (ب)) نمایش داده شده است. از آنجا که رفتار الگوریتم‌ها با تغییر این پارامتر ثابت است، از تکرار آن‌هـــا برای دیگر پارامترها خودداری شده است. همان‌طور که در شکل‌های ۵ و ٦ مشخص است، هنگامی که اندازه حافظه نهان کمتر از اندازه یک فایل است، تنها دو روش CC-ndn و ProbCache دارای احتمال برخوردی بیشتر از صفر هستند. ولی سایر روش‌ها (LCD، MCD، LCE) با مشکل مواجه می‌شوند. دلیل آن، مشکل به‌وجـــود آمدن پدیده thrashing برای حافظه است.

در این حالت چون حافظه کافی برای ذخیره تمام بسته‌های یک فایل درخواستی در یک گره وجود ندارد، بسته‌های یک فایل جایگزین بسته‌های اولیه همان فایل می‌شوند. بنابراین فرصتی برای این بسته‌ها وجود ندارد تا بتوانند جوابگوی درخواست‌های دیگر کاربران باشند. اما دو روش CCndn و ProbCache تمام بسته‌های یک فایل را در یک گره ذخیره نمی‌کنند. در CCndn فقط یک و یا تعداد محدودی از تکّه‌ها (بسته به طول مسیر) در یک گره ذخیره می‌شوند. به‌عنوان مثال در این شبیه‌ساز، همان‌طور که در جدول ۱ نیز گزارش شده، طول هر تکّه ۱۰۰ بسته می‌باشد و بنابراین اگر اندازه حافظه بیشتر از یکصد بسته باشد پدیده کوبیدگی برای آن رخ نخواهد داد. در الگوریتم ProbCache نیز از آنجا که ذخیره بسته‌ها به‌صـــورت احتمالی صورت می‌گیرد، احتمال این‌کـــه تمام بسته‌های تمام فایل‌ها در تمام گره‌ها ذخیره شوند بسیار ناچیز است. نتایج نشان می‌دهند در صورتی‌که اندازه حافظه‌نهان کافی نباشد، باقی الگوریتم‌ها نرخ برخوردی نزدیک به صفر خواهند داشت. با افزایش اندازه حافظه‌نهان، یک تغییر اساسی در رفتار تمام الگوریتم‌ها ایجاد می‌شود. در این حالت دو الگوریتم CCndn و LCD عملکرد بهتری از بقیه دارند.

دلیل آن استفاده بهینه‌تر از حافظه بـــه‌وسیله این دو الگوریتم می‌باشد. از آنجا که تنها داده‌های محبوب در الگوریتم LCD وارد گره لبه می‌شوند، این داده‌ها نیاز به رقابت جهت حفظ مکان خود در گره لبه را ندارند. به بیان دیگر، با استفاده از الگوریتم،LCD مقدار داده ورودی به گره لبه بسیار کمتر از بقیه الگوریتم‌ها است و به همین دلیل تعداد بیشتری از بسته‌های محبوب می‌توانند در این گره‌ها ذخیره شوند. این تحلیـــل درمورد الگوریتم CCndn نیز صادق است. از آنجا که با استفاده از این دو الگوریتم، تعداد بیشتری از داده‌های محبوب می‌توانند در شبکه



شکل ۵: مقایسه پنج الگوریتم تصمیم گیری حافظه نسبت به احتمال یافتن داده از یکی از گره‌های شبکه، (انتخاب کاربر – کارساز ثابت)



شکل ۶: مقایسه پنج الگوریتم تصمیم گیری حافظه نسبت به احتمال یافتن داده از یکی از گره‌های شبکه، (انتخاب کاربر – کارساز تصادفی)

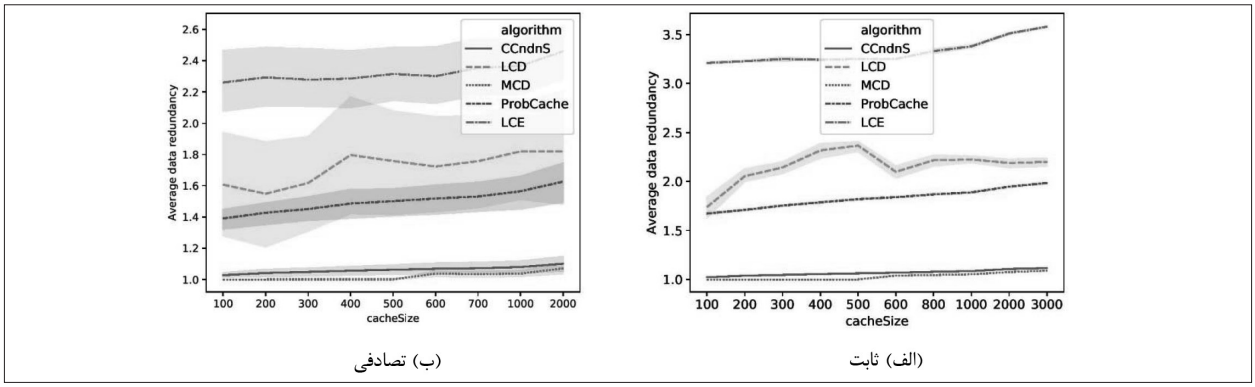
ذخیره شوند، عملکرد این دو الگوریتم از بقیه بسیار بهتر است. همان‌طــورکه قبلا نیز گفته شد در شکل ٦ کاربران و کارسازها به‌صورت تصادفی به مسیریاب‌ها اختصاص داده شده‌اند. لازم به ذکر است‌برای بالا بردن دقت نتایج، نتایج نمایش‌داده شده در تمامی اشکال تصادفی، میانگین نتایج حاصل از چندین بار اجرای شبیه‌ساز است.

مطابق با آنچه که در ابتدای این بخش گفته شد و چنانچه در شکل‌هــای ٥ و ٦ می‌بینیم، برخلاف روش‌های CCndn و ProbCache، در روش‌های MCD، LCE، LCD پس از افزایش ظرفیت حافظه نهــان به اندازه بیش از یک فایل، فضای کافی برای ذخیره محبوب‌تریــن داده در گره لبه ایجاد شده و فرصتی برای این بسته‌های داده ایجاد می‌شود تا بتوانند جوابگوی درخواست‌های دیگر کاربران باشند. و بدین ترتیب یک جهش بزرگ در نمودار آن‌ها قابل مشاهده است.

۶-۲ تعداد نسخهٔ بسته‌ها در شبکه

تعداد نســخهٔ داده‌ها در شبکه معیاری است جهت نشان دادن کارآیی الگوریتم در استفاده بهینه از حافظه نهان موجود در شبکه. کاهش تعداد نســخه‌ها باعث افزایش تنوع داده‌ها در حافظه نهان موجود در شبکه خواهد شد. همان‌طــور که شکل ۷ (الــف) نشان می‌دهد، دو الگوریتم CCndn و MCD کمترین تعداد نســخــه در شبکه را باعث می‌شوند. این مقدار برای الگوریتم LCE که در تمام گره‌ها یک نسخه از هر بسته باقی می‌گذارد از همه بیشتر است. در مورد الگوریتم،LCD هنگامی که اندازه حافظه نهان کوچک است، احتمال برخورد داده در حافظه نهان تقریبا صفر است (شکل ٥). در نتیجه تمامی درخواست‌ها باید به کارســاز رسیده که منجر به افزایش تعداد نسخه‌ها در شبکه می‌شوند. در صورتی‌که، با افزایش حافظه نهان، داده‌های محبوب از شبکه عبور نمی‌کنند و تعداد نسخهٔ آن‌ها کم می‌شوند.

با مقایسه شکل‌های ٥ و ٧ برای دو الگوریتم LCD و MCD مشخص می‌شود که تعداد نسخــه داده کمی بیشتر از یک، منجر به افزایش کارآیی می‌شود. با وجود

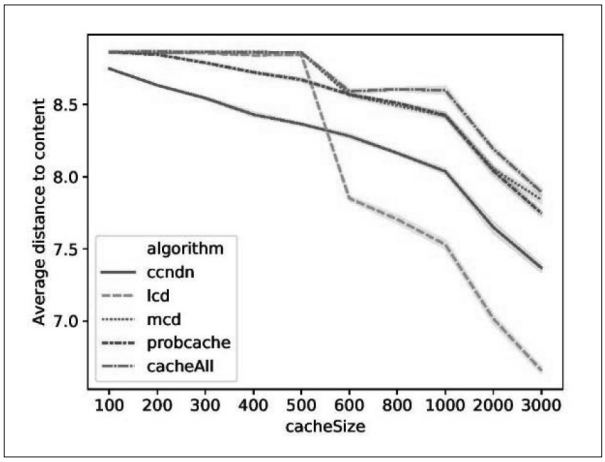


شکل ۷: مقایسه تعداد نسخه‌ها در شبکه

این‌که تعداد نسخـــه‌ها در الگوریتم MCD تقریبا یک است، احتمال برخورد داده در آن بسیار کمتر است. دلیل آن احتمال استفاده نسخـــه داده در یک گره میانی بـــه‌وسیله درخواست‌هایی است که از طرف کاربرانی که از یک منطقه جغرافیایی دیگر درخواست ارسال می‌کنند.

۶-۳ متوسط فاصله کاربران تا داده‌ها

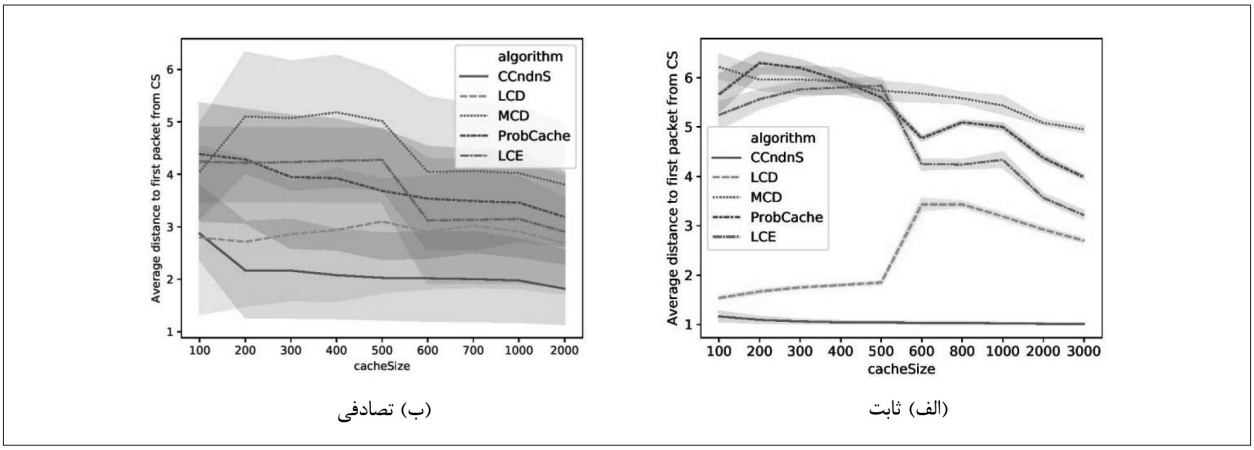
یکی از مهم‌تـــرین موارد استفاده از حافظه‌های‌نهان، کاهش فاصله کاربران تا داده می‌باشد. بنابراین، الگوریتم تصمیم‌گیری داده در حافظه نهان باید سعی نماید تا این مقدار را هرچه بیشتر کاهش دهد. شکل ۸ به مقایسه این پارامتر برای الگوریتم‌های مورد بررسی پرداخته است. الگوریتم CCndn هنگامـــی کـــه اندازه حافظه نهان کوچک است، بهترین عملکرد را دارد. اما هنگامـــی که اندازه حافظه افزایش می یابد، LCD بهتر عمل خواهد کرد. همان‌طـــور که ذکر شد، LCD سعی در انباشت داده‌های محبوب در گره لبه می‌نماید. بنابراین، با استفاده از این روش داده‌های محبوب در کمترین فاصله از کاربران قرار دارند که منجر به کاهش چشمگیر متوسط فاصله کاربران تا داده می‌شود.



شکل ۸: متوسط فاصله کاربران تا داده‌ها

۶-۴ متوسط فاصله کاربر تا اولین بسته در شبکه

اگرچه شکل ۸ نشان می‌دهد که الگوریتم LCD عملکرد بهتری در کاهش فاصله داده تا کاربر دارد، نتایج شکل ۹، عملکرد بهتر CCndn را نشان می‌دهد. همان‌طـــور که در بخش ٤- ٥ اشاره شد، روش CCndn جهت جایگذاری داده‌ها، توزیع آن‌هـــا درطول مسیر مسیریابی است. دلیل اصلی آن قابلیت همپوشانی تاخیر دسترسی به داده‌های انتهایی با داده‌های ابتدایی می‌باشد. لذا شکل ۹ (الـــف) وسط فاصله کاربران به اولین بسته داده‌ها را گزارش می نماید. از آنجا که تنها قسمتی از فایل‌ها با استفاده از روش CCndn در یک گره ذخیره می‌شود، تعداد فایل‌های محبوب بیشتری می‌توانند قسمت ابتدایی فایل خود را در گره لبه ذخیره نمایند. بنابراین، این الگوریتم می‌تواند فاصله کاربران تا اولین بسته (و تمام بسته‌های اولین تکّه) را تقریبا به یک کاهش دهد. همان‌طـــور که در شکل ۸ قابل مشاهده است، پس از افزایش ظرفیت حافظهٔ نهان به اندازه حداقل یک فایل، هر بار که ظرفیت حافظـــهٔ نهان افزایش می‌یابد یک فایل اضافه می‌شود، یک جهش بزرگ در جهت کاهش فاصله داده تا کاربر در نمودار رخ می‌دهد. نتایج ارائه شده در شکل‌های ۸ و ۹ نشان می‌دهد که الگوریتم MCD موفق به کاهش فاصله کاربر تا داده نشده و از



(الف) ثابت (ب) تصادفی

شکل ۹: متوسط فاصله کاربران تا اولین بسته در شبکه

بین بردن نسخــه داده جهت کاهش تعداد نسخه‌ها انتخاب مناسبی نبوده است. الگوریتم ProbCache نیز همان‌گونه که از یک روش احتمالاتی انتظار مــی‌رود، از کارآیی متوسطی برخوردار است.

۷- نتیجه گیری

این مقاله به بررسی و مقایسه اجمالی عملکرد چند الگوریتم تصمیم‌گیری جهت جایگذاری داده در حافظه نهان در شبکه‌های مبتنی بر داده پرداخته است. شبکه‌های مبتنی بر داده، با اختصاص نام به داده به جای اختصاص نام به مکان داده، از ویژگی‌های ممتازی برخوردار هستند. از جمله این ویژگی‌ها امکان استفاده از حافظه نهان در لایه شبکه و در داخل دستگاه‌های مسیریاب است. ازآنجاکه مسیریاب‌ها توان عملیاتی بسیار بالایی دارند و مقدار داده زیادی در هر لحظه وارد این دستگاه‌ها می‌شوند، باید سازوکــارهایی اتخاذ شود تا کارآیی حافظه نهان در آن‌هــا افزایش یابد. از جمله این سازوکارها استفاده از الگوریتم‌های تصمیم‌گیری جهت جایگذاری داده می‌باشد تا بتوان در حافظه نهان مسیریاب‌ها، بهترین چیدمان را به‌دســت آورد. در این مقاله، تعدادی از الگوریتم‌هایی که هرکدام نماینده دسته‌ای از این الگوریتم‌ها می‌باشند و بهترین کارآیی را در دسته خود دارند انتخاب شدند تا با یکدیگر مقایسه گردند. این دسته‌ها شامل جایگذاری داده‌ها ۱- به‌صورت توزیعی، ۲-با درنظر گرفتن محبوبیت داده ۳- به‌صــورت تصادفی و ٤- بدون هیچ سیاستی جهت افزایش سادگی و کاهش پیچیدگی می‌باشند. نتایج نشان دادند که دو الگوریتم CCndn و LCD که به ترتیب نماینده دسته الگوریتم‌های توزیعی و برمبنای محبوبیت می‌باشند بهترین عملکرد را دارا هستند. مزیت CCndn نسبت به LCD در آن است که این الگوریتم حتی برای حافظه‌های بسیار کوچک نیز عملکرد قابل‌قبولی دارد. درضمن، این الگوریتم می‌تواند زمان دسترسی به بسته‌های اولیه یک فایل را بسیار کاهش دهد و درمجموع با همپوشانی زمان دسترسی داده‌ها می‌تواند فایل‌ها را با تاخیر کمتری در اختیار کاربران قرار دهد. نقطه قوت الگوریتم LCD، در کاهش فاصله کاربران تا فایل‌های محبوب می‌باشد. درضمن کارآیی گره‌های لبه با استفاده از این الگوریتم بسیار بالا است. نتایج این پژوهش نشان می‌دهد که از بین‌بردن داده‌ها بــه‌وسیله الگوریتم تصمیم‌گیری انتخاب درستی نمی‌باشد. این عمل که در الگوریتم MCD صورت می‌گیرد باعث کاهش راندمان حافظه نهان چــه از لحاظ برخــورد و چه از لحاظ کاهش فاصله کاربران تا داده می‌گردد.

مراجع

1. H. Che, Y. Tung, and Z. Wang, “Hierarchical Web caching systems: modeling, design and experimental results,” JSAC, vol.20, no.7, pp.1305–1314.
2. L. Zhang, A. Afanasyev, J. Burke, et al., “Named Data Networking,” SIGCOMM Computer Communication Review, vol.44, pp.66–73, Jul 2014.

3. V. Jacobson, D. K. Smetters, J. D. Thornton, et al., "Networking named content," in Proc. CoNEXT, pp.1–12, 2009.
4. C. Bernardini, T. Silverston, and O. Festor, "Mpc: Popularity-based caching strategy for content centric net-works," in 2013 IEEE International Conference on Com-munications (ICC), pp.3619–3623, June 2013.
5. N. Kamiyama and M. Murata, "Spatially-dispersed caching in information-centric networking," in 2018 IEEE International Conference on Communications (ICC), pp.1–7, May 2018.
6. J. Li, H. Wu, B. Liu, J. Lu, Y. Wang, X. Wang, Y. Zhang, and L. Dong, "Popularity-driven coordinated caching in named data networking," in Proc. ANCS, pp.15–26, 2012.
7. Z. Ming, M. Xu, and D. Wang, "Age-based coopera-tive caching in information-centric networks," in 2012 Proceedings IEEE INFOCOM Workshops, pp.268–273, March 2012.
8. H. Wu, J. Li, T. Pan, and B. Liu, "A novel caching scheme for the backbone of named data networking," in 2013 IEEE International Conference on Communications (ICC), pp.3634–3638, June 2013.
9. H. Wu, J. Li, Y. Wang, and B. Liu, "EMC: The effective multi-path caching scheme for named data networking," in Proc. ICCCN, pp.1–7, July 2013.
10. J. A. Khan, C. Westphal, J. Garcia-Luna-Aceves, and Y. Ghamri-Doudane, "Nice: Network-oriented information-centric centrality for efficiency in cache management," in Proceedings of the 5th ACM Conference on Information-Centric Networking, pp.31–42, 2018.
11. S. H. Ahmed and D. Kim, "Named data networking-based smart home," ICT Express ELSEVIER, vol.2, no.3, pp.130–134, 2016.
12. R. A. Rehman, J. Kim, and B.-S. Kim, "NDN-CRAHNs: Named data networking for cognitive radio ad hoc networks," Mobile Information Systems, vol.2015, 2015.
13. X. Liu, M. J. Nicolau, A. Costa, J. Macedo, and A. Santos, "A geographic opportunistic forwarding strategy for vehicular named data networking," in Intelligent Distributed Computing IX, pp.509–521, Springer, 2016.
14. M. Amadeo, C. Campo, A. Molinaro, and N. Mitton, "Named data networking: A natural design for data collection in wireless sensor networks," in IEEE Wireless Days (WD), 2013.
15. M. A. Hail, M. Amadeo, A. Molinaro, and S. Fischer, "Caching in named data networking for the wireless internet of things," in 2015 International Conference on Recent Advances in Internet of Things (RIoT), pp.1–6, April 2015.
16. B. Ager, F. Schneider, J. Kim, and A. Feldmann, "Revisiting cacheability in times of user generated content," in INFOCOM IEEE Conference on Computer Communications Workshops , 2010, pp.1–6, March 2010.
17. D. Rossi and G. Rossini, "Caching performance of con-tent centric networks under multi-path routing (and more)," Tech. Rep. Telecom ParisTech, 2011.
18. D. Perino and M. Varvello, "A reality check for content centric networking," in Proc. ICN, pp.44–49, 2011.
19. F. Qazi, O. Khalid, R. N. B. Rais, I. A. Khan, et al., "Caching in content-centric networks," Wireless Communications and Mobile Computing, vol.2019, 2019.
20. L. Fan, P. Cao, J. Almeida, and A. Z. Broder, "Summary cache: A scalable wide-area web cache sharing protocol," IEEE/ACM Trans. Netw., vol.8, pp.281–293, June 2000.
21. S. Michel, K. Nguyen, A. Rosenstein, et al., "Adaptive web caching: Towards a new global caching architecture," Comput. Netw. ISDN Syst., vol.30, pp.2169–2177, Nov. 1998.
22. M. Rabinovich, J. S. Chase, and S. Gadde, "Not all hits are created equal: Cooperative proxy caching over a wide-area network," Computer Networks, vol.30, no.22-23, pp.2253–2259, 1998.
23. A. Wolman, M. Voelker, N. Sharma, et al., "On the scale and performance of cooperative web proxy caching," in Proc. SOSP, pp.16–31, 1999.
24. Z. Li and G. Simon, "Time-shifted TV in content centric networks: The case for cooperative in-network caching," in ICC, pp.1–6, 2011.
25. Y. Wang, Z. Li, G. Tyson, S. Uhlig, and G. Xie, "De-sign and evaluation of the optimal cache allocation for content-centric networking," IEEE Transactions on Com-puters, vol.65, no.1, pp.95–107, 2015.
26. S. Naz, R. N. B. Rais, P. A. Shah, S. Yasmin, A. Qayyum, S. Rho, and Y. Nam, "A dynamic caching strategy for ccn-based MANETs," Computer Networks, vol.142, pp.93–107, 2018.
27. D. Grewe, M. Wagner, M. Arumaithurai, I. Psaras, and Kutscher, "Information-centric mobile edge computing for connected vehicle environments: Challenges and re-search directions," in Proceedings of the Workshop on Mobile Edge Communications, pp.7–12, ACM, 2017.
28. Z. Jia, P. Zhang, J. Huang, et al., "Modeling hierarchi-cal caches in content-centric networks," in Proc. ICCCN, pp.1–7, 2013.
29. I. Psaras, R. G. Clegg, R. Landa, et al., "Modelling and evaluation of CCN-caching trees," in Proc. IFIP Networking, pp.78–91, 2011.
30. S. K. Fayazbakhsh, Y. Lin, et al., "Less pain, most of the gain: Incrementally deployable ICN," in SIGCOMM, pp.147–158, 2013.
31. G. Tyson, S. Kaune, S. Miles, et al., "A trace-driven analysis of caching in content-centric networks," in Proc. ICCCN, pp.1–7, 2012.
32. E. Aubry, T. Silverston, and I. Chrisment, "Green growth in ndn: Deployment of content stores," in 2016 IEEE International Symposium on Local and Metropolitan Area Networks (LANMAN), pp.1–6, June 2016.
33. N. Laoutaris, H. Che, and I. Stavrakakis, "The LCD interconnection of LRU caches and its analysis," Perform. Eval., vol.63, no.7, pp.609–634, 2006.
34. N. Laoutaris, S. Syntila, and I. Stavrakakis, "Meta algorithms for hierarchical web caches," in Proc. IPCC, pp.445–452, 2004.
35. I. Psaras, W. K. Chai, and G. Pavlou, "Probabilistic in-network caching for information-centric networks," in Proc. ICN, pp.55–60, 2012.
36. M. Rezazad and Y. C. Tay, "CCndnS: A strategy for spreading content and decoupling NDN caches," in 2015 IFIP Networking Conference (IFIP Networking), pp.1–9, May 2015.