حداقل چارچوب مدیریت هزینه کش برای شبکه های اطلاعات محور با برنامه نویسی شبکه

چکیده

در سال های اخیر، افزایش تقاضا برای محتوای رسانه های غنی، تلاش های زیادی را برای طراحی مجدد معماری اینترنت به ارمغان آورده است. به عنوان یکی از نامزدهای اصلی، *شبکه اطلاعات محور* (ICN)[1] توجهات زیادی را به خود جلب کرده است، جایی که *کش در شبکه* یک جزء کلیدی در معماری های مختلف ICN است. در این مقاله، ما یک چارچوب جدید برای مدیریت حافظه پنهان در ICN مطرح می کنیم که به طور مشترک استراتژی ذخیره سازی و مسیریابی محتوا را در نظر می گیرد. به طور خاص، چارچوب ما مبتنی بر *شبکه بندی تعریف شده توسط نرم افزار* (SDN)[2]است که در آن یک کنترل کننده مسئول تعیین استراتژی ذخیره سازی مطلوب و مسیریابی محتوا از طریق *کدگذاری شبکه خطی* (LNC)[3] است. برای چارچوب مدیریت حافظه پنهان پیشنهادی، ما ابتدا یک مشکل بهینه سازی را برای به حداقل رساندن هزینه پهنای باند شبکه و هزینه نگهداری با بررسی مشترک استراتژی ذخیره کردن و مسیر یابی محتوایی باLNC ، طراحی کردیم. سپس یک الگوریتم *مدیریت کارآیی مخزن مبتنی بر برنامه نویسی شبکه* (NCCM)[4]را برای به دست آوردن یک راه حل ذخیره سازی نزدیک و بهینه سازی برای ICN ها ایجاد کردیم. علاوه بر این، مراتب بالایی و پایین تر مشکل را استنتاج می کنیم و آزمایش های گسترده ای را برای مقایسه عملکرد الگوریتم NCCM با این محدودیت ها انجام می دهیم. نتایج شبیه سازی اثربخشی الگوریتم NCCM و چارچوب را تأیید می کند.

1 .مقدمه

[1] Information-centric network
[2] Software-defined networking
[3] Linear network coding
[4] Network coding based cache management

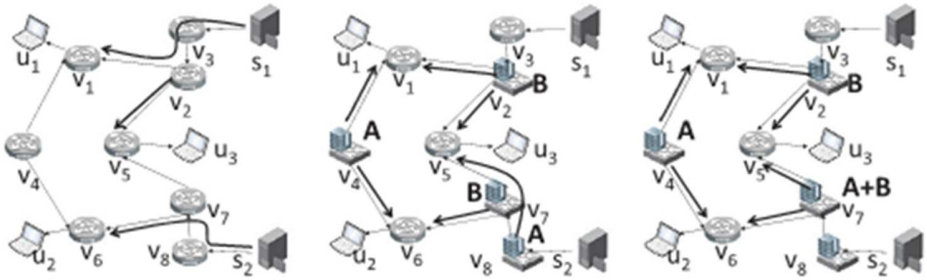
در دهه گذشته، محتوای چند رسانه ای ترافیک غالب اینترنت شده است [2-4]. افزایش تقاضا برای محتوای غنی از رسانه ها، مستلزم روش های کارآمدتر برای بازیابی محتوا است. برای این منظور، *شبکه اطلاعاتی محور* (ICN) یک رویکرد امیدوار کننده طراحی است که این تقاضا را با ارائه دسترسی محتوا به نام و امکان *ذخیره در شبکه* در اختیار قرار می دهد [5،6]. در ICN ها، یک *روتر محتوا* (CR)[5] با قابلیت ذخیره در شبکه می تواند برخی از (معمولا محبوبیت) تکه های داده را برای دسترسی را وساطت کند [7]. ذخیره سازی در شبکه می تواند تاخیر بازیابی محتوا، ترافیک شبکه و بار سرویس بر روی سرورها را تا حد زیادی کاهش دهد8] ، [9.

برای مدیریت مخازن در شبکه درICN ، دو مساله مهم باید بصورت مشترک مورد توجه قرار گیرد. یکی *استراتژی ذخیره سازی* است که تعیین می کند که چه مقدار از داده ها باید در هر CR ذخیره شود، و دیگری *مسیریابی محتوا* است که تعیین می کند کجا برای مسیریابی محتوا لازم است و نحوه ارائه مطالب چگونه است.

در متون، دو نوع استراتژی ذخیره سازی وجود دارد: غیر تعاونی و تعاونی. در استراتژی های ذخیره سازی غیر تعاونی، CR به طور تصادفی داده های دریافتی را ذخیره می کند، که ممکن است منجر به به روز رسانی حافظه پنهان مکرر، تخصیص حافظه پنهان غیرقابل کپی و کپی مجدد شود [8]. در استراتژی های ذخیره سازی تعاونی، یک CR می تواند با CR های همسایه خود همکاری کند تا تعیین کند که کدام مجموعه داده ها برای حافظه پنهان است [9-12].

برای مسیریابی محتوا، دو روش مختلف برای استفاده از مخازن در شبکه وجود دارد. یکی از اینها فقط استفاده از حافظه های ذخیره شده در مسیر سرور اصلی محتوا برای آن درخواست است و دیگری استفاده از تمام مخازن ذخیره سازی نزدیک است. اولی نیازمند هیچ نوع همکاری بین CR ها نیست اما ممکن است بصورت بالقوه تاخیر بازیابی محتوایی بیشتری را نشان دهد. دومی نیازمند همکاری میان CR ها برای ارسال درخواست به نزدیک ترین حافظه خارج از مسیر است [13]. هر دو حالت با ذخیره محتوا ارتباط نزدیکی دارند. در این مقاله، ما بر روی استراتژی ذخیره سازی همکاری و مسیریابی محتوا تمرکز خواهیم کرد تا تمام مخازن توزیع شده در شبکه را به طور کامل استفاده کنیم .

[5] Content router



(a) With no In-network Cache. (b) ICNs without LNC. (c) ICNs with LNC.

شکل 1. نمونه ای از درخواست محتوا در سناریوهای مختلف شبکه

برای فعال کردن همکاری در میان CR های توزیع شده، برای جمع آوری اطلاعات مربوط به همکاری (به عنوان مثال، نرخ درخواست و وضعیت فعلی کش)، یک چارچوب مدیریت کش لازم است و تصمیم گیری های ذخیره سازی و مسیریابی را می گیرند. نرم افزار *تعریف شبکه بندی* (SDN)، که از نظر فیزیکی کنترل کننده هواپیما و هواپیما داده را جدا می کند، می تواند این نیاز را برآورده کند [14،15]. به طور معمول، در کنترل هواپیما، یک کنترل کننده مسئول جمع آوری اطلاعات شبکه و تصمیم گیری مسیریابی است که در روتر ها پیکربندی می شود. در هواپیما داده، روترها بسته های روبرو را با توجه به تنظیمات جریان توسط کنترل کننده تنظیم می کنند. طی چند سال گذشته، بسیاری از کنترل کننده های جدید با استفاده از سرورهای چند هسته ای قدرتمند برای رسیدگی به تعداد زیادی جریان داده در شبکه های بزرگ طراحی شده اند. به عنوان مثال، McNettle [16] می تواند حدود 20 میلیون درخواست در هر ثانیه را برای یک شبکه با 5000 سوئیچ مدیریت کند .

به تازگی، مطالعات مقدماتی برای امکان مدیریت کش در ICN ها براساس SDN انجام شده است [17،18]. با این حال، این مطالعات عمدتا بر چگونگی ادغام عملیات مرتبط با حافظه پنهان در معماری SDN موجود متمرکز شده و در مورد استراتژی ذخیره سازی واقعی بحث نمی کنند. در این مقاله، برای مطالعه استراتژی ذخیره سازی و مسیریابی محتوا ICN ها بر اساس SDN با هدف به حداقل رساندن پهنای باند شبکه و هزینه کش یک گام به جلو بر می داریم، که کل هزینه پهنای باند و مصرف حافظه پنهان در کل شبکه است.

به طور خاص، ما *برنامه نویسی خطی شبکه* (LNC) را برای بهینه سازی مشترک استراتژی ذخیره سازی و مسیریابی محتوا برای به حداقل رساندن پهنای باند شبکه و هزینه کش بکار می بریم. ما از نمونه ای که در شکل 1 نشان داده

شده است برای نشان دادن مزایای استفاده از ذخیره سازی و LNC در ICN ها استفاده می کنیم. در این شکل، یک شبکه شامل هشت روتر (v 1 -v 8) و دو سرور (s 1 وs 2) است. کاربران همه به روترهایv1 ، v5 وv6 متصل می شوند و یک تکه محتوا را به عنوان1 f ، که شامل دو تکه داده ای با اندازه یکسان، A وB است، درخواست می کنند. ما فرض می کنیم که هر لینک یک هزینه واحد برای ارسال یک تکه داده است و یک روتر یک هزینه واحد برای ذخیره یک تکه داده دارد. از لحاظ هزینه کل، یعنی مجموع هزینه پهنای باند و هزینه کش، ما در سه سناریوی تحویل محتوا مختلف، موارد زیر را داریم :

- در شکل 1 (a) سناریوی پایه ای را بدون کش در شبکه بررسی می کنیم، بنابراین بهترین راه برای به دست آوردن محتوای تعیین شده، استفاده از چندپخشی است که هفت پیوند در مسیریابی استفاده می شود.در این مورد، مخزن ذخیره شده در شبکه وجود ندارد. برای هر تکه داده، 7 پیوند استفاده می شود و هر لینک دارای ظرفیت واحد است. بنابراین، برای انتقال دو قطعه داده، هزینه حافظه کش 0 و هزینه پهنای باند 2 × 7 = 14 است. هزینه کل 14= 14 + 0 است .
- در شکل 1 (b)، ما فرض می کنیم که چهار CR (v 2، v 4، v 7 و v 8) وجود دارد و هر یک از آنها می توانند تنها یک تکه داده را ذخیره کنند. در این سناریو، ICN را بدون LNC در نظر می گیریم، بنابراین هر CR می تواند یک تکه داده اولیه را ذخیره کند. شکل 1 (b) استراتژی ذخیره سازی بهینه و مسیریابی محتوا را نشان می دهد، که در آن نماد پررنگ نشان داده شده در هر CR نشان دهنده داده های تکه ای ذخیره شده در CR است. در این حالت، مجموع 4 تکرار داده در CR ها ذخیره می شود، و انتقال دو تکه داده نیاز به 7 واحد مصرف پهنای باند دارد. بنابراین، برای انتقال دو قطعه داده، هزینه ذخیره سازی 4 است و هزینه پهنای باند 7 است. هزینه کل 4 + 7 = 11 است، که نشان دهنده پیشرفت 21.42 درصدی است.
- شکل 1 (c) سناریو را با مدیریت کش بهینه در ICN ها با LNC نشان می دهد. در این مورد، CR ها می توانند ترکیب خطی داده های اصلی را ذخیره کنند؛ و برای بازیابی داده های اصلی داده های A و B، کاربر فقط باید هر دو تکه داده های مستقل خطی مستقل را بدست آورد. با راه حل مطلوب، هر روتر (i.e.، v1، v5 و v6) می تواند دو تکه داده های

کد شده را از دو تا از نزدیکترین CR ها به آنها دانلود کند، بنابراین CR فقط نیاز به ذخیره 3 تکه داده دارد و هزینه پهنای باند 6 واحد است. بنابراین، هزینه کل برابر 3 + 6 = 9 در مقایسه با بهترین راه حل در سناریو 1 است، راه حل بهینه برای سناریو 3 به افزایش 35.71٪ می رسد؛ و در مقایسه با بهترین راه حل در سناریو 2، به افزایش 18.18٪ می رسد.

مثال بالا مزیت مشترک در نظر گرفتن استراتژی ذخیره سازی در شبکه و مسیریابی محتوا با LNC در ICN ها، را نشان می دهد که کار این مقاله را پیش می برد. مفاد اصلی این مقاله به شرح زیر خلاصه شده است .

- ما یک چارچوب جدید مبتنی بر SDN را برای تسهیل در پیاده سازی استراتژی ذخیره سازی و مسیر محتوا در ICN ها با LNC پیشنهاد می کنیم. چارچوب مبتنی بر مفهوم در حال ظهور SDN است، که در آن یک کنترل کننده مسئول تعیین استراتژی ذخیره سازی بهینه و همچنین مسیریابی محتوای بهینه از طریق LNC است .
- ما یک مشکل مدیریت بهینه حافظه پنهان برای ICN ها را با LNC تحت یک استراتژی کش به عنوان یک مسئله *برنامه نویسی خطی صحیح* (ILP)[6] تشکیل می دهیم. بر اساس این ILP پایه، ما فرموله سازی ILP را به منظور کاهش هزینه کل پهنای باند شبکه و هزینه ذخیره سازی به صورت مشترک با توجه به استراتژی ذخیره سازی و مسیر محتوا محاسبه می کنیم.
- ما یک الگوریتم *مدیریت رمزنگاری مبتنی بر کدگذاری شبکه* (NCCM) را برای رسیدن به یک راه حل مدیریت نزدیک به مطلوب به کار می بریم. بر اساس آرامش لاگرانژی، مشکل فرموله شده را می توان کم کرد و سپس به یک مسئله برنامه ریزی خطی و چندین مسئله ساده بزرگ شدن وزن با عدد صحیح ساده، که می تواند به طور مطلوب در طی زمان چندجمله ای حل شود، تجزیه می شود .
- ما انجام آزمایش های گسترده ای را برای مقایسه عملکرد الگوریتم NCCM پیشنهادی با حد پایین فرموله سازی ILP انجام می دهیم. ما همچنین عملکرد الگوریتم NCCM پیشنهاد شده را با سه حد بالایی از مشکل مقایسه می کنیم،

[6] Integer linear programming

یعنی *بدون کش* (بدون حافظه کش)، *حافظه تصادفی* (r-Cache) و *کش های حریص* (g-Cache). نتایج شبیه سازی اثربخشی الگوریتم NCCM پیشنهادی و چارچوب را اثبات می کند.

بقیه مقاله به شرح زیر است: ما درباره کار مرتبط در بخش 2 بحث می کنیم. در بخش 3، یک چارچوب مدیریت کل کش برای ICN ها را بر اساس SDN معرفی می کنیم. سپس ما مشکل مدیریت بهینه حافظه پنهان برای ICN ها با LNCرا مطرح می کنیم که هدف آن کاهش پهنای باند شبکه و هزینه ذخیره با استفاده از مخازن در شبکه و LNC در بخش 4 می باشد. برای حل مشکل در عمل، در بخش 5، ما یک الگوریتم کارآمد را بر اساس کاهش لاگرانژی طراحی می کنیم. سپس تجربه های گسترده ای را برای نشان دادن عملکرد چارچوب مان در بخش 6 انجام می دهیم. در نهایت در مورد کاربرد طرح پیشنهادی در بخش 7 بحث خواهیم کرد و مقاله را در بخش 8 به پایان می رسانیم .

2. کار مرتبط

به تازگی نشان داده شده است که SDN می تواند عملکرد توزیع کش و کارایی مدیریت محتوا را به طور قابل توجهی بهبود بخشد. در [14] و [15]، راه حل ها به ترتیب برای حمایت از ICN با استفاده از مفاهیم SDN در شبکه های سیمی و بی سیم پیشنهاد و مورد بحث قرار می گیرند. با این حال، بحث در مورد بررسی مشترک مدیریت کش و تصمیم گیری مسیریابی به منظور کاهش هزینه های شبکه وجود ندارد.

چندین سیستم مدیریت حافظه پنهان در ICN ها [10،12،17،18] پیشنهاد شده است. در [17]، مدیریت حافظه پنهان با کنترل کننده یکپارچه شده است، اما استراتژی ذخیره سازی فعلی نامشخص است. در[18] API ها برای پشتیبانی از عملیات مرتبط با حافظه پنهان، از جمله تصمیم گیری های ذخیره سازی، اعلان های حافظه پنهان و ذخیره سازی پیشگیرانه تعریف شده اند. با این حال، هیچ بحثی در مورد چگونگی استفاده از این API ها برای مدیریت حافظه های ذخیره شده وجود ندارد، و هیچ الگوریتم استراتژی ذخیره سازی بتونی پیشنهاد نشده است .

همچنین برخی از منافع در ذخیره سازی تعاونی برای بهبود کارایی cache در ICN ها وجود داشته است [9-12]. در[9] CR ها در یک مسیر در یک روش توزیع هماهنگ شده است. داده های موجود در CRs پایین نگه داشته شده توسط CRs upstream توصیه می شود. در [10] یک سیستم مدیریت حافظه توزیع شده برای تغییر اطلاعات کش

در میان تمام حافظه های ذخیره سازی برای دریافت اطلاعات جهانی از شبکه توسعه داده شده است. سپس، تصمیم گیری های پنهان براساس اطلاعات جهانی شبکه توسط هر حافظه مستقل ساخته می شود. در [11]، داده ها تکه ها براساس شماره تکه و برچسب CRS ذخیره می شوند. یک الگوریتم پیچیده طراحی شده است تا مجوز CR را برای ذخیره سازی موثر داده ها از تکه های داده طراحی کند. این روشهای توزیع شده ممکن است باعث پیچیدگی و سربار اضافی برای تبادل اطلاعات بین CR ها شود. علاوه بر این، تاخیر همگرایی برای روش توزیع وجود دارد. در [12]، چندین الگوریتم تکاملی غیر خطی متمرکز استفاده شده است. سپس یک مکانیزم تبلیغ و درخواست / پاسخ برای کشف محتوای ذخیره شده استفاده می شود. بر خلاف آنها، ما به طور کلی استراتژی ذخیره سازی و مسیریابی محتوا را بهینه می کنیم .

استراتژی های ذخیره سازی نیز می توانند در CDN و ذخیره سازی وب یافت شوند [19]. مدل های مختلف بر اساس محدودیت های مختلفی نظیر ظرفیت پیوند، ظرفیت ذخیره سازی و تقاضای درخواست پیش بینی شده برای به حداقل رساندن میانگین پرش متقابل، مصرف پهنای باند و غیره مورد مطالعه قرار گرفته است. با این حال، استراتژی ذخیره سازی و مسائل مسیریابی محتوا به طور مشترک برای توپولوژی های دلخواه شبکه بهینه نمی شود. بعلاوه، این آثار استراتژی ذخیره سازی مبتنی بر فایل هستند و برنامه نویسی شبکه را انجام نمی دهند.

امکان و مزایای استفاده از کدگذاری شبکه در ICN ها در [20] بیان شده است. این تمرکز بر یک رویکرد توزیع شده و برخی نتایج ارزیابی اولیه داده می شود. [21] ICN ها را با برنامه نویسی داخلی ساخته می شود که بر طرح امضا، طرح ارتباطی، مکانیابی حمل و نقل و مکانیسم تصدیق متمرکز می شود تا بهینه سازی حمل و نقل سود را بهینه سازد. در [22]، ما یک طرح جدید مسیریابی K-anycast جدید کش را پیشنهاد کردیم، که می تواند به طور قابل توجهی بهبود عملکرد ارسال محتوا را برای انتشار / ICNN مبتنی بر اشتراک ارائه دهد. در مقایسه با این آثار، چارچوب مبتنی بر SDN پیشنهادی انعطاف پذیر است و می تواند اطلاعات مربوط به همکاری را جمع آوری کند. همچنین می تواند پیکربندی ذخیره و سیاست مسیریابی در CRها را ممکن سازد. علاوه بر این، ما نیز از لحاظ تئوری مشکل مدیریت

حافظه پنهان در ICN ها را با LNC مدلسازی کرده و یک الگوریتم NCCM کارآمد مبتنی بر آرام سازی لاگرانژی برای بهینه سازی استراتژی ذخیره سازی و مسیریابی محتوا را به طور مشترک طراحی می کنیم.

3 .چارچوب مدیریت جدید کش برای ICN ها با LNC

در این بخش ابتدا ایده اصلی چارچوب مدیریت کش را معرفی می کنیم که بر اساس مفهوم SDN است. سپس چندین عملیات مهم را برای اجرای چارچوب پیشنهادی مورد بحث قرار می دهیم .

3.1. ایده اصلی چارچوب

در چارچوب ما، ICNرا بررسی می کنیم که شاملCRs ، یک کنترل کننده و سرورهای فعال LNC است را بررسی می کنیم، همانطور که در شکل 2 نشان داده شده است. توجه داشته باشید که هر روتر را می توان به عنوان یک CR در نظر گرفت ، حتی اگر دارای قابلیت حافظه پنهان نباشد. ما آن را به عنوان CR با ظرفیت کش صفر در نظر می گیریم. فرض می کنیم که محتوای F_n شامل چندین تکه داده است. ما عملکرد اصلی مربوط به مدیریت کش را برای ICNها بر اساس SDN به صورت زیر مطرح می کنیم .

3.1.1. عملکرد CR

در چارچوب ما، CRبرای عملکرد زیر مسئولیت خواهد داشت:

·نظارت بر محتویات دریافت شده از کاربران نهایی محلی خود در سطح محتوا (نه در سطح بسته) .

- ارسال آمار درخواست محتوا به صورت دوره ای به کنترل کننده .
- بازگشت بسته داده ها به صورت مستقیم برای کاربران نهایی در صورتی که بسته داده ها در حافظه داخلی آن موجود باشد .
- ارائه داده های دریافت شده برای کاربران نهایی .
- ارسال درخواست ها برای محتوای شناخته شده به سایر CR ها با توجه به جدول جریان .
- ارسال درخواست برای محتوای نامشخص به کنترل کننده .

• بازیابی تکه داده های کد شده دلخواه از سرورهای تعیین شده در صورت لزوم .

3.1.2. قابلیت های کنترل کننده

در طرح ما، کنترل کننده باید مسئول قابلیت ING پیگیری شود :

• جمع آوری آمار درخواست محتوا در سطح محتوا (نه در سطح تکه است) از هرCR .

• با تعیینQ k ، درخواست محتوای مشهور محلی برای هر CR v k تنظیم می شود. مسیرهای مربوط به محتوا در Q kباید قبل از زمان پیکربندی شوند .

• پیش بینی یک مجموعه محتوای محبوب، به عنوان F نشان داده شده، برای ذخیره سازی .

• استفاده از طرح مدیریت کش بهینه برای بهینه سازی استراتژی ذخیره سازی و مسیریابی محتوا .

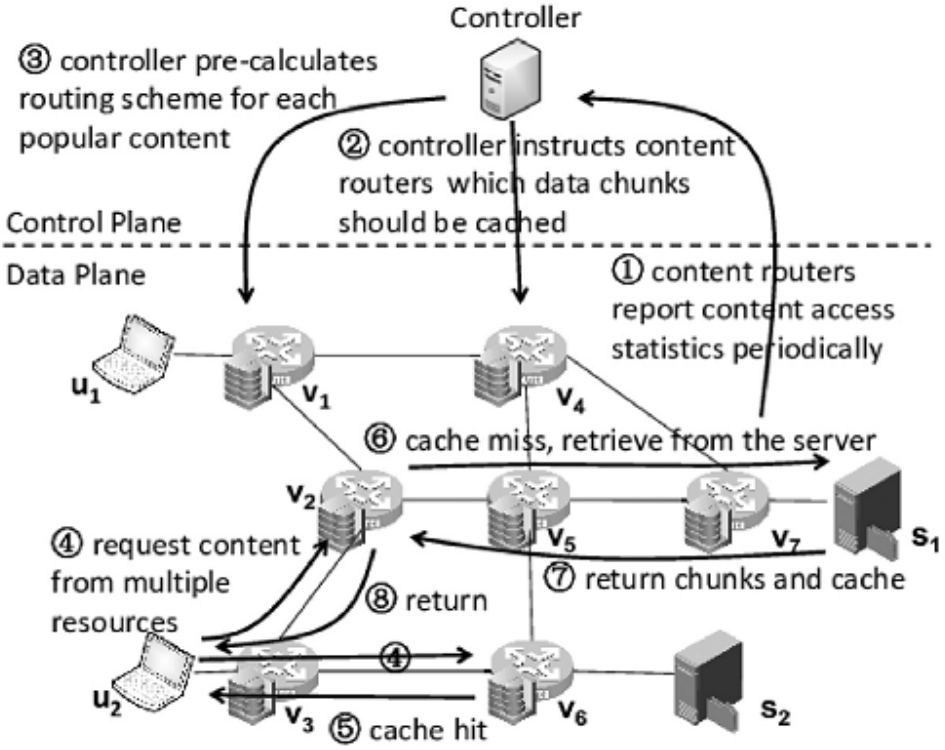
• پیکربندی CR ها برای ذخیره محتوای محبوب، درخواست مسیر محتوا و ارائه محتوا .

• پیکربندی CR ها به طوری که درخواست ها برای محتوای غیر محبوب را بتوان به سرورها ارسال کرد .

برای یک ICN در مقیاس بزرگ، عملکرد کنترل کننده همچنین می تواند توسط چندین کنترل کننده همکاری انجام شود، که هر کدام مسئولیت مدیریت CR را در حوزه ی خود خواهند داشت و اطلاعات را با یکدیگر تبادل می کنند .

3.2. عملیات اساسی

در این بخش ما عملیات اصلی مربوط به مدیریت کش را توضیح خواهیم داد، از جمله نحوه به دست آوردن آمار مربوط به محتوا، نحوه همکاری در ذخیره سازی محتوا در CR های مختلف، جایی که برای درخواست مسیرمحتوا است و اینکه ارائه محتوا به طور کامل از تمام مخازن در شبکه استفاده می کند.



شکل 2. چهارچوب مدیریت کش.

3.2.1 مجموعه آمار درخواست محتوا

برای به دست آوردن استراتژی ذخیره سازی مطلوب و تصمیم در مورد مسیریابی، تمام CR ها باید با یک دوره زمانی مشخص (مرحله 1 در شکل 2) آمار درخواست محتوا را گزارش دهند. در اینجا ما توجه می کنیم که با آمار آمار درخواست محتوا، کنترل کننده می تواند هر الگوریتم تجزیه و تحلیل داده ها را برای درک الگوی درخواست محتوا، مانند محبوبیت و موقعیت مکانی در طول زمان [23] اتخاذ کند. با این وجود، جزئیات مربوط به تجزیه و تحلیل آمار خارج از محدوده این مقاله است .

3.2.2. استراتژی ذخیره سازی

براساس آمار درخواست محتوا، کنترل کننده می تواند درخواست محتوای مشهور محلی (Q k) را برای هر CR v k و مجموعه ای از محتوای محبوب ($F,F=U_{VK}\ Q_K$) تعیین کند. برای هر محتوا درF ، کنترل کننده بیشتر مجموعه ای از داده ها را تعیین می کند که هر CR باید ذخیره کند (مرحله 2 در شکل 2). بدون استفاده از برنامه نویسی شبکه،

چنین طرح های ذخیره سازی دوره ای و غیر خطی در [12،24] مطرح شده اند. در عمل، دوره زمانی را می توان برای کاهش هزینه های پهنای باند و هزینه های مربوط به تعرفه / ارتباطات سیستم تعیین کرد.

برای اعمالLNC ، کنترل کننده ممکن است LNC قطعی یا LNC تصادفی را انتخاب کند. اگر LNC قطعی [25-28] استفاده شود، کنترل کننده باید مجموعه ای از *بردارهای رمزگذاری جهانی* (GEVs)[7] را به هر CR ارسال کند تا CR بتواند درخواست سرورهای فعال LNC را برای به دست آوردن بخشهای داده شده مورد نیاز رمزگذاری کند. در این مورد، کنترل کننده می تواند اطمینان دهد که هر مجموعه ای از داده های کد شده از هر محتوی ذخیره شده در ICNها با اندازه کمتر از اندازه محتوا به طور خطی وابسته نیست. از سوی دیگر، اگر LNC تصادفی [26-29] استفاده شود، کنترل کننده می تواند تعداد تکه های داده های کدگذاری شده لازم را به هر CR ارسال کند، که سپس درخواست را به سرورهای فعال LNC ارسال می کند که می تواند تکه های داده های کد شده را به GEVs تصادفی تعمیم دهد.در چنین مواردی، داده های رمزی شده از همان محتوا ممکن است به طور خطی با یک احتمال قطعی (بالا) از همدیگر جدا باشند. مزیت طرح دوم این است که سربار کردن محاسبات کنترل کننده را می توان در خطر سازش احتمالی استقلال خطی تکه های داده های کدگذاری کاهش داد.

3.2.3 .مسیریابی محتوا

در چارچوب ، محتویات درخواست شده توسط کاربران نهایی محلی را در هر CR به دو نوع، محتوای محبوب و غیر محبوب تقسیم می کنیم. برای هرk CR v ، کنترل کننده باید مسیر مطلوب را برای هر محتوای محبوب در Q k پیدا کند و جدول های جریان را در CR ها بر روی مسیر پیمایش کند. ورودی جریان برای یک محتوا F n یک لیست از واسط های خروجی داشته، که هر کدام به یک تکه داده های رمز شده f n از منتهی می شود. این مرحله (مرحله 3 در شکل 2) را می توان پس از رسیدن به استراتژی ذخیره سازی مطلوب توصیف شده در عملیات قبلی انجام داد.

در چارچوب ما، فقط لازم است کاربران نهایی محتویات داده Mn برای محتویات f n را دریافت کنند. آنها می توانند یک به یک بصورت عادی تکه داده ها را درخواست کنند. ما استراتژی ارسال ICN را کمی تغییر می دهیم (برای مثال،

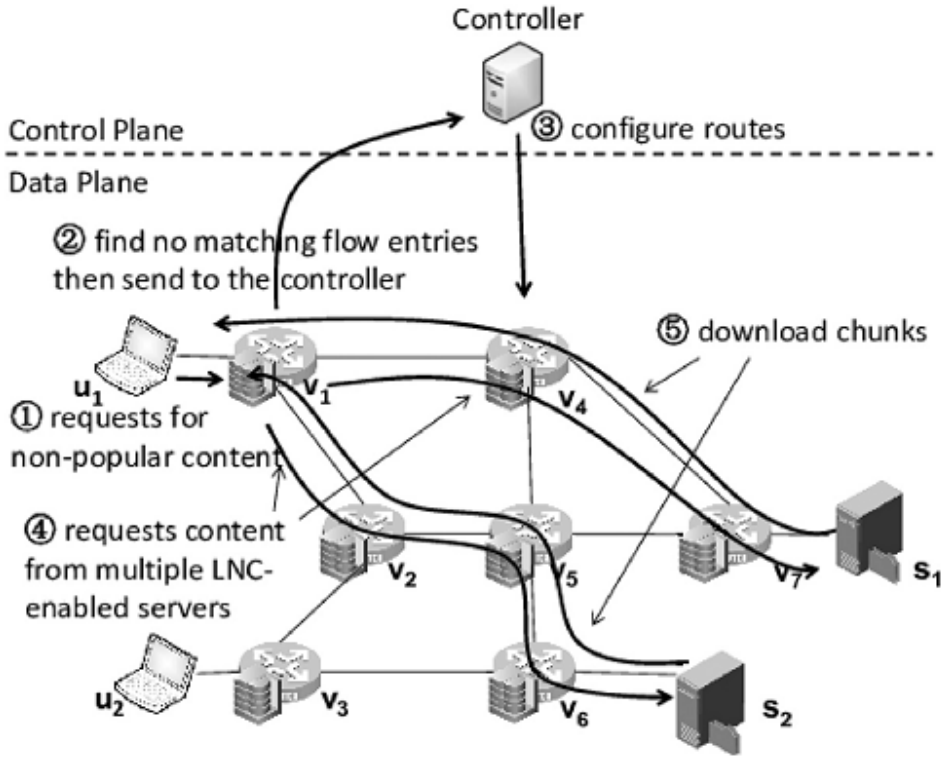
[7] Global encoding vectors

نام گذاری شبکه بندی داده ها (NDN)[8] تا بتوانیم با ورودی جریان روتر تعداد تکه های داده ای که می توان آن را بدست آورد و از هر یک از واسط های خروجی دریافت کرد را ثبت کنیم. . هنگامی که CR vk یک درخواست از کاربران نهایی محلی خود را برای یک بخش از محتوای Q k دریافت می کند، ابتدا بررسی می کند که آیا یک درخواست دیگر برای همان محتوا را با همان شماره توزیع ارسال کرده است و یا داده های بازگشت داده شده را دریافت نکرده است. اگر چنین است، این درخواست را ارسال نمی کند. اگر نه، CR درخواست را از طریق یک رابط خروجی ارسال می کند اگر تعداد تکه های داده حاصل از این رابط خروجی کمتر از تعداد تکه های داده است که می تواند از این رابط بدست آید (مرحله 4 در شکل 2).

اگر حافظه پنهان در یک CR به کار گرفته شود، CRیک تکه داده جدید رمز شده را با ترکیب تصادفی داده های تکراری کش شده از محتویات fn ایجاد می کندو آن را به عقب ارسال می کند (مرحله 5 در شکل 2). اگر حافظه پنهان در CR گم شود، CRباید داده های مورد نیاز را از یک یا چند سرور فعال با LNC (مرحله 6 در شکل 2) با توجه به جدول جریان بدست آورد. هنگامی که CR تکه های داده های کد شده را از سرورها دریافت می کند، تکه های داده کد شده را (گام 7 در شکل 2) ذخیره می کند و آنها را به کاربران نهایی (مرحله 8 در شکل 2) برمی گرداند .

برای درخواست به محتوای غیر محبوب که در Q k نیست، شکل 3 مراحل عملیات را نشان می دهد. به طور خاص، هنگامی که لبه CR v 1 یک درخواست دریافت می کند، ابتدا جدول جریان آن را بررسی می کند. اگر یک ورودی پیدا شود، درخواست متعاقبا فرستاده می شود. اگر یک ورودی برای محتوا خاص پیدا نشود، لبه CR فرمان درخواست را به کنترل کننده خواهد داد، همانطور که در مرحله 2 نشان داده شده است. کنترل کننده سپس یک طرح مسیر یابی مطلوب را تعیین می کند و تمام CR های متناظر را یاد آوری می کند که مرحله 3 است. بعد، می توان درخواست را برای چندین سرور متصل به LNC به طور همزمان ارسال کرد، همانطور که در مرحله 4 نشان داده شده است، برای به دست آوردن محتوا به طور موثر با استفاده ازLNC ، که مرحله 5 است.

[8] Named data networking



شکل 3. نمونه ای از درخواست محتوای غیر مشهور.

4 -فرمول بندی مدیریت بهینه کش برای ICN با LNC

در این بخش ابتدا مدل سیستم و فرضیه های مهم را بیان می کنیم. سپس در مورد فرمول بندی یک مشکل بهینه برای مدیریت حافظه پنهان در ICN ها با LNC توضیح می دهیم .

4.1. مدل سیستم

در این مقاله ICN به عنوان گراف هدایت شده $G =< V, E >$ را نشان می دهیم، جایی که V مجموعه ای از CR ها و E مجموعه ای از پیوندهای بین CR است .

4.1.1. نشانه گذاری ها

برای تسهیل بحث های بیشتر، نکته اصلی را که در مقاله به کار می رود به شرح زیر خلاصه می کنیم .

- V : مجموعه ای از CR ها $V = \{v_1, \ldots, v_{|V|}\}$.

- E : مجموعه ای از ارتباط بین CR ها. هر دو e i ، e j وj ، در E ،iff هستند، ارتباط بین CR v i و CR v j وجود دارد .
- c k ظرفیت ذخیره سازی CR v k از لحاظ داده ها0 ≥ c k ،. ∀ v k ∈ V.
- $C_{i,j}$ هزینه پیوند لینک i j ، i ∈ E از نظر تکه های داده C i ،. 0 ≥ j ، ∀ e i ∈ E، j.
- $C'_{k,n}$ هزینه cache زمانی که CR v k یک تکه داده از محتوا f n را ذخیره می کند .
- F: مجموعه ای از محتوای محبوب مورد نیاز برای ذخیره سازی، $F = \{f_1, \ldots, f_{|F|}\}$
- m n: اندازه محتوای f n از لحاظ داده ها، ∀ f n ∈ F است .
- S n: مجموعه ای از سرور ها برای محتوا f n ، ∀ f n ∈ F. .
- Q k: مجموعه ای از درخواست های محتوای محبوب محلی CR v k ، Q k ⊆F، ∀ v k ∈ V. .

4.1.2 فرضیه ها

در طرح مان، در نظر می گیریم که هر CR با (1) یک حافظه کش با ظرفیت مشخص همراه است و (2) مجموعه ای از درخواست ها (تولید شده توسط کاربران نهایی محلی). همانطور که قبلا توضیح داده شد، چنین تنظیم کلی نیز می تواند یک روتر بدون حافظه پنهان داشته باشد که ظرفیت حافظه آن صفر است. علاوه بر این، ما همچنین می توانیم از آن برای نشان دادن یک سرور استفاده کنیم که نه تنها همیشه محتوای مورد استفاده را ذخیره می کند، بلکه ممکن است دارای سطح مشخصی از ظرفیت حافظه پنهان برای ذخیره محتوایی باشد که آن انجام نمی دهد.

برای تسهیل پیش بینی محتوای محبوبF ، همانطور که در کار فعلی نشان داده شده است [30]، هر CR (که یک سوئیچ جریان باز است) می تواند یک شمارنده برای بازیابی آمار درخواست را حفظ کند و در نتیجه محبوبیت محتوا را می توان مستقیما از این آمار با استفاده از ماژول اندازه گیری در کنترل کننده استفاده کرد. کنترل کننده می تواند محتویات محبوب را به صورت زیر پیش بینی کند :

- هر CR vk به صورت دوره ای تعداد درخواست های تاریخی برای محتویات دریافت شده f_n از کاربران نهایی محلی خود را که به عنوان N k ، n خوانده می شود، شمارش می کند .

• هر CR vk به صورت دوره ای مقادیر N k ، nرا به کنترل کننده می فرستد .

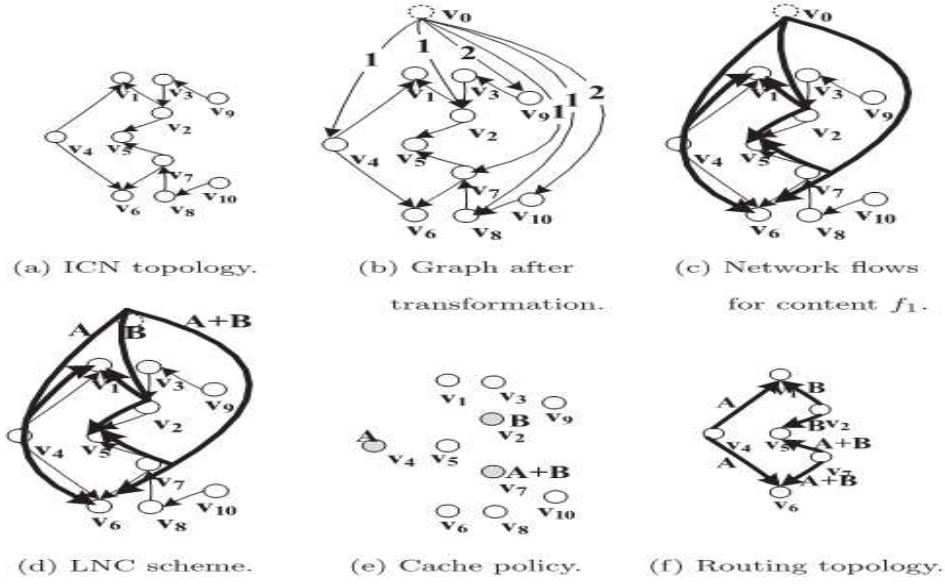
• کنترل کننده محبوبیت درخواستی محتوای f_n را محاسبه می کند که در آن N k ، nنشان دهنده fn در هر CR vkبه عنوان، $p_{k,n} = \frac{\dot{N}_{k,n}}{\sum_{v_k \in V, fn \in \bar{F}} N_{k,n}}$ تعداد درخواست های محتوا FN دریافت شده توسط CR vk از کاربران نهایی محلی آن را نشان می دهد و $\bar{F}$ مجموعه تشکیل شده از تمام محتواهای درخواستی از کاربران را نشان می دهد. بنابراین، ما $\sum_{v_k \in V, f_n \in \bar{F}} p_{k,n} = 1.$ داریم .

• *مجموعه درخواست های محتوی محلی محبوب* از هر $\{f_n | p_{k,n} \geq \overline{p}\}$است، که در آن p یک آستانه برای محبوبیت محتوا است.

• کنترل کننده محتوای مجموعه محبوب را پیش بینی می کند، به عنوان $F = \bigcup_{v_k \in V} Q_k$ مشخص می شود. فقط محتویات F در ICN ذخیره می شود .

از آنجا که هر دو آمار درخواست در هر CR و محاسبه مجموعه محتوای محبوب در کنترل کننده در سطح محتوا (نه در سطح ردیف) هستند، فرآیند فوق می تواند به طور موثر در شبکه اجرا شود. علاوه بر این، ما توجه می کنیم که کنترل کننده می تواند هر الگوریتم تجزیه و تحلیل داده ها را برای درک الگوی درخواست محتوا و تصمیم گیری در مورد مجموعه ای از محتویات که در ICN ذخیره می شود، اتخاذ کنند.

در طرح ما، SDN برای تسهیل مدیریت کش و مسیریابی محتوا مورد استفاده قرار می گیرد. در SDN، صفحه کنترل می تواند به صورت فیزیکی از صفحه داده جدا شود، که می توان برای دستیابی به تصمیم گیری در مورد ذخیره سازی و تصمیم گیری مسیریابی با استفاده از اطلاعات سطح بالا مانند ترافیک و آمار درخواست، مورد استفاده قرار داد. در نوشته ها، بسیاری از مطالعات موجود ICN بر اساس SDN برای ارائه قابلیت ICN با استفاده از سوئیچ جریان باز فعلی [15،30،31] اجرا شده است.



شکل 4. جریانات شبکه و روش ذخیره سازی در مسئله P.

در کنترل کننده، دو ماژول برای طرح پیشنهادی NCCM را می توان اجرا کرد. اولین قدم برای محاسبه محتوای محبوب F است. دیگر تصمیم گیری در مورد استراتژی ذخیره سازی و مسیریابی محتوا بر اساس وضعیت شبکه با توجه به الگوریتم NCCM است. برای هرCR ، نیاز به گزارش شمارش درخواست های تاریخی و دیگر وضعیت ها مانند ظرفیت حافظه پنهان موجود در کنترل کننده است .

هنگام استفاده از LNC در ICN ها، ما فرض می کنیم که هر محتوی در درجه اول به تکه های داده های اندازه ثابت تقسیم می شود. بعد، داده های کد شده از هر محتوی توسط سرورهای فعال LNC تولید می شود. در نتیجه، تکه های داده ذخیره شده در CR ها و انتقال در شبکه، ترکیب خطی داده های تکراری هستند. از آنجاییکه تکه های داده های مختلف کدگذاری مستقیما به صورت خطی مستقل یا مستقل با احتمال بالا هستند، کاربر فقط باید تعداد زیادی از داده های کدگذاری شده را از هر مجموعه ای از CR ها به دست آورد تا محتوای اصلی را بازیابی کند. برای تسهیل فرمول بندی بیشتر، ما فرض می کنیم که هر CR درخواست محتوا f n از طرف کاربران نهایی محلی می تواند تکه داده های اصلی را رمز گشایی و بازیابی کرده و آن را کمتر از m n داده های رمزی شده از محتویات محتوا f n دریافت نمی کند.

در نهایت، از آنجا که بیشتر ترافیک در شبکه متعلق به دانلود محتویات عمومی است، در این مقاله، ما تلاش می کنیم تا به طور بهینه تکه داده های کد شده محتوای محبوب در ICN را برای به حداقل رساندن هزینه های پهنای باند شبکه و هزینه های کش با همکاری بهینه سازی ذخیره سازی استراتژی و مسیریابی محتوا ذخیره کنیم. برای محتویات غیرمعمول، کنترل کننده یا لبهCR ، درخواستها را براساس چارچوب ما درخواست می کند .

4.2. فرمول بندی مشکل

بر اساس مدل چارچوب و سیستم پیشنهادی سابق ما، فرموله سازی *برنامه نویسی خطی عدد صحیح (ILP)* برای مشکل مدیریت بهینه حافظه را ارائه کرده ایم. به طور خاص، اهداف ILP این است که هزینه پهنای باند شبکه و هزینه کش را به حداقل برساند.

در چارچوب ما، CR v k می تواند محتوا را از چندین CR دریافت کند. چنین سناریویی نوعی ارتباط متداول برای CRاست. برای تدوین مشکل، می توانیم یک گره مجازی v 0 اضافه کنیم که دارای تمام داده های اصلی c است برای تمام قطعات محتوا در F قرار گرفته است. علاوه بر این، می توانیم یک لینک مجازی با ظرفیت محدود از v 0 به هر CR v d ∈ V اضافه کنیم. ما محدودیت بار ترافیکی را از این پیوند مجازی برابر باβd ، n، یعنی مقدار محدوده داده های کد شده از محتویات fn cached در CR vd را تنظیم می کنیم. گراف را به صورت $\overline{G} = < \overline{V}, \overline{E} >$ بعد از انتقال نشان می دهیم، که در آن $\overline{V} = V \cup \{v_0\}$ and $\overline{E} = \bigcup_{v_k \in V} \{e_{0,i}\} \cup E$ برای مورد CR vk بارهای داده ای از محتویات fn را از چندین CR در G بارگیری می کند، برابر با دانلود از v 0 در .G پس از چنین تحول، سناریو ارتباط چندین به یک معادل یک unicast از v 0 است به CR vk در .G .

شکل. 4 (a) و (b) نمودار اصلی G را برای مثال نشان داده شده در شکل 1 را نشان می دهد و گراف G با تبدیل گراف G را افزایش می دهد. به طور خاص، مقادیر نشان داده شده در لینک از گره مجازی V 0 به دیگر CR ها محدودیت بار ترافیک در لینک هستند. از آنجا کهCRs v 2 ، v 4، v 7و v 8 دارای ظرفیت ذخیره سازی واحد هستند، محدودیت بار ترافیک این لینک ها یکی است. از طرف دیگر، از آنجا که ما گره های سرور v 9 و v 10 را به عنوان CR ها که همیشه تمام داده های محتوا را در اختیار دارند، پردازش می کنیم، محدودیت بار ترافیکی این لینک

ها دو است. برای CRs v 1، v 3، v 5 و v 6 بدون ظرفیت ذخیره سازی موجود، محدودیت بار ترافیکی در این لینک ها صفر است و چنین پیوندهایی در شکل 4 (b) نشان داده نمی شود.

4.2.1 فرمول بندی مسئله برای به حداقل رساندن هزینه پهنای باند شبکه بر اساس یک استراتژی ذخیره سازی معین

ابتدا فرمول بندی مسئله را برای یک مورد ساده تر ارائه می کنیم که استراتژی ذخیره سازی تعیین کرده است. فرض کنید که مقادیر $\beta_{d,n}$ توسط یک سیاست ذخیره سازی داده شده تعیین شده است. هدف ما یافتن مسیریابی محتوا برای به حداقل رساندن هزینه پهنای باند شبکه است در حالی که نیازهای دانلود محتوای کاربران نهایی برآورده می شود. برای فرمول بندی مسئله، تعاریف زیر را ارائه می دهیم :

- $\theta_{k,n}$: مقدار محدوده داده های کد گذاری شده محتوا FN بارگیری شده توسط .CR v k
- $\beta_{d,n}$:مقدار داده های کد شده محتوای f_n ذخیره شده در v_d. $\beta_{d,n}$ یک عدد صحیح غیرمنفی است .
- $l_{k,n}^{i,j}$ بار ترافیکی در لینک $e_{i,j}$ برای CR vk که محتوای f_n را دانلود می کند.مخصوصاً $l_{k,n}^{0,j}$ مقدار بار ترافیکی را از CR v 0تا CR v j نشان می دهد، یعنی مقدار تکه های داده های کد شده از محتویات fn بارگیری شده توسط CR vkاز CR v j در عمل.
- $\ell_n^{i,j}$ مصرف پهنای باند ترافیک در لینک $e_{i,j}$ برای بارگیری محتوای f n .

توجه داشته باشید که جریان شبکه برای محتوا ف n به معنی جریان ترافیک برای بارگیری تکه داده های کد شده محتوای FN از CR های مختلف (به عنوان مثال، دانلود از گره مجازی0 V) به مقصد است. برای انتقال محتویات fn در $\overline{G}$، با محدودیت جریان شبکه (در اشکال (2) - (4) در زیر نشان داده شده است) در تک پخشی برای دانلود محتویات fn بین 0 v وvk ، بار ترافیکی انتقال در پیوند $e_{0,j}$ از0 v به vj در $\overline{G}$، یعنی $l_{k,n}^{0,j}$ معادل مقدار داده های کد شده محتوی fn است که از CR vj توسط CR vk در G دریافت می شود .

برای یک استراتژی ذخیره سازی داده شده، می توانیم برنامه ریزی خطی (LP) را به صورت زیر شرح دهیم :

$$\sum_{e_{i,j}\in E}\sum_{f_n\in F} C_{i,j}\ell_n^{i,j} \qquad (1)$$

حداقل:

به شرطی که: :

$$\sum_{v_j\in V} l_{k,n}^{0,j} = \theta_{k,n}, \forall v_k \in V, \forall f_n \in Q_k \qquad (2)$$

$$\sum_{j:e_{i,j}\in \overline{E}} l_{k,n}^{i,j} = \sum_{j:e_{j,i}\in \overline{E}} l_{k,n}^{j,i}, \forall v_k \in V, v_i \in \overline{V} - \{v_k, v_0\}, \forall f_n \in Q_k \qquad (3)$$

$$\sum_{j:e_{j,k}\in \overline{E}} l_{k,n}^{j,k} = \theta_{k,n}, \forall v_k \in V, \forall f_n \in Q_k \qquad (4)$$

$$l_{k,n}^{i,j} \le \ell_n^{i,j}, \forall v_k \in V, \forall e_{i,j} \in E, \forall f_n \in Q_k \qquad (5)$$

$$\theta_{k,n} \ge m_n, \forall v_k \in V, \forall f_n \in Q_k \qquad (6)$$

$$l_{k,n}^{0,d} \le \beta_{d,n}, \forall v_k, v_d \in V, \forall f_n \in F \qquad (7)$$

$$0 \le l_{k,n}^{i,j}, \forall v_k \in V, \forall e_{i,j} \in \overline{E}, \forall f_n \in F \qquad (8)$$

در فرمول بالا، هدف این است که هزینه پهنای باند را به حداقل برسانیم. محدودیت های معادله (2) - (4) محدودیت جریان شبکه در هر CR برای هر جلسه دانلود محتوا را نشان می دهد .

باLNC ، اگر CR های مختلف تکه داده های مختلف کدگذاری شده از همان محتویات fn را دانلود کند، و تکه داده های دانلود شده باید از همان لینک $e_{i,j}$عبورکنند، سپس بار ترافیکی را می توان با تولید و انتقال ترکیب ترکیب خطی تصادفی تکه های داده بارگیری شده به اشتراک گذاشت . علاوه بر این، تکه های داده های جدید تولید شده کدگذاری شده از طریق لینک $e_{i,j}$ برای افزودن درجه جدید آزادی با احتمال بالا برای بازبینی تمامی تکه های داده محتوا fn در هر CR درخواست آن مفید است . بنابراین، فقط لازم است $\max_{v_k\in V} l_{k,n}^{i,j}$ تکه داده ها از طریق$e_{i,j}$ منتقل شوند،یعنی کل مصرف پهنای باند شبکه از ترافیک در لینک $e_{i,j}$ برای دانلود f_n $\max_{v_k\in V} l_{k,n}^{i,j}$ است. از سوی دیگر، محدودیت در معادله. (5) برابر است با:

$$\max_{v_k \in V} l_{k,n}^{i,j} \leq \ell_n^{i,j}, \forall e_{i,j} \in E, \forall f_n \in Q_k.$$

از آنجا که هدف ما به حداقل رساندن $\sum_{e_{i,j} \in E} \sum_{f_n \in F} C_{i,j} \ell_n^{i,j}$ است، ما داریم $\ell_n^{i,j} = \max_{v_k \in V} l_{k,n}^{i,j}$. بنابراین، هدف $\sum_{e_{i,j} \in E} \sum_{f_n \in F} C_{i,j} \ell_n^{i,j}$ به معنی هزینه کل پهنای باند شبکه است .

محدودیت ها در معادله (6) و (7) محدودیت دانلود محتوا را ارائه می دهند. به طور خاص، محدودیت در معادله (6) بدین معناست که هر CR نیاز به بارگیری تعداد کافی از تکه های داده های کد شده دارد تا کد های داده های اصلی را برای هر یک از مواد درخواست شده بازنویسی و بازیابی کند. محدودیت در معادله (7) نشان می دهد که مقدار تکه های داده های کد شده برای هر محتوایی که از CR دریافت می شود، بیشتر از مقدار داده های محتوا ذخیره شده در CRنیست. محدودیت در معادله (8) محدوده ارزش متغیرها را اجرا می کند.

پس از حل مسئله بالا، برای هر محتوی fn و هر CR $v_k \in V$ درخواست $f_n \in F$، f_n می شود، یک جریان یکپارچه را با ظرفیت جریان کمتر از m_n بین گره v_0 و CR v_k بدست می آوریم. بنابراین، یک چند پخشی با LNC را می توانبدست آورد تا اطمینان حاصل کرد که هر CR درخواست محتوا می تواند رمزگشایی و بازیابی تکه اولیه داده های m_n را رمزگشایی و بازیابی کند [32].

4.2.2 یک فرمول ILP برای به حداقل رساندن هزینه پهنای باند و هزینه نگهداری در ICN با NC

هنگامی که به طور مشترک استراتژی ذخیره سازی و مسیریابی محتوا را برای به حداقل رساندن هزینه های (1)پهنای باند و هزینه حافظه پنهان بررسی می کنیم، پارامتر $\beta_{d,n}, \forall v_d \in V, f_n \in F$ که نشان دهنده مقدار داده های کد گذاری شده از محتویات در CR v_d محسوب می شود یک متغیر عدد صحیح نامنفی است .

(2) در زیر، ما می توانیم مشکل را به عنوان برنامه ریزی خطی عدد صحیح (ILP) تشکیل دهیم:

$$\mathbf{P}: \quad \textbf{Minimize:} \sum_{e_{i,j} \in E} \sum_{f_n \in F} C_{i,j} \ell_n^{i,j} + \sum_{v_d \in V} \sum_{f_n \in F} C'_{d,n} \beta_{d,n} \tag{3}$$

به شرطی که:

(4) $Eq.(2) - Ineq.(8)$

$$\beta_{d,n} \leq m_n, \forall v_d \in V - S_n, \forall f_n \in F \quad (9)$$

(5)
$$\beta_{d,n} = m_n, \forall v_d \in S_n, \forall f_n \in F \quad (10)$$

(6)
$$\sum_{f_n \in F} \beta_{d,n} \leq c_d, \forall v_d \in V \quad (11)$$

(7)
$$\beta_{d,n} \in \mathbb{N}, \forall v_d \in V, \forall f_n \in F \quad (12)$$

قسمت اول هدف، هزینه پهنای باند را نشان می دهد و بخش دوم نشان دهنده هزینه کش است. محدودیت در معادله (9) محدودیت ظرفیت حافظه پنهان برای هر محتوا در هرCR است، که نشان دهنده تعداد تکه های داده ذخیره شده توسط هر CR بیشترین مقدار فایل است. محدودیت در معادله (10) نشان می دهد که سرورها برای هر محتوا همیشه دارای تمام تکه داده های اصلی برای این محتوا هستند. محدودیت در معادله (11) محدودیتی را به وجود می آورد که مقدار داده های موجود ذخیره شده در هر CR از ظرفیت ذخیره سازی آن فراتر نمی رود. محدودیت ها در معادله (12) محدوده ارزش متغیرها را اعمال می کند .

در طرح LNC بدست آمده، تعداد تکه داده های کد شده در لینک $e_{0,j},\ \forall v_j \in V$ در $\overline{G}$ ارسال شده، تعداد تکه های داده کد شده ای هستند که در CR_{vj} در G ذخیره شده اند.در معماری ما، اگر کنترل کننده LNC قطعی را برای پیاده سازی سیاست ذخیره سازی انتخاب کند، باید مجموعه ای از *بردارهای رمزگذاری جهانی* [25] (GEVs) را به هر CR ارسال کند، که نشان می دهد داده های کدگذاری شده CR باید حافظه پنهان داشته باشد. به عبارت دیگر، اگر کنترل کننده LNC تصادفی را انتخاب کند، تنها تعداد تکه داده های کدگذاری شده را به هر CR ارسال می کند. برای هر $CR_{vi} \in V$، مقدار داده های محدوده محتوا f_n منتقل شده به مجاور CR_{vj} است .اگر $\ell_n^{i,j}$ کمتر از مقدار داده های دریافت شده از محتویات محتوی f_n است، پس آن $\ell_n^{i,j}$ داده های کد شده محتوای f_n با ترکیب خطی تصادفی تکه داده های دریافت شده از محتویاتf_n را افزایش می دهد و آنها را به v_j منتقل می کند. در غیر این صورت، آن می تواند به سادگی $\ell_n^{i,j}$ تکه داده های کد شده را ارسال کند که برای CR_{vj} دریافت می کند.

شکل 4 (c) جریان شبکه یکپارچه را بین v_0 تا هر CR_{v1} ، v_5و v_6 برای محتوای f_1 نشان می دهد. به طور خاص، ترافیک در لینک $e_{0,2}$ به CRs_{v1} و v_5 هدف قرار گرفته است؛ ترافیک در لینک $e_{0,7}$در مورد CRs_{v1} و v_6 توزیع شده است و ترافیک در لینک $e_{0,7}$به CRs_{v5} و v_6 تقسیم می شود. با توجه به شکل 4(c) ، برای سه گره با ظرفیت موجود، ما $\beta_{2,1}=1, \beta_{4,1}=1$ و $\beta_{7,1}=1$ را داریم، به این معنی که هر CRs_{v2v4} of و v_7 یک تکه داده داده شده را برای محتویات f_1 ذخیره می کند. بر اساس الگوریتم LNC نشان داده شده در شکل 4 (d) ما می توانیم سیاست ذخیره سازی بهینه را بدست آوریم که تکه داده های $A + B$ باید در CR_{v7} ذخیره شوند، داده ها باید A در CR_{v4} ذخیره شوند و اطلاعات B باید در CR_{v2} ذخیره شود، که در شکل 4 (e) نشان داده شده است. علاوه بر این، توپولوژی مسیریابی نشان داده شده در شکل 4 (f) همچنین می تواند با توجه به شکل 4 (c) با حذف گره مجازی v_0 و پیوندهایی از v_0 به دیگر CR ها لازم شود .

5. الگوریتم مدیریت کش مبتنی بر کد گذاری موثر شبکه

در این بخش، ما یک الگوریتم *مدیریت رمزنگاری مبتنی بر کدگذاری شبکه* (NCCM) را برای به دست آوردن راه حل تقریبا مطلوب ILP **P**در پیچیدگی محاسباتی چندجمل های طراحی می کنیم. بر اساس الگوریتم آرامش لاگرانژی و الگوریتم زیر گرادیانی، الگوریتم NCCM را می توان به طور موثر اجرا کرد.

5.1 .مشکل دوگانه لاگرانژی

ما محدودیت در معادله را (7) برای به دست آوردن مشکل دوگانه لاگرانژی به عنوان مشکل اصلی که می تواند به دو زیر پروتکل تقسیم شود کم می کنیم، که هر کدام می توانند با پیچیدگی محاسباتی چند جمله ای حل شوند .

به طور خاص، ابتدا محدودیت در معادله (7) را با حرکت آن به تابع هدف با ضرایب لاگرانژی مرتبط $\lambda_{k,d,n} \geq 0, \forall v_k, v_d \in V, f_n \in F$ را کم می کنیم. ما تابع هدف ILP **P** را به صورت زیر مجددا فرمول بندی می کنیم:

$$\sum_{e_{i,j}\in E}\sum_{f_n\in F} C_{i,j}\ell_n^{i,j} + \sum_{v_d\in V}\sum_{f_n\in F} C'_{d,n}\beta_{d,n} + \sum_{v_k,v_d\in V}\sum_{f_n\in F}\lambda_{k,d,n}(l_{k,n}^{0,d}-\beta_{d,n})$$

$$=\sum_{f_n\in F}\left(\sum_{e_{i,j}\in E} C_{i,j}\ell_n^{i,j}+\sum_{v_k,v_d\in V}\lambda_{k,d,n}l_{k,n}^{0,d}\right)-\sum_{v_d\in V}\sum_{f_n\in F}\left(\sum_{v_k\in V}\lambda_{k,d,n}-C'_{d,n}\right)\beta_{d,n}.$$

اجازه دهید λ نشان دهنده بردار تشکیل شده از عناصر $\lambda_{k,d,n}, \forall v_k, v_d \in V, f_n \in F$ باشد. سپس، مسئله دوگانه لاگرانژی **P** ILP :

$$\max_{\boldsymbol{\lambda}>\mathbf{0}} L(\boldsymbol{\lambda}), \tag{13}$$

که در آن

$$\mathbf{P}^{\boldsymbol{\lambda}}: \; L(\boldsymbol{\lambda}) = \min \sum_{f_n \in F} \left(\sum_{e_{i,j} \in E} C_{i,j} \ell_n^{i,j} + \sum_{v_k, v_d \in V} \lambda_{k,d,n} l_{k,n}^{0,d} \right) - \sum_{v_d \in V} \sum_{f_n \in F} \left(\sum_{v_k \in V} \lambda_{k,d,n} - C'_{d,n} \right) \beta_{d,n}$$

تحت محدودیت در معادلات (2) - (6) و معادلات (8)-(12) .

بدیهی است، محدودیت در معادلات (2) - (6) و معادله (8) تنها به گروه متغیرهای $\{l_{k,n}^{i,j}, \ell_n^{i,j}, \theta_{k,n}\}$ مرتبط است و معادلات (9) - (12) تنها با گروه متغیرهای $\{\beta_{d,n}\}$ ارتباط دارد. بنابراین، مشکل $\mathbf{P}^{\lambda}$ را می توان به دو زیر مشکلی تجزیه کرد، $\mathbf{P}_1^{\boldsymbol{\lambda}}$ و $\mathbf{P}_2^{\boldsymbol{\lambda}}$ به شرح زیر است :

$$\mathbf{P}_1^{\boldsymbol{\lambda}}: \; \text{Minimize}: \sum_{f_n \in F} \left(\sum_{e_{i,j} \in E} C_{i,j} \ell_n^{i,j} + \sum_{v_k, v_d \in V} \lambda_{k,d,n} l_{k,n}^{0,d} \right)$$

تحت محدودیت در معادلات (2) - (6) و معادله (8). و

$$\mathbf{P}_2^{\boldsymbol{\lambda}}: \; \text{Minimize}: \; - \sum_{v_d \in V} \sum_{f_n \in F} \left(\sum_{v_k \in V} \lambda_{k,d,n} - C'_{d,n} \right) \beta_{d,n}$$

تحت محدودیت در معادلات (9) .(12).

اولا، مشکل $\mathbf{P}_1^{\boldsymbol{\lambda}}$ یک مشکل برنامه نویسی خطی (LP) است که می تواند به طور موثر حل شود. برای مشکل $\mathbf{P}_2^{\boldsymbol{\lambda}}$ ، ما همچنین می توانیم یک الگوریتم حریص ساده برای به دست آوردن راه حل بهینه آن، که در بخش 5.2 نشان داده شده است، طراحی کنیم.

با توجه به**λ** ، B^{λ} ارزش هدف را برای مشکل P^{λ} نشان می دهد و این یک مرز پایین برای مشکل [33] **P** است B^{λ} . را می توان با استفاده از مقادیر اهداف برای مشکل P و P λ2 بدست آورد، به ترتیب به عنوان $B^{\lambda}{}_1$ و $B\ \lambda_2$ تعریف می شود. بنابراین، $B^{\lambda} = \dot{B}_1^{\lambda} + B_2^{\lambda}$.

5.2.الگوریتم بهینه برای مشکل P_2^{λ}

برای داده**λ** ، مشکل P_2^{λ} به صورت زیر می تواند به طور مطلوب حل شود. ما نشان می دهیم $\bar{\lambda}_{d,n}^{*} = \sum_{v_k \in V} \lambda_{k,d,n}$. ما داریم :

$$-\mathbf{P}_2^{\boldsymbol{\lambda}}: \qquad \text{Maximize}: \sum_{v_d \in V} \sum_{f_n \in F} \left(\lambda_{d,n}^{*} - C'_{d,n}\right) \beta_{d,n} \qquad (14)$$

به شرط: محدودیت در معادلات (9) .(12).

توجه داشته باشید که راه حل مشکل $-\mathbf{P}_2^{\boldsymbol{\lambda}}$ معادل با حل مشکل $P\ \lambda_2$ است. با این حال، ارزش هدف $-B_2^{\boldsymbol{\lambda}}$ -P λ_2 است .مشکل $-\mathbf{P}_2^{\boldsymbol{\lambda}}$ می تواند بیشتر به | V | زیر مشکلات تقسیم شود. برای هر $CR_{v\,d} \in V$ ، فقط باید مشکل زیر را حل کنیم .

$$-\mathbf{P}_{2,d}^{\boldsymbol{\lambda}}: \qquad \text{Maximize}: \sum_{f_n \in F} (\lambda_{d,n}^{*} - C'_{d,n}) \beta_{d,n} \qquad (15)$$

به شرطی که:

$$\beta_{d,n} \le m_n, \forall f_n \in F, \text{ if } v_d \in V - S_n$$
$$\beta_{d,n} = m_n, \forall f_n \in F, \text{ if } v_d \in S_n$$
$$\sum_{f_n \in F} \beta_{d,n} \le c_d$$
$$\beta_{d,n} \in \mathbb{N}, \forall f_n \in F$$

مشکل فوق یک مشکل بزرگ سازی وزن با محدودیت های ظرفیت است. الگوریتم بهینه برای حل مسئله در الگوریتم 1 نشان داده شده است. ایده اصلی این است که ابتدا مجموعه ای از محتوای محبوب در ترتیب کاهش وزن شان را $\lambda_{d,n}^{*} - C'_{d,n}$ مرتب کنیم. سپس، CR_{vd} به عنوان تعداد زیادی از داده ها برای هر قطعه محتوا در مجموعه ای مرتب

شده به صورت متوالی ذخیره می شود. فرض کنید $B^{\lambda}_{2,d}$ مقدار اهداف برای مشکل $-\bar{\mathbf{P}}^{\lambda}_{2,d}$ را نشان می دهد. سپس، $B^{\lambda}_{2} = -\sum_{v_d \in V} B^{\lambda}_{2,d}$.

5.3 انتخاب ضرب کننده ها

برای پیدا کردن یک مرز خوب پایین، انتخاب ضریب بردار λ مهم است. ما از بهینه سازی زیر گرادیانی برای انتخاب تکراری λ استفاده می کنیم. در تکرار t ، بردار زیر گرادیان γ^t توسط $\bar{\gamma^t_{k,d,n}} = l^{0,d,t}_{k,n} - \beta^t_{d,n}, \forall v_k, v_d \in V, \bar{\forall} f_n \in F,$ محاسبه می شود که در آن $l^{0,d,t}_{k,n}$ و $\beta^t_{d,n}$ مقادیر متغیرهای $l^{0,d}_{k,n}$ و $\beta_{d,n}$ را در حل مطلوب مسئله $\mathbf{P}^{\lambda}$ نشان می دهد که در تکرار t حاصل شده است. γ^t نشان دهنده برداری است که توسط عناصر $\gamma^t_{k,d,n}, \forall v_k, v_d \in V, f_n \in \bar{F},$ تعریف شده است .

بردار ضرب کننده λ^{t+1} مورد استفاده در تکرار t + 1 را می توان توسط $\boldsymbol{\lambda}^{t+1} = \max\{\boldsymbol{\lambda}^t + s_t \boldsymbol{\gamma}^t, 0\}$ بدست آورد که در آن λ^t یک بردار چندتایی است که در تکرار $t\text{-th}$ بکار می رود و s_t یک اندازه گام مثبت است . s_t را می توان با روش معمولی [33] بدست آورد : $s_t = \frac{\eta_t (z^t_{UP} - z^t_{LB})}{\|\boldsymbol{\gamma}^t\|^2}$، که در آن $0 \le \eta_t \le 2$ and $z_{UP}{}^t$ (resp. $z_{LB}{}^t$) نشان دهنده یک حد بالا (یا پایین تر) بر روی هدف مطلوب مسئله **P** در تکرار t-th است. به ویژه،

$$\eta_t = \begin{cases} 2, & \text{if } t = 1 \\ \eta_{t-1}, & \text{if } z^t_{LB} \ge z^{t-1}_{LB} \\ \eta_{t-1}/2, & \text{if } z^t_{LB} = z^{t-1}_{LB} = z^{t-2}_{LB} \end{cases} . \qquad (16)$$

اجازه دهید β^t بردار تشکیل شده توسط عناصر $\beta^t_{d,n}, \forall v_d \in V, \forall f_n \in F$ باشد. برای به دست آوردن یک راه حل عملی برای مشکل **P** در تکرار t ، ما اجازه می دهیم β^t پارامترهای شناخته شده و حل مشکل **P** با محدودیت در معادلات (2) - (7) و معادله (8) باشند. از آنجائیکه مقادیر βt محدودیت های معادلات (9) - (12) را رفع می کنند، مقدار به دست آمده برای مشکل **P** که به صورت $B^{\lambda^t}(\boldsymbol{\beta}^t)$ نشان داده شده است مرز بالایی مسئله اصلی **p** بدست آمده در تکرار t است.علاوه بر این، راه حل به دست آمده، بدیهی است که یک راه حل قابل حل برای مشکل **P** باشد.

Algorithm 1 The greedy algorithm for optimally solving problem $\mathbf{P}_{2,d}^{\boldsymbol{\lambda}}$.

For a given node $v_d \in V$ and multiplier vector $\boldsymbol{\lambda}$, compute $\lambda_{d,n}^* = \sum_{v_k \in V} \lambda_{k,d,n}$;
Sort $\lambda_{d,n}^* - C'_{d,n}$ in decreasing order. Suppose that the i^{th} largest value is $\lambda_{d,n_i}^* - C'_{d,n_i}$;
Let $\beta_{d,n} = 0, \forall f_n \in F$ and $C = c_d$;
if $\lambda_{d,n_1}^* - C'_{d,n_1} < 0$ **then**
  for $i = 1$ to $|F|$ **do**
    if $v_d \in S_i$ **then**
      $\beta_{d,i} = m_i$;
      $C = C - m_i$;
    end if
  end for
else
  for $i = 1$ to $|F|$ **do**
    if $v_d \in S_i$ **then**
      $\beta_{d,i} = m_i$;
      $C = C - m_i$;
    end if
  end for
  $j = 1$;
  while $j \leq |F|$ and $C \geq 1$ and $\lambda_{d,n_j}^* - C'_{d,n_j} \geq 0$ **do**
    if $v_d \notin S_{n_j}$ **then**
      Cache $\beta_{d,n_j} = \min(C, m_{n_j})$ data chunks of content f_{n_j};
      $C = C - \beta_{d,n_j}$;
    end if
    $j = j + 1$;
  end while
end if
return $B_{2,d}^{\boldsymbol{\lambda}}$ and $\beta_{d,n}, \forall f_n \in F$;

5.4 الگوریتم NCCM بر اساس آرامش لاگرانژی

در حال حاضر الگوریتم NCCM که براساس آرام سازی لاگرانژی برای حل مسئله **P** طراحی شده است را توصیف می کنیم. به طور خاص، زیرمشکلات P^{λ}_1 و P^{λ}_2 با بردار چندتایی λ^t در تکرار t حل می شود. در هر تکرار t، می توان یک راه حل امکان پذیر برای مسئله P بر اساس βt و مرز بالایی از مسئله اصلی P بدست آورد. ما مرز بالایی z^t_{UP}

را به عنوان کوچکترین مرز بالایی نگه می داریم که در تکرارهای t به دست آمده است. از سوی دیگر، z^t_{LB} حداکثر مقدار هدف مسئله P^{λ} را پس از تکرار t نشان می دهد، که مرز پایین مسئله **P** است.

Algorithm 2 Network coding based cache management (NCCM) algorithm.

1: $t = 1$ and $t' = 0$; $z^0_{UP} = +\infty$ and $z^0_{LB} = -\infty$; $\eta_1 = 2$;
2: Let $\lambda^1_{k,d,n} = 10^{-5}, \forall v_k, v_d \in V, f_n \in F$ and $\epsilon^1 = +\infty$;
3: **while** $t < T$ and $\epsilon^t > \epsilon^*$ and $t' < T'$ **do**
4: Solve problem $\mathbf{P}^{\lambda}_1$; Obtain $B^{\lambda^t}_1$ and $l^{0,d,t}_{k,n}, \forall v_k, v_d \in V, \forall f_n \in F$;
5: Solve problem $\mathbf{P}^{\lambda}_2$; Obtain $B^{\lambda^t}_2$ and $\beta^t_{d,n}, \forall v_d \in V, \forall f_n \in F$;
6: $B^{\lambda^t} = B^{\lambda^t}_1 + B^{\lambda^t}_2$;
7: Let $z^t_{UP} = \min(B^{\lambda^t}(\boldsymbol{\beta}^t), z^{t-1}_{UP})$;
8: **if** $z^t_{UP} < z^{t-1}_{UP}$ **then**
9: Let π^* be the feasible solution of problem **P** obtained at the tth iteration in which we let $\boldsymbol{\beta}^t$ be given parameters and solve problem **P** with constraints Eq. (2)– Ineq. (7) and Ineq. (8);
10: **end if**
11: Let $z^t_{LB} = \max(B^{\lambda^t}, z^{t-1}_{LB})$;
12: **if** $z^t_{LB} > z^{t-1}_{LB}$ **then**
13: $t' = 0$;
14: **else**
15: $t' = t' + 1$;
16: **end if**
17: **if** $t' \geq 3$ **then**
18: $\eta_{t+1} = \eta_t / 2$;
19: **end if**
20: $\epsilon^{t+1} = z^t_{UP} - z^t_{LB}$;
21: Update Lagrangian multiplier vector $\boldsymbol{\lambda}^{t+1}$ according to

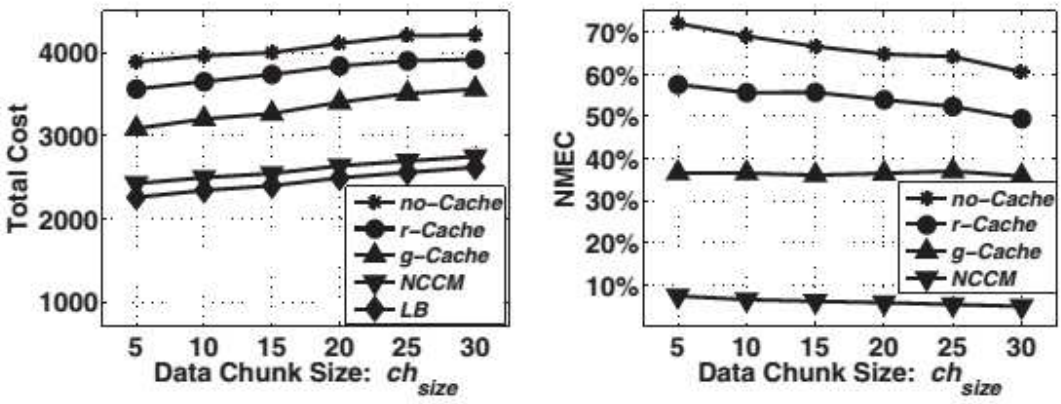
$$\lambda^{t+1}_{k,d,n} = \max\{\lambda^t_{k,d,n} + s_t \gamma^t_{k,d,n}, 0\}, \forall v_k, v_d \in V,$$

$\forall f_n \in F$, where
22: $t = t + 1$; $\gamma^t_{k,d,n} = l^{0,d,t}_{k,n} - \beta^t_{d,n}$ and $s_t = \dfrac{\eta_t (z^t_{UP} - z^t_{LB})}{\|\boldsymbol{\gamma}^t\|^2}$;
23: **end while**
24: **return** z^{t-1}_{UP} and π^*;

الگوریتم NCCM نشان داده شده در الگوریتم 2 هنگامی که یکی از شرایط برآورده شود متوقف می شود: (1) تعداد تکرار t به حد تکرار T می رسد؛ (2) تفاوت بین z^t_{LB} و z^t_{UP} کمتر از آستانه ϵ^* است ؛ (3) مرز پایین برای بیش از تعدادی تکرار T افزایش نمی یابد. پس از اینکه الگوریتم خاتمه یافت، به راه حل عملی π^* باز می گردد که در طی تکرار به مرز کمترین حد می رسد .

6. نتایج عددی

در این بخش، آزمایشهای شبیه سازی برای ارزیابی عملکرد الگوریتم NCCM انجام می شود. به طور خاص، ما عملکرد الگوریتم NCCM را با مرز پایین و سه مرز بالایی مقایسه می کنیم. مرز پایینی، به عنوان LB نشان داده شده، نشان دهنده یک راه حل بهینه است که می تواند به راحتی با کاهش محدودیت عدد صحیح در معادله (12) در **P** ILP بدست آید. برای اولین مرز بالا، ما یک شبکه سنتی را بدون استفاده از برنامه نویسی کش و شبکه در نظر می گیریم، جایی که داده ها فقط در سرورهای اصلی خود در دسترس هستند. ما عملکرد را در این مورد به عنوان مرز فوقانی، *بدون حافظه* در نظر گرفته ایم. دو مرز بالایی دیگر توسط (1) تکه های تصادفی ذخیره شده بر روی تمام CR ها به دست می آیند و (2) ذخیره حریصانه محتویات بصورت داخلی که برای هر CR لازم است (مانند روش جایگزینی کش LFU در ICN های سنتی بدون LNC) . در این موارد، اگر چه تمام مخازن در شبکه استفاده می شود، پتانسیل مدیریت حافظه پنهان و برنامه نویسی شبکه بر هزینه پهنای باند و هزینه ذخیره سازی به طور کامل در نظر گرفته نمی شود. ما آنها را به ترتیب *r-cache* و *g-cache* نامگذاری می کنیم.



(a) Total cost. (b) NMEC.

شکل 5. هزینه در مقابل اندازه بسته داده ها.

در آزمایشات ما، شبکه G به طور تصادفی با استفاده از یک روش به طور گسترده مورد استفاده توسط واکس من [34]تولید می شود. این روش بر محل پیوندها تاکید می کند، که یکی از ویژگی های مهم بسیاری از شبکه های بصورت فیزیکی جاسازی شده و شبکه های پوشش داده شده مانند شبکه های همتا (همراه به همراه) می باشد .

به طور خاص، موقعیت گرهها، یعنیCRs ، در G یک فرآیند پوآسون در صفحه با اندازه گیری میانگین مقیاس پذیر شده Lebesgue است، که η نشان دهنده شدت فرآیند پوآسون است. دو v_i (CR) و v_j با احتمال $P(v_i, v_j) = \alpha e^{-d(v_i,v_j)/(\mu * L)}$ مرتبط است جایی که $\alpha > 0, \mu \leq 1, d(v_i, v_j)$ فاصله اقلیدس بین vi و vj است و L حداکثر فاصله بین هر دو CR است. اجازه دهید منطقه گرافیت شبکه تصادفی G 20×20 باشد. در آزمایشات نشان داده شده در شکل های 5-13، پارامترهای شبکه برای مدل شبکه واسمن را به شرح زیر تنظیم شده اند: شدت فرایند پواسون $\alpha \in [0.5, 1.5]$ and $\mu = 0.7, \eta \in [0.01, 0.125]$.

علاوه بر پارامترهای شبکه، ما شش پارامتر دیگر را در شبیه سازی های مان داریم .

- ch_{size}: اندازه شکاف داده، $ch_{size} \in [5\ \text{Mb}, 30\ \text{Mb}]$.
- ca_{size} : ظرفیت حافظه پنهان، $ca_{size} \in [50\ \text{Mb}, 300\ \text{Mb}]$.
- co_{size}: اندازه محتوا، $co_{size} \in [200\ \text{Mb}, 450\ \text{Mb}]$.
- | F |: اندازه مجموعه محتوای محبوب، $|F| \in [10, 35]$.
- q: اندازه مجموعه درخواست محتوای محبوب محلی، $q \in [2, 12]$.
- C: هزینه cache درMb ، $c \in [0.01, 5]$.

هزینه پهنای باند در هر لینک به صورت تصادفی از [0، 1] (در مگابایت) انتخاب شده است. مقدار پارامترc_{ij} (در هر قطعه داده) را می توان با ضرب کردن هزینه پیوند در(Mb) با اندازه تکه داده محاسبه کرد. در مقایسه با هزینه پهنای باند شبکه، هزینه ذخیره سازی پایین تر است. هزینه ذخیره هر فایل در هر CR به صورت تصادفی از [0، c] (در مگابایت) انتخاب می شود و مقدار پارامترd ، n(در هر قطعه داده) را می توان با ضرب کردن هزینه کش) در (Mb و اندازه تکه داده حساب کرد. از آنجا که محتوای متفاوت دارای اندازه های مختلف هستند و معمولا تنها چند گره در شبکه سرور اصلی هستند، ما اندازه هر محتوا را در F از [1، اندازه سایز] تنظیم می کنیم و به صورت تصادفی یک گره

CR از اولین گره 10٪ به عنوان سرور آن را انتخاب می کنیم. برای هر CR، ظرفیت حافظه پنهان از [0، ca اندازه] تنظیم شده است. مجموعه محتوای هر CR بصورت تصادفی انتخاب می شود.

برای هر نمونه شبیه سازی، ما نه تنها هزینه کل را مقایسه می کنیم، یعنی مجموع هزینه پهنای باند شبکه و هزینه کش، که توسط الگوریتم NCCM با مرز پایین LB و مرز بالایی بدست می آید، بلکه همچنین *حداکثر هزینه اضافی عادی شده*(NMEC) را هم نشان می دهد. به طور خاص، اجازه دهید S_h کل هزینه NCCM (یا *بدون Cache یا r-Cache یا g-Cache*) را نشان دهد و S_l نشان دهنده LB است. NCCM NMEC (یا بدون کش یا r-Cache یا g-Cache) به عنوان $\frac{S_h - S_l}{S_l}$ تعریف شده است. به غیر از موارد دیگر مشخص شده، تعدادی تکرار در NCCM T = 30 را تعیین می کنیم. مقادیر پیش فرض پارامترها عبارتند از$ch_{size} = 10$:. $ca_{size} = 200$، co size = 300، = | F | 20، q = 4، c = 0. 05، η = 0. 025، α = 0. 7و μ = 0. 7 . در بحث های زیر، هر آزمایش دارای یک پارامتر تغییر یافته از مقدار پیش فرض آن است. این به ما کمک می کند که نحوه عملکرد هر استراتژی را با یک پارامتر خاص تحت تاثیر قرار دهیم.

در شکل 5، هزینه های کلی NCCM و تمام مرزها به آرامی با افزایش ch_{size}افزایش می یابد و عملکرد NCCM بسیار نزدیک به حد پایین است. نتایج را می توان به شرح زیر توضیح داد. در مرحله اول، از آنجا که یک تکرار داده، واحد داده ای است که در شبکه انتقال داده شده و در هر CR ذخیره می شود، هرچقدر تراکم اطلاعات بیشتر باشد، میزان استفاده از حافظه داخلی در شبکه پایین تر است، زیرا CR نمیتواند بخشی از یک تکه داده را ذخیره کند. بنابراین، میزان استفاده از حافظه در شبکه در این مورد کاهش می یابد. ثانیا، برای هر یک از مواد، باید آن را به تعدادی از تکه های داده با اندازه ثابت تقسیم کرد. اگر اندازه محتوا را نمی توان با اندازه تکه های داده تقسیم کرد، یک تکه داده باید برای استفاده از برنامه نویسی شبکه استفاده شود. بنابراین، در این مورد پهنای باند کل برای جمع آوری محتوا برای هر CR افزایش می یابد. با توجه به این دو دلیل، هر دو هزینه کش و هزینه پهنای باند با افزایش ch_{size} افزایش می یابد. برایNCCM ، از آنجا که اندازه هر فایل ثابت است، هر چه بسته داده ها بزرگتر باشد، تکه های داده کمتری به یک تکه محتوا متعلق است، که بیشتر عملکرد LNC را کاهش می دهد. بنابراین، کاهش میزان استفاده از

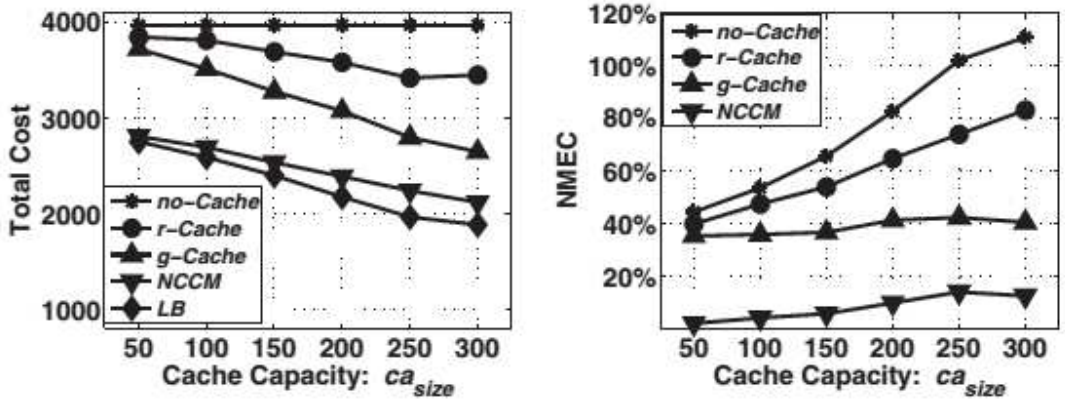
حافظه در شبکه وLNC ، هزینه کل NCCM را کاهش می دهد. در شکل 5، هزینه کل *بدون ذخیره سازی* بزرگترین است، که انتظار ما را برآورده می کند، استفاده از حافظه داخلی در شبکه، کل هزینه را کاهش می دهد. هزینه کلی-r cacheبالاتر از هزینه کلی g-cache است که بسیار بالاتر از هزینه کلی NCCM است، که نشان می دهد که این امر برای بررسی مشترک استراتژی ذخیره سازی، مسیریابی محتوا و برنامه نویسی شبکه مفید می باشد تا بیشتر هزینه کل را کاهش دهد. شکل 5(b) NMEC مرزهای بالا و NCCM را نشان می دهد. ما به وضوح می توانیم مشاهده کنیم که NMEC NCCM همیشه زیر 10٪ است .

شکل 6 (a) نشان می دهد که هزینه های کلr-cache ، g-cache، NCCMو LB با افزایش اندازه ca کاهش می یابد. واضح است که وقتی ظرفیت حافظه پنهان هر CR افزایش می یابد، داده های بیشتری ازCR نزدیک را می توان کم کرد. کل هزینه بدون حافظه پنهان است پایدار است زیرا حافظه داخلی در شبکه استفاده نمی شود. شکل 6 (ب) نشان می دهد که NMEC از NCCM برای اولین بار افزایش می یابد و سپس تمایل دارد که پایدار باشد. از سوی دیگر، NMECsاز سه مرز بالایی افزایش می یابد، زمانی که اندازه بزرگتر بزرگتر می شود .

در شکل 7(a) ، کل هزینه های سه مرز بالایی، NCCMو LB به صورت خطی با افزایش حجم ترکیب افزایش می یابد. از آنجا که تعداد محتویات درخواست شده توسط هر CR ثابت شده است، هر چه حجم محتوا بزرگتر باشد، پهنای باند بیشتر و ظرفیت حافظه پنهان مصرف می شود. از سوی دیگر، از آنجا که ظرفیت حافظه پنهان هر CR نیز ثابت شده است، هنگامی که اندازه شرکت افزایش می یابد، تأثیر حافظه در شبکه در کاهش هزینه پهنای باند کوچکتر می شود. بنابراین، همانطور که در شکل 7 (b) نشان داده شده است، NMECs سه مرز بالایی یکسان می شود، وقتی که co_{size} به اندازه کافی بزرگ می شود. برایNCCM ، زمانی که اندازه شرکت به اندازه کافی بزرگ می شود، هر محتوا را می توان به قطعات داده ای بیشتر تقسیم کرد که مزایای LNC را افزایش می دهد. بنابراین، NMECاز NCCM بسیار کوچکتر از مرزهای بالایی است .

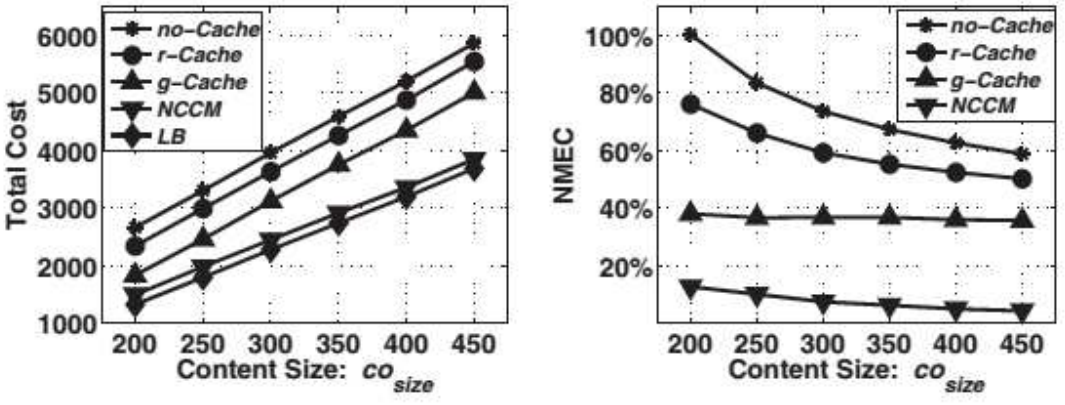
در شکل 8(a) ، هنگامی که q ثابت است، محتوای محبوب تر در F است، CRهای کمتر همان محتوا را می خواهند. برای هر محتوا، تعداد تکه های ذخیره شده در تمام حافظه های موجود در شبکه کاهش می یابد. بنابراین، سه مرز

بالایی به آرامی افزایش می یابد. علاوه بر این، افزایش | F | همچنین منجر به تکه های داده کمتر کد شده منتقل شده در ICN ها با LNC می شود که می تواند بین CR های مختلف درخواست شده به اشتراک گذاشته شود. بنابراین، کل هزینه های NCCM و LB با افزایش $|F|$. افزایش می یابد. شکل 8 (b) نشان می دهد که NMEC ها از مرزهای بالایی خیلی بزرگتر از NCCM NMEC هستند و با افزایش | F | آنها کاهش می یابند زیرا مزایای LNC در NCCM و LB کاهش می یابد. با این وجود، NMEC از NCCM همیشه زیر 10٪ است و در مقایسه با مرزهای بالایی، می تواند حداقل 15٪ کل هزینه را ذخیره کند.



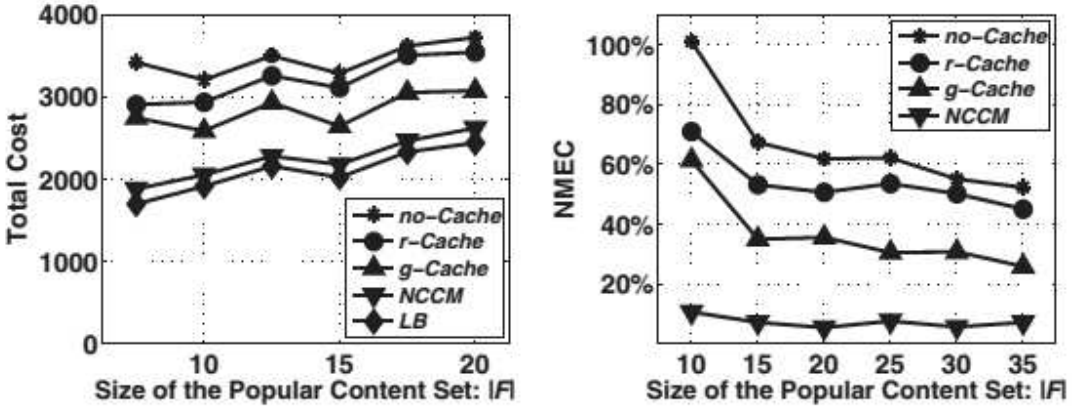
(a) Total cost. (b) NMEC.

شکل 6. هزینه در مقابل ظرفیت کش داده ها.



(a) Total cost. (b) NMEC.

شکل 7. هزینه در مقابل اندازه محتوای داده ها.



(a) Total cost. (b) NMEC.

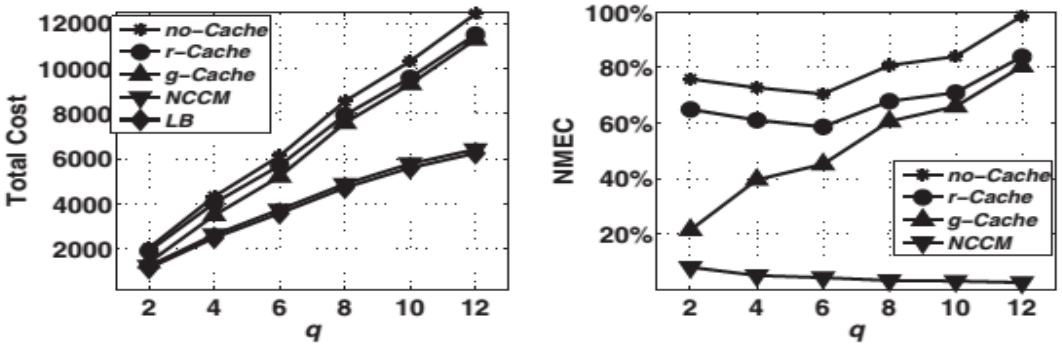
شکل 8. هزینه در مقابل تعداد محتویات محبوب.

شکل 9 (a) نشان می دهد که هزینه های کل سه مرز بالایی، NCCMو LB با افزایش q افزایش می یابد. وقتی F | |ثابت است، همان طور که تعدادی از مطالب درخواست شده در هر CR افزایش می یابد، پهنای باند بیشتری مصرف می شود. از سوی دیگر، از آنجا که تکه های داده های کدگذاری شده که در ICN ها با LNC منتقل می شود می تواند بین درخواست CR های مختلف تقسیم شود، پهنای باند شبکه می تواند با استفاده از LNC کاهش یابد. نرخ رشد NCCMو LB پایین تر از حد بالایی است. بنابراین، شکل 9 (b) نشان می دهد که NMEC از سه مرز بالایی با افزایش q افزایش می یابد. علاوه بر این می توان دید که NMEC از NCCM با افزایش q افزایش می یابد که مزایای NCCMرا نشان می دهد .

شکل 10 (a) نشان می دهد که هزینه های کلr-cache ، g-cache، NCCMو LB با افزایش c افزایش می یابد. این منطقی است زیرا وقتی هزینه ذخیره سازی در هر مگابایت افزایش می یابد هزینه های کل هم افزایش می یابد. هزینه کل بدون حافظه ذخیره شده پایدار می باشد زیرا از کش در شبکه استفاده نمی کند. از آنجائیکه r-cache و g-cache همیشه در هر CR ذخیره داده می شوند، هزینه های کل با افزایش c افزایش می یابد و هزینه ها می توانند بزرگتر از عدم وجود کش باشند. در مقایسه، از آنجا که NCCM و LB به طور مشترک استراتژی ذخیره سازی و محتوای مسیریابی برای به حداقل رساندن کل هزینه را در نظر می گیرند، آنها بخش های داده ای کمتری را در CR

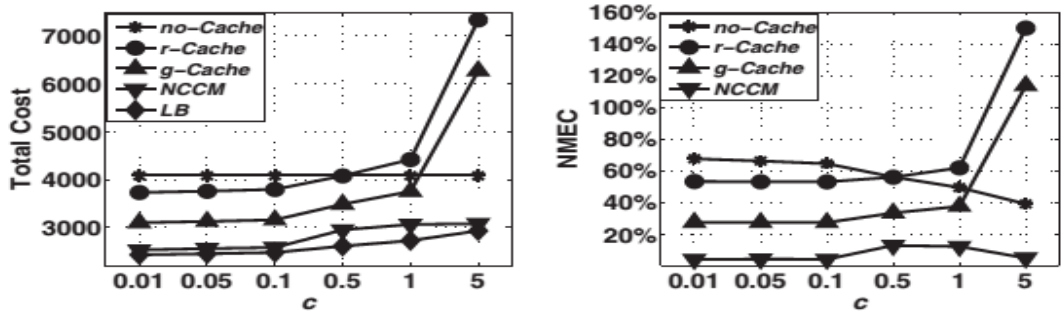
ها ذخیره می کنند وقتی که c بزرگتر می شود. شکل 10 (b) نشان می دهد که NMEC NCCM همیشه زیر 15٪ است.

شکل 11 (a) نشان می دهد که هزینه های کل سه مرز بالایی، NCCM و LB با افزایش η افزایش می یابد. بر اساس مدل Waxman، هنگامی که η بزرگتر می شود، CR در شبکه بیشتر است. CR بیشتر دارای ظرفیت ذخیره سازی بیشتری است. از طرف دیگر CR بیشتر محتوای بیشتری را می کند که هزینه پهنای باند را افزایش می دهد. بنابراین، در این مورد، هزینه کل سه مرز بالایی افزایش می طلبد. برای NCCM و LB، هنگامی که تعداد CR ها افزایش می یابد، تکه های داده های کدگذاری شده می تواند بین CR های مختلف درخواستی تقسیم شود، بنابراین پهنای باند شبکه می تواند با استفاده از LNC کاهش یابد. بنابراین، هزینه های کل NCCM و LB در ابتدا افزایش می یابد و پس از آن تمایل به ثبات دارد. شکل 11 (ب) نشان می دهد که NMEC از NCCM همیشه کمتر از 10٪ است و می تواند حداقل 30٪ کل هزینه در مقایسه با مرزهای بالاتری را ذخیره کند.



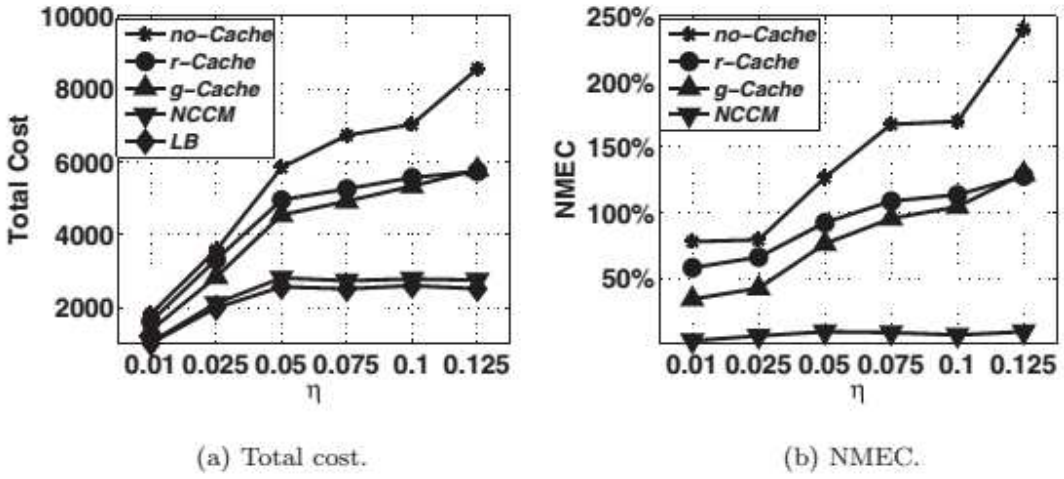
(a) Total cost. (b) NMEC.

شکل 9. هزینه در مقابل اندازه مجموعه درخواستی محتویات محبوب داخلی.



(a) Total cost. (b) NMEC.

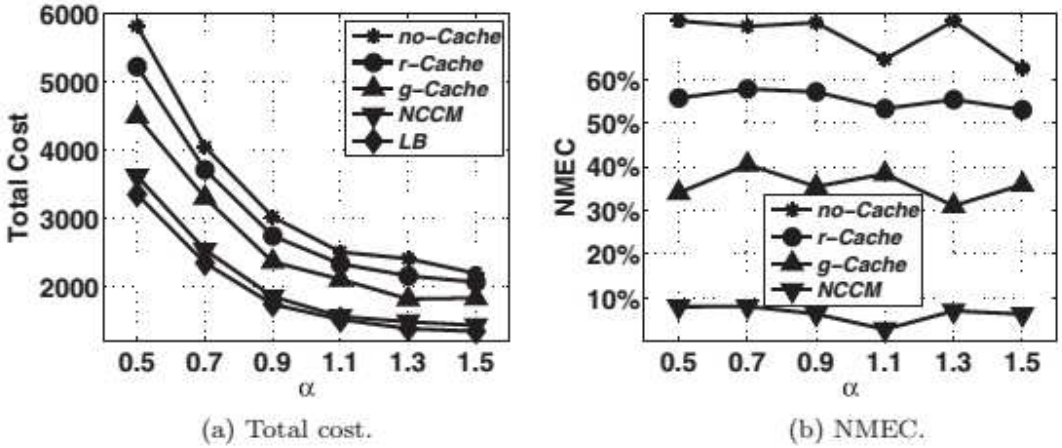
شکل 10. هزینه در مقابل هزینه کش.



(a) Total cost. (b) NMEC.

شکل 11. هزینه در مقابل پارامتر شبکه : η.

شکل 12 (a) نشان می دهد که هزینه های کل سه مرز بالایی، NCCMو LB با افزایش α کاهش می یابد. بر اساس مدلWaxman ، هنگامی که α یا μ بزرگتر می شود، از آنجا که η ثابت است، تعداد CR در شبکه ثابت می شود و احتمال پذیرش هر دوCR ، افزایش می یابد، به این معنی که ICN توپولوژی متراکم تر است . در این حالت، مسیر بین CR درخواست کننده و یکCR خدمت کننده به صورت کوتاهتر خواهد بود و داده های بیشتری از CR مجزایی می توانند جمع آوری شوند. بنابراین هزینه پهنای باند شبکه NCCM یا هر محدودیت می تواند کاهش یابد. شکل 12 (ب) به وضوح نشان می دهد که NMEC از NCCM همیشه زیر 10٪ است و همیشه می تواند حداقل 20٪ هزینه های کلی را در مقایسه با مرزهای بالاتری ذخیره کند. از آنجا که μ بزرگتر می شود، پیوندهای بیشتری را در شبکه نیز افزایش می دهد، نتایج شبیه سازی برای هزینه های نسبت بهμ ، روند مشابهی را نسبت به α دارند.



(a) Total cost. (b) NMEC.

شکل 12. هزینه در مقابل پارامتر شبکه: α

به طور خلاصه، تصاویر 12- 5 نشان می دهند که الگوریتم NCCM در بیشتر مقیاس های شبیه سازی 10 درصد از حد پایین تر از راه حل بهینه را بر اساس اکثر سناریوهای شبیه سازی می گیرد .

در آزمایشات ما نیز NMEC ها را بررسی و زمان اجرای الگوریتم NCCM پیشنهاد شده را در مقایسه با تعداد تکرارها در شکل 13 نشان می دهیم. به طور خاص، وقتی تعداد مطالب محبوب، () F |و اندازه درخواست محلی محتوای محبوب، q، کوچک، متوسط و بزرگ هستند، نتایج به ترتیب در شکل 13(b) - (a) ، (d) - (c)و (e) نشان داده شده استf) - .در تمام این ارقام، پارامترهای دیگر را تنظیم می کنیم و مقادیر | را | F | و q فقط به طور خاص تغییر می دهیم، ما الگوریتم NCCM را بر روی MacBook Pro-2.8GHz با پردازنده Intel Core i7 اجرا می کنیم. همانطور که در شکل 13 نشان داده شده، NMECبا افزایش زمان تکرار و زمان اجرا در تمام موارد کاهش می یابد. از آنجا که الگوریتم NCCM براساس آرام سازی لاگرانژی طراحی شده است، نتیجه به محل مطلوب نزدیک تر می شود وقتی که زمان تکرار برای تعداد بیشتری تنظیم می شود. اگر چه الگوریتم NCCM نیاز به اجرا بصورت تکراری دارد و در نهایت راه حل های غیرقطعی را بدست آورد، در بیشتر موارد NMEC یک راه حل غیرقطعی می تواند در چهار تکرار همگام شود. شکل 13 همچنین نشان می دهد که زمان اجرا در مرحله تکرار بیشتر می شود هنگامی که مقادیر | F | و q بزرگتر شوند. با این حال، برای رسیدن به NMEC کافی پایین (به عنوان مثال، کمتر از 3٪)، الگوریتم NCCM تنها

2.37 ثانیه نیاز دارد، حتی اگر تعداد مطالب محبوب، | F | به اندازه کافی بزرگ است (به عنوان مثال، | 0 10 = | F 0 و 20 = q). بنابراین، الگوریتم NCCM پیشنهاد شده را می توان به طور موثر پیاده سازی کرد.

7 .بحث ها

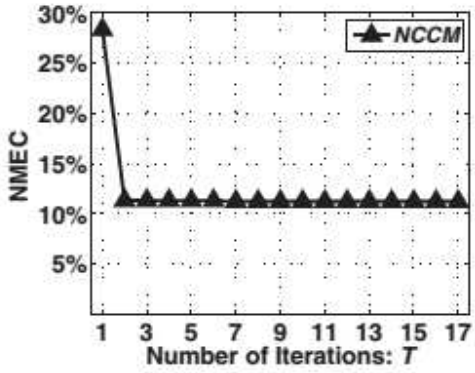
در این بخش، ما درباره آستانه محبوبیت، هزینه بالای محاسبات کنترل، دستگاه های چند منظوره، بازیابی شکست و تحرک دستگاه ها بحث خواهیم کرد .

آستانه تعداد محتویاتی را که می تواند در CR از ICN ذخیره شود، تعیین می کند، یعنی آستانه مقدار | F | را تعیین می کند .الگوریتم NCCM پیشنهاد شده را می توان در مورد کلی استفاده کرد که آستانه بیشتر یا برابر صفر است. اگر آستانه صفر باشد، تمام محتویات را می توان در ICN ذخیره سازی کرد، بنابراین عملکرد شبکه می تواند بهینه شود، اما پیچیدگی محاسباتی بالا است. در صورتی که آستانه بیشتر از صفر باشد، مقادیر | F | کاهش می یابد و تعدادی از متغیرها در الگوریتم NCCM پیشنهادی کاهش می یابد که منجر به عملکرد شبکه زیر بهینه با پیچیدگی محاسباتی کمتری می شود. در اینجا ما می خواهیم توجه کنیم که همیشه بین شبکه و پیچیدگی محاسباتی الگوریتم NCCMپیشنهاد شده رابطه جایگزینی وجود دارد. با این وجود، معمولا تعداد کمی از محتوای محبوب (14٪ - 30٪) نسبت زیادی (90٪) از پهنای باند در شبکه های سنتی را مصرف می کنند [35]. بنابراین، منطقی است که ما محتوای محبوب را فقط در ICN ذخیره کنیم. در عمل، آستانه را می توان با در نظر گرفتن عملکرد شبکه و پیچیدگی محاسباتی طرح پیشنهادی تعیین کرد .

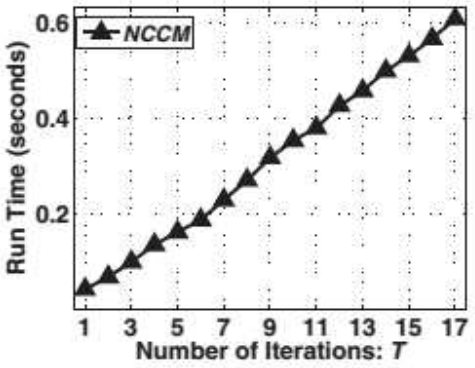
در طرح ما، پیچیدگی محاسباتی اصلی با اجرای الگوریتم NCCM ایجاد می شود. ما همچنین بر این باوریم که طراحی پیشنهادی را می توان در کنترل کننده واقعی SDN به دلایل زیر انجام داد. اولا، ما یادآوری می کنیم که در طرح پیشنهادیNCCM ، محتویات محتوا و محتوای مسیریابی برای دوره زمانی بعدی بر اساس درخواست های تاریخی در دوره زمان فعلی پیش محاسبه می شوند. بنابراین، الگوریتم NCCM خارج از خط اجرا می شود، که نیاز به ظرفیت محاسباتی کنترل کننده را کاهش می دهد. ثانیا، برای یک ICN در مقیاس بزرگ، عملکرد کنترل کننده همچنین می تواند توسط تعدادی از کنترل کننده های همکاری انجام شود، که هر کدام مسئولیت مدیریت CR را در حوزه ی خود

خواهند داشت و اطلاعات را با یکدیگر مبادله می کنند [36،37] . سوم، پیچیدگی محاسباتی الگوریتم NCCM پیشنهادی متناسب با تعداد محتویاتی است که باید در ICN ذخیره شوند، یعنی اندازه محتویات محبوب F. از آنجا که مقدار | F | توسط آستانه برای پهنای باند محلی محاسبه می شود، آستانه می تواند برای تعادل عملکرد شبکه و پیچیدگی محاسبات الگوریتم NCCM انتخاب شود. سرانجام، طی چند سال گذشته، بسیاری از کنترلرهای جدید با استفاده از سرورهای چند هسته ای قدرتمند برای رسیدگی به تعداد زیادی جریان داده در شبکه های بزرگ طراحی شده اند. برای مثال، McNettle می تواند حدود 20 میلیون درخواست در هر ثانیه برای یک شبکه با 50 0 0 سوئیچ را مدیریت کند [16].

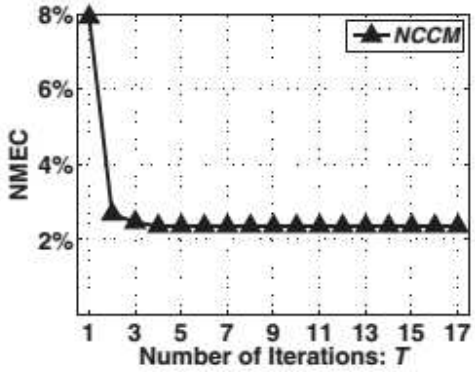
برای بررسی دستگاه های چند منظوره، ابتدا می توانیم CR مجازی با ظرفیت ذخیره سازی صفر را برای هر یک از دستگاههای چند منظوره معرفی کنیم. ما فرض می کنیم که دستگاه چند منظوره فقط با CR مجازی متصل است و مجازی CR متصل به چند CR است. سپس تمامی درخواست های دستگاه چند منظوره به عنوان درخواست CR مجاز مربوطه مورد بررسی قرار می گیرد. پیوندهای بین هر CR مجازی و سایر CR ها دارای هزینه پهنای باند صفر است. با چنین تغییری، هر دستگاه چند منظوره را می توان به عنوان یک دستگاه تک منفرد تلقی کرد و ما می توانیم یک گراف G را برای ورود الگوریتم NCCM به دست آوریم. پس از این که راه حل را بدست آوریم، تکه های داده منتقل شده از چندین CR به CR مجازی به این معنی است که این تکه های داده به طور مستقیم به دستگاه چند منظوره مربوط به CR مجازی منتقل می شود. یک مثال در شکل 14 نشان داده شده است: همان طور که در شکل 14 نشان داده شده است، ده دستگاه وجود دارد که در مربع های خاکستری نشان داده شده اند و هفت CR که در دایره های سفید نشان داده شده اند. برای دستگاه چند منظوره u_1، یک CR v_8 مجازی با ظرفیت ذخیره سازی صفر اضافه می کنیم. سپس توپولوژی اصلی شبکه را می توان به یک توپولوژی شبکه ای بدون دستگاه های چند منظوره تبدیل کرد. پس از آن ما می توانیم بر اساس آن گراف G را بدست آوریم. ما اجازه می دهیم هزینه پهنای باند ارتباط بین v_1 ,v_8 ، v_3 و v_8 صفر باشد.



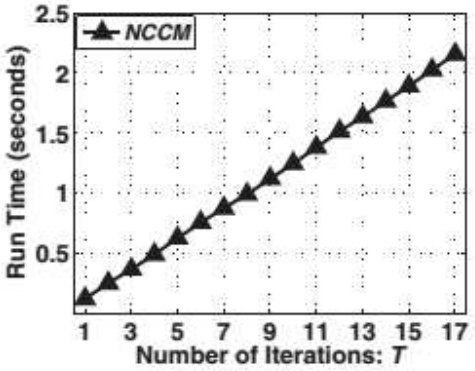
(a) NMEC vs Number of iterations ($|F| = 10, q = 5$)



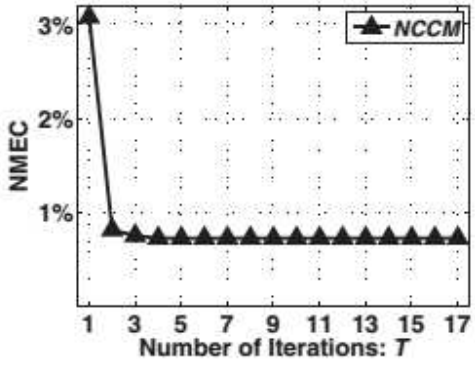
(b) Run time vs Number of iterations ($|F| = 10, q = 5$)



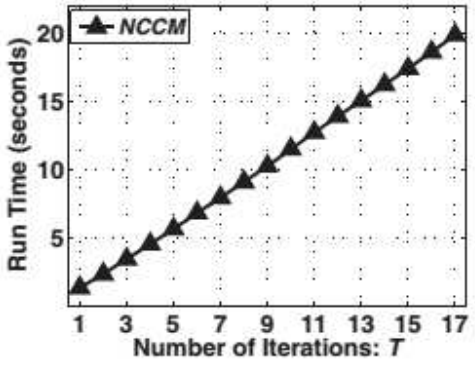
(c) NMEC vs Number of iterations ($|F| = 100, q = 10$)



(d) Run time vs Number of iterations ($|F| = 100, q = 10$)



(e) NMEC vs Number of iterations ($|F| = 1000, q = 20$)

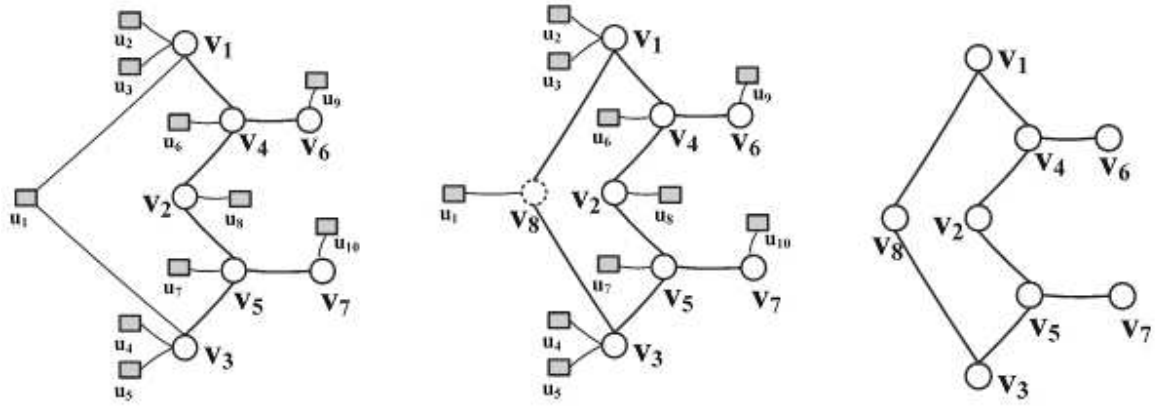


(f) Run time vs Number of iterations ($|F| = 1000, q = 20$)

شکل 13. NMEC در مقابل تعداد تکرارها (زمان اجرا).

ناکامی در بازیابی سخت افزارها یا پروتکل ها یک مسئله مهم است و به خوبی برای SDN [38]،ICN [39]، ICN [40] و ICN مبتنی بر SDN [41] مورد مطالعه قرار گرفته است. به عنوان مثال، [20] نشان می دهد که برای خراب شدن داده ها به علت برخی از خرابی ها (مانند شکست گره، شکست پیوند، قفل کردن رابط، خطای انتقال) و بسته شدن بسته به دلیل تراکم و غیره، برنامه نویسی شبکه می تواند عملکرد بهتری نسبت به ICN بدون NC داشته باشد به دلیل اینکه داده های کدگذاری شده توسط CR توسط گیرنده ها بارگیری می شوند برای همه دستگاه هایی که هنوز هم نیاز به تکه هایی از یک محتوای مشابه دارند مفیدند. علاوه بر این، پیاده سازی ICN با NC در [21] پیشنهاد شده است که در آن مسئله بازیابی شکست با استفاده از ارسال مجدد در بسته ها مورد بررسی قرار گرفته است و دوباره به مسیرهای جایگزین بازگشته است. از آنجا که چنین سخت افزاری یا پروتکل های مبتنی بر SDN و ICN، الگوریتم NCCM پیشنهاد شده می تواند به خوبی با این تکنیک های بازیابی شکست سازگار ترکیب شود.

در طرح پیشنهادیNCCM ، درخواست ها از طریق پرونده ها ارسال می شود. هنگامی که یک کاربر به یک مکان دیگر منتقل می شود و به یک CR جدید متصل می شود، می تواند به سادگی چندین تکه داده های کد شده از یک محتوا که هنوز رمزگشایی نشده اند و دوباره به دست آمده، را مجددا درخواست کند. در این حالت، می توان کاربر را به عنوان یک کاربر جدیدتلقی کرد و درخواست تنها نیاز به مشخص کردن تعداد تکه های داده کد شده دارد، زیرا در NCابICN ، کاربر می تواند محتوای اصلی را رمزگشایی کند و آن را بدست آورد، در صورتی که بتواند تعداد کافی از بسته های داده کد شده را بدست آورد. بنابراین، این ساده تر از آن در ICN بدونNC ، مانند CCNx است. درCCNx ، هنگامی که یک کاربر به مکان دیگری منتقل می شود، تمام تکه های داده ای که دریافت نشده اند دوباره درخواست می شود. نشان داده شده است که CCNx همچنان می تواند تا 97٪ درخواست ها را در هنگام تحرک بالا نگه دارد [42،43].



(a) Original Network Topology. (b) Network Topology Transformation. (c) Graph G.

شکل 14. نمونه ای از تغییر شکل توپولوژی شبکه برای ICN با دستگاههای چند منظوره.

8 .نتیجه گیری

در این مقاله، ما به طور سیستماتیک قدرت LNC را بررسی کردیم تا هزینه پهنای باند شبکه و هزینه ذخیره در ICN ها کاهش یابد. به طور خاص، ابتدا یک چارچوب جدید مبتنی بر SDN برای مدیریت کش در ICN ها با LNC پیشنهاد کردیم. سپس یک مشکل مدیریت حافظه پنهان برای ICN ها با LNC را برای کمینه کردن هزینه پهنای باند شبکه و هزینه های ذخیره سازی به صورت مشترک با در نظر گرفتن استراتژی ذخیره سازی و مسیریابی محتوا تشکیل دادیم. ما همچنین یک الگوریتم مبتنی بر آرامش لاگرانژی کارایی یعنی NCCM را برای به دست آوردن راه حل مدیریت نزدیک به مطلوب کشف کردیم. برای ارزیابی عملکرد چارچوب و الگوریتمNCCM ، مرزهای بالایی و پایین تر از مشکل را به دست آوردیم و آزمایش های گسترده ای را برای مقایسه عملکرد الگوریتم NCCM پیشنهادی با این محدودیت ها انجام دادیم. نتایج شبیه سازی اثربخشی الگوریتم NCCM پیشنهاد شده و چارچوب را تایید می کند. به طور خاص، نتایج عددی گسترده نشان می دهد که الگوریتم NCCM پیشنهاد شده به 10٪ از حد پایین تر از راه حل بهینه در بیشتر سناریوهای شبیه سازی می رسد. علاوه بر کاهش هزینه پهنای باند شبکه، امنیت شبکه یکی دیگر از چالش های اصلی شبکه های ارتباطی است. طرح های جستجو بر روی داده های رمز شده در محیط ابری بررسی شده اند [44-47]. در آینده، ما طرحهای جستجو را بر روی داده های رمزگذاری شده ذخیره شده در CRs بررسی خواهیم کرد .

تقدیر و تشکر

این کار تا حدودی از سوی مرکز نوآوری مشارکتی فناوری نوین نرم افزار و صنعتی سازی، بنیاد علوم طبیعی چین (شماره 61202378، شماره 61272462، شماره 61271165، شماره 61301153، شماره 61572310) پشتیبانی شده است. برنامه 973 چین (CB3291032013)، برنامه پژوهشی شرق شانگهای ، بنیاد علوم طبیعی موسسات آموزش عالی استان جیانگ سو شماره KJB52004016، تحقیقات بنیاد برنامه کاربردی سوژو شماره SYG201401، شورای تحقیقاتی هنگ کنگ تحتCRF C7036-15G ، و آزمایشگاه Key of Tianjin AdvancedNetworking (TANK)، دانشکده کامپیوتر و فناوری، دانشگاه تیانجین، تیانجین چین، 300350 پشتیبانی شده است.

مواد تکمیلی

مواد تکمیلی مرتبط با این مقاله را می توان در نسخه آنلاین در 10.1016/j.comnet.2016.08.004 پیدا کرد.