

# تحلیل مسأله بهینه‌سازی تخصیص ظرفیت در پروکسی‌های SIP و ارائه الگوریتم کنترل اضافه بار با عدالت Max-Min

مجتبی جهانبخش، سیدوحید ازهری و وحید قاسم‌خانی

جریان‌های محلی اولویت بالاتری اختصاص می‌دهد و بنابراین جریان‌های بین دامنه‌ای به گذردهی مناسبی دست پیدا نمی‌کنند. برای حل این چالش در ادامه تابع هدف حداقل مربعات را بررسی می‌نماییم که نتیجه آن فراهم‌شدن عدالت max-min بین جریان‌ها می‌باشد. این نوع عدالت هنگامی فراهم می‌شود که افزایش تخصیص به یک جریان سبب کاهش تخصیص به جریان‌های دیگر با سهم کمتر یا برابر نشود. این نکته تضمین می‌نماید که جریان‌های بین دامنه‌ای آسیبی نمی‌بینند.

با توجه به این که در شرایط عملیاتی، حل مرکزی و یکپارچه مسأله بهینه‌سازی توسط حل‌کننده، سربار قابل توجهی داشته و از طرف دیگر با افزایش تعداد جریان‌ها و پروکسی‌ها، زمان محاسباتی نیز بالاتر خواهد رفت، بنابراین برای تخصیص عادلانه بین جریان‌ها از کلاس‌بندی و زمان‌بندی استفاده کرده‌ایم. از طرف دیگر اضافه بار به صورت ناگهانی و پیش‌بینی نشده به وقوع می‌پیوندد و پروکسی‌ها نیازمند واکنش سریع نیز می‌باشند، در غیر این صورت ارسال مجدد کاربرها اتفاق افتاده و شرایط وخیم‌تر خواهد شد. در این مقاله به منظور پوشش‌دادن به تخصیص با زمان محاسباتی پایین و همچنین کنترل اضافه بار با واکنش سریع از راهکارهای زیر استفاده شده است:

(۱) تخصیص منابع در هر پروکسی با استفاده از یک زمان‌بند نوبت گردشی (RR) انجام می‌شود که تضمین‌کننده عدالت max-min از دید هر پروکسی (به صورت محلی) بوده و همچنین پیچیدگی محاسباتی پایین  $O(1)$  دارد.

(۲) یک روش کنترل اضافه بار ارائه می‌شود که از طریق کنترل طول صف جریان‌ها، اضافه بار را کنترل کرده و از ارسال مجدددها جلوگیری می‌کند و علاوه بر آن از حداکثر سهم اختصاص داده شده نیز استفاده می‌نماید.

در ادامه این مقاله، در بخش دوم تحقیقات پیشین مورد ارزیابی قرار می‌گیرد و مسأله بهینه‌سازی NUM و روش‌های کنترل اضافه بار موجود بررسی می‌شوند. در بخش سوم توپولوژی و مسأله بهینه‌سازی مربوطه مطرح و همچنین دو تابع هدف مجموع گذردهی جریان‌ها و حداقل مربعات با ابزار حل‌کننده بررسی می‌شود. در بخش چهارم روش کنترل اضافه بار ارائه و نتایج آن با نتایج بهینه‌سازی و روش OCC مقایسه می‌شود و در انتها در بخش پنجم جمع‌بندی انجام می‌گردد.

## ۲- مرور تحقیقات پیشین

در سال ۲۰۰۲ نسخه به روز شده استاندارد SIP [۱] ارائه شد و در آن به منظور کنترل اضافه بار محلی از پیام پاسخ ۵۰۳ برای رد کردن درخواست‌های مازاد بر ظرفیت استفاده شد، هر چند که در تحقیقاتی مثل [۲] و [۳] عدم پایداری استفاده از روش محلی (به تنهایی) مشخص شد و

چکیده: پروتکل SIP یک پروتکل لایه کاربرد است که برای ایجاد، مدیریت و اتمام جلسات چندرسان‌های در زیرسیستم‌های چندرسان‌های IP در نظر گرفته شده است. استفاده وسیع از این پروتکل منجر به حجم بالای ترافیک در پروکسی‌های SIP شده و تخصیص با دقت منابع پردازشی به جریان‌ها را به یک مسأله با اهمیت بالا تبدیل می‌کند. در این مقاله به تحلیل مسأله بهینه‌سازی تخصیص منابع پردازشی در پروکسی‌های SIP می‌پردازیم و دو تابع هدف حداکثرسازی مجموع گذردهی و حداقل‌سازی مربعات را مورد ارزیابی قرار می‌دهیم. تابع هدف حداکثرسازی مجموع گذردهی، به جریان‌های بین دامنه‌ای اولویت پایین‌تری را اختصاص می‌دهد زیرا این جریان‌ها برخلاف جریان‌های داخل دامنه‌ای از دو پروکسی میانی عبور می‌کنند. از سوی دیگر، تابع هدف حداقل‌سازی مربعات به سیاست با عدالت max-min منتهی می‌شود و بنابراین به منظور عملیاتی‌سازی این نوع سیاست، در پروکسی‌ها از سرویس‌دهی نوبت گردشی (RR) استفاده می‌کنیم و یک الگوریتم ارائه می‌دهیم که با کنترل طول صف جریان‌ها، اضافه بار را کنترل کرده و از ارسال مجدد و ناپایداری جلوگیری می‌کند. این الگوریتم در مقایسه با روش‌های موجود به استفاده بهتر از منابع پردازشی در پروکسی‌ها منتهی می‌شود.

کلیدواژه: اضافه بار، بهینه‌سازی، پروتکل SIP، پروکسی، عدالت max-min.

## ۱- مقدمه

گسترده‌گی و تنوع سرویس‌های فراهم‌شده توسط شبکه‌های IP منجر به روی آوردن تکنولوژی‌های مختلف به یکپارچه‌سازی انواع شبکه‌های دسترسی و گرویدن به شبکه‌های نسل آینده ارتباطات شده است. پروتکل SIP [۱] به عنوان پروتکل سیگنالینگ در IMS که بستر پیام‌رسانی در شبکه‌های نسل آینده است در نظر گرفته شده است. سرور اصلی ارائه‌دهنده سرویس در پروتکل SIP، پروکسی نامیده می‌شود. ظرفیت هر پروکسی در ارتباط با دیگر پروکسی‌ها (چه پروکسی‌های اپراتور یکسان و یا دیگر اپراتورها) محدود است و بنابراین پروکسی باید در شرایط اضافه بار علاوه بر استفاده از یک الگوریتم کنترل اضافه بار کارآمد که پایداری مناسبی دارد، از حداکثر منابع پردازشی خود نیز استفاده نموده و به پارامترهایی مثل عدالت بین جریان‌های مختلف نیز توجه داشته باشد.

در این مقاله ابتدا با تحلیل مسأله بهینه‌سازی کنترل اضافه بار با تابع هدف مجموع گذردهی جریان‌ها نشان می‌دهیم که این تابع هدف به

این مقاله در تاریخ ۶ مرداد ماه ۱۳۹۵ دریافت و در تاریخ ۲۱ مرداد ماه ۱۳۹۶ بازنگری شد.

مجتبی جهانبخش، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران،  
(email: m\_jahanbakhsh@comp.iust.ac.ir)

سیدوحید ازهری، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران،  
(email: azharivs@iust.ac.ir)

وحید قاسم‌خانی، دانشکده مهندسی کامپیوتر، دانشگاه علم و صنعت ایران، تهران،  
(email: vahidgk@gmail.com)

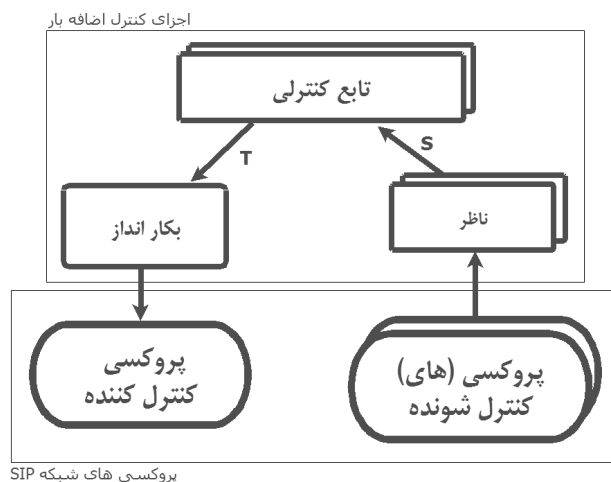
استفاده برای کنترل T می‌تواند مواردی مثل سطل سوراخ‌دار<sup>۵</sup>، رد کردن درصدی، مبتنی بر پنجره، مبتنی بر سطل نشانه<sup>۶</sup> و فراخوانی شکاف‌دار<sup>۷</sup> باشد.

در روش‌های محلی هر پروکسی هم نقش کنترل‌کننده را دارد و هم نقش کنترل‌شونده و بنابراین هر سه بخش ناظر، تابع کنترل و بکارانداز در هر پروکسی به صورت مجزا اجرا می‌شوند و عملاً بازخوردی ارسال نمی‌شود. با توجه به این که هزینه رد کردن تماس‌های مازاد بر ظرفیت، قابل توجه می‌باشد اگر اضافه بار از حد مشخصی بالاتر باشد حتی رد کردن تمامی تقاضاها نیز کنترل اضافه بار و پایداری را به همراه نخواهد داشت [۷]. در روش‌های انتها به انتها واحد ناظر در همه پروکسی‌های یک مسیر واقع شده است حال آن که واحد بکارانداز در پروکسی مبدأ ارسال‌کننده تقاضاها قرار گرفته است. اشکال این روش پیچیدگی زیاد آن جهت یافتن کلیه مسیرهای منتهی به مقصد مورد نظر و جمع‌بندی اطلاعات به دست آمده برای تصمیم‌گیری در مورد محدود کردن ارسال بسته‌ها به یک مسیر خاص است [۸]. روش‌های انتها به انتها همیشه نمی‌توانند در سطح تک تک مسیرها به کنترل اضافه بار بپردازند. به عنوان مثال وقتی که یک تقاضا مربوط به یک دامنه مدیریتی دیگر باشد، مسیر در پروکسی لبه دامنه تقاضاکننده قابل کشف نمی‌باشد، مورد دیگر هنگامی است که سیاست پروکسی‌های میانی مشخص نباشد.

با توجه به این محدودیت‌ها قسمت عمده روش‌ها و راهکارهای مطرح‌شده در مقالات و بدنه استانداردهای مرتبط به روش‌های گام به گام می‌باشد (به عنوان مثال [۹] تا [۱۱]). در این روش‌ها بخش بکارانداز بر روی پروکسی بالادست (در نقش کنترل‌کننده) و بخش ناظر بر روی پروکسی پایین‌دست (در نقش کنترل‌شونده) اجرا می‌شود. عملیاتی‌تر بودن روش‌های گام به گام منجر به استانداردسازی نحوه تعبیه بازخورد در سرایند VIA بسته‌های SIP در [۹] شد. در این کار همچنین یک روش مبتنی بر کاهش درصد نرخ ارسالی نیز معرفی شده است. در روش‌های مبتنی بر کاهش درصد نرخ ارسالی، پروکسی پایین‌دست تعیین می‌کند که پروکسی بالادست چه درصدی از تقاضاهای خود را رد کند در حالی که مکانیزم استفاده‌شده در [۱۰] مبتنی بر نرخ می‌باشد که در آن سرور پایین‌دست مرتباً به نرخ تقاضاهای دریافتی از پروکسی‌های بالادست خود نظارت می‌کند و میزان نرخ قابل قبول دریافتی خود را به آنها اعلام می‌نماید. این مرجع تنها به نحوه پیاده‌سازی بخش بکارانداز در پروکسی بالادست و مکانیزم ارسال بازخورد تمرکز داشته و بخش ناظر و تابع کنترلی جهت تعیین درجه اضافه بار را مورد بررسی قرار نمی‌دهد.

روش معرفی‌شده در مقاله حاضر به صورت گام به گام و مبتنی بر طول صف می‌باشد که در آن پروکسی بالادست، اضافه بار جریان‌ها را با توجه به بازخورد دریافتی، گزردهی و نرخ ورودی کنترل می‌کند. این روش پوشش‌دهنده تمامی اجزای شکل ۱ می‌باشد. بخش ناظر در پروکسی پایین‌دست هر جریان قرار دارد و به صورت دوره‌ای طول صف جریان مربوطه را به پروکسی بالادست ارسال می‌کند. بخش تابع کنترلی در پروکسی بالادست قرار دارد و با توجه به بازخورد پایین‌دست و پارامترهای محاسبه‌شده محلی، احتمال رد کردن را برای بخش بکارانداز تنظیم می‌نماید.

در حالت اضافه بار از یک پروکسی جریان‌های متعددی عبور می‌کند و



شکل ۱: اجزای کنترل اضافه بار SIP.

نیاز به روش‌های توزیع‌شده مشخص گردید. روش‌های توزیع‌شده به دو صورت گام به گام و انتها به انتها پیاده‌سازی می‌شوند. در روش‌های گام به گام مقدار بازخورد<sup>۱</sup> به طور مستقیم بین هر جفت از پروکسی‌های مجاور منتقل می‌شود در حالی که در روش‌های انتها به انتها، هر پروکسی مربوط به یک مسیر خاص، بازخورد خود را به پروکسی کنترل‌کننده که پروکسی ارسال‌کننده اولیه است ارسال می‌کند. شکل ۱ اجزای روش‌های کنترل اضافه بار را نشان می‌دهد. در این شکل، کنترل اضافه بار یک یا چند پروکسی کنترل‌شونده که مربوط به یک مسیر مشخص هستند توسط یک پروکسی کنترل‌کننده انجام می‌شود. کنترل اضافه بار عموماً از سه بخش ناظر<sup>۲</sup>، تابع کنترلی<sup>۳</sup> و بکارانداز<sup>۴</sup> تشکیل شده است.

- ناظر: وظیفه این بخش اندازه‌گیری مقدار بار فعلی پروکسی دارای اضافه بار و ارائه نمونه‌های کنترلی S (Sample) است. پارامترهای کنترلی استفاده‌شده معمول در مقالات مواردی مثل میزان بار پردازشی CPU (به عنوان مثال [۴])، طول صف (به عنوان مثال [۵]) و تأخیر برقراری جلسات قبلی (به عنوان مثال [۶]) می‌باشند. محاسبه پارامتر کنترلی در بخش ناظر، عموماً به صورت دوره‌ای انجام می‌شود.

- تابع کنترلی: پیاده‌سازی الگوریتم کنترل اضافه بار در این بخش قرار دارد. این بخش با توجه به نمونه‌های کنترلی S که از یک یا چند پروکسی جمع‌آوری می‌شود نسبت به میزان پارامتر کاهش T (Throttle) تصمیم‌گیری می‌نماید. این بخش هم می‌تواند در پروکسی کنترل‌کننده واقع باشد و هم در پروکسی‌های کنترل‌شونده. در حالت اول پروکسی کنترل‌کننده بازخوردی که دریافت می‌کند مقدار خام نمونه‌های کنترلی S از جانب یک یا چند پروکسی است (به عنوان مثال CPU) و با توجه به این بازخوردها، مقدار T را به دست می‌آورد. در حالت دوم، تابع کنترلی مستقر در هر پروکسی کنترل‌شونده، با توجه به پارامتر S بخش ناظر خود (به عنوان مثال CPU) مقدار قابل قبول T را محاسبه کرده و به عنوان بازخورد به پروکسی کنترل‌کننده ارسال می‌کند.

- بکارانداز: هدف این بخش اعمال پارامتر کاهش T بر روی تقاضاهای ورودی در پروکسی کنترل‌کننده می‌باشد. روش مورد

1. Feedback
2. Monitor
3. Control Function
4. Actuator

5. Leaky Bucket  
6. Token Bucket  
7. Call Gapping

که هر تقاضا مربوط به یک کاربر مجزا است و مدل کردن کنترل اضافه بار چنین سناریویی عملاً غیر ممکن می‌باشد، پروکسی در نقش بالادست به عنوان منبع تولید ترافیک در نظر گرفته می‌شود.

– پروکسی‌های شبکه SIP باید به نحوی هزینه رد کردن بار مازاد بر ظرفیت را نیز به عهده بگیرند. در TCP منابع تولید ترافیک، خود باید بار ارسالی را کنترل کنند اما در SIP مسأله به این سادگی نیست و منبع تولید ترافیک کاربران هستند که هر کدام تنها به اندازه یک جلسه بار ایجاد می‌کنند. رد کردن بار مازاد بر ظرفیت در پروکسی‌ها، مستلزم استفاده از منابع پردازشی می‌باشد و باید در مسأله بهینه‌سازی اضافه شود.

– سربرار روش‌های کنترل اضافه بار کارایی کنترل اضافه بار را تحت تأثیر قرار می‌دهد. در TCP هزینه لینک‌ها عموماً توسط روش‌های ضمنی در منابع تولید ترافیک مشاهده می‌شود و سربرار روش استفاده شده در منابع تولید ترافیک نقشی در کارایی لینک‌ها نخواهد داشت در حالی که در SIP، هزینه روش کنترل اضافه بار (ضمنی و یا صریح) تأثیر به‌سزایی در کارایی پروکسی دارد.

توپولوژی مطرح شده در این مقاله در قیاس با توپولوژی‌های مطالعه شده در مقالات، توپولوژی پیچیده‌تری می‌باشد. اکثر روش‌های ارائه شده جهت کنترل اضافه بار به بررسی در توپولوژی‌های ساده (مثل دوزنقه و محلی) می‌پردازند [۱۴] تا [۱۷]. تعداد دیگری از مقالات به بررسی توپولوژی‌های کامل‌تری نیز پرداخته‌اند اما ظرفیت پروکسی‌های بالادست را نامحدود در نظر گرفته‌اند و جریان معکوس را نیز در نظر نمی‌گیرند [۸] و [۱۸]. مسأله بهینه‌سازی بررسی شده در این مقاله مربوط به توپولوژی پیچیده‌ای است که در آن ظرفیت همه پروکسی‌ها محدود است و ترافیک بین پروکسی‌ها به صورت گراف کامل با یک یا دو پروکسی میانی و دوطرفه است. نکته دیگر این است که عدالت مد نظر اکثر روش‌های موجود عدالت نسبی است [۱۹] تا [۲۱] که در آن کنترل اضافه بار و تخصیص منابع به جریان‌های با نرخ کمتر آسیب می‌رساند. روش ارائه شده در این مقاله علاوه بر کارایی و پایداری مناسب، دارای عدالت max-min [۲۲] بین جریان‌های هر پروکسی است.

در بخش سوم مسأله بهینه‌سازی تخصیص منابع پردازشی SIP مطرح می‌گردد و دو تابع هدف حداکثرسازی مجموع گذردهی‌ها و همچنین تابع هدف حداقل مربعات بررسی می‌شوند.

### ۳- مسأله بهینه‌سازی تخصیص منابع SIP

در حالت کلی شبکه SIP شامل یک مجموعه  $L$  تایی از پروکسی‌های SIP با ظرفیت  $C_i$  می‌باشد. هر پروکسی این ظرفیت را در دو نقش بالادست و پایین‌دست بین دیگر پروکسی‌ها تقسیم می‌کند. در شکل ۲ یک شبکه SIP با چهار پروکسی نمایش داده شده است ( $L=4$ ). به عنوان مثال در این شکل نرخ تقاضاهای کاربران پروکسی ۱ به کاربران پروکسی ۲ با  $\lambda_{1,2}$  مشخص شده و در حالت اضافه بار پروکسی ۱ به میزان  $T_{1,2}$  را به پروکسی ۲ ارسال کرده و بقیه را رد می‌کند (بر طبق نتایج مطرح شده در [۲۳] رد کردن تقاضاهای مازاد بر ظرفیت تنها در پروکسی با نقش بالادست انجام می‌شود).

جزء اساسی مکانیزم کنترل اضافه بار، تنظیم نرخ ارسالی به هر پروکسی ( $T_{i,j}$ ) می‌باشد. بنابراین در مسأله بهینه‌سازی باید این پارامتر به نحوی تنظیم شود که تابع هدف حداکثر و یا حداقل شود. نکته دیگر این است که با توجه به این که در حالت اضافه بار ممکن است نرخ تقاضای

پروکسی علاوه بر کنترل مناسب اضافه بار هر جریان، باید در مورد نحوه تعیین سهم جریان‌ها نیز تصمیم‌گیری نماید. سیاست تخصیص سهم جریان‌ها، یک مسأله بهینه‌سازی می‌باشد. در این مقاله با به کارگیری و توسعه مسأله بهینه‌سازی کنترل ازدحام پروتکل TCP به ارائه مسأله بهینه‌سازی تخصیص منابع پردازشی در پروکسی‌های SIP می‌پردازیم. کنترل ازدحام TCP یک مکانیزم توزیع شده جهت اشتراک ظرفیت لینک‌ها در بین کاربرهای در حال رقابت برای به دست آوردن لینک‌ها می‌باشد و شامل دو مؤلفه اساسی است: الگوریتم منبع تولید ترافیک که به طور پویا به تنظیم نرخ ارسالی می‌پردازد و الگوریتم لینک که به صورت ضمنی هزینه را به روز می‌کند. در اینترنت فعلی الگوریتم منبع توسط پروتکل TCP و الگوریتم لینک‌ها توسط روش‌های مدیریت صف (AQM) مثل Drop-Tail و RED ارائه می‌شود. روش‌های مختلف TCP از متریک‌های متفاوتی برای اندازه‌گیری میزان ازدحام استفاده می‌کنند. به عنوان مثال روش Reno از احتمال گم شدن بسته‌ها و Vegas از تأخیر صف به عنوان ملاک ازدحام استفاده می‌کند [۱۲]. کنترل ازدحام TCP از مدل شبکه‌ای با مجموعه  $L$  لینک با ظرفیت محدود  $C = (c_l, l \in L)$  استفاده می‌کند که به صورت مشترک توسط  $N$  منبع تولید ترافیک استفاده می‌شوند (با اندیس  $s$ ). هر کدام از منابع تولید ترافیک از یک سری از لینک‌ها استفاده می‌کنند:  $S(l) = \{s \in N | l \in L(s)\}$  که در این عبارت مجموعه  $\{L(s)\}$  مشخص‌کننده یک ماتریس  $L \times N$  است ( $R_{ls}$ ) که در آن اگر منبع  $s$  از لینک  $l$  استفاده کند یک و در غیر این صورت صفر است. هر منبع تولید ترافیک  $s$  در هر لحظه به اندازه  $x_s(t)$  تولید ترافیک می‌کند و هر لینک  $l$  در هر لحظه یک مقدار به اندازه  $\lambda_l(l) > 0$  را به عنوان هزینه لینک به روز می‌کند. جمع ترافیک تولید شده در هر لینک باید کوچک‌تر از ظرفیت این لینک باشد. این شبکه به وسیله مسأله بهینه‌سازی NUM<sup>1</sup> [۱۳] مدل می‌شود که در (۱) نشان داده شده است

$$\begin{cases} \text{maximize} & \sum_s U_s(x_s) \\ \text{subject to} & Rx \leq c \\ \text{minimize}_{\lambda \geq 0} & D(\lambda) = \sum_s \max_{x_s \geq 0} (U_s(x) - x_s \sum_l R_{ls} \lambda_l) + \sum_l c_l \lambda_l \end{cases} \quad (1)$$

با توجه به مسأله بهینه‌سازی (۱) و دوگان آن می‌توان مدل عمومی الگوریتم‌های کنترل ازدحام را استخراج کرده و نشان داد که روش‌های کنترل ازدحام به الگوریتم‌های توزیع شده‌ای تفسیر می‌شوند که (۱) را حل می‌کنند. کنترل ازدحام پروتکل TCP و کنترل اضافه بار پروتکل SIP از نقطه نظر بهینه‌سازی دارای این شباهت می‌باشند که مسأله آنها توزیع شده بوده و با هدف حداکثرسازی استفاده از منابع می‌باشد. اما تفاوت کنترل ازدحام TCP و کنترل اضافه بار SIP در این است که کنترل ازدحام TCP یک مکانیزم توزیع شده برای استفاده حداکثری از ظرفیت لینک‌ها است در حالی که کنترل اضافه بار SIP یک مکانیزم توزیع شده برای استفاده حداکثری از ظرفیت پروکسی‌ها و علاوه بر آن جلوگیری از اضافه بار در آنها است و بنابراین مسأله اضافه بار SIP دارای پیچیدگی‌های زیر نیز می‌باشد:

– هر پروکسی هم نقش منبع تولید ترافیک را دارد و هم نقش لینک. البته منبع تولید ترافیک واقعی کاربرها می‌باشند اما با توجه به این

جدول ۱: پارامترهای آزمایش مسأله بهینه‌سازی.

$\lambda_{1,1} = 100 \text{ cps}$ , $\lambda_{1,2} = 180 \text{ cps}$ , $C_1 = 460 \text{ cps}$	۱ پروکسی
$\lambda_{2,1} = 50 \text{ cps}$ , $\lambda_{2,2} = 180 \text{ cps}$ , $C_2 = 460 \text{ cps}$	۲ پروکسی
$k = 0$ , $w_{\text{rej}} = 0.1$ , $w_{\text{rej}} = 0.2$	پارامترهای مشترک

### ۳-۱ بهینه‌سازی با تابع هدف مجموع گذردهای

فرض کنید که  $L=2$  و هر یک از دو پروکسی در ارتباط با پروکسی دیگر هم در نقش بالادست و هم پایین دست ایفای نقش می‌کند. علاوه بر این هر پروکسی دارای بار محلی نیز می‌باشد. این توپولوژی در شکل ۳ نشان داده شده است. می‌خواهیم مسأله بهینه‌سازی (۲) را در دو حالت عمده مورد بررسی قرار دهیم: هنگامی که یک پروکسی در محدوده اضافه بار قرار دارد و هنگامی که هر دو پروکسی در حالت اضافه بار قرار دارند. نتایج این بخش حاصل حل مسأله بهینه‌سازی توسط ابزار CVX می‌باشد.

در صورتی که تابع هدف مجموع گذردهای باشد مسأله بهینه‌سازی برای  $L=2$  به صورت (۳) درخواهد آمد

$$\begin{aligned} \max & T_{1,1} + T_{1,2} + T_{2,1} + T_{2,2} \\ \text{s.t.} & T_{1,1} + 0.2(\lambda_{1,1} - T_{1,1}) + T_{1,2} + 0.2(\lambda_{1,2} - T_{1,2}) + T_{2,1} \leq C_1 \\ & T_{2,1} + 0.2(\lambda_{2,1} - T_{2,1}) + T_{2,2} + 0.2(\lambda_{2,2} - T_{2,2}) + T_{1,2} \leq C_2 \\ & 0 \leq T_{1,1} \leq \lambda_{1,1}, 0 \leq T_{1,2} \leq \lambda_{1,2}, 0 \leq T_{2,1} \leq \lambda_{2,1}, 0 \leq T_{2,2} \leq \lambda_{2,2} \end{aligned} \quad (3)$$

با توجه به چهار جریان نشان داده شده در شکل ۳ امکان دارد که یک یا هر دو پروکسی در ناحیه اضافه بار قرار گرفته و نیاز به رد کردن تعدادی از تقاضاها در پروکسی‌ها باشد. در ادامه این دو حالت مورد بررسی قرار می‌گیرند.

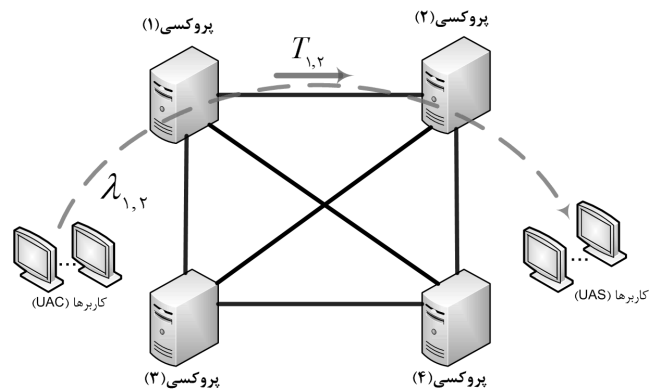
### ۳-۱-۱ یک پروکسی در اضافه بار

فرض کنید پارامترهای مسأله بهینه‌سازی (۳) به صورت جدول ۱ باشند. با توجه به پارامترهای جدول ۱، مجموع بار ورودی پروکسی ۱ با ۴۶۰ تماس در ثانیه (۱۸۰+۱۸۰+۱۰۰) و مجموع بار ورودی پروکسی ۲ برابر با ۴۱۰ تماس در ثانیه (۱۸۰+۱۸۰+۵۰) می‌باشد و بنابراین میزان بار پردازشی پروکسی ۱ و پروکسی ۲ به ترتیب برابر با ۱ و ۰.۸۹ می‌شود. بار پردازشی مطابق با (۴) به طور ساده برابر با مجموع گذردهای عبوری از پروکسی به علاوه هزینه رد کردن بار مازاد و هزینه ارسال مجدد، تقسیم بر ظرفیت آن در نظر گرفته شده است

$$U_i = \frac{\sum_{j \in L} (T_{i,j} + w_{\text{rej}}(\lambda_{i,j} - T_{i,j}) + \lambda_{i,j} k w_{\text{rej}}) + \sum_{j \in L, j \neq i} T_{j,i}}{C_i} \quad (4)$$

با اضافه شدن نرخ هر کدام از جریان‌های مربوط به پروکسی ۱، این سرور در محدوده اضافه بار قرار گرفته و نیاز به کنترل اضافه بار آن می‌باشد. سؤالات زیر قابل تأمل است:

- بالاتر رفتن هر کدام از بارهای محلی  $(\lambda_{1,1})$ ، خروجی  $(\lambda_{1,2})$  و ورودی  $(\lambda_{2,1})$  چه تأثیری در گذردهای پروکسی دارای اضافه بار دارد؟
- پروکسی ۲ در بالاتر رفتن تابع هدف چه نقشی می‌تواند داشته باشد؟
- در چه حالت‌هایی پروکسی ۲ نیز دچار اضافه بار خواهد شد؟
- بر طبق نتیجه حل (۳) توسط ابزار CVX، بالاتر رفتن ترافیکی که از پروکسی ۱ شروع می‌شود  $(\lambda_{1,1})$  و رفتار متفاوتی با ترافیک دریافتی  $(\lambda_{2,1})$  خواهد داشت. با بالاتر رفتن بار  $\lambda_{1,2}$  و یا  $\lambda_{1,1}$  گذردهای  $T_{2,1}$  کاهش



شکل ۲: سناریوی شبکه مبتنی بر SIP با  $L=4$ .

ارسالی در نقش بالادست از یک پروکسی به پروکسی دیگر  $(\lambda_{i,j})$  از ظرفیت اختصاص داده شده بالاتر باشد، مقداری از تقاضاها با هزینه  $w_{\text{rej}}$  رد می‌شوند و میزان گذردهای عبوری برابر با  $T_{i,j}$  خواهد بود. با توجه به این فرضیات مسأله بهینه‌سازی به صورت (۲) تعریف خواهد شد

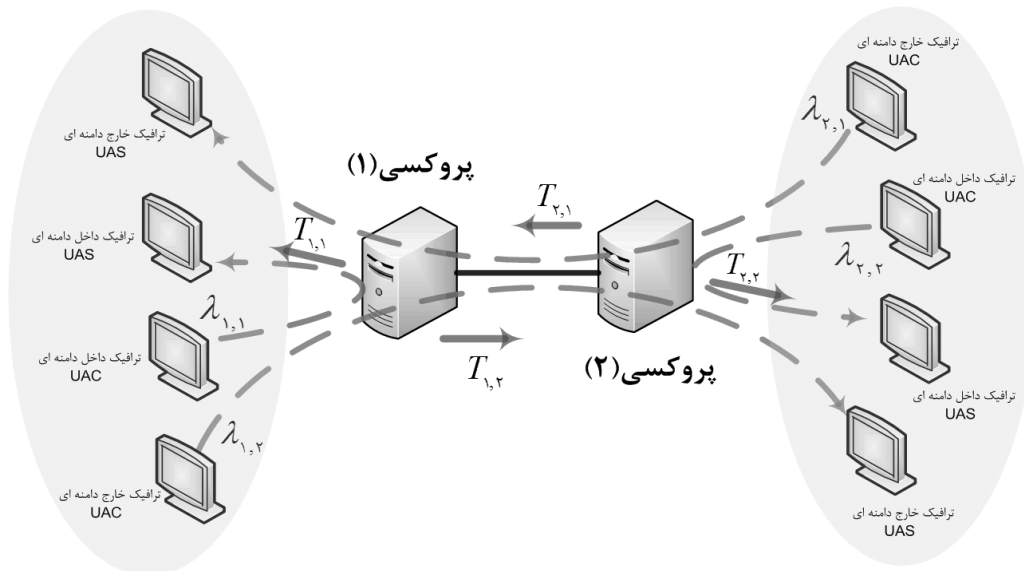
$$\begin{aligned} \max & \sum_{i,j \in L} U_{i,j}(T_{i,j}) \\ \text{s.t.} & \sum_{j \in L} (T_{i,j} + w_{\text{rej}}(\lambda_{i,j} - T_{i,j}) + \lambda_{i,j} k w_{\text{rej}}) + \\ & \sum_{j \in L, j \neq i} T_{j,i} \leq C_i, \quad i \in L \\ & 0 \leq T_{i,j} \leq \lambda_{i,j}, \quad i, j \in L \end{aligned} \quad (2)$$

قید اول این مسأله به منظور تقسیم ظرفیت پروکسی بین دیگر پروکسی‌ها در دو نقش بالادست و پایین دست نوشته شده است. در این عبارت ظرفیت پروکسی  $i$  باید بزرگتر یا مساوی با مجموع هزینه‌های زیر باشد:

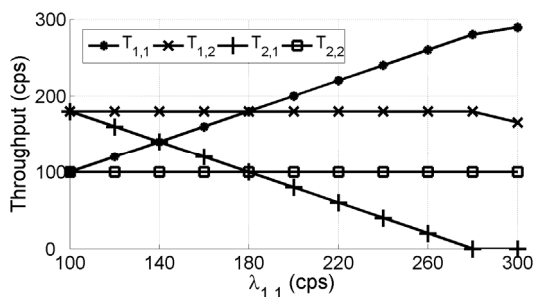
- $T_{i,j}$ : گذردهای ارسالی پروکسی به سمت دیگر پایین دست‌ها.
- $(\lambda_{i,j} - T_{i,j}) \cdot w_{\text{rej}}$ : سربار رد کردن بار مازاد ظرفیت پایین دست‌ها.
- $\lambda_{i,j} k w_{\text{rej}}$ : هزینه فرستادن ارسال مجدد تقاضاها هنگامی که پایین دست‌ها به علت واقع شدن در اضافه بار فرصت نمی‌کنند پیام موقت مثل Trying-۱۰۰ و یا Ringing-۱۸۰ را به موقع ارسال کنند. ترم سوم این عبارت برای حالتی نوشته شده است که بار دریافتی در پروکسی بالادست به متوسط تعداد  $k$  بار ارسال مجدد شده باشد.

$T_{j,i}$ : گذردهای دریافتی از پروکسی‌های بالادست.  
قید دوم مسأله بهینه‌سازی تعیین می‌کند که تمامی گذردهای  $(T_{i,j})$  و بارهای ورودی  $(\lambda_{i,j})$  باید یک مقدار غیر منفی باشند و هیچ کدام از گذردهای آنها از ترافیک ورودی متناظر خود نمی‌تواند بالاتر باشد. تابع هدف (۲) مجموع بهره‌وری هر کدام از جریان‌های ترافیکی است و بهره‌وری هر جریان ترافیکی تابعی از گذردهای ارسال شده می‌باشد.

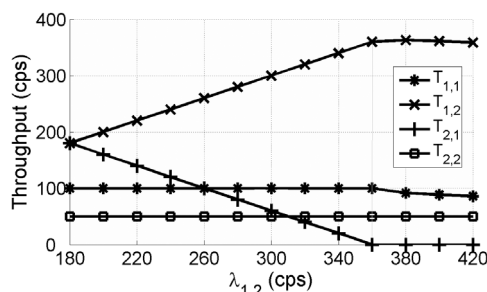
در ادامه این بخش ابتدا مسأله بهینه‌سازی با تابع هدف مجموع گذردهای مورد بررسی قرار می‌گیرد و در ادامه عدالت max-min بین جریان‌ها مطالعه می‌شود. قابل ذکر است که به منظور بررسی ساده‌تر، تنها سناریوی با دو پروکسی مورد ارزیابی قرار می‌گیرد، هر چند که با توجه به جامع بودن مسأله بهینه‌سازی، نتایج برای هر تعداد پروکسی معتبر و قابل تعمیم خواهد بود. نکته دیگر این است که در هر آزمایش تنها یکی از نرخ‌های ورودی که نقش کلیدی در رفتار پروکسی دارد تغییر می‌کند، هر چند که تغییر هم‌زمان همه نرخ‌ها با نتایج به دست آمده در این بخش مطابقت می‌کند.



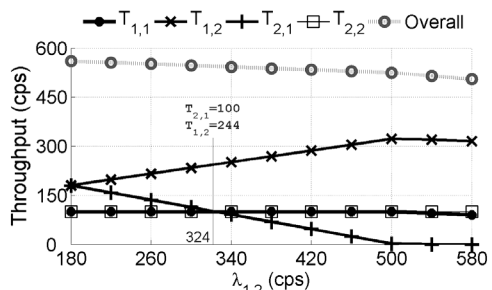
شکل ۳: سناریوی شبکه SIP با  $L = 2$ .



شکل ۵: تأثیر افزایش بار محلی پروکسی ۱ وقتی که هر دو پروکسی در حالت اضافه بار قرار دارند.



شکل ۴: نتیجه بهینه‌سازی هنگامی که پروکسی ۱ دارای اضافه بار باشد و  $\lambda_{1,2}$  افزایش یابد.



شکل ۶: تأثیر افزایش بار  $\lambda_{1,2}$  وقتی که هر دو پروکسی در حالت اضافه بار قرار دارند.

- بار محلی افزایش یابد.

- ترافیک بین دامنه‌ای افزایش یابد.

طبق شکل ۵ با افزایش بار محلی پروکسی ۱ ( $\lambda_{1,1}$ ), پروکسی ۲ با رد کردن بار ارسالی خود به پروکسی ۱ و کاهش گذردهی  $T_{1,1}$  به افزایش گذردهی مجموع کمک می‌کند. این نتیجه با توجه به این نکته حاصل می‌شود که بار محلی تنها از یک پروکسی عبور می‌کند و بنابراین پروکسی دیگر با رد کردن باری که از هر دو پروکسی عبور می‌کند ( $\lambda_{1,1}$ ) به افزایش گذردهی مجموع کمک می‌کند. نکته جالب توجه دیگر این است که در این سناریو منابع پردازشی پروکسی ۲ پایین‌تر آمده و از حالت اضافه بار خارج می‌شود. در نرخ ۲۸۰ تماس در ثانیه استفاده از منابع پردازشی در پروکسی ۲ تا مقدار ۰/۶۸ پایین می‌آید که می‌تواند در ارتباط با دیگر پروکسی‌ها و یا بار محلی به طور مناسبی از آن استفاده کرد.

در شکل ۶ تأثیر افزایش بار بین دامنه‌ای  $\lambda_{1,2}$  نشان داده شده است. در این آزمایش به منظور رسیدن به حداکثر گذردهی مجموع، نرخ رد کردن و

می‌یابد. در حقیقت وظیفه رد کردن تقاضاها به پروکسی ۲ موقوف می‌شود تا سر بار رد کردن در پروکسی دارای اضافه بار وجود نداشته باشد. به عنوان مثال مطابق با شکل ۴ در صورت اضافه شدن  $\lambda_{1,2}$ ، فرایند کاهش  $T_{1,1}$  تا جایی ادامه پیدا می‌کند که گذردهی ارسالی از پروکسی ۲ به پروکسی ۱ به صفر برسد. در نرخ‌های بالاتر از  $\lambda_{1,2} = 360$  گذردهی  $T_{1,1}$  برابر صفر می‌باشد و پروکسی ۱ به ناچار سر بار رد کردن تقاضاهای مازاد بر ظرفیت ( $\lambda_{1,1}$  یا  $\lambda_{1,2}$ ) را به قیمت کاهش گذردهی خود متحمل می‌شود.

در آزمایش دیگر فرض کنید که  $\lambda_{1,2} = 180$  تماس در ثانیه است و ترافیک  $\lambda_{1,1}$  افزایش یابد. در این حالت پروکسی ۲ تا جایی که ظرفیت پردازشی داشته باشد با کنترل گذردهی ارسالی به پروکسی ۱ ( $T_{1,1}$ ) از ایجاد بار اضافه در پروکسی ۱ جلوگیری می‌کند. با بالا رفتن  $\lambda_{1,1}$  به نقطه‌ای خواهیم رسید که منابع پردازشی هر دو پروکسی برابر یک خواهد شد. در صورت حل مسأله بهینه‌سازی با فرضیات جدول ۱ و  $\lambda_{1,1} = 430$  تماس در ثانیه در نقطه‌ای قرار داریم که هر دو پروکسی در نقطه بار پردازشی برابر با یک قرار گرفته و  $T_{1,1} = 180$  تماس در ثانیه است. در این نقطه منابع پردازشی پروکسی ۲ طبق (۴) یک است  $(U_p = (180 + (430 - 180) \times 0.72 + 50 + 180) / 460 = 1)$ .

### ۳-۱-۲ هر دو پروکسی در اضافه بار

فرض کنید که پارامترها مشابه جدول ۱ و تنها  $\lambda_{1,1} = 100$  تماس در ثانیه است. با توجه به این مقادیر بار ورودی هر دو پروکسی برابر با ۴۶۰ تماس در ثانیه و میزان بار پردازشی هر دو ۱ می‌باشد. با هر تماس دیگر چه محلی و یا غیر محلی اضافه بار ایجاد خواهد شد. بر اساس نتایج CVX دو حالت متفاوت ایجاد خواهد شد:

که در این عبارت  $C_{i,j}^{(i)}$  سهم اختصاص داده شده در پروکسی  $i$  به جریان  $i, j$  و همچنین  $L$  تعداد جریان‌های عبورکننده از این پروکسی می‌باشد. تابع هدف (۶) به این منظور انتخاب شده که مجموع اختلاف بین سهم فرضی عادلانه جریان‌ها  $(C_i/L)$  و سهم اختصاص‌یافته، حداقل شود. مسأله بهینه‌سازی (۶) نسبت به (۲) دو تغییر دارد. تفاوت اول قید تخصیص ظرفیت است که به مساوی تبدیل شده است. تابع حداقل مربعات در صورتی به عدالت max-min منجر می‌شود که مسأله بهینه‌سازی به صورت مفید به ظرفیت (work-conserving) باشد و به عبارت دیگر پروکسی از تمامی ظرفیت خود استفاده کند. بنابراین حیطه مورد مطالعه مسأله بهینه‌سازی را متوجه نقطه‌ای می‌کنیم که مجموع بار ورودی بیشتر از ظرفیت باشد  $(\sum_{j \in L} \lambda_{i,j} + \sum_{j \in L, j \neq i} \lambda_{j,i} \geq C_i)$  و به منظور استفاده از تمامی ظرفیت، قید ظرفیت را مساوی در نظر می‌گیریم. تفاوت دیگر این است که برخلاف مسأله بهینه‌سازی (۲) که تخصیص منابع همه پروکسی‌ها را به صورت سراسری و یک جا انجام می‌دهد، در (۶) هر پروکسی  $(i \in L)$  به طور جداگانه به حل مسأله مربوط به خود می‌پردازد. این انتخاب به این علت صورت گرفته است که با توجه به توزیعی بودن مسأله کنترل اضافه بار SIP، حل جداگانه تخصیص منابع در هر پروکسی به صورت مجزا از نظر پیاده‌سازی عملیاتی‌تر می‌باشد.

فرض کنید پارامترهای مسأله مشابه جدول ۱ هستند با این تفاوت که  $\lambda_{1,2} = 100$  و  $\lambda_{2,1} = 324$  می‌باشند. در جدول ۲ نتیجه حل برای سه مقدار متفاوت  $\lambda_{1,1}$  و سه سیاست متفاوت قرار دارد: طبق مسأله بهینه‌سازی (۲)، مسأله بهینه‌سازی (۶) و عدالت max-min سراسری. هنگامی که  $\lambda_{1,1}$  برابر با ۱۸۰ تماس در ثانیه است نتیجه حل مسأله بهینه‌سازی (۶) توسط پروکسی ۱ برای  $T_{1,1}$ ،  $T_{1,2}$  و  $T_{2,1}$  به ترتیب برابر با ۱۰۰، ۱۴۴ و ۱۸۰ تماس در ثانیه به دست می‌آید. در حقیقت پروکسی ۱ به منظور رعایت عدالت max-min بعد از کنار گذاشتن سهم جریان  $\lambda_{1,1}$  که کمترین تقاضا است بقیه ظرفیت باقیمانده (۳۶۰) را به صورت یکسان بین دو جریان دیگر تقسیم می‌کند و به هر جریان گذردهی ۱۸۰ اختصاص می‌دهد. بنابراین جریان  $\lambda_{1,2}$  موظف است که این ظرفیت را صرف پذیرش ۱۴۴ تماس و رد کردن بار مازاد خود بکند  $(180 = 144 + (324 - 144) \times 0.7)$  در حالی که جریان  $\lambda_{2,1}$  که تقاضایی در حد سهم عادلانه خود دارد می‌تواند از سهم خود برای رسیدن به گذردهی ۱۸۰ استفاده نماید. حل مسأله بهینه‌سازی در پروکسی ۲ برای  $T_{2,1}$ ،  $T_{2,2}$  و  $T_{1,2}$  به ترتیب برابر با ۱۰۰، ۱۸۰ و ۱۸۰ تماس در ثانیه به دست می‌آید. هر چند که پروکسی ۲ برای جریان  $\lambda_{1,2}$  گذردهی ۱۸۰ در نظر می‌گیرد اما پروکسی ۱ به علت رعایت عدالت بین جریان‌های خود موفق به ارسال بیشتر از ۱۴۴ تماس نمی‌شود. نتیجه جالب این است که در صورت افزایش  $\lambda_{1,1}$ ، هر دو پروکسی دچار کم‌باری می‌شوند. مثلاً هنگامی که  $\lambda_{1,1}$  برابر با ۳۲۴ است هر دو پروکسی برای جریان ارسالی خود به پروکسی دیگر گذردهی ۱۴۴ را محاسبه کرده و برای جریان دریافتی از پروکسی دیگر سهم فرضی ۱۸۰ را در نظر می‌گیرند، هر چند که پروکسی دیگر به دلیل رعایت عدالت تنها از ۱۴۴ سهم خود استفاده می‌کند.

در ستون سوم جدول ۲ نتیجه قابل قبول برای عدالت سراسری با توجه به مفهوم عدالت max-min استخراج شده است. به عنوان مثال هنگامی که  $\lambda_{1,1}$  برابر با ۲۳۲ باشد، عادلانه‌ترین حالت وقتی اتفاق خواهد افتاد که بعد از اختصاص سهم ۱۰۰ برای هر یک از دو جریان محلی، بقیه ظرفیت دو پروکسی به نحوی تقسیم شوند که ابتدا سهم جریان کمتر ( $\lambda_{1,1}$ ) مشخص شود و جریان بیشتر ( $\lambda_{1,2}$ ) از سهم مشخص شده عادلانه خود

جدول ۲: حل مسأله با سه سیاست متفاوت.

$\lambda_{1,1}$	عدالت max-min سراسری	عدالت در هر پروکسی طبق (۶)	حداکثر مجموع گذردهی طبق (۲)
۱۸۰	$T = \begin{bmatrix} 100 & 144 \\ 180 & 100 \end{bmatrix}$ $T_{Overall} = 524$	$T = \begin{bmatrix} 100 & 144 \\ 180 & 100 \end{bmatrix}$ $T_{Overall} = 524$	$T = \begin{bmatrix} 100 & 244 \\ 100 & 100 \end{bmatrix}$ $T_{Overall} = 544$
۲۳۲	$T = \begin{bmatrix} 100 & 151.2 \\ 174.2 & 100 \end{bmatrix}$ $T_{Overall} = 525.4$	$T = \begin{bmatrix} 100 & 144 \\ 167 & 100 \end{bmatrix}$ $T_{Overall} = 511$	$T = \begin{bmatrix} 100 & 215.1 \\ 123.1 & 100 \end{bmatrix}$ $T_{Overall} = 528.2$
۳۲۴	$T = \begin{bmatrix} 100 & 164 \\ 164 & 100 \end{bmatrix}$ $T_{Overall} = 528$	$T = \begin{bmatrix} 100 & 144 \\ 144 & 100 \end{bmatrix}$ $T_{Overall} = 488$	$T = \begin{bmatrix} 100 & 164 \\ 164 & 100 \end{bmatrix}$ $T_{Overall} = 528$

گذردهی به نحوی انتخاب می‌شود که از حداکثر منابع دو پروکسی استفاده شود. در حقیقت در این حالت دو پروکسی سربار رد کردن نرخ مازاد را بین هم تقسیم می‌کنند. به عنوان مثال در نرخ  $\lambda_{1,2} = 324$  برای رسیدن به حداکثر مقدار تحلیلی گذردهی باید (۵) حل شود

$$\begin{cases} T_{1,2} + (324 - T_{1,2}) \times 0.7 + T_{2,1} = 360 \\ T_{2,1} + (180 - T_{2,1}) \times 0.7 + T_{1,2} = 360 \end{cases} \quad (5)$$

$$\Rightarrow T_{1,2} = 244, T_{2,1} = 100 \text{ cps}$$

### ۳-۲ رعایت عدالت max-min بین جریان‌ها

مطابق نتایج بخش قبل در صورتی که تابع هدف مسأله، حداکثرسازی مجموع گذردهی‌ها باشد خواهیم داشت:

- اگر یک پروکسی دارای اضافه بار باشد بهتر است که دیگر پروکسی‌ها بار ارسالی خود به پروکسی مذکور را کنترل کنند تا مجموع گذردهی بالاتری حاصل شود.
- اگر تمامی پروکسی‌ها دارای اضافه بار باشند ترافیک محلی از اولویت بالاتری برخوردار می‌باشد. همچنین ترافیک ارسالی و دریافتی بین پروکسی‌ها باید به نحوی تنظیم شود که همچنان از حداکثر منابع پردازشی استفاده شود.

تابع حداکثرسازی مجموع گذردهی‌ها برای سروهای یک خوشه که نقش یکسانی دارند مناسب می‌باشد اما هنگامی که پروکسی‌های مربوط به نقاط جغرافیایی غیر یکسان، ارائه‌دهندگان سرویس غیر یکسان و یا دارای نقش غیر یکسانی باشند، در نظر داشتن پارامترهایی مثل رعایت عدالت بین جریان‌ها نیز قابل اهمیت می‌شود. برای اضافه شدن عدالت به مسأله بهینه‌سازی لازم است که تابع هدف تغییر کند. با استناد به [۲۲] در مسأله تخصیص منابع (۲)، تابع هدف حداقل مربعات<sup>۱</sup> با رعایت شرایطی (که در ادامه توضیح داده می‌شود) به عدالت max-min منتهی می‌شود. بنابراین (۲) را به صورت (۶) بازنویسی می‌کنیم

$$\min \sum_{j \in L} (C_{i,j}^{(i)} - \frac{C_i}{L})^2 + \sum_{j \in L, j \neq i} (T_{j,i} - \frac{C_i}{L})^2$$

$$\text{s.t. } C_{i,j}^{(i)} = T_{i,j} + w_{rej} (\lambda_{i,j} - T_{i,j}) + \lambda_{i,j} kW_{retx}, j \in L \quad (6)$$

$$\sum_{j \in L} C_{i,j}^{(i)} + \sum_{j \in L, j \neq i} T_{j,i} = C_i$$

$$0 \leq T_{i,j} \leq \lambda_{i,j}, 0 \leq T_{j,i} \leq \lambda_{j,i}, j \in L$$

$$\rho_{i,j} = \max\left[\frac{1 - \frac{C_{i,j}^{(i)}}{\lambda_{i,j}}}{1 - w_{rej}}, 1 - \frac{C_{i,j}^{(j)}}{\lambda_{i,j}}\right], \quad i, j \in L \quad (8)$$

که در این عبارت احتمال رد کردن  $(\rho_{i,j})$  برابر با مقدار بیشتر از بین دو احتمال به دست آمده برای کنترل اضافه بار در پروکسی بالادست و پایین‌دست (به ترتیب) است. به منظور این که احتمالات محاسبه شده در (۸) به استفاده کارا از منابع و همچنین جلوگیری از ناپایداری، ارسال مجدد و افت گذردهی منجر شود دو پارامتر دیگر نیز مورد استفاده قرار می‌گیرد:

(۱)  $\alpha C_{i,j}$ : مقدار آستانه جهت پذیرش بار مازاد که در صف قرار می‌گیرد به این امید که در آینده نزدیک برای پردازش آن ظرفیت آزاد شود. مقدار  $\alpha$  وابسته به تأخیر شبکه و مهم‌تر از آن تایمر  $T_1$  مورد استفاده در پروتکل SIP است. پارامتر  $T_1$  عموماً برابر با ۵۰۰ میلی‌ثانیه در نظر گرفته می‌شود و در صورتی که منقضی شود تقاضای مربوطه ارسال مجدد می‌شود. ارسال مجدد سبب وخیم‌تر شدن شرایط اضافه بار می‌گردد و بنابراین باید تا حد امکان از آن جلوگیری کرد. به طور مثال برای جلوگیری از ارسال مجدد به یک پروکسی با فقط یک جریان و با فرض تأخیر ناچیز شبکه، طول صف حداکثر می‌تواند برابر با  $0.5C$  باشد. مقدار  $\alpha$  در حالت کلی  $(T_1 - D_{RTT})/L$  است که در آن  $L$  تعداد جریان‌های عبوری از پروکسی می‌باشد و  $D_{RTT}$  تأخیر شبکه برای دریافت پاسخ موقت در ارسال‌کننده تقاضا و جلوگیری از ارسال مجدد در آن است.

(۲)  $Q_{i,j}$ : این پارامتر مشخص‌کننده طول فعلی صف تقاضا می‌باشد و باید از پارامتر  $\alpha C_{i,j}$  کم شود تا در صورتی که در برهه‌های قبل مقداری بار مازاد پذیرش شده و پروکسی موفق به کاهش طول صف نشده است، در برهه فعلی دوباره ظرفیت مازاد اختصاص داده نشود.

بنابراین فرمول نرخ رد کردن بار مازاد جریان  $i$  به  $j$  در پروکسی کنترل‌کننده اضافه بار  $i$  به صورت (۹) می‌باشد

$$\rho_{i,j} = \max[\rho_{i,j}^{(i)}, \rho_{i,j}^{(j)}], \quad i, j \in L$$

$$\rho_{i,j}^{(i)} = \frac{1 - \frac{[(1 + \alpha)C_{i,j}^{(i)} - Q_{i,j}]}{\lambda_{i,j}}}{1 - w_{rej}} \quad (9)$$

$$\rho_{i,j}^{(j)} = 1 - \frac{(1 + \alpha)C_{i,j}^{(j)} - Q_{i,j}}{\lambda_{i,j}}$$

در شکل ۷ شبه‌کد الگوریتم استفاده شده در پروکسی بالادست نشان داده شده است. بخش تابع کنترلی پروکسی بالادست به صورت دوره‌ای (۰/۱ ثانیه در شبیه‌سازی) اجرا می‌شود و در هر برهه با توجه به پارامتر  $Q_{i,j}^{(j)}$  که از پروکسی پایین‌دست دریافت کرده است و همچنین مقادیر محاسبه شده  $C_{i,j}^{(i)}$  و  $C_{i,j}^{(j)}$  و طول صف فعلی خود  $(Q_{i,j}^{(i)})$  مقدار  $\rho_{i,j}$  را محاسبه می‌کند. قابل ذکر است هنگام شروع به کار و یا بعد از صرف‌بودن بار ورودی برای این که روش بر روی گذردهی صفر ثابت نماند در صورتی که مقدار گذردهی صفر برابر با مقدار  $T_{min}$  تنظیم می‌شود.

بخش تابع کنترلی بعد از محاسبه  $\rho_{i,j}$ ، پارامترهای محاسبه شده  $\lambda_{i,j}$  و  $T_{i,j}$  را برابر با صفر قرار می‌دهد تا این مقادیر مجدداً برای برهه بعد محاسبه شوند. بخش پردازش پروکسی به منظور تخمین  $\lambda_{i,j}$  و  $T_{i,j}$  از دو شمارنده ساده استفاده می‌کند که با دریافت هر INVITE

برای رد کردن بار مازادش استفاده کند. به عبارت دیگر

$$T_{1,2} + (232 - T_{1,2}) \times 0.2 + T_{1,2} = 360 \Rightarrow T_{1,2} = 174.2 \quad (7)$$

$$T_{1,2} + (324 - T_{1,2}) \times 0.2 + 174.2 = 360 \Rightarrow T_{1,2} = 151.2$$

در بخش بعد یک الگوریتم عادلانه max-min ارائه خواهد شد که مسأله بهینه‌سازی (۶) را حل می‌کند. همان طور که مشاهده شد (۶) به علت محاسبه مجزا در پروکسی‌ها و رعایت عدالت، سبب کم‌باری پروکسی‌ها خواهد شد. البته در شرایط واقعی در هنگام اضافه بار، ظرفیت مازاد تابعی از روش کنترل اضافه بار بوده و به نحوی بین پروکسی‌ها تقسیم می‌شود. هدف الگوریتم ارائه شده نزدیک‌بودن به تخصیص مطابق با سیاست رعایت‌شده در ستون سوم جدول ۲ می‌باشد.

## ۴- الگوریتم کنترل اضافه بار

هزینه هر تماس در مسأله بهینه‌سازی (۶) برابر با دیگر تماس‌ها است و بنابراین یک راه ساده و کارا برای رسیدن به عدالت max-min استفاده از سیاست پردازشی نوبت گردشی<sup>۱</sup> در هر پروکسی می‌باشد. با چنین سیاستی جریان‌هایی که بار ورودی بیشتر از سهم عادلانه داشته باشند تنها به خود لطمه می‌زنند. پردازش نوبتی جریان‌ها در هر پروکسی نیازمند یک بخش کلاس‌بندی‌کننده<sup>۲</sup> جریان‌ها است که وظیفه دارد تقاضاها را از صف اصلی پروکسی برداشته و بر اساس نام پروکسی بالادست و پایین‌دست جدا کرده و در صف جریان مربوطه قرار دهد  $(Q_{i,j})$ . بخش مورد نیاز دیگر زمان‌بند<sup>۳</sup> می‌باشد که به صورت نوبت گردشی تقاضاها را از صف جریان‌ها برمی‌دارد.

همان طور که در بخش سوم ذکر شد، در حالت اضافه بار بعضی از پیام‌ها با هزینه  $w_{rej}$  رد می‌شوند و بنابراین دو نوع پیام با وزن ۱ و  $w_{rej}$  (به ترتیب برای تماس‌های موفق و رد شده) وجود دارد و پردازش RR به تنهایی منجر به عدالت max-min نمی‌شود. راهکار استفاده شده برای هزینه متفاوت تقاضاهایی که رد می‌شوند استفاده از اعتبار مجدد<sup>۴</sup> در بخش زمان‌بند پروکسی می‌باشد. به عبارت دیگر هنگامی که یک تقاضا رد می‌شود، پروکسی به جریان مربوطه به اندازه  $1 - w_{rej}$  اعتبار مجدد اختصاص می‌دهد و بعد از این که مجموع اعتبار مجدد یک جریان برابر با ۱ شد به آن یک نوبت جدید تعلق می‌گیرد. این راهکار مشابه با روش DRR در [۲۵] بوده و پیچیدگی محاسباتی  $O(1)$  دارد.

با توجه به توپولوژی مورد مطالعه، امکان دارد که اضافه بار یک جریان ناشی از کمبود ظرفیت در پروکسی بالادست و یا پایین‌دست باشد. سهم اختصاص داده شده به یک جریان در پروکسی بالادست طبق فرمول  $C_{i,j}^{(i)} = T_{i,j} + w_{rej}(\lambda_{i,j} - T_{i,j}) + \lambda_{i,j}kw_{retx}$  و در پروکسی پایین‌دست طبق فرمول  $T_{i,j} = C_{i,j}^{(j)}$  می‌باشد. هدف روش ارائه شده در این مقاله کنترل طول صف و جلوگیری از ارسال مجدد است و بنابراین پارامتر  $k$  را برابر صفر در نظر می‌گیریم (در ادامه در مورد پارامتر  $\alpha$  و مقدار قابل قبول برای جلوگیری از ارسال مجدد نیز صحبت می‌شود). احتمال رد کردن برابر با  $\rho_{i,j} = 1 - T_{i,j}/\lambda_{i,j}$  می‌باشد و بنابراین بسته به این که اضافه بار ناشی از کمبود ظرفیت در پروکسی بالادست و یا پایین‌دست باشد می‌توان احتمال رد کردن را به صورت (۸) نوشت

1. Round Robin
2. Classifier
3. Scheduler
4. Credit

پروکسی ۴۶۰ تماس در ثانیه در نظر گرفته شده است. هر یک از نتایج حاصل میانگین گیری پنج بار آزمایش می‌باشد.

در نمونه نرخ ورودی اول، میانگین گذردهی الگوریتم برای نرخ‌های آزمایش تقریباً مشابه با جواب مسأله بهینه‌سازی می‌باشد و این مورد نشان‌دهنده قابلیت روش ارائه‌شده در شرایط نزدیک به اضافه بار است در حالی که مجموع گذردهی‌ها در روش OCC بسیار پایین‌تر از روش ارائه‌شده است. مهم‌ترین پارامتر این روش  $\rho_{target}$  است که به عنوان میزان بهره‌وری هدف از منابع پردازشی تنظیم می‌شود. مطابق [۸] این پارامتر برابر با ۰.۹ تنظیم شده است. به عبارت ساده در روش OCC در شرایط اضافه بار، پروکسی‌ها حداکثر از ۰.۹ ظرفیت خود استفاده می‌کنند. البته بالاتر بردن این مقدار در شرایطی که بار ناگهانی به پروکسی وارد می‌شود آن را دچار ناپایداری می‌کند.

در نمونه نرخ ورودی دوم، جواب حل (۶) برای هر چهار جریان برابر با ۱۴۱/۶۷ می‌باشد در حالی که گذردهی میانگین به دست آمده از الگوریتم برابر با ۱۴۳ تماس در ثانیه به دست می‌آید. علت بالاتر بودن جواب الگوریتم از حل (۶) همان طور که در بخش ۳-۲ ذکر شد از کم‌باری ناشی از حل محلی مسأله بهینه‌سازی نتیجه می‌شود. به عبارت دیگر حل مسأله بهینه‌سازی در پروکسی ۱ برای هر جریان، سهم ۱۵۳/۳۳ را در نظر می‌گیرد. جریان‌های  $\lambda_{1,r}$  و  $\lambda_{2,r}$  باید از این سهم برای رسیدن به گذردهی ۱۴۱/۶۷ و همچنین رد کردن ۵۸/۳۳ تماس با هزینه ۱۱/۶۷ (هر تماس هزینه ۰/۲) استفاده کنند در حالی که جریان ورودی  $\lambda_{3,r}$  حداکثر می‌تواند از مقدار ۱۴۱/۶۷ از سهم خود استفاده کند ( $C_{r,i}^{(0)} = T_{r,i}$ ) و بنابراین مقدار ۱۱/۶۷ از سهم در نظر گرفته شده در پروکسی ۱ بلااستفاده می‌ماند. گذردهی بالاتر الگوریتم ارائه‌شده ناشی از تقسیم این مقدار بلااستفاده در دو پروکسی به طور مساوی در بین جریان‌ها می‌باشد. همان طور که مشاهده می‌شود روش OCC در این برهه مجموع گذردهی ۴۹۶ دارد در حالی که گذردهی روش ارائه‌شده مقدار ۵۷۲ می‌باشد.

در نمونه سوم که نرخ  $\lambda_{1,r} = ۳۲۴$  است گذردهی به دست آمده از حل مسأله بهینه‌سازی برابر با  $T_{1,r} = ۱۴۴$  می‌باشد. این جواب نزدیک به مقدار به دست آمده الگوریتم ( $T_{1,r} = ۱۳۹$ ) می‌باشد در حالی که روش OCC حتی قادر به کنترل اضافه بار مناسب نشده است. در این بازه زمانی، سه جریان دیگر برای الگوریتم و مسأله بهینه‌سازی گذردهی مشابهی دارند.

بر خلاف روش OCC و اغلب الگوریتم‌های کنترل اضافه بار، روش ارائه‌شده در این مقاله نیاز چندانی به پارامترهای تنظیم عملکرد ندارد و البته برای تخمین بهتر  $\alpha$  می‌توان مقادیر  $D_{RTT}$  و  $T_1$  را تنظیم کرد. پارامتر  $T_1$  عموماً ۵۰۰ میلی‌ثانیه است و مطابق با آزمایش‌های انجام‌شده، تغییر  $D_{RTT}$  تأثیر چندانی در پایداری الگوریتم ندارد هر چند که بالا بودن تأخیر، سبب کنترل محتاطانه‌تر و گذردهی مجموع پایین‌تر می‌شود. نکته قابل توجه در مورد الگوریتم ارائه‌شده در این مقاله این است که بدون حل مسأله بهینه‌سازی، به جواب نزدیک به آن دست می‌یابد. حل مسأله بهینه‌سازی در سناریوهای با تعداد پروکسی و جریان زیاد دارای سربار و پیچیدگی محاسباتی بالا است اما روش ارائه‌شده در این مقاله بسیار ساده می‌باشد و با استفاده از پارامتر طول صف جریان‌ها و دو شمارنده برای نرخ ورودی و گذردهی در کنار استفاده از یک زمان‌بند نوبت‌گردشی در پروکسی از نتیجه حل مسأله (۶) پیروی می‌کند.

در آزمایش دوم می‌خواهیم نحوه تخصیص منابع در زمان‌بند نوبت‌گردشی را با مسأله (۶) مقایسه نماییم. فرض کنید سناریوی آزمایش مطابق با شکل ۱ و فقط پروکسی ۱ دارای اضافه بار باشد. در این آزمایش

### Control Function in Proxy(i)

```

1: for all  $j \in L$  do
2:   if  $T_{i,j} = 0$  then
3:      $T_{i,j} \leftarrow T_{min}$ 
4:   end if
5:    $C_{i,j}^{(i)} \leftarrow T_{i,j} + (\lambda_{i,j} - T_{i,j}) \cdot w_{rej}$ 
6:    $C_{i,j}^{(j)} \leftarrow T_{i,j}$ 
7:   Extract  $Q_{i,j}^{(j)}$  from Proxy(j) feedback
8:    $\rho_{i,j}^{(i)} \leftarrow \frac{1 - [(1+\alpha)C_{i,j}^{(i)} - Q_{i,j}^{(i)}] / \lambda_{i,j}}{1 - w_{rej}}$ 
9:    $\rho_{i,j}^{(j)} \leftarrow 1 - \frac{(1+\alpha)C_{i,j}^{(j)} - Q_{i,j}^{(j)}}{\lambda_{i,j}}$ 
10:   $\rho_{i,j} \leftarrow \max[\rho_{i,j}^{(i)}, \rho_{i,j}^{(j)}]$ 
11:   $\lambda_{i,j} \leftarrow 0$ 
12:   $T_{i,j} \leftarrow 0$ 
13: end for

```

### Processing & Actuator in Proxy(i)

```

1: if Message is INVITE then
2:    $\lambda_{i,j} \leftarrow \lambda_{i,j} + 1$ 
3:   if  $\text{rnd}[0..1] \geq \rho_{i,j}$  then
4:     Accept & Process INVITE request
5:   else
6:     Reject INVITE request
7:   end if
8: else if Message is 200-OK then
9:    $T_{i,j} \leftarrow T_{i,j} + 1$ 
10:  Process 200-OK
11: else
12:  Process message
13: end if

```

شکل ۷: شبه‌کد الگوریتم دنبال‌شده در پروکسی بالادست.

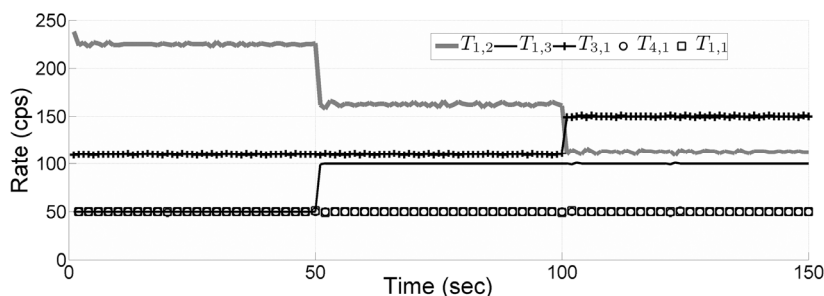
و یا ۲۰۰-OK به ترتیب مقادیر  $\lambda_{i,j}$  و  $T_{i,j}$  را به اندازه یک واحد اضافه می‌کنند.

مطابق با قسمت دوم شکل ۷، بخش بکارانداز روش کنترل اضافه بار تنها بر روی تقاضاهای INVITE اجرا شده و با احتمال  $\rho_{i,j}$  تقاضای جدید را رد می‌کند. در صورت پذیرش یک INVITE، پروکسی پردازش آن و بقیه پیام‌های مربوط به آن تماس را به عهده می‌گیرد. بخش ناظر پروکسی نیز مکانیزم ساده‌ای دارد و از نمایش شبه‌کد آن صرف نظر کرده‌ایم. این بخش در هر برهه طول صف جریان مربوطه را در سرآیند VIA پیام‌های پاسخ (خصوصاً Trying-۱۰۰ و یا Ringing-۱۸۰) تعبیه می‌کند (مطابق با [۹]).

به منظور بررسی روش ارائه‌شده، توپولوژی شکل ۱ را در نرم‌افزار ns شبیه‌سازی کرده‌ایم. در این شبیه‌سازی، پیاده‌سازی کاربران و پروکسی‌های شبکه SIP مطابق با RFC-۳۲۶۱ [۱] انجام شده است. علاوه بر پیاده‌سازی روش ارائه‌شده، روش کنترل اضافه بار OCC نیز به منظور مقایسه کارایی، شبیه‌سازی شده است. پیاده‌سازی روش OCC و پارامترهای استفاده‌شده در آن مطابق با [۸] انجام شده است.

در جدول ۳ کارایی روش ارائه‌شده، روش OCC و حل مسأله بهینه‌سازی (۶) به ازای سه مدل ترافیک مختلف مقایسه شده است. توپولوژی آزمایش مطابق با شکل ۳ می‌باشد و همچنین ظرفیت هر دو





شکل ۸: بررسی کارایی الگوریتم ارائه‌شده در سناریوی شکل ۱ هنگامی که فقط پروکسی ۱ دارای اضافه بار باشد و نرخ‌های ورودی مطابق با جدول ۴ باشند.

جدول ۳: مقایسه کارایی الگوریتم ارائه‌شده با روش OCC و حل مسأله بهینه‌سازی.

$\lambda$	الگوریتم ارائه‌شده	روش OCC	حل مسأله بهینه‌سازی (۶)
$\lambda = \begin{bmatrix} 100 & 180 \\ 180 & 100 \end{bmatrix}$	$T = \begin{bmatrix} 100 & 179 \\ 179 & 100 \end{bmatrix}$ $T_{Overall} = 558$	$T = \begin{bmatrix} 89 & 160 \\ 160 & 89 \end{bmatrix}$ $T_{Overall} = 498$	$T = \begin{bmatrix} 100 & 180 \\ 180 & 100 \end{bmatrix}$ $T_{Overall} = 558$
$\lambda = \begin{bmatrix} 200 & 200 \\ 200 & 200 \end{bmatrix}$	$T = \begin{bmatrix} 143 & 143 \\ 143 & 143 \end{bmatrix}$ $T_{Overall} = 572$	$T = \begin{bmatrix} 124 & 124 \\ 124 & 124 \end{bmatrix}$ $T_{Overall} = 496$	$T = \begin{bmatrix} 141,67 & 141,67 \\ 141,67 & 141,67 \end{bmatrix}$ $T_{Overall} = 566,66$
$\lambda = \begin{bmatrix} 100 & 324 \\ 180 & 100 \end{bmatrix}$	$T = \begin{bmatrix} 100 & 139 \\ 180 & 100 \end{bmatrix}$ $T_{Overall} = 519$	$T_{Overall} < 420$	$T = \begin{bmatrix} 100 & 144 \\ 180 & 100 \end{bmatrix}$ $T_{Overall} = 524$

جدول ۵: گذردهی حاصل از حل مسأله بهینه‌سازی (۶).

$T_{1,1}$	$T_{1,2}$	$T_{1,3}$	$T_{3,1}$	$T_{4,1}$	زمان (ثانیه)
۵۰	۲۲۵	۵۰	۱۱۰	۵۰	۵۰-۱
۵۰	۱۶۲,۵	۱۰۰	۱۱۰	۵۰	۱۰۰-۵۰
۵۰	۱۱۲,۵	۱۰۰	۱۵۰	۵۰	۱۵۰-۱۰۰

جدول ۴: نرخ ورودی آزمایش در یک پروکسی با پنج جریان.

$\lambda_{1,1}$	$\lambda_{1,2}$	$\lambda_{1,3}$	$\lambda_{3,1}$	$\lambda_{4,1}$	زمان (ثانیه)
۵۰	۳۰۰	۵۰	۱۱۰	۵۰	۵۰-۱
۵۰	۳۰۰	۱۰۰	۱۱۰	۵۰	۱۰۰-۵۰
۵۰	۳۰۰	۱۰۰	۲۰۰	۵۰	۱۵۰-۱۰۰

قبل بسیار دقیق عمل می‌کند و به عدالت max-min منتهی می‌شود. همچنین عملکرد روش کنترل اضافه بار ارائه‌شده نیز در کارکرد آن خلی ایجاد نمی‌کند. به عنوان مثال در برهه ۵۰-۱۰۰ که تنها جریان  $\lambda_{1,3}$  بیشتر از سهم خود بار ورودی دارد، گذردهی در اطراف ۱۶۲/۵ دریافت می‌کند و چهار جریان دیگر که نرخ ورودی کمتر از سهم دارند گذردهی کامل و بدون نوسان دریافت می‌کنند. به عبارتی سیاست نوبت گردشی تنها وقتی نوبت این جریان‌ها را به جریان  $\lambda_{1,3}$  اختصاص داده است که طول صف آنها برابر با صفر بوده و تقاضایی برای پردازش نداشته‌اند. در برهه سوم (۱۰۰-۱۵۰)، جریان  $\lambda_{1,1}$  به نرخ ۲۰۰ تماس در ثانیه و بیشتر از سهم خود افزایش می‌یابد و بنابراین سهم جریان‌های  $\lambda_{1,2}$  و  $\lambda_{1,3}$  برابر با ۱۵۰ می‌باشد. در این برهه با توجه به این که رد کردن بار مازاد جریان  $\lambda_{1,1}$  در پروکسی ۳ انجام می‌شود (که اضافه بار ندارد)، این جریان به گذردهی برابر با سهم اختصاصی در پروکسی ۱ دست پیدا می‌کند (۱۵۰) در حالی که جریان  $\lambda_{1,3}$  به گذردهی ۱۱۲/۵ دست پیدا کرده و پروکسی ۱ مقدار سهم ۳۷/۵ را صرف رد کردن ۱۸۷/۵ تقاضای مازاد بر ظرفیت می‌کند.

## ۵- نتیجه‌گیری

در این مقاله مسأله بهینه‌سازی شبکه‌های مبتنی بر SIP در قالب سناریوی با یک یا دو پروکسی میانی مدل شده که پیچیده‌ترین توپولوژی مطالعه‌شده در بحث کنترل اضافه بار SIP می‌باشد. همچنین با توجه به این که سربار رد کردن تقاضاهای مازاد بر ظرفیت، قابل چشم‌پوشی نیست در مسأله بهینه‌سازی اضافه بار و مورد بررسی قرار گرفته است.

نرخ جریان‌های عبوری از پروکسی ۱ مطابق با جدول ۴ و ظرفیت پروکسی ۱ برابر با ۵۰۰ تماس در ثانیه تنظیم شده است. در جدول ۵ گذردهی حاصل از حل مسأله بهینه‌سازی (۶) توسط ابزار CVX در هر یک از سه برهه آزمایش نشان داده شده است. همان طور که ذکر شد جواب حل مسأله بهینه‌سازی (۶) به عدالت max-min منتهی خواهد شد. به عبارتی پروکسی ۱ به منظور رعایت سهم عادلانه جریان‌ها برای هر یک سهم فرضی  $(C_i/L)$  را در نظر می‌گیرد، هر چند که تمامی جریان‌ها به اندازه سهم فرضی نرخ ورودی ندارند. بنابراین ابتدا به جریان‌هایی که کمتر از سهم فرضی نرخ ورودی دارند گذردهی کامل را اختصاص می‌دهد و با ظرفیت باقیمانده به جریان‌های دیگر که بیشتر از سهم فرضی نرخ ورودی دارند به مقدار مساوی سهم اختصاص می‌دهد. به عنوان مثال در برهه ۵۰-۱۰۰ آزمایش، سهم فرضی اولیه هر جریان برابر با ۱۰۰ می‌باشد. هر چند که سه جریان  $\lambda_{1,1}$ ،  $\lambda_{1,2}$ ،  $\lambda_{1,3}$  نرخ کمتر و یا برابر با این سهم دارند. بنابراین به گذردهی کامل دست پیدا می‌کنند و باقیمانده سهم آنها به دیگر جریان‌ها تعلق می‌گیرد. ظرفیت باقیمانده بعد از تخصیص به این سه جریان برابر با ۳۰۰ و بنابراین سهم فرضی هر یک از دو جریان دیگر برابر با ۱۵۰ می‌باشد. هر چند که جریان  $\lambda_{1,1}$  نیز بار ورودی ۱۱۰ و کمتر از سهم فرضی محاسبه‌شده در این مرحله دارد و بنابراین گذردهی کامل دریافت می‌کند و در انتها ظرفیت باقیمانده ۱۹۰ به جریان  $\lambda_{1,3}$  اختصاص می‌یابد که صرف پذیرش ۱۶۲/۵ تماس در ثانیه و رد کردن ۱۳۷/۵ تماس با مجموع هزینه ۲۷/۵ می‌شود.

در شکل ۸ نتیجه شبیه‌سازی برای این آزمایش نشان داده شده است. طبق این شکل، سرویس‌دهی نوبت گردشی با جزئیات ذکرشده در بخش

*Communications and Signal Processing, ICCSP'15*, pp. 1296-1300, Melmaruvathur, India, 2-4 Apr. 2015.

- [16] G. Mishra, S. Dharmaraja, and S. Kar, "Reducing session establishment delay using timed out packets in SIP signaling network," *International J. of Communication Systems*, vol. 29, no. 2, pp. 262-276, Jan. 2016.
- [17] B. Upadhyay, A. Mishra, and S. B. Upadhyay, "AIPC: counteractive analysis of overload control mechanism for sip server," *International J. of Computer Engineering & Technology*, vol. 1, no. 1, pp. 128-140, Jan. 2014.
- [18] J. Liao, et al., "A distributed end-to-end overload control mechanism for networks of SIP servers," *The International J. of Computer and Telecommunications Networking*, vol. 56, no. 12, pp. 2847-2868, Aug. 2012.
- [19] M. Jahanbakhsh, S. V. Azhari, J. Enayati-Zadeh, and M. Baghdadi, "Local and distributed SIP overload control solution improving sustainability of sip networks," *Int. J. Commun. Syst.*, vol. 30, 27 pp., Apr. 2017.
- [20] J. Wang, J. Liao, T. Li, J. Wang, J. Wang, and Q. Qi, "Probe-based end-to-end overload control for networks of SIP servers," *J. of Network and Computer Applications*, vol. 41, pp. 114-125, May 2014.
- [21] M. Homayouni, H. Nemati, V. Azhari, and A. Akbari, "Controlling overload in SIP proxies: an adaptive window based approach using no explicit feedback," in *Proc. IEEE Global Communications Conf., GLOBECOM'10*, 5 pp. Miami, FL, USA, 6-10 Dec. 2010.
- [22] A. Coluccia, A. D. Alconzo, and F. Ricciato, "On the optimality of max-min fairness in resource allocation," *Annals of Telecommunications*, vol. 67, no. 1-2, pp. 15-26, Feb. 2012.
- [23] S. V. Azhari and H. Nemati, "Stability analysis of tandem SIP proxies," in *Proc. IEEE Int Conf. on Communications, ICC'12*, pp. 1244-1248, Ottawa, ON, Canada, 10-15 Jun. 2012.
- [24] M. Grant and S. Boyd, *CVX: MATLAB software for Disciplined Convex Programming*, <http://stanford.edu/~boyd/cvx>
- [25] M. Shreedhar and G. Varghese, "Efficient fair queueing using deficit round-robin," *IEEE/ACM Trans. on Networking*, vol. 4, no. 3, pp. 375-385, Jun. 1996.

**مجتبی جهانبخش** در سال ۱۳۸۵ مدرک کارشناسی مهندسی کامپیوتر خود را از دانشگاه اصفهان و در سال ۱۳۸۸ مدرک کارشناسی ارشد خود را از دانشگاه علم و صنعت ایران دریافت نمود و در حال تحصیل در مقطع دکترا در این دانشگاه می‌باشد. زمینه‌های تحقیقاتی مورد علاقه ایشان شامل پروتکل‌ها و سرویس‌های وابسته به سیستم‌های چندرسانه‌ای، کنترل اضافه بار و اینترنت اشیا است. ایشان توجه خاصی به ارزیابی عملیاتی راه‌کارهای ارائه شده دارد که حاصل آن توسعه چندین بستر آزمایش همه منظوره در محیط دانشگاهی است.

**سید وحید ازهری** استادیار دانشکده کامپیوتر دانشگاه علم و صنعت ایران می‌باشد. وی دکترای تخصصی خود را در رشته مهندسی برق و کامپیوتر از دانشگاه مک‌مستر کانادا در سال ۱۳۸۶ دریافت نمود. زمینه تخصصی وی شبکه‌های بی‌سیم، شبکه‌های کامپیوتری و رایانش سبز می‌باشد. ایشان بالغ بر ۱۰ سال است که در زمینه پروتکل SIP مشغول فعالیت پژوهشی می‌باشد. حاصل فعالیت‌های ایشان در این حوزه شامل الگوریتم‌های متعدد کنترل اضافه بار و همچنین توسعه اولین بستر تست IMS در ایران است.

**وحید قاسم‌خانی** در سال ۱۳۸۹ مدرک کارشناسی مهندسی نرم افزار خود را از دانشکده فنی شمسی‌پور تهران و در سال ۱۳۹۲ مدرک کارشناسی ارشد شبکه‌های کامپیوتری خود را از دانشگاه علم و صنعت ایران دریافت نمود. نام‌برده موفق شد در هر دو دوره تحصیلی مذکور به عنوان دانشجوی ممتاز فارغ التحصیل شود. وی با توجه به داشتن علاقه زیاد در زمینه شبکه‌های کامپیوتری و سیستم‌های چند رسانه‌ای بر بستر IP بلافاصله وارد این صنعت شد و هم‌اکنون به عنوان مدیر طراحی و توسعه سیستم‌های IPTV/OTT فعالیت می‌کند.

حداکثرسازی مجموع گذردهی در اغلب سناریوها منجر به لطمه خوردن به ترافیک بین دامنه‌ای پروکسی می‌شود و بنابراین در ادامه به منظور اضافه نمودن عدالت max-min به بررسی تحلیلی تابع هدف حداقل مربعات پرداخته شد و یک روش کنترل اضافه بار با عدالت max-min معرفی شد که روشی وفق‌پذیر بوده و برخلاف روش‌هایی مثل OCC در انواع توپولوژی‌ها از جمله توپولوژی پیچیده مطرح شده در این مقاله کارایی مناسبی دارد. از کارهای آتی در راستای بهینه‌سازی کنترل اضافه بار در شبکه‌های مبتنی بر SIP، بررسی دقیق‌تر عدالت سراسری max-min و مطالعه بر روی دیگر توابع هدف جالب مثل میانگین هندسی، جهت رعایت حداقل گذردهی تعیین شده برای جریان‌ها می‌باشد.

## مراجع

- [1] J. Rosenberg, et al., *SIP: Session Initiation Protocol*, IETF (RFC 3261), Jun. 2002.
- [2] J. Rosenberg, *Requirements for Management of Overload in the Session Initiation Protocol*, RFC-5390 (Informational), Dec. 2008.
- [3] E. Noel and C. Johnson, "Novel overload controls for SIP networks," in *Proc. 21st Int. Teletraffic Congress*, 8 pp., Sept. 2009.
- [4] M. Ohta, "Overload control in a SIP signaling network," *International Journal of Computer, Electrical, Automation, Control and Information Engineering*, vol. 1, no. 12, pp. 4076-4081, Dec. 2007.
- [5] S. Montagna and M. Pignolo, "Load control techniques in SIP signaling servers using multiple thresholds," in *Proc. the 13th Int. Telecommunications Network Strategy and Planning Symp.*, 6 pp., Sept. 2008.
- [6] M. Khazaei and N. Mozayani, "A dynamic distributed overload control mechanism in SIP networks with holonic multi-agent systems," *Telecommun Syst.*, DOI 10.1007/s11235-015-0133-3, Dec. 2015.
- [7] L. De Cicco, G. Cofano, and S. Mascolo, "Local SIP overload control: controller design and optimization by extremum seeking," *IEEE Trans. on Control of Network Systems*, vol. 2, no. 3, pp. 267-277, Feb. 2015.
- [8] V. Hilt and I. Widjaja, "Controlling overload in networks of SIP servers," in *Proc. of IEEE Int. Conf. on Network Protocols, ICNP'08*, pp. 83-93, Orlando, FL, USA, 19-22 Oct. 2008.
- [9] V. Gurbani, V. Hilt, and H. Schulzrinne, *Session Initiation Protocol (SIP) Overload Control*, Internet Engineering Task Force (IETF), RFC-7339, ISSN: 2070-1721, Sep. 2014.
- [10] E. Noel and P. M. Williams, "Session Initiation Protocol (SIP) Rate Control," Internet Engineering Task Force (IETF), RFC-7415, Feb. 2015.
- [11] C. Shen, H. Schulzrinne, and E. Nahum, "Session initiation protocol (SIP) server overload control: design and evaluation," in *Principles, Systems and Applications of IP Telecommunications, Services and Security for Next Generation Networks*, Springer, pp. 149-173, 2008.
- [12] M. Chiang, S. H. Low, A. R. Calderbank, and J. C. Doyle, "Layering as optimization decomposition: a mathematical theory of network architectures," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 255-312, Jan. 2007.
- [13] F. P. Kelly, A. Maulloo, and D. Tan, "Rate control for communication networks: shadow prices, proportional fairness and stability," *J. Operations Res. Soc.*, vol. 49, no. 3, pp. 237-252, Mar. 1998.
- [14] R. G. Garroppo, S. Giordano, S. Niccolini, and S. Spagna, "A prediction-based overload control algorithm for SIP servers," *IEEE Trans. on Network and Service Management*, vol. 8, no. 1, pp. 39-51, Mar. 2011.
- [15] A. Akbar, S. Mahaboob Basha, and S. A. Satar, "A cooperative overload control method for SIP servers," in *Proc. Int. Conf. on*