علوم رایانش و فناوری اطلاعات
نشریه علمی انجمن کامپیوتر ایران
مجلد ۲۲، شماره ۱، ۱۴۰۳
صفحات ۲۰-۳۰
**مقاله پژوهشی**

# تجزیه‌وتحلیل سن اطلاعات و تأخیر برای ذخیره‌سازی جزئی لبه‌ای - ابری

مهران رهنمانیا[۱]، فرید آشتیانی[۲*]

*نویسنده مسئول، دریافت: ۰۳/۰۳/۳۰، پذیرش: ۰۳/۰۸/۰۷

[۱] دانشجوی کارشناسی ارشد، مهندسی برق مخابرات سیستم، دانشگاه صنعتی شریف، تهران، ایران
[۲] دانشیار، دانشکده مهندسی برق ، دانشگاه صنعتی شریف، تهران، ایران

## چکیده

در مقاله حاضر، یک سیستم ذخیره‌سازی ساده لبه‌ای - ابری شامل یک حافظه در ابر (تأمین‌کننده محتوا) و یک حافظه پنهان در لبه (ایستگاه پایه)، همراه با ذخیره‌سازی جزئی برای محتویات پویا را در نظر می‌گیریم. در صورت درخواست فایل توسط هر کاربر، اجزاء آن به سمت کاربر ارسال می‌شوند، به‌این‌ترتیب که جزئی که در حافظه پنهان است، مبتنی بر مکانیسم نشاندن (push-based) (قرار داشتن در حافظه پنهان و به‌روزرسانی مرتب آن) و جزء دیگر مبتنی بر مکانیسم کشیدن (pull-based) (واکشی از حافظه ابری توسط ایستگاه پایه و ارسال به سمت کاربر نهایی) در اختیار کاربر قرار می‌گیرد. بنابراین نیمی از فایل‌ها در حافظه پنهان قرار می‌گیرند و به‌صورت تصادفی به‌روز می‌شوند و نیمه دیگر باید قبل از ارسال به کاربر واکشی شوند. متوسط تأخیر و میانگین سن اطلاعات را برای سه طرح ذخیره‌سازی در حافظه پنهان و تحویل استخراج می‌کنیم: ذخیره‌سازی کامل تحویل کامل (WCWD)، ذخیره‌سازی جزئی تحویل کامل (PCWD) و ذخیره‌سازی جزئی تحویل جزئی (PCPD). برای این منظور، یک مدل شبکه صف با خاصیت ضربی به همراه ورودی‌های اضافی پیشنهاد می‌کنیم تا ورودی‌های وابسته ناشی از واکشی هم‌زمان و همچنین بازیابی موازی هر دو جزء فایل‌ها (درنتیجه تقسیم فایل و ذخیره جزئی) را مدل نماییم. شبیه‌سازی‌ها دقت بالای مدل تحلیلی را نشان می‌دهد. خواهیم دید که طرح PCPD برتر از سایر طرح‌ها است.

**کلمات کلیدی: حافظه پنهان، ذخیره‌سازی جزئی، سن اطلاعات، تأخیر، شبکهٔ صف.**

## ۱- مقدمه

امروزه، در دنیایی که بیشتر ابزارها از طریق شبکه‌های مختلف به یکدیگر متصل هستند، دسترسی فوری به داده‌ها اهمیت زیادی پیداکرده است که در آن حافظه پنهان [۱] می‌تواند نقش مهمی در کاهش تأخیر [۲] و بار شبکه‌ها [۳] به‌منظور افزایش کیفیت خدمات [۴] ایفا کند. ذخیره‌سازی داده‌ها در حافظه پنهان، روشی است که در آن محتوایی که محبوبیت و تقاضای بیشتری دارد را در نزدیکی کاربر نهایی ذخیره می‌کند تا همان‌طور که گفته شد تأخیر دسترسی به محتوای موردنظر برای کاربر نهایی و همچنین ترافیک کلی شبکه کاهش یابد.

امروزه به دلیل افزایش تصاعدی ترافیک داده‌های تلفن همراه (ناشی از وجود میلیاردها دستگاه متصل به این شبکه‌ها)، چالش‌های بسیاری برای ذخیره‌سازی داده‌ها در حافظه‌های پنهان در سطح شبکه‌های بی‌سیم به وجود آمده است [۱, ۲]. ازاین‌رو بسیاری از مقاله‌ها به دنبال راه‌های نوآورانه‌ای برای طراحی الگوریتم‌های ذخیره‌سازی به جهت پیش‌بینی و برآورده کردن نیازهای آینده کاربران می‌باشند. برخی مقاله‌ها نیز به بررسی مشکلات سنتی ذخیره‌سازی بر روی حافظه پنهان، قابلیت‌ها و محدودیت‌های بک‌هال [۵]، همکاری ایستگاه‌های پایه [۶]، فن‌های کدگذاری شده/غیر کدگذاری شده، تحرک کاربران [۷]، مسائل اقتصادی، و قرار دادن محتوا در

لبه شبکه‌های بی‌سیم[۸] (یعنی در ایستگاه‌های پایه و دستگاه‌های کاربر) توجه نموده‌اند [۳-۶].

اگرچه در تحقیقات قبلی روی حافظه پنهان، به‌روزرسانی فرایندی است که درزمانی که ترافیک شبکه کم است انجام می‌شود، اما برای محتوای پویا[۹]، وضعیت متفاوت است. اخیراً برخی از مقاله‌ها، محتوای پویا و همچنین فایل‌های محبوب بسیار پویا را در نظر گرفته‌اند، بنابراین برای تأمین کردن فایل‌های پویای مورد درخواست ، لازم است حافظه پنهان هم‌زمان با تحویل محتوا به کاربر به‌روزرسانی شود [۷-۱۳]. درواقع، برای محتوای بسیار پویا مانند برخی از وب‌سایت‌ها، اخبار ورزشی دربازی‌های المپیک، نتایج رأی‌گیری در انتخابات ریاست جمهوری، وضعیت بورس و غیره، تازگی اطلاعات یکی از ویژگی‌های مهم و تأثیرگذار برای اتخاذ سیاست مناسب ذخیره‌سازی در حافظه پنهان است. از سوی دیگر، اهمیت سامانه‌های کنترل شبکه‌ای در فناوری‌های جدید مانند وسایل نقلیه بدون راننده، شبکه‌های اینترنت اشیاء و غیره، که در آن‌ها اطلاعات تازه و به‌روز در فرآیندهای تصمیم‌گیری بسیار حیاتی است، روزبه‌روز در حال افزایش است. ازاین‌رو تازگی اطلاعات را می‌توان بعد جدیدی در طراحی الگوریتم‌های ذخیره‌سازی در نظر گرفت. در [۱۴] معیار سن اطلاعات[۱۰] برای اندازه‌گیری تازگی داده‌ها معرفی‌شده است که برابر بازمان سپری‌شده از لحظه تولید یا به‌روزرسانی اطلاعات (که با $\mathrm{u}(t)$ نشان داده می‌شود) تا زمان فعلی (که با t نشان داده می‌شود) است یعنی ، $\Delta(t) \triangleq t - \mathrm{u}(t)$.

برخی از تحقیقات انجام‌شده، به ذخیره‌سازی محتوای پویا در حافظه پنهان و همچنین پرداختن به مشکلات کهنگی داده اختصاص دارند. چند مقاله با معرفی تابع‌های هزینه‌ای مرتبط با واکشی[۱۱] داده‌ها و سن اطلاعات، به بررسی و بهینه کردن آن‌ها در قالب مسائل بهینه‌سازی پرداخته‌اند [۸, ۹]. علاوه بر این، برخی از محققین از تئوری صف استفاده کرده و سعی کرده‌اند سناریوهای ذخیره‌سازی مختلف را مدل و سن اطلاعات را محاسبه کنند [۱۲, ۱۳]. از سوی دیگر، سامانه‌های ذخیره‌سازی توزیع‌شده[۱۲] برای افزایش عملکرد حافظه پنهان و سرعت بازیابی داده‌ها با ارسال یک فایل از طریق مسیرهای متعدد به کاربر نهایی، پیشنهادشده است. در این راستا، برای کاهش تأخیر، فن‌های مختلف شامل درخواست‌های اضافی[۱۳] داده‌ها، تکثیر[۱۴] یا ذخیره‌سازی کد شده و غیره می‌توانند فایل‌ها را از منابع ذخیره‌سازی توزیع‌شده برای کاربران ارسال کنند (به‌عنوان‌مثال، [۱۵, ۱۶]).

در اکثر آثار گزارش‌شده که در آن از حافظه پنهان و ذخیره‌سازی کد نشده استفاده‌شده است، فایل‌ها به‌طور کامل در حافظه پنهان ذخیره می‌شوند. بااین‌حال، در برخی از آثار، فایل‌ها به قسمت‌های[۱۵] کوچک‌تری تقسیم می‌شوند، و از میان آن‌ها، تنها تعداد کمی در حافظه پنهان ذخیره می‌شوند (به‌عنوان‌مثال، [۱۷-۱۹]). این طرح‌ها می‌توانند در عمل کارآمدتر باشند، زیرا در بیشتر موقعیت‌هایی که یک فایل، به‌عنوان‌مثال یک ویدیو درخواست می‌شود، برخی از قسمت‌های آن دارای اهمیت بیشتری می‌باشند. به‌طور مثال برخی اوقات پس از مشاهده قسمت‌های اولیه، درخواست‌کننده تصمیم می‌گیرد که باقیمانده فایل درخواستی را رها کند. بنابراین، به نظر می‌رسد که ذخیره‌سازی تنها برخی از قسمت‌های فایل‌های محبوب در حافظه پنهان کارآمدتر است. اگرچه تعداد کمی از کارها بر روی طراحی الگوریتم‌های ذخیره‌سازی جزئی در حافظه پنهان[۱۶] متمرکزشده‌اند، تا جایی که می‌دانیم، تحلیل عملکرد در شرایطی که درخواست‌ها به‌صورت تصادفی ارسال می‌شوند (که منجر به تأخیر در صف می‌شود) هنوز گزارش نشده است. علاوه بر این، کارایی آن‌ها برای فایل‌های بسیار پویا مشخص نیست. واضح است که هنگام ذخیره کردن محتوای بسیار پویا در حافظه پنهان، ضروری است که آن را به‌طور مرتب به‌روز کنیم تا محتوای ذخیره‌شده به‌اندازه کافی تازه بماند. از سوی دیگر برای آنکه از مزیت‌های ذخیره‌سازی در حافظه پنهان (به‌طور ویژه صرفه‌جویی در ترافیک بک‌هال) بهره‌مند گردیم، به‌روزرسانی را باید به حداقل برسانیم.

از این نظر، دو طرح کلی برای به‌روزرسانی حافظه پنهان وجود دارد: بر پایه مکانیسم کشیدن[۱۷]، بر پایه مکانیسم نشاندن[۱۸] [۱۲]. در طرح اول، هنگام درخواست یک فایل، ممکن است آن فایل از تأمین‌کننده محتوا[۱۹] واکشی شود (به‌عنوان‌مثال، زمانی که سن آن بزرگ‌تر از یک آستانه است [۱۲]) و از طریق ایستگاه پایه برای کاربر ارسال شود. بدیهی است که در بازیابی مرسوم فایل (به‌عنوان‌مثال در حالت بدون حافظه پنهان)، به همه درخواست‌ها به‌نوعی بر پایه مکانیسم کشیدن پاسخ داده می‌شود. بااین‌حال، در طرح دوم، فایل‌های به‌روز شده به‌طور منظم به حافظه پنهان فرستاده می‌شوند تا جای نسخه‌های قدیمی‌تر را بگیرند (یعنی بر پایه مکانیسم نشاندن)، بنابراین فایل‌های درخواستی کاربر مستقیماً از حافظه پنهان موجود در ایستگاه پایه ارسال می‌شوند.

ما در مقاله حاضر یک سناریوی ساده برای ذخیره‌سازی ابری-لبه‌ای[۲۰] همراه با سه طرح ذخیره‌سازی را در نظر می‌گیریم. ما برای محاسبه تأخیر و سن اطلاعات یک مدل تحلیلی جدید با الهام گرفتن از کار قبلی خود [۲۰] با استفاده از یک شبکه صف با ورودی‌های اضافی[۲۱] که خاصیت ضربی[۲۲] را حفظ می‌کند پیشنهاد می‌دهیم. سپس میزان سن اطلاعات و تأخیر طرح‌های ذخیره‌سازی را با استفاده از مدل تحلیلی ارائه‌شده به دست آورده و آن را با طرح ذخیره‌سازی سنتی که شامل ذخیره‌سازی کامل فایل است، مقایسه می‌کنیم. چالش اصلی که در تحلیل وجود دارد، ورودهای هم‌زمان موجود درون شبکه صف معادل است که به علت بازیابی هم‌زمان بخش‌های مختلف یک فایل از ابر و لبه به وجود می‌آید.

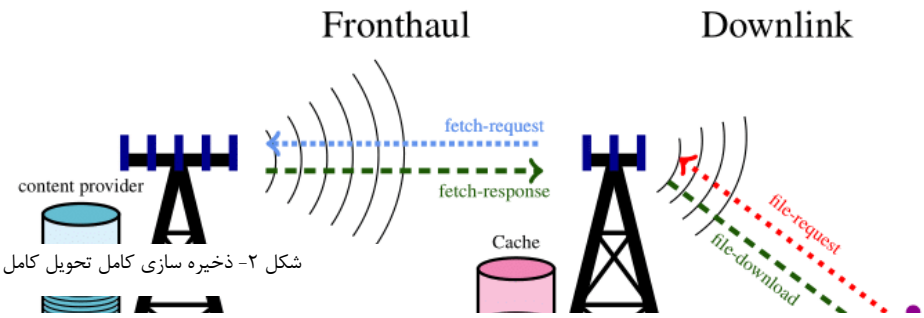
کارهای اصلی این مقاله به شرح زیر است:

- یک سناریوی ذخیره‌سازی ابری - لبه‌ای را در نظر می‌گیریم که شامل تأمین‌کننده محتوا در ابر و یک حافظه پنهان در ایستگاه پایه است. علاوه بر این، سه طرح ساده ذخیره‌سازی (تا حدودی مشابه طرح‌های [۱۹]) در نظر گرفته می‌شود، یعنی ذخیره‌سازی کامل تحویل کامل[۲۳](WCWD)، ذخیره‌سازی جزئی تحویل کامل[۲۴] (PCWD) و ذخیره‌سازی جزئی تحویل جزئی[۲۵] (PCPD). اولین طرح برای نصف فایل‌ها (یعنی ذخیره‌شده در تأمین‌کننده محتوا) بر پایهٔ مکانیسم کشیدن عمل می‌کند و برای نصف دیگر (یعنی ذخیره‌شده در حافظه پنهان در ایستگاه پایه) بر پایهٔ مکانیسم نشاندن عمل می‌کند. در دو طرح ذخیره‌سازی دیگر (ذخیره‌سازی جزئی) که از یک طرح ترکیبی بر پایهٔ هر دو مکانیسم کشیدن و نشاندن[۲۶] پیروی می‌کند، هر فایل به دو جزء تقسیم می‌شود به‌گونه‌ای که یکی از اجزاء بر اساس مکانیسم نشاندن و دیگری بر اساس مکانیسم کشیدن در اختیار کاربر قرار می‌گیرد.

- میانگین تأخیر و سن اطلاعات را برای طرح‌های ذخیره‌سازی فوق به دست می‌آوریم. به این منظور، یک مدل شبکه صف خاص را پیشنهاد می‌دهیم که قادر به مدل‌سازی ورودی‌های وابسته (ورودی‌های هم‌زمان در صف‌های مختلف) و تأثیر آن‌ها بر صف‌های دیگر است.

- با استفاده از شبیه‌سازی و نتایج عددی مدل تحلیلی نشان می‌دهیم که مدل تحلیلی ارائه‌شده از دقت بالایی برخوردار است. همچنین خواهیم دید که طرح PCPD در کاهش تأخیر و سن اطلاعات بهتر از طرح‌های دیگر عمل می‌کند.

ساختار این مقاله به‌صورت زیر سازمان‌دهی شده است: بخش دوم مدل سیستم را معرفی می‌کند. در بخش سوم، ما مدل تحلیلی‌مان را پیشنهاد می‌نماییم و توضیح می‌دهیم که چگونه می‌تواند انواع مختلف وابستگی‌هایی را که در طرح‌های




شکل ۱- سناریوی مورد بررسی در مقاله.

ذخیره‌سازی وجود دارند، در نظر بگیرد. بخش چهارم با استفاده از مدل تحلیلی پیشنهادی، تأخیر و سن اطلاعات را تجزیه‌وتحلیل می‌کنیم. نتایج عددی در بخش پنجم ارائه می‌شود.

## ۲- مدل سیستم

ما بر روی یک شبکه متشکل از یک ایستگاه پایه که مجهز به حافظه پنهان است و یک تأمین‌کننده محتوا از راه دور (به‌طور مثال، سرور مرکزی و یا سرور ابری) متمرکز می‌شویم که در آن برخی از فایل‌ها را می‌توان نزدیک‌تر به کاربر نهایی در حافظه پنهان در ایستگاه پایه ذخیره کرد تا تأخیر دستیابی به محتوا را به حداقل برسانیم. شایان‌ذکر است که تأمین‌کننده محتوا ممکن است به‌عنوان یک حافظه پنهان ابری در نظر گرفته شود که بسیار نزدیک به سرور اصلی قرارگرفته است (به شکل ۱ مراجعه شود).

یک مجموعه $F = \{1,2,3,\dots,2N\}$ از فایل‌های متمایز و بسیار پویا با اندازه‌های یکسان در نظر می‌گیریم. اگرچه ممکن است برخی فایل‌های دیگر نیز توسط کاربران درخواست شوند، ولی ما تمرکز خود را روی مجموعه $F$ که فقط حاوی فایل‌هایی با محتوای بسیار پویا است، می‌گذاریم. حافظه پنهان قادر است $N$ فایل کامل را از میان فایل‌های بسیار پویا ذخیره کند. برای سادگی، فرض بر این است که محبوبیت فایل‌ها در $F$ یکسان است. بنابراین، فرآیند رسیدن درخواست فایل‌های مختلف به‌عنوان فرآیندهای پواسون[۲۷] مستقل با نرخ‌های برابر $\frac{\lambda}{2N}$ مدل‌سازی می‌شود. هر فایل به دو جزء مساوی تقسیم می‌شود. همان‌طور که در بخش قبل بحث شد، یک توجیه پشت سر این نوع تقسیم‌بندی این است که گاهی اوقات پس از دریافت جزء اولیه فایل، ممکن است که کاربر درخواست‌کننده، دیگر نیازی به‌جزءهای دیگر فایل نداشته باشد. بااین‌حال، فرض می‌کنیم زمانی که کاربر درخواست فایلی را می‌کند، هردو جزء آن باید توسط شبکه به سمت کاربر ارسال شوند.

باتوجه‌به پویایی بالای محتوای فایل‌ها، فرض بر این است که فایل‌های قرار داده‌شده در حافظه پنهان موجود در ایستگاه پایه، صرف‌نظر از درخواست‌های مربوطه، به‌صورت مستقل به‌روزرسانی می‌شوند. همچنین فرض می‌کنیم که این فایل‌ها به روش چرخشی[۲۸] و با نرخ یکسان به‌روز می‌شوند، به‌گونه‌ای که زمان لازم برای هر به‌روزرسانی از یک متغیر تصادفی نمایی[۲۹] پیروی می‌کند. هدف ما این است که برخی از فایل‌ها را در حافظه پنهان در نزدیکی کاربر نهایی قرار دهیم تا بتوانیم تأخیر مرتبط با دریافت فایل توسط کاربر نهایی که به دلیل بازیابی محتوا از یک تأمین‌کننده محتوا در راه دور ناشی می‌شود، را به حداقل برسانیم. بااین‌حال، محتوای حافظه پنهان ممکن است کهنه شود. به همین علت آن‌ها را مستقلاً به‌روزرسانی می‌کنیم. در این راستا از معیار سن اطلاعات برای ارزیابی تازگی محتوا استفاده می‌نماییم.

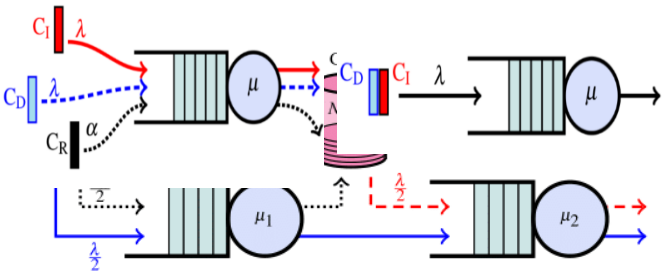
ما دو کانال اصلی در نظر می‌گیریم، فرانت‌هال[۳۰] و فروسو[۳۱]. فرانت‌هال، ایستگاه پایه را به تأمین‌کننده محتوا متصل می‌کند، درحالی‌که فروسو، ایستگاه پایه را به کاربر نهایی متصل می‌کند [۱۲]. ازآنجایی‌که اندازه درخواست در مقایسه با اندازه فایل‌های اصلی بسیار کوچک است، تأثیر آن‌ها را در تأخیر نادیده گرفته‌ایم.

فرض بر این است که اگر فایل‌ها در حافظه پنهان در ایستگاه پایه، موجود باشند، فایل‌های درخواستی از طریق کانال فروسو برای کاربران ارسال شود. در غیر این صورت، ایستگاه پایه درخواست‌ها را برای دریافت محتوای موردنظر از تأمین‌کننده محتوا، به ابر ارسال می‌کند و سپس، پس از دریافت فایل درخواستی، آن‌ها را برای کاربران ارسال می‌نماید. همچنین فرض می‌شود که زمان لازم برای ارسال فایل از هر کانال به‌صورت یک متغیر تصادفی با توزیع نمایی است. این فرض با در نظر گرفتن اندازه‌ی ثابت برای قطعات فایل‌های مختلف و همچنین نرخ بیت ثابت در هر دو کانال (البته با مقادیر مختلف در حالت کلی) توجیه می‌شود. درواقع، ما یک مدل محو شونده بلوکی[۳۲] را برای هر دو کانال فرانت‌هال و فروسو فرض می‌کنیم. بنابراین، با توجه به طرح ارسال مجدد خودکار (ARQ)[۳۳]، تعداد ارسال موردنیاز برای یک انتقال موفق، یک متغیر تصادفی هندسی است، و لذا زمان کل ارسال یعنی مجموع این تعداد بازه زمانی ثابت را با یک متغیر تصادفی پیوسته نمایی تقریب می‌زنیم (می‌دانیم هر دو متغیر هندسی و متغیر تصادفی نمایی بی حافظه هستند و اگر طول زمانی بازه ثابت برای هر ارسال به سمت صفر میل کند این تقریب دقیق می‌شود [۲۱]). ازآنجایی‌که درخواست‌ها به‌صورت تصادفی و تحت یک فرآیند پواسون می‌رسند، ما دو صف مربوط به لینک فرانت‌هال و فروسو خواهیم داشت. بنابراین، نرخ انتقال در هر لینک معادل نرخ سرویس در صف مربوطه است. نرخ سرویس برای لینک فرانت‌هال ، $\mu_1$ و برای فروسو ، $\mu_2$ است. علاوه بر این، $\lambda_u$ به‌عنوان نرخ متوسط برای به‌روزرسانی حافظه پنهان (برای محتوای بسیار پویا) از طریق فرانت‌هال در نظر گرفته می‌شود که معمولاً $\lambda_u < \lambda$ است.

در ادامه سه طرح برای ذخیره‌سازی داده‌ها در حافظه پنهان را در نظر می‌گیریم.

هر فایل را به دو جزء تقسیم می‌کنیم. درواقع از تئوری صف می‌دانیم که اگر زمان سرویس یک فایل نمایی با نرخ $\mu$ باشد، و همچنین اگر فایل‌ها را به دو جزء تقسیم کنیم به‌طوری‌که نرخ سرویس هر مشتری $2\mu$ شود (و بالطبع نرخ

شکل ۲- ذخیره سازی کامل تحویل کامل (WCWD).




ورود مشتریان هم $2\lambda$ می‌شود) در صف جدید که نرخ مشتری و نرخ سرویس هر دو، دو برابر شده است، متوسط تأخیر کاهش می‌یابد [۲۲]. به همین دلیل، برای منصفانه بودن مقایسه نتیجه‌ها، در تمام طرح‌ها فایل‌ها را به دو جزء تقسیم می‌کنیم حتی در طرح‌هایی که کل فایل در حافظه پنهان ذخیره می‌شود فرض می‌کنیم که درخواست فایل متناظر، درخواست دو جزء فایل است که هر جزء با نرخ $2\mu$ سرویس داده می‌شود.

## ۲-۱- ذخیره‌سازی کامل تحویل کامل (WCWD)

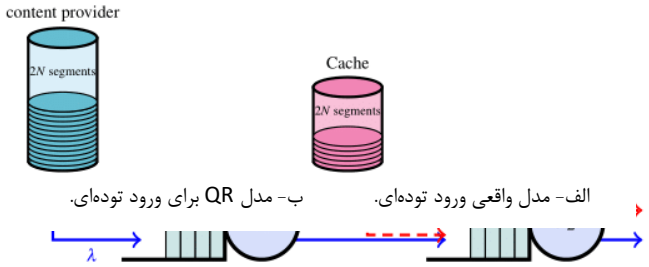
در این طرح، $N$ فایل را به‌طور کامل در حافظه پنهان قرار می‌دهیم و سایر فایل‌ها باید از تأمین‌کننده محتوا واکشی شوند. بنابراین، هر دو جزء از یک فایل کامل در تأمین‌کننده محتوا و یا در حافظه پنهان قرار داده‌شده‌اند. نرخ کل درخواست‌های ورود به هر صف برابر $\frac{\lambda}{2}$ می‌شود که در شکل۲ نشان داده‌شده است. بدیهی است که هر درخواست برای بازیابی یک فایل کامل است، که به معنای دریافت هر دو جزء از فایل است. همچنین به‌طور مشابه می‌توان نتیجه گرفت، نرخ به‌روزرسانی کل برای حافظه پنهان نیز برابر $\frac{\lambda_u}{2}$ است.

## ۲-۲- ذخیره‌سازی جزئی تحویل کامل (PCWC)

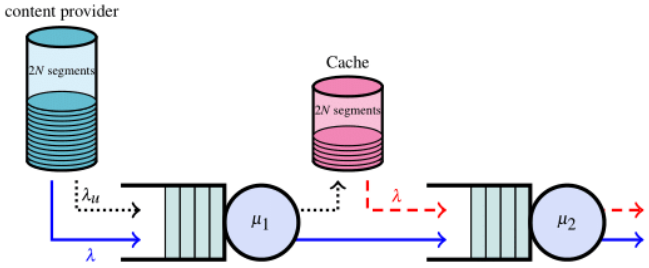
در این طرح، نصف هر فایل را در حافظه پنهان قرار می‌دهیم. هنگامی‌که کاربر، درخواست فایلی را می‌کند، تأمین‌کننده محتوا جزء دوم فایل را به ایستگاه پایه می‌فرستد، سپس این جزء‌به‌جزء اول فایل موجود در حافظه پنهان در ایستگاه پایه ملحق می‌شود و درنهایت کل فایل برای کاربر ارسال می‌شود. علاوه بر این، همان‌طور که در شکل ۳ نشان داده‌شده است، واضح است که نرخ درخواست‌های به‌روزرسانی و ورودی به ترتیب $\lambda_u$ و $\lambda$ می‌شود (زیرا به‌نوعی همه فایل‌ها هم در حافظه پنهان هستند و هم نیاز به واکشی تأمین‌کننده محتوا دارند) . البته ازآنجاکه یک جزء فایل به‌روزرسانی و یا واکشی می‌شود پس هر درخواست جهت دستیابی و یا به‌روزرسانی یک جزء فایل است.

## ۲-۳- ذخیره‌سازی جزئی تحویل جزئی (PCPD)

در این طرح مانند طرح قبلی نصف هر فایل را در حافظه پنهان قرار می‌دهیم. زمانی که کاربر فایلی را درخواست می‌کند، حافظه پنهان موجود در ایستگاه پایه در کنار تأمین‌کننده محتوا، درخواست‌ها را به‌طور مستقل و موازی پردازش می‌کند. بنابراین پس از هر درخواست، جزء ذخیره‌شده در حافظه پنهان، مستقیماً برای کاربر ارسال می‌شود و جزء دیگر هم مستقلاً پس از واکشی از تأمین‌کننده محتوا از طریق ایستگاه پایه برای کاربر ارسال می‌گردد. در یک نگاه این طرح بهتر است چراکه جزء ذخیره‌شده از ایستگاه پایه زودتر ارسال می‌شود و درنتیجه کاربر می‌تواند از آن بهره ببرد تا جزء دوم هم برسد. البته اگر فایل از نوع ویدئو باشد ممکن است وقفه‌ای بین دو جزء ویدئو حاصل گردد. همان‌طور که در شکل ۴ نشان داده‌شده است نرخ درخواست‌های به‌روزرسانی و ورود به ترتیب $\lambda_u$ و $\lambda$ می‌شود.




شکل ۳- ذخیره سازی جزئی تحویل کامل (PCWC).




شکل ۴- ذخیره سازی جزئی تحویل جزئی (PCPD).

باتوجه‌به انگیزه ذخیره‌سازی جزئی توضیح داده‌شده در مقدمه، هر جزء از فایل قابل تحویل به کاربر فرض می‌شود، بنابراین ما تأخیر $\overline{D}$ و سن اطلاعات $\overline{A}$ یک فایل را به‌صورت زیر محاسبه می‌کنیم.

$$\overline{A} = \frac{\overline{A}_{\mathrm{BS}} + \overline{A}_{\mathrm{CP}}}{2}, \qquad \overline{D} = \frac{\overline{D}_{\mathrm{BS}} + \overline{D}_{\mathrm{CP}}}{2}, \tag{۱}$$

که در آن زیرنویس BS برای جزء فایل‌های ذخیره‌شده در حافظه پنهان در ایستگاه پایه و CP برای جزء فایل‌های موجود در تأمین‌کننده محتوا است.

# ۳- مدل تحلیل ارائه‌شده مبتنی بر شبکه صف

با توجه به طرح‌های ذخیره‌سازی فوق و همچنین سناریوی در نظر گرفته‌شده، برای به دست آوردن میانگین تأخیر باید دو صف پشت سر هم [۳۴] ، شامل یک صف برای فرآیند انتقال فایل‌هایی که از تأمین‌کننده محتوا واکشی می‌شوند و یک صف برای فایل‌هایی که از ایستگاه پایه ارسال می‌شوند، در نظر بگیریم. همان‌طور که در شکل‌های ۳ تا ۵ نشان داده‌شده است با توجه به فرض‌های بخش قبل، هر صف به‌صورت یک صف ساده M/M/1 مدل می‌شود. بنابراین به نظر می‌رسد که ما یک شبکه صف جکسون[۳۵] داریم که خاصیت ضربی دارد [۲۲].

در یک شبکه صفی که خاصیت ضربی دارد، رفتار سیستم به‌گونه‌ای است که گویی هر صف به‌صورت مستقل کار می‌کند، بنابراین ما می‌توانیم توزیع حالت دائمی را (پس از حل معادلات ترافیک که بیانگر معادلات تعادل نرخ ورود در صف‌ها است) به دست آوریم و قانون لیتل[۳۶] را برای استخراج میانگین تأخیر اعمال کنیم [۲۲]. بااین‌حال، با بررسی طرح‌های ذخیره‌سازی بخش قبل، می‌توانیم دو رویداد اساسی را مشاهده کنیم که مدل‌سازی را پیچیده‌تر می‌کند، به‌طوری‌که نمی‌توانیم آن را یک

شکل ۵- ورود توده‌ای با سایز دو.

شبکه صف جکسون ساده در نظر بگیریم. درواقع، زمانی که هر دو جزء فایل در تأمین‌کننده محتوا و یا در ایستگاه پایه هستند، مانند طرح WCWD، درخواست فایل به معنای ورود توده‌ای[۳۷] در صف مربوطه است زیرا هر دو جزء یک فایل به‌طور

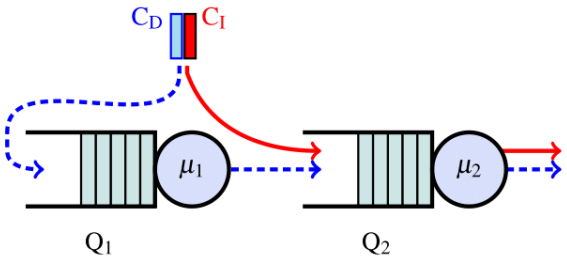
همزمان درخواست و درنتیجه به سمت کاربر ارسال می‌شود. علاوه بر این، در حالت PCPD، درخواست یک فایل به معنای پردازش موازی درخواست‌ها به‌صورت همزمان در لبه و ابر است، یعنی ورود همزمان دو مشتری در دو صف پشت سرهم. با توجه به‌نوعی وابستگی بین ورودی‌ها در هر دو رویداد، باید شبکه صف جکسون را تغییر دهیم تا چنین وابستگی‌هایی را در برگیرد. در ادامه، توضیح می‌دهیم که چگونه یک شبکه صف جکسون را برای مدل‌سازی ورودی‌های همزمان در صف‌های پشت سرهم تغییر می‌دهیم.

در مدل‌سازی با استفاده از شبکه‌های صف، معمولاً از سیگنال‌های مثبت[۳۸] و منفی[۳۹] برای نمایش حرکت‌های همزمان استفاده می‌شود [۲۲]. سیگنال‌های مثبت موجودیت‌هایی هستند که وارد صف می‌شوند و یک مشتری را اضافه می‌کنند و متعاقباً موجودیت دیگری را همزمان در خروجی ایجاد می‌کنند. هنگامی‌که در ورودهای یک صف M/M/1 سیگنال مثبت داریم، برای اینکه صف شبه برگشت‌پذیر[۴۰] (QR) شود (QR خاصیتی است که ورود پواسون به خروج پواسون منجر می‌شود [۲۲])، که یک شرط کافی برای داشتن خاصیت ضربی در شبکه‌های صف است، باید یک نرخ اضافی برای سیگنال مثبت در خروجی صف به هنگامی‌که صف خالی است در نظر بگیریم. سیگنال‌های اضافی در خروجی صف موجب بیشتر شدن شلوغی شبکه نسبت به وضعیت واقعی می‌شوند. اگرچه با اعمال سیگنال مثبت در صف‌های M/M/1، صف‌های QR با ورودی‌های مستقل خواهیم داشت و به‌عبارت‌دیگر، هیچ ورود همزمان در صف‌های نهایی M/M/1 در محاسبات وجود ندارد و فقط نرخ ورود افزایش‌یافته و محاسبات بسیار راحت شده است، ولی نتایج به‌دست‌آمده از شبیه‌سازی‌ها نشان می‌دهد که تأخیر متوسط در این روش به علت وجود نرخ‌های اضافی دقت کمی دارد. در اینجا، ما یک رویکرد جدید را برای اصلاح ورودی‌های صف دنبال می‌کنیم تا خاصیت QR حفظ شود و میانگین تأخیر به‌دست‌آمده بسیار دقیق باشد.

رویکرد ما مبتنی بر در نظر گرفتن ورودی‌های اضافی با بررسی وابستگی بین ورودی‌های همزمان و با توجه به مشاهده وضعیت صف‌ها است. ابتدا مروری بر کار قبلی خود در [۲۰] که به بررسی ورودی‌های وابسته در صف M/M/1 پرداخته است می‌پردازیم و سعی می‌کنیم آن را برای استفاده در مقاله حاضر تطبیق دهیم.

همان‌طور که در [۲۰] بحث شده است برای مدل‌سازی ورود توده‌ای به یک صف M/M/1 که ساده‌ترین نوع ورود وابسته مشتری‌ها است (به‌گونه‌ای که برخی ورودی‌ها مستقل می‌آیند و به دنبال هر ورودی مستقل بلافاصله کاربری به‌صورت وابسته به آن می‌آید)، از روش‌های سنتی مانند زنجیره‌های مارکوف می‌توان بهره جست و با استفاده از تبدیل Z [۲۳] و یا فرآیند شبه زاد و مرگ[۴۱] و روش‌های تحلیل ماتریسی[۴۲] [۲۴] به حل زنجیره مارکوف پرداخت. اما راه‌حل‌های موجود بین دو مشتری در یک توده تمایز قائل نمی‌شوند و به‌راحتی قابل‌تعمیم به یک شبکه صف (مانند صف‌های پشت سر هم) نیستند. ما در [۲۰] روشی را ارائه دادیم که از طریق آن توانستیم، ورود توده‌ای به یک صف (شکل۵-الف) را با استفاده از یک صف QR (شکل۵-ب) با ورودی‌های مستقل و ورودی‌های اضافی، به‌گونه‌ای مدل کنیم که متوسط زمان حضور هر مشتری مستقل در صف QR برابر با متوسط زمان حضور هر مشتری در صف اصلی باشد.

اساس محاسبه نرخ ورودی‌های اضافی $\alpha$ در مدل QR بر این مبنا است که با توجه به ویژگی PASTA (ورود پواسون میانگین‌های زمانی را مشاهده می‌کند[۴۳] [۲۱])، یک مشتری مستقل $C_I$ همیشه در هنگام ورود صف را در حالت دائمی [۴۴] می‌بینند. بااین‌حال، یک مشتری $C_D$ (مشتری وابسته) فقط می‌تواند یک مشتری بیشتر از حالت‌هایی از صف را ببیند که مشتری مستقل همراه آن (مشتری جلویی در توده) دیده است. این محدودیت نشان می‌دهد که مشتریان $C_D$ نمی‌توانند به یک صف خالی برسند و ما برای مدل‌سازی این وضعیت در [۲۰]، $\alpha$ را به‌صورت زیر به دست آوردیم:




شکل ۶ - ورودی همزمان به دو صف پشت سر هم.

شکل ۶ - ورودی همزمان به دو صف پشت سر هم.

$$\alpha = \lambda(1-\rho), \qquad (۲)$$

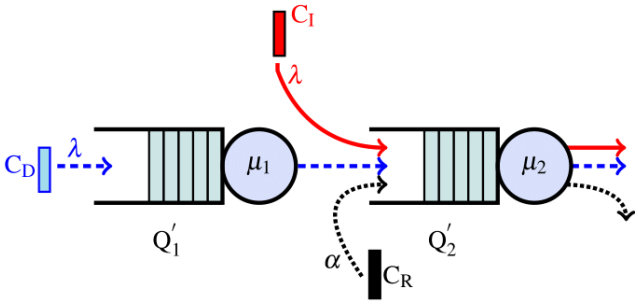
که $\rho$، ضریب بهره‌وری[۴۵] و $\lambda$ نرخ ورود به‌صف در مدل QR است. پس ورود توده‌ای به یک صف را با استفاده از یک نرخ اضافه توانستیم به‌صورت یک صف QR مدل کنیم.

حال به سراغ ورودی همزمان به دو صف پشت سرهم نشان داده‌شده در شکل ۶ می‌رویم و سعی می‌کنیم وابستگی بین ورودی‌های همزمان را با استفاده از یکسری نرخ‌های اضافی مدل کنیم. همان‌طور که در شکل ۶ نشان داده‌شده است هر مشتری $C_D$ و $C_I$ به‌طور همزمان به صف‌های مربوطه خود وارد می‌شوند.

لازم به ذکر است که اگرچه توزیع حالت دائمی صف‌های QR پشت سر هم (مانند دو صف M/M/1)، به علت وجود خاصیت ضربی به‌سادگی به دست می‌آید [۲۲]، اما به دلیل وجود ورودهای همزمان، سناریوی ما، با دو صف QR پشت سر هم متفاوت است. به‌عنوان‌مثال، در دو صف QR پشت سر هم، هر مشتری که صف اول را ترک می‌کند، صف بعدی را با احتمال $\pi_2(0)$ خالی مشاهده می‌کند، این امر به این دلیل است که در یک شبکه صف جکسون مشتریان ورودی از خارج شبکه و یا از صف‌های دیگر، طبق خاصیت ASTA (ورود پواسون میانگین‌های زمانی را مشاهده می‌کند[۴۶]) قبل از ورود به یک صف، آن را در حالت دائمی می‌بینند [۲۲]. بااین‌حال، زمانی که یک مشتری $C_D$ و یک مشتری $C_I$ به‌صورت همزمان به صف‌های مربوط به خود وارد شوند، احتمال کمتری وجود دارد که مشتری $C_D$ پس از ترک صف $Q_1$ ، صف روبه روی خود ($Q_2$) را خالی ببینند. این امر به خاطر این است که ما با اطمینان می‌دانیم که یک مشتری $C_I$ همزمان با ورود مشتری $C_D$ ، وارد صف $Q_2$شده است، بنابراین ممکن است مشتری $C_I$ هنوز در صف $Q_2$ وجود داشته باشد و مشتری $C_D$ اثر آن را حس کند.

شایان‌ذکر است که فرآیند خروج از صف $Q_1$ که M/M/1 است، یک فرآیند پواسون است [۲۲]. به‌عبارت‌دیگر، فرآیند نقطه‌ای متناظر با لحظات رسیدن مشتریان $C_D$ به‌صف $Q_2$ یک فرآیند پواسون است، اما مستقل از فرآیند نقطه‌ای لحظات رسیدن مشتریان $C_I$ به‌صف $Q_2$ نیست. همان‌طور توضیح دادیم اگر یک مشتری $C_D$ همیشه بلافاصله بعد از یک مشتری $C_I$ برسد (ورود توده‌ای)، مشتری $C_D$ صف را در حالت دائمی به‌اضافه یک می‌بینند. بااین‌حال، در این مورد، هنگامی‌که یک مشتری $C_D$ از صف $Q_1$ خارج می‌شود، اگر مشتری $C_I$ متناظر آن هنوز در صف $Q_2$ باشد، مشتری $C_D$، صف $Q_2$ را در حالت دائمی به‌اضافه یک می‌بینند. این باعث می‌شود که صف $Q_2$ نسبت به حالتی که ورودها، فرآیندهای پواسون مستقل هستند شلوغ‌تر به نظر برسد.

حال، اجازه دهید احتمال خروج مشتری $C_D$ از صف $Q_1$ درحالی‌که مشتری $C_I$ همچنان در صف $Q_2$ وجود دارد را $\beta$ در نظر بگیریم. به‌عبارت‌دیگر، مشتری $C_D$، صف $Q_2$ را در حالت دائمی به‌علاوه یک، با احتمال $\beta$ مشاهده می‌کند. درنتیجه، میانگین زمان پاسخ برای مشتریان $C_I$ و $C_D$ در صف $Q_2$ به $T_2$ و $T_2 + \beta X_2$ تبدیل می‌شود، که در آن $X_2 = \frac{1}{\mu_2}$ میانگین زمان سرویس در صف $Q_2$ است و $T_2$




متوسط زمان ماندن مشتریانی است که هنگام ورود، صف $Q_2$ را در حالت دائمی می‌بینند.

با عنایت به بحث‌های فوق و با الهام [۲۰] از شبکه صف QR پیشنهادی شکل۷ استفاده می‌کنیم. درواقع چون با احتمال $\beta$، در صف $Q_2$ ورود توده‌ای داریم، ورود توده‌ای را با یک نرخ اضافه مدل می‌کنیم. در این مدل، توزیع زمان ماندن در صف اول که مانند یک صف M/M/1 ساده است، به فرم زیر است[۲۲] :

$$T(t) = \mu(1-\rho)e^{-\mu(1-\rho)t}. \qquad (۳)$$

پس برای محاسبه $\beta$ داریم

$$\beta \approx P(Y < Z), \qquad (۴)$$

که در آن Y زمان حضور مشتریان $C_D$ در صف $Q_1'$ و Z زمان حضور مشتریان $C_I$ در صف $Q_2'$ است و هر دو از توزیع نمایی مانند (۳) با پارامترهای متناظر پیروی می‌کند. شایان‌ذکر است که محاسبه دقیق $\beta$ کار پیچیده‌ای است به‌عبارت‌دیگر ممکن است در لحظه ورود مشتری $C_D$ به‌صف $Q_2$، مشتری $C_I$ وابسته به آن، از صف $Q_2$ خارج‌شده باشد ولی مشتری $C_D$ همچنان اثر آن را ببیند که بایستی چنین احتمالی را در $\beta$ محاسبه نمود. رابطه (۴) یکی از ترم‌های مؤثر در دیدن اثر مشتری $C_I$ توسط مشتری $C_D$ در لحظه ورود به‌صف $Q_2$ را مدل می‌کند و شبیه‌سازی‌ها نشان از دقت تقریب مزبور دارد. ازاین‌رو داریم

$$\beta \approx \frac{\mu_1(1-\rho_1)}{\mu_1(1-\rho_1)+\mu_2(1-\rho_2)}. \qquad (۵)$$

که $\rho_1$ و $\mu_1$ به ترتیب ضریب بهره‌وری و نرخ سرویس در صف $Q_1'$ و همچنین $\rho_2$ و $\mu_2$ به ترتیب ضریب بهره‌وری و نرخ سرویس در صف $Q_2'$ می‌باشند. حال با حل نمودن معادلات ترافیک، تمام پارامترهای موردنیاز برای به دست آوردن $T_2$ و $T_2 + \beta X_2$ که بیانگر میانگین زمان پاسخ برای مشتریان $C_I$ و $C_D$ در صف $Q_2$ است استخراج می‌شود.

همان‌طور که در قسمت قبلی این بخش توضیح داده شد، هنگامی‌که یک ورود توده‌ای در صف اول داریم، اگرچه ورودی‌های اضافی باید در خروجی صف حذف شوند، اما اثر ورود توده (یعنی وابستگی بین مشتری اول و دوم در توده) به‌صف دوم انتشار می‌یابد. به‌عبارت‌دیگر، وقتی مشتری $C_D$ به‌صف دوم می‌رسد، با احتمال کمتری در مقایسه باحالتی که ورودی‌های صف اول مستقل هستند صف دوم را خالی می‌بینند، زیرا ما می‌دانیم وقتی مشتری $C_D$ صف اول را ترک می‌کند و وارد صف دوم می‌شود یک مشتری $C_I$ قطعاً به‌اندازه یک‌زمان سرویس زودتر وارد صف دوم شده است پس $C_D$ با یک احتمال صف را برتر می‌بینند. بنابراین مشابه با بحث قبل، با احتمال $\frac{\mu_1}{\mu_1+\mu_2(1-\rho_2)}$، مشتری $C_D$ دوباره با مشتری $C_I$ در صف دوم ملاقات می‌کند، که معادل با رسیدن توده در صف $Q_2$ می‌شود. لازم به ذکر است که این احتمال بیانگر این است که مشتری $C_I$ پس از وارد شدن به‌صف دوم، تا زمانی که مشتری $C_D$ از صف اول خارج نشده خارج نشود. بنابراین برای لحاظ نمودن اثر وابستگی ورود توده‌ای در خروجی صف اول، ورودی‌های اضافی را با نرخ $\lambda(1-\rho_2)\frac{\mu_1}{\mu_1+\mu_2(1-\rho_2)}$ در صف دوم نظر می‌گیریم. در بخش بعد از مدل‌سازی سناریوهای وابستگی در این بخش بهره می‌بریم.

## ۴- تجزیه‌وتحلیل تأخیر و سن اطلاعات

در این بخش، میانگین تأخیر و سن اطلاعات را برای سه طرح ذخیره‌سازی در بخش دوم استخراج می‌کنیم. برای این منظور، ما از مدل‌های تحلیلی پیشنهادی که در بخش قبل توضیح داده شد، بهره می‌بریم. با توجه به فرم ضربی شبکه‌ی صف‌های ارائه‌شده، می‌توانیم توزیع حالت دائمی تعداد مشتریان (اجزاء فایل) را در هر صف استخراج‌کنیم و با اعمال قانون لیتل، میانگین تأخیر را محاسبه کنیم. علاوه بر این، با توجه به قضیه عمر باقیمانده[۴۷] مربوط به فرآیند تجدید[۴۸] به‌روزرسانی حافظه پنهان، می‌توانیم میانگین سن اطلاعات را مشابه [۱۲] استخراج‌کنیم. لازم به ذکر است سن اطلاعات در بحث ذخیره‌سازی، در لحظه‌ای که محتوا توسط کاربر دریافت می‌شود در نظر گرفته می‌شود.

### ۴-۱- ذخیره‌سازی کامل تحویل کامل

در این طرح، ورودی‌های توده‌ای در هر دو صف برای فایل‌های به‌روزرسانی و فایل‌های درخواستی داریم. متعاقباً، نرخ ورود در صف اول به $2\times\frac{\lambda}{2}$ و $2\times\frac{\lambda_u}{2}$ تبدیل می‌شود زیرا هر فایل از دو جزء تشکیل‌شده است و ما در همه سناریوها، هر جزء فایل را به‌عنوان ورودی در نظر می‌گیریم. با توجه به بخش قبل یک مدل شبکه صف پشت سر هم QR، متشکل از دو صف M/M/1 با ورود مشتریان اضافی و مستقل پیشنهاد می‌دهیم. علاوه بر این، برای گنجاندن اثر وابستگی ناشی از ورودهای توده‌ای در صف اول، باید تعدادی مشتری اضافی را در صف دوم نیز اضافه کنیم، یعنی دو نوع ورود اضافی در صف دوم داریم، یکی برای مدل‌سازی وابستگی بین ورودی‌های خودش (یعنی ورود توده‌ای به دلیل ارسال هر دو جزء فایل ذخیره‌شده در حافظه پنهان در ایستگاه پایه) و نوع دیگر به دلیل انتشار اثر وابستگی ورودی‌ها در اولین صف (یعنی ورود توده‌ای به دلیل واکشی هر دو جزء فایل از تأمین‌کننده محتوا). بنابراین پس با استفاده از معادلات ترافیک برای اولین صف‌داریم:

$$\boldsymbol{\rho_1 = \frac{\lambda+\lambda_u+\frac{\lambda}{2}(1-\rho_1)+\frac{\lambda_u}{2}(1-\rho_1)}{\mu_1}} \qquad (۶)$$

$$\boldsymbol{\Longrightarrow \rho_1 = \frac{3(\lambda+\lambda_u)}{2\mu_1+\lambda+\lambda_u}},$$

و برای صف دوم داریم

$$\boldsymbol{\rho_2 = \frac{2\lambda+\frac{\lambda}{2}(1-\rho_2)+\frac{\lambda}{2}(1-\rho_2)\frac{\mu_1}{\mu_1+\mu_2(1-\rho_2)}}{\mu_2}}$$

$$\boldsymbol{\Longrightarrow \rho_2 = \frac{5\lambda+\lambda\frac{\mu_1}{\mu_1+\mu_2(1-\rho_2)}}{2\mu_2+\lambda\left(1+\frac{\mu_1}{\mu_1+\mu_2(1-\rho_2)}\right)}}, \qquad (۷)$$

متعاقباً از [۲۳] متوسط زمان ماندن در صف‌های M/M/1 فوق به‌صورت

$$\boldsymbol{D_1 = \frac{\rho_1}{1-\rho_1}\times\frac{1}{\lambda+\lambda_u}}, \quad \boldsymbol{D_2 = \frac{\rho_2}{1-\rho_2}\times\frac{1}{2\lambda}}, \qquad (۸)$$

به دست می‌آید که $D_1$ بیانگر متوسط زمان ماندن مشتری‌ها در صف اول و $D_2$ بیانگر متوسط زمان ماندن مشتری‌ها در صف دوم است. بدیهی است که ما ورودهای اضافی را در قانون لیتل در نظر نمی‌گیریم زیرا ورودهای واقعی شامل ورودی‌های اضافی نیستند . ازاین‌رو برای محاسبه متوسط زمان بین درخواست و دریافت فایل توسط کاربر نهایی برای جزء فایل‌هایی که از تأمین‌کننده محتوا فرستاده می‌شوند، داریم

$$\overline{D}_{\mathrm{CP}} = D_1 + D_2 \,, \tag{۹}$$

و متعاقباً برای جزء فایل‌های ارسالی از ایستگاه پایه‌داریم

$$\overline{D}_{\mathrm{BS}} = D_2 \,, \tag{۱۰}$$

درنتیجه باتوجه‌به (۱)، (۸)، (۹) و (۱۰) داریم

$$\overline{D} = \frac{D_1}{2} + D_2 = \frac{\rho_1}{2(1-\rho_1)} \times \frac{1}{\lambda + \lambda_u} + \frac{\rho_2}{(1-\rho_2)} \times \frac{1}{2\lambda} \,. \tag{۱۱}$$

اگرچه مشتریان $\mathrm{C_I}$ و $\mathrm{C_D}$ در یک توده، زمان‌های انتظار متفاوتی در صف‌دارند، ولی باتوجه‌به (۱) فقط نیاز داریم میانگین تأخیرهایی را داشته باشیم که همه مشتری‌ها تجربه می‌کنند و نیازی نیست که هرکدام را جداگانه مشخص کنیم. بااین‌حال، میانگین تأخیرهایی که مشتریان $\mathrm{C_I}$ و $\mathrm{C_D}$ به‌طور جداگانه تجربه می‌کنند، قابل‌محاسبه هستند.

در مورد سن اطلاعات، می‌توانیم بین میانگین کل تأخیر مشتریان مختلف متناظر با اجزاء فایل‌ها تمایز قائل شویم، چراکه میزان تازگی اجزاء فایل‌ها می‌تواند متفاوت باشد. اما به خاطر سادگی محاسبات، ما تأخیر کل یکسانی را برای مشتریان در همه طرح‌های پیشنهادی فرض می‌کنیم. برای ارزیابی عملکرد میانگین سن اطلاعات مانند [۱۲]، ما مدت‌زمان بین دو به‌روزرسانی پیاپی فایل‌ها در حافظه پنهان را به‌صورت زیر در نظر می‌گیریم:

$$T_{\mathrm{update}} = \sum_{s=1}^{N} t_s \,, \tag{۱۲}$$

که در آن $\mathrm{t_s}$ زمان بین به‌روزرسانی برای فایل s-ام است، و $(\mathrm{t_1}, \mathrm{t_2}, \dots \mathrm{t_N})$ مستقل و هم توزیع با توزیع ارلانگ-۲[۴۹] و نرخ $\frac{\lambda_u}{2}$ هستند. نصف شدن $\lambda_u$ به این دلیل است که در این طرح، ما فایل‌های به‌روزرسانی را به‌صورت توده‌هایی با اندازه دو می‌فرستیم، و همچنین یک فایل وقتی به‌روز می‌شود که هر دو جزء آن به‌روز شده باشند. پس برای چرخه به‌روزرسانی ($\mathrm{T_{update}}$) مربوط به هر فایل، داریم:

$$E\left[T_{\mathrm{update}}\right] = \frac{2N}{\lambda_u} \,, \tag{۱۳}$$

$$E\left[T_{\mathrm{update}}^2\right] - E^2\left[T_{\mathrm{update}}\right] = \frac{4N}{\lambda_u^2} \,.$$

در ارائه دهنده محتوا، درخواست‌ها به‌صف مربوطه می‌رسند و منتظر می‌مانند تا فایل‌هایی تازه از طریق فرانت‌هال ارسال شوند. علاوه بر این، سن اطلاعات داده‌های موجود در ایستگاه پایه از سه قسمت تشکیل‌شده است: (۱) زمان ارسال برای تحویل جدیدترین محتوا به ایستگاه پایه برای ذخیره‌سازی در حافظه پنهان از طریق فرانت‌هال، (۲) زمانی که از آخرین به‌روزرسانی فایل در حافظه پنهان می‌گذرد تا زمان ارسال به کاربر از طریق سرور صف دوم (لینک فروسو)، که با $T_t$ نشان داده می‌شود و (۳) زمان انتقال برای تحویل محتوا به کاربر از طریق کانال فروسو. با توجه به نظریه تجدید و قضیه عمر باقیمانده [۲۱] ، هنگامی‌که یک درخواست بین دو چرخه به‌روزرسانی وارد سرویس می‌شود، سن یا زمان صرف شده را می‌توان به‌صورت زیر محاسبه کرد (درخواست‌ها به‌صورت پواسون می‌آیند و متناظرا لحظات شروع ارسال به کاربر نیز پواسون است [۱۲]):

$$\mathbb{E}[T_t] = \frac{\mathbb{E}\left[T_{\mathrm{update}}^2\right]}{2\mathbb{E}\left[T_{\mathrm{update}}\right]} = \frac{1+N}{\lambda_u} \,. \tag{۱۴}$$

ازاین‌رو داریم

$$\overline{\boldsymbol{A}}_{\mathrm{BS}} = \boldsymbol{X_1} + \mathbb{E}[\boldsymbol{T_t}] + \boldsymbol{X_2} = \frac{\boldsymbol{2}}{\boldsymbol{\mu_1}} + \frac{\boldsymbol{1+N}}{\boldsymbol{\lambda_u}} + \frac{\boldsymbol{2}}{\boldsymbol{\mu_2}} \,, \tag{۱۵}$$

که در آن $\boldsymbol{X_1}$ زمان سرویس برای به‌روزرسانی یک فایل کامل است (یک فایل کامل شامل دو جزء)، و $\boldsymbol{X_2}$ زمان ارسال فایل کامل درخواستی از ایستگاه پایه است. علاوه بر این،

$$\overline{\boldsymbol{A}}_{\mathrm{CP}} = \boldsymbol{X_1} + \boldsymbol{D_2} \,. \tag{۱۶}$$

بنابراین، بر اساس (۱)، (۸)، (۱۵) و (۱۶) می‌توانیم میانگین سن اطلاعات را به‌راحتی به دست آوریم. زمان سرویس اول و دوم نشان‌دهنده مدت‌زمان ارسال یک جزء فایل است. شایان‌ذکر است، زمان انتظار برای درخواست‌های صف اول و دوم متناظرا بر سن اطلاعات فایل‌های واکشی شده و ذخیره‌شده تأثیر نمی‌گذارد به این دلیل که آخرین نسخه به‌روزرسانی فایل مرتبط با درخواست در زمان ارسال، برای کاربران فرستاده می‌شود.

## ۴-۲- ذخیره‌سازی جزئی تحویل کامل

در اینجا، هیچ‌گونه ورود توده‌ای در صف اول نداریم. ولی در صف دوم ورود توده‌ای داریم، زیرا باید هر دو جزء فایل به کاربر نهایی تحویل داده شود (درواقع جزء دوم فایل به‌جزء اول فایل در حافظه پنهان ملحق می‌شود و باهم به سمت کاربر ارسال می‌گردد). به‌این‌ترتیب،

$$\boldsymbol{D_1} = \frac{\boldsymbol{1}}{\boldsymbol{\mu_1 - \lambda - \lambda_u}} \,, \quad \boldsymbol{D_2} = \frac{\boldsymbol{\rho_2}}{\boldsymbol{1-\rho_2}} \times \frac{\boldsymbol{1}}{\boldsymbol{2\lambda}} \,, \tag{۱۷}$$

$$\boldsymbol{\rho_2} = \frac{\boldsymbol{2\lambda + \lambda(1-\rho_2)}}{\boldsymbol{\mu_2}} \Rightarrow \boldsymbol{\rho_2} = \frac{\boldsymbol{3\lambda}}{\boldsymbol{\mu_2 + \lambda}} \,,$$

پس برای محاسبه متوسط زمان بین درخواست و دریافت فایل توسط کاربر نهایی برای جزء فایل‌هایی که از تأمین‌کننده محتوا فرستاده می‌شوند داریم

$$\overline{D}_{\mathrm{CP}} = D_1 + D_2 \,, \tag{۱۸}$$

و ازآنجایی‌که جزء فایل‌هایی که در ایستگاه پایه هستند باید منتظر بایستند تا جزء دیگرشان از تأمین‌کننده محتوا واکشی شود و سپس باهم ارسال شوند، داریم

$$\overline{D}_{\mathrm{BS}} = D_1 + D_2. \tag{۱۹}$$

درنتیجه باتوجه‌به (۱)، (۱۷)، (۱۸) و (۱۹) داریم

$$\overline{D} = D_1 + D_2 = \frac{1}{\mu_1 - \lambda - \lambda_u} + \frac{\rho_2}{1-\rho_2} \times \frac{1}{2\lambda}. \quad (۲۰)$$

برای عملکرد سن اطلاعات، در ایستگاه پایه، جزء اول فایل‌ها به‌روز می‌شود، درنتیجه چرخه به‌روزرسانی برابر است با:

$$T_{\text{update}} = \sum_{s=1}^{2N} t_s , \quad (۲۱)$$

که $t_s$ زمان به‌روزرسانی برای جزء فایل s-ام است، و $t_1 t_2, \dots t_{2N}$ با توزیع نمایی $i.i.d$ با نرخ $\lambda_u$ هستند. بدین ترتیب،

$$\overline{A}_{\text{BS}} = X_1 + \mathbb{E}[T_t] + X_2 = \frac{1}{\mu_1} + \frac{1+2N}{2\lambda_u} + \frac{1}{\mu_2}, \quad (۲۲)$$

$$\overline{A}_{\text{CP}} = X_1 + D_2 , \quad (۲۳)$$

که در آن $D_2$ میانگین کل تأخیری است که مشتری‌ها در صف دوم تجربه می‌کند و $X_1$ , $X_2$ به ترتیب میانگین زمان سرویس جزء فایل در صف اول و دوم است. ازاین‌رو با توجه به (۱)، (۱۷)، (۲۲) و (۲۳) سن اطلاعات به دست می‌آید.

## ۴-۳- ذخیره‌سازی جزئی تحویل جزئی

در این طرح، ما ورودی‌های هم‌زمان به هر دو صف‌داریم (به دلیل آنکه درخواست اجزاء فایل هم‌زمان رخ می‌دهد، ولی اجزاء فایل مستقل ارسال می‌شوند)، درواقع جزء فایل ذخیره‌شده می‌تواند زودتر به کاربر تحویل داده شود، حال‌آنکه در PCWD، هر دو جزء فایل باهم به سمت کاربر ارسال می‌شدند و درنتیجه جزء فایل ذخیره‌شده مستقل به سمت کاربر ارسال نمی‌شد. بنابراین در مقایسه با طرح قبلی، به برخی از ورودهای اضافی در صف دوم نیاز داریم (به بخش ۳ مراجعه کنید). بنابراین داریم

$$D_1 = \frac{1}{\mu_1 - \lambda - \lambda_u} , \quad D_2 = \frac{\rho_2}{1-\rho_2} \times \frac{1}{2\lambda} \quad (۲۴)$$

$$\rho_2 = \frac{2\lambda + \lambda(1-\rho_2)\beta}{\mu_2} \Rightarrow \rho_2 = \frac{\lambda(2+\beta)}{\mu_2 + \lambda\beta}$$

ازاین‌رو برای محاسبه متوسط زمان بین درخواست و دریافت فایل توسط کاربر نهایی برای جزء فایل‌هایی که از تأمین‌کننده محتوا فرستاده می‌شوند داریم

$$\overline{D}_{\text{CP}} = D_1 + D_2 , \quad (۲۵)$$

و ازآنجایی‌که جزء فایل‌هایی که در ایستگاه پایه هستند به‌طور مستقل از ایستگاه پایه ارسال می‌شوند، داریم

$$\overline{D}_{\text{BS}} = D_2 , \quad (۲۶)$$

درنتیجه باتوجه‌به (۱)، (۲۴)، (۲۵) و (۲۶) داریم

$$\overline{D} = \frac{D_1}{2} + D_2 = \frac{1}{2(\mu_1 - \lambda - \lambda_u)} + \frac{\rho_2}{1-\rho_2} \times \frac{1}{2\lambda}.$$

سن اطلاعات در ایستگاه پایه مشابه با طرح قبلی است به‌جز زمان ماندن در صف دوم، زیرا ما اجزاء فایل را به‌طور مستقل ارسال می‌کنیم. به‌این‌ترتیب،

$$\overline{A}_{\text{BS}} = X_1 + E[T_t] + X_2 = \frac{1}{\mu_1} + \frac{1+2N}{2\lambda_u} + \frac{1}{\mu_2},$$

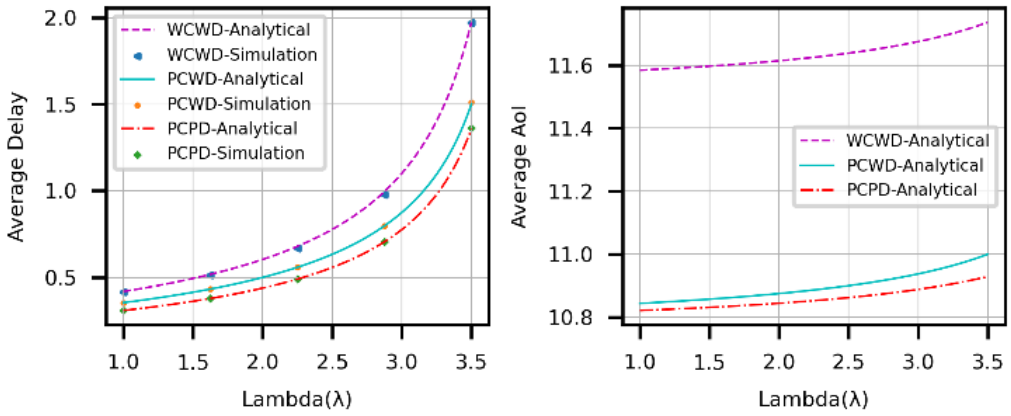
$$\overline{A}_{\text{CP}} = X_1 + D_2$$

که در آن $D_2$ میانگین کل زمانی است که مشتری در صف دوم تجربه می‌کند، و , $X_1$ $X_2$ به ترتیب میانگین زمان سرویس جزء فایل در صف اول و دوم است.

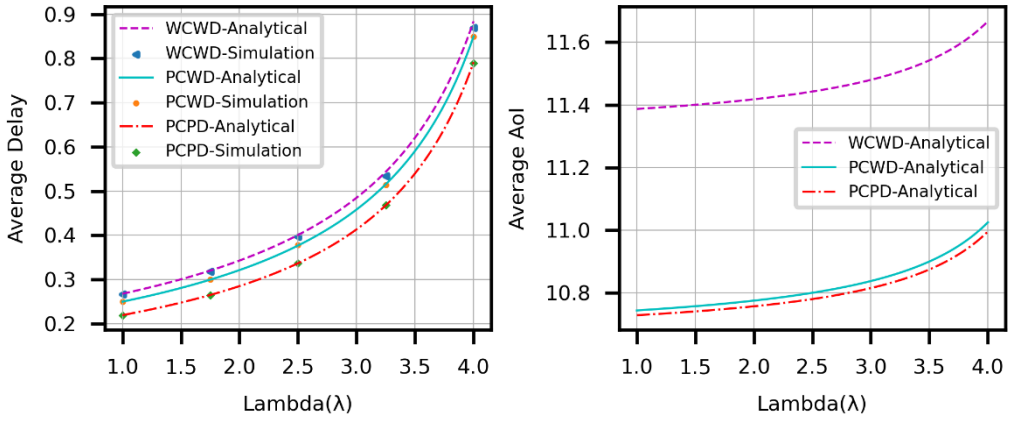
## ۵- شبیه‌سازی و نتایج عددی

در این بخش، ما سناریوهای خود را با پیاده‌سازی یک شبیه‌سازی رویداد گسسته [۵۰] موردبررسی قرار می‌دهیم و در این راستا از کتابخانه SimPy در زبان برنامه‌نویسی پایتون بهره می‌بریم. همان‌طور که در شکل ۸ مشخص است از مقایسه نتایج به‌دست‌آمده از شبیه‌سازی‌ها و مدل تحلیلی شبکه صف ارائه‌شده، اختلاف‌هایی کمتر از ۲ درصد به ازای نرخ‌های ورود و سرویس مختلف دیده می‌شود که نشان‌دهنده دقت بسیار بالای مدل تحلیلی صف ارائه‌شده است. لازم به ذکر است که نرخ‌های ورود و سرویس استفاده‌شده در نتایج عددی به‌گونه‌ای انتخاب‌شده‌اند تا پایداری صف‌ها تضمین شود. به‌طور مثال، با توجه مقادیر نرخ‌های سرویس و ورود میزان بهره‌وری هر دو صف در محدوده $0.2 < \rho < 0.92$ قرار دارد.

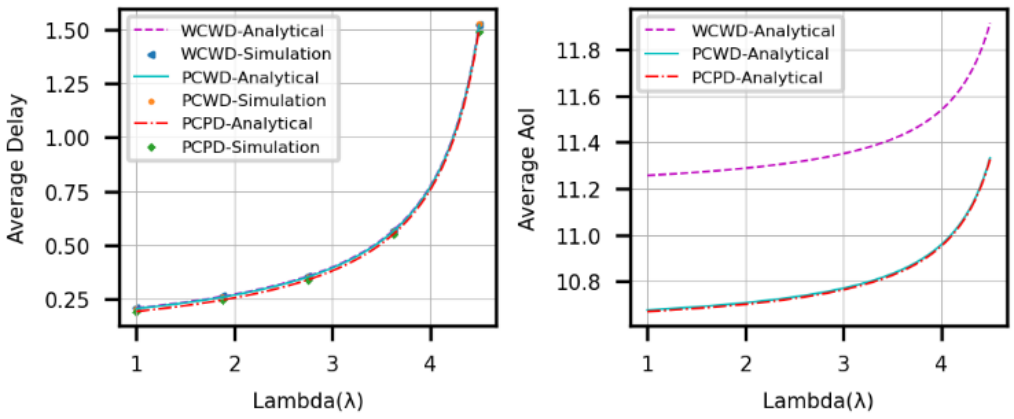
علاوه بر این همان‌طور که در شکل ۸ نشان داده‌شده است. با درنظرگرفتن تأخیر و سن اطلاعات طرح PCPD عملکرد بهتری را در مقایسه با طرح‌های دیگر نشان می‌دهد. همچنین تأثیر $N$ , $\lambda_u$ , $\mu_2$ , $\mu_1$ بر عملکرد سیستم مشهود است. این نتایج نشان می‌دهند که طراحی بهینه سیستم باید با در نظر گرفتن این پارامترها صورت گیرد تا بهره‌وری بالاتری داشته باشد.




الف- $\mu_1 = 5,\ \mu_2 = 10,\ \lambda_u = 0.5,\ N = 10$




$\mu_1 = 10,\ \mu_2 = 10,\ \lambda_u =$




$\mu_1 = 30,\ \mu_2 = 10,\ \lambda_u = 0.$

شکل۸- نتایج شبیه‌سازی و مدل تحلیلی برای سن اطلاعات و تاخیر.

## ۶- نتیجه‌گیری و کارهای آینده

در مقاله حاضر، یک سناریوی ذخیره‌سازی برای فایل‌های بسیار پویا در نظر گرفتیم. در سناریوی خود، یک تأمین‌کننده محتوا در ابر و یک حافظه پنهان در ایستگاه پایه وجود دارد. سه طرح ذخیره‌سازی بررسی شد، WCWD، PCWD و PCPD که در دو طرح آخر فایل‌ها به دو جزء تقسیم شدند. در ذخیره‌سازی جزئی، هر جزء از فایل در حافظه پنهان در ایستگاه پایه ذخیره شد و جزء دیگر از تأمین‌کننده محتوا واکشی گردید. برای تجزیه‌وتحلیل تأخیر و سن اطلاعات، یک مدل صف تحلیلی جدید ارائه کردیم که توسط برخی از ورودی‌های اضافی شلوغ شده است تا ورود هم‌زمان را در طرح‌های ذخیره‌سازی مدل کند. نتایج عددی ما دقت بالای مدل صف پیشنهادی و همچنین برتری PCPD را نسبت به دیگر طرح‌های ذخیره‌سازی در شرایط مختلف نشان داد.

## ۷- تشکر و قدردانی

این تحقیق تحت شماره قرارداد RD-510206-1007 توسط مرکز تحقیق و توسعه شرکت ارتباطات سیار ایران (همراه اول) و جهت پیشرفت حوزه فناوری اطلاعات و ارتباطات موردحمایت قرارگرفته است.

## ۸- مآخذ

[1] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. on inf. theory,* vol. 60, no. 5, pp. 2856-2867, 2014.

[2] M. Zhang, H. Luo, and H. Zhang, "A survey of caching mechanisms in information-centric networking," *IEEE Commun. Surveys & Tuts.,* vol. 17, no. 3, pp. 1473-1499, 2015.

[3] M. Javedankherad, Z. Zeinalpour-Yazdi, and F. Ashtiani, "Content placement in cache networks using graph coloring," *IEEE Sys. J.,* vol. 14, no. 3, pp. 3129-3138, 2020.

[4] M. Javedankherad, Z. Zeinalpour-Yazdi, and F. Ashtiani, "Mobility-Aware Content Caching Using Graph-Coloring," *IEEE Trans. on Vehicular Technol.,* vol. 71, no. 5, pp. 5666-5670, 2022.

[5] S. Li, J. Xu, M. Van Der Schaar, and W. Li, "Popularity-driven content caching," in *Proc. IEEE INFOCOM*, 2016, pp. 1-9.

[6] G. S. Paschos, G. Iosifidis, M. Tao, D. Towsley, and G. Caire, "The role of caching in future communication systems and networks," *IEEE J. on Sel. Areas in Commun.,* vol. 36, no. 6, pp. 1111-1125, 2018.

[7] B. Abolhassani, J. Tadrous, and A. Eryilmaz, "Optimal load-splitting and distributed-caching for dynamic content over the wireless edge," *IEEE/ACM Trans. on Netw.,* vol. 31, no. 5, pp. 2178-2190, 2023.

[8] B. Abolhassani, J. Tadrous, A. Eryilmaz, and E. Yeh, "Fresh caching for dynamic content," in *Proc. IEEE INFOCOM*, 2021, pp. 1-10.

[9] G. Ahani, D. Yuan, and S. Sun, "Optimal scheduling of age-centric caching: Tractability and computation," *IEEE Trans. on Mobile Comput.,* vol. 21, no. 8, pp. 2939-2954, 2020.

[10] S. M. Azimi, O. Simeone, A. Sengupta, and R. Tandon, "Online edge caching and wireless delivery in fog-aided networks with dynamic content popularity," *IEEE J. on Sel. Areas in Commun.,* vol. 36, no. 6, pp. 1189-1202, 2018.

[11] K. Kazari, F. Ashtiani, and M. Mirmohseni, "Cache update and delivery of dynamic contents: A stochastic game approach," *IEEE Trans. on Mobile Comput.,* vol. 23, no. 4, pp. 3035-3047, 2024.

[12] S. Zhang, J. Li, H. Luo, J. Gao, L. Zhao, and X. S. Shen, "Low-latency and fresh content provision in information-centric vehicular networks," *IEEE Trans. on Mobile Comput.,* vol. 21, no. 5, pp. 1723-1738, 2020.

[13] S. Zhang, L. Wang, H. Luo, X. Ma, and S. Zhou, "AoI-delay tradeoff in mobile edge caching with freshness-aware content refreshing," *IEEE Trans. on Wireless Commun.,* vol. 20, no. 8, pp. 5329-5342, 2021.

[14] S. Kaul, R. Yates, and M. Gruteser, "Real-time status: How often should one update?," in *Proc. IEEE INFOCOM*, pp. 2731-2735.

[15] P. Parag, A. Bura, and J.-F. Chamberland, "Latency analysis for distributed storage," in *Proc. IEEE INFOCOM,* 2017, pp. 1-9.

[16] Q. Shuai, V. O. Li, and Z. Lu, "Which achieves lower latency with redundant requests, replication or coding?," in *Proc. IEEE GLOBECOM*, 2017, pp. 1-6.

[17] N. Abani, G. Farhadi, A. Ito, and M. Gerla, "Popularity-based partial caching for information centric networks," in *Proc. Med-Hoc-Net*, 2016, pp. 1-8.

[18] S. He *et al.*, "Cloud-edge coordinated processing: Low-latency multicasting transmission," *IEEE J. on Sel. Areas in Commun.,* vol. 37, no. 5, pp. 1144-1158, 2019.

[19] J.-M. Hsu, H.-Y. Chiu, and Y.-S. Ye, "A partial cache for multimedia content in named data networking," in *Proc. International Conference on Platform Technology and Service*, 2015, pp. 37-38.

[20] M. Rahnamania and F. Ashtiani, "A New Analytical Approach for Delay Analysis in the Presence of Correlated Arrivals," in *Proc. Iran Workshop on Commun. and Inf. Theory*, 2024, pp. 1-6.

[21] M. Harchol-Balter, *Performance modeling and design of computer systems: queueing theory in action*. Cambridge University Press, 2013.

[22] X. Chao and M. Pinedo, *Queueing networks: Customers, signals, and product form solutions,* 1999.

[23] L. Kleinrock, *Queueing Systems: Theory*. Wiley, 1974.

[24] G. Latouche and V. Ramaswami, *Introduction to matrix analytic methods in stochastic modeling*, Society for Industrial and Applied Mathematics, 1999.

## معرفی نویسندگان:

مهران رهنمانیا فارغ‌التحصیل مقطع کارشناسی ارشد رشته مهندسی برق گرایش مخابرات سیستم در دانشگاه صنعتی شریف است. علاقه‌مندی‌های پژوهشی وی مدل‌سازی تصافی شبکه‌های مخابراتی با استفاده از ابزارهای مانند تئوری صف است. پژوهشهای وی در حال حاضر بر بررسی سن اطلاعات در سناریوهای ذخیره سازی مختلف در حافظه پنهان متمرکز است. او همچنین دارای تجربه در زمینه‌های تئوری بازی وMDP است.

**ایمیل:** mehran.rahnamania@ee.sharif.edu

فرید آشتیانی (عضو ارشد IEEE) دکترای خود را در رشته مهندسی برق از دانشگاه صنعتی شریف، تهران، ایران، در سال ۱۳۸۲ دریافت نمود. وی از سال ۱۳۷۴ تا ۱۳۷۸ در مرکز تحقیقات نیرو (P.R.C.) و پژوهشگاه نیرو (N.R.I.) ایران مشغول به کار شد. وی از سال ۱۳۷۸ تا ۱۳۸۰ به‌عنوان عضو پژوهشی آزمایشگاه تحقیقات علوم ارتباطات پیشرفته، مرکز تحقیقات مخابرات ایران (I.T.R.C.)، تهران، ایران فعالیت داشت. وی از سال ۱۳۸۲ در گروه مهندسی برق دانشگاه صنعتی شریف مشغول به کار بوده و در حال حاضر دانشیار همین گروه می باشد. زمینه های تحقیقاتی او شامل تئوری صف، مدل سازی، تحلیل و طراحی انواع مختلف شبکه های بی سیم و تحلیل عملکرد است.

**ایمیل:** ashtianimt@sharif.edu

---

[1] Cache
[2] Delay
[3] Load of networks
[4] Quality of Service
[5] Backhaul
[6] Base station
[7] Mobility of users
[8] Wireless edge
[9] Dynamic contents
[10] Age of information (AoI)

[11] Fetch
[12] Distributed storage
[13] Redundant request
[14] Replication
[15] Segments
[16] Partial caching
[17] pull-based
[18] push-based
[19] Content provider
[20] Cloud-edge caching
[21] Redundant arrivals
[22] Product form
[23] Whole caching whole delivery
[24] Partial caching whole delivery
[25] Partial caching partial delivery
[26] pull-push-based
[27] Poisson process
[28] Round-robin
[29] Negative exponential
[30] Fronthaul
[31] Downlink
[32] Block-fading
[33] Automatic repeat request
[34] Tandem queues
[35] Jackson queueing network
[36] Little's law
[37] Batch arrival
[38] Positive signal
[39] Negative signal
[40] Quasi-reversible
[41] Quasi-birth-death
[42] Matrix analytic methods
[43] Poisson arrivals see time averages
[44] Steady state
[45] Utilization factor
[46] Arrivals See Time Averages
[47] Residual life
[48] Renewal processes
[49] Erlang-2
[50] Discrete event simulation