بسم الله الرحمن الرحیم

دانشگاه یزد

دانشکده مهندسی برق و کامپیوتر

پایان‌نامه دوره کارشناسی ارشد مهندسی فناوری اطلاعات

گرایش شبکه‌های کامپیوتری

بهبود الگوریتم کنترل ازدحام TCP-Vegas

توسط:

شیرین اسدالهی

استاد راهنما:

دکتر فضل‌الله ادیب‌نیا

استاد مشاور:

دکتر مهدی آقا صرام – دکتر کیارش میزانیان

تابستان ۱۳۹۲

دانشکده مهندسی برق و کامپیوتر

جلسه دفاعیه

پایان‌نامه کارشناسی ارشد فناوری اطلاعات

خانم/آقای **.....شیرین اسدالهی.....**

با عنوان

بهبود الگوریتم کنترل ازدحام TCP-Vegas

در تاریخ ....۱۳۹۲/۷/۲..... با حضور استاد داور برگزار گردید و پس از ارزیابی مورد پذیرش قرار گرفت.

استاد راهنمای پایان‌نامه

دکتر فضل‌الله ادیب‌نیا

تاریخ و امضاء

کلیه حقوق مادی و معنوی مترتب بر نتایج مطالعات، ابتکارات و نوآوری‌های ناشی از تحقیق موضوع این پایان‌نامه/ رساله متعلق به دانشگاه یزد است و هرگونه استفاده از نتایج علمی و عملی از این پایان‌نامه/ رساله برای تولید دانش فنی، ثبت اختراع، ثبت اثر بدیع هنری، همچنین چاپ و تکثیر، نسخه برداری، ترجمه و اقتباس و ارائه مقاله در سمینارها و مجلات علمی از این پایان‌نامه/ رساله منوط به موافقت کتبی دانشگاه یزد است.

تقدیر و تشکر

به مصداق «من لم یشکر المخلوق لم یشکر الخالق » بسی شایسته است که از استاد راهنمای فرهیخته و فرزانه خود جناب آقای دکتر ادیب‌نیا که با کرامتی چون خورشید ، سرزمین دل را روشنی بخشیدند و گلشن سرای علم و دانش را با راهنمایی های کارساز و سازنده خود، بارور ساختند، تقدیر و تشکر نمایم.

از اساتید مشاور جناب آقای دکتر میزانیان و جناب آقای دکتر صرام نیز کمال امتنان را دارم.

همچنین از جناب آقای دکتر میرجلیلی و جناب آقای دکتر ظریف‌زاده که زحمت داوری این پایان‌نامه را بر عهده داشتند، تشکر می‌نمایم.

در نهایت از پدر و مادر عزیز، دلسوز و مهربانم که آرامش روحی و آسایش فکری فراهم نمودند تا با حمایت های همه جانبه در محیطی مطلوب ، مراتب تحصیلی و نیز پایان نامه درسی را به نحو احسن به اتمام برسانم، سپاسگزارم.

چکیده:

ازدحام یکی از عواملی است که می‌تواند کارایی را در شبکه‌های کامپیوتری کاهش دهد. جهت اداره کردن ازدحام در یک شبکه کامپیوتری از الگوریتم‌های کنترل ازدحام استفاده می‌شود. برخی از این الگوریتم‌ها عبارتند از TCP Tahoe، TCP Reno، TCP New-Reno، SACK و TCP-Vegas. مطالعات تاکنون، نشان می‌دهد که در شبکه‌های همگن، TCP-Vegas در بسیاری از موارد، بهتر از TCP-Reno عمل می-کند. اما هنگامی‌که در اینترنت از هر دو الگوریتم Vegas و Reno با هم استفاده می‌شود، Reno بهتر از Vegas عمل می‌کند. به همین دلیل، هیچ انگیزه‌ای برای سیستم عامل‌ها وجود ندارد که از Vegas بعنوان پروتکل پیش‌فرض لایه انتقال استفاده کنند. ما در این پایان نامه، یک Vegas بهبود یافته پیشنهاد می‌کنیم که با Reno سازگار است و زمانی که با Reno رقابت می‌کند، می‌تواند یک سهم عادلانه از منابع شبکه را بدست آورد. الگوریتم پیشنهادی، یک الگوریتم مبتنی بر فرستنده است. برخلاف TCP-Vegas که از پارامترهای α و β ثابت استفاده می‌کند، الگوریتم پیشنهادی، این پارامترها را با توجه به شرایط شبکه و به صورت پویا تعیین می‌کند. الگوریتم پیشنهادی با Vegas اصلی و سایر الگوریتم‌های بهبود یافته Vegas، نظیر Vegas-A و CODE TCP مقایسه شد. نتایج شبیه‌سازی نشان می‌دهد که الگوریتم پیشنهادی در رقابت با Reno، نسبت به سایر الگوریتم‌های بهبود یافته، سهم منصفانه‌تری از پهنای باند را بدست می‌آورد و دارای توان عملیاتی بالاتری است. بعلاوه الگوریتم پیشنهادی خصوصیات خوب الگوریتم Vegas را نیز حفظ می‌کند.

واژه‌های کلیدی:کنترل ازدحام، توان عملیاتی، عدالت[1]، TCP Vegas، TCP Reno

[1] . Fairness

فهرست مطالب

عنوان **صفحه**

فصل ۱- مقدمه ۱

فصل۲- الگوریتم‌های کنترل ازدحام ۴

۲-۱- مقدمه ۵

۲-۲- راهکارهای کنترل ازدحام ۶

۲-۲-۱- کنترل ازدحام حلقه-باز ۶

۲-۲-۲- کنترل ازدحام حلقه- بسته ۶

۲-۳- سیاست ازدحام ۷

۲-۴- تکنیک‌های کنترل ازدحام ۹

۲-۴-۱- TCP Tahoe ۹

۲-۴-۲- TCP Reno ۱۰

۲-۴-۳- TCP New-Reno ۱۱

۲-۴-۴- TCP SACK ۱۳

۲-۴-۵- TCP Vegas ۱۴

۲-۵- مقایسه الگوریتم‌های کنترل ازدحام ۱۵

۲-۵-۱- بهره‌وری منابع ۱۵

۲-۵-۱-۱- بهره‌وری پهنای باند ۱۶

۲-۵-۱-۲- تغییرات اندازه پنجره ازدحام ۱۷

۲-۵-۱-۳- نرخ انتقال مجدد ۱۸

۲-۵-۲- عدالت بین اتصالاتی با تاخیرهای مختلف ۱۹

۲-۵-۲-۱- رفتار اتصالاتی با تاخیرهای طولانی ۱۹

۲-۵-۲-۲- تغییر عدالت با تغییرات تاخیرها ۲۲

۲-۵-۲-۳- تاثیر الگوریتم‌های صف ۲۳
۲-۵-۳- عدالت بین انواع متفاوت TCP زمانی‌که روی یک لینک با هم رقابت می‌کنند..... ۲۵
۲-۵-۳-۱- رقابت کردن دو اتصال، زمانی‌که با هم در حال اجرا هستند ۲۶
۲-۵-۳-۲- تاثیر الگوریتم‌های صف ۲۸
۲-۵-۳-۳- بهبودهای دیگر..... ۳۱
۲-۵-۴- Reno-SACK-Vegas..... ۳۲
۲-۵-۵- نتیجه‌گیری ۳۴

فصل۳- الگوریتم کنترل ازدحام TCP Vegas ۳۵

۳-۱- مقدمه ۳۶
۳-۲- معرفی الگوریتم کنترل ازدحام TCP Vegas ۳۶
۳-۲-۱- چالش‌های TCP Vegas..... ۳۷
۳-۲-۱-۱- عدالت ۳۷
۳-۲-۱-۲- رفتار نامناسب اتصال‌های قدیمی‌تر ۳۸
۳-۲-۱-۳- دیگر مشکلات ۳۹
۳-۳- سوابق پژوهشی ۳۹
۳-۳-۱- TCP-Vegas-A ۴۰
۳-۳-۲- CODE TCP ۴۱
۳-۴- نتیجه‌گیری ۴۲

فصل۴- الگوریتم پیشنهادی ۴۳

۴-۱- مقدمه ۴۴
۴-۲- الگوریتم پیشنهادی ۴۴
۴-۳- شبیه‌سازی ۴۶
۴-۳-۱- اتصال Reno در مقابل یک اتصال Vegas\Vegas-A\CODE TCP\Proposed Algorithm ۴۷
۴-۳-۱-۱- تاثیر الگوریتم صف RED ۴۹
۴-۳-۲- پیوند گلوگاه با اندازه‌های بافر متفاوت ۴۹
۴-۳-۳- اتصالات با RTTهای متفاوت ۵۱
۴-۳-۴- نسبت متفاوت تعداد اتصالات ۵۳
۴-۳-۵- عدالت در سناریوهای همگن ۵۵
۴-۳-۶- مقایسه الگوریتم‌های کنترل ازدحام در شبکه‌های همگن ۵۸

فصل۵- نتایج و پیشنهادات ۶۵

۵-۱- نتایج ۶۶

۵-۲- پیشنهادات ۶۸

لغت‌نامه فارسی به انگلیسی ۶۹

لغت‌نامه انگلیسی به فارسی ۷۳

منابع ۷۷

فهرست جدول‌ها

عنوان **صفحه**

جدول ۲-۱: نرخ بهره‌وری پهنای باند با نرخ loss متفاوت..... ۱۶

جدول ۲-۲ نرخ انتقال مجدد برای اتصالات one-on-one ۱۸

جدول ۲-۳ تغییر عدالت با تغییرات RTT ۲۲

جدول ۲-۴-الف: تاثیر Maxthresh روی عدالت در رقابت Reno و Vegas ۳۱

جدول ۲-۴-ب: تاثیر Maxthresh روی عدالت در رقابت SACK و Vegas ۳۱

جدول ۴-۱: **مقدار شاخص عدالت ۱۰ اتصال در حال رقابت با مقادیر متغیر برای پهنای باند** Bottleneck..۵۶

جدول ۴-۲ میانگین توان عملیاتی اتصال مسیریاب برای الگوریتم‌های کنترل ازدحام..... ۵۹

فهرست شکل‌ها

عنوان **صفحه**

شکل ۲-۱: بار شبکه و تاثیر آن روی توان عملیاتی و تاخیر ۱۲
شکل ۲-۲: توپولوژی ۱ ..... ۱۶
شکل ۲-۳: تغییرات پنجره ازدحام با چهار packet loss ۱۷
شکل ۲-۴: توپولوژی ۲ ..... ۱۸
شکل ۲-۵: توپولوژی ۳ ..... ۱۹
شکل ۲-۶: اشغال پهنای باند توسط دو اتصال Reno ۲۰
شکل ۲-۷: اشغال پهنای باند توسط دو اتصال Vegas ۲۱
شکل ۲-۸: اشغال پهنای باند توسط دو اتصال SACK ۲۲
شکل ۲-۹: عدالت بین اتصالاتی با تاخیر متفاوت ۲۳
شکل ۲-۱۰: تاثیر صف Drop Tail بین اتصالاتی با تاخیر متفاوت ۲۴
شکل ۲-۱۱: تاثیر صف RED بین اتصالاتی با تاخیر متفاوت ۲۵
شکل ۲-۱۲-الف: اشغال پهنای باند توسط دو اتصال Reno و SACK ۲۶
شکل ۲-۱۲-ب: اشغال پهنای باند توسط دو اتصال Reno و Vegas ۲۷
شکل ۲-۱۲-ج: اشغال پهنای باند توسط دو اتصال SACK و Vegas ۲۷
شکل ۲-۱۳: نرخ اشغال پهنای باند برای سایزهای بافر متفاوت با صف Drop Tail ۲۹
شکل ۲-۱۴: نرخ اشغال پهنای باند برای سایزهای بافر متفاوت با صف RED ۳۰
شکل ۲-۱۵: نرخ‌های اشغال پهنای باند توسط پیاده‌سازی‌های Vegas اصلی و بهبود یافته ۳۲
شکل ۲-۱۶: توپولوژی ۴ ..... ۳۳
شکل ۲-۱۷: نرخ اشغال پهنای باند Reno، SACK و Vegas در رقابت با هم ۳۳
شکل ۴-۱: توپولوژی شبکه ۴۶
شکل ۴-۲: توان عملیاتی Reno و الگوریتم پیشنهادی با پهنای باند متفاوت ۴۷
شکل ۴-۳: میزان بهبود الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌ها ۴۷
شکل ۴-۴: توان عملیاتی Reno و الگوریتم پیشنهادی با پهنای باند متفاوت ۴۸
شکل ۴-۵: توان عملیاتی Reno و الگوریتم پیشنهادی با پهنای باند متفاوت و صف RED ۴۹
شکل ۴-۶: توان عملیاتی Reno و الگوریتم پیشنهادی با اندازه‌های بافر متفاوت ۵۰
شکل ۴-۷: میزان بهبود الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌ها ۵۰
شکل ۴-۸: توان عملیاتی دو اتصال Reno و الگوریتم پیشنهادی با RTT متفاوت ۵۲

شکل ۴-۹: میزان بهبود الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌ها ۵۲
شکل ۴-۱۰: تغییر پذیری در میانگین توان عملیاتی با نسبت متفاوت تعداد دو نوع اتصال TCP ۵۳
شکل ۴-۱۱: میزان بهبود الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌ها ۵۴
شکل ۴-۱۲: توان عملیاتی الگوریتم پیشنهادی با تعداد اتصالات متفاوت ۵۴
شکل ۴-۱۳: تغییرات CWND دو اتصال الگوریتم پیشنهادی روی یک پیوند گلوگاه ۵۵
شکل ۴-۱۴: تغییرات CWND دو اتصال روی یک پیوند گلوگاه..... ۵۷
شکل ۴-۱۵: توپولوژی شبکه ۵۸
شکل ۴-۱۶: توان عملیاتی الگوریتم‌های کنترل ازدحام..... ۵۹
شکل ۴-۱۷: میانگین طول صف الگوریتم TCP-Reno..... ۶۰
شکل ۴-۱۸: میانگین طول صف الگوریتم TCP-New Reno..... ۶۰
شکل ۴-۱۹: میانگین طول صف الگوریتم TCP-Vegas..... ۶۱
شکل ۴-۲۰: میانگین طول صف الگوریتم پیشنهادی..... ۶۱
شکل ۴-۲۱: تغییرات طول پنجره الگوریتم TCP-Reno ۶۲
شکل ۴-۲۲: تغییرات طول پنجره الگوریتم TCP-New Reno ۶۲
شکل ۴-۲۳: تغییرات طول پنجره الگوریتم TCP-Vegas ۶۳
شکل ۴-۲۴: تغییرات طول پنجره الگوریتم پیشنهادی ۶۳

فصل اول

مقدمه

ازدحام در شبکه‌های کامپیوتری، زمانی رخ می‌دهد که تعداد بسیار زیادی بسته، به یک زیرشبکه تحویل داده شود. زمانی که در شبکه ازدحام رخ می‌دهد، تاخیر افزایش می‌یابد و احتمال مفقود شدن بسته‌ها نیز در شبکه بالا می‌رود و در نتیجه توان عملیاتی و کارایی شبکه به شدت کاهش می‌یابد. این مسئله باعث شد تا برای مقابله با ازدحام راهکارهایی ارائه شود. یکی از این راهکارها، افزایش منابع شبکه بود که طبق مطالعات انجام شده[۱]، پردازنده‌های سریع و حافظه زیاد نمی-توانند مشکل ازدحام را به طور کامل حل کنند. در نتیجه با وجود منابع زیاد، باید از الگوریتم‌های کنترل ازدحام جهت مقابله با ازدحام استفاده شود. TCP یک پروتکل اتصال‌گرای انتها به انتها[۲] و قابل اطمینان است. به‌منظور بهبود کارایی شبکه، انواع مختلفی از پروتکل‌های کنترل ازدحام پیشنهاد شده است. دو نوعی که امروزه استفاده می‌شود، یکی Reno [۲] است که به طور گسترده در اینترنت به کار گرفته می‌شود و دیگری Vegas است که ادعا می‌شود، توان عملیاتی آن بیش‌تر از Reno است.

TCP-Vegas یکی از الگوریتم‌های کنترل ازدحام است که ازدحام را قبل از وقوع تشخیص می-دهد یعنی از وقوع ازدحام جلوگیری می‌کند در حالی که الگوریتم‌های TCP-Reno و TCP-New Reno بعد از وقوع ازدحام، از ادامه ازدحام پیشگیری می‌کنند. بین چندین خصوصیت جدید پیاده-سازی شده در TCP-Vegas که در [۳] مطرح شدند، یکی از مهمترین تفاوت‌های بین TCP-Vegas و TCP-Reno فاز اجتناب از ازدحام آن است. TCP-Reno وابسته به گم شدن بسته‌ها[۳] است تا ازدحام شبکه را تشخیص دهد درحالی که TCP-Vegas از یک طرح تخمین پهنای باند استفاده می‌کند تا ازدحام شبکه را حدس بزند.

TCP-Vegas یک الگوریتم کنترل ازدحام مبتنی بر تاخیر است. برخلاف TCP-Reno که برای تنظیم اندازه پنجره‌اش وابسته به گم شدن بسته‌ها است، TCP-Vegas از تاخیر صف به‌منظور اجتناب از ازدحام استفاده می‌کند. Vegas می‌تواند ازدحام شبکه را در مرحله اولیه تشخیص دهد و به طور متناوب از گم شدن بسته‌ها جلوگیری کند. در حالی که در Reno، معمولا بسته‌ها گم می‌شوند.

Vegas در مقایسه با Reno از نظر بهره‌وری شبکه[۴]، پایداری[۵][۴و ۵]، عدالت[۶][۴و ۵]، توان عملیاتی[۷] و گم شدن بسته‌ها در محیط‌های همگن بهتر عمل می‌کند. اگرچه مطالعات انجام شده

[2] . End-to-End
[3] . Packet Loss
[4] . Network Utilization

نشان داده است، زمانی که اتصالات Reno و Vegas در یک محیط شبکه با هم رقابت می‌کنند، Reno در مقایسه با Vegas سهم بیش‌تری از پهنای باند را بدست می‌آورد[۵، ۶و ۷]. به دلیل این ناسازگاری، Vegas به طور وسیع به کار گرفته نمی‌شود. در این پایان نامه، یک مکانیسم جدید از Vegas پیشنهاد می‌شود که اکثر عملکردهای آن مشابه با Vegas است اما بر خلاف Vegas که از پارامترهای ثابت α و β استفاده می‌کند، این الگوریتم، پارامترهای α و β را به طور پویا و با توجه به شرایط شبکه تغییر می‌دهد. زمانی که الگوریتم پیشنهادی، وقوع ازدحام را احساس می‌کند، به جای اینکه نرخ جریانش را کاهش دهد، پارامترهای α و β را افزایش می‌دهد. این باعث می‌شود که الگوریتم پیشنهادی بیش‌تر شبیه Reno رفتار کند. بنابراین الگوریتم پیشنهادی در رقابت با Reno طوری عمل می‌کند که کارایی آن کاهش نمی‌یابد.

بعد از این فصل که مقدمه‌ای بر الگوریتم کنترل ازدحام TCP Vegas و معرفی جایگاه این پروتکل است، در فصل دوم به معرفی الگوریتم‌های کنترل ازدحام پرداخته می‌شود و هر یک از آن‌ها به صورت مفصل توضیح داده خواهد شد. فصل سوم به بررسی دقیق الگوریتم کنترل ازدحام TCP Vegas و همچنین مروری بر کارهای انجام شده در راستای بهبود این الگوریتم اختصاص یافته است. در فصل چهارم، الگوریتم پیشنهادی ارائه شده جهت بهبود الگوریتم کنترل ازدحام TCP Vegas معرفی شده است و با Vegas اصلی و سایر الگوریتم‌های بهبود یافته Vegas نظیر Vegas-A و CODE TCP از جنبه توان عملیاتی مقایسه شده است. سرانجام در فصل پنجم، نتیجه‌گیری و جمع‌بندی مطالب مطرح شده در این پایان‌نامه ارائه گردیده و راهکارهایی جهت بهبود این الگوریتم در آینده پیشنهاد می‌شود.

---

[5] . Stability
[6] . Fairness
[7] . Throughput

فصل دوم

الگوریتم‌های کنترل ازدحام

۲-۱ مقدمه

یکی از نکات مهم در شبکه، ازدحام است. ازدحام در شبکه وقتی رخ می‌دهد که بار شبکه (تعداد بسته‌های ارسال شده به شبکه) بیشتر از ظرفیت شبکه باشد. کنترل ازدحام، راهکار و تکنیکی جهت کنترل ازدحام و نگهداشتن بار شبکه کمتر از ظرفیت شبکه است.

شاید این پرسش مطرح شود که چرا ازدحام در شبکه وجود دارد. ازدحام در شبکه به دلایل بعدی به وجود می‌آید [۲۴]:

- نرخ ترافیک ورودی از ظرفیت خطوط خروجی تجاوز کند. اگر یک جریان بسته از ۳ یا ۴ خط ورودی، وارد شود و همه آن‌ها نیاز به یک خط خروجی داشته باشند، یک صف تشکیل خواهد شد. اگر حافظه کافی برای نگهداری همه بسته‌ها وجود نداشته باشد، بسته گم خواهد شد. افزایش ظرفیت حافظه به میزان نامحدود این مشکل را حل نخواهد کرد. چون زمانی که بسته‌ها به جلوی صف می‌رسند، آن‌ها Timeout می‌شوند، در نتیجه مبدا بسته‌های تکراری را ارسال می‌کند که این بسته‌ها نیز به صف اضافه شده و باعث طولانی-تر شدن صف می‌شوند.

- بافر مسیریاب‌ها خیلی محدود باشد.
- اگر پردازنده‌ها در یک زیر شبکه دارای سرعت پایین باشند، ازدحام می‌تواند رخ دهد. با وجود پردازنده‌هایی با سرعت پایین در مسیریاب‌ها، کارهای عادی مانند صف‌بندی بافرها، بروزرسانی جداول و ... به آهستگی انجام خواهد شد.
- ازدحام به وسیله اتصال‌های[۸] آهسته نیز می‌تواند ایجاد شود. این مشکل زمانی حل خواهد شد که اتصال‌هایی با سرعت بالا استفاده شود. البته این راه حل در همه موارد صادق نیست. گاهی افزایش پهنای باند اتصال، می‌تواند مسئله ازدحام را بدتر کند. ازدحام هنگامی می‌تواند بدتر شود، که یک مسیریاب بافرهای آزاد نداشته باشد و شروع به دور انداختن بسته‌های ورودی کند. وقتی این بسته‌ها دور انداخته می‌شوند، فرستنده بعد از انقضای زمان آن‌ها را دوباره ارسال می‌کند. بنابراین بسته توسط فرستنده بارها دوباره فرستاده می‌شود تا زمانی که مبدا ACK این بسته‌ها را دریافت کند. ارسال چندین باره این بسته‌ها موجب طولانی شدن صف‌ها و بدتر شدن ازدحام می‌شود.

[8] . Link

۲-۲ راهکارهای کنترل ازدحام

راهکارهای کنترل ازدحام را می‌توان به دو دسته کنترل ازدحام حلقه- باز[۹] (جلوگیری) و کنترل ازدحام حلقه- بسته[۱۰] (برطرف کردن ازدحام) تقسیم کرد [۲۵].

۲-۲-۱ کنترل ازدحام حلقه- باز

در کنترل ازدحام حلقه- باز [۲۵]، سیاست‌هایی برای جلوگیری از ازدحام اعمال می‌شود. در این راهکارها، کنترل ازدحام توسط مبدا یا مقصد صورت می‌گیرد. لیستی از سیاست‌های جلوگیری از ازدحام به شرح زیر می‌باشند:

- **سیاست ارسال مجدد**[۱۱]: سیاست ارسال مجدد خوب، می‌تواند از ازدحام جلوگیری کند. سیاست ارسال مجدد و تایمرهای ارسال مجدد باید برای بهینه‌سازی کارایی و جلوگیری از ازدحام طراحی شوند.
- **سیاست اعلام وصول**[۱۲]: سیاست اعلام وصول که توسط گیرنده اعمال می‌شود، ممکن است بر ازدحام موثر باشد. اگر گیرنده هر بسته‌ای را که دریافت می‌کند، اعلام وصول نکند، ممکن است فرستنده را کند نماید و به جلوگیری از ازدحام کمک کند.
- **سیاست حذف**[۱۳]: سیاست حذف خوب که توسط مسیریاب‌ها اعمال می‌شود، ممکن است از ازدحام جلوگیری نماید ولی به جامعیت انتقال نیز آسیب نرساند. به عنوان مثال، در انتقال صوت، اگر سیاست موردنظر در هنگام ازدحام، بسته‌هایی با اهمیت کمتر را حذف نماید، کیفیت صوت حفظ می‌گردد و از ازدحام جلوگیری می‌شود.

۲-۲-۲ کنترل ازدحام حلقه- بسته

[9] . Open Loop
[10] . Closed Loop
[11] . Retransmission Policy
[12] . Acknowledgement Policy
[13] . Discarding Policy

راهکارهای کنترل ازدحام حلقه- بسته [۲۵] سعی می‌کنند، پس از ایجاد ازدحام آن را برطرف کنند. پروتکل‌های مختلف، از راهکارهای متفاوتی استفاده می‌کنند که بعضی از آن‌ها را شرح می-دهیم.

- **فشار عقبگرد**[۱۴]: وقتی مسیریاب دچار ازدحام شد، می‌تواند به اطلاع مسیریاب قبلی برساند که از نرخ بسته‌های خروجی بکاهد. این روند می‌تواند به مسیریاب‌های قبلی دیگر اعمال شود. این راهکار را فشار عقبگرد گویند.
- **نقطه انسداد:** نقطه انسداد، بسته‌ای است که مسیریاب به مبدا می‌فرستد تا اعلان کند که دچار ازدحام شده است. این نوع کنترل مانند کند شدن مبدا در ICMP است.
- **سیگنال‌دهی ضمنی**[۱۵]: مبدا می‌تواند سیگنال ضمنی منتظر ازدحام را تشخیص دهد و از نرخ ارسال خود بکاهد. به عنوان مثال، تاخیر اندکی در دریافت اعلام وصول می‌تواند سیگنالی برای ازدحام شبکه باشد.
- **سیگنال‌دهی صریح**[۱۶]: مسیریاب‌هایی که با ازدحام مواجه می‌شوند، می‌توانند سیگنال صریح را ارسال کنند. مثلا بیتی را در بسته مقداردهی کنند تا ازدحام را به اطلاع فرستنده یا گیرنده برساند.

۲-۳ سیاست ازدحام

سیاست کلی TCP برای اداره کردن ازدحام در سه فاز شروع کند[۱۷]، اجتناب از ازدحام[۱۸] و تشخیص ازدحام[۱۹] صورت می‌گیرد [۲۵]. در فاز شروع کند، فرستنده با نرخ انتقال کند شروع می-کند، ولی نرخ را با سرعت افزایش می‌دهد تا به حد آستانه[۲۰] برسد. در فاز دوم، پس از رسیدن به حد آستانه، نرخ داده کاهش می‌یابد تا از ازدحام اجتناب شود. سرانجام اگر ازدحام تشخیص داده شد، فرستنده به فاز شروع کند یا فاز اجتناب از ازدحام می‌رود که به چگونگی تشخیص ازدحام بستگی دارد.

---

[14] . Backpressure
[15] . Implicit Signaling
[16] . Explicit Signaling
[17] . Slow Start
[18] . congestion avoidance
[19] . congestion detection
[20] . Threshold

- **شروع کند (افزایش توانی):** یکی از الگوریتم‌هایی که در کنترل ازدحام TCP به کار می‌رود، شروع کند است. این الگوریتم بر این ایده استوار است که اندازه پنجره ازدحام (cwnd) با حداکثر اندازه یک قطعه (MSS) شروع می‌شود. MSS در اثنای برقراری اتصال، توسط انتخابی به همین نام تعیین می‌شود. اندازه پنجره، هر وقت که قطعه‌ای اعلام وصول می‌شود، یک MSS افزایش می‌یابد. این الگوریتم به کندی شروع می‌شود ولی به صورت توانی رشد می‌کند. شروع کند نمی‌تواند دائما ادامه داشته باشد. باید حد آستانه‌ای برای توقف این فاز وجود داشته باشد. فرستنده متغیری به نام حد آستانه شروع کند (ssThreshold) را نگهداری می‌نماید. وقتی اندازه پنجره برحسب بایت به این حد آستانه می‌رسد، شروع کند متوقف می‌شود و فاز بعدی شروع می‌گردد. در اغلب پیاده-سازی‌ها، مقدار آستانه ۶۵۵۳۵ بایت است.
- **اجتناب از ازدحام (رشد افزایشی):** اگر با الگوریتم شروع کند، آغاز کنیم، اندازه پنجره ازدحام به صورت توانی افزایش می‌یابد. برای اجتناب از ازدحام، باید رشد توانی کند شود. TCP الگوریتم دیگری به نام اجتناب از ازدحام را معرفی می‌کند که به جای توانی، رشد افزایشی دارد. وقتی اندازه پنجره ازدحام به آستانه شروع کند می‌رسد، فاز شروع کند متوقف و فاز افزایشی آغاز می‌شود. در این الگوریتم، هر وقت پنجره قطعات اعلام وصول شد، اندازه پنجره ازدحام یک واحد اضافه می‌گردد. در این مورد، پس از این که فرستنده، اعلام وصول‌های مربوط به تمام قطعات پنجره را دریافت کرد، اندازه پنجره به اندازه یک قطعه افزایش می‌یابد.
- **تشخیص ازدحام (کاهش ضربی):** اگر ازدحام به وجود آید، اندازه پنجره ازدحام باید کاهش یابد. تنها راهی که فرستنده می‌تواند حدس بزند که ازدحام رخ داده است، نیاز به ارسال مجدد قطعه است. اما ارسال مجدد در دو حالت رخ می‌دهد: وقتی زمان تایمر RTO به اتمام برسد یا سه ACK دریافت شود. در هر دو مورد، اندازه آستانه به نصف تقلیل می-یابد (کاهش ضربی). اغلب پیاده‌سازی‌های TCP دو عکس العمل دارند:

۱- اگر مهلت به اتمام برسد، احتمال ازدحام بیشتر است. احتمالا قطعه‌ای در شبکه حذف شده است و اطلاعی از قطعات ارسالی بعدی وجود ندارد. در این مورد، TCP با قدرت پاسخ می‌دهد و عملیات بعدی را انجام می‌دهد:

الف) مقدار آستانه را به نصف اندازه فعلی پنجره تقلیل می‌دهد.