

# مروری بر پروتکل های کنترل ازدحام

### ۳-۱ پیشگفتار

همانطور که در رویکردهای کنترل ازدحام اشاره شد، عملیات کنترل ازدحام در دو جایگاه اساسی مورد بحث و بررسی قرار گرفته و به اهم موضوعات به تفکیک جایگاه و روش اشاره می گردد. هدف غایی این فصل معرفی پروتکل های کنترل ازدحام است.

### ۳-۲ دسته بندی روش های کنترل ازدحام در لایه انتقال

روش های کنترل ازدحام از لحاظ نوع بازخورد دریافتی از شبکه به سه گروه تقسیم می شوند، که در ادامه این بخش هر گروه به اختصار شرح داده می شود و برای هر گروه چند نمونه از پروتکل های معروف بررسی می گردد [۱۲].

#### ۳-۲-۱ کنترل ازدحام بر پایه مهلت زمانی<sup>۱</sup>

در این روش پس از ارسال داده و قبل از ارسال مجدد داده به میزان زمان که معمولاً دوبرابر حداکثر مدت زمان رفت و برگشت بسته است، صبر خواهیم کرد. در صورتی که پس از این زمان، تصدیق<sup>۲</sup> دریافت بسته ها از مقصد به دست فرستنده نرسد، به منزله وقوع ازدحام در شبکه تلقی شده و میزان داده ارسالی (پنجره ارسال) در بازه ی زمانی بعدی کاهش می یابد.

#### ۳-۲-۱-۱ معرفی TCP-Tahoe

یکی از اولین راهکارهایی که توسط جکوبسون<sup>۳</sup> ارائه گردید. الگوریتمی که در این روش ارائه گردید از ۳ بخش اساسی تشکیل شده که در ادامه به هر کدام از آنها اشاره خواهیم کرد [۱۳].

۱. تخمین زمان ارسال مجدد<sup>۴</sup> (RTO): عامل TCP پس از ارسال هر بسته زمان ارسال بسته را ثبت خواهد کرد. در صورتی که پس از گذشت یک زمان تخمینی تصدیق دریافت بسته

<sup>۱</sup> Timeout-Based Congestion Control

<sup>۲</sup> Acknowledgment

<sup>۳</sup> Jacobson

<sup>۴</sup> Retransmission Time Estimation

دریافت نشود، فرض بر این است که بسته در جایی از مسیر به دلیل ازدحام ازدست رفته، لذا عامل، بسته را مجددا ارسال خواهد کرد. تخمین صحیح این زمان از اهمیت بالای برخوردار است. در صورتیکه این زمان کوچک باشد باعث ارسال چندباره یک بسته خواهد شد، که موجب تشدید ازدحام می گردد و در صورت بزرگ بودن عامل ازدسته رفتن بسته ها بیشتر به طول می انجامد و گذردهی جریان را کاهش خواهد داد.

۲. بخش دوم در واقع بهبودی برای دسته اول است، هر چند به کمک RTO ازدست رفتن بسته را می توان با اطمینان بالایی تشخیص داد، اما این روش بسیار کند به نظر می رسد. در TCP استاندارد، تنها راه برای تشخیص ازدست رفتن بسته دریافت نشدن Ack پس از گذر زمان RTO است، در حالیکه از بسته های Ack تکراری می توان به این منظور استفاده کرد. به زبان ساده تر در صورت دریافت بسته های Ack تکراری می توان به سرعت بسته ای که ازدست رفته را تشخیص داده و مجددا ارسال نمود. لازم به توضیح است که بسته ای Ack حداکثر پس از مدت زمان  $RTT^1$  به دست عامل خواهد رسید.

۳. آخرین و مهمترین بخش در این پروتکل، فاز آغاز آهسته<sup>۲</sup> (SS) و فاز جلوگیری از ازدحام<sup>۳</sup> (CA) است. در فاز آغاز آهسته عامل TCP سعی خواهد کرد که حداکثر ظرفیت کانال را شناسایی کند و در صورت وقوع ازدحام پنجره ارسال خود را به صورت ضربی کاهش<sup>۴</sup> می دهد، (اندازه پنجره ارسال نصف خواهد شد.) و در فاز جلوگیری از ازدحام اندازه پنجره ارسال با رشد خطی افزایش خواهد یافت، در این فاز عامل سعی خواهد کرد که گذردهی جریان TCP را افزایش دهد و بیشترین داده ممکن را قبل از وقوع ازدحام ارسال کند.

در شکل ۳-۱ روش آغاز آهسته و فاز پرهیز از ازدحام در TCP-Tahoe نشان داده شده

است.

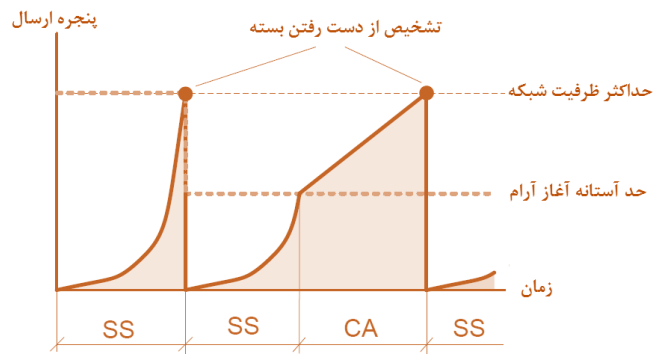
---

<sup>۱</sup> Round Trip Time

<sup>۲</sup> Slow Start

<sup>۳</sup> Congestion Avoidance

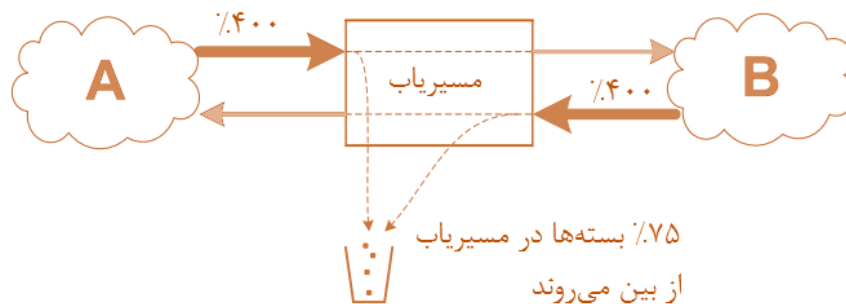
<sup>۴</sup> Multiplicative Decrease



شکل ۱-۳: روش آغاز آهسته و فاز پرهیز از ازدحام

### ۲-۱-۲-۳ معرفی TCP Reno

همانطور که در TCP Tahoe مشاهده شد در صورت از دست رفتن یک بسته، اندازه پنجره ارسال بازنشانی شده و مجدداً از مقدار ۱ افزایش می‌یابد، این واکنش سخت‌گیرانه و انفجاری، گذردهی جریان را که از الگوریتم Tahoe استفاده می‌کند، را کاهش می‌دهد. به عنوان مثال شکل ۲-۳ را در نظر بگیرید، در صورتیکه نرخ ارسال ما ۴ برابر گذردهی مسیریاب باشد، گذردهی جریان را تا ۷۵٪ کاهش را تجربه خواهد کرد.



شکل ۲-۳: افزایش نمای Packet Loss در صورت افزایش سربار مسیریاب

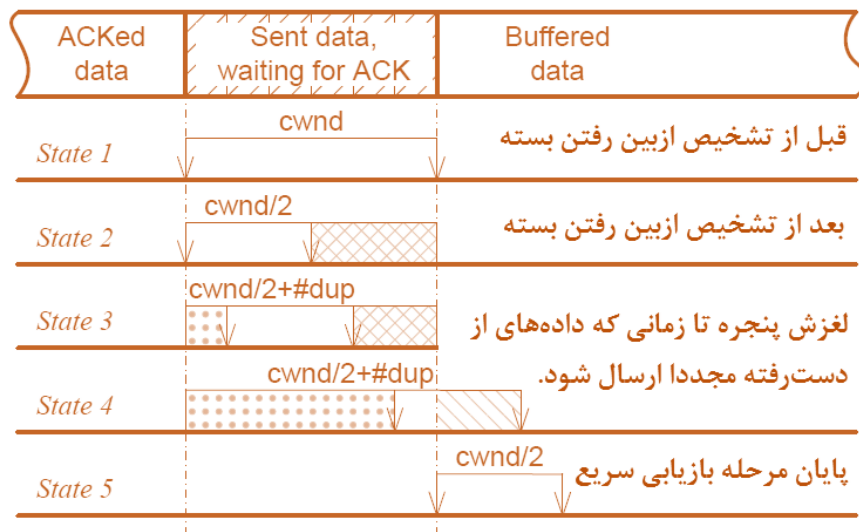
برای حل این مشکل جکوبسون دو مفهوم جدید را معرفی نمود:

۱. وقوع ازدحام بزرگ<sup>۱</sup>: عدم دریافت تصدیق دریافت بسته‌ها (Ack) پس از گذشت یک بازه‌ی زمانی خاص (مانند: RTO و یا RTT) نشان می‌دهد که برای یک بازه‌ی زمانی محدود هیچکدام

<sup>۱</sup> Major Congestion Event

از بسته‌ها به مقصد نرسیده‌اند، می‌توان گفت که احتمالاً شبکه شدیداً دچار ازدحام شده است، در چنین شرایطی، بازنشانی<sup>۱</sup> پنجره ارسال (آغاز آهسته با اندازه اولیه ۱ واحد برای پنجره ارسال) معقول به نظر می‌رسد.

۲. وقوع ازدحام کوچک<sup>۲</sup>: یک حالت کاملاً متفاوت در شبکه دریافت بسته‌های Ack تکراری است. به عنوان نمونه دریافت ۴ بسته Ack تکراری به این معناست که بسته‌ها به مقصد می‌رسند و گاهی ۱ یا ۲ بسته از دست رفته‌اند، در چنین شرایطی خواهیم گفت که شبکه به صورت جزئی دچار ازدحام گردیده، زیرا توانایی انتقال بسته‌ها کاملاً از بین نرفته و تعدادی از بسته‌ها به مقصد نرسیده‌اند. در چنین شرایطی الگوریتم Reno وارد فاز بازیابی سریع<sup>۳</sup> خواهد شد.



شکل ۳-۳: روش رشد و حرکت پنجره لغزان در TCP Reno

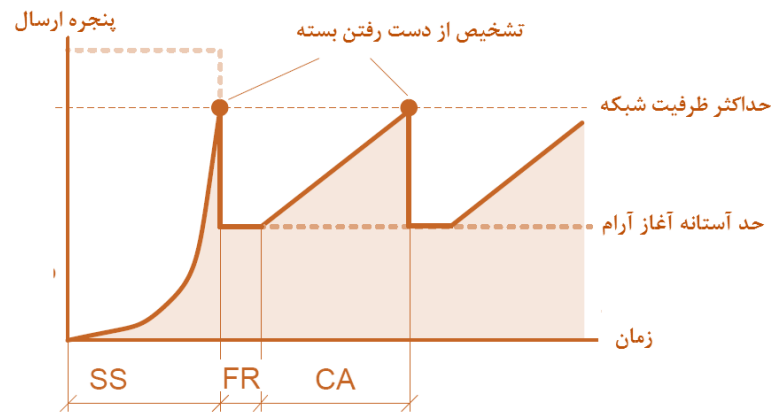
همانطور که در شکل ۳-۳ و شکل ۴-۳ خواهید دید، در صورت وقوع ازدحام کوچک، پنجره ارسال نصف خواهد شد و در ادامه تمامی بسته‌های که از دست رفته‌اند، ارسال می‌شوند، این فاز

<sup>۱</sup> Reset

<sup>۲</sup> Minor Congestion Event

<sup>۳</sup> Fast Recovery

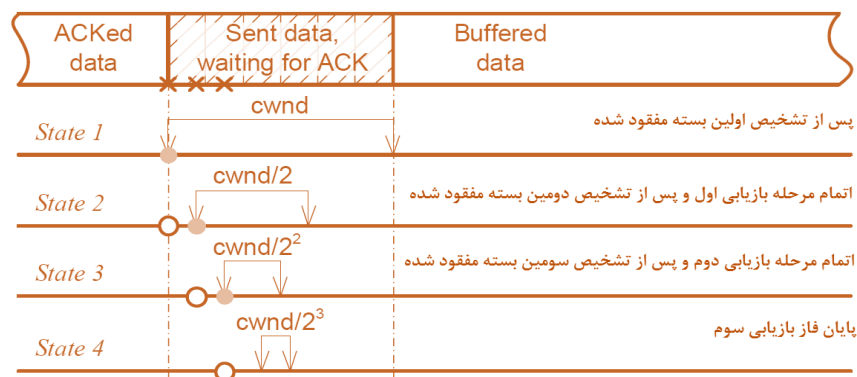
زمانی به پایان می‌رسد که دیگر Ack تکراری دریافت نشود، پس از پایان این فاز وارد فاز پرهیز از ازدحام خواهیم شد.



شکل ۳-۴: نمایش فازهای آغاز آهسته و پرهیز از ازدحام در پروتکل Reno

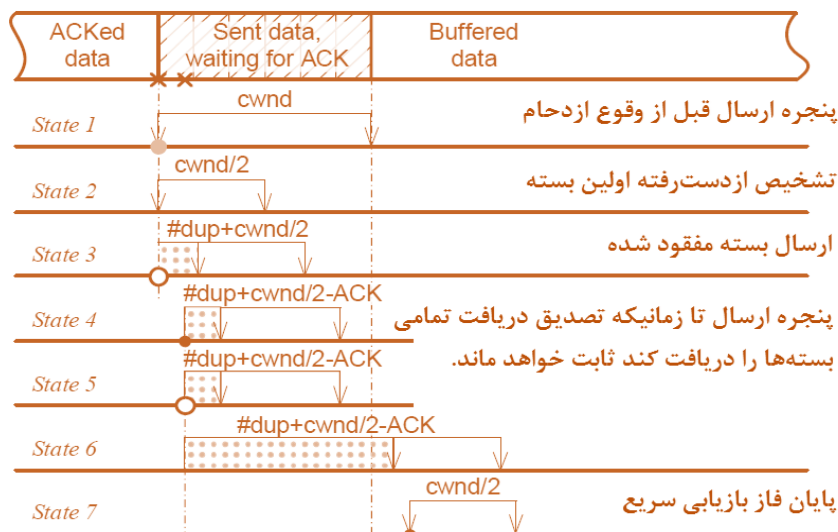
### ۳-۱-۲-۳ معرفی TCP NewReno

یکی از نقاط ضعف در فاز بازیابی سریع (FR) هنگامی است، که چندین بسته به صورت زنجیره‌ای از بین می‌روند. که باعث افزایش زمان بازیابی خواهد شد. در شکل ۳-۵، فاز بازیابی سریع ۳ مرتبه انجام شده و اندازه پنجره ارسال ۳ مرتبه نصف گردیده است، در نتیجه گذردهی الگوریتم در محیط‌های که بارکاری بالای دارند، کاهش می‌یابد.



شکل ۳-۵: فاز بازیابی سریع در TCP Reno

به جهت رفع این مشکل الگوریتم NewReno ارائه گردید [۱۳]. فلوید<sup>۱</sup> الگوریتم بازیابی سریع را اصلاح کرد، در شکل ۳-۶ الگوریتم اصلاح شده را مشاهده می کنید. با ذکر یک مثال این الگوریتم را شرح می دهیم، فرض کنید پنجره ای به طول ۱۰۰ بلوک از داده های بافر شده را ارسال نموده و قطعه ای از ۱۰ بلوک داده از دست رفته است. برای جلوگیری از کاهش ۱۰ باره پنجره ارسال در هنگام ورود به فاز بازیابی سریع تا هنگام دریافت تصدیق تمامی بلوک های داده موجود در این پنجره، از فاز بازیابی سریع (FR) خارج نخواهیم شد. به زبان ساده در صورت از دست رفتن بسته های یک پنجره فرض بر بروز ازدحام است و اندازه پنجره ارسال یک مرتبه به نصف تقلیل می یابد.



شکل ۳-۶: روش رشد و حرکت پنجره لغزان در TCP NewReno

### ۳-۲-۲ اجتناب از ازدحام بر پایه اعلان صریح ازدحام<sup>۲</sup>

در طول ۳ دهه گذشته، تلاش های فراوانی در جهت بهبود الگوریتم های کنترل ازدحام صورت پذیرفته است. تقریباً کلیه روش هایی که در بخش قبل به آنها اشاره شد در شبکه هایی با پهنای باند بالا دچار افت شدید گزردهی می شوند. فرض از دست رفتن بسته ها به دلیل ازدحام در شبکه های گسترده<sup>۳</sup> و یا بیسیم کامل نیست و بسته ها به دلایلی مانند اختلال<sup>۴</sup> و خطا در هنگام انتقال حذف

<sup>۱</sup> Floyd

<sup>۲</sup> ECN-Based Congestion Avoidance (ECN stands for: Explicit Congestion Notification)

<sup>۳</sup> Wide Area Network

<sup>۴</sup> Noise

می شوند . پروتکل های برپایه اعلان صریح ازدحام تلاش خواهند کرد که در شبکه های با پهنای باند بالا، گذردهی جریان ها را تضمین کرده و از ایجاد ازدحام و افزایش تاخیر صف و ایجاد صف های طولانی جلوگیری نمایند [۱ و ۱۵].

به کمک این روش برای ارسال پیام اخطار به فرستنده، بسته ها حذف نخواهند شد، بلکه با قرار دادن اطلاعاتی در سرآیند بسته ها به فرستنده در مورد میزان احتمال وقوع ازدحام اخطار خواهیم داد. در این روش گره های میانی شبکه نسبت به بار کاری و حداکثر ظرفیت شبکه آگاه و حساس هستند. در صورتیکه بار کاری ارتباط از یک آستانه خاص عبور کند، قبل از وقوع ازدحام به فرستنده ها در شبکه اطلاع داده می شود. هر فرستنده پس از دریافت پیام اخطار متناسب با سطح اخطار دریافتی پنجره ارسال خود را کاهش می دهد.

### ۳-۲-۱ معرفی XCP<sup>۱</sup>

این پروتکل با هدف بهبود گذردهی در شبکه هایی با پهنای باند بالا و تاخیر بالا طراحی گردید [۱۶]. در این روش فرستنده از درجهی ازدحام در شبکه توسط بازخوردی که از مسیر یاب ها دریافت می کند، آگاه خواهد شد. که به فرستنده کمک می کند، در صورتیکه شبکه شدیداً دچار ازدحام شده باشد با سرعت بیشتری اندازه پنجره خود را کاهش دهد و در صورت وقوع ازدحام جزئی با سرعت کمتری اندازه پنجره خود را کاهش دهد. در نتیجه گذردهی جریان بالا خواهد رفت و همچنین پاسخگویی نیز افزایش می یابد.

در صورت افزایش تاخیر بازخورد، سیستم ناپایدار خواهد شد. در نتیجه در صورت افزایش بیش از حد تاخیر، باید سرعت ارسال داده را کاهش دهیم. سوال اساسی یافتن ارتباط بهینه بین میزان تاخیر و بازخورد است، که سیستم در حالت پایدار بماند.

یکی دیگر از نکات مثبت در این روش، جدا شدن روش کنترل گذردهی از روش کنترل عدالت است. جهت افزایش گذردهی، از میزان پهنای باند غیرفعال استفاده می شود. در حالیکه برای افزایش

---

<sup>۱</sup> eXplicit Congestion Protocol



عدالت؛ پهنای باند جریان‌هایی که بیش از سهم عادلانه خود از پهنای باند در گلوگاه استفاده می‌کنند، اصلاح خواهند شد و در بازه‌ای منصفانه قرار خواهند گرفت و پهنای باند آزاد شده به سایر جریان‌ها اختصاص خواهد یافت.

از دیگر مزایای این روش نگهداری وضعیت شبکه<sup>۱</sup> در بسته‌های هر جریان است، بنابراین مسیریاب نیازی ندارد که به ازاء هر جریان وضعیت را در داخل مسیریاب نگهداری نماید. بدین صورت تعداد جریان‌های که می‌توان با این روش مدیریت کرد، افزایش خواهد یافت.

### ۳-۲-۲-۲ معرفی VCP<sup>۲</sup>

الگوریتم VCP با در نظر داشتن استانداردهای موجود در سرآیند لایه‌ی شبکه یک راه‌کار عملی ارائه می‌نماید [۱۷]. در این روش با کمک گرفتن از دو بیت اعلان صریح ازدحام (که در سرآیند لایه IP قرار دارند) به عنوان شناسه‌ای جهت تشخیص وضعیت ازدحام در شبکه، استفاده شده است. در اینجا خلاصه‌ای از چگونگی عملکرد این الگوریتم ارائه می‌شود.

هر مسیریاب بارکاری<sup>۳</sup> را به صورت بلادرنگ محاسبه می‌نماید، گیرنده جریان TCP پس از دریافت بسته، بازخورد دریافت شده از مسیریاب را از آن خارج می‌کند و در قالب بسته‌ی Ack برای فرستنده جریان ارسال می‌نماید.

بارکاری در سه سطح پایین، بالا و سربار بالا<sup>۴</sup> دسته‌بندی می‌گردد. سپس این سه سطح به صورت یک کد باینری دوبیتی به عنوان بیت‌های ECN<sup>۵</sup> در سرآیند لایه IP قرار می‌گیرد. در ادامه فرستنده با توجه به بازخوردی که از شبکه دریافت می‌کند، سرعت افزایش و کاهش پنجره ارسال را تنظیم می‌نماید. به عنوان مثال در وضعیت بارکاری پایین؛ پنجره ارسال به صورت ضربی<sup>۶</sup> افزایش

<sup>۱</sup> Network State – (Congestion Free , Congestion Experienced)

<sup>۲</sup> Variable-structure congestion Control Protocol

<sup>۳</sup> Load Factor

<sup>۴</sup> Low-Load, High-Load, Overload

<sup>۵</sup> Explicit Congestion Notification

<sup>۶</sup> Multiplicative

خواهد یافت و یا در حالت بارکاری بالا؛ پنجره ارسال به صورت خطی<sup>۱</sup> افزایش می یابد و در صورت وقوع ازدحام و افزایش بیشتر بارکاری پنجره ارسال به صورت ضربی کاهش می یابد.

از مزایای این روش می توان به عدم تغییر در سرآیند IP اشاره کرد. همچنین بارمحاسباتی این روش در مسیرپابها بسیار اندک است و نیاز به نگهداری وضعیت هر جریان در مسیرپاب نیست. در ضمن میزان تغییرات در الگوریتم های TCP در سمت گیرنده و فرستنده بسیار جزئی است. در کنار این خصوصیات، گذردهی و عدالت<sup>۲</sup> در تسهیم منابع الگوریتم VCP تقریباً با پروتکل XCP برابری می کند.

### ۳-۲-۲-۳ معرفی الگوریتم کنترل ازدحام به کمک بازخورد چند سطحی<sup>۳</sup>

الگوریتم MLCP نیز همانند الگوریتم VCP از شناسه ی بارکاری که توسط مسیرپابها محاسبه می شود، استفاده می نماید [۱۸]. گیرنده بسته داده اطلاعات بازخورد دریافتی از مسیرپاب را در بسته های Ack قرار می دهد. الگوریتم کنترل جریان در فرستنده به کمک همین بازخورد نرخ ارسال داده (پنجره ارسال) را تنظیم می نماید.

تفاوت این روش با پروتکل VCP در تعداد بیت هایی است که برای انعکاس وضعیت شبکه به دست فرستنده جریان می رسد، MLCP از ۴ بیت برای سطوح مختلف ازدحام در شبکه استفاده می نماید. روش VCP در فاز پرهیز از ازدحام از روش AIMD<sup>۴</sup> استفاده میکند، در حالیکه MLCP از روش AIIMD<sup>۵</sup> استفاده می نماید. در ادامه به اختصار به این مفاهیم و نتیجه استفاده از آنها اشاره شده است.

دلیل استفاده از ۴ بیت برای تعیین سطوح ازدحام چیست؟ نتایج آزمایشات و شبیه سازی ها نشان داد که در صورت استفاده از ۴ بیت می توان به گذردهی بسیار مطلوب و نزدیک به حالت بهینه

<sup>۱</sup> Addictive Increase

<sup>۲</sup> Fairness

<sup>۳</sup> Multi-level Feedback Congestion Control Protocol

<sup>۴</sup> Addictive Increase - Multiplicative Decrease

<sup>۵</sup> Addictive Increase – Inversely Proportional Increase - Multiplicative Decrease

دست یافت [۱۸]. در ضمن زمان همگرایی به سهم عادلانه و زمان مورد نیاز جهت کاهش نرخ ارسال در هنگام وقوع ازدحام کاهش خواهد یافت.

در این روش نرخ ارسال با توجه به میزان پهنای باند آزاد و همچنین تاخیر بازخورد تعیین می شود، که از «نوسانات شدید گذردهی»<sup>۱</sup> جلوگیری خواهد کرد.

بارکاری در مسیریاب ها به صورت پویا و بر اساس RTT هر جریان محاسبه می گردد، به همین جهت تسهیم عادلانه پهنای باند در میان جریان هایی با RTT متفاوت با دقت بیشتری انجام خواهد شد.

پروتکل های کنترل ازدحام بر پایه اعلان صریح ازدحام در مراکز داده در فصل چهارم تشریح شده است.

### ۳-۲-۳ اجتناب از ازدحام بر پایه تاخیر<sup>۲</sup>

روشی که در بالا به آن اشاره شد، کارکرد بسیار موثری در کنترل ازدحام دارد، اما در مواقعی که داده بعد از عبور از شبکه های ناهمگن با معماری متفاوت به مقصد می رسند، نمی توان از اعلان صریح ازدحام استفاده کرد. دلیل این امر ساده است، ممکن است بازخورد ارسال شده از یک گره، برای منبع دیگر قابل درک نباشد، و عملکرد سیستم را با مشکل مواجه کند.

از سوی دیگر در روش های کنترل ازدحام که بر پایه مهلت زمانی هستند، فرستنده ازدحام را پس از وقوع آن تشخیص می دهد، که در نتیجه آن اندازه پنجره ارسال خود را به نصف کاهش می دهد. در هنگام وقوع ازدحام عامل TCP به دلیل بی اطلاعی از میزان بارکاری شبکه بر شدت ازدحام می افزاید، این امر علاوه بر افزایش تاخیر صف و از بین بردن بسته ها، باعث کاهش شدید گذردهی خط هم خواهد شد. در بخش قبل به مسئله تغییرات دوره ای نرخ ارسال، RTT، گذردهی و طول صف اشاره شد.

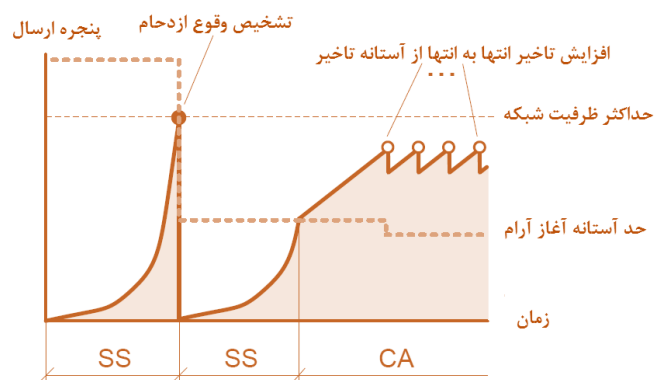
<sup>۱</sup> Throughput Oscillation

<sup>۲</sup> Delay-Based Congestion Avoidance

خوشبختانه از یک روش جایگزین می توان بهره جست. ایده اصلی در این روش بر تخمین و پیش بینی تاخیر بسته ها استوار است. سپس به کمک این اطلاعات نرخ مناسب ارسال داده با در نظر داشتن میزان بار کاری شبکه تعیین می شود. این دسته از الگوریتم های کنترل ازدحام پیش از وقوع ازدحام نرخ ارسال داده خود را کاهش می دهند.

### ۳-۲-۳-۱ معرفی TCP DUAL

TCP-Tahoe در جلوگیری از تشدید ازدحام در شبکه اینترنت سهم بسزایی را ایفا کرد. هر چند گذردهی شبکه به صورت قابل ملاحظه ای افزایش یافت، لیکن دامنه ی تغییرات گذردهی و دامنه تغییرات تاخیر بسیار بالا بود. مشکلاتی نظیر: کاهش کارایی بافرها<sup>۱</sup>، تغییرات تاخیر بالا<sup>۲</sup> و بالا بودن نرخ ازدست رفتن بسته ها همچنان پابرجا بود.



شکل ۳-۷: نمایش فازهای آغاز آهسته و پرهیز از ازدحام در TCP DUAL

وانگ<sup>۳</sup> و کراوکروفت<sup>۴</sup> پروتکل کنترل ازدحام TCP DUAL را معرفی کردند [۱۳]. در شکل ۳-۷ ایده الگوریتم DUAL مشاهده می شود. در این روش فاز CA اصلاح شد. فرض کنید گیرنده TCP با دریافت هر بسته پیام Ack بدون تاخیر برای فرستنده ارسال می کند، ایده این روش در

<sup>۱</sup> Buffer Utilization

<sup>۲</sup> High jitter

<sup>۳</sup> Wang

<sup>۴</sup> Crowcroft

اینجاست؛ عامل با در نظر گرفتن «تغییرات RTT»<sup>۱</sup> (اختلاف زمان ارسال بسته و دریافت Ack) تخمینی از تاخیر صفبندی<sup>۲</sup> را بدست می آورد و در صورت افزایش RTT و گذر از یک آستانه، قبل از وقوع ازدحام و از دست رفتن بسته ها اندازه پنجره ارسال خود را کاهش می دهد. در واقع با استفاده از تاخیر صفبندی در شبکه آستانه ازدحام را پیش بینی نموده و از وقوع ازدحام پرهیز خواهد کرد.

الگوریتم DUAL نوسانات گذردهی کانال و همچنین نوسانات طول صف را مهار کرد. تلاش های صورت گرفته در مهار و کنترل بهینه منابع در این روش بسیار موثر هستند، اما متأسفانه به طور کامل مشکلات مذکور برطرف نگردید. همچنین تخصیص عادلانه منابع به کمک این روش به درستی صورت نمی پذیرد.

### ۳-۲-۳ معرفی TCP Vegas

الگوریتم Vegas توسط Brakmo و Peterson ارائه گردید [۱۹]. ایده اصلی در این الگوریتم، بر پایه تخمین میزان بافر اشغال شده در مسیر یاب استوار است. مشابه روش DUAL این الگوریتم هم بر پایه اندازه گیری RTT عمل می کند. کمینه RTT مشاهده شده در طول اتصال به معنای یک مقدار مرجع برای تشخیص حالت «فارغ از ازدحام»<sup>۳</sup> شبکه به کار می رود. به عبارت دیگر در صورت افزایش RTT در طول اتصال، این مسئله به عنوان افزایش طول صف و کم شدن فضای آزاد بافر در مسیر یاب تفسیر می گردد. برخلاف الگوریتم DUAL در الگوریتم Vegas سعی شده که تعداد بسته های موجود در صف به صورت کمی تفسیر شده و یک مقدار قطعی و نه نسبی برای تعداد بسته های موجود در صف مسیر یاب گلوگاه<sup>۴</sup> بدست آید.

الگوریتم TCP Vegas با ثابت نگه داشتن نرخ ارسال در ناحیه ای پایدار به نحوی بسیار موثر گذردهی جریان TCP را بهبود می بخشد. متأسفانه، تحقیقات اخیر نشان می دهد که جریان های

<sup>۱</sup> Jitter

<sup>۲</sup> Queuing Delay

<sup>۳</sup> Congestion-Free State

<sup>۴</sup> Bottleneck Router

Vegas در رقابت با جریان های Reno در کسب سهم عادلانه پهنای باند موفق نیستند [۱۳]. به عنوان مثال در شرایطی که چندین مسیر<sup>۱</sup> برای رسیدن به مقصد وجود دارد، الگوریتم در تخمین صحیح منابع آزاد شبکه دچار خطا خواهد شد. جریان های جدید به دلیل مشاهده ناصحیح  $RTT_{min}$  سهم بیشتری از پهنای باند را اشغال می نمایند.

### ۳-۲-۳ معرفی TCP Vegas+

همانطور که در بخش قبلی به آن اشاره شد، الگوریتم Vegas در رقابت برای تصاحب منابع شبکه در مقابل الگوریتم Reno شکست خواهد خورد. دلیل این شکست بسیار ساده است، در ابتدا دو جریان به سرعت پنجره ارسال خود را افزایش می دهند، دیری نمی پاید که با افزایش حجم بافر اشغال شده و افزایش  $RTT$  الگوریتم Vegas وارد فاز پرهیز از ازدحام خواهد شد، در حالیکه الگوریتم Reno بدون توجه به چنین مسئله ای تا پر شدن کامل بافر و از دست رفتن بسته ها پنجره ارسال خود را افزایش می دهد. (دو الگوریتم که یکی به دلیل پرهیز از ازدحام پنجره خود را کاهش می دهد و دیگری که به دلیل نداشتن وضعیت شبکه پنجره خود را افزایش می دهد). نهایتاً بافر توسط جریان های Reno اشغال خواهد شد.

هاسه گاو<sup>۲</sup> با تشخیص و مشاهده این مسئله در شبکه، راه حلی ساده و موثر را پیشنهاد کرد [۱۳]. راه حل پیشنهادی شامل تقسیم بندی ترافیک ها در شبکه به ترافیک های سازگار با Vegas و ناسازگار با Vegas است. در ابتدا الگوریتم با فرض اینکه تمامی جریان ها با او سازگار هستند، شروع به فعالیت خواهد کرد و در ادامه در صورتیکه احتمال دهد؛ در رقابت با جریان های ناسازگار است، الگوریتم کنترل ازدحام خود را به الگوریتم Reno تغییر می دهد. با این روش رقابت ناعادلانه در تخصیص پهنای باند از بین خواهد رفت.

<sup>۱</sup> Multipath Routing

<sup>۲</sup> Hasegawa