# فصل سیزدهم

## مدیریت فعال صف و زمانبندی درشبکه های IP

### ۱۳-۱- مقدمه

یکی از مکانیزم های مهم تامین کیفیت سرویس وجلوگیری از وقوع ازدحام درشبکه های IP ، استفاده از مکانیزم های مدیریت فعال صف و زمانبندی درمسیریاب های IP می باشد. با کمک این مکانیزم ها، وقوع تراکم کنترل شده و از کاهش کارایی شبکه جلوگیری می شود. درادامه به بررسی مفاهیم و مکانیزم های مدیریت فعال صف و زمانبندی درشبکه های IP می پردازیم.

### ۱۳-۲- مدیریت فعال صف

همانطور که در فصل قبل به آن اشاره شد، رفتاری که در هرنود برروی جریان های ترافیکی عبوری ازآن اعمال میشود، توسط PHB مشخص می گردد. درPHB نحوه تخصیص پهنای باند و نحوه مدیریت صف مشخص می شود.

ازدحام در شبکه زمانی اتفاق می‌افتد که بسته‌ها با سرعتی بالاتر از آنچه بتوان آنها را انتقال داد به درگاه خروجی وارد شوند. در حالت کلی دو کلاس الگوریتم وجود دارند که مسیریابها به منظور کنترل ترافیک شبکه از آنها استفاده می‌کنند:

۱- الگوریتمهای زمانبندی صف که میزان پهنای باند تخصیص یافته به هر کلاس سرویس را در درگاه خروجی مدیریت می‌کنند. زمانبند صف این امکان را فراهم می‌کند که بتوان دستیابی کلاسهای سرویس را به یک منبع محدود شبکه یعنی پهنای باند اتصال، مدیریت و سازماندهی نمود.

۲- الگوریتمهای مدیریت حافظه صف، مقدار بسته‌های موجود در صف (عمق صف) را با مشخص نمودن اینکه چه زمان و کدامین بسته‌ها در صورت بروز ازدحام باید دور ریخته شوند کنترل می‌کنند. مدیریت حافظه صف این امکان را فراهم می‌کند که بتوان دستیابی کلاسهای سرویس را به یک منبع محدود شبکه یعنی حافظه بافر صف مدیریت نمود.

با وجود اینکه این دو مکانیزم بسیار به یکدیگر مرتبط می‌باشند، مباحثی کاملا مجزا را تحت پوشش قرار می‌دهند. زمانبندی صف این امکان را بوجود می‌آورد که بتوان ازدحام را از طریق کنترل تخصیص پهنای باند درگاه خروجی مابین کلاسهای سرویس مختلف مدیریت نمود. مدیریت حافظه صف، جلوگیری از ازدحام را با استفاده از کنترل طول متوسط صف فراهم می‌آورد.

یکی از ساده ترین مکانیزم های مدیریت صف، روش Tail Drop می باشد.Tail Drop فقدان کامل مدیریت حافظه صف می‌باشد. هنگامیکه بسته‌ای به انتهای یک صف که منابع آن کاملا مصرف شده‌اند می‌رسد، بسته به همراه همه بسته‌های بعدی، تا زمانی که فضایی در صف فراهم شود دور ریخته می‌شوند.

فواید روش Tail Drop عبارتند از:

۱- پیاده‌سازی این روش برای سازندگان و فهم آن برای کاربران ساده است.

۲- Tail Drop می‌تواند تعداد بسته‌های دور ریخته شده را کاهش دهد، بخصوص اگر به صف، حجم حافظه زیادی تخصیص داده شده باشد، البته صفوف با طول زیاد منجر به افزایش تأخیر انتها به انتها برای کلیه جریانهایی می‌شود که از این صف استفاده می‌کنند.

محدودیتهای Tail Drop عبارتند از:

۱- Tail Drop تا زمانیکه صف صددرصد پر نشده و منابع کاملا مصرف نشده باشند، بسته‌ها را دور نمی‌ریزد. در نتیجه پس از پرشدن صف، تا زمانی که فضایی در صف خالی نشود قادر به جذب مابقی ترافیک نیست. این وضع به علت فقدان فضای بافر برای ذخیره بسته‌های وارده منجر به بلاک شدن صف می‌شود. در نتیجه تعداد کمی از جریانها می‌توانند تمام ظرفیت بافر را اشغال کرده و مانع از دسترسی مابقی جریانها به صف شوند.

۲- Tail Drop باعث می‌شود که صفوف برای بازه‌های زمانی طولانی پر بمانند؛ چرا که سیستمهای میزبان، زمانی قادر به تشخیص ازدحام (از طریق دور ریختن بسته‌ها) خواهند بود که صف به صددرصد ظرفیت خود رسیده و منابع، کاملا اشغال شده باشند.

۳- Tail Drop الگوریتمی بسیار ضعیف برای ترافیک بر پایه TCP است. تقریبا حدود هشتادوپنج تا نودوپنج درصد ترافیکهای شبکه IP ، بر پایه TCP هستند. TCP فرض می‌کند که بسته به دلیل بروز ازدحام در شبکه توسط مسیریاب دور ریخته است. این امر موجب می‌شود که جریانات TCP نرخ خود را کنترل نمایند. Tail Drop باعث می‌شود که کلیه جریانات TCP که از صف مازدحام عبور می‌کنند، نرخ خود را همزمان کاهش دهند که خود باعث ایجاد پدیده‌ای به نام همزمانی سراسری[1] TCP می‌گردد. این امر موجب ایجاد نوسانات مخرب در ترافیک شده و منجر به استفاده غیرمؤثر از پهنای باند درگاه خروجی می‌گردد. چرا که چندین جریان، همزمان پنجره ازدحام TCP خود را به نصف کاهش می‌دهند.

۴- جریانهای مجزای TCP، با سرعت کمتری از عواقب ناشی از دور ریخته شدن چندین بسته نسبت به گم شدن یک تک بسته بهبود می‌یابند. این مساله می‌تواند تا حد زیادی بهره‌وری کلی جریانهای TCP را کاهش دهد.

۵- بالا بودن عمق متوسط صف باعث افزایش تأخیر انتها به انتهای جریانهایی می‌شود که از شبکه عبور می‌کنند. دور ریختن بسته‌ها باعث هدر رفتن منابعی می‌شود که برای انتقال بسته تا نقطه دور ریخته شدن آن صرف شده‌اند.

مدیریت فعال صف به معنی دستکاری صف دریک مسیریاب با هدف بالابردن کارآیی جریان هایی است که مسیریاب آنها را حمل می نماید.

اهداف اصلی مدیریت فعال صف عبارتند از:

- کاهش متوسط طول صف درمسیریاب های میانی شبکه و کاهش تاخیر انتها به انتها بسته های متعلق به جریان های ترافیکی
- استفاده بهتر از منابع شبکه وافزایش کارآیی شبکه با کاستن تعداد بسته هایی که به صورت لبریزشدن از بافرازبین می روند.

الگوریتم های متعددی برای مدیریت فعال صف ارائه شده است که درادامه به بررسی اجمالی هریک از الگوریتم های فوق می پردازیم.

### ۱۳-۲-۱- مکانیزم RED[2]

بر خلاف Tail Drop که هیچ مدیریت صفی را تأمین نمی‌نماید، RED یک تکنیک مدیریت فعال صف است که هم اکنون در شبکه‌های بزرگ IP بکار می‌رود. با استفاده از RED دور ریختن یک تک بسته کافیست تا بروز ازدحام به میزبانها اطلاع داده شود. با دور ریختن یک بسته، مسیریاب به طور غیر واضح به مبدا TCP هشدار می‌دهد که بسته دور ریخته شده جایی در طول مسیرش به مقصد TCP با ازدحام مواجه شده است. در پاسخ به این هشدار غیر واضح، مبدا TCP نرخ ارسال خود را کاهش می‌دهد تا در نهایت بافر صف مسیریاب، سرریز نشود.

---

[1] global synchronization
[2] Random Early Detection

درالگوریتم RED با محاسبه متوسط وزن دار طول صف، ازدحام احتمالی موجود درشبکه شناسایی شده و درصورت تشخیص وقوع ازدحام درشبکه ، بسته ورودی حذف ویاعلامت گذاری می شود. ویژگی اصلی الگوریتم RED این است که متوسط طول صف همواره مقدار پایین نگاه داشته می شود ودرعین حال با انفجارهای لحظه ای و ازدحام های زودگذر بسته ها برخورد نمی کند. ازدحام های زودگذر با یک افزایش موقتی طول صف بوجود می آیند. درصورت وقوع ازدحام های طولانی مدت، الگوریتم RED عکس العمل نشان داده و با علامت زدن بسته ورودی، به فرستنده خبر می دهد که ازدحام بوجودآمده است و ازآن درخواست می کند که طول پنجره ارسال خود را کاهش دهد. البته در صورت بروز ازدحام درصف، تمام اتصالات موجود، اندازه پنجره ارسال خود را کاهش نمی دهند بلکه فقط برخی از اتصالات به طور تصادفی انتخاب شده و اندازه پنجره ارسال خودرا کم میکنند.

RED از یک متوسط وزن دار طول صف (avg) برای تصمیم گیری برای حذف یا علامت زنی بسته ها استفاده می کند. هنگام ورود یک بسته، درصورتیکه متوسط وزن دار طول صف از یک حد آستانه پایین( $\min_{th}$) کمتر باشد، بسته ورودی درون صف قرار می گیرد ولی چنانچه avg از یک حد بالای آستانه ($\max_{th}$) بیشتر باشد، حتماً بسته ورودی حذف ویا علامت زده می شود. بدین ترتیب این تضمین وجود دارد که همواره متوسط طول صف از مقدار حد آستانه بالا تجاوز نخواهد کرد. درصورتیکه avg بین این دوحد آستانه باشد، معرف این است که ازدحام درحال وقوع می باشد وباید به نوعی به برخی از اتصالات موجود این موضوع اطلاع داده شود. دراین حالت بسته های ورودی با احتمال$P_a$ علامت زده می شود که $P_a$ تابعی از avg است. احتمال علامت زدن بسته های متعلق به یک اتصال خاص، بامیزان پهنای باند اشغال شده اتصال نسبت مستقیم دارد.

متوسط وزن دارصف، میزان درجه ترافیک انفجاری که اجازه ورود به صف دارد را تعیین می کند. باتوجه به سطح ازدحام فعلی درصف، مقدار احتمال علامت زنی بسته ها تعیین می شود. درالگوریتم RED هدف اصلی این است که بسته های موجود درصف کاملاً تصادفی علامت زده شوند تا از یکطرفه کردن و یا همگام سازی سراسری برای کاهش اندازه پنجره ارسال اجتناب شود و بسته ها به تناوب و به طور موثر علامت زده شوند. دراین صورت می توان طول صف را به طور بهینه کنترل نمود.

محاسبه متوسط طول صف با توجه به دوره های زمانی بی کاربودن صف و باتخمین تعداد m بسته که می توانند در طی دوره بیکاربودن صف ازمسیریاب عبورنمایند، انجام می شود.بعد از هردوره بیکاری، متوسط وزن دار طول صف محاسبه می شود.برای محاسبه احتمال علامت زنی بسته ها ($P_a$) ابتدا احتمال $P_b$ به صورت زیر بدست می آید:

$$P_b = \frac{\max p(avg - \min_{th})}{(\max_{th} - \min_{th})}$$

درفرمول فوق پارامتر maxp یک پارامتر کنترلی است که توسط آن حداکثر احتمال دورریختن بسته ها مشخص می شود. باتوجه به فرمول فوق می توان به این نتیجه رسیدکه مقداراحتمال $P_b$ بین ۰ و $maxp$ تغییر می کند.احتمال علامت زنی بسته ها، $P_a$ ، باتوجه به اینکه از آخرین علامت زنی بسته قبلی تاکنون چند بسته وارد صف شده است (count) محاسبه می شود. این احتمال به تدریج با افزایش count افزایش می یابد. مقدار احتمال $P_a$ برحسب احتمال$P_b$ ومقدار count به صورت زیرمحاسبه می گردد:

$$P_a = \frac{P_b}{(1 - count.P_b)}$$

برای اندازه گیری طول صف، می توان به جای شماره تعداد بسته ها، تعداد بایت های موجود درآن را شمارش نمود. دراینصورت می توان ازروی متوسط طول صف، مستقیماً تاخیر صف را محاسبه نمود. درصورتیکه طول صف برحسب بایت اندازه گیری شود، فرمول های فوق به صورت زیر اصلاح می شوند:

$$P_b = \frac{\max p(avg - \min_{th})}{(\max_{th} - \min_{th})}$$

$$P_b = \frac{P_b.PacketSize}{MaximumPacketSize}$$

$$P_a = \frac{P_b}{(1 - count.P_b)}$$

با توجه به فرمول های فوق میتوان نتیجه گرفت که هنگامی که طول صف برحسب بایت اندازه گیری می شود، احتمال علامت زنی یک بسته بزرگ ( مثل بسته FTP) ازیک بسته کوچک (مثل یک بسته TELNET) به مراتب بیشتر است. در شکل(۱۳-۱)، شبه کد الگوریتم RED و نمودار احتمال دور ریختن ( $P_b$ ) در این الگوریتم نشان داده شده است.

فواید الگوریتم RED عبارتند از:

۱- RED نیازی به تغییر در پروتکل لایه انتقال میزبانهای فعلی شبکه ندارد.

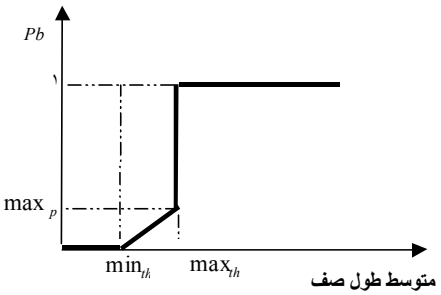
۲- RED اولین مراحل ازدحام را تشخیص داده و با دور ریختن تصادفی بسته‌ها، به آن عکس‌العمل نشان می‌دهد. اگر حجم ازدحام رو به افزایش باشد، RED بسته‌ها را با شدت بیشتری دور می‌ریزد تا از پر شدن کامل صف جلوگیری کند.

۳- از آنجا که RED برای شروع دور ریختن بسته‌ها تا پر شدن کامل صف منتظر نمی‌ماند، به صف این امکان را می‌دهد که انفجارهای ترافیک را پذیرفته و کلیه بسته‌ها را در هنگام بروز انفجار دور نریزد. در نتیجه، رفتارRED با جریانهای TCP دوستانه است؛ چرا که در RED، دسته‌ای از بسته‌ها که همگی متعلق به یک جریان TCP است دور ریخته نمی‌شوند. در نتیجه RED از همزمانی سراسری TCP جلوگیری می‌کند.

۴- RED باعث می‌شود که حجم ترافیک صف در سطحی متوسط باقی بماند. حجم ترافیک صف نه آنقدر پایین است که موجب عدم بهره‌وری پهنای باند شود و نه آنقدر بالا است که دور ریختن بالای بسته‌ها منجر به این شود که تعداد زیادی از اتصال های TCP نرخ خود را همزمان کاهش دهند و میزان بهره‌وری پهنای باند پایین بیاید.

For each packet arrival
Calculate the average queue size (*avg*)
If $\min_{th} \le avg \le \max_{th}$
Calculate probability $p_b$
With probability $p_b$ :
Mark the arriving packet
Else if $\max_{th} \le avg$
Mark the arriving packet

(ب)




الف)

**شکل(۱۳-۱): (الف) احتمال دور ریختن بسته‌ها در RED، (ب) شبه کد الگوریتم RED**

محدودیتهای RED عبارتند از:

۱- بهره‌وری RED بسیار حساس به انتخاب پارامترهای آن است. انتخاب نادرست پارامترهای RED ممکن است به بهره‌وری حتی پایین‌تر از Tail Drop نیز منجر شود. همچنین متوسط طول صف RED بسته به سطح ازدحامی که با آن مواجه می‌شود تغییر می‌کند.

۲- پیکربندی RED برای دستیابی به یک کارایی قابل پیش‌بینی، کار پیچیده و مشکلی است.

پارامتر وزن صف( $w_q$) برحسب اندازه و تعداد دوره های انفجار که اجازه ورود به صف را دارد محاسبه می شود. پارامترهای $max_{th}$ و $min_{th}$ برحسب متوسط اندازه صف مورد نظرانتخاب میشوند. با تنظیم متوسط طول صف، می توان بین گذردهی و تاخیر صف مصالحه نمود.

با اعمال تغییرات کوچک و جابجا کردن دستورات آن، می توان الگوریتم RED را به طور موثرپیاده سازی نمود. علاوه برآن لزومی ندارد که همه دستورات الگوریتم RED با رسیدن یک بسته اجرا گردند.بیشتر کارهای الگوریتم فوق، از قبیل محاسبه متوسط طول صف و احتمال علامت زنی بسته ، را می توان به موازات هدایت پیشرو بسته ها انجام داد. بنابراین میتوان الگوریتم RED را درمسیریاب هایی با خطوط خروجی پرظرفیت به خوبی پیاده سازی نمود. چنانچه به جای علامت زدن سرآیند بسته ها ازحذف صریح بسته ها درالگوریتم RED استفاده شود، بارعملیاتی الگوریتم به شدت کاهش می یابد.

مقادیر پیشنهادی برای پارامترهای RED به شرح زیر است:

$$\min_{th} = 5 \quad packets$$
$$\max_{th} = 2 \ or \ 3 \ times \ \min_{th}$$
$$\max p = 0.1;$$
$$w_q = 0.002$$

هنگامی که متوسط طول صف دراواسط بین $max_{th}$ و $min_{th}$ است، بهتر است که این احتمال به آرامی با نزدیک شدن مقدار متوسط طول صف به هریک از دو آستانه $max_{th}$ و $min_{th}$ تغییر یابد. مثلاً با نزدیک شدن متوسط طول صف به $max_{th}$ این احتمال کمی افزایش یابد . البته نباید مقدار maxp از $0.1$ فراتر رود. هنگامی که maxp برابر با $0.1$ است، با نزدیک شدن متوسط طول صف به $max_{th}$ تقریباً ۲۰ درصد بسته ها علامت زده می شوند.

### ۱۳-۲-۲- مکانیزم FRED[1]

این روش سعی دارد تمامی جریانهایی که از اتصال می‌گذرند سهم عادلانه خود را از ظرفیت اتصال دریافت کنند. برای تحقق این هدف، این الگوریتم صف را بر اساس هر جریان مدیریت می‌کند. FRED برای هر جریان که در صف مسیریاب بسته‌ای دارد، آماری نگهداری می‌کند. در طول بازه‌های زمانی ازدحام (که با بررسی طول صف مشخص می‌شود)، الگوریتم، بسته‌های مربوط به جریانهایی را دور می‌ریزد که بیشتر از دو برابر سهم عادلانه[2] خود از صف اشغال کرده باشند. سهم عادلانه از صف عبارتست از: $1/n^{th}$ متوسط طول صف که $n$ عبارتست از تعداد جریانهایی که در صف بسته دارند. محدودیت، دو برابر سهم عادلانه است تا بتوان بسته‌های ایجاد شده در اثر ترافیک انفجاری را در صف جا داد. اگر جریانی از این محدودیت سهم عادلانه، برای یکبار تجاوز کند، آنگاه برای دفعات بعد بصورت کامل و دقیق محدود به سهم عادلانه خود شده و حق اشغال بیش از سهم خود را نخواهد داشت. این محدودیت تنها زمانی برداشته خواهد شد که جریان هیچ بسته‌ای در صف نداشته باشد. از آنجا که جریانهای غیر مسئول، عموما بار ترافیکی پیوسته دارند و معمولا همواره حداقل یک بسته در صف دارند، این محدودیت بطور مداوم برای آنها باقی می‌ماند.

---

[1] Flow RED

[2] fair share

برعکس اگر یک جریان مسئول از سهم عادلانه خود تجاوز کند، باید به بسته‌های دور ریخته‌ای که در اثر این تخلف ایجاد می‌شود با کاهش بار خود عکس‌العمل نشان دهد. که این واکنش منجر به این می‌شود که در یک تناوب، این جریان هیچ بسته‌ای در صف نداشته باشد و در نتیجه محدودیت قطعی و محکم برداشته می‌شود. این مکانیزم باید بصورت محکم جریانهای غیر مسئول را محدود نماید و در همان حال به جریانهای مسئول امکان دهد که سهم بیشتری از ظرفیت اتصال را دریافت کنند.

### ۱۳-۲-۳- مکانیزم WRED[1]

WRED توسعه‌ای از RED است که این امکان را فراهم می‌کند تا بتوان به انواع مختلف ترافیک، سطوح مختلف دور ریختن را نسبت داد. توانایی تعریف سطوح مختلف دور ریختن برای صف های مختلف یا انواع مختلف ترافیک در یک صف، کنترل بهتری از RED را فراهم می‌سازد.

به عنوان مثال فرض کنید دو سطح دور ریختن برای یک صف تعریف شده باشد. در این حالت می‌توان به یک دسته از بسته‌ها که دارای اولویت بالاتر هستند، سطح دور ریختن پایین‌تر و به دسته دیگر از بسته‌ها که دارای اولویت پایین‌تر هستند، سطح دور ریختن بالاتری را اعمال کرد.

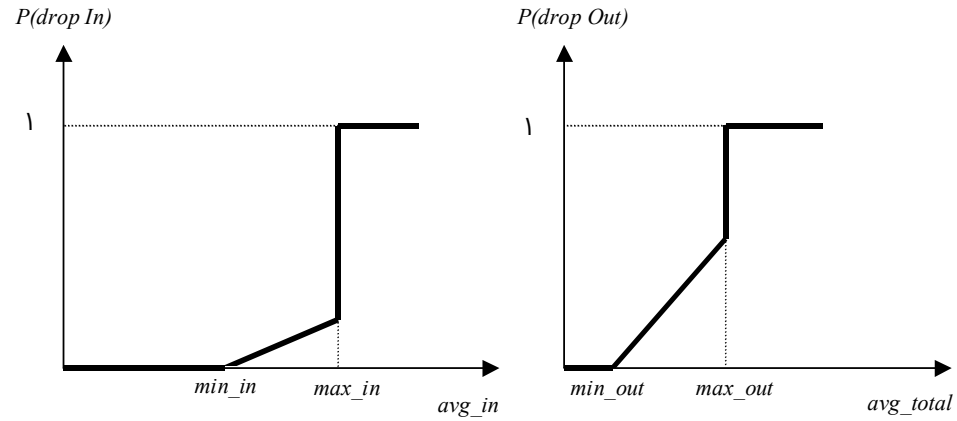
### ۱۳-۲-۴- مکانیزم RIO[2]

در این روش بسته‌ها به دو کلاس $In$ و $Out$ تقسیم می‌شوند که اولی بیان کننده بسته‌هایی است که در داخل پروفایل ترافیکی هستند و دومی، مشخصه بسته‌هایی است که در خارج پروفایل قرار گرفته و از قرارداد ترافیکی تخلف نموده‌اند. مکانیزم بکار رفته در این الگوریتم همانند RED است. اما این الگوریتم با دو مجموعه پارامتر پیکربندی می‌شود. یک مجموعه برای بسته‌های $In$ و دیگری برای بسته‌های $Out$. به ازای ورود هر بسته، دروازه ورودی شبکه چک می‌کند که آیا بسته $In$ است یا $Out$ ؟ اگر بسته $In$ باشد، دروازه، $avg\_in$ یعنی متوسط طول صف بسته‌های $In$ و همچنین $avg\_total$ که متوسط طول کلی صف برای کلیه بسته‌ها (هم بسته‌های $In$ و هم $Out$) است محاسبه می‌کند. احتمال دور ریختن بسته‌های $In$ بستگی به $avg\_in$ و احتمال دور ریختن بسته‌های $Out$ به $avg\_total$ بستگی دارد.

سه پارامتر برای هر یک از کلاسها وجود دارند. پارامترهای $min\_in$ ، $max\_in$ و $P_{\max\_in}$ مشخص کننده فازهای : عملکرد نرمال $[0,min\_in]$ ، جلوگیری از ازدحام $[min\_in,max\_in]$ و کنترل ازدحام $[max\_in, \infty]$ برای بسته‌های $In$ می‌باشند. بطور مشابه، $min\_out$ و $max\_out$ و $P_{\max\_out}$ مشخص کننده فازهای مربوط به بسته‌های $Out$ می‌باشند.

تفاوت رفتار نسبت به بسته‌های $Out$ با انتخاب دقیق پارامترهای ذکر شده ممکن می‌گردد. همانگونه که در شکل(۱۳-۲) نشان داده شده است، دروازه RIO ، در سه حالت، بسته‌های $Out$ را با شدت بیشتری کنترل می‌کند : اول اینکه بسته‌های $Out$ را زودتر از بسته‌های $In$ دور می‌ریزد. این امر با انتخاب $min\_out$ کوچکتر از $min\_in$ صورت می‌گیرد. دوم در فاز جلوگیری از ازدحام، بسته‌های $Out$ را با احتمال بالاتری دور می‌ریزد. چرا که $P_{\max\_out}$ بالاتر از $P_{\max\_in}$ انتخاب شده است. سوم RIO برای بسته‌های $Out$ زودتر وارد فاز کنترل ازدحام می‌شود تا برای بسته‌های $In$ (چرا که $max\_out$ بسیار کوچکتر از $max\_in$ انتخاب شده است). در کل RIO در صورت بروز احتمال ازدحام بسته‌های $Out$ را ابتدا دور می‌ریزد و در صورتیکه ازدحام باقی بماند، کلیه بسته‌های $Out$ دور ریخته می‌شوند.

[1] Weighted RED
[2] RED in Input and Output




**شکل(۱۳-۲): نمودار احتمال دور ریختن در RIO**

### ۱۳-۲-۵- مکانیزم ARED[1]

همانگونه که گفته شد یکی از معایب الگوریتم RED این است که متوسط طول صف آن با توجه به سطح ازدحام تغییر می‌کند و منجر به پایین آمدن بهره‌وری اتصال می‌گردد. ARED تلاش دارد تا با تنظیم پارامتر $\max_p$ بر اساس بار ترافیکی شبکه، متوسط طول صف را همواره بین $\min_{th}$ و $\max_{th}$ نگهدارد. در این روش بر اساس شدت بار ترافیکی موجود، اگر متوسط طول صف بیشتر از $\max_{th}$ باشد، $\max_p$ بصورت ضربی با فاکتور $\alpha$ افزایش و در صورتیکه متوسط طول صف از $\min_{th}$ کمتر شود، با فاکتور $\beta$ کاهش می‌یابد.

در نسخه جدیدتری که از این روش ، سعی بر این شده $\max_p$ طوری تنظیم شود که متوسط طول صف را نه تنها بین $\min_{th}$ و $\max_{th}$، بلکه در یک دامنه هدف و در نیمه فاصله بین این دو حد آستانه نگهدارد. تغییر دیگری که در این الگوریتم لحاظ شده است، این است که بجای افزایش و کاهش ضربی، از روش افزایش جمعی و کاهش ضربی[2] (AIMD) استفاده شده است. در این توسعه، $\alpha < 0.25$ و $\beta > 0.83$ (پیش‌فرض $0.9$) در نظر گرفته شده است.

### ۱۳-۲-۶- مکانیزم SRED[3]

در این روش سعی شده است که طول صف در سطحی مستقل از تعداد اتصالات فعال تثبیت شود. این امر در SRED با تخمین تعداد جریانات فعال صورت می‌گیرد. این تخمین بدون جمع‌آوری و تحلیل اطلاعات آماری مربوط به جریانها انجام می‌شود. با استفاده از همین تخمین می‌توان جریانهای متخلف را نیز شناسایی نمود.

در این روش لیستی به نام zombie وجود دارد که حاوی اطلاعات آخرین M بسته وارد شده می‌باشد. با ورود هر بسته، اطلاعات شناسایی بسته (آدرس مبدا، آدرس مقصد و ...) به این لیست اضافه می‌شود. این کار تا پر شدن کامل لیست ادامه می‌یابد. پس از پر شدن لیست، با ورود هر بسته اطلاعات آن بسته با اطلاعات یکی از بسته‌هایی که بطور تصادفی از لیست انتخاب شده مقایسه می‌گردد. اگر اطلاعات بسته ورودی بر اطلاعات بسته انتخاب شده منطبق بود، به این معناست که این دو بسته متعلق به یک جریان هستند و در نتیجه متغیری به نام hit، ۱ می‌گردد. اگر دو بسته متعلق به جریان یکسانی نبودند hit برابر صفر می‌گردد. در این حالت با احتمال $P$ اطلاعات جریان بسته بر روی اطلاعاتی که بصورت تصادفی از لیست انتخاب شده بود نوشته می‌شود و با احتمال $1\text{-}P$ تغییری در لیست بوجود نمی‌آید.

---

[1] Adaptive RED
[2] Additive Increase Multiplicative Decrease (AIMD)
[3] Stablized RED

بدون توجه به اینکه آیا hit رخ داده است یا نه، اگر سیستم در حالت دور ریختن تصادفی باشد، بسته ورودی با احتمال مشخصی دور ریخته می‌شود که این احتمال، بستگی به وجود یا عدم وجود hit دارد.
در SRED می‌توان تعداد جریانهای فعال را از روی نرخ متوسط hit تخمین زد.با فرض:

$$Hit(i) = \begin{cases} 0 & if\ no\ hit \\ 1 & if\ hit \end{cases}$$

فرکانس hit عبارتست از:

$$p(t) = (1-\alpha)P(t-1) + \alpha Hit(t) \qquad \text{که در آن } 1 < \alpha < 1$$

$P(t)^{-1}$ تخمینی از تعداد جریانهای فعال موجود پیش از ورود بسته t ام می‌باشد.
احتمال دور ریختن در SRED عبارتست از:

$$P_{zap} = P_{sred}(q) \times \min(1, \frac{1}{(256 \times P(t))^2})$$

که در آن:

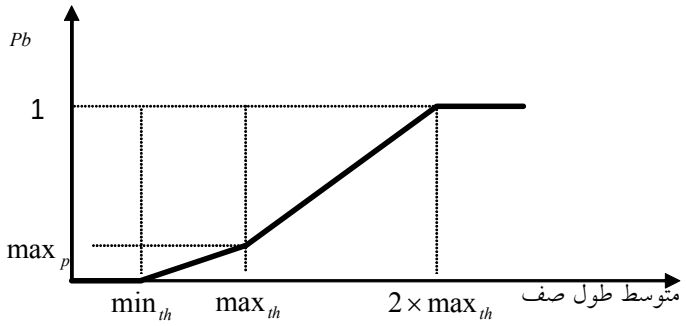
$$P_{sred}(q) = \begin{cases} P_{\max} & if\ B/3 \le q \le B \\ 0.25 \times P_{\max} & if\ B/6 \le q \le 3/B \\ 0 & otherwise \end{cases}$$

بعلاوه مکانیزم hit می‌تواند در تشخیص جریانهای متخلف نیز مفید باشد. جریانهای متخلف معمولا تعداد بیشتری hit دارند. چرا که بسته بیشتری ارسال می‌کنند. (و در نتیجه تعداد مقایسه‌های بیشتری را موجب می‌شوند) و همچنین ورودی‌های بیشتری به لیست zombie دارند.

### ۱۳-۲-۷- مکانیزم GRED[1]

همانگونه که بیان شد، در الگوریتم RED اگر متوسط طول صف از $\max_{th}$ تجاوز کند تمامی بسته‌ها دور ریخته می‌شوند. این حالت ممکن است منجر به رفتار نوسانی RED شود. در الگوریتم GRED ، منحنی احتمال دور ریختن( $P_b$ ) تغییر یافته است. به این صورت که برای طول صف بین $\max_{th}$ و ۲× $\max_{th}$ نیز بسته‌ها با احتمال $P_b$ دور ریخته می‌شوند . این الگوریتم، نسبت به نوسانات ناگهانی طول صف و تنظیم پارامترها در مقایسه با RED، مستحکمتر است. شکل (۱۳-۳)، احتمال دور ریختن بسته‌ها را در این الگوریتم نشان می‌دهد.

[1] Gentle RED




**شکل(۱۳-۳): احتمال دورریختن بسته‌ها در GRED**

### ۱۳-۲-۸- مکانیزم ECN [1]

ECN بعنوان توسعه‌ای آزمایشی بر IP ارائه گردید، شیوه‌ای متفاوت برای مدیریت فعال صف تأمین می‌کند. بر خلاف RED که با دور ریختن بسته‌ها به ازدحام واکنش نشان می‌دهد، ECN با علامت زدن بسته‌ها باعث می‌شود که میزبان مقصد، یک اخطار واضح به معنی وجود ازدحام به مبدا بفرستد. وقتی مبدا، یک ECN دریافت نمود، نرخ ارسال خود را کاهش می‌دهد.

### ۱۳-۲-۸-۱- توسعه ECN در IP

ECN نیاز به تعریف یک فیلد دو بیتی به نام ECN در سرآیند IP دارد.مطابق شکل(۱۳-۴)، بیتهای شش و هفت فیلد DS بصورت آزمایشی به فیلد ECN در IPv4 و IPv6 تخصیص یافته‌اند. بیت شش، ECT [2] و بیت هفت، CE [3] نامیده می‌شود. بیت انتقال با قابلیت ECN، (ECT)،توسط منبع یک می‌شود و نشان دهنده این مطلب است که نقاط پایانی پروتکل لایه انتقال، دارای قابلیت ECN می‌باشند. وقتی که مسیریاب با ازدحام مواجه می‌شود، به منظور مشخص نمودن وقوع ازدحام برای سیستمهای میزبان، بیت مواجهه با ازدحام (CE) را یک می‌کند. بیت CE فقط وقتی باید توسط مسیریاب یک شود که در حالت معمول (بدون پشتیبانی از ECN)، به عنوان نشان ازدحام، بسته را دور می‌ریخت.
به طور خاص، زمانی که مقصد بسته‌ای با شناسه ECN (بیت CE مساوی یک) دریافت می‌کند، الگوریتم کنترل ازدحامی که دنبال می‌کند باید دقیقا همان الگوریتمی باشد که در پاسخ به یک بسته دور ریخته شده اجرا می‌شود. در نتیجه زمانی که منبع، بسته‌ای با شناسه ECN دریافت می‌کند باید با نصف کردن پنجره ازدحام TCP مربوط به هر پنجره داده‌ای که حاوی بسته با شناسه ECN است به ازدحام پاسخ گوید.


**شکل(۱۳-۴): ECN در فیلد DS یک سرآیند IPv4 یا IPv6**

[1] Explicit Congestion Notification
[2] Ecn Capable Transmit (ECT)
[3] Congestion Experienced (CE)

### ۱۳-۲-۸-۲- توسعه ECN در TCP

علاوه بر فیلد ECN در سرآیند بسته IP ، باید تغییراتی در پروتکل لایه حمل نیز به منظور پشتیبانی ECN اعمال نمود. بکارگیری ECN در TCP نیازمند سه تغییر است:

۱- در طول روال برقراری ارتباط TCP، سیستمهای میزبان نیاز به مکانیزمی دارند تا تعیین کنند آیا هر دو دارای قابلیت ECN هستند یا خیر.

۲- تعریف فیلد ECN-echo در سرآیند TCP که از طریق آن، گیرنده TCP پس از دریافت یک بسته CE به مبدا اطلاع می‌دهد که یک بسته CE دریافت نموده است.

۳- تعریف فیلد" کاهش پنجره ازدحام (CWR) [۱]" در سرآیند TCP تا از طریق آن، فرستنده بیت CE بتواند به مقصد اطلاع دهد که پنجره ازدحام TCP کاهش یافته است.

### ۱۳-۲-۸-۳- مزایا و محدودیتهای ECN

علاوه بر مزایایی که ECN مانند سایر الگوریتمهای مدیریت فعال دارا می‌باشد، این مکانیزم به علت علامت زدن بسته‌ها بجای دور ریختن آنها، مانع از هدررفتن پهنای باندی که برای انتقال بسته تا نقطه ازدحام بکار رفته می‌گردد.

ECN زمان انتقال جریانهای کوتاه TCP را در حالتی که آخرین بسته آنها دور ریخته شده کاهش می‌دهد؛ چرا که با دارا بودن قابلیت ECN، این جریانها دیگر نیاز به انتظار تا منقضی شدن زمان‌سنج ارسال مجدد و ارسال مجدد آخرین بسته خود نخواهد داشت.

محدودیت ECN این است که پیاده سازی آن، نیاز به تغییر در پروتکلهای فعلی لایه حمل دارد تا بتواند آنها را دارای قابلیت ECN نماید.

### ۱۳-۲-۹- مکانیزم BLUE

عملکرد مطلوب RED نیاز به عملکرد صحیح دامنه وسیعی از پارامترها تحت سناریوهای ازدحام مختلف دارد. RED زمانی می‌تواند عملکرد مطلوبی داشته باشد که اولا: میزان فضای بافر کافی در اختیار داشته باشد وثانیا پارامترها به نحو صحیحی مقداردهی شده باشد. فضای بافر بزرگ از این جهت برای RED ضروری است که بتواند بار بالاتر از ظرفیت اتصال را از زمان تشخیص ازدحام از طریق پارامتر طول صف تا زمانی که بار مربوط، در پاسخ به هشدار ازدحام توسط مبدا کاهش می‌یابد، در خود جای دهد. همچنین RED باید اطمینان داشته باشد که علامت ازدحام با نرخی ارسال شود که موجب کاهش نرخ منبع شود بدون اینکه منجر به عدم بهره‌وری اتصال گردد. متاسفانه، زمانی که تعداد زیادی منبع TCP فعال هستند، ترافیک حاصله به شدت انفجاری است. ترافیک انفجاری عموما تکنیکهای مدیریت صف مانند RED را بی‌اثر می‌کند. چرا که طول صف با سرعت بسیار بالاتر از آنکه RED بتواند به آن عکس‌العمل نشان دهد بالا و پایین می‌رود. یک روش حل این معضل، بکارگیری بافر بزرگ برای دروازه‌های RED است. بعنوان مثال پیشنهاد شده است که به منظور عملکرد مطلوب RED، مسیریابهای میانی فضای بافری معادل دو برابرحاصلضرب پهنای باند و تأخیر داشته باشند. این شیوه توسط بسیاری از ISP ها انتخاب شده است. اما متاسفانه در شبکه‌های بزرگ، استفاده از طول صف بزرگ موجب بالا رفتن تأخیر و واریانس تأخیر خواهد شد.

به منظور رفع نواقص RED، الگوریتمی متفاوت به نام BLUE در سال ۱۹۹۹ ارائه گردید. ایده کلیدی BLUE اجرای مدیریت صف مستقیما بر پایه نرخ از دست دادن بسته و بهره‌وری اتصال است. BLUE ، یک احتمال دور ریختن $P_m$ را نگهداری می‌کند که برای علامت زدن یا دور ریختن بسته‌های وارد شده به صف بکار می‌رود. اگر صف مرتبا به علت سرریز بافر بسته‌ها را دور بریزد، $P_m$ افزایش می‌یابد و در نتیجه نرخ ارسال به عقب علامت ازدحام، افزایش می‌یابد. برعکس اگر صف

[۱] Congestion Window Reduced (CWR)

خالی شود یا اتصال بیکار گردد، BLUE احتمال $P_m$ را کاهش می‌دهد. این عمل بطور مؤثری به BLUE امکان می‌دهد که نرخ صحیحی را که برای ارسال به عقب علائم ازدحام نیاز دارد مشخص کند. شکل(۱۳-۵) شبه کد الگوریتم BLUE را نشان می‌دهد.

Upon packet loss or $(Q_{len} > L)$ event:
If ( now-*last_update* ) > *freeze_time* )
$P_m = P_m + \delta_1$
*last_update* = now
Upon link idle event:
If ( now-*last_update* ) > *freeze_time* )
$P_m = P_m - \delta_2$
*last_update* = now

**شکل (۱۳-۵): شبه کد الگوریتم BLUE**

تفاوت دیگری که در این الگوریتم وجود دارد، نشاندهنده حالتی دیگر برای به‌روز کردن احتمال دور ریختن است و آن وقتی است که طول صف از مقدار مشخص $L$ تجاوز کند. این تغییر باعث می‌شود که فضای کافی برای ترافیک انفجاری در صف باقی بماند.

علاوه بر احتمال دور ریختن، BLUE دو پارامتر دیگر نیز بکار می‌برد که میزان سرعت تغییر احتمال را در طول زمان مشخص می‌کند. اولین پارامتر *freeze_time* است. این پارامتر، حداقل بازه زمانی مابین دو به‌روز رسانی پشت‌سر هم $P_m$ را نشان می‌دهد. این امر امکان می‌دهد که تغییرات ایجاد شده در احتمال پیش از به‌روز شدن دوباره آن، تاثیر خود را اعمال کند. دو پارامتر دیگری که انتخاب شده‌اند $\delta_1$ و $\delta_2$ می‌باشند که بترتیب، مشخص کننده دو مقداری هستند که $P_m$ در صورت سرریز شدن بافر، به اندازه آن مقدار افزایش می‌یابد و یا در صورت بیکار بودن اتصال، به اندازه آن مقدار کاهش می‌یابد. به منظور پیکربندی مناسب BLUE برای دستیابی به اهداف کنترل ازدحام، باید پارامترهای آن را به نحو مناسبی انتخاب نمود. اولین پارامتر، *freeze_time* باید براساس RTT های مؤثر اتصال های موجود تنظیم شود، تا بتوان تاثیر تغییرات ایجاد شده در $P_m$ را پیش از تغییرات بعدی آن، در منابع ترافیک اعمال نمود. دسته دوم پارامترها، $\delta_1$ و $\delta_2$ باید طوری انتخاب شوند که به اتصال، توانایی انطباق با تغییرات بزرگ در میزان بار گذرنده از آن را بدهد.

برای اتصالاتی که تغییرات بسیار بزرگ در بار، در طول فقط چند دقیقه صورت می‌گیرند، $\delta_1$ و $\delta_2$ باید طوری انتخاب شوند که امکان تغییر $P_m$ در دامنه صفر تا یک را در مدت چند دقیقه فراهم آورند. این برخلاف شیوه‌های فعلی است که احتمالات دور ریختن در آنها در طول چند میلی‌ثانیه، حتی وقتی بار ثابت است از صفر تا یک تغییر می‌کند. در مورد اتصالات معمولی، انتخاب مقادیر بین ۱۰ ms و ۵۰۰ms برای *freeze_time* و تنظیم $\delta_1$ و $\delta_2$ به نحوی که به $P_m$ ، امکان تغییر در [۰و۱] را در طول ۵ تا ۳۰ ثانیه بدهند، منجر به کارایی مؤثر BLUE خواهد شد.

به منظور بررسی کارایی BLUE، آزمایشات مختلفی تحت توپولوژی‌های مختلف و با پیکربندی‌های گوناگون صورت گرفته است. یکی از نتایج مهم این آزمایشات، این است که می‌توان BLUE را با حداقل میزان فضای بافر بکار برد و در عین حال، کارایی مطلوب را نیز بدست آورد. بکارگیری فضای بافر کم، تأخیر موجود در شبکه را کاهش داده و در عوض میزان مؤثر بودن الگوریتم کنترل ازدحام BLUE را افزایش می‌بخشد. بعلاوه نیاز به بافر کوچکتر تخصیص حافظه بیشتر به بسته‌های با اولویت بالا را نیز ممکن می‌سازد و حافظه مناسبی برای دیگر عملیات مسیریاب از جمله ذخیره جداول بزرگ مسیریابی آزاد می‌نماید.

از دیگر نتایج بررسی‌های صورت گرفته پایین بودن نرخ از دست دادن سلول و بالاتر بودن میزان بهره‌وری اتصال در BLUE در مقایسه با RED (تحت سناریوها و پیکربندی‌های مختلف) می‌باشد.

براساس شبیه سازی های انجام شده مکانیزم BLUE دارای مزایای زیر می باشد:

- نیاز به بافرکم
- توانایی اجرای کاربردهای محاوره ای دراینترنت
- تخصیص حافظه بیشتر به بسته های با اولویت بالا
- تاخیر کم

### ۱۳-۲-۱۰- مکانیزم SFB[1]

تا سالهای اخیر اینترنت بیشتر بر روی مکانیزمهای کنترلی TCP به منظور محدود کردن از دست دادن سلول و تقسیم عادلانه منابع شبکه تکیه داشت. اما اخیرا کاربردهایی به اینترنت وارد شده‌اند که از مکانیزمهای کنترلیTCP استفاده نمی‌کنند و به سیگنالهای ازدحام که از طریق شبکه ارسال می‌شود بی‌توجه هستند. این کاربردها ذاتا خطرناک هستند؛ چرا که باعث بالا رفتن نرخ از دست دادن سلول در شبکه می‌شوند. به منظور رفع مشکل جریانهای غیر مسئول، کارهای زیادی صورت گرفته که مسیریابها را در مقابل جریانهای غیرمسئول حفظ کنند. ایده اصلی این روشها تشخیص جریانهای غیرمسئول و محدود کردن نرخ آنهاست تا از تاثیر آنها بر روی کارایی جریانهای مسئول جلوگیری شود. SFB یکی از تکنیکهای جدید است که برای حفظ جریانهای TCP از جریانهای غیرمسئول طراحی شده است.

SFB یک الگوریتم صف‌بندی FIFO است که جریانهای غیرمسئول را شناسایی و محدود می‌کند. این عمل از طریق مکانیزمهای محاسبه، مشابه آنچه در BLUE بکار می‌رود صورت می‌گیرد. SFB به تعداد N*L عدد محفظه[2] در نظر می‌گیرد. این محفظه‌ها در L سطح با N محفظه در هر سطح سازماندهی می‌شوند. به علاوه SFB، L تابع hash متفاوت نگهداری می‌کند که هر یک مربوط به یک سطح از محفظه‌ها است. هر تابع hash، جریانی را از طریق شناسه ارتباط آن (آدرس مبدا، آدرس مقصد، درگاه مبدا، درگاه مقصد و پروتکل) به یکی از N محفظه موجود در آن سطح نگاشت می‌کند. محفظه‌ها به منظور دنبال کردن وضعیت اشغال بودن صف بسته‌های مربوط به یک جریان خاص بکار می‌روند. هر محفظه در SFB، یک احتمال دور ریختن $P_m$ مانند BLUE نگهداری می‌کند که بر اساس میزان پر بودن محفظه به‌روز می‌شود.

همزمان با ورود هر بسته به صف، بسته به یکی از N محفظه موجود در هر یک از L سطح، hash می‌شود. اگر تعداد بسته‌های نگاشت شده به یکی از محفظه‌ها، از یک حد آستانه مشخص (یعنی اندازه محفظه) فراتر رود، $P_m$ مربوط به آن محفظه افزایش می‌یابد. اگر تعداد بسته‌ها صفر شود، $P_m$ کاهش می‌یابد. واقعیت این است که یک جریان غیرمسئول به سرعت، $P_m$ را در کلیه L محفظه‌ای که به آنها نگاشت شده، به یک می‌رساند.

جریانهای مسئول ممکن است در یک یا دو محفظه با جریانهای غیرمسئول مشترک باشند. اما برای هر جریان مسئول حداقل یک محفظه وجود دارد که توسط جریان غیرمسئول اشغال نشده و در نتیجه دارای $P_m$ نرمال می‌باشد. تصمیم گیری در مورد دور ریختن بسته‌ها بر اساس $P_{\min}$، مینیمم مقدار $P_m$ مربوط به تمام محفظه‌هایی است که جریان به آن محفظه‌ها، نگاشت شده است. اگر $P_{\min}$ یک شود، بسته بعنوان غیرمسئول شناخته شده و نرخ آن محدود می‌شود.

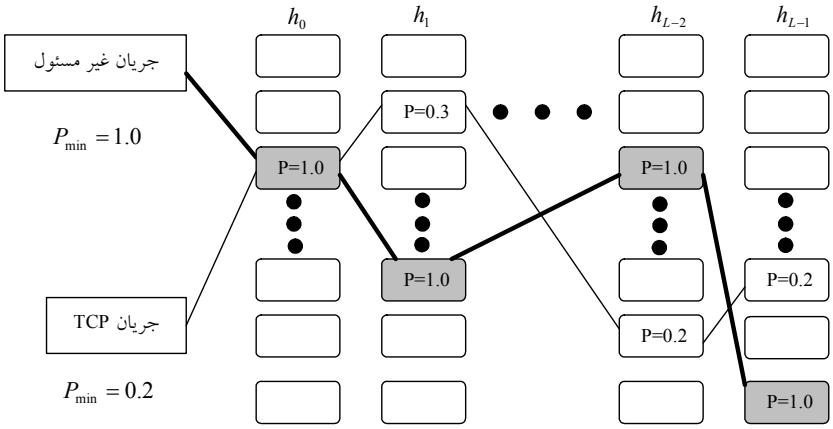
شکل(۱۳-۶)، مثالی از نحوه عملکرد SFB نشان می‌دهد. همانگونه که در شکل مشخص است، جریانی غیر مسئول، احتمالات دور ریختن کلیه محفظه‌هایی که به آنها نگاشت شده است، بالا برده است. جریان TCP نشان داده شده در شکل، اگرچه در یک سطح به محفظه‌ای مشترک با جریان غیرمسئول نگاشت شده است، اما در سطوح دیگر، به محفظه‌هایی نرمال نگاشت شده است. به همین دلیل، احتمال دور ریختن مینیمم برای جریان TCP، کمتر از یک است و در نتیجه به عنوان

---

[1] Stochastic Fair Blue

[2] bin

غیرمسئول شناخته نمی‌شود. از سوی دیگر از آنجا که مینیمم احتمالات دور ریختن جریانهای غیرمسئول، یک است، بعنوان غیرمسئول شناخته شده و نرخ آن محدود می‌شود.




**شکل(۱۳-۶): مثالی از SFB**

با وجود اینکه SFB به‌خوبی قادر است جریانهای TCP را از جریانهای غیرمسئول جدا نماید اما دارای محدودیتهایی است. از جمله اینکه با افزایش جریانهای غیرمسئول، تعداد محفظه‌هایی که توسط آنها پر شده و دارای $P_m$ برابر یک است افزایش می‌یابد. در نتیجه احتمال اینکه یک جریان مسئول به درون محفظه‌هایی نگاشت شود که همگی توسط جریانهای غیرمسئول اشغال شده باشند و در نتیجه اشتباها بعنوان غیرمسئول شناسایی شود بالا می‌رود. به عبارت دقیق‌تر احتمال اینکه یک جریان TCP بعنوان جریانی غیرمسئول شناسایی شود عبارتست از: $P = \left[1 - \left(1 - \frac{1}{B}\right)^M\right]^L$

که L تعداد سطوح، B تعداد محفظه‌های در هر سطح و M تعداد جریانهای غیرمسئول می‌باشد.

برای رفع این محدودیت SFB ، از توابع hash متغیر استفاده می‌شود. به این ترتیب که بصورت دوره‌ای و تصادفی، محفظه‌ها reset شده و توابع hash تغییر می‌یابند. جریان غیرمسئول (بدون توجه به تابع hash) مرتبا شناسایی و نرخ آن محدود خواهد شد. اما با تغییر تابع hash ، جریانهای مسئولی که بطور کامل و تصادفی به درون محفظه‌های اشغال شده توسط جریانهای غیرمسئول نگاشت شده‌اند مجددا به حداقل یک محفظه خالی نگاشت می‌شود. با وجود اینکه بکارگیری توابع hash متغیر، عدالت را بین جریانهای موجود افزایش می‌دهند، ذکر این مطلب لازم است که هر زمان که تابع hash تغییر می‌نماید و محفظه‌ها ری ست می‌شوند، جریانهای غیرمسئول به طور موقت دارای آزادی می‌شوند. به این مفهوم که برای مدت کوتاهی در حالت شک قرار می‌گیرند و محدودیت نرخ از آنها برداشته می‌شود. تنها پس از آنکه این جریانها موجب دور ریختن سلول شوند، مجدد شناسایی و نرخ آنها محدود می‌شود. این امر موجب می‌شود که جریانهای غیرمسئول در همان بازه کوتاه، سهم بیشتری از پهنای باند را اشغال کنند. یک راه حل برای این مشکل، استفاده از دو مجموعه محفظه است. همزمان با بکارگیری یکی از دو مجموعه محفظه برای مدیریت صف، مجموعه دیگر با استفاده از سری دیگر توابع hash، گرم می‌شود. در این حالت هر زمان که جریانی بعنوان غیرمسئول شناسایی می‌شود، با استفاده از مجموعه دوم توابع hash، مجدد hash شده و احتمالات دور ریختن محفظه‌های مربوطه در مجموعه دوم به‌روز می‌شوند. وقتی توابع hash سوئیچ می‌شوند محفظه‌هایی که گرم شده‌اند مورد استفاده قرار می‌گیرند. در نتیجه جریانهای غیرمسئول درست از همان ابتدا محدودیت نرخ پیدا می‌کنند.

### ۱۳-۲-۱۱- مکانیزم REM

این روش توسط Sanjeewa Athuraliya وهمکارانش در سال ۲۰۰۱ ارائه گردید. جنبه های اصلی و کلیدی مکانیزم REM عبارتند از:

۱. این روش سعی دارد که نرخ ارسال کاربر را با ظرفیت شبکه انطباق دهد و درعین حال بافرها را تا حد امکان خالی نگه دارد.

۲. احتمال دورریختن /علامت زدن کلی به صورت مجموع ارزش های اتصالات موجود درمسیرارسال بسته های کاربر محاسبه می شود.

طراحی یک الگوریتم مدیریت صف، باید به سه سوال اصلی زیر پاسخ دهد:

- ازدحام چگونه اندازه گیری می شود؟
- این اندازه، چگونه درتابع احتمال به کارمی رود؟
- این اندازه چگونه به کاربر، برمی گردد؟

به عنوان مثال، RED ازدحام را با متوسط طول صف اندازه گیری می کند. تابع احتمال نیز درRED، تابعی خطی و صعودی می باشد و نهایتاً اطلاعات ازدحام به یکی از دو طریق دورریختن بسته ها یا ۱کردن بیت ECN آنها به کاربرانتقال می یابد. مکانیزم REM دردو سوال اول با RED تفاوت دارد. تعریف اندازه ازدحام و تابع احتمال دورریختن در REM متفاوت است.

همانطور که گفته شد، اولین جنبه کلیدی REM، تثبیت نرخ ورودی حول و حوش ظرفیت اتصال و همچنین تثبیت صف حول یک هدف کوچک ، بدون درنظرگرفتن تعداد کاربران استفاده کننده ازکانال، است. دراین الگوریتم، هرصف متغیری به نام price را به عنوان اندازه ازدحام مشخص می کند. این متغیر درتعیین احتمال دورریختن/علامت زدن به کار خواهد رفت. Price، براساس عدم تطبیق نرخ(یعنی تفاوت بین نرخ ورودی و ظرفیت اتصال) و همچنین عدم تطبیق صف (یعنی تفاوت بین طول صف و طول هدف) محاسبه می شود. Price درصورتی افزایش می یابد که مجموع این دوعدم تطبیق، مثبت باشد ودرغیراینصورت کاهش می یابد. این مجموع وقتی که نرخ ورودی از ظرفیت اتصال تجاوز کند مثبت می باشد ودرغیراینصورت منفی است. وقتی تعداد کاربران افزایش می یابد، عدم تطابق درنرخ وصف گسترش یافته و متغیر price و درنتیجه احتمال دورریختن رابالا می برند. این روند، سیگنال ازدحام قویتری را به منابع ترافیک می فرستد و آنها نیز نرخ خود را کاهش می دهند. وقتی نرخ ارسال منابع خیلی پایین است، این عدم تطابق منفی شده و منجر به پایین آمدن price و احتمال دورریختن بسته ها و بالارفتن نرخ ارسال منابع می شود، تازمانیکه درنهایت عدم تطابق به صفر متمایل شده و باعث بهره وری بالا و تاخیر قابل چشم پوشی می گردند.درمکانیزم REM، نحوه به روز رسانی متغیر price به صورت زیر است:

برروی صف l، price یا $P_l(t)$ درپریود t به صورت زیر بروز رسانی می شود:

**فرمول A** :
$$P_l(t+1) = [Pl(t) + \gamma(\alpha_l(b_l(t) - b_l^*) + x_l(t) - c_l(t))]^+$$

درفرمول فوق $\gamma > 0$ و $\alpha_l > 0$ ثابت های کوچکی هستند و $b_l(t)$ میزان پربودن بافر مربوط به صف l درپریود t است. $b_l^*$ طول صف هدف می باشد، $x_l(t)$ نرخ ورودی صف l درپرویود t و $c_l(t)$ پهنای باند موجوددر صف l درپریود t است.تفاضل $x_l(t) - c_l(t)$، عدم تطبیق نرخ و تفاضل $b_l(t) - b_l^*$، عدم تطبیق صف را اندازه می گیرد. بنابه رابطه بالا، price وقتی افزایش می یابد که مجموع عدم تطابق نرخ و صف با وزن $\alpha_l$مثبت باشد. درحالت تعادل، price تثبیت شده و این مجموع باید صفر شود( یعنی $\alpha_l(b_l - b_l^*) + x_l - c_l = 0$ ). این عبارت وقتی برقرار است که نرخ ورودی برابر ظرفیت کانال ( $x_l = c_l$ ) و طول صف برابر طول هدف ( $b_l = b_l^*$ ) باشد، که منجر به ویژگی اول REM می گردد.

ازآنجائیکه درعمل اندازه گیری طول صف از نرخ ورودی ساده تر است، می توان عبارت $x_l(t)-c_l(t)$ را با $b_l(t+1)-b_l(t)$ جایگزین نمود. دراین صورت فرمول A قبلی به صورت زیر درمی آید:

**فرمول B** :
$$P_l(t+1)=[P_l(t)+\gamma(b_l(t+1)-(1-\alpha_l)b_l(t)-\alpha_l b^*)]^+$$

دومین جنبه REM، بکارگیری مجموع price های اتصالات موجود درمسیر، به عنوان اندازه ازدحام مسیر است و استفاده ازاین اندازه در محاسبه احتمال دورریختن انتها به انتها است. بافرض اینکه بسته ای از اتصالات l (l=1,2,3,….) با price های $p_l(t)$ عبور می کند، احتمال دورریختن $m_l(t)$ برای صف l درپریود t به صورت زیر بدست می آید:

**فرمول C** :
$$m_l(t)=1-\Phi^{-p_l(t)}$$

که درآن $\Phi>1$ یک ثابت است. بنابراین احتمال دورریختن انتهابه انتها برای این بسته به صورت زیر محاسبه می شود:

$$1-\prod_{l=1}^{L}(1-m_l(t))=1-\Phi^{-\sum^{l}p_l(t)}$$

احتمال دور ریختن انتها به انتها وقتی بالا است که اندازه ازدحام مسیرآن ، یعنی $\sum p_l(t)$ بزرگ باشد. درصورتیکه احتمالات دور ریختن دراتصالات ، یعنی $m_l(t)$ کوچک باشند، price های اتصالات یعنی $p_l(t)$ نیز کوچک بوده و درنتیجه احتمال دورریختن انتها به انتهاتقریباً معادل است با مجموع price های اتصالات موجود درمسیر یعنی :

$$\text{احتمال علامت زنی انتها به انتهای } (\log_e \Phi)\sum_l p_l(t) \approx$$

دوجنبه فوق که درفرمول های B و C خلاصه شده اند را می توان مستقل از هم پیاده سازی نمود. به عنوان مثال ممکن است تنها از price برای اندازه گیری ازدحام استفاده شود، اما تابع احتمال دورریختن متفاوت ( مانند تابع احتمال RED) استفاده گردد . ویا ممکن است که به روش دیگری ازدحام اندازه گیری گردد (مثلاً بااستفاده از تاخیر، طول صف و ...) امابرای علامت زدن بسته ها از تابع احتمال نمایی استفاده شود.

### ۱۳-۲-۱۲- مکانیزم GREEN

GREEN یک تابع کنترلی فیدبکی می باشد که نرخ اطلاع رسانی ازدحام را با استفاده از نرخ تخمینی ورودداده تنظیم می کند. GREEN برپایه تابع حدآستانه عمل می کند. اگر نرخ تخمینی داده اتصال( $x_{est}$ ) بالاتر از ظرفیت اتصال ( $C_l$ ) باشد، نرخ اطلاع رسانی ازدحام (P) با نرخ $\frac{1}{\Delta T}$ به اندازه $\Delta P$ افزایش می یابد. برعکس اگر $x_{est}$ کمتر از $C_l$ باشد،P با همین نرخ به اندازه $\Delta p$ کاهش خواهد یافت.

چنانچه تابع $U(x)$ به صورت زیر تعریف شود:

$$U(x)=\begin{cases}+1 & x\geq 0\\ -1 & x<0\end{cases}$$

آنگاه می تون مقدار P را به صورت زیر محاسبه نمود:

$$P = P + \Delta P.U(x_{est} - C_l)$$

ظرفیت اتصال ( $C_l$ ) مقداری کمتر ازظرفیت واقعی آن است ( معمولاً $0.97C$ )، بنابراین طول متوسط صف به صفر نزدیک می شود. تخمین نرخ ورود داده با استفاده از میانگین گیری نمایی به صورت زیر محاسبه می شود:

$$x_{est} = (1 - \exp(-Del / K)) * (B / Del) + \exp(-Del / K).x_{est}$$

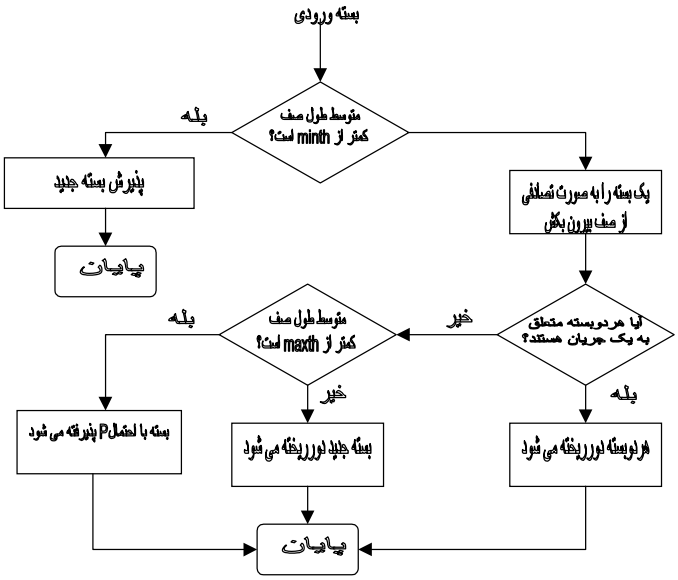
که دراین فرمول *Del*، تاخیر بین بسته ها، B اندازه بسته و K یک ثابت زمانی است. ازروش های دیگری نیز می توان برای تخمین نرخ ورود داده استفاده نمود.
براساس شبیه سازی های انجام شده ، مکانیزم GREEN به خوبی توانسته است که به بهره وری مناسب و حداقل احتمال اتلاف بسته ها دست یابد.

### ۱۳-۲-۱۳- مکانیزم CHOKE

این روش در سال ۲۰۰۱ توسط Rong Pan و همکارانش دردانشگاه استانفورد ارائه گردید. دراین روش سعی شده است که بتوان جریانات (کاربران ) غیرمسئول[1] و بدرفتار را شناسایی کرد و درصورت بروز ازدحام، بسته های مربوط به این دسته از کاربران را دورریخت. فرض کنید که مسیریاب دارای یک صف FIFO برای صف بندی بسته های مربوط به تمامی جریانات تقسیم کننده اتصال خروجی است. CHOKE به مانند RED با بکارگیری از یک میانگین نمایی، میانگین اشغال بودن بافر FIFO را محاسبه می کند و به علاوه دو حد آستانه $max_{th}$ و $min_{th}$ را نیز برای بافر مشخص می کند.اگر متوسط طول صف کمتر از $min_{th}$ باشد، تمامی بسته های ورودی در بافر FIFO قرار می گیرند.(اگر نرخ ورودی کمتر از ظرفیت اتصال خروجی باشد، متوسط طول صف اغلب از $min_{th}$ تجاوز نخواهد کرد). اگر متوسط طول صف از $max_{th}$ تجاوز کند، تمامی بسته های رسیده دورانداخته می شوند. این عمل میزان اشغال بودن صف را به کمتر از$max_{th}$ میرساند. وقتی که متوسط طول صف بیشتر از $min_{th}$ باشد، هربسته ورودی با یک بسته به نام بسته کاندیدای دورریختن که به طور تصادفی از بافر انتخاب شده است، مقایسه می گردد. اگر این دو بسته دارای شناسه جریان یکسان باشند، هر دو دورریخته می شوند، درغیر اینصورت بسته تصادفی انتخاب شده دربافرنگهداری شده (همان محل قبل) و بسته ورودی با احتمالی که وابسته به متوسط طول صف است دورریخته می شود. احتمال دورریختن دقیقاً مشابه RED محاسبه می گردد. به عبارت دقیق تر اگردرهنگام ورود بسته ها به صف متوسط طول صف از $max_{th}$ تجاوز کرده باشد، با احتمال ۱۰۰درصد دورریخته می شوند.دلیل اصلی این کاراین است که بافراحتمالاً دارای بسته هایی است که متعلق به جریان بدرفتارهستند و درنتیجه این بسته ها با احتمال بیشتری برای مقایسه انتخاب می شوند. به علاوه، بسته های متعلق به یک جریان بدرفتار، بسیار بیشتر دریافت می شوند و با احتمال بیشتری در مقایسه شرکت می کنند. اشتراک این دو احتمال قوی منجر به این امر می شود که بسته های متعلق به جریان بدرفتار دورریخته شوند.
دیاگرام الگوریتم CHOKE درشکل(۱۳-۷) آورده شده است. درحالت کلی می توان $m>1$ بسته را از بافرانتخاب نموده وهمگی آنها را با بسته ورودی مقایسه نمود و بسته هایی را که دارای شناسه جریان یکسان با بسته ورودی هستند را دورریخت. می توان دریافت که انتخاب بیش از یک کاندیدای دورریختن، کارآیی مکانیزم CHOKE را افزایش می دهد. به خصوص وقتی که چندین جریان غیرمسئول وجود دارند. درحقیقت وقتی تعداد جریان های غیرمسئول افزایش می یابد، لازم است که بسته های کاندید بیشتری انتخاب گردند. اما ازآنجائیکه دراین شیوه، تاکید بروابسته نبودن به وضعیت سیستم است، هیچ اطلاعی راجع به تعداد اتصالات غیرمسئول فعال درهرلحظه دردست نیست.

[1] unresponsive: درصورت بروز تراکم درشبکه، پاسخ طبیعی به تراکم کاهش نرخ ارسال و تلاش برای انطباق با ظرفیت موجود درکانال می باشد. پروتکل هایی نظیر TCP که چنین شیوه ای را استفاده می کنند، و فرمی از کنترل ترافیک را اجرا می نمایند، مسئول (responsive) نامیده می شوند. درعوض پروتکل هایی که این حالت را تشخیص نداده و یا روش نادیده گرفتن تراکم و ارسال با نرخ قبلی را انتخاب می کنند (مانند UDP) ، غیرمسئول نامیده می شوند.

می توان یک حدآستانه میانی به نام $in_{th}$ که فاصله بین $min_{th}$ و $max_{th}$ رابه دومنطقه تقسیم می کند، درنظرگرفت. هنگامیکه متوسط اشغال بودن بافر بین $in_{th}$ و $min_{th}$ است، m=1 و درصورتیکه این مقدار بین $in_{th}$ و $max_{th}$ است، مقدار m=64 درنظرگرفته می شود. درحالت کلی تر می توان چندین حدآستانه را بین$min_{th}$ و $max_{th}$ نیز درنظر گرفت.




**شکل(۱۳-۷): فلوچارت مکانیزم CHOKE**

مکانیزم CHOKE کاملاً بدون وضعیت بوده وبه هیچ ساختار داده ای نیاز ندارد بلکه فقط تعدادی محاسبه ساده دارد. انتخاب تصادفی بسته از صف، مقایسه شناسه های جریان و احتمالاً دورریختن هردوبسته ، نمونه ای از محاسبات این مکانیزم می باشند. ازآنجائیکه این روش از RED کمک گرفته است، جنبه های مثبت آن را نیز دارا می باشد.

### ۱۳-۲-۱۴-مکانیزم AVQ

این مکانیزم توسط Srisankar Kunniyur و R. Srikant درسال ۲۰۰۱ ارائه گردید. دراین روش از یک صف مجازی استفاده می شود که ظرفیت آن ( ظرفیت مجازی) کمتر از ظرفیت واقعی اتصال است. هنگامی که بسته ای به صف واقعی می رسد، صف مجازی نیز به روزآوری شده تاورود بسته جدید را منعکس کند. هنگامی که بافر مجازی پرشود، بسته هادرصف واقعی دورریخته/علامت زده می شوند،سپس ظرفیت مجازی درهراتصال به گونه ای تغییرمی کند که مجموع جریان های ورودی به اتصال به یک بهره وری مطلوب برسد.

فرض کنید که C نشاندهنده ظرفیت اتصال، $\gamma$ بهره وری مطلوب اتصال باشد، دراین صورت روش AVQ به صورت زیر درهر مسیریاب عمل می کند:

مسیریاب یک صف مجازی درنظر می گیرد که ظرفیت آن $\widetilde{C} \le C$ و اندازه بافرآن مساوی با اندازه بافرصف واقعی باشد. به ازای ورود هربسته، یک بسته مجازی درصف مجازی درج می شود(درصورت وجود فضای کافی دربافر). اگربسته جدید منجر به سرریزشدن بافرمجازی شود، بسته از بافرمجازی دورریخته می شودو بسته واقعی نیز دورریخته شده ویا بیت ECN آن علامت زده می شود. به ازای ورودی هر بسته ، ظرفیت صف مجازی براساس رابطه زیر به روز آوری می شود:

$$\widetilde{C} = \alpha(\gamma C - \lambda)$$

که درآن $\lambda$ نرخ ورودی اتصال است. لازم به ذکر است که هیچگونه ورود و خروج واقعی بسته درصف مجازی صورت نمی گیرد و فقط تغییرات لازم در طول صف مجازی اعمال می شود.

الگوریتم AVQ به صورت زیر است:

- تعاریف :

B = طول بافر

S= زمان ورودی بسته قبلی

t= زمان فعلی

b = تعداد بایت های بسته جدید

VQ = تعداد بایتهای موجود درصف مجازی درحال حاضر

- به ازای ورود هربسته اعمال زیر انجام می شود:

VQ ←-max(VQ- $\widetilde{C}(t-s)$ ,0) /*به روزرسانی طول صف مجازی */

If VQ+b > B

Mark or drop packet in the real queue

Else

VQ←-VQ + b /* به روزرسانی طول صف مجازی */

**Endif**

$\widetilde{C}$ =max(min( $\widetilde{C} + \alpha.\gamma.C(t-s)$ ,C)- $\alpha.b$ ,0) /* به روزرسانی ظرفیت مجازی */

s←t /* به روزرسانی زمان ورود آخرین بسته */

پیچیدگی پیاده سازی مکانیزم AVQ قابل قیاس با RED می باشد. مکانیزم RED اعمالی نظیر: میانگین گیری طول صف، محاسبه احتمال دورریختن و تولید اعداد تصادفی را برای انجام تصمیم گیری دردورریختن اجرا می کند. این اعمال درAVQ با محاسبه ظرفیت مجازی جایگزین شده اند.

### ۱۳-۲-۱۵- مکانیزم CBT[1]

در این الگوریتم سعی بر این است که ظرفیت بافر بین تمامی انواع مختلف جریانها تقسیم گردد. هدف این الگوریتم تفکیک بین کلاسهای مختلف ترافیک و کاهش تاثیر گرسنگی[2] جریانهای غیرمسئول بر جریانهای مسئول می‌باشد. علاوه بر این، به علت اینکه کاربردهایی وجود دارند که دلایلی منطقی برای استفاده از پروتکلهای غیرمسئول دارند، (مانند کاربردهای چندرسانه‌ای حساس به تأخیر) چنین کاربردهایی نباید بطور کامل و یکطرفه از استفاده از پهنای باند محدود شوند، بلکه در عین حال که باید جریانهای مسئول را از تاثیرات مخرب جریانهای غیرمسئول حفظ نمود، از طرف دیگر باید امکان این را فراهم آورد که جریانهایی از قبیل آنچه در بالا گفته شد بتوانند سهم مشخص شده خود از ظرفیت اتصال را دریافت دارند.

در این روش برای کلیه کلاسهای ترافیکی، صفی واحد وجود دارد؛ اما به هر یک از کلاسهای ترافیکی مجوز استفاده بخشی از ظرفیت اتصال داده می‌شود. این مجوز از طریق انتساب حدود آستانه به هر یک از کلاسهای ترافیک صورت می‌گیرد. این حدود آستانه بر روی متوسط طول صف مربوط به بسته‌های هر کلاس تنظیم می‌گردند. به عبارت دیگر برای هر کلاس سرویس، حد آستانه‌ای بر روی طول صف بسته‌های مربوط به آن کلاس وجود دارد، که سهم آن کلاس را از ظرفیت اتصال کنترل می‌کند.

CBT سه کلاس تعریف می‌کند: UDP چند رسانه‌ای، جریانهای دیگر UDP و TCP. برای هر یک از دو کلاسUDP ، CBT یک حد آستانه ایستا تعریف کرده و متوسط طول صف مربوط به این کلاسها را نگهداری می‌کند و زمانی که متوسط طول صف بسته‌های مربوط به هر کلاس از حد آستانه آن صف تجاوز کند، بسته‌های ورودی مربوط به این کلاس دور ریخته می‌شوند. برای کلاس TCP ، این الگوریتم، مکانیزم RED را با دو حد آستانه مربوطه، اعمال می‌کند و بدین ترتیب بروز

[1] Class-Based Thresholds

[2] starvation

ازدحام را از پیش اطلاع می‌دهد. با اجرای تست بر روی حد آستانه هر کلاسUDP ، CBT جریانهای TCP را از جریانهای غیرمسئول یا بدرفتار UDP و همچنین جریانهای چندرسانه‌ای UDP را از مابقی جریانهای UDP حفظ می‌کند.
نسبت حدود آستانه کلاسها مشخص کننده نسبت پهنای باند تخصیص یافته به هر کلاس در طول بازه‌های زمانی ازدحام است. مزایای این روش، کمتر بودن میزان تأخیر و تعداد بسته‌های دور ریخته شده در مقایسه با RED و FRED است. بعلاوه با بکارگیری CBT بجای RED، TCP سهم بیشتری از پهنای باند را در مواجهه با ترافیکهای غیرمسئول دریافت می‌نماید. یکی از محدودیتهای این روش، ایستا بودن حدود آستانه است که برای ترافیک بسیار متغیر اینترنت منجر به کارایی پایین می‌گردد.

### ۱۳-۲-۱۶- مکانیزم D-CBT[1]

در این الگوریتم به منظور رفع محدودیتهای CBT حدود آستانه بطور پویا و عادلانه بر اساس شرایط موجود شبکه برای دو کلاس UDP تعیین می‌گردند. به منظور محاسبه این حدود آستانه (که سهم عادلانه هر کلاس از متوسط بافر خروجی را مشخص می‌کنند) D-CBT نیاز به اطلاعات وضعیت هر کلاس دارد. در نتیجه باید تعداد جریانهای فعال هر کلاس را محاسبه کند. همانند FRED جریانهای فعال جریانهایی هستند که بسته‌ای در صف داشته باشند. با ورود هر بسته ابتدا در واحدی به نام واحد حسابداری، اطلاعات مربوط به تعداد جریانهای فعال به‌روز شده و سپس متوسط طول صف سه کلاس به‌روز می‌گردد.
با استفاده از طول صف متوسط سه کلاس و تعداد جریانهای فعال موجود در هر کلاس، حدود آستانه عادلانه برای کلاسهای UDP محاسبه شده و سپس تست حد آستانه بر روی بسته‌های ورودی کلاس صورت می‌گیرد. بسته‌های UDP که از این تست عبور می‌کنند به همراه بسته‌های TCP وارد واحد RED می‌شوند که در آن، تست دور ریختن زود هنگام برروی آنها صورت خواهد گرفت.

### ۱۳-۲-۱۷-کنترل کننده *PI* [2]

یکی از محدودیتهایی که RED با آن مواجه است، وجود trade off بین سرعت پاسخگویی و ثبات آن است. یک طراحی که دارای زمان پاسخگویی سریعی است، معمولا ثبات نسبتا پایین‌تری را داراست و بالعکس، طراحی‌ای که دارای ثبات و پایداری مناسبی است، معمولا دارای زمان پاسخگویی بالایی است. محدودیت دیگری که در RED وجود دارد، ارتباط مستقیم بین طول صف و احتمال دور ریختن است. این ارتباط مستقیم باعث می‌شود در حالتی که سیستم با بار زیادی روبروست، جریانهای وارده دچار تأخیر و از دست دادن سلول بیشتری گردند. درکنترل کننده *PI* ،کنترل کننده‌ای طراحی شده است که با مشاهده و بررسی پارامترهایی از سیستم، سعی در نگهداری طول صف در حول مقدار مرجع (q_ ref) داشته و در عین حال محدودیتهای بیان شده برای RED را نیز مرتفع سازد.
کنترل کننده *PI* در بازه‌های زمانی مشخص، پارامترهایی همچون طول فعلی صف، طول صف و احتمال دور ریختن مربوط به بازه زمانی قبل و طول صف مطلوب، مقدار جدید احتمال دور ریختن بسته را طبق فرمولی محاسبه و بصورت بازخورد به الگوریتم RED بر می‌گرداند. شبه‌کد این الگوریتم بصورت زیر است:

```
at every sampling interval
        {
                p= a* (q-q_ref)-b* (q_old – q_ref) + p_old
                q _old = q
                p _old =p
        }
```

[1] Dynamic CBT
[2] Proportional Integral Controller

در این شبه‌کد، q طول صف فعلی، q _ref طول صف مطلوب (مرجع)، $q\_old$ و p_old به ترتیب طول صف و احتمال مربوط با دوره تناوب قبلی و a و b اعدادی ثابت هستند.

## ۱۳-۳- مکانیزم های زمانبندی درشبکه های IP

نسل بعدی اینترنت دو نوع کاربرد را پشتیبانی می‌کند: کاربردهای بهترین تلاش و کاربردهای سرویس تضمین‌شده. کاربردهای بهترین تلاش که هم اکنون کاربرد متداول اینترنت هستند، هر مقدار کارایی که شبکه به آنها تخصیص دهد برای آنها راضی کننده است. به عنوان مثال با وجود اینکه یک کاربرد انتقال فایل ترجیح می‌دهد که با حداقل تأخیر و پهنای باند نامحدود مواجه شود، اما می‌تواند خود را با منابع موجود و محدود در شبکه انطباق دهد.

علاوه بر این کاربردها، امروزه اینترنت باید ترافیکهایی را حمل کند که نیاز به محدوده‌های کارایی دارند. به عنوان مثال کاربردی که حاوی صوتی بصورت یک رشته داده ۶۴ کیلو بایت در هر ثانیه می‌باشد، در صورتی که شبکه نتواند پهنای باندی معادل ۶۴ کیلو بایت در ثانیه را برای آن فراهم کند دیگر قابل استفاده نخواهد بود.کارایی دریافت شده توسط ارتباط، بسته به مکانیزم زمانبندی دارد. این مکانیزمها توسط نودهای میانی که در مسیر بین مبدا و مقصد ارتباط قرار دارند اجرا می‌شوند. یک مسیریاب منبعی اشتراکی در شبکه محسوب می‌شود. بسیاری از مشکلات که در شبکه با آن مواجه می‌شویم مربوط به تخصیص مقدار محدودی منابع اشتراکی (حافظه بافر و پهنای باند درگاه خروجی) بین کاربران، کاربردها و کلاسهای سرویس می‌باشد. الگوریتمهای زمانبندی صف از طریق انتخاب بسته بعدی که باید از طریق درگاه انتقال یابد، امکان دستیابی به میزان ثابتی از پهنای باند خروجی را مدیریت می‌نماید.

زمانبند می‌تواند تأخیر صف‌بندی و پهنای باندهای متفاوت را به ارتباطهای مختلف تخصیص دهد. این عمل از طریق انتخاب ترتیب سرویس‌دهی و با سرویس‌دهی تعداد مشخصی بسته از یک ارتباط خاص صورت گیرد. همچنین زمانبند می‌تواند نرخ دورریختن‌های متفاوتی را با نسبت دادن مقدار مشخصی فضای بافر به ارتباط‌ها به آنها تخصیص دهد. به علاوه زمانبند می‌تواند منابع را بصورت عادلانه بین ارتباط‌های بهترین تلاش تقسیم نماید. در نتیجه، نسل بعدی اینترنت به مکانیزمهای زمانبندی به دو منظور نیازمند است:

۱- پشتیبانی از محدوده‌های تأخیر، پهنای باند و دور ریختن سلول برای کاربردهای با سرویس تضمین‌شده.

۲- پشتیبانی از تخصیص عادلانه منابع مورد نیاز کاربردهای بهترین تلاش .

الگوریتمهای زمانبندی مختلفی وجود دارند که همگی به دنبال یافتن یک تعادل صحیح بین پیچیدگی، کنترل و عدالت می‌باشند. همانگونه که گفته شد ازدحام زمانی اتفاق می‌افتد که بسته‌ها با سرعت بالاتر از آنچه بتوان آنها را انتقال داد، به درگاه خروجی وارد شوند. به حداقل رساندن ازدحام در شبکه مساله مهمی است. چرا که ازدحام بهره‌وری را کاهش داده، تأخیر را افزایش می‌دهد و موجب واریانس بالا در تأخیر می‌شود و ممکن است در اثر نبود فضای کافی برای ذخیره کلیه بسته‌هایی که منتظر ارسال هستند، منجر به از دست رفتن سلول شود.

در طراحی یک زمانبند، باید یکسری نیازمندیها در نظر گرفته شوند. در شرایطی خاص ممکن است برخی از این نیازمندیها مهمتر از بقیه باشند و در نتیجه تصمیم‌گیری درباره اولویت دادن به هر یک از این نیازمندیها بستگی به شرایط خاص داده شده دارد. این نیازمندیها عبارتند از:

- پیچیدگی : روشهای زمانبندی از لحاظ پیچیدگی پیاده‌سازی سخت‌افزاری و کنترلی متفاوتند. پیچیدگی الگوریتم دارای اهمیت ویژه‌ای است چرا که این الگوریتم باید هر زمان که مسیریاب بخواهد بسته‌ای را انتخاب و خارج کند اجرا گردد. فرکانس خروج بسته بستگی به سرعت اتصال خروجی دارد و می‌تواند یکی در هر چند میکروثانیه باشد. در نتیجه یک مکانیزم زمانبندی فقط باید چند عمل ساده را انجام دهد. به علاوه زمانبند باید قابل پیاده‌سازی ارزان بصورت سخت‌افزاری نیز باشد.

- عدالت : از آنجا که یک مکانیزم زمانبندی سهمی از منابع شبکه مانند پهنای باند درگاه خروجی و ظرفیت بافر را به جریانها نسبت می‌دهد، عدالت مساله مهمی در پشتیبانی از کاربردهای مختلف می‌باشد. زمانبند باید بتواند توزیع عادلانه پهنای باند را برای تمامی کلاس سرویسهای مختلف که برای دستیابی به پهنای باند با یکدیگر در رقابت هستند تضمین کند.
- تفکیک (حفاظت) : که عبارتست از جلوگیری از تاثیر مخرب جریانهای بدرفتار بر روی جریانهای خوش‌رفتار. جریانهای بدرفتار آنهایی هستند که بیشتر از سهم عادلانه خود بسته ارسال می‌کنند.
- کارایی : همانگونه که پیش از این گفته شد، هدف اصلی الگوریتم زمانبندی تضمین محدوده‌های کارایی برای اتصال، مانند بهره‌وری، تأخیر و دور ریختن بسته است. شبکه و کاربران آن بر روی یک سری پارامترهای ترافیکی مشخص توافق می‌نمایند. کاربران نباید از محدوده‌های مشخص شده در قرارداد تجاوز کنند و شبکه نیز باید تضمین نماید که نیازهای سرویس آنها را تأمین نماید. تضمین محدوده‌های کارایی در شبکه‌های امروزی کار مشکلی است؛ چرا که همه زمانبندهای موجود در مسیر ارتباط باید در تأمین آن مشارکت کنند.
- استفاده از پهنای باند اضافی : الگوریتم زمانی باید به کلاسهای سرویس امکان دهد تا در صورتی که یک کلاس سرویس از تمامی پهنای باند تخصیص یافته به آن استفاده نکند، به آن پهنای باند مازاد، دستیابی داشته باشند.

مکانیزمهای زمانبندی به دو دسته تقسیم می‌شوند که عبارتند از: work conserving و non work conserving . یک زمانبند work conserving زمانی بیکار است که هیچ بسته‌ای در صفهای مسیریاب منتظر نباشد. در یک زمانبند non work conserving، به هر بسته یک زمان نسبت داده می‌شود که باید در آن زمان به درگاه خروجی ارسال شود. زمانبند تا زمانی که بسته بعدی برای انتقال آماده نشود بیکار می‌ماند.

یکی از خواص الگوریتمهای work conserving این است که آنها دارای حداقل متوسط کل تأخیر صف‌بندی هستند. متوسط کل تأخیر صف‌بندی، بر روی تمامی جریانهایی که باید سرویس داده شوند محاسبه می‌شود. به علاوه متوسط کل تأخیر صف‌بندی برای تمامی الگوریتمهای work conserving مساوی است. به عبارت دیگر، با وجود اینکه زمانبندهای مختلف الگوریتمهای مختلفی را برای انتخاب بسته بعدی بکار می‌برند، تأخیر متوسط صف‌بندی همیشه یکسان است.

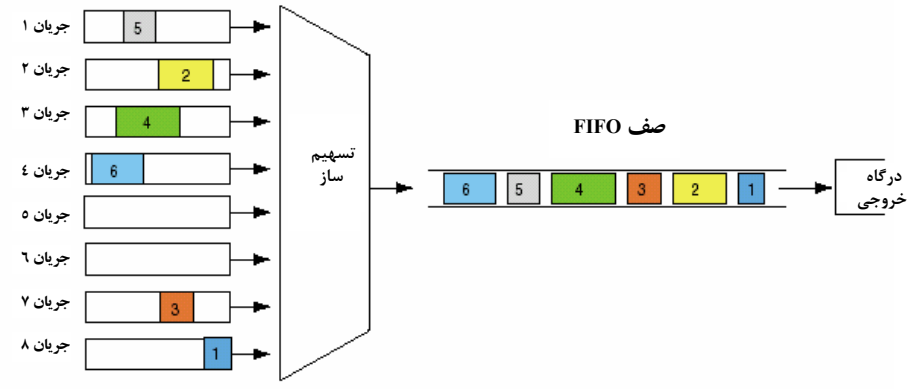
یک سؤال در اینجا مطرح می‌شود و آن این است که: چرا باید از الگوریتمهای non work conserving استفاده کنیم و در نتیجه با بیکار گذاشتن اتصال، پهنای باند را هدر دهیم. پاسخ این است که الگوریتمهای non work conserving جریان ترافیکی وارده به مسیریابها را قابل پیش‌بینی‌تر می‌نمایند. چرا که اتصال را برای دوره تناوبهای کوتاهی از زمان بیکار نگه می‌دارند. این الگوریتمها عموما بهترین انتخاب برای شبکه‌های با ترافیک بلادرنگ هستند. چرا که می‌توانند محدوده‌هایی مشخص بر روی تأخیر و واریانس تأخیر ایجاد کنند. این امر از طریق تأخیر دادن بسته‌ها به منظور دستیابی به نیازهای تأخیر خاص، صورت می‌گیرد.

در ادامه مهمترین و متداولترین مکانیزمهای زمانبندی بررسی می‌شوند:

### ۱۳-۳-۱- زمانبندی FIFO

این الگوریتم ، یکی از ساده‌ترین ساختارهای زمانبندی صف است. در صف‌بندی FIFO، تمام بسته‌ها در یک صف مشترک قرار گرفته و با آنها بطور یکسان برخورد می‌شود. در این روش، سرویس‌دهی بسته‌ها به ترتیب ورود آنها به صف انجام شده و به آن، صف‌بندی اولین ورودی، اولین سرویس‌دهی (FIFS[1]) نیز گفته می‌شود. درشکل(۱۳-۸) مثالی از صف بندی FIFO آورده شده است.

[1] Firs In First Serve




**شکل(۱۳-۸): مثالی از صف بندی FIFO**

مکانیزم FIFO دارای مزایای زیر است:

- در مقایسه با سایر روشها این صف‌بندی بار محاسباتی بسیار کمی در مسیریابهای مبتنی برنرم افزار دارد.
- در یک صف FIFO ترتیب بسته ها به هم نمی خورد وحداکثر تاخیر، با حداکثر اندازه صف تعیین می‌شود.
- تا زمانی که اندازه صف کوتاه باشد صف FIFO عملکرد خوبی دارد چرا که در هرنود شبکه، تاخیر صف‌بندی زیادی وجود ندارد.

همچنین این مکانیزم دارای محدودیتهای زیر است.

- نمی‌توان یک کلاس ترافیک را بطور جداگانه‌ای از سایر کلاسهای ترافیکی سرویس‌داد.
- با اینکه تاخیر صف‌بندی متوسط برای تمام جریانها با افزایش ازدحام، زیاد می‌شود تاخیر مشاهده شده برای تمام جریانها با هم مساوی است. در نتیجه، صف‌بندی FIFO می‌تواند برای کاربردهای بلادرنگی که از یک صف FIFO عبور می‌کنند موجب افزایش اتلاف ، تاخیر و تغییرات تاخیر بسته‌ها می‌شود و از آنجا که این کاربردها به این مسئله، حساس هستند موجب مشکل می‌شود.
- در زمان هایی که شبکه با ازدحام مواجه می شود، صف‌بندی FIFO بیشتر به نفع جریانهای UDP عمل می‌کند تا جریانهای TCP . وقتیکه بخاطر ازدحام، تلف بسته‌ای مشاهده شد کاربردهای برمبنای TCP نرخ ارسالشان را کاهش می‌دهند اما کاربردهای برمبنای UDP به تلف بسته‌ها بی‌توجه بوده و با نرخ همیشگی و معمول خود به ارسال بسته‌ها ادامه می‌دهند.
- یک جریان انفجاری می‌تواند کل فضای بافر یک صف FIFO را اشغال کند و باعث عدم سرویس‌دهی مناسب به سایر جریانها شود.

وقتیکه از این الگو استفاده می‌شود، اغلب دو صف در روی درگاه خروجی استفاده می‌شود. این صفها عبارتند از:

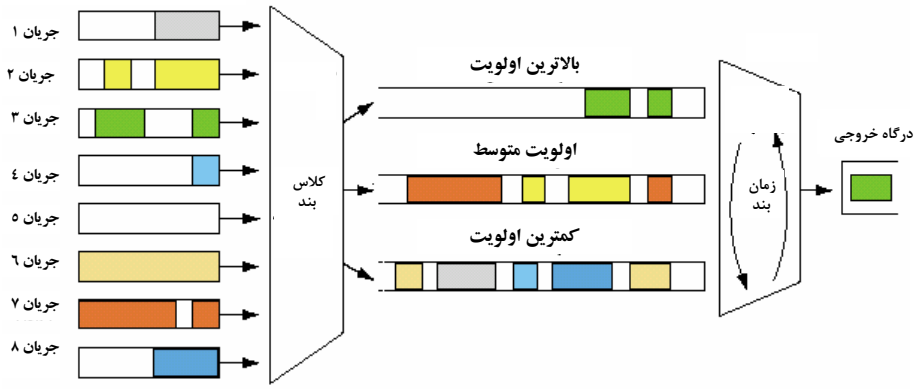
۱. یک صف اولویت بالا که برای زمانبندی کنترل ترافیک شبکه استفاده می‌شود.
۲. یک صف FIFO که سایر انواع ترافیک را زمانبندی می‌کند

### ۱۳-۳-۲- صف‌بندی با اولویت (PQ[1])

این نوع صف بندی، پایه‌ای برای روشهایی از الگوریتمهای زمانبندی صف است که هدف از آنها مهیا کردن روشی نسبتاً ساده

[1] Priority Queuing

برای پشتیبانی از کلاسهای سرویس جدا شده می‌باشد. در PQ کلاسیک، ابتدا سیستم، بسته‌ها را طبقه‌بندی کرده و سپس آنها را درصفهایی با اولویت مختلف قرار می‌دهد. بسته‌های سر هر صف در صورتی زمانبندی می‌شوندکه تمام صفهای دارای اولویت بالاتر خالی باشند. لازم بذکر است که در داخل هر صف بااولویت، بسته‌ها به صورت FIFO زمانبندی می‌شوند. درشکل(۱۳-۹) مثالی از صف بندی PQ آورده شده است.




**شکل(۱۳-۹): مثالی از صف بندی PQ**

مکانیزم PQدارای دو مزیت زیر است:

- در مقایسه با خیلی از الگوهای صف‌بندی جدید، PQ بار محاسباتی کمی برای مسیریابهای مبتنی برنرم افزار دارد.
- به مسیریابها اجازه سازماندهی بسته‌های بافر شده را می‌دهد تا بتوانند یک کلاس ترافیک را جدا از سایر کلاسهای ترافیک، سرویس‌دهی کنند. برای مثال می‌توان اولویتها را طوری فعال کرد که کاربردهای بلادرنگ، از قبیل ویدئو و صوت ، دارای اولویت بالاتری از کاربردهای غیربلادرنگ باشند.

محدویتهای PQ از قرار زیر است:

- اگر در لبه شبکه، کنترلی بر مقدار ترافیک اولویت بالا انجام نشود ممکن است بسته‌های با اولویت پایینتر، تاخیر اضافی را مشاهده کنند، چرا که باید برای اتمام سرویس‌دهی ترافیک های اولویت بالای کنترل نشده منتظر بمانند.
- اگر حجم ترافیک اولویت بالا خیلی زیاد شود، باعث سرریزشدن ترافیک اولویت پایین و در نتیجه تلف بسته‌های اولویت پایین می‌شود.
- یک جریان ترافیکی اولویت بالای متخلف می‌تواند بطور چشمگیری بر تاخیرو تغییرات تاخیرسایرجریانهای اولویت بالا که از همان صف استفاده می‌کنند بیفزاید.
- از آنجا که درهنگام ازدحام درشبکه، ترافیک UDP بر ترافیک TCP برتری دارد، PQ راه حلی برای رفع مشکلات صف‌بندی FIFO بحساب نمی‌آید. اگر در مقایسه با ترافیک UDP، به ترافیک TCP اولویت بالاتری داده شود، مکانیزمهای کنترل جریان و مدیریت پنجره TCP تلاش بر مصرف تمام پهنای‌باند در دسترس کرده و از اینرو باعث کاهش شدید پهنای‌باندردسترس برای جریانهای UDP اولویت پایین می‌شود.

سازندگان سوئیچها ومسیریاب های شبکه غالباًمکانیزم PQ را طوری پیاده سازی می کنندکه دریکی ازدوحالت زیرکارکند:

- صف‌بندی اولویت شدید[۱]
- صف‌بندی اولویت با نرخ کنترل شده

درصف بندی با اولویت شدید، این اطمینان وجوددارد که بسته‌های صف اولویت بالا، همیشه قبل از بسته‌های صف دارای

[۱] strict priority queuing

اولویت پایین زمانبندی شوند. البته، مشکل این روش این است که ترافیک اولویت بالای زیاد، می‌تواند منجر به کمبود پهنای‌باند برای کلاسهای سرویس اولویت پایینترشود. با این وجود ممکن است که بعضی از متصدیان شبکه، بخواهند که شبکه‌هایشان از این نوع رفتار پشتیبانی کند.

درصف بندی اولویت بانرخ کنترل شده ،فقط درصورتیکه مقدار ترافیک موجود درصف اولویت بالا کمتر از یک مقدار آستانه (تعیین شده توسط کاربر) شود به بسته‌های صف اولویت بالا اجازه می‌دهد که قبل از بسته‌های با اولویت پایینتر زمانبندی شوند.

برای مثال، یک صف دارای اولویت بالا را در نظر بگیرید که دارای نرخ محدود شده‌ای به ۲۰ درصد پهنای‌باند درگاه خروجی است. تا زمانیکه صف اولویت بالا کمتر از ۲۰ درصد پهنای‌باند درگاه خروجی را مصرف می‌کند بسته‌های این صف، قبل از بسته‌های اولویت بالا زمانبندی می‌شوند. زمانیکه صف اولویت بالا بیشتر از ۲۰ درصد پهنای‌باند درگاه خروجی را مصرف می‌کند بسته‌های اولویت پایین، قبل از بسته‌های اولویت بالا زمانبندی می‌شوند. وقتی این مسئله اتفاق بیفتد دیگر هیچ استانداردی وجود نداشته و هرسازنده سوئیچ، چگونگی زمانبندی بسته‌های اولویت پایین را (قبل از بسته‌های اولویت بالا) تعیین می‌کند.

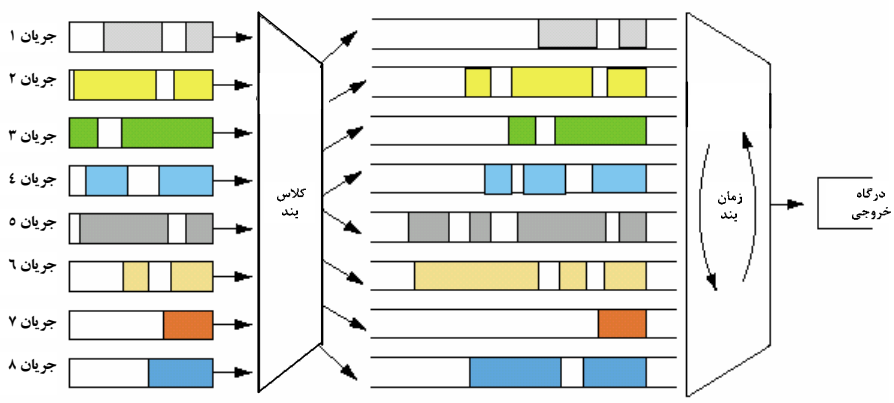
### ۱۳-۳-۳- مکانیزم صف بندی عادل (FQ[1])

این روش توسط John Nagle در سال ۱۹۸۷ پیشنهاد گردید. FQ ، زیربنایی برای کلاسی از الگوهای زمانبندی است که :

۱- اطمینان می‌دهند هر جریان بطور منصفانه‌ای به منابع شبکه دسترسی داشته باشد.

۲- از صرف پهنای‌باند بیشترازتسهیم‌منصفانه، توسط یک جریان انفجاری جلوگیری می‌کنند.

در این روش، ابتدا بسته‌ها توسط سیستم به جریانهایی کلاس‌بندی شده و سپس به صفی که برای آن جریان، پیش‌بینی شده است ملحق می‌شوند. سپس یک بسته در هر Round-Robin سرویس‌دهی می‌شود. همچنین صفهای خالی در نظر گرفته نمی‌شوند. به FQ ، صف‌بندی برمبنای جریان نیز گفته می‌شود. درشکل(۱۳-۱۰) مثالی از عملکرد الگوریتم FQ آورده شده است:




**شکل(۱۳-۱۰): مثالی از الگوریتم FQ**

مکانیزم FQ دارای مزیت زیر می باشد:

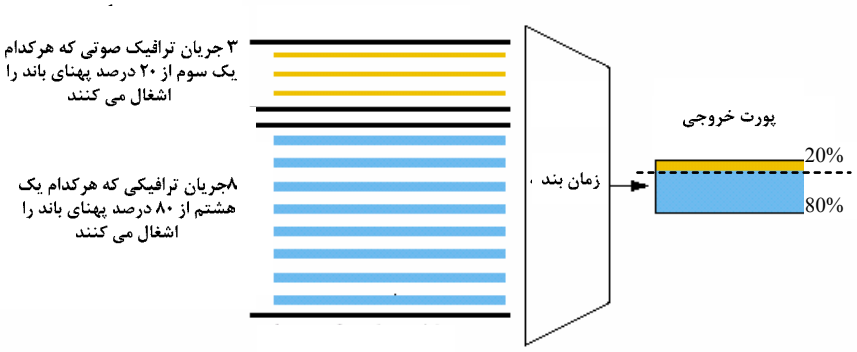
- با توجه به اینکه هر جریان، بطور مجزا و در صفی مخصوص به خودش قرار دارد یک جریان بشدت انفجاری یا متخلف نمی‌تواند باعث کاهش کیفیت سرویس تحویلی به سایر جریانها شود و اگر جریانی سعی در مصرف بیشتری از مقدار

[1] Fair Queuing

تسهیم منصفانه‌اش داشته باشد فقط صف مربوط به همان جریان، تاثیر گرفته و اثری بر عملکرد سایر صفها ندارد.
محدودیتهای این روش عبارتند از :

- این صف‌بندی بصورت نرم‌افزاری (نه سخت‌افزاری) پیاده سازی می‌شود. این محدودیت، باعث کاهش سرعت کار می‌شود.
- هدف از FQ تخصیص مقدارپهنای‌باندمساوی به تمام جریانها در هر لحظه از زمان است و برای پشتیبانی از جریانهای دارای پهنای‌باندمختلف، طراحی نشده است.
- اگر تمام بسته‌ها درتمام صفها دارای اندازه یکسانی باشند، FQ برای تمام جریانها پهنای‌باند یکسانی را مهیا می‌کند. در غیر اینصورت، جریانهایی که اغلب دارای بسته‌های بزرگی هستند سهم بیشتری از پهنای‌باندرگاه خروجی را در مقایسه با جریانهایی که دارای بسته‌های کوچک هستند بدست می‌آورند.
- FQ به ترتیب ورود بسته‌ها حساس است. اگر بلافاصله بعد از اینکه زمانبند round-robin، صف را ملاقات کرد بسته‌ای به یک صف خالی برسد، باید برای ارسال، تا اتمام سرویس‌دهی صفهای دیگر منتظر بماند.
- در FQ زمینه پشتیبانی آسان از سرویسهای بلادرنگ (همچون VoIP) وجود ندارد.
- در FQ فرض می‌شود که براحتی می‌توان ترافیک شبکه را به جریانهای دلخواه، کلاس‌بندی کرد، اما این مسئله، در شبکه‌های IP آنقدرساده نیست.
- FQ را نمی‌توان در مسیریاب مرکزی استفاده کرد. چرا که مسیریاب مرکزی باید بتواند برای هردرگاه، از هزاران یا ده‌ها هزار صف مجزا پشتیبانی کند. این مسئله، در شبکه‌های IP بزرگ باعث افزایش پیچیدگی مدیریت سرآیند ها شده و درنتیجه می‌تواند اثر معکوسی بر مقیاس‌پذیری FQ داشته باشد.

اگر n صف را در نظر بگیرید به هر صف، 1/n پهنای‌باند درگاه خروجی تخصیص داده می‌شود. اگر تعداد صفها از n به n+1 تغییر کند مقدار پهنای‌باند تخصیص داده شده به هر صف، از 1/n به 1/(n+1) پهنای‌باندرگاه خروجی کاهش می‌یابد.
در FQ برمبنای کلاس، درگاه خروجی به تعدادی از کلاسهای سرویس مختلف تقسیم می‌شود. هر کلاس سرویس، دارای درصد از پیش تعیین شده‌ای از پهنای‌باندرگاه خروجی می‌باشدکه توسط کاربر تعیین می‌شود. سپس FQ در داخل بلاک پهنای‌باندی که برای هر کلاس سرویس، تخصیص داده شده بکار برده میشود. در نتیجه تمام جریانهای تعیین شده برای یک کلاس سرویس، سهم منصفانه‌ای از پهنای‌باند مجموع تعیین شده برای آن کلاس سرویس مشخص را بدست می‌آورند.درشکل (۱۳-۱۱) مثالی از عملکرد صف بندی FQ آورده شده است.




**شکل(۱۳-۱۱): مثالی از صف بندی FQ**

در شکل فوق فرض شده که فقط دو سرویس، از یک درگاه خروجی استفاده می‌کنند. بدینصورت که به VoIP، ۲۰ درصد

وبه سایرجریان های ترافیکی، ۸۰ درصد پهنای‌باند لینک خروجی تخصیص داده شده است. در مدل FQ برمبنای کلاس، به هر جریان VoIP، نسبتی مساوی از ۲۰ درصد از پهنای‌باند لینک خروجی، و به هر جریان ترافیکی دیگرنسبتی مساوی از ۸۰ درصد از پهنای‌باند لینک خروجی تخصیص می‌یابد. بنابراین جریانهای VoIP برخوردی با سایر جریان ها ندارند و بالعکس.

### ۱۳-۳-۴- زمانبندی RR[1]

در این روش، محور زمان به چارچوبهایی با طول حداکثر F تقسیم می‌شود. بسته‌های مربوط به جریانهای مختلف بصورت چرخشی سرویس می‌گیرند. بعلاوه از آنجاییکه جریانهای مختلف ممکن است نیازهای متفاوتی داشته باشند، تعداد بسته‌های مربوط به یک جریان که در طول یک چارچوب سرویس می‌گیرند، محدود است. یک شمارنده اعتباری به هر ارتباط مربوط می‌شود و هر زمان که بسته‌ای از آن ارتباط سرویس گرفت، این شمارنده یک واحد کاهش می‌یابد. در آغاز هر چارچوب، شمارنده‌ها برابر با حداکثر میزان ترافیکی که یک ارتباط ممکن است در طول یک چارچوب ارسال کند، می‌شوند. در صورتی که شمارنده یک ارتباط صفر شود، ارتباط، دیگر قادر به سرویس‌گیری نمی‌باشد. سادگی این الگوریتم پیاده‌سازی سریع آنرا ممکن ساخته است.

مانند روش Stop and Go حداکثر تأخیر معادل است با حداکثر طول چارچوب. نیاز به تخصیص خوب و دقیق پهنای باند، منجر به طول چارچوب بزرگ و در نتیجه محدوده بزرگ تأخیر می‌گردد.

### ۱۳-۳-۵- زمانبندی WRR[2] (یا همان CBQ[3])

این روش، زیربنای کلاسی از الگوهای زمانبندی صف است که هدف از آنها رفع محدودیتهای مدلهای FQ و PQ می‌باشد که دارای مشخصات زیر است:

- این روش، محدودیتهای مدل FQ را با پشتیبانی از جریانهای با احتیاجات پهنای‌باند مختلف حل می‌کند. در واقع، در این روش می‌توان به هر صف، درصدی مختلفی از پهنای‌باند درگاه خروجی را تخصیص داد.
- همچنین این روش، محدودیتهای مدل PQ را با اطمینان به صفهای با اولویت پایینتر برای دستیابی دائمی به فضای بافر و پهنای‌باند درگاه خروجی حل می‌کند.

در صف بندی WRR ،ابتدا بسته‌ها به کلاسهای سرویس مختلفی تقسیم شده (برای مثال، بلادرنگ، interactive، و انتقال فایل) و سپس به صفی که برای آن کلاس سرویس درنظر گرفته شده است، انتقال می‌یابند. هرکدام از این صفها یکبار در هر round robin سرویس‌دهی شده و مشابه با PQ و FQ، صفهای خالی در نظر گرفته نمی‌شوند. به این صف‌بندی، صف‌بندی برمبنای کلاس (CBQ) نیز گفته می‌شود.

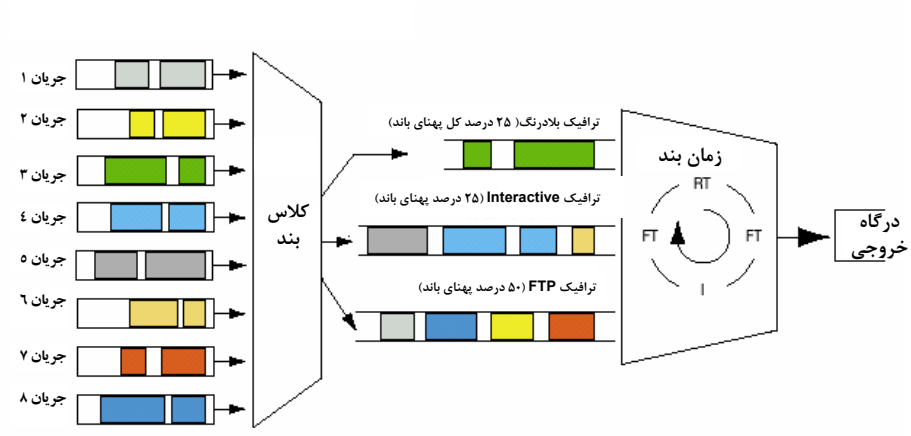
در صف‌بندی WRR، ارائه مقادیر مختلف پهنای‌باند به یکی از دو صورت زیر انجام می‌شود.

- در مدت یک چرخه سرویس، به صفهای با پهنای‌باند بیشتر اجازه ارسال بیشتر از یک بسته داده می‌شود.
- هر صف که در هر زمانی که ملاقات شد فقط یک بسته ارسال کند، اما در هرچرخه سرویس، صفهای با پهنای‌باند بزرگتر چندین بار ملاقات شوند.

در شکل(۱۳-۱۲)، ۲۵ درصد پهنای‌باند درگاه خروجی به صف ترافیک بلادرنگ، ۲۵ در صد به صف ترافیک interactive و۵۰ درصد به صف ترافیک انتقال داده، اختصاص یافته است. صف‌بندی WRR این تخصیص پهنای‌باند وزن‌دار شده را به این صورت پشتیبانی می‌کند که در هر چرخه سرویس، صف انتقال داده را ۲ بار ملاقات می‌کند.

---

[1] Round Robin
[2] Weighted Round Robin
[3] Class Based Queuing




**شکل(۱۳-۱۲): مثالی از صف بندی WRR**

برای تعدیل سازی میزان منابع تخصیص یافته به هرکلاس سرویس، می توان پارامترهایی رابرای کنترل رفتار هر صف تنظیم کرد:

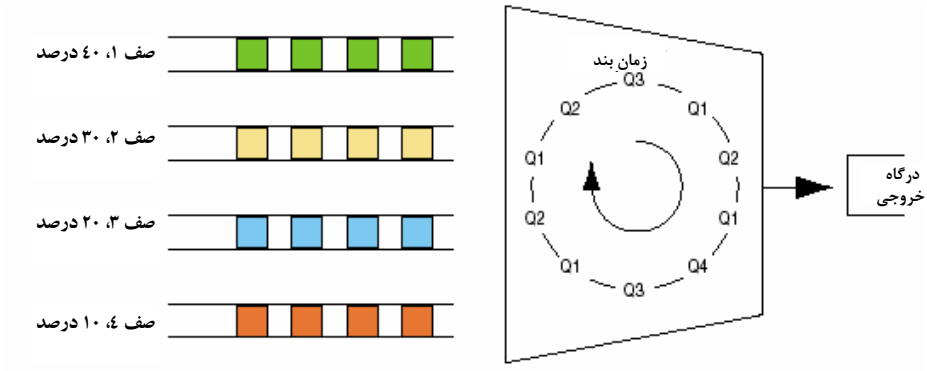
- مقدار تاخیر بسته ها درهرصف به وسیله پارامترهایی نظیر: نرخ ورود بسته ها به صف، عمق صف، مقدارترافیکی که درهرچرخه سرویس از صف برداشته می شود وتعداد کلاس های سرویس (صف) دیگر که در درگاه خروجی وجود دارد، تعیین می گردد.
- مقدار تغییرات تاخیربسته ها در هر صف بوسیله تغییرپذیری تاخیر درآن صف، تغییرپذیری تاخیر در تمام صف های دیگر و تغییرپذیری درفاصله زمانی بین هر چرخه سرویس دهی صف تعیین می شود.
- مقدار اتلاف بسته های هر صف توسط نرخ ورود بسته ها به صف، عمق صف، میزان تجاوز از پروفایل RED تنظیم شده برای آن صف ومیزان ترافیک برداشته شده از صف در هر چرخه سرویس دهی، تعیین می شود.

این صف‌بندی دارای مزایای زیر است:

- صف‌بندی WRR را می‌توان بصورت سخت‌افزاری انجام داد، بنابراین می‌توان آنرا در رابطهای سرعت‌بالا در هسته و لبه شبکه بکار برد.
- صف‌بندی WRR ، یک پهنای‌باندحداقل را برای تمام کلاسهای سرویس تضمین می‌کند.
- صف‌بندی WRR، مکانیزم مؤثری برای پشتیبانی از کلاسهای سرویس جدا شده می‌باشد.
- کلاس بندی ترافیک به کلاس های سرویس گوناگون نسبت به اولویت گذاری بسته ها، مدیریت منصفانه تر وپایداری فراهم می آورد. به عنوان مثال اگر به ترافیک های بلادرنگ بالاترین اولویت داده شود، دراینصورت ترافیک های بلادرنگ اضافی، باعث حذف ترافیک های کم اولویت ارسال فایل می شوند. به عبارت دیگر مکانیزم WRR به این امر اعتقاد دارد که کاهش منابع به مراتب بهتر ازحذف منابع قادر به کنترل ازدحام در شبکه می باشد.
- یکی از محدودیت های بزرگ مکانیزم WRR این است که این مکانیزم فقط درصورتیکه طول همه بسته ها یکسان باشد و یا مقدار میانگین طول بسته ها از قبل مشخص باشد، قادربه تخصیص میزان صحیح پهنای باندبه هرکلاس می باشد. به عنوان مثال فرض کنید که ازمکانیزم WRR برای سرویس دهی به ۴ صف ۱،۲،۳ و۴ به میزان ۴۰،۳۰،۲۰ و ۱۰ درصد پهنای باند استفاده می شود. فرض براین است که تمام بسته ها دارای طول ثابت ۱۰۰ بایت می باشند. این حالت درشکل(۱۳-۱۳) نشان داده شده است

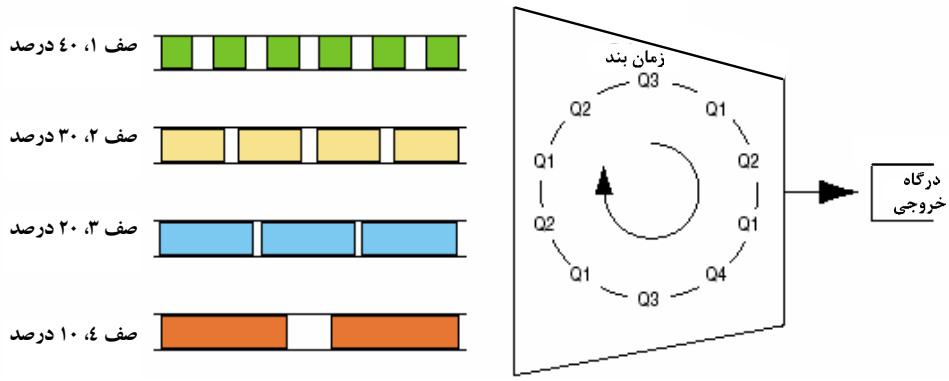
درپایان هرچرخه سرویس،صفهای ۱،۲،۳ و۴ به ترتیب اقدام به ارسال ۴ بسته (۴۰۰ بایت)،۳ بسته (۳۰۰ بایت)، ۲بسته (۲۰۰بایت) و ۱بسته (۱۰۰ بایت) می کنند. ازآنجائیکه دریک چرخه سرویس دهی کلاً ۱۰۰۰ بایت ارسال شده است، بنابراین صف های ۱،۲،۳ و۴ به ترتیب ۴۰،۳۰،۲۰ و۱۰ درصد از کل پهنای باند خروجی را به خود اختصاص داده اند.

ملاحظه می شودکه دراین مثال مکانیزم WRR به خوبی موفق به تخصیص پهنای باند بین صف های فوق شده است. بنابراین درحالتیکه طول بسته ها ثابت است ( مثل شبکه های ATM) مکانیزم WRR به خوبی منصفانه عمل می کند.




**شکل(۱۳-۱۳): مکانیزم WRR برای صف های با طول بسته های ثابت**

اما درحالتیکه یک کلاس سرویس از بسته های بزرگتری نسبت به کلاس سرویس دیگر بهره می برد، کلاس سرویسی که بسته های بزرگتری دارد، از پهنای باند بیشتری نسبت به آن مقداری که برای آن درنظرگرفته شده است، بهره می برد. به عنوان مثال فرض کنیددرمثال قبلی، طول متوسط بسته های صف های ۱،۲،۳و۴ به ترتیب برابر با ۱۰۰،۲۰۰،۳۰۰ و۴۰۰ بایت باشد. این حالت در شکل (۱۳-۱۴) نشان داده شده است.




**شکل (۱۳-۱۴): مکانیزم WRR برای صف های با طول بسته های متغیر**

دراین حالت درپایان یک چرخه سرویس، صفهای ۱،۲،۳و۴ به ترتیب اقدام به ارسال ۴ بسته (۴۰۰ بایت)،۳بسته (۶۰۰ بایت)، ۲بسته(۶۰۰ بایت) و ۱ بسته (۴۰۰ بایت) می نمایند. ازآنجائیکه دریک چرخه سرویس کلاً ۲۰۰۰بایت ارسال شده است، بنابراین صفهای ۱،۲،۳ و۴ به ترتیب۲۰،۳۰،۳۰ و۲۰ درصد کل پهنای باند را به خود اختصاص داده اند. بنابراین درحالتیکه طول بسته های ارسالی متغیراست، مکانیزم WRR قادربه تخصیص پهنای باند به میزان لازم که هنگام پیکره بندی مشخص می شود، نمی باشد.

از آنجائیکه می توان مکانیزم WRR را به صورت سخت افزاری پیاده سازی نمود، از آن می توان هم در مسیریاب های مرزی و هم درمسیریاب های مرکزی شبکه استفاده نمود. این مکانیزم به طور موثر محدودیت روش FQ را با زمانبندی کلاس های سرویس بانیازهای پهنای باند متفاوت، رفع می نماید. همچنین این مکانیزم مشکل اصلی روش PQ را نیز ازبین می برد. اما همانطور که اشاره شد، مشکل اصلی این روش عدم کارآیی لازم در حالتیکه طول بسته ها متغیراست، می باشد.

### ۱۳-۳-۶- زمانبندی DWRR[1]

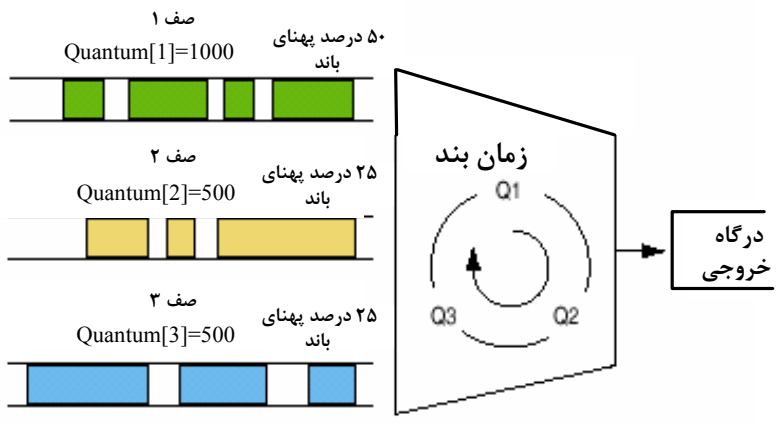
این مکانیزم ،زیربنای کلاسی از الگوهای صف‌بندی است که برای حل محدودیتهای مدلهای WRR و WFQ[2] طراحی شده اند ودارای مشخصات زیر است:

- DWRR، محدودیتهای WRR را با پشتیبانی دقیق توزیع منصفانه وزندار شده پهنای‌باند حل می‌کند، موقعیکه سرویس‌دهی صف‌ها با بسته‌های دارای طول متغیر انجام می‌شود از توزیع منصفانه وزن‌دار شده پهنای‌باند پشتیبانی می‌کند (محدودیت WRR).
- DWRR، محدودیتهای WFQ را با تعریف یک الگوریتم زمانبندی که دارای یک پیچیدگی محاسباتی کمتر است و می‌توان آنرا بصورت سخت‌افزاری بکار برد حل می‌کند.

در صف‌بندی DWRR هر صف با تعدادی از پارامترها شکل می‌گیرد که عبارتند از:

- weight = درصد پهنای‌باند درگاه خروجی که به صف تخصیص داده می‌شود.
- Deficitcounter = تعداد کل بایتهایی که صف اجازه دارد در هر زمانی که توسط زمانبند ملاقات می‌شود ارسال کند.
- Quantum سرویس، که متناسب با weight صف بوده و بر حسب بایت بیان می‌شود. هر بار که هر صف، با زمانبند ملاقات شد، deficitcounter آن به اندازه quantum افزایش می یابد. اگر $quantum[i] = 2 * quantum[x]$ باشد صف i، دو برابر پهنای‌باند صف x دریافت می‌کند (به شرطی که هر دو صف، فعال باشند).

در الگوریتم DWRR کلاسیک، زمانبند، هر صف ناخالی را ملافات کرده و تعداد بایتهای موجود در بسته سر صف را تعیین می‌کند. اگر اندازه بسته در سر صف، بزرگتر از متغیر deficitcounter بود متغیر deficitcounter به اندازه quantum افزوده می‌شود. سپس زمانبند برای سرویس‌دهی صف بعدی حرکت می‌کند. اگر اندازه بسته سر صف، کوچکتر یا مساوی متغیر deficitcounter بود، متغیر deficitcounter به اندازه تعداد بایتهای بسته، کم شده و بسته از درگاه خروجی ارسال می‌گردد. الگوریتم به تخلیه بسته‌ها از صف ادامه می‌دهد تا اینکه یا اندازه بسته سر صف، بزرگتر از اندازه deficitcounter شود یا صف خالی گردد. حال در صورت خالی بودن صف، متغیر deficitcounter صفر می‌شود. وقتیکه این مسئله اتفاق افتاد زمانبند برای سرویس‌دهی، به صف ناخالی بعدی حرکت می‌کند. درشکل(۱۳-۱۵)، مثالی از صف بندی DWRR آورده شده است.




**شکل(۱۳-۱۵): مثالی از مکانیزم DWRR**

[1] Deficit WRR
[2] Weighted Fair Queuing

در این قسمت به بررسی مثالی درزمینه پیاده سازی مکانیزم DWRR پرداخته شده است. البته باید توجه داشت که ممکن است که پیاده سازی تشریح شده دراین قسمت با آنچه که سازندگان مسیریاب ها درطرح خود استفاده می نمایند، متفاوت باشد.

به متغیر صف deficitcounter در ابتدا مقدار صفر داده می‌شود. در این مثال، صفها از 1 تا n شماره‌گذاری می‌شوند، که در آن n، حداکثر تعداد صفهای درگاه خروجی است.

```
for i=1 to n
        DeficitCounter[i]=0
END
```

تابع *Enqueue*(*i*)، بسته‌های تازه وارد را به صفشان هدایت کرده و قسمتی را با نام activelist مدیریت می‌کند. activelist برای جلوگیری از بررسی صفهای خالی استفاده می‌شود. activelist داری لیستی از صفهایی است که دارای حداقل یک بسته هستند. هرگاه که یک بسته، در صفی که قبلاً خالی بوده قرار بگیرد شاخصی برای آن صف به انتهای activelist ، توسط تابعی که Insertactivelist(i) نام دارد، اضافه می‌گردد. بطور مشابهی هرگاه که صفی خالی شود شاخص مربوط به آن صف، توسط تابع Removefromactivelist(i) از activelist حذف می‌گردد.

```
Enqueue(i)
        i=The index of the queue that will hold the new packet
        IF (ExistInActiveList[i]=FALSE) THEN
                InsertActiveList(I)
                DeficitCounter(I)=0
        ENDFOR
        Enqueue packet to Queue[I]
END Enqueue
```

هرگاه یک شاخص، در سر activelist باشد تابع Dequeue() به مقدار $Defictcounter[i] + Quantum[i]$ بایت ازصف، تخلیه می‌کند. اگر در انتهای چرخه سرویس، queue[i] هنوز بسته‌هایی برای ارسال داشته باشد تابع Insertactivelist(i) شاخص i را به انتهای activelist اضافه می‌کند. با این وجود اگر Queue[i] در انتهای چرخه سرویس، خالی باشد، deficitcounter[i] مساوی صفر شده و تابع Removefromactivelist(i) شاخص i را از activelist حذف می‌کند.

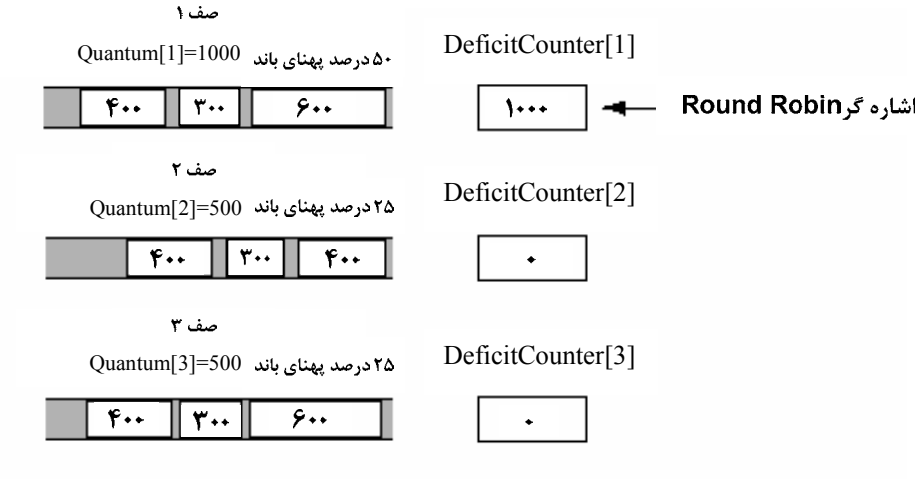
```
Dequeue()
        While (TRUE) DO
                IF(ActiveList is NotEmpty) THEN
                        i=The index at the head of the ActiveList
                        DeficitCounter[i]=DeficitCounter[i]+Quantum[i]
                        While(DeficitCounter[i]>0 and not empty(Queue[i])) DO
                                PacketSize=Size(head(Queue[i]))
                                IF (PacketSize <=DeficitCounter[i]) THEN
                                        Transmit Packet at head of Queue[i]
                                        DeficitCounter[i]=DeficitCounter[i]-PacketSize
                                ELSE
                                        Break
                                ENDIF
                        ENDWHILE
                        IF (Emoty(Queue[i]) THEN
                                DeficitCounter[i]=0
                                RemoveFromActiveList[i]
                        ELSE
                                InsertActiveList[i]
                        ENDIF
                ENDIF
```

```
        ENDWHILE
END Dequeue
```

استفاده از این کد، در یک الگوی زمانبندی صف DWRR یک شبکه، دارای محدودیتهای زیر است:

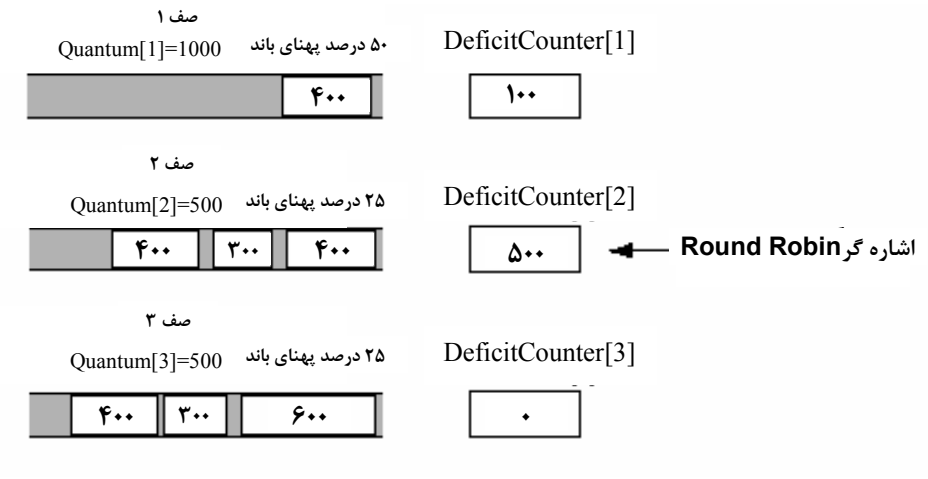
- ادامه سرویس‌دهی چندین بسته از یک صف تنها تا کمتر شدن deficitcounter[i] از اندازه بسته سر صف Queue[i]، می‌تواند منجر به تغییرات تاخیر شده و این مسئله باعث عدم استفاده از آن برای پشتیبانی ترافیک بلادرنگ شود.
- ناتوانی مدل برای پشتیبانی از یک deficitcounter[i] منفی می‌تواند به این معنی باشد که اگر Queue[i] دارای اعتبارات کافی برای انتقال یک بسته نباشد، ممکن است که پهنای‌باند ضعیفی به صف، تخصیص داده شود. چراکه تا زمان تجمع اعتبارکافی، به آن اجازه ارسال داده نمی‌شود. این مسئله، ممکن است تا چند چرخه سرویس طول بکشد.

برای این مثال، فرض کنید که دردرگاه خروجی که شامل سه صف۱،۲ و۳ است از مکانیزم زمانبندی DWRR فوق استفاده شود. فرض براین است که صف۱،۲ و۳ به ترتیب ۵۰،۲۵ و۲۵ درصدازکل پهنای باند درگاه خروجی رابه خوداختصاص داده است. درابتدا متغیرDeficitCounter به صفرمقداردهی می شود. فرض می شود که اشاره گرچرخه سرویس به صف۱ اشاره می کند که دربالای لیست ActiveList قراردارد. قبل ازآنکه زمانبند DWRR شروع به سرویس دهی صف۱کند، به مقدار Quantum[1]=1000 به DeficitCounter[1] اضافه می شود.درشکل(۱۳-۱۶) مثال فوق نشان داده شده است.



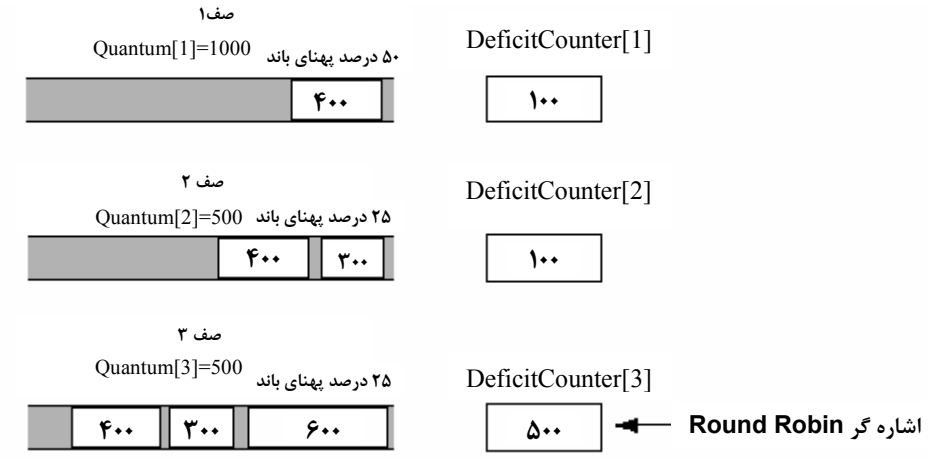

**شکل (۱۳-۱۶): مثال DWRR، چرخه اول، اشاره گربرابر۱ میباشد**

ازآنجائیکه بسته ۶۰۰ بایتی موجود درسرصف۱ از مقدار DeficitCounter[1]=1000 بیشتر است، بسته ۶۰۰بایتی ارسال می شود. این امر باعث می شود که مقدار DeficitCounter[1] به اندازه ۶۰۰بایت کم شود که مقدار جدید آن ۴۰۰ می شود. اکنون ازآنجائیکه بسته ۳۰۰ بایتی موجود درسرصف ازمقدار DeficitCounter[1]=400 کمتراست، بنابراین بسته ۳۰۰بایتی ارسال می شود ودراین حالت مقدار DeficitCounter[1] به اندازه ۳۰۰ کم می شود. ازآنجائیکه بسته ۴۰۰بایتی موجود درسرصف۱ ازمقدارDeficitCounter[1]=100 بزرگتراست، بسته ۴۰۰ بایتی قابل ارسال نمی باشد. این مسئله باعث می شود که اشاره گرچرخه به صف۲ که اکنون دربالای صف ActiveList قراردارد ، اشاره می کند. این مسئله درشکل(۱۳-۱۷) نشان داده شده است.



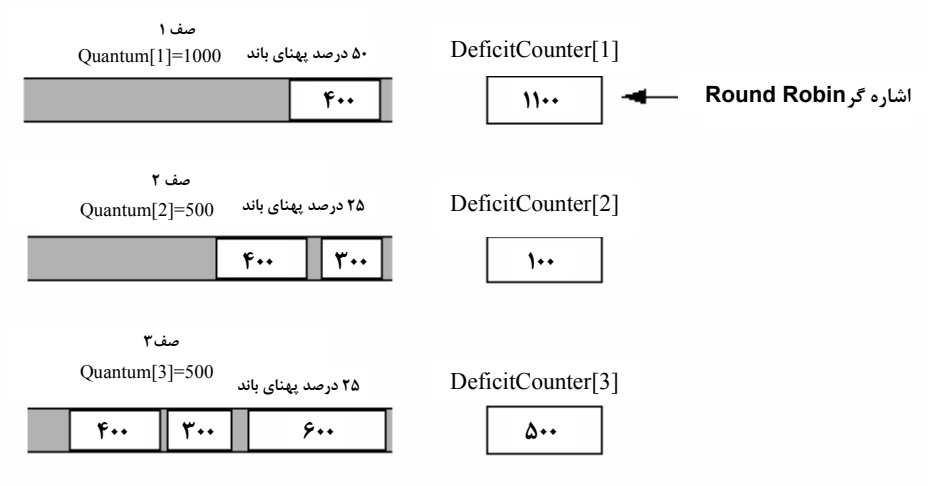

**شکل(۱۳-۱۷): مثال DWRR، اولین چرخه ، اشاره گرصف به صف۲ اشاره می کند**

قبل ازآنکه زمانبند DWRR شروع به سرویس دهی صف۲ کند، به مقدار Quantum[2]=500 به DeficitCounter[2] اضافه می شود. ازآنجائیکه بسته ۴۰۰ بایتی موجود درسرصف۲ از مقدار DeficitCounter[2]=500 کمتر است، بسته ۴۰۰بایتی ارسال می شود. این امر باعث می شود که مقدار DeficitCounter[2] به اندازه ۴۰۰بایت کم شود که مقدار جدید آن ۱۰۰ مقدارداده می شود. اکنون ازآنجائیکه بسته ۳۰۰ بایتی موجود درسرصف۲ ازمقدار DeficitCounter[2]=100 بیشتراست، بنابراین بسته ۳۰۰بایتی قابل ارسال نمی باشد. این مسئله باعث می شود که اشاره گرچرخه به صف۳ که اکنون دربالای صف ActiveList قراردارد ، اشاره می کند. این مسئله درشکل (۱۳-۱۸) نشان داده شده است.




**شکل(۱۳-۱۸): مثال DWRR، اولین چرخه ، اشاره گرصف به صف ۳ اشاره می کند**

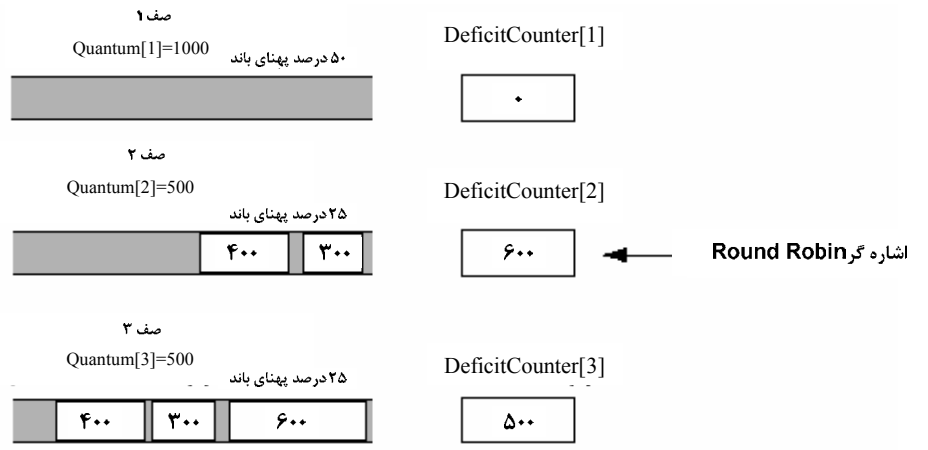
قبل ازآنکه زمانبند DWRR شروع به سرویس دهی صف۳ کند، به مقدار Quantum[3]=500 به DeficitCounter[3] اضافه می شود، که مقدارآن برابر با ۵۰۰ می شود. ازآنجائیکه بسته ۶۰۰ بایتی موجود درسرصف۳ از مقدار DeficitCounter[3]=500 بیشر است ، بنابراین بسته ۶۰۰ بایتی قابل ارسال نمی باشد. این مسئله باعث می شود که اشاره گرچرخه به صف۱ که اکنون دربالای صف ActiveList قراردارد ، اشاره می کند. این مسئله درشکل (۱۳-۱۹) نشان داده شده است.



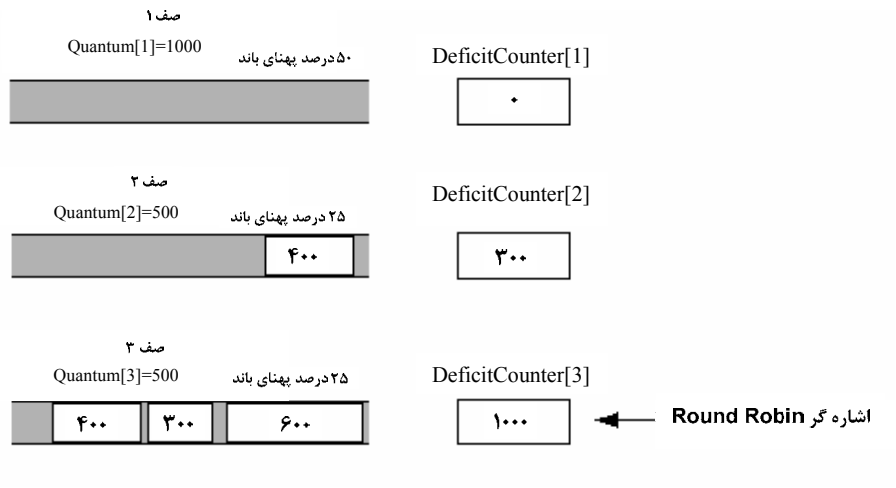

**شکل(۱۳-۱۹): مثال DWRR، اولین چرخه، اشاره گرصف به صف۱ اشاره می کند**

قبل ازآنکه زمانبند DWRR شروع به سرویس دهی صف۱ کند، به مقدار Quantum[1]=1000 به DeficitCounter[1] اضافه می شود که مقدارجدید DeficitCounter[1] برابر با ۱۱۰۰ میگردد.ازآنجائیکه بسته ۴۰۰ بایتی موجود درسرصف۱ از مقدار DeficitCounter[1]=1100 کمتر است، بسته ۴۰۰بایتی ارسال می شود. این امر باعث می شود که مقدار DeficitCounter[1] به اندازه ۴۰۰بایت کم شود که مقدار جدید آن ۷۰۰ مقدارداده می شود. اکنون صف۱ خالی است. این مسئله باعث می شودکه DeficitCounter[1]=0 گرددوصف۱ ازلیست ActiveList خارج می شود واشاره گرچرخه به صف ۲اشاره می نماید. درشکل (۱۳-۲۰) این حالت نشان داده شده است.

قبل ازآنکه زمانبند DWRR شروع به سرویس دهی صف۲ کند، به مقدار Quantum[2]=500 به DeficitCounter[2] اضافه می شود ومقدارجدید آن برابر با ۶۰۰ می گردد. ازآنجائیکه بسته ۳۰۰ بایتی موجود درسرصف۲ از مقدار DeficitCounter[2]=600 کمتر است، بسته ۳۰۰بایتی ارسال می شود. این امر باعث می شود که مقدار DeficitCounter[2] به اندازه ۳۰۰بایت کم شود که مقدار جدید آن ۳۰۰ مقدارداده می شود. اکنون ازآنجائیکه بسته ۴۰۰ بایتی موجود درسرصف۲ ازمقدار DeficitCounter[2]=300 بیشتراست، بنابراین بسته ۴۰۰بایتی فوق قابل ارسال نمی باشد. این مسئله باعث می شود که اشاره گرچرخه به صف۳ که اکنون دربالای صف ActiveList قراردارد ، اشاره کند. این مسئله درشکل (۱۳-۲۱) نشان داده شده است.

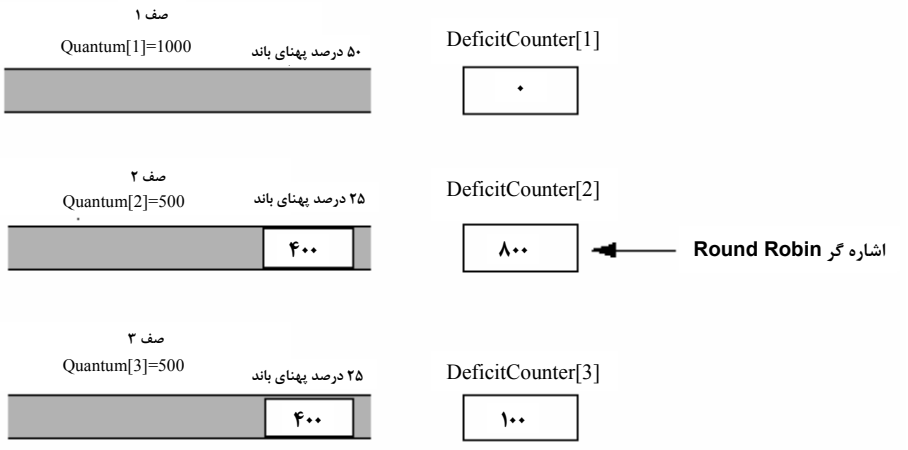



**شکل(۱۳-۲۰): مثال DWRR، چرخه دوم، اشاره گرصف به صف۲ اشاره می کند**




**شکل(۱۳-۲۱): مثال DWRR، چرخه دوم، اشاره گرصف به صف ۳ اشاره می کند**

قبل ازآنکه زمانبند DWRR شروع به سرویس دهی صف۳ کند، به مقدار Quantum[3]=500 به DeficitCounter[3] اضافه می شود، که مقدارآن برابر با ۱۰۰۰ می شود. ازآنجائیکه بسته ۶۰۰ بایتی موجود درسرصف۳ از مقدار DeficitCounter[3]=1000 کمتر است ، بنابراین بسته ۶۰۰ بایتی ارسال می شود و ازمقدارDeficitCounter[3] به اندازه ۶۰۰ بایت کم می شود و مقدار جدید آن برابر با ۴۰۰ میگردد. ازآنجائیکه بسته ۳۰۰ بایتی موجود درسرصف۳ از مقدار DeficitCounter[3]=400 کمتر است بسته ۳۰۰ بایتی فوق نیز ارسال می گرددواز مقدارDeficitCounter[3] به اندازه ۳۰۰ بایت کم شده و مقدارجدید آن برابر با ۱۰۰ می گردد. ازآنجائیکه بسته ۴۰۰بایتی موجود درسرصف ۳ از مقدار DeficitCounter[3]=100 بیشتر است، بنابراین بسته ۴۰۰بایتی قابل ارسال نمی باشد. این مسئله باعث می شود که اشاره گرچرخه به صف۲ که اکنون دربالای صف ActiveList قراردارد ، اشاره می کند. این مسئله درشکل(۱۳-۲۲) نشان داده شده است.




**شکل(۱۳-۲۲): مثال DWRR، چرخه سوم، اشاره گرصف به صف ۲ اشاره می کند**

قبل ازآنکه زمانبند DWRR شروع به سرویس دهی صف۲ کند، به مقدار Quantum[2]=500 به DeficitCounter[2] اضافه می شود ومقدارجدید آن برابر با ۸۰۰ می گردد. دراین نقطه مکانیزم زمانبند DWRR به سرویس دهی صف ها براساس پهنای باند تخصیص یافته برای هرکلاس سرویس، ادامه می دهد.

مکانیزم DWRR دارای مزایای زیر می باشد:

- محافظت جریان های ترافیکی: چنانچه یک کلاس سرویس ضعیف دریک صف وجود داشته باشد، هیچ تاثیر منفی برکارآیی سایر کلاس های سرویس صف های دیگر نمی گذارد.
- حل مشکل WRR درسرویس دهی به بسته های به طول متغیر
- حل مشکل مکانیزم PQ با این تضمین که تمام کلاس های سرویس به حداقل پهنای باند تخصیص یافته به آنها دسترسی پیدا کنند.
- پیاده سازی نسبتاً ساده

### ۱۳-۳-۷- زمانبندی GPS[1]

GPS یک مدل ایده‌آل work conserving است. در این الگوریتم بسته‌ها متعلق به هر جریان در صفوف منطقی متفاوت نگهداری می‌شوند. GPS مقدار بسیار کوچکی از داده هر صف را به گونه‌ای سرویس می‌دهد که در یک بازه زمانی محدود، تمامی صف‌های غیرخالی را ملاقات می‌نماید. هر صف می‌تواند دارای وزنی باشد و صفوف را می‌توان معادل وزن آنها سرویس داد. اگر فرض کنیم که k جریان فعال وجود دارند. آنگاه هر جریان 1/k سهم از منابع دریافت می‌کند. اگر صفی خالی باشد منبع استفاده نشده مربوط به آن صف بین جریانهای موجود بصورت مساوی تقسیم می‌شود.

فرض کنید که k جریان با وزنهای w(1) ، w(2) ، .. ، w(k) وجود دارند که باید توسط یک سرویس‌دهنده از طریق GPS سرویس داده شوند. نرخ سرویس i امین جریان در بازه زمانی $[t, \tau]$ مساوی است با $R(i,\tau,t)$. برای هر جریان i و j در بازه $[t, \tau]$ معادله زیر برقرار است:

$$\frac{R(i,\tau,t)}{R(j,\tau,t)} \geq \frac{w(i)}{w(j)}$$

از آنجا که GPS دارای خواص عدالت ایده‌آل و جداسازی کامل جریانها می‌باشد، تحقیقات زیادی بر روی آن صورت گرفته است. البته GPS قابل پیاده‌سازی نیست. چرا که سرویس‌دهی مقدار بسیار کوچک داده از هر صف، غیرممکن است.

با این وجود می‌توان عملکرد GPS را مشابه عملکرد روش زمانبندی چرخشی بیت به بیت وزن‌دار تصور نمود. در یک زمانبند چرخشی بیت به بیت، بیتهای مجزا از بسته‌های سر هر صف به فرم WRR انتقال می‌یابند. این شیوه تخصیص عادلانه پهنای باند را پشتیبانی می‌کند. چرا که طول بسته، در نظر گرفته می‌شود. در نتیجه در هر لحظه از زمان، هر صف سهم تعیین شده خود از پهنای باند را دریافت می‌کند. با وجود اینکه انتقال بسته‌ها بصورت یک بیت در هر لحظه از طریق یک شبکه TDM امکان‌پذیر است؛ اما نمی‌توان این انتقال را در شبکه‌های مالتی‌پلکس شده آماری پشتیبانی نمود. البته اگر تصور کنیم که یک سرجمع کننده بسته در انتهای اتصال وجود دارد، ترتیبی که هر بسته نهایتا سرهم می‌شود از طریق انتقال آخرین بیت هر بسته مشخص می‌گردد.

### ۱۳-۳-۸- زمانبندی VC[2]

این الگوریتم در سال ۱۹۹۰ توسط L.Zhang ارائه گردید.

الگوریتم به هر بسته، یک زمان انتقال مجازی نسبت می‌دهد که بیان کننده زمانی است که تا آن زمان بسته باید انتقال یابد. بسته‌ها به ترتیب صعودی برچسب زمانی سرویس می‌گیرند. زمان مجازی، بر پایه نرخ ورود بسته‌های قبلی (که توسط سیستم اندازه‌گیری شده) و متوسط نرخ ورودی که توسط کاربر تعیین شده است، محاسبه می‌شود. یک برچسب زمانی

[1] Generalized Processor Sharing
[2] Virtual Clock

$TS_i^k$ به k امین بسته اتصال i نسبت داده می‌شود. اگر $P_i$ نرخ رزرو شده برای اتصال $i$ و $AT$ زمان واقعی ورود بسته با طول $L_i^k$ به سوئیچ باشد، آنگاه برچسب زمانی مربوط به آن بسته بصورت زیر مشخص می‌شود:

$$TS_i^k \leftarrow \max(AT, TS_i^{k-1}) + \frac{L_i^k}{P_i}$$

مشکل این روش، عدم رعایت عدالت در توزیع پهنای باند و عدم تضمین محدوده‌های تأخیر می‌باشد.

### ۱۳-۳-۹- زمانبندی Stop and Go

در این روش، که در سال ۱۹۹۰ توسط S. J. Golestani ارائه گردیده است ، محور زمان هر دو اتصال ورودی و خروجی به بازه‌های زمانی با طول ثابت T تقسیم می‌شود. این بازه زمانی یک چارچوب نامیده می‌شود. بسته‌هایی که در طول یک چارچوب به اتصال ورودی داده می‌شوند، آمادگی انتقال در چارچوب بعدی را دارا هستند. در داخل یک چارچوب، بسته‌های آماده را می‌توان به هر ترتیبی انتقال داد. در نتیجه یک تأخیر ثابت $\theta$ برای هر بسته وجود دارد.

از آنجا که انتقال هر بسته که در یک چارچوب وارد می‌شود تا چارچوب بعدی به تأخیر می‌افتد، الگوریتم، non work conserving می‌باشد. اگر حداکثر تعداد بسته‌ای که در طول یک چارچوب وارد شده بیشتر از مکانهای رزرو شده در یک چارچوب نباشد، الگوریتم قادر است تأخیر و واریانس تأخیر محدود را تأمین نماید. مشکل اساسی این الگوریتم مربوط به دقت در تخصیص پهنای باند است. تأخیر $\theta$ معادل طول چارچوب است و در نتیجه چارچوب کوچک، ارجح است اما از طرف دیگر به منظور تخصیص پهنای باند دقیق‌تر، طول چارچوب باید بزرگ باشد.

### ۱۳-۳-۱۰- زمابندی Delay-EDD[1]

این الگوریتم در سال ۱۹۹۰ توسط D.Ferrari و D.Verma ارائه گردید. ایده اصلی الگوریتم این است که اگر همه بسته‌ها با نرخی ثابت وارد شوند، این تضمین برای آنها وجود دارد که همگی، سرویسگر را در یک زمان محدود ترک کنند. این زمان، پارامتر تأخیری است که به اتصال تخصیص داده می‌شود.

به طور خاص، یک متغیر حالت به نام Expected Deadline($ExD_i$) به هر ارتباط i نسبت داده می‌شود. اگر مینیمم زمان بین ورود بسته‌ها $X\min_i$ و محدوده تأخیر داخلی یک ارتباط $d_i$ باشد، ضرب‌الاجل مورد انتظار بصورت زیر محاسبه می‌شود:

$$ExD_i \leftarrow \max(AT + d_i, ExD_i + X\min_i)$$

که $AT$ زمان واقعی ورود بسته است. بر اساس این تعریف، تمامی بسته‌ها با طول یکسان فرض شده‌اند. متغیر $X\min_i$ بیان کننده پهنای باند تخصیص یافته به ارتباط i است. اگر همه بسته‌ها دارای طول یکسان L باشند و همگی با مینیمم بازه زمانی $X\min_i$ وارد شوند، آنگاه پهنای باند تخصیص یافته به ارتباط، $\frac{L}{X\min_i}$ است.

### ۱۳-۳-۱۱- زمانبندی WFQ

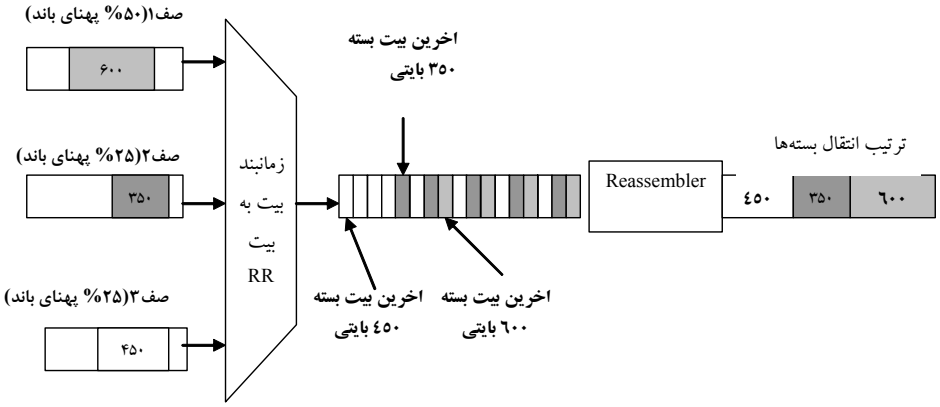
این روش در سال ۱۹۸۹ ارائه گردید. WFQ اساس کلاسی از الگوریتمهای زمانبندی است که به منظور رفع محدودیتهای FQ طراحی شده‌اند. WFQ به هر صف وزنی می‌دهد که مشخص کننده درصد متفاوتی از پهنای باند درگاه خروجی است که به آن تخصیص می‌یابد. از این طریق WFQ قادر است جریانهایی با نیازهای مختلف پهنای باند را پشتیبانی نماید.

---

[1] Delay- Earliest Due Date

WFQ همچنین بسته‌های با طول متغیر را نیز پشتیبانی می‌کند؛ تا جریانهایی با بسته‌های بزرگتر، سهم پهنای باند بیشتری از جریانهای با بسته‌های کوچکتر دریافت نکنند. پشتیبانی از تخصیص عادلانه پهنای باند، حین ارسال بسته‌های با طول متغیر به نحو قابل ملاحظه‌ای به پیچیدگی محاسباتی مکانیزم زمانبندی صف می‌افزاید. دلیل اصلی اینکه پیاده‌سازی مکانیزم زمانبندی صف در شبکه‌های ATM بر پایه سلولهای با طول ثابت بسیار ساده‌تر از پیاده‌سازی آنها در شبکه‌های IP بر پایه بسته با طول متغیر می‌باشد نیز همین امر است.

شکل(۱۳-۲۳) ، یک زمانبند چرخشی بیت به بیت وزن‌دار را نشان می‌دهد که در حال سرویس‌دهی به سه صف است. فرض کنید که صف یک ۵۰٪ پهنای باند و صفهای دو و سه هر یک ۲۵٪ پهنای باند را دریافت می‌کنند. زمانبند، دو بیت از صف یک و یک بیت از صف دو و یک بیت از صف سه انتقال داده و سپس مجدد به صف یک بر می‌گردد. در نتیجه این خط‌مشی زمانبندی، آخرین بیت بسته ۶۰۰ بایتی قبل از آخرین بیت بسته ۳۵۰ بایتی انتقال می‌یابد و آخرین بیت بسته ۳۵۰ بایتی قبل از آخرین بیت بسته ۴۵۰ بایتی انتقال می‌یابد. این امر موجب می‌شود سرویس بسته ۶۰۰ بایتی زودتر از بسته ۳۵۰ بایتی و بسته ۳۵۰ بایتی زودتر از بسته ۴۵۰ بایتی پایان پذیرد.




**شکل(۱۳-۲۳): زمانبند چرخشی بیت به بیت دوار**

### ۱۳-۳-۱۱-۱- الگوریتم WFQ

WFQ با تقریب زدن و شبیه‌سازی سیستم GPS، توزیع عادلانه پهنای باند را برای بسته‌های با طول متغیر پشتیبانی می‌کند. این تقریب با محاسبه و انتساب یک" زمان پایان" به هر بسته صورت می‌گیرد. این زمان پایان، زمانی است که سرویس‌دهی به آن بسته در سیستم GPS معادل، پایان می‌پذیرد. به عبارت دیگر، WFQ، GPS را شبیه‌سازی کرده و نتیجه این شبیه‌سازی را به منظور تعیین ترتیب سرویس بسته‌ها بکار می‌برد.

در سال ۱۹۹۳ روش دیگری به نام PGPS ارائه گردید که نحوه زمانبندی یکسانی با WFQ دارد. در این الگوریتم برای محاسبه زمان پایان بسته متغیرها و تعاریف زیر باید مورد نظر قرار گیرند:

با فرض:

$w(i)$ : وزن اتصال $i$ ام

$F(i,k,t)$ : زمان پایان بسته $k$ ام که در زمان $t$ به صف $i$ وارد می‌شود.

$P(i,k,t)$ : طول بسته $k$ ام که در زمان $t$ به صف $i$ وارد می‌شود.

$V(t)$ : زمان مجازی که به ازای وقوع هر رویداد (مانند ورود و خروج بسته‌ها) به‌روز می‌شود.

نحوه به‌روز رسانی $V(t)$ بصورت زیر است:

$$V(0)=0, V(t+\tau)=V(t)+\frac{\tau}{\sum_{i\in B_i} w(i)}$$

که $B_i$ مجموعه جریانهای فعال موجود در زمان t است.

منظور از جریان فعال، جریانی است که ماکزیمم زمان پایان بسته‌های موجود در صف آن و یا آخرین بسته‌ای که از آن صف به منظور سرویس‌گیری خارج شده بزرگتر از زمان مجازی فعلی باشد.

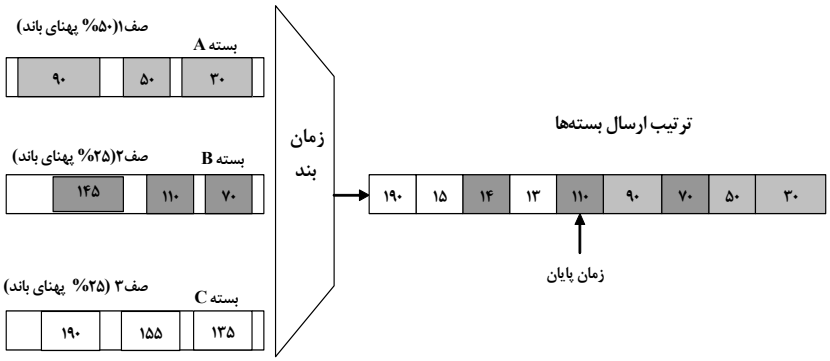
زمان مجازی برای تمامی زمانهایی که سرویس‌دهنده بیکار است، برابر صفر می‌باشد.

با این فرضیات، زمان پایان بسته $i$ بصورت زیر محاسبه می‌شود:

$$F(i,k,t)=\max\{F(i,k-1,t),V(t)\}+\frac{P(i,k,t)}{w(i)}$$

با ورود هر بسته، زمان پایان آن طبق فرمول فوق محاسبه شده و به آن نسبت داده می‌شود. سپس زمانبند از میان تمامی بسته‌های موجود در صف‌ها، بسته‌ای را که دارای کمترین زمان پایان صف است، انتخاب و ارسال می‌کند. نکته‌ای که باید مورد توجه قرار گیرد این است که زمان پایان، زمان واقعی انتقال بسته نیست. بلکه عددی است که به هر بسته نسبت داده می‌شود و نشاندهنده ترتیب ارسال بسته‌ها در درگاه خروجی می‌باشد.

در شکل(۱۳-۲۴)، مثالی از الگوریتم WFQ آورده شده است. همانگونه که در این شکل مشخص است، پس از کلاسه‌بندی و قرار گرفتن یک بسته در صف مربوط به خود، زمان پایان بسته محاسبه و به آن نسبت داده می‌شود. سپس زمانبندWFQ بسته با کوچکترین (زودترین) زمان پایان را به عنوان بسته بعدی برای انتقال به درگاه خروجی انتخاب می‌کند. به عنوان مثال اگر WFQ مشخص نماید که بسته A دارای زمان پایان ۳۰ و بسته B دارای زمان پایان ۷۰ و بسته C دارای زمان پایان ۱۳۵ باشد، آنگاه بسته A قبل از B یا C انتقال می‌یابد. تخصیص وزن مناسب به صفها به زمانبند WFQ امکان می‌دهد که دو یا چند بسته متوالی از یک صف را انتقال دهد.




**شکل(۱۳-۲۴):مثالی از زمانبند WFQ ( ارسال بسته‌ها بر اساس زمان پایان بسته)**

### ۱۳-۳-۱۱-۲- مزایا و محدودیتهای WFQ

WFQ دارای دو مزیت اصلی است:

- WFQ با تضمین یک سطح حداقل از پهنای باند برای هر کلاس سرویس، مستقل از رفتار سایر کلاسها، حفاظی مطمئن برای هر کلاس سرویس ایجاد می‌کند.
- در صورت ترکیب با نظارت کننده ترافیک در لبه‌های شبکه، WFQ قادر است سهمی عادلانه و وزن‌دار و با تأخیری محدود را از پهنای باند درگاه خروجی، برای هر کلاس سرویس تضمین نماید.

اما WFQ دارای محدودیتهای زیر نیز می‌باشد:

- پیاده‌سازی WFQ در محصولات بصورت نرم‌افزاری انجام شده و نه سخت‌افزاری. این امر کاربرد WFQ را برای واسط‌های با سرعت پایین در لبه‌های شبکه محدود می‌کند.
- جریانهای بدرفتار در یک کلاس سرویس می‌توانند بر روی کارایی سایر جریانهای موجود در آن کلاس تاثیر داشته باشند.
- WFQ الگوریتمی پیچیده را پیاده‌سازی می‌کند که نیاز به نگهداری میزان قابل توجهی اطلاعات وضعیت برای هر کلاس سرویس و بررسی مرتب اطلاعات به ازای هر ورود و خروج بسته دارد.
- پیچیدگی محاسباتی، هنگامی که نیاز به پشتیبانی تعداد زیادی کلاس سرویس است بر روی قیاس‌پذیری WFQ آن تاثیر منفی می‌گذارد.
- و نهایتا با وجود اینکه محدوده تأخیر پشتیبانی شده توسط WFQ بهتر از زمانبندی‌های دیگر است، اما این محدوده هنوز هم بزرگ می‌باشد.

### ۱۳-۳-۱۱-۳- توسعه‌های WFQ

از آنجا که WFQ در ابتدا در سال ۱۹۸۹ ارائه شده است، توسعه‌های زیادی بر آن ایجاد شده‌اند که هر یک سعی در ایجاد تعادل بین پیچیدگی، دقت و کارایی آن دارند. برخی از این توسعه‌ها درادامه توضیح داده شده اند.

### ۱۳-۳-۱۲- زمانبندی SCFQ[1]

همانگونه که بیان شد، یکی از مشکلات اصلی WFQ پیچیدگی و هزینه محاسبه زمان مجازی به ازای ورود هر بسته است. SCFQ که در سال ۱۹۹۴ توسط Golestani ارائه شد، راه حلی بود برای سرعت بخشیدن به محاسبه زمان پایان بسته‌ها. در این روش، به منظور محاسبه زمان پایان بسته، بجای استفاده از زمان مجازی، زمان پایان بسته‌ای که هم اکنون در حال سرویس‌گیری است بکار می‌رود. به عبارت دیگر، زمان پایان بسته i بصورت زیر محاسبه می‌شود:

$$F(i,k,t) = \max\{F(i,k-1,t), CF\} + \frac{P(i,k,t)}{w(i)}$$

زمان پایان یک بسته عبارتست از ماکزیمم $CF$ (زمان پایان بسته‌ای که هم اکنون در حال سرویس‌گیری است) و زمان پایان آخرین بسته موجود در صف، به اضافه زمانی که برای پایان سرویس بسته لازم است.

با وجود اینکه SCFQ پیچیدگی محاسباتی WFQ را ندارد، در بازه‌های زمانی کوتاه غیر عادلانه بوده و سطح تفکیک بین جریانهای مختلف در آن پایین‌تر می‌باشد. به علاوه این روش دارای محدوده تأخیر بیشتری نسبت به WFQ می‌باشد.

### ۱۳-۳-۱۳- زمانبندی $WF^2Q$[2]

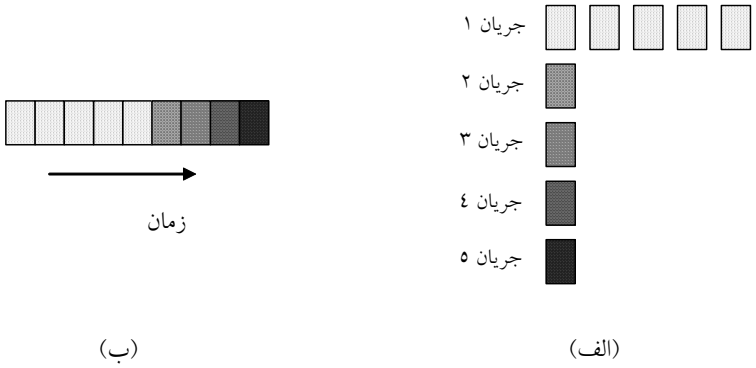
همانگونه که گفته شد، هدف WFQ شبیه‌سازی و ایجاد رفتاری تخمینی از عملکرد ایده‌آل GPS است. در WFQ برای انتخاب بسته بعدی به منظور زمانبندی، از میان کلیه بسته‌های موجود بسته با کوچکترین زمان پایان انتخاب می‌شود. این نحوه عملکرد گاهی اوقات کاملا متفاوت از GPS است. برای فهم بهتر به مثال زیر توجه کنید.

در شکل(۱۳-۲۵) (الف)، ۵ جریان از یک اتصال بصورت مشترک استفاده می‌کنند. همه بسته‌ها در زمان صفر وارد شده و همگی دارای اندازه ۱ واحد هستند. در این مثال جریان شماره یک، ۵ بسته پشت سرهم را با شروع از زمان صفر ارسال می‌کند. در حالیکه ۴ جریان دیگر فقط یک بسته در زمان صفر می‌فرستند. همانطور که در شکل(۱۳-۲۵) (ب) نشان داده

[1] Self Clocked Fair Queueing

[2] worst-case Fair Weighted Fair Queueing

شده است، اگر سرویس‌گر WFQ باشد، ابتدا ۵ بسته جریان اول پشت سرهم ارسال می‌شوند و سپس نوبت به بسته‌های جریانهای بعد می‌رسد.



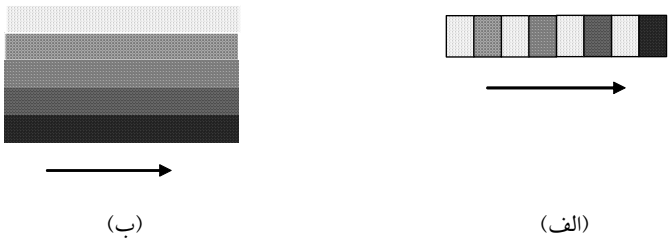

**شکل(۱۳-۲۵): (الف) ترتیب ورود بسته‌ها، (ب)ترتیب سرویس دهی در WFQ**

این نحوه عملکرد ممکن است منجر به بروز ازدحام در شبکه شود و با رفتار ایده‌آل GPS که در شکل (۱۳-۲۶) (ب) نمایش داده شده است تفاوت دارد.

الگوریتم $WF^2Q$ که در سال ۱۹۹۶ توسط Bennett J. و Zhang H. ارائه گردید، سعی در رفع این مشکل و ارائه تخمینی دقیق‌تر برای GPS دارد. در این الگوریتم، هنگامی که بسته‌ای برای سرویس‌گیری در زمان t انتخاب می‌شود، بجای انتخاب بسته از بین همه بسته‌های موجود در سرویسگر به مانند WFQ ، سرویسگر تنها بسته‌هایی را در نظر می‌گیرد که در سیستم GPS مربوط، در زمان t سرویس‌گیری خود را آغاز کرده باشند. به عبارت دیگر $\left\{P_i^k \mid b_{i,GPS}^k < \tau \le b_{i,WFQ}^k\right\}$ که $b_{i,s}^k$ زمان شروع سرویس بسته k ام از جریان i تحت سرویسگر s است و آنگاه از میان این بسته‌ها، بسته با کوچکترین زمان پایان را برای سرویس‌دهی انتقال می‌دهد.

بر اساس این عملکرد، ترتیب سرویس‌گیری بسته‌ها در مثال فوق تحت سرویسگر $WF^2Q$ بصورت شکل(۱۳-۲۶) (الف) است.

$WF^2Q$ تخمین بهتری برای GPS می‌باشد. این روش، سرویس تقریبا یکسانی با GPS فراهم می‌کند که حداکثر تفاوت آن با GPS بیشتر از طول یک بسته نیست. مشکل این روش پیچیدگی زمانی آن برای محاسبه زمان مجازی است. این پیچیدگی بیشتر از پیچیدگی زمانی WFQ می‌باشد. به منظور کاهش پیچیدگی زمانی این الگوریتم توسعه دیگری به نام $WF^2Q+$ توسط ارائه‌دهندگان $WF^2Q$ انجام شد که در آن روشی جدید برای محاسبه زمان مجازی ارائه گردیده است که منجر به پیچیدگی کمتر و دقت بیشتر می‌گردد.




**شکل(۱۳-۲۶): (الف) ترتیب سرویس گیری در $WF^2Q$، (ب) ترتیب سرویس گیری در GPS**

### ۱۳-۳-۱۴- زمانبندی SFQ[1]

یکی دیگر از توسعه‌های زمانبند WFQ که در سال ۱۹۹۶ توسط Goyal et .al ارائه گردید، SFQ نام دارد. در این الگوریتم به هنگام ورود هر بسته دو برچسب، برچسب شروع و برچسب پایان به آن مربوط می‌شوند. البته برخلاف WFQ، SFQ بسته‌ها به ترتیب صعودی برچسبهای شروع زمانبندی می‌شوند. به علاوه $V(t)$ (زمان مجازی) بصورت برچسب شروع بسته‌ای که در زمان t در حال سرویس‌گیری است تعریف می‌شود. الگوریتم این روش بصورت زیر است:
با ورود هر بسته، برچسب شروع بسته بصورت زیر محاسبه می‌شود:

$$S(i,k,t) = \max\{V(t), F(i,k-1,t)\}$$

که $F(i,k,t)$ برچسب پایان بسته عبارتست از:

$$F(i,k,t) = S(i,k,t) + \frac{P(i,k,t)}{w(i)}$$

زمان مجازی در ابتدا صفر است. در طول یک دوره زمانی مشغول بودن سرویس‌گر، زمان مجازی سرویس‌گر در لحظه t مساوی است با برچسب شروع بسته‌ای که در لحظه t سرویس می‌گیرد. در پایان دوره زمانی مشغول بودن، زمان مجازی برابر ماکزیمم برچسبهای پایان تمامی بسته‌هایی که تا لحظه t سرویس گرفته‌اند می‌گردد. بسته‌ها به ترتیب صعودی برچسب شروع خود سرویس می‌گیرند.
همانگونه که در الگوریتم مشخص است، محاسبه $V(t)$ در SFQ ساده‌تر و کم هزینه‌تر است. چرا که این محاسبه تنها نیاز به بررسی برچسب شروع بسته در حال سرویس‌گیری دارد. در نتیجه پیچیدگی محاسباتی SFQ مانند SCFQ یعنی $O(\log Q)$ در هر بسته است که Q تعداد جریانهای موجود در یک سرویس‌گر می‌باشد. به علاوه تضمین عدالت در این روش بهتر از WFQ بوده و متوسط تأخیر آن نیز پایین‌تر از WFQ می‌باشد.

### ۱۳-۳-۱۵- زمانبندی LFVC[2]

این الگوریتم در سال ۱۹۹۷ توسط S.Suri و G.Varghese ارائه گردید. اساس این الگوریتم مانند روش VC است. با این تغییر که این الگوریتم عدالت را در بهره‌وری تضمین می‌کند بدون اینکه از محدوده‌های تأخیر تجاوز کند. در این الگوریتم اگر جریانی از سهم عادلانه خود از پهنای باند تجاوز کند، آن جریان به عنوان متخلف شناخته شده و تمامی بسته‌های مربوط به آن جریان در ناحیه‌ای به نام L نگهداری می‌شوند. بقیه جریانهای خوش‌رفتار در یک صف اولویت H نگهداری می‌شوند. ساختار داده H و L را می‌توان دو صف اولویت در نظر گرفت. سرویسگر، از میان بسته‌های موجود در H ، بسته با کوچکترین برچسب زمانی را به منظور سرویس‌دهی انتخاب می‌کند. ممکن است که کوچکترین برچسب زمانی در H بزرگتر از کوچکترین برچسب زمانی موجود در L باشد. با این وجود باز هم ابتدا بسته‌های موجود در H ارسال می‌شوند. پس از گذشت مدت زمانی، باید یکی از جریانهای فعال را از L به H منتقل نمود. چرا که تا حد امکان نباید بسته‌ای دیرتر از برچسب زمانی خود سرویس بگیرد، به عبارت دیگر این انتقال باید پیش از بهم خوردن شرط تأخیر زیر صورت گیرد:

$$T(p) - AT \geq \Delta_f$$

---

[1] Start-time Fair Queueing
[2] Leap Forward Virtual Clock

$T(p)$ برچسب زمانی بسته $p$ است، که $p$، بسته سر صف جریان f می‌باشد. $AT$ زمان واقعی و $\Delta_f = \frac{l_f^{\max}}{r_f}$ زمان لازم برای ارسال بزرگترین بسته جریان با نرخ تضمین شده آن جریان است.( $l_f^{\max}$ اندازه بزرگترین بسته جریان $f$ و $r_f$ نرخ جریان $f$ می‌باشد.)

ممکن است در شرایطی تمامی جریانها، متخلف شناخته شده و در نتیجه H خالی بماند. در این حالت به منظور work conserve ماندن، باید بسته‌ای ارسال شود. بدین منظور، ساعت سرویسگر را تا حد امکان و تا جایی که شرط تأخیر هیچ یک از جریانهای موجود در L بهم نخورد، جلو برده می‌شود. در این زمان، حداقل یکی از جریانهای فعال L برای انتقال به H واجد شرایط می‌گردد.

می‌توان ساعت سرویسگر را تا مقدار مینیمم $(t_f - \Delta_f)$ بین کلیه جریانهای فعال $f$ موجود در L جلو برد که $t_f$ برچسب فعلی جریان $f$ است و بصورت زیر محاسبه می‌شود:

$$t_f = \begin{cases} T(P_f) & if\ f\ is\ active \\ \max(t_f^{prev}, AT) & otherwise \end{cases}$$

که $P_f$ بسته سر صف جریان f است. در نتیجه بهتر است جریانهای موجود در L را بر اساس اولویت $(t_f - \Delta_f)$ سازماندهی نمود (بجای اینکه همانند H بر اساس $t_f$ در صف قرار گیرند). این امر امکان می‌دهد که برای یافتن جریان منتخب برای انتقال به H، تنها جریانهای سر صف L را چک نمود.

### ۱۳-۳-۱۶- زمانبندی CBWFQ[1]

در این روش، کاربر کلاسهای ترافیکی را مشخص کرده و سپس ویژگیهای هر کلاس را تعیین می‌نماید. از جمله این ویژگیها، پهنای باند، وزن حداکثر طول بسته و حداکثر طول صف برای آن کلاس است. بسته‌های متعلق به هر کلاس محدود به پهنای باند و محدودیت طول صف مربوط به آن کلاس می‌باشند. پس از آنکه بسته‌ها در صفهای مربوطه جای گرفتند، بر اساس وزنی که به صفهای آنها تخصیص داده شده است، بر اساس الگوریتم WFQ سرویس خواهند گرفت.

CBWFQ این امکان را فراهم می‌آورد که بتوان میزان دقیق پهنای باندی که باید به هر کلاس تخصیص یابد را مشخص نمود.

### ۱۳-۳-۱۷- زمانبندی CSFQ[2]

این الگوریتم که در سال ۱۹۹۸ توسط .Stoica et al ارائه گردید، روشی مناسب برای برقراری عدالت بین جریانهای مختلف می‌باشد که هیچ نیازی به نگهداری اطلاعات وضعیت در مسیریابهای میانی شبکه ندارد و در نتیجه پیچیدگی محاسباتی آن بسیار کمتر می‌باشد.

در این الگوریتم هر نود لبه، نرخ ورود هر جریان را تخمین زده و سپس بسته‌های آن جریان را با این نرخ ورود برچسب می‌زند.

در عوض تمامی نودها شامل نودهای لبه اعمال زیر را انجام می‌دهند:

- بصورت دوره‌ای نرخ عادلانه $f$ را برای اتصال خروجی محاسبه می‌کنند.

---

[1] Class Based Weighted Fair Queueing

[2] Core Stateless Fair Queueing

- با ورود هر بسته، احتمال ارسال بسته، $p$ را طبق فرمول زیر محاسبه می‌کنند: $p = \min(1, \frac{f}{r})$ که $f$ نرخ عادلانه اتصال و $r$ نرخ ورود جریان است که بصورت برچسب در بسته ذخیره شده است. و سپس بسته را با احتمال $p$ به جلو ارسال می‌کنند. در صورتیکه بسته به جلو ارسال شود، برچسب آنرا با مینیمم مقدار قبلی آن برچسب و سهم عادلانه اتصال خروجی جایگزین می‌کنند. به عبارت دیگر : $r' = \min(f, r)$

بدین ترتیب سازگاری برچسب، تضمین می‌شود. بدین معنی که در نود بعدی برچسب هنوز هم بیانگر نرخ تخمینی ترافیک ورودی جریان می‌باشد.

در این روش تنها نودهای لبه نیاز به مدیریت بر روی تک تک جریانها را دارند؛ چرا که باید نرخ هر جریان ورودی را تخمین بزنند. نودهای میانی نیاز به مدیریت تک تک جریانها ندارند؛ چرا که برای محاسبه احتمال $p$ تنها نیازبه دانستن برچسب بسته و نرخ عادلانه اتصال خروجی دارند. در نتیجه پیچیدگی وضعیت CSFQ در نودهای لبه $O(n)$ و در نودهای میانی $O(1)$ است.

مشکل این روش این است که باید فیلدی اضافی در سرآیند هر بسته اضافه شود و نیاز به تغییر تمامی مسیریابهای شبکه دارد.

**پرسش های فصل**

۱. مشکلات و مزایای روش Tail Drop را توضیح دهید.
۲. مشکل همزمانی سراسری TCP را توضیح دهید. به نظر شما راه حل این مشکل چه می باشد؟
۳. اهداف اصلی مکانیزم های مدیریت فعال صف را توضیح دهید.
۴. به نظر شما تاثیر پارامترmaxp درمکانیزم RED چه می باشد؟ باتغییر maxp درفاصله ۰ تا ۱ مقدار متوسط طول صف و تاخیر صف بندی چه تغییری می نمایند.
۵. مزایا و معایب مکانیزم RED را بنویسید.
۶. تغییر پارامترهای $min_{th}$ و $max_{th}$ چه تاثیری بر کارایی مکانیزم RED می گذارد؟
۷. برتری مکانیزم ARED را بر RED معمولی بنویسید.
۸. برای پیاده سازی مکانیزم ECN چه توسعه هایی در پروتکل های IP و TCP نیاز است؟
۹. مزایای مکانیزم BLUE نسبت به RED را بنویسید.
۱۰. جنبه های تشابه وتفاوت مکانیسم REM را نسبت به مکانیزم RED بنویسید.
۱۱. نیازمندی های مکانیزم های زمانبندی را تشریح نمایید.
۱۲. تفاوت مکانیزم های زمانبندی work conserving و non work conserving را با ذکر یک مثال توضیح دهید.
۱۳. محدودیت ها و مزایای مکانیزم FIFO را توضیح دهید.
۱۴. استفاده از روش زمانبندی PQ درمسیریاب های شبکه، چه دستاوردها و چه مشکلاتی خواهد داشت؟
۱۵. تفاوت روش FQ را با روش WFQ با ذکر یک مثال توضیح دهید.
۱۶. تفاوت روش RR را با روش WRR با ذکر یک مثال توضیح دهید.
۱۷. بافرض اینکه مقادیر وزن صف های ۱و۲و۳ درمثال شکل (۱۳-۱۶) به ترتیب برابر با ۴۰٪، ۳۵٪ و ۲۵٪ باشد، مثال تشریح شده را با کمک مکانیزم DWRR مجددا تکرار نموده و توضیح دهید.
۱۸. مشکلات پیاده سازی زمانبند GPS را بنویسید.