دانشکده‌ی مهندسی برق و رباتیک

پایان‌نامه کارشناسی ارشد

مهندسی برق-گرایش مخابرات

عنوان

بررسی روش‌های کنترل جریان برمبنای روش تخصیص منابع اشتراک یافته عام

نگارش

آزاده اصغری

استاد راهنما:

دکتر امید رضا معروضی

بهمن ۱۳۹۴

شماره : ۱۳۱۰/آ.ت.ب
تاریخ : ۹۴/۰۶/۳۰
ویرایش : ------

بسمه تعالی

مدیریت تحصیلات تکمیلی
فرم شماره (۶)

فرم صورتجلسه دفاع پایان نامه تحصیلی دوره کارشناسی ارشد

با تأییدات خداوند متعال و با استعانت از حضرت ولی عصر (عج) جلسه دفاع از پایان نامه کارشناسی ارشد خانم / آقای :

محمد علی دانش پژوه رشته : **مهندسی برق** گرایش : **کنتـرل**

تحت عنوان : **کنترل تعامل دینامیکی غیر خطی بازوهای ماهـر رباتیک**

که در تاریخ **۹۴/۰۶/۳۰** با حضور هیأت محترم داوران در دانشگاه صنعتی شاهرود برگزار گردید به شرح زیر است :

قبول (با درجه عالی) امتیاز ۱۸/۷۸ ☑ دفاع مجدد ☐ مردود ☐

۱ـ عالی (۲۰ ـ ۱۹) ۲ـ بسیار خوب (۱۸/۹۹ ـ ۱۸)

۳ـ خوب (۱۷/۹۹ ـ ۱۶) ۴ـ قابل قبول (۱۵/۹۹ ـ ۱۴)

۵- نمره کمتر از ۱۴ غیر قابل قبول

عضو هیأت داوران	نام ونام خانوادگی	مرتبهٔ علمی	امضـــاء
۱ـ استاد راهنمـا	دکتر محمدرضا خاتم	استاد	
۲ـ استاد مشاور	ـــــــ	ـــــ	ـــــــ
۳ـ نماینده شورای تحصیلات تکمیلی	دکتر حداد ظریف	استادیار	
۴ـ استاد ممتحن	دکتر حبیب احمدی	استادیار	
۵ ـ استاد ممتحن	دکتر علی زاده زم	دانشیار	

رئیس دانشکـده :

دانشکده: مهندسی برق و رباتیک

گروه: مخابرات

پایان‌نامه کارشناسی ارشد خانم آزاده اصغری به شماره دانشجویی ۹۲۰۲۲۰۴ تحت عنوان بررسی روش های کنترل جریان برمبنای روش تخصیص منابع اشتراک یافته عام در تاریخ ۱۳۹۴/۱۲/ توسط کمیته تخصصی زیر جهت اخذ مدرک کارشناسی ارشد در مهندسی برق-مخابرات مورد ارزیابی و با درجه‌ی مورد‌پذیرش قرار گرفت.

استاد راهنما	امضاء	استاد مشاور	امضاء
دکتر امیدرضا معروضی			

اساتید داور	امضاء	نماینده تحصیلات تکمیلی	امضاء

تعهدنامه

این‌جانب آزاده اصغری دانشجوی دوره کارشناسی ارشد رشته برق-مخابرات دانشکده‌ی برق و رباتیک دانشگاه صنعتی شاهرود نویسنده پایان‌نامه **بررسی روش های کنترل جریان برمبنای روش تخصیص منابع اشتراک یافته عام** تحت راهنمایی دکتر امیدرضا معروضی متعهد می‌شوم:

- تحقیقات در این پایان‌نامه توسط این‌جانب انجام‌شده است و از صحت و اصالت برخوردار است.
- در استفاده از نتایج پژوهش‌های محققان دیگر به مرجع مورداستفاده استناد شده است.
- مطالب مندرج در پایان‌نامه تاکنون توسط خود یا فرد دیگری برای دریافت هیچ نوع مدرک یا امتیازی در هیچ جا ارائه نشده است.
- کلیه حقوق معنوی این اثر متعلق به دانشگاه شاهرود می‌باشد و مقالات مستخرج بانام «دانشگاه صنعتی شاهرود» و یا «Shahrood University of Technology» به چاپ خواهد رسید.
- حقوق معنوی تمام افرادی که در به دست آمدن نتایج اصلی پایان‌نامه تأثیرگذار بوده‌اند در مقالات مستخرج از پایان‌نامه رعایت می‌گردد.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که از موجود زنده (بافت‌های آن‌ها) استفاده‌شده است ضوابط و اصول اخلاقی رعایت شده است.
- در کلیه مراحل انجام این پایان‌نامه، در مواردی که به حوزه اطلاعات شخصی افراد دسترسی یافته یا استفاده‌شده است، اصل رازداری، ضوابط و اصول اخلاق انسانی رعایت شده است.

تاریخ

امضای دانشجو

مالکیت نتایج و حق نشر

کلیه حقوق معنوی این اثر و محصولات آن (مقالات مستخرج، کتاب، برنامه‌های رایانه‌ای، نرم‌افزارها و تجهیزات ساخته‌شده است) متعلق به دانشگاه شاهرود می‌باشد. این مطلب باید به نحو مقتضی در تولیدات علمی مربوطه ذکر شود.

- استفاده از اطلاعات و نتایج موجود در پایان‌نامه بدون ذکر مرجع مجاز نمی‌باشد.

این مدعیان در طلبش بی خبرانند کان را که خبر شد خبری باز نیامد . . .

الهی گفتی کریم، امید بدان تمام است، تا کرم تو در میان است، ناامیدی حرام است.

"سپاس **خدای** را که اول اوست، پیش از آنکه قبل از او اولی باشد و آخر است بدون این که بعد از او آخری باشد. حمد و سپاس بی پایان، ذات بی نیاز معبودی را که ذکرش

تجلی دل عاشقان و آرام بخش قلب عارفان است."

با تشکر از

استاد فرزانه جناب آقای **دکتر امید رضا معروضی** که در کلیه مراحل انجام این پژوهش با خوشروئی بنده را یاری و راهنمایی نموده اند،

و همچنین اساتید محترم داور، جناب آقایان **دکتر محمد رضا جوان** و **دکتر سید مسعود میر رضایی** که زحمت باز خوانی این پایان نامه را بر عهده داشته اند،

و در نهایت با تشکر خالصانه خدمت همه کسانی که به نوعی مرا در انجام این مهم یاری نمودند.

ماحصل آموخته هایم را تقدیم می کنم به آن ها که مهر آسمانی شان آرام بخش آلام زمینی ام است

به استوارترین تکیه گاهم، دستان پر مهر پدرم، به مهربان ترین نگاه زندگیم، چشمان آرامش بخش مادرم، که هر چه آموختم در مکتب عشق از او آموختم و هر چه بکوشم قطره ای از

دریای بی کران مهربانیش را سپاس نتوانم بگویم. امروز، هستی ام به امید اوست و فردا کلید باغ بهشتم رضای او.

چکیده:

مسئله‌ی زمان‌بندی ترافیک و الگوریتم‌های زمان‌بندی جهت سرویس‌دهی به داده‌های شبکه همواره از جنبه‌های مختلف پارامترهای کیفیت سرویس مورد توجه قرار گرفته‌اند. امروزه تلفن‌های هوشمند ما دارای چندین رابط از قبیل 4G,3G,wifiو... هستند و تعداد این رابط‌ها روز به روز در حال افزایش است. برای هر رابط می‌توان یک سری کاربردهایی را برتری داد. اگر تلفن‌های هوشمند از چندین رابط به طور همزمان استفاده کنند، بازده و اتصال کاربردهایی همچون رشته‌های ویدئویی و تماس‌های VOIP افزایش می‌یابد و میزان تأخیر در شبکه کاهش می‌یابد. این بدین معناست که وسایل موبایل ما هنگام استفاده از تمام ظرفیت موجود نیاز به مدیریت بسته‌ها مطابق با این اولویت‌ها دارند. زمان‌بندهای صف‌بندی منصفانه قبلی نمی-توانستند از اولویت یک کاربرد برای بعضی از رابط‌ها، در برابر بقیه استفاده کنند. در این پایان‌نامه ضمن بررسی الگوریتم‌های FCFS، SPN، RR، DRR، در حالت چندسروره و معرفی الگوریتم پیشنهادی بر مبنای الگوریتم DRR، پارامترهای میانگین زمان تأخیر و میانگین زمان انتظار را برای آن‌ها به دست آورده و با استفاده از ورودی پوآسن و کنترلر ورودی سطل نشانه، نشان می‌دهیم الگوریتم پیشنهادی دارای عملکرد بهتری در رویارویی با پارامترهای کیفیت سرویس نسبت به ۴ الگوریتم ارائه شده دارد.

کلمات کلیدی: صف‌بندی منصفانه چندسروره، الگوریتم‌های زمان‌بند بسته‌ای، زمان مجازی.

لیست مقالات مستخرج شده از پایان نامه:

۱- آزاده اصغری، امیدرضا معروضی " مقایسه‌ی الگوریتم‌های زمان‌بندی چند سروره با در نظر گرفتن اولویت سرور"**سومین کنفرانس ملی و اولین کنفرانس بین المللی پژوهش های کاربردی در مهندسی برق، مکانیک و مکاترونیک، دانشگاه مالک اشتر. بهمن ۱۳۹۴.**

۲- آزاده اصغری، امیدرضا معروضی " بهبود پارامترهای کیفیت سرویس در کنترل جریان شبکه‌های چندسروره با اولویت سرور بر مبنای الگوریتم DRR" **دومین کنفرانس بین المللی و سومین همایش ملی کاربرد فناوری‌های نوین در علوم مهندسی، دانشگاه تربت حیدریه، اسفند ۱۳۹۴.**

فهرست مطالب:

فصل ۱ مقدمه ۱

۱-۱ تفاوت کنترل جریان و کنترل ازدحام ۲

۱-۲ کلیدزنی بسته ای و کلیدزنی مداری ۳

۱-۳ اهمیت الگوریتم های زمان بندی ترافیک ۴

۱-۴ زمان بندی سرویس ۴

۱-۵ مشخصات مطلوب الگوریتم زمان بندی ۶

۱-۶ دسته بندی الگوریتم های زمان بندی ۹

۱-۷ کنترل نرخ ورودی ۱۰

۱-۸ زمان بندی سرورها و تخصیص توان ۱۲

۱-۹ زمان بندی در حالت چند سرویس دهنده ۱۳

۱-۱۰ هدف تحقیق ۱۵

۱-۱۱ ساختار پایان نامه ۱۵

فصل ۲ مروری بر گذشته ۱۷

۲-۱ الگوریتم های مبتنی بر قاب بندی ۱۸

۲-۲ الگوریتم های مبتنی بر اولویت مرتب‌شده ۱۸

۲-۳ دو دیدگاه کلی در طراحی الگوریتم های مبتنی بر اولویت مرتب‌شده ۲۵

فصل ۳ مفاهیم پایه ۲۷

۳-۱ جریان های ترافیکی σ, ρ_regular ۲۸
۳-۱-۱ الگوریتم سطل سوراخ دار ۳۱
۳-۱-۲ الگوریتم سطل نشانه ۳۲
۳-۲ سیستم های صف Fair Queuing ۳۴
۳-۲-۱ ساختار صف بندی FQ ۳۵
۳-۲-۲ روش زمان بندی WFQ ۳۸
۳-۲-۳ ساختار صف بندی GPS ۴۱
۳-۲-۴ روش زمان بندی SCFQ ۴۳
3-2-5 روش زمان بندی WF^2Q ۴۷
۳-۲-۶ روش زمان بندی FCFS ۵۰
۳-۲-۷ روش زمان بندی SPN ۵۳
۳-۲-۸ روش زمان بندی RR ۵۴
۳-۲-۹ روش زمانبندی DRR ۵۶
۳-۳ ضرورت وجود الگوریتم های چندسروره ۵۷
فصل ۴ الگوریتم پیشنهادی ۶۵
۴-۱ مقدمه ۶۶
۴-۲ شرح مسئله ۶۵
۴-۳ عدم کار آیی الگوریتم های منصفانه تک سروره ۷۱
۴-۴ روش پیشنهادی ۷۲

۴-۵ بررسی الگوریتم ها ۷۵

فصل ۵ نتیجه گیری و پیشنهادها ۸۳

۵-۱ نتیجه گیری ۸۴

۵-۲ پیشنهادها ۸۵

فصل ۶ مراجع ۸۶

فهرست شکل ها

شکل ۱-۱ مسئله ی زمان بندی ترافیک ۵

شکل ۱-۲ مدل زمان بند چندسروره ۱۳

شکل ۱-۳ مدل زمان بند تکسروره ۱۴

شکل ۱-۴ مشکل ناترتیبی در الگوریتم چندسروره ۱۵

شکل ۳-۱- مدل سطل سوراخ دار ۳۲

شکل ۳-۲- مدل سطل نشانه ۳۳

شکل ۳-۳ نمایش ساختار یک سیستم FQ ۳۵

شکل ۳-۴- مثالی از سیستم FQ ۳۶

شکل ۳-۵-مثالی از صف بندی FQ ۳۸

شکل ۳-۶- مثالی از زمان بند WFQ (ارسال بسته بر اساس زمان پایان) ۴۰

شکل ۳-۷- الف) ترتیب ورود بسته ها ب) ترتیب سرویس دهی در WFQ ۴۷

شکل ۳-۸- الف) ترتیب سرویس گیری در WF^2Q ب) ترتیب سرویس گیری در GPS ۴۹

شکل ۳-۹- مثالی از صف بندی FCFS ۵۰

شکل ۳-۱۰- مثال عددی از نحوهی سرویس دهی FCFS ۵۲

شکل ۳-۱۱- مثال عددی از الگوریتم SPN ۵۴

شکل۳-۱۲- مثال عددی از الگوریتم RR با q=1 ..... ۵۵

شکل۳-۱۳- مثال عددی از الگوریتم RR با q=4 ..... ۵۵

شکل ۳-۱۴- سیستم مدل چند سروره با در نظرگرفتن اولویت سرورها ۵۹

شکل ۳-۱۵- نمایش $Si\tau 1,t$ و $Ai\tau 1,t$ و حد بالا و پایین قابل قبول آن‌ها ۶۳

شکل ۴-۱- مثال های زمان بند بسته ای الف) حالت استاندارد(دوجریان و یک سرور). ب)دو جریان و دو سرور بدون اولویت باینری. ج) دو جریان و دو سرور با اولویت باینری ۶۸

شکل ۴-۲: سیستم مدل فرضی برای زمان بند بسته ای در حالت چند سروره ۷۰

شکل ۴-۳- میانگین زمان انتظار برای ۴ الگوریتم در مدت T=0-100(s) ۷۹

شکل ۴-۴ میانگین زمان تأخیر برای ۴ الگوریتم در مدت T=0-100(s) ۸۰

شکل ۴- ۵- مقایسه میانگین زمان تأخیر الگوریتم های DRRN و DRRNQ با ۴ الگوریتم موردنظر ۸۱

شکل ۴- ۶- مقایسه میانگین زمان تأخیر الگوریتم ها در D_pهای بیشتر از ۰/۰۲۴ ..... ۸۲

شکل ۴-۷- نتایج میانگین زمان انتظار در حالت تک سروره در T=0-100 ..... ۸۲

شکل ۴-۸- نتایج میانگین زمان تأخیر در حالت تک سروره در T=0-100 ..... ۸۳

فهرست جدول‌ها:

جدول ۴-۱- مقایسه تأثیر سطل نشانه در حالت چندسروره ۸۰

جدول ۴-۲- مقایسه اثر سطل نشتی در حالت تک سروره ۷۴

اختصارات

GPS	General Process Sharing
PGPS	Packetized General Process Sharing
FQ	Fair Queuing
WFQ	Weighted Fair Queuing
RPS	Proportional Servers Rate
FCFS	First Come First Service
SPN	Shortest Process Next
RR	Round Robin
DRR	Deficit Round Robin
QOS	Quality Of Service
DC	Deficit Counter
Q	Quantum
AWT	Average Waiting Time
ATT	Average Deley Time
VOIP	Voice Over Internet Protocol
Skype	Sky Peer-To-Peer
TCP	Transmission Control Protocol
IP	Internet Protocol
UDP	User Datagram Protocol
CDN	Content Delivery Network
ATM	Asynchronous Transfer Mode

فصل ۱

مقدمه

امروزه سهم عمده‌ای از ارتباطات در جهان، توسط شبکه‌های پرسرعت بسته‌ای[۱] صورت می‌گیرد، و این شبکه‌ها به‌سرعت در حال جایگزینی شبکه‌های قدیمی‌تر مبتنی بر کلید زنی مداری[۲] هستند. علت این امر را می‌توان ظهور کاربردهای جدید و متنوع چندرسانه‌ای مانند کنفرانس راه دور[۳] و آموزش مجازی[۴]، صوت، ویدئو بلادرنگ[۵] و همچنین افزایش چشم‌گیر کاربران این‌گونه کاربردها دانست[۱]. مسلماً با افزایش تعداد کاربران این کاربردها و تنوع سرویس‌های مورد درخواست و افزایش سرعت سرورهای ارتباطی، کنترل این شبکه‌ها از اهمیت زیادی برخوردار می‌شود. منظور از کنترل شبکه، کنترل جنبه‌های مختلف شامل کنترل نرخ[۶] ارسال کاربران، کنترل ازدحام[۷]، کنترل جریان[۸]، کنترل پذیرش[۹] ورودی جدید و موارد دیگر است. کنترل هر یک از موارد مذکور نقش عمده‌ای در بهبود کارایی شبکه و استفاده بهتر از منابع شبکه خواهد داشت.

[1] Packet based
[2] Circuit switching
[3] Teleconference
[4] E-learning
[5] Real time
[6] Rate control
[7] Congestion control
[8] Flow control
[9] Admission control

۱-۱ تفاوت کنترل جریان و کنترل ازدحام

در کنترل شبکه‌های کامپیوتری با دو مبحث روبرو هستیم: کنترل جریان و کنترل ازدحام. در کنترل جریان هدف آن است که از هماهنگی سرعت دریافت گیرنده و سرعت ارسال فرستنده اطمینان حاصل کنیم. بدین معنی که فرستنده اینقدر سریع بسته‌ها را ارسال نکند که بافر گیرنده سرریز و پر شود. اما در کنترل ازدحام، هدف اطمینان از تحویل داده شدن صحیح بسته‌های داده به گیرنده توسط شبکه است. ازدحام به دلیل اشباع شدن منابع شبکه رخ می‌دهد. وقتی ازدحام رخ می‌دهد مسیریاب‌ها ناچارند بسته‌ها را در بافر خود صف کنند اما بعد از مدتی بافر پرشده و مسیریاب ناچار می‌شود بسته‌های اضافی را دور بریزد. اما مسئله چالش‌برانگیز تخصیص یک پهنای باند مناسب به هر یک از کاربران در مسیریاب‌ها برای ایجاد عدالت بین آن‌هاست. منظور از عدالت بیشینه‌سازی، حداقل پهنای باند تخصیص‌یافته به کاربران مختلف می‌باشد به‌نحوی‌که هیچ کاربری بدون کاهش سرویس دیگر کاربران قادر به دریافت پهنای باند بیشتری نباشد[۲].

کنترل جریان در شبکه‌های بسته‌ای، ازجمله مهم‌ترین موضوعات موردتوجه محققین از همان سال‌های ابتدایی ظهور شبکه‌های کامپیوتری بوده است. مسئله تخصیص منابع به کاربران در لحظه‌ی ورود آن‌ها به شبکه، از دیدگاه کنترل جریان موردبررسی قرار می‌گیرد. اگرچه تاکنون مکانیزم‌های مختلف و متنوعی برای کنترل جریان در شبکه‌های بسته‌ای ارائه‌شده است، این موضوع کماکان از موضوعات محوری موردتوجه محققین در شبکه‌های بسته‌ای سیمی و بی‌سیم است. آنچه مسلم است، این است که وجود یک الگوریتم زمان‌بندی ترافیک[1] با طراحی صحیح و مناسب، در بسیاری از روش‌ها و مکانیزم‌های ارائه‌شده تاکنون، ضروری به نظر می‌رسد.

در ادامه‌ی این فصل، به مقایسه‌ی دو نوع کلیدزنی بسته‌ای و مداری می‌پردازیم و در مورد اهمیت وجود الگوریتم زمان‌بندی در شبکه‌های بسته‌ای، ویژگی‌ها، ملزومات و انواع آن‌ها بحث خواهد شد.

[1] Traffic scheduling

۱-۲ کلیدزنی بسته‌ای و کلیدزنی مداری

شبکه تلفنی یا شبکه PSTN[1] مبتنی بر استفاده از کلیدزنی مداری است. در کلیدزنی مداری بر اساس حداکثر نرخ ارسال کاربرد موردنظر، مسیری (مداری) بین فرستنده و گیرنده برقرار می‌شود. شبکه مسئول تخصیص منابع کافی در طی مسیر برقرار شده است. تخصیص منابع کافی می‌تواند به‌صورت مختلف مانند TDM[2] و FDM[3] صورت پذیرد. واضح است که هم ترافیک داده و هم ترافیک کاربردهای بلادرنگ مانند صوت و ویدئو در همه‌ی لحظات دارای نرخ ارسال بیشینه نیستند. به‌عبارت‌دیگر تنها در لحظات کوتاهی از ارسال، نرخ ارسال ممکن است به حداکثر خود برسد. آشکار است که استفاده از کلیدزنی مداری، با حضور کاربردهایی با ویژگی‌های مذکور، باعث اتلاف و عدم استفاده بهینه از منابع شبکه می‌شود. برای استفاده بهینه‌تر از منابع شبکه از کلیدزنی بسته‌ای[4] استفاده می‌شود. با استفاده از کلیدزنی بسته‌ای، کاربر قادر است داده خود را با نرخ حداکثر ارسال کند درحالی‌که منابع بر اساس نرخ متوسط ارسال به آن تخصیص داده شده است. در اینصورت، نیاز به بافرینگ در نقاط میانی شبکه وجود دارد تا انبار توده[5] بسته‌های ارسالی باشد و از گم‌شدن و از دست رفتن بسته‌ها جلوگیری کند. این ایده بر اساس استفاده از اصل ادغام آماری[6] [۱] است. در کلیدزنی بسته‌ای، اطلاعات (داده، صوت، ویدئو، سیگنالینگ، هم‌زمان‌سازی و ...) به‌صورت بسته‌های IP و یا ATM و یا هر پروتکل دیگری بین فرستنده و گیرنده، تبادل می‌شود. مزیت عمده‌ی این روش کلیدزنی، قابلیت ذخیره کردن بسته‌ها در هر گره‌ی شبکه و در مواقع نیاز است.

[1] Public switch telephone network
[2] Time division multiplexing
[3] Frequency division multiplexing
[4] Packet switching
[5] Burst
[6] Statistical multiplexing

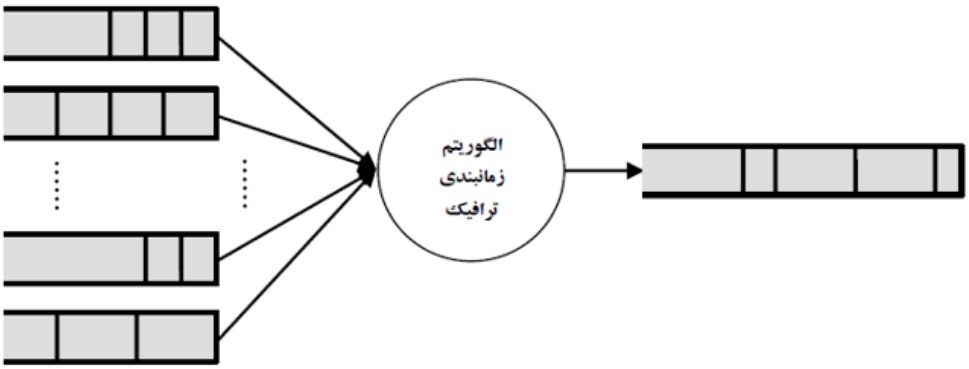
۱-۳ اهمیت الگوریتم‌های زمان‌بندی ترافیک

در ادغام آماری، گرچه شبکه ممکن است در بسیاری از اوقات، کارایی خوبی داشته باشد، ازدحام می‌تواند در بازه‌های کوتاه یا بلندی از زمان، در شبکه رخ دهد و کارایی کلی شبکه را تنزل دهد. این بازه‌های زمانی معمولاً در زمان‌هایی رخ می‌دهند که کاربران به‌صورت همزمان، بسته‌های خود را به‌صورت توده‌ای ارسال کنند. علاوه بر این، بعضی کاربران ممکن است بدرفتار[1] باشند. بدین معنا که ممکن است، کاربر، رفتار حریصانه‌ای[2] در ارسال بسته‌هایش داشته باشد و بسته‌های خود را با نرخی بیشتر از نرخ جاگیری شده، برای آن ارسال کند. این رفتار حریصانه ممکن است باعث ازدحام در گره‌های میانی شبکه گردد. عملکرد جریان‌های بدرفتار، علاوه بر ایجاد ازدحام در گره-های میانی شبکه، باعث عدم تخصیص عادلانه منابع شبکه نیز می‌شود. چراکه یک جریان بدرفتار سعی در استفاده بیش از سهم عادلانه[3] خود از منابع شبکه دارد که این باعث می‌شود، سایر جریان‌ها بسیار کمتر از سهم عادلانه خود از شبکه سرویس بگیرند. در این حالت اصطلاحاً گفته می‌شود که برای جریان‌ها در شبکه قحطی[4] رخ داده است. برای جلوگیری از بروز مشکلات فوق، استفاده از یک الگوریتم زمان‌بندی ترافیک ضروری است.

۱-۴ زمان‌بندی سرویس

دسته مهمی از مسائل زمان‌بندی ترافیک را می‌توان به‌صورت یک مسئله صف‌بندی مدل کرد که در آن ظرفیت یک محاوره خروجی (سرویس‌دهنده) باید به‌طور عادلانه بین چند محاوره (صف) ورودی تقسیم گردد. شکل (۱-۱) این مسئله را بهتر نشان می‌دهد.

[1] Miss behaving flows
[2] Greedy
[3] Fair share
[4] Starvation



شکل ۱-۱ مسئله‌ی زمان‌بندی ترافیک

به دلیل محدود بودن ظرفیت سرور خروجی در هرلحظه از زمان، تنها یک صف ورودی قادر به ارسال یک بسته در محاوره خروجی است. الگوریتم زمان‌بندی ترافیک، تعیین کننده‌ی بسته‌ای است که در هرلحظه از زمان باید روی محاوره خروجی ارسال شود.

منظور از زمان‌بندی سرویس، اصطلاحاً نوبت‌دهی به کاربران مختلف جهت دریافت سرویس بر روی یک سرور می‌باشد. هدف از این نوع زمان‌بندی، توزیع ظرفیت سرویس‌دهی یک سرور بین محاوره‌های مختلف گذرنده از آن سرور می‌باشد. یک دسته از روش‌های زمان‌بندی سرویس، روش‌های سرویس‌دهی عادلانه، FQ[1]، است که در آن‌ها ظرفیت سرویس‌دهی سرور به‌صورت عادلانه و متناسب با سهم هریک از محاوره‌ها بین آن‌ها تقسیم می‌شود.

بر اساس به‌کارگیری روش‌های FQ می‌توان تضمین نمود که یک محاوره مادامی‌که بسته‌ای جهت ارسال داشته باشد، مستقل از وضعیت سایر محاوره‌های موجود، می‌تواند از کسر مشخصی از ظرفیت سرویس‌دهی سرویس‌دهنده استفاده کند. البته، در غیاب یک محاوره سهم آن بین بقیه محاوره‌ها تقسیم می‌شود. ازجمله شناخته‌شده‌ترین روش‌های FQ که با پیچیدگی محاسباتی کمی قابل پیاده‌سازی است می‌توان به روش SCFQ اشاره کرد [۳].

[1] Fair queuing

۱-۵ مشخصات مطلوب الگوریتم زمان‌بندی

یک الگوریتم زمان‌بندی ترافیک، وظیفه دارد برای هر جریان در شبکه، سرویسی متناسب با نرخ جاگیری شده توسط آن جریان فراهم کند و مکانیزمی برای ایجاد نوعی جداسازی بین جریان‌ها ایجاد کند تا در صورت وجود بدرفتاری، در رفتار یک جریان از تأثیر آن بر سرویس سایر جریان‌ها کاسته شود؛ بنابراین ویژگی‌ها و نیازمندی‌های یک الگوریتم زمان‌بندی ترافیک را می‌توان به‌صورت خلاصه به‌صورت زیر بیان داشت:

۱. ایجاد جداسازی بین رفتار جریان‌ها: منظور، ایجاد مصونیت سرویس برای جریان‌ها در مقابل جریان‌های بدرفتار است. به‌بیان‌دیگر، یک الگوریتم زمان‌بندی ترافیک باید قادر به برآورده کردن معیارها و پارامترهای کیفیت خدمات، QOS[۱] برای هر جریان در شبکه و مستقل از رفتار سایر جریان‌ها در شبکه باشد.

۲. استفاده مؤثر از سرویس‌دهنده: یک الگوریتم زمان‌بندی ترافیک باید استفاده مؤثر از ظرفیت سرور خروجی و بیشترین بهره را از ادغام آماری داشته باشد. بر این اساس، الگوریتم‌های زمان‌بندی ترافیک را می‌توان به دو دسته کار مداوم[۲] و کار گسسته[۳] تقسیم کرد.

الگوریتم‌های کار مداوم: یک الگوریتم کار مداوم زمانی بیکار خواهد بود که هیچ بسته‌ای برای سرویس در سیستم نباشد.

الگوریتم‌های کار گسسته: یک الگوریتم کار گسسته، ممکن است بسته‌ای برای سرویس در سیستم داشته باشد ولی اصلاً سرویسی ارائه ندهد.

واضح است که الگوریتم‌های کار مداوم، دارای بهینگی در استفاده از ظرفیت سرویس‌دهنده هستند. چراکه در صورت وجود درخواست سرویس، سرویس‌دهنده هیچ‌گاه

[1] Quality of service
[2] work conserving
[3] Non-work conserving

بیکار نخواهد ماند و با حداکثر ظرفیت خود در حال سرویس‌دهی به جریان‌هاست. الگوریتم‌های WFQ [۴]، WF^2Q[۵]، SCFQ[۳]، URR [۷]، DRR [۸]، همگی از الگوریتم‌های کار مداوم هستند. در نگاه اول، استفاده از الگوریتم‌های کار گسسته، ممکن است چندان توجیهی نداشته باشد، زیرا سرویس‌دهنده حداکثر بازدهی را نخواهد داشت. ولی باید توجه داشت که الگوریتم‌های کار گسسته، در ازای تأخیر در ارائه سرویس به بسته‌ها، (بیکار بودن سرویس‌دهنده باوجود بسته در انتظار سرویس در سیستم) مزایای دیگری به دست می‌آورند. ازجمله کنترل شکل‌گیری رفتار توده‌ای بسته‌ها در خروجی سرویس‌دهنده و درنتیجه، قابلیت بیشتر در پیش‌بینی ورود بسته‌ها در سرویس‌دهنده‌های پایین دست[1] و پیش‌بینی حداکثر انباشت موردنیاز در سرویس‌دهنده‌های مسیر جریان است. ازجمله دیگر ویژگی‌های الگوریتم‌های کار گسسته، کنترل تغییرات تأخیر زمانی بسته‌هاست که ازجمله پارامترهای مهم QOS، بخصوص برای کاربردهای بلادرنگ است. الگوریتم‌های EDD-jitter [۹] و RCSP[۱۰] از این دسته‌اند.

۳. تأخیر انتها به انتهای کم: ازجمله پارامترهای دیگر QOS در شبکه‌ها، تأخیر انتها به انتهاست. الگوریتم‌های زمان‌بندی ترافیک، باید تأخیر انتها به انتهای ارسال بسته‌ها را کنترل نمایند. منظور از تأخیر انتها به انتها، تأخیر یک‌طرفه از زمان ارسال بسته به داخل شبکه تا زمان دریافت کامل آن در گیرنده است. توجه کنید که تأخیر انتها به انتهای کم، در الگوریتم‌های زمان‌بندی ترافیک می‌تواند خود باعث کاهش حجم حافظه انباشت موردنیاز شود؛ بنابراین انتخاب یک الگوریتم زمان‌بندی ترافیک با تأخیر انتها به انتهای کم می‌تواند به‌طور مستقیم در کاهش هزینه پیاده‌سازی در قالب کاهش حجم حافظه‌ی موردنیاز نیز مؤثر باشد. در مورد اندازه‌گیری تأخیر، پارامترهای مختلفی قابل اندازه‌گیری

[1] Down stream

است. ازجمله این پارامترها، می‌توان حداکثر زمان تأخیر، متوسط زمان تأخیر و توزیع زمان تأخیر را نام برد.

۴. عدالت[1]: همان‌طور که گفته شد، هدف اصلی یک الگوریتم زمان‌بندی ترافیک، تسهیم عادلانه‌ی پهنای باند محاوره خروجی (سرویس‌دهنده) بین جریان‌هایی است که آن را به اشتراک گذارده‌اند. این تسهیم پهنای باند با توجه به سهم سرویسی که هر جریان جاگیری کرده است، انجام می‌شود. در یک الگوریتم با رفتار ناعادلانه، در یک بازه زمانی کوتاه، میزان سرویس ارائه‌شده به دو جریان با نرخ جاگیری شده‌ی یکسان می‌تواند بسیار متفاوت باشد. نکته‌ی دیگر آنکه رفتار عادلانه یک الگوریتم در تسهیم پهنای باند می‌تواند خود منجر به ایجاد جداسازی مناسب بین سرویس جریان‌ها شود. توجه شود که دو الگوریتم گرچه می‌توانند دارای حداکثر تأخیر یکسان در سرویس به بسته‌ها باشند، ولی ممکن است ازنظر ویژگی عدالت، رفتار کاملاً متفاوتی داشته باشند.

۵. سادگی پیاده‌سازی و قابلیت گسترش آن[2]: ازجمله ویژگی‌های یک الگوریتم زمان‌بندی ترافیک، سادگی پیاده‌سازی آن است. این موضوع وقتی اهمیتی دو چندان پیدا می‌کند که الگوریتم طراحی‌شده قرار باشد در شبکه‌های باند وسیع[3] مورداستفاده قرار گیرد. در این شبکه‌ها نرخ ارائه سرویس بالاست. لذا زمان سرویس دادن به یک بسته، بسیار کوچک است. ازآنجاکه الگوریتم زمان‌بندی ترافیک، در هر ارسال بسته، باید بسته کاندید برای دریافت سرویس بعدی را انتخاب کند، لذا پیچیدگی زمانی الگوریتم بسیار اهمیت پیدا می‌کند. همچنین علاوه بر پیچیدگی زمانی، پیچیدگی و هزینه‌ی پیاده‌سازی سخت‌افزاری الگوریتم نیز باید موردتوجه قرار گیرد. علاوه بر افزایش نرخ سرویس، در شبکه‌های باند وسیع، در این شبکه‌ها تعداد جریان‌های ورودی و درنتیجه تعداد صف‌هایی که برای سرویس تشکیل می‌شوند نیز زیاد است؛ بنابراین قسمتی از هزینه‌ی سخت‌افزاری طراحی،

[1] Fairness
[2] Scalability
[3] Broad-band

مربوط به حجم حافظه‌ی موردنیاز است. نکته‌ی مهم دیگر در طراحی الگوریتم‌های زمان-بندی ترافیک، طراحی قابل‌گسترش است. با طراحی یک الگوریتم ساده، قابل‌گسترش ولی کارامد می‌توان آن را با اندک تغییرات کم‌هزینه در سطوح بالاتر (ازنظر نرخ ارسال و تعداد صف‌های بیشتر) نیز گسترش داد. در مورد پیچیدگی الگوریتم‌های زمان‌بندی ترافیک و انواع آن در ادامه توضیحاتی ارائه خواهد شد.

۱-۶ دسته‌بندی الگوریتم‌های زمان‌بندی

الگوریتم‌های زمان‌بندی ازنظر سازوکار در فراهم آوردن نرخ‌های سرویس‌دهی به چهار دسته تقسیم می‌شوند. دسته‌ی اول الگوریتم‌های مبتنی بر زمان مجازی هستند؛ مانند الگوریتم‌های GPS، PGPS، WF^2Q، SCFQ. در این دسته از نظم‌بندی‌ها بسته‌ها با توجه به نگاشتی از زمان مجازی که به آن‌ها تخصیص داده می‌شود، سرویس می‌گیرند. دسته‌ی دوم نظم‌بندی‌های منصفانه‌ای هستند که بر پایه‌ی نوبت‌دهی چرخشی استوارند، مانند DRR، URR، WRR [۱۱] و... . دسته‌ی سوم، الگوریتم‌های بر پایه‌ی EDD[1] هستند. درواقع در این الگوریتم‌ها به هر بسته یک موعد زمانی برای سرویس‌گیری داده می‌شود و بسته‌ها با توجه به افزایش این پارامتر سرویس می‌گیرند. Jitter-EDD و delay-EDD از این دسته‌اند. دسته آخر نظم‌بندی‌ها بر پایه‌ی حق تقدم بسته‌هاست. بسته‌هایی با حق تقدم بالاتر دارای اولویت سرویس‌گیری هستند.

علاوه بر الگوریتم زمان‌بندی، کنترل نرخ ورود به صف‌های محاوره نیز در کارآیی روش‌های کنترل جریان، تأثیر بسزایی دارند. در ادامه به معرفی دو رهیافت متفاوت به مسئله‌ی کنترل نرخ ورود می‌پردازیم.

[1] Earliest Due Date

۱-۷ کنترل نرخ ورودی

کنترل نرخ ورودی عبارت است از چگونگی تنظیم نرخ‌های ورود به شبکه برای هر محاوره، به‌نحوی‌که شبکه پایدار باشد، یعنی صف‌ها دچار سرریز نشوند. تنظیم نرخ‌های ورود معمولاً با در نظر گرفتن یک معیار تسهیم عادلانه صورت می‌گیرد. افزون براین ممکن است کنترل ازدحام در سطح شبکه نیز مدنظر قرار گیرد. منظور از کنترل ازدحام در یک الگوریتم کنترل نرخ ورود، روش‌هایی است که در یک شبکه برای جلوگیری و مقابله با سرریز بافرها یا افزایش بیش‌ازحد تأخیرها به کار می‌رود [۱۲].

یکی از روش‌های کنترل نرخ ورود، روش پنجره است [۱۲]. در این روش گیرنده پس از دریافت هر بسته، پیغام دریافت هر بسته یا Ack را برای فرستنده ارسال می‌کند. فرستنده نیز مجاز است تنها به‌اندازه‌ی مشخصی (طول پنجره) بسته‌ی ارسال‌شده بدون Ack داشته باشد؛ بنابراین با کاهش آهنگ دریافت Ack از گیرنده، نرخ ارسال اطلاعات فرستنده کاهش می‌یابد.

اولین بار، ایده‌ی پیاده‌سازی مسئله‌ی کنترل نرخ ورودی به‌صورت یک مسئله‌ی بهینه‌سازی کلی در کار پژوهشی آقای گلستانی [۱۳] مطرح گردید. در این روش مسئله‌ی مسیریابی و کنترل نرخ ورودی به‌صورت توزیع‌شده، توأمان فرمول‌بندی شده و حل‌شده‌اند. ترافیک ورودی از یک مدل آماری دلخواه تبعیت می‌کند و تنها متوسط نرخ ورود به شبکه کنترل می‌شود. تنظیم متوسط نرخ ورود، بر اساس طول مسیر هر محاوره انجام می‌شود، به‌نحوی‌که طول مسیر در این فرمول‌بندی بر اساس متوسط ازدحام ایجادشده در طی مسیر یک محاوره مشخص می‌شود. با تعیین طول مسیر، مسیریابی باهدف یافتن کوتاه‌ترین مسیر انجام می‌شود. بدین ترتیب، مسیریابی و کنترل نرخ ورودی به نحوی انجام می‌شود که حتی‌الامکان از افزایش بیش‌ازحد ازدحام و تأخیر در کل شبکه جلوگیری شود.

تحقیقات دیگری نیز در این زمینه انجام‌شده‌اند که اکثر آن‌ها مبتنی بر تنظیم متوسط نرخ ورود هستند. در مرجع [۱۴] سعی شده است تا با استفاده از تئوری دوگانی مسئله بهینه‌سازی کنترل نرخ ورودی به یک مسئله توزیع‌شده در سطح شبکه تبدیل شود که در آن محاوره‌ها با دریافت اطلاعاتی درباره‌ی وضعیت سرورهای مسیر، قادر به حل مسئله‌ی بهینه‌سازی کلی باشند.

اگرچه روش ارائه‌شده در مرجع [۱۳] با در نظر گرفتن متوسط تأخیر محاوره‌ها به‌عنوان جزئی از تابع هزینه‌ی مسئله‌ی بهینه‌سازی، سعی در پایین نگه‌داشتن متوسط ازدحام در شبکه دارد، بااین‌وجود با این رویکرد نمی‌توان تضمین نمود که ماکزیمم تأخیر یک محاوره محدود باشد. علت این امر، ماهیت آماری ترافیک ورودی به شبکه است.

درواقع، نمی‌توان تنها با کنترل متوسط نرخ ورودی به شبکه، از ورود سیلابی ترافیک به شبکه جلوگیری نمود. درنتیجه برای ترافیک ورودی با یک مدل آماری دلخواه، حتی اگر متوسط آماری تأخیر محدود شود، بازهم امکان بی‌نهایت شدن ماکزیمم تأخیر در بدترین شرایط وجود دارد. برای حل این مشکل، استفاده از روش‌های شکل‌دهی ترافیک در مبادی ورودی به شبکه پیشنهادشده است [۱۵،۱۷].

در این روش برخلاف مدل آماری ترافیک، مقدار ترافیک ورودی در یک بازه نمی‌تواند از مقدار مشخصی تجاوز کند. یکی از متداول‌ترین مشخصه‌هایی که در این زمینه مطرح می‌شود، مشخصه‌ی (σ,ρ)_regular است. اگر ترافیک با چنین مشخصه‌ای به یک سطل با حجم σ وارد شود و این سطل با نرخ ρ از طریق سوراخی که در آن وجود دارد تخلیه شود، در این صورت هیچ‌گاه سرریز نخواهد داد [۱۵،۱۶].

با استفاده از روش‌های سرویس‌دهی عادلانه در کلیه‌ی سرورهای واقع در مسیر یک محاوره و با فرض شکل‌دهی ترافیک ورودی محاوره مثلاً به فرم (σ,ρ)_regular ،می‌توان کران بالایی

مشخصی برای تأخیر آن به دست آورد. آنالیز تأخیر یک محاوره‌ی (σ,ρ)_regular در عبور از یک مسیر واحد در یک شبکه سیمی چندگامی در مراجع [۱۸]،[۲] آمده است.

در شرایطی که محاوره‌ای از چنین مشخصه‌ی ترافیکی تبعیت کند و درصورتی‌که بتوان در هر گام از مسیر آن، مقدار سرویس مشخصی برای آن فراهم آورد، می‌توان تضمین نمود که ماکزیمم تأخیر این محاوره محدود بماند. البته شرط تحقق این امر، فراهم آوردن مقدار سرویس مشخصی برای این محاوره در هر گام از مسیر است که از طریق روش‌های زمان‌بندی سرویس، میسر می-شود [۱۸]،[۳]. در فصل ۳ راجع به کنترل نرخ ورودی در این پژوهش توضیحات بیشتری ارائه می‌گردد.

۱-۸ زمان‌بندی سرورها و تخصیص توان

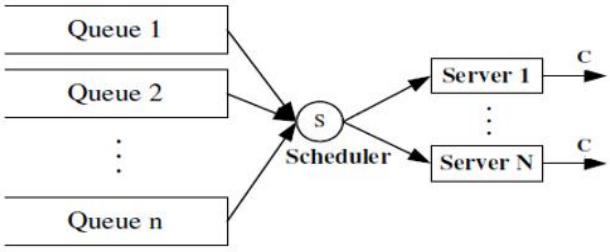
همان‌طور که مطرح شد زمان‌بندی در شبکه‌های سیمی به معنای نحوه نوبت‌بندی محاوره‌های مختلف برای سرویس‌دهی بر روی یک سرور است؛ ولی در شبکه‌های بی‌سیم، علاوه بر جنبه‌ی سابق به معنای این است که در هر بازه‌ی زمانی با توجه به وجود تداخل، کدام‌یک از خطوط اجازه فعالیت داشته باشند. در شبکه‌های سیمی با تغییر نحوه زمان‌بندی می‌توان میزان تأخیر و گذردهی محاوره‌ها بلکه گذردهی کل شبکه هم‌تغییر می‌کند [۱۲].

درواقع، یکی از چالش‌های مهم طراحی شبکه‌های بی‌سیم، زمان‌بندی ارسال روی خطوط برای استفاده‌ی بهینه از محیط مشترک دسترسی بی‌سیم است. یک رویه متمرکز زمان‌بندی که بیشترین گذردهی قابل حصول را ارائه می‌کند، اولین بار در [۱۹] ارائه‌شده است. متأسفانه مسئله ارائه‌شده در الگوریتم بهینه، از نوع NP-Hard است که ماهیتاً باید به‌صورت متمرکز اجرا شود. بر این اساس، طراحی یک الگوریتم زمان‌بندی تقریبی که به‌صورت توزیع‌شده قابل‌اجرا باشد، برای این شبکه‌ها ضروری به نظر می‌رسد. درعین‌حال این الگوریتم باید گذردهی حتی‌الامکان نزدیک به حد ماکزیمم را ارائه نماید. از دیگر عوامل مؤثر بر گذردهی و تأخیر در شبکه‌های بی‌سیم،

استراتژی تخصیص توان است. در [۶] ثابت‌شده است که جهت دست یافتن به گذردهی بهینه، لازم است مسئله‌ی کنترل توان به‌صورت توأم با زمان‌بندی حل شود. به‌علاوه در آنجا الگوریتم بهینه‌ی مسیریابی، زمان‌بندی و کنترل توان ارائه‌شده است.

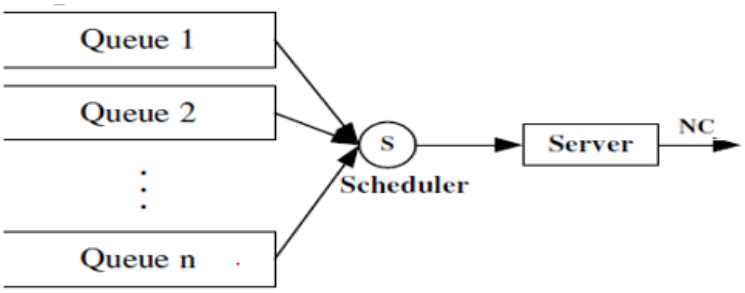
۱-۹ زمان‌بندی در حالت چند سرویس‌دهنده

با افزایش کاربران، سرویس‌های اینترنتی در سال‌های اخیر و ضرورت وجود کاربرهای چندرسانه‌ای که حجم عظیمی از اطلاعات را حمل می‌کنند، ترافیک اینترنتی رشد زیادی داشته است؛ بنابراین یک سرور تکی ظرفیت کافی را برای مدیریت این حجم وسیع از ترافیک نمی‌تواند داشته باشد؛ بنابراین باوجود ضرورت سیستم‌های چند سروره، چگونگی فراهم آوردن پارامترهای کیفیت سرویس در این سیستم‌ها، هدف اصلی این تحقیق قرار گرفته است. دو تفاوت اصلی در الگوریتم-های تک سروره و چند سروره وجود دارد. اول اینکه سیستم‌های چند سروره ازنظر تعداد سرورها و نرخ سرویس با سیستم‌های تک سروره متفاوت است. شکل (۲-۱) را در نظر بگیرید:



شکل ۲-۱ مدل زمان‌بند چندسروره

در حالت تک سروره یک بسته همیشه با نرخ NC ارسال می‌شود. درحالی‌که در حالت چند سروره بسته همیشه از طریق یکی از سرویس‌دهنده‌ها ارسال می‌شود، درنتیجه نرخ ارسال همیشه کمتر از NC است.

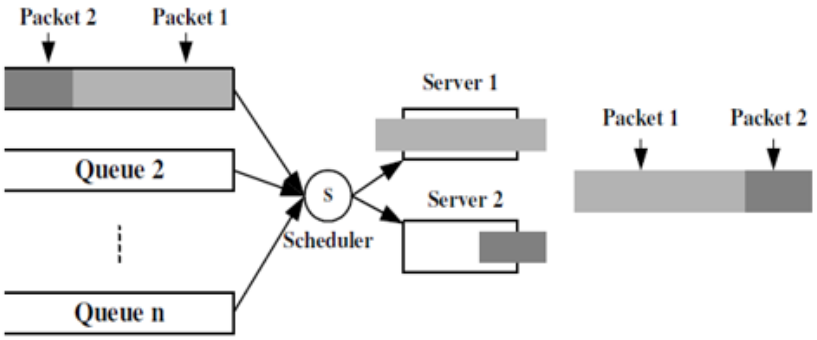


شکل ۳-۱ مدل زمان‌بند تک‌سروره

بنابراین نتایج حاصل از سیستم‌های تک سروره، نمی‌تواند قابل ارجاع به سیستم‌های چند سروره باشد؛ بنابراین برای به دست آوردن ویژگی‌های سرویس‌دهی چندسروره، تحقیقات مستقلی لازم است.

الگوریتم‌های نظم‌بندی منصفانه در حالت چند سروره بر پایه‌ی دوم (RR) در اینجا موردبحث قرار گرفته‌اند، زیرا الگوریتم‌های دسته اول (الگوریتم‌های مبتنی بر زمان مجازی) دارای پیچیدگی بالایی هستند و برای تحلیل شبکه‌های سرعت‌بالا مناسب نیستند. برای مثال MSFQ[۲۰] یک الگوریتم زمان‌بندی بر اساس زمان مجازی است که دارای پیچیدگی از مرتبه‌ی O(n) می‌باشد که متناسب با تعداد جریان‌ها در شبکه است. باوجوداینکه الگوریتم‌هایی از دسته اول، موجود هستند که با پیچیدگی کم، WFQ را تخمین می‌زنند و قابل‌بررسی در حالت چند سروره نیز هستند، اما بازهم مرتبه‌ی پیچیدگی آن‌ها نهایتاً به O(log(n)) کاهش می‌یابد [۲۱]،[۲۲] که بازهم اگر تعداد جریان‌ها در شبکه بالا باشد، پیچیدگی پیاده‌سازی همچنان در مقدار بالایی قرار دارد. برعکس الگوریتم‌های بر پایه‌ی RR دارای پیچیدگی بسیار پایینی هستند. برای مثال DRR و MSDRR دارای پیچیدگی از مرتبه O(1) می‌باشد که مقداری ثابت است و با افزایش جریان تغییر نمی‌کند؛ بنابراین باوجوداینکه در گذشته کارهای بسیاری در زمینه الگوریتم‌های دسته اول در حالت تک سروره انجام شده (برای مثال مشاهده شده که این دسته از الگوریتم‌ها حد بالای تأخیر بهتری را نسبت به دسته‌ی دوم می‌دهند [۲۳]). این پژوهش سعی در انجام تحقیقاتی درزمینه‌ی گسترش حالت تک سروره‌ی زمان‌بندهای RR به حالت چند سروره دارد.

دومین تفاوت بین الگوریتم‌های زمان‌بند تک سروره و چند سروره این است که زمان‌بندهای چندسروره مشکل ناترتیبی[1] دارند. این مشکل جدا از اینکه الگوریتم‌ها در کدام‌یک از ۴ دسته قرار دارند، در سیستم‌های چند سروره با طول بسته متفاوت به وجود می‌آید. شکل (۴-۱) بیانگر این مطلب است.



شکل ۴-۱ مشکل ناترتیبی در الگوریتم چندسروره

۱-۱۰ هدف تحقیق

هدف این پایان‌نامه، بررسی چند الگوریتم زمان‌بندی خاص در حالت چندسروره برمبنای میزان حداقل زمان میانگین انتظار و حداقل زمان تأخیر بسته‌ها، در حالتی که جریان‌ها سهم برابر در استفاده از سرورها ندارند، می‌باشد. ضمن آن‌که با تغییر روش سرویس‌دهی الگوریتم چندسروره، برمبنای روش DRR، یک الگوریتم پیشنهادی با پارامترهای کیفیت سرویس مطلوب‌تر معرفی می‌شود.

۱-۱۱ ساختار پایان‌نامه

همان‌گونه که دیده‌شد فصل اول پایان‌نامه شامل پیش‌درآمدی بر موضوع تحقیق، بیان مفهوم زمان‌بندی سرویس‌دهی عادلانه، بیان موضوع زمان‌بندی در حالت چند سروره، هدف پایان‌نامه و

[1] Miss-ordering

ساختار پایان‌نامه می‌باشد. در فصل دوم ضمن تقسیم‌بندی الگوریتم‌های زمان‌بندی به بیان انواع الگوریتم‌ها در دسته‌های مورد نظر می‌پردازیم. در فصل سوم با بیان مفاهیم نظری مورد نیاز برای آشنایی با فضای کاری، زمینه‌ای برای ایجاد درک بهتر مفاهیم فصل چهارم می‌پردازیم. فصل چهارم به توصیف الگوریتم پیشنهادی و بیان نتایج شبیه‌سازی اختصاص داده شده است. در نهایت در فصل پنجم نتایج به دست آمده در این پایان‌نامه و نیز پیشنهادهایی برای ادامه کار در این زمینه ارائه می‌شود.

فصل ۲

مروری بر گذشته

در فصل پیش الگوریتم‌های زمان‌بندی ترافیک را بر اساس کارکرد سرویس‌دهنده به دو دسته‌ی کار مداوم و کار گسسته تقسیم نمودیم. در این فصل، الگوریتم‌های زمان‌بندی ترافیک را بر اساس ساختار سرویس، به دو دسته الگوریتم‌های مبتنی بر فریم[۱] و الگوریتم‌های مبتنی بر اولویت مرتب-شده[۲] تقسیم می‌کنیم. سپس دو دیدگاه کلی در طراحی الگوریتم‌های زمان‌بندی ترافیک مطرح می‌شود. در دیدگاه اول، هدف طراحی الگوریتمی با حداکثر معیار عدالت و پارامتر تعویق محدود است و در دیدگاه دوم، هدف طراح برای رسیدن به الگوریتمی با پارامتر تعویق تا حد امکان کم و معیار عدالت محدود است.

الگوریتم‌های زمان‌بندی ترافیک را می‌توان بر اساس ساختار سرویس به دو دسته کلی الگوریتم-های مبتنی بر قاب‌بندی و الگوریتم‌های مبتنی بر اولویت مرتب‌شده طبقه‌بندی کرد.

[1] Frame-based scheduling algorithms
[2] Sorted-priority scheduling algorithms

۲-۱ الگوریتم‌های مبتنی بر قاب‌بندی

در الگوریتم‌های مبتنی بر قاب‌بندی، صف‌ها در قاب‌هایی با اندازه ثابت و یا متغیر از زمان و با توجه به ماکزیمم ترافیک قابل دریافت برای آن صف، سرویس می‌گیرند. ازجمله الگوریتم‌های این دسته می‌توان به Stop_&_Go [۲۴]، WRR [۱۱]، DRR [۸] اشاره کرد. استفاده از قاب‌بندی ثابت در چنین الگوریتم‌هایی ممکن است منجر به کار گسسته شدن الگوریتم شود (مانند Stop&Go). چراکه ممکن است صفی که قرار بوده در یک قاب ترافیک خود را ارسال کند، در طول آن قاب بسته‌ای برای ارسال نداشته باشد و لذا سرویس‌دهنده در طول آن قاب بیکار می‌ماند. در الگوریتم‌هایی مانند WRR و DRR که از قاب‌بندی با طول متغیر استفاده می‌کنند، چنین مشکلی وجود ندارد.

۲-۲ الگوریتم‌های مبتنی بر اولویت مرتب‌شده

در الگوریتم‌های کار مداوم مبتنی بر اولویت مرتب‌شده، تابعی از زمان با نام زمان مجازی سیستم معرفی می‌شود. این تابع که تابعی غیر نزولی است برای دنبال کردن پیشرفت کار سرویس‌دهنده در سیستم استفاده می‌شود. به‌عبارت‌دیگر تابع زمان مجازی سیستم که با $v(t)$ نمایش داده می‌شود، باید دنبال‌کننده سرویس نرمالیزه شده ارائه‌شده توسط سرویس‌دهنده به کل سیستم باشد[۲۵]. به‌طور مشابه برای هر صف در سیستم، یک تابع زمان مجازی صف تعریف می‌شود که دنبال‌کننده سرویس نرمالیزه شده دریافت شده توسط آن صف در مواقع غیر خالی[1] بودن آن است. ایده اساسی در طراحی الگوریتم‌های مبتنی بر اولویت مرتب‌شده، فراهم آوردن شرایطی است که تابع زمان مجازی هر صف غیر خالی بتواند تابع زمان مجازی سیستم را تعقیب کند. به‌عبارت‌دیگر اختلاف این دو دارای یک کران بالایی باشد. وجود این کران بالایی به نوعی تضمین‌کننده محدود

[1] Backlogged

بودن معیار عدالت الگوریتم است. چراکه باوجوداین شرایط، اختلاف بین زمان مجازی هر دو صف غیر خالی نیز دارای یک کران بالا می‌شود و این به معنای محدود بودن معیار عدالت [۳] برای الگوریتم است. اصلی‌ترین عامل در به وجود آوردن شرایط مذکور، تعریف مناسب تابع زمان مجازی سیستم است.

در الگوریتم‌های مبتنی بر اولویت مرتب‌شده در هنگام ورود هر بسته، یک برچسب زمانی[1] بر اساس تابع زمان مجازی سیستم، طول بسته، برچسب بسته‌های قبلی پذیرفته‌شده در سیستم و نرخ جاگیری شده صف مربوطه محاسبه می‌شود و به بسته تخصیص داده می‌شود. بسته‌ها سپس به ترتیب صعودی برچسب، سرویس می‌گیرند و به سرور خروجی ارسال می‌شوند. بدین معنا که بسته با برچسب کوچک‌تر نسبت به بسته با برحسب بزرگ‌تر دارای اولویت سرویس است. چگونگی محاسبه‌ی مقدار برچسب برای اکثر الگوریتم‌های مبتنی بر اولویت مرتب‌شده یکسان و به‌صورت زیر است[۲۵]:

فرض کنید r_k نرخ جاگیری شده برای صف k-ام و p^i_k بسته i-ام از صف k م در یک دوره کار[2] باشد. یک دوره کار در یک سرویس‌دهنده کار مداوم، بزرگ‌ترین دوره‌ای از زمان است که سرویس‌دهنده در آن به‌طور پیوسته مشغول سرویس است. همچنین فرض کنید که aik، d^i_k و L^i_k به ترتیب زمان ورود، زمان خروج و طول بسته p^i_k باشند. در این صورت بسته‌ها بر اساس رابطه زیر برچسب می‌خورند:

$$F_k^i = \frac{1}{r_k} L_k^i + \max\left(F_k^{i-1}, v\left(a_k^i\right)\right), \qquad k \in K, i = 1,2,3,\dots \tag{۱-۲}$$

$$F_k^0 = 0$$

[1] Time stamp
[2] Busy period

که در آن K تعداد کل صف‌هایی است که سرور خروجی را به اشتراک گذارده‌اند و F^i_k برچسب محاسبه‌شده منتسب به بسته p^i_k است.

تنوع خدماتی که از طریق شبکه‌های سوئیچ بسته‌ای ارائه می‌شود، موجب آن شده است که کاربران مختلف، سطح کیفی متفاوتی را به لحاظ شاخصه‌های کیفی خدمات مانند تأخیر، جیتر و پهنای باند از شبکه مطالبه نمایند. تأمین این نیاز با روش‌های سنتی کنترل جریان که مبتنی بر تخصیص منابع بر اساس سیاست، تا حد امکان[1]، است مقدور نیست، چراکه این روش‌ها شاخصه-های کیفیت خدمات را به‌طور میانگین تأمین می‌سازند.

روش اشتراک منابع عام (GPS) روش دیگری برای تخصیص منابع به مشترکین متقاضی خدمات می‌باشد که شاخصه‌های کیفی را بر اساس سیاست، بدترین حالت[2]، در نظر می‌گیرد و لذا امکان تضمین حداقلی منابع موردنیاز را برای سرویس‌های مختلف فراهم می‌سازد. مدل ریاضی این روش با فرض امکان تحقق جریان‌های ترافیکی شبه سیال به‌دست‌آمده است که به خاطر محدودیت در انشقاق بسته‌ها در مقیاس بسیار کوچک، در سطح بیت، به لحاظ عملی قابل پیاده سازی نیست. همان گونه که اشاره شد عامل اصلی در بوجود آمدن شرایط مختلف تأخیر و عدالت در الگوریتم-های زمان‌بندی ترافیک مبتنی بر اولویت مرتب‌شده، تعریف تابع زمان مجازی سیستم است. در سال ۱۹۹۱، ژانگ[3] از زمان حقیقی به عنوان تابع زمان مجازی سیستم استفاده کرد [۲۶] به‌عبارت‌دیگر، در الگوریتم او که با نام virtual-clock شناخته می‌شود، تابع زمان مجازی سیستم به‌صورت زیر است:

$$v(t) = t \qquad (۲-۲)$$

[1] Best effort
[2] Worst case
[3] Zhang

گرچه الگوریتم وی ازنظر کمینه‌سازی حداکثر تأخیر دارای عملکرد بهینه است، ولی معیار عدالت در این الگوریتم نادیده گرفته شده است. اولین بار در سال ۱۹۹۳، پرخ و گلیگر[1] الگوریتم خود را با نام PGPS[2]، طی دو مقاله ارائه کردند [۲]،[۲۷]. الگوریتم مشابهی توسط دیمرس، کشاو و شنکر[3] با نام[4](WFQ) در [۴] ارائه شد. الگوریتم WFQ درواقع نسخه بسته‌ای الگوریتم GPS است. الگوریتم GPS یک الگوریتم ایده‌آل زمان‌بندی ترافیک سیال است. به‌عبارت‌دیگر، در GPS فرض بر این است که بسته‌ها بینهایت کوچک هستند، طوری که ترافیک ورودی به یک صف را می‌توان به‌صورت یک جریان سیال در نظر گرفت. همان‌طور که در فصل پیش بیان شد، الگوریتم WFQ در سیستم بسته‌ای، در هر لحظه بسته‌ای را برای سرویس انتخاب می‌کند که در سیستم GPS سرویسش را زودتر تمام می‌کند با این فرض اضافه که در طی ارسال این بسته در هیچ بسته‌ای وارد سرویس‌دهنده نشود. درواقع این فرض اضافه برای کار مداوم شدن الگوریتم WFQ لازم است. درواقع همین فرض است که باعث اختلاف سرویس یک صف در دو سیستم GPS,PGPS می‌شود. به این منظور، تابع زمان مجازی سیستم که بیان کننده سرویس بهنجار ارائه‌شده توسط سرویس‌دهنده در سیستم GPS است، به‌صورت زیر تعریف می‌شود:

$$v(t_2) - v(t_1) = C(t_2 - t_1)\left(\sum_{k \in B(t_1,t_2)} r_k\right)^{-1} \qquad (۳-۲)$$

که در آن C نرخ سرویس‌دهنده و $B(t_1, t_2)$مجموعه صف‌های غیر خالی در فاصله زمانی (t_1, t_2) است.

[1] Parekh & gallager
[2] Packet-based generalized process sharing
[3] Demers& Keshav& Shenker
[4] Weighted Fair Queuing

در [۹] اختلاف سرویس در دو سیستم سیال و بسته‌ای محاسبه شده است و کران بالایی برای این اختلاف به دست آمده است. سپس از این کران بالا، برای محاسبه معیار عدالت و کران بالای تأخیر یک بسته در سیستم استفاده شده است.

الگوریتم WFQ یک الگوریتم بسته‌ای وابسته به سیستم سیال متناظر است. این وابستگی می‌تواند منجر به پیچیدگی زیاد پیاده‌سازی WFQ شود. شرایطی را در نظر بگیرید که تعداد زیادی بسته در یک فاصله زمانی کوتاه ε سرویسشان در سیستم سیال تمام شود و صفشان خالی شود. در این وضعیت، تابع زمان مجازی سیستم می‌بایستی در یک فاصله زمانی به تعداد صف‌هایی که خالی شده‌اند بهنگام شود و این به معنای پیچیدگی از مرتبه O(K) در نگه داشتن تابع زمان مجازی سیستم است. در فصل بعد این الگوریتم‌ها به‌طور مفصل‌تر مورد بررسی قرار می‌گیرند.

گلستانی در سال ۱۹۹۴ با ارائه الگوریتم SCFQ[1] سعی در رفع این مشکل کرد[۳]. در الگوریتم SCFQ سعی بر این است که تابع زمان مجازی سیستم به جای اینکه از یک سیستم سیال گرفته شود، از خود سیستم بسته‌ای استخراج گردد. به همین منظور در این الگوریتم، تابع زمان مجازی در بازه سرویس هر بسته برابر برچسب همان بسته در حال سرویس تعریف می‌شود. بدیهی است در چنین سیستمی شکل تابع زمان مجازی به‌صورت یک تابع پله‌ای در خواهد آمد.

الگوریتم SCFQ دارای معیار عدالت بهتری نسبت به سایر الگوریتم‌ها، حتی نسبت به WFQ است[۳]. همچنین در مورد کران بالای تأخیر آن بیان شد که این الگوریتم دارای پارامتر تعویق محدودی است و لذا کران بالای تأخیر آن نیز محدود می‌باشد؛ بنابراین حداکثر تأخیری که هر بسته متحمل می‌شود نمی‌تواند بی‌نهایت شود. ولی پارامتر تعویق این الگوریتم وابسته به تعداد کل صف‌های موجود در سیستم است و می‌تواند بسیار بزرگتر از WFQ شود. در سال ۱۹۹۷ نسخه

[1] Self-clocked fair queuing

دیگری از این الگوریتم توسط گویل، وین و چنگ[1] با نام SFQ ارائه شد که در آن به جای استفاده از زمان مجازی پایان سرویس بسته در حال سرویس به عنوان تابع زمان مجازی سیستم، از زمان مجازی آغاز سرویس بسته در حال سرویس استفاده شد [۲۸]،[۲۹]. در این الگوریتم هر بسته دو برچسب دارد: زمان مجازی آغاز سرویس بسته و زمان مجازی پایان سرویس بسته. بسته‌ها نیز در این سیستم به ترتیب صعودی برچسب زمان مجازی آغاز سرویس خود سرویس می‌گیرند.

پس از دو الگوریتم WFQ و SCFQ، تلاش‌های بسیاری برای ارائه الگوریتم‌هایی با پیچیدگی پیاده‌سازی پایین و کارایی رفتار عدالت و تأخیر مناسب به عمل آمد. ازجمله مهمترین تلاش‌ها توسط وارمه و استادیالیس[2] در سال ۱۹۹۸ انجام شد ک منجر به معرفی کلاس خاصی از الگوریتم‌های زمان‌بندی ترافیک [۳۰] با نام الگوریتم‌های به اختصار RPS[3] شد. الگوریتم‌های این کلاس دارای این ویژگی مشترک هستند که همگی دارای پارامتر برابر پارامتر تعویق الگوریتم WFQ هستند. به‌عبارت‌دیگر،

$$\Theta_i^{\text{RPS class}} = \frac{L_i}{r_i} + \frac{L_{max}}{C} \qquad (۴-۲)$$

معرفی کنندگان این کلاس از الگوریتم‌های زمان‌بندی علاوه بر معرفی آن، چارچوبی نیز برای طراحی الگوریتم در آن ارائه کردند که البته معرفی کامل کلاس الگوریتم‌های RPS، ویژگی‌های آن‌ها و روش طراحی آن‌ها خارج از مطالب این پایان‌نامه است [۳۱]. پس از معرفی RPS در سال ۱۹۹۸، الگوریتم‌های زیادی از این کلاس ارائه شد که از مهمترین آن‌ها می‌توان به SPFQ [۳۲]، MD_SCFQ [۳۳]، SWFQ [۳۴]، EWFQ [۳۵] و ... اشاره کرد. در ارائه همه این الگوریتم-ها سعی محققین بر معرفی الگوریتمی با پیچیدگی پیاده‌سازی تابع زمان مجازی کم و معیار عدالت محدود بود.

[1] Goyal & vin & Cheng
[2] Varma &Stilliadis
[3] Proportional Servers Rate

رشد روز افزون کاربران اینترنتی و همچنین حجم وسیع داده‌های ارسالی به سرورها، محققین را بر آن داشت تا تحقیقات خود را از الگوریتم‌های زمان‌بندی تک سروره به حالت‌های چند سروره معطوف کنند. هرچند پژوهش در زمینه‌ی زمان‌بندهای تک سروره بسیار وسیع‌تر بوده و هنوز هم مورد توجه محققین قرار دارد، اما اقداماتی نیز در زمینه الگوریتم‌های زمان‌بندی چند سروره انجام شده است. ازجمله کارهای اولیه در این زمینه می‌توان به الگوریتم MSF^2Q [۳۶] اشاره کرد که در آن GPS را به حالتی تعمیم دادند که چندین سرور از ظرفیت سرویس های مساوی به‌صورت موازی با هم کار می‌کنند تا سرویسی را فراهم کنند که بدون هیچ گونه قید مسیریابی بین جریان ها و سرورها، جریان پیدا کنند. درواقع همه‌ی جریان‌ها با نرخ برابر می‌توانند از همه‌ی سرورها استفاده کنند. در این سیستم مدل N سرور به‌طور همزمان با نرخ r کار می‌کنند و نرخ عدالت محدود برای آن‌ها محاسبه می‌شود. درواقع MSF^2Q وقتی که تعداد سرورها یک باشد به WFQ تنزل می‌یابد. همچنین مسئله بار ترافیک در شبکه های بی‌سیم، اغلب به‌طور ناهمگن در بین نقاط دسترسی توزیع می شود که نتیجه آن، تخصیص پهنای باند ناعادلانه بین کاربران است که دراین راستا [۳۷] یک الگوریتم نقطه دسترسی کارآمدی را بیان می کند که تخصیص پهنای باند را با بیشینه سازی حداقل پهنای باند تخصیص یافته به‌طور منصفانه‌ای تضمین می کند. در [۳۸] تحقیق شده است که تخصیص منصفانه منابع همگن به چه معنی است. درواقع مسئله تخصیص منصفانه چند نوع از منابع را به کاربرهایی با تقاضای ناهمگن آدرس دهی کرده است و در حالت خاص، عدالت منبع غالب و برتر را پیشنهاد داده است که تعمیمی از بیشینه سازی حداقل پهنای باند تخصیص یافته برای حالت چندمنبعه است. همچنین مدل DRGPS (منبع غالب در GPS) را که یک ایده آل سازی صف بندی منصفانه برپایه ی جریان های ترافیکی شبه سیال می باشد را بیان می کند و صفوف منصفانه وزن دار را بر مبنای حالت بدترین شرایط برای چند منبع در [۳۹] مورد تحلیل قرار می دهد. در [۴۰] شبیه سازی هیبرید در تخمین GPS مورد بررسی قرار گرفت. ازجمله جدیدترین فعالیت هایی که در این زمینه صورت گرفته است، می توان به تخمین وزن

های بهینه برای GPS زمان گسسته [۴۱] اشاره کرد که از تکنیک Walraevens برای یافتن مجموعه ای از تخمین‌ها برای یک عملکرد خاص استفاده کرده است. در [۴۲] طراحی، اجرا و ارزیابی مسئله کنترل ازدحام در شبکه‌های چند مسیره TCP مورد بررسی قرار گرفت. مرجع [۴۳] دو الگوریتم URR و DRR از دسته الگوریتم‌های مبتنی بر فریم در حالت چند سروره، از لحاظ درصد پیچیدگی نسبت به الگوریتم‌های دسته اول بررسی شد که بازهم در آن همه‌ی سرورها سهم برابر در سرویس دهی به جریان‌ها را داشتند و بسته‌ها همانند نظم صف بندی حالت تک سروره‌ی خود، سرویس‌دهی شدند.

۲-۳ دو دیدگاه کلی در طراحی الگوریتم‌های مبتنی بر اولویت مرتب‌شده

از بحث بالا می‌توان دو دیدگاه کلی را در طراحی الگوریتم‌های زمان‌بندی ترافیک مبتنی بر اولویت مرتب‌شده، مشاهده کرد:

۱- طراحی الگوریتم با تأکید بر کاهش معیار بی‌عدالتی

۲- طراحی الگوریتم با تأکید بر کاهش حداکثر تأخیر

الگوریتم‌هایی مانند SCFQ و SFQ از دسته اول و الگوریتم‌هایی مانند WFQ و SPFQ از دسته دوم‌اند. در این دسته‌بندی به خوبی می‌توان دید که الگوریتم‌های دسته اول گرچه دارای حداکثر تأخیر محدود هستند ولی پارامتر تعویق آن‌ها به خوبی الگوریتم‌های دسته دوم نیست. همچنین الگوریتم‌های دسته دوم، گرچه دارای معیار عدالت محدود هستند، ولی معیار عدالتشان به خوبی دسته اول نیست. به‌طور شهودی هم می‌توان چنین بده بستانی را بین عدالت و تأخیر مشاهده کرد. چراکه رسیدن به عدالت بهتر بین صف‌ها در سیستم بسته‌ای به معنای آن است که صف‌ها همیشه باید صبر کنند و سرویسشان را با هم جلو ببرند. به‌عبارت‌دیگر اگر صفی، سرویسش

از بقیه جلو افتاد (به دلیل ماهیت بسته‌ای سرویس) باید صبر کند تا بقیه صف‌ها نیز سرویسشان به سرویس او برسد و این به معنای ایجاد تأخیر در سرویس است.

بنابراین تحقیق در زمینه‌ی بهبود الگوریتم‌های زمان‌بندی ترافیک مبتنی بر اولویت مرتب‌شده را می‌توان در دو جهت کلی پیگیری کرد:

۱- بهبود رفتار تأخیر الگوریتم‌های دسته اول

۲- بهبود رفتار عدالت الگوریتم‌های دسته دوم

نکته مهمی که لازم به ذکر است، آن است که در همه فعالیت‌های انجام شده در این زمینه همواره تلاش برای کاهش پیچیدگی الگوریتم به‌طور همزمان وجود داشته است.

فصل ۳

مفاهیم پایه

مسئله‌ی الگوریتم‌های زمان‌بندی تقریباً در سال ۱۹۸۹ به‌طور جدی از حالت تک‌سروره موردبحث بسیاری از محققین قرار گرفته است و با توجه به تحقیقات زیادی که در این زمینه انجام شده، بهبود چشم‌گیری داشته است. در این فصل ضمن بیان مبانی نظری مربوطه جهت آشنایی خواننده با فضای کاری الگوریتم‌های زمان‌بندی، تاریخچه‌ی ظهور آن‌ها را از ابتدائی‌ترین الگوریتم بیان کرده و با تعمیم آن تعاریف به حالت چند سروره، سعی در ایجاد زمینه‌ای برای بیان الگوریتم پیشنهادی در فصل ۴ خواهیم داشت.

در این قسمت به بررسی مدلی از ترافیک خواهیم پرداخت که در آن برخلاف مدل آماری ترافیک می‌توان یک پوش بر روی ترافیک ورودی در نظر گرفت. ضمن معرفی این کلاس از جریان های ترافیکی، مروری بر روش‌های سرویس دهی عادلانه به محاوره ها خواهیم داشت.

۳-۱ جریان‌های ترافیکی (σ, ρ)_regular

به منظور توصیف دقیق ترافیک، فرض می‌شود که $A(t)$ بیانگر حجم ترافیک ورودی یک جریان ترافیکی تا زمان t و نه خود لحظه‌ی t، باشد؛ به‌بیان‌دیگر $A(t)$ تابعی پیوسته از چپ باشد، یعنی

$f(t_1, t_2) \triangleq f(t_1, t_2)$ ، $f(t)$ بلاوه در حالت کلی برای تابع $A(t^-) = A(t)$. به‌صورت $f(t_2) - f(t_1)$ تعریف می‌شود. بدین ترتیب، حجم ترافیک ورودی در بازه‌ی $[t_1, t_2)$ به‌صورت $A(t_1, t_2)$ مشخص می‌شود.

تعریف(σ, ρ)_regular

جریان ترافیکی $A(t)$ را(σ, ρ)_regular گوئیم هرگاه برای هر بازه‌ی دلخواه $[t_1, t_2)$ داشته باشیم [۱۸]:

$$A(t_1, t_2) \leq \sigma + \rho(t_2 - t_1) \qquad (۱-۳)$$

در این تعریف، پارامتر ρ را اصطلاحاً ماکزیمم نرخ ارسال پیوسته[1] و پارامتر σ را ماکزیمم انباشتگی[2] ترافیک می‌نامیم. فرض کنید تابع $W(t)$ *نظیر* جریان ترافیکی$A(t)$ به فرم زیر تعریف شود:

$$w(t) = \max_{\forall \tau \leq t} [A(\tau, t) - \rho(t - \tau)] \qquad (۲-۳)$$

دو لم زیر به بررسی ویژگی‌های این تابع می‌پردازند.

لم ۳-۱: در تعریف $W(t)$، مقدار ماکزیمم در رابطه‌ی (۳-۲)، درτ_t ای حاصل می‌شود که $w(\tau_t) = 0$ است. (در زیر نویس : از آنجا که $A(t)$ تابعی پیوسته از چپ است نتیجه می‌شود که $w(t)$ نیز پیوسته از چپ است و لذا $w(t^-) = w(t)$)

اثبات: فرض کنید مقدار ماکزیمم در رابطه‌ی (۳-۲) در لحظه‌یτ_t حاصل شود، داریم:

[1] Sustainable Rate
[2] Burstiness

$$w(t) = \max_{\forall \tau \le t}[A(\tau, t) - \rho(t-\tau)] \qquad (۴-۳)$$
$$\geq [A(\tau_t, t) - \rho(t-\tau_t)]$$
$$+ [A(\tau, \tau_t) - \rho(\tau_t - \tau)], \quad \forall \tau \le \tau_t$$
$$\Longrightarrow w(t) \geq [A(\tau_t, t) - \rho(t-\tau_t)]$$
$$+ \operatorname*{Sup}_{\forall \tau \le \tau_t} [A(\tau, \tau_t) - \rho(\tau_t, \tau)] \geq w(t) + w(\tau_t)$$
$$\Longrightarrow w(\tau_t) = 0 \qquad \blacksquare$$

لم ۳-۲: در صورتی که $w(t)$ براساس رابطه‌ی (۳-۲) تعریف شود، می‌توان اثبات نمود که در هر بازه‌ی دلخواه $[\tilde{t}, t]$، جریان ترافیکی $A(t)$ در رابطه‌ی زیر صدق می‌کند.

$$A(\tilde{t}, t) \le \rho(t-\tilde{t}) + w(\tilde{t}, t) \qquad (۵-۳)$$

اثبات: فرض کنید τ_t و $\tau_{\tilde{t}}$ ، لحظاتی از زمان باشند، که به ترتیب در تعریف $w(t)$ و $w(\tilde{t})$ به مقدار ماکزیمم منجر می‌شوند. برای $\forall \tau \in (\tau_t, t)$ می‌توان نتیجه گرفت که : $w(\tau_t) = 0$ است). بدین ترتیب، می‌توان استدلال نمود که $\tau_{\tilde{t}} \le \tau_t$ است و درنتیجه:

$$w(t) \geq A(\tau_{\tilde{t}}, t) - \rho(t - \tau_{\tilde{t}}) \qquad (۶-۳)$$
$$= A(\tau_{\tilde{t}}, \tilde{t}) - \rho(\tilde{t} - \tau_{\tilde{t}}) + A(\tilde{t}, t) - \rho(t-\tilde{t})$$
$$= w(\tilde{t}) + A(\tilde{t}, t) - \rho(t-\tilde{t})$$
$$\Longrightarrow A(\tilde{t}, t) - \rho(t-\tilde{t}) \le w(\tilde{t}, t) \quad \blacksquare$$

لم ۳-۳ : جریان ترافیکی $A(t)$ دارای مشخصه‌ی (σ, ρ)_regular است اگر و تنها اگر وجود داشته باشد تابع $\beta(t)$ به نحوی که:

$$0 \le \beta(t) \le \sigma \qquad (۷-۳)$$

$$A(t_1, t_2) \le \rho(t_2 - t_1) + \beta(t_1, t_2) \qquad (۸-۳)$$

اثبات: کفایت شرط فوق به وضوح برقرار است. برای اثبات لزوم شرط، می‌توان تابع $\beta(t)$ را برابر $w(t)$ اختیار نمود. می‌توان به سادگی ملاحظه کرد که $0 \le w(t) \le \sigma$ است (درواقع، از آنجا که در رابطه‌ی (۳-۲) عبارت $A(\tau,t) - \rho(t-\tau)$ بازای $\tau = t$ برابر صفر می‌شود، کران پایینی $0 \le w(t)$ نتیجه می‌شود). برقراری رابطه‌ی (۳-۸) نیز از لم ۳-۲ نتیجه می‌گردد. ■

می‌توان نظیر یک جریان ترافیکی (σ,ρ)_regular ، یک مدل سطل سوراخ‌دار ارائه کرد. از این منظر، جریان ترافیکی $A(t)$ ، (σ,ρ)_regular است اگر و تنها اگر ورود آن به یک سطل با حجم σ و نرخ خروج ρ هیچ‌گاه ایجاد سرریز ننماید [۱۸].

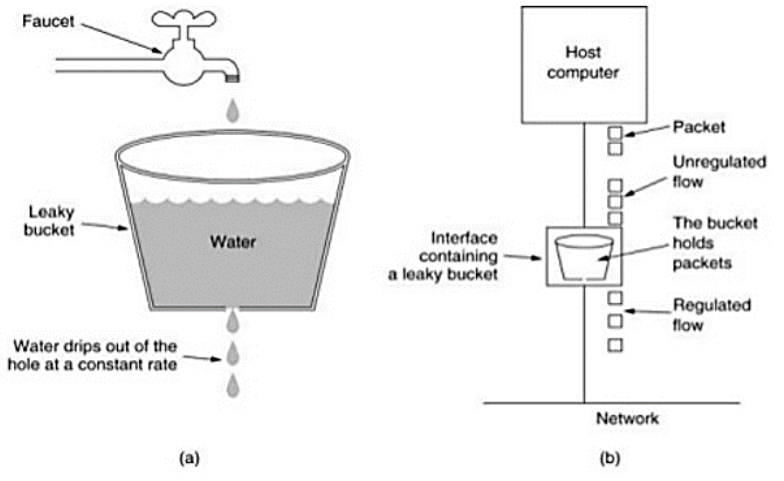
اگر یک سطل با حجم σ و نرخ خروج ρ را در نظر بگیریم، تابع $w(t)$ در هر لحظه بیانگر مقدار ترافیک موجود داخل سطل است (شکل ۳-۱). بدین ترتیب، اگر ترافیک ورودی به یک سطل با حجم σ و نرخ خروج ρ ، (σ,ρ)_regular باشد طبق لم ۳-۳، سطل هیچ‌گاه سرریز نخواهد کرد. به علاوه اگر سطل هیچ‌گاه سرریز نکند $w(t) \le \sigma$ ، نتیجه می‌شود که ترافیک ورودی (σ,ρ)_regular است. بر این اساس، $w(t)$ را اصطلاحاً انباشتگی ترافیک در لحظه‌ی t می‌نامیم. این ایده مارا به سمت دو الگوریتم کنترل ورودی سوق می‌دهد که در ادامه به معرفی آن‌ها می‌پردازیم:

۳-۱-۱ الگوریتم سطل سوراخ‌دار

مانند شکل (۳-۱) سطلی را در نظر بگیرید که منفذ کوچکی در کف آن وجود دارد. هنگامی که آن را با آب پر می‌کنید، آب بطور منظم از کف آن چکه می‌کند. اگر آن را سریع‌تر پر کنید، میزان نشتی آب همان مقدار قبلی خواهد ماند. چنانچه سرعت پر کردن سطل را کاهش دهید، باز آب با همان نرخ خواهد چکید. در شبکه‌های کامپیوتری نیز اگر بافر یا صفی داشته باشیم (سطل) که داده‌ها را با نرخ ثابتی ارسال کند (منفذ ته سطل)، می‌توانیم نحوه‌ی ارسال داده‌ها را کنترل کرده و به ترافیک خروجی سرور نظم دهیم.

فرض کنید هر سرور شبکه شامل یک سطل نشتی باشد و از گام‌های زیر پیروی کند.

- وقتی گره میزبان[1] بسته‌ای را میفرستد، این بسته به داخل سطل انداخته می‌شود.
- بسته در یک نرخ ثابتی نشت پیدا می‌کند و از سطل خارج می‌شود. به‌طوریکه رابط‌های شبکه بسته‌ها را در یک نرخ ثابت ارسال می‌کنند.
- ترافیک انباشته توسط سطل نشتی به یک ترافیک یکنواخت تبدیل می‌شود.
- در عمل صف یک صف محدود است که در یک نرخ محدودی بسته را ارسال می‌کند.



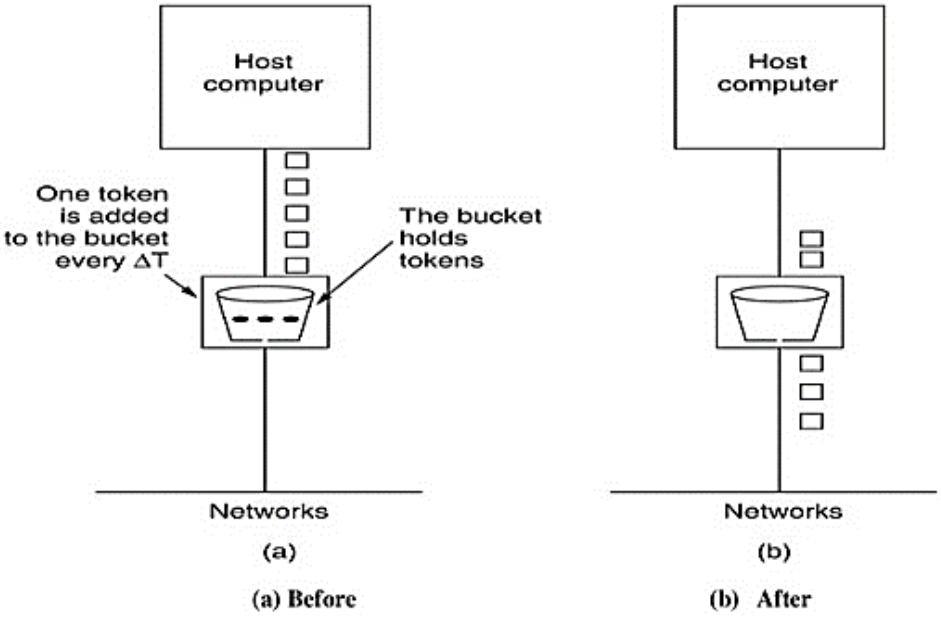
شکل ۳-۱- مدل سطل سوراخ‌دار

۳-۱-۲ الگوریتم سطل نشانه

الگوریتم پیشین نرخ خروج ثابتی را به داده‌ها تحمیل می‌کند و اینکه ترافیک چه اندازه انفجاری باشد، برایش اهمیتی ندارد. سطل نشان‌دار انعطاف بیشتری را نسبت به سطل سوراخ‌دار فراهم می‌کند. در این الگوریتم، سطل نشانه‌هایی را بطور متناوب تولید می‌کند و به هر بسته که قصد خروج دارد یک نشانه می‌دهد. در صورتی که هیچ نشانه‌ای موجود نباشد، بسته‌ها باید منتظر تولید نشانه‌های جدید شوند. این تکنیک علاوه بر اینکه مانع ارسال انفجاری داده‌ها می‌شود، به ماشین

[1] Host

اجازه می‌دهد که سهمیه‌ی خود را حداکثر تا اندازه‌ی حجم سطل (اندازه‌ی بافر) پس‌انداز کرده و مجموعه‌ای از بسته‌ها را بطور یکجا ارسال کند. شکل زیر دو سناریو را قبل و بعد از اینکه نشانه‌ها وارد سطل شوند نشان می‌دهد. در شکل اول سطل دارای سه نشانه است و پنج بسته منتظر برای ارسال وجود دارد. در شکل b سه بسته با حذف سه نشانه از سطل خارج شده‌اند و دو بسته همچنان بیرون از سطل باقی مانده‌اند. الگوریتم سطل نشانه نسبت به سطل نشتی دارای محدودکنندگی کمتری است و حد عبور یک توده به وسیله‌ی نشانه‌های در دسترس در سطل در بازه‌های ثابتی از زمان محدود می‌شود. شکل زیر بیانگر این مطلب است.



شکل ۳-۲- مدل سطل نشانه

ما در این پایان‌نامه از فرمول سطل نشانه برای کنترل ورودی پوآسن و جلوگیری از انباشتگی استفاده کرده‌ایم. در اثبات GPS تک سروره کلاسیک، می‌توان نشان داد که تأخیر یک بسته (بازه‌ی زمانی هنگامی که آخرین بیت بسته می‌رسد تا هنگامی که آخرین بیت آن سرویس داده می‌شود) در جریان i بیشتر از $\frac{\sigma_i}{\rho_i}$ نیست. این کنترل پذیرش بسیار ساده است: اگر $\sum_i \rho_i < r$ و $\sum_i \sigma_i < B$ باشد، که در آن B سایز بافر بسته می‌باشد، آنگاه جریان i می‌تواند وارد سیستم شود

و تأخیر محدود حاصل می‌شود. نشانه‌ها در یک نرخ ثابت ρ تولید می‌شوند و بسته‌ها بعد از خارج شدن تعداد نشانه‌های موردنیاز از سطل نشانه، وارد شبکه می‌شوند. در مورد تعداد بسته‌هایی که می‌توانند بافر شوند محدودیتی وجود ندارد. اما تعداد نشانه‌های موجود در سطل حداکثر σ بیت می‌باشد. علاوه بر محدودیتی که سرعت ρ در تولید نشانه و خروج داده از صف ایجاد می‌کند، ترافیک ورودی برای ترک کردن سطل نیز دارای حداکثر سرعت مجاز C می‌باشد که در آن $C > \rho$ است. محدودیت اعمال شده توسط سطل نشانه به این صورت است: اگر $A_i(t,\tau)$ مقدار جریان محاوره iم باشد که در بازه $(t,\tau]$ سطل نشانه را ترک می‌کند و وارد شبکه می‌شود، آنگاه برای هر محاوره:

$$A_i(\tau,t) \le \min\{(t-\tau)C_i, \sigma_i + \rho_i(t-\tau)\}, \forall t \ge \tau \ge 0 \qquad (۹-۳)$$

که آنرا به‌صورت $A_i \sim (\sigma_i, \rho_i, C_i)$ نشان می‌دهیم. $l_i(t)$ مقدار نشانه‌های باقی مانده در سطل در لحظه t می‌باشد. فرض می‌کنیم که محاوره‌ها در لحظه ۰ با ماکزیمم نرخ نشانه‌ی موجود در سطل شروع به خروج بسته‌ها می‌کنند. اگر $K_i(t)$ تعداد کل نشانه‌های پذیرفته شده در بازه $(0,t]$ در سطل محاوره iم باشند (که البته شامل سطل پر نشانه که محاوره iم با آن شروع کرده نمی‌باشد. و همچنین شامل نشانه‌هایی که می‌رسند و سطل پر، ظرفیت آن‌ها را ندارد، نیز نمی-باشد). آنگاه :

$$K_i(t) = \min_{0 \le \tau \le t}\{A_i(0,\tau) + \rho_i(t-\tau)\} \qquad (۱۰-۳)$$

بنابراین برای همه‌ی $\tau \le t$:

$$K_i(t) - K_i(\tau) \le \rho_i(t-\tau) \qquad (۱۱-۳)$$

حال می‌توان $l_i(t)$ را به‌صورت زیر تعریف کرد:

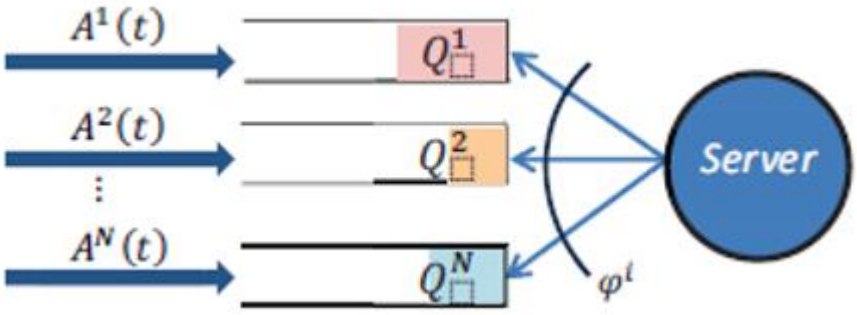
$$l_i(t) = \sigma_i + K_i(t) - A_i(0,t) \qquad (۱۲-۳)$$

از رابطه ی (۳-۱۱) و (۳-۱۲) نامساوی مفید زیر به دست می‌آید:

$$A_i(\tau, t) \leq l_i(\tau) + \rho_i(t-\tau) - l_i(t) \qquad (۱۳-۳)$$

۳-۲ سیستم‌های صف Fair Queuing

در سیستم‌های صف FQ، هدف آن است که ظرفیت سرویس‌دهی سرور متناسب با سهم هریک از محاوره‌های فعال در سیستم، بین آن‌ها تقسیم شود. شکل (۳-۳). برای این منظور، پارامترهای φ^i که $\varphi^i \geq 0$ و $\sum_i \varphi^i = 1$ است، به عنوان سهم هر یک از محاوره‌ها از ظرفیت سرویس‌دهی سرور تعریف می‌شود.



شکل ۳-۳ نمایش ساختار یک سیستم FQ

برای برقرار بودن خاصیت پایستکاری کار[1] در سیستم، فرض می‌شود که در غیاب یک محاوره، سهم آن به‌صورت متناسب بین محاوره‌های دیگر تقسیم شود. در این مبحث، یک محاوره را در یک لحظه غائب گوییم هرگاه داده‌ای (بسته‌ای) از آن جهت ارسال در سیستم موجود نباشد. به‌طور متناظر، یک محاوره را در یک لحظه فعال و یا انباشته گوییم هرگاه داده‌ای از آن در سیستم منتظر ارسال باشد. بدین ترتیب، مجموعه محاوره‌های انبااشته در لحظه‌ی t را با $B(t)$ نشان می-دهیم.

[1] Work conserving

در ادامه ضمن معرفی انواع روش‌های FQ ارائه‌شده، با ارائه‌ی معیارهایی به بررسی کارآئی آن‌ها خواهیم پرداخت.

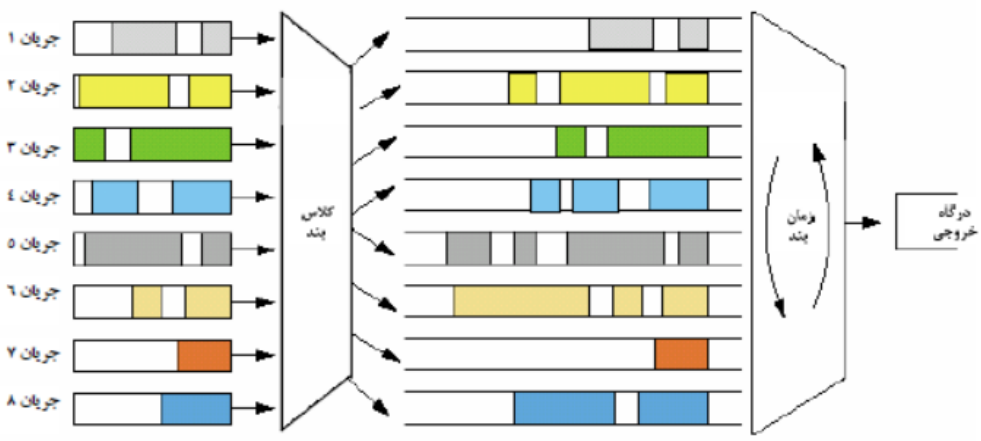
۳-۲-۱ ساختار صف‌بندی FQ

این روش توسط جان نگل[1] در سال ۱۹۸۷ پیشنهاد گردید. FQ، زیربنایی برای کلاسی از الگوهای زمان‌بندی است که:

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

۲- از طرف پهنای باند بیشتر از تسهیم منصفانه، 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، پیش‌بینی‌شده است ملحق می‌شوند. سپس یک بسته در هر RR سرویس‌دهی می‌شود. همچنین صف‌های خالی در نظر گرفته نمی‌شوند. به FQ، صف‌بندی برمبنای جریان نیز گفته می‌شود. در شکل (۴-۳) مثالی از عملکرد الگوریتم FQ آورده شده است:



شکل ۳-۴- مثالی از سیستم FQ

[1] John Nagle

ساختار FQ دارای مزایای زیر می‌باشد:

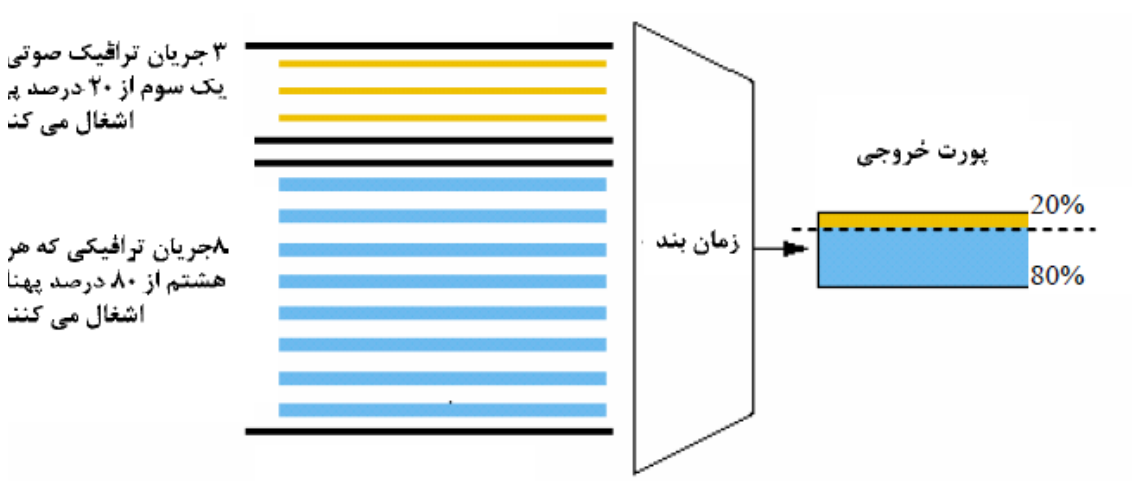
- با توجه به اینکه هر جریان، به‌طور مجزا و در صفی مخصوص به خودش قرار دارد یک جریان به‌شدت انفجاری یا مختلف نمی‌تواند باعث کاهش کیفیت سرویس تحویلی به سایر جریان‌ها شود و اگر جریانی سعی در مصرف بیشتری از مقدار تسهیم منصفانه‌اش داشته باشد فقط صف مربوط به همان جریان، تأثیر گرفته و اثری بر عملکرد سایر صف‌ها ندارد.

محدودیت‌های این روش عبارت‌اند از:

- این صف‌بندی به‌صورت نرم‌افزاری (نه سخت‌افزاری) پیاده‌سازی می‌شود. این محدودیت، باعث کاهش سرعت کار می‌شود.
- هدف از FQ تخصیص مقدار پهنای باند مساوی به تمام جریان‌ها در هرلحظه از زمان است و برای پشتیبانی از جریان‌های دارای پهنای باند مختلف، طراحی نشده است.
- اگر تمام بسته‌ها در تمام صف‌ها دارای اندازه یکسانی باشند، FQ برای تمام جریان‌ها پهنای باند یکسانی را مهیا می‌کند. در غیر این صورت، جریان‌هایی که اغلب دارای بسته‌های بزرگی هستند سهم بیشتری از پهنای باند درگاه خروجی را در مقایسه با جریان‌هایی که دارای بسته‌های کوچک هستند به دست می‌آورند.
- FQ به ترتیب ورود بسته‌ها حساس است. اگر بلافاصله بعداز 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 به‌راحتی می‌توان ترافیک شبکه را به جریان‌های دلخواه، کلاس‌بندی کرد، اما این مسئله، در شبکه‌های IP آن‌قدر ساده نیست.
- FQ را نمی‌توان در مسیریاب مرکزی استفاده کرد. چراکه مسیریاب مرکزی باید بتواند برای هر درگاه، از هزاران یا ده‌ها هزار صف مجزا پشتیبانی کند. این مسئله، در شبکه‌های

IP بزرگ باعث افزایش پیچیدگی مدیریت سرآیندها شده و درنتیجه می‌تواند اثر معکوسی بر مقیاس‌پذیری FQ داشته باشد.

- اگر n صف را در نظر بگیرید به هر صف، 1/n پهنای باند درگاه خروجی تخصیص داده می‌شود. اگر تعداد صف‌ها از n به n+1 تغییر کند مقدار پهنای باند تخصیص داده‌شده به هر صف، از 1/n به 1/(n+1) پهنای باند درگاه خروجی به تعدادی از کلاس‌های سرویس مختلف تقسیم می‌شود. هر کلاس سرویس، دارای درصد از پیش تعیین‌شده‌ای از پهنای باند درگاه خروجی هست که توسط کاربر تعیین می‌شود. سپس FQ در داخل بلاک پهنای باندی که برای هر کلاس سرویس، تخصیص داده‌شده به‌کاربرده می‌شود. درنتیجه تمام جریان‌های تعیین‌شده برای یک کلاس سرویس، سهم منصفانه‌ای از پهنای باند مجموع تعیین‌شده برای آن کلاس سرویس مشخص را به دست می‌آورند. در شکل (۳-۵) مثالی از عملکرد صف‌بندی FQ آورده شده است.



شکل ۳-۵-مثالی از صف‌بندی FQ

در شکل فوق فرض شده که فقط دو سرویس، از یک درگاه خروجی استفاده کنند. بدین‌صورت که به VoIP، ۲۰ درصد و به سایر جریان‌های ترافیکی، ۸۰ درصد پهنای باند سرور خروجی تخصیص داده‌شده است. در مدل FQ بر مبنای کلاس، به هر جریان VoIP، نسبتی مساوی از ۲۰ درصد از پهنای باند سرور خروجی، و به هر جریان ترافیکی دیگر نسبتی مساوی از ۸۰ درصد

از پهنای باند سرور خروجی تخصیص می‌یابد. بنابراین جریان‌های VOIP برخوردی با سایر جریان‌ها ندارند و بالعکس.

۳-۲-۲ روش زمان‌بندی WFQ

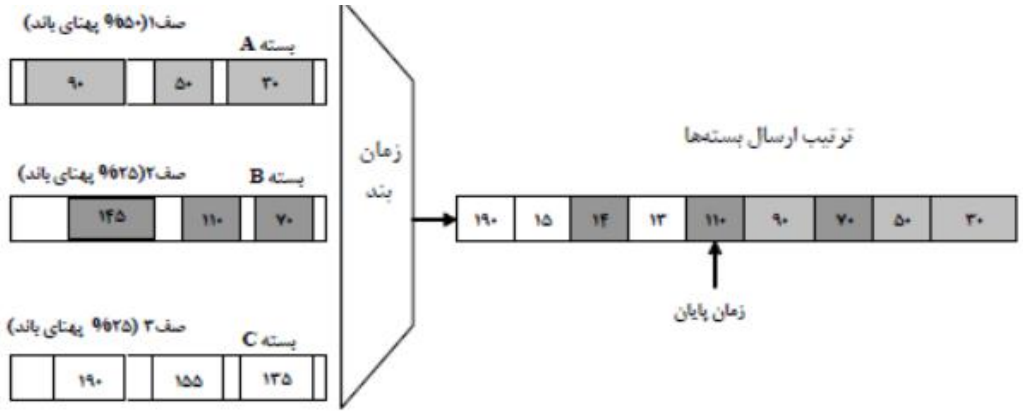
این روش در سال ۱۹۸۹ ارائه گردید[۴]. WFQ اساس کلاسی از الگوریتم‌های زمان‌بندی است که به منظور رفع محدودیت‌های FQ طراحی شده‌اند. WFQ به هر صف وزنی می‌دهد که مشخص کننده درصد متفاوتی از پهنای باند درگاه خروجی است که به آن تخصیص می‌یابد. از این طریق WFQ قادر است جریان‌هایی با نیازهای مختلف پهنای باند را پشتیبانی نماید.

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 ساختار زمان‌بندی صف می‌افزاید. دلیل اصلی اینکه پیاده‌سازی ساختار زمان-بندی صف در شبکه‌های ATM بر پایه‌ی سلول‌های با طول ثابت بسیار ساده‌تر از پیاده‌سازی آن‌ها در شبکه‌های IP بر پایه بسته با طول متغیر می‌باشد نیز همین امر است.

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. به‌عبارت‌دیگر، WFQ، GPS را شبیه‌سازی کرده و نتیجه این شبیه‌سازی را به‌منظور تعیین ترتیب سرویس بسته‌ها به کار می‌برد.

در شکل (۳-۶)، 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 زمان‌بند WFQ بسته با کوچک‌ترین (زودترین) زمان پایان را به‌عنوان بسته بعدی برای انتقال به درگاه خروجی انتخاب می‌کند. به‌عنوان‌مثال 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 پایان می‌یابد. تخصیص وزن مناسب به صف‌ها به زمان‌بند WFQ امکان می‌دهد که دو یا چند بسته متوالی از یک صف را انتقال دهد.



شکل ۳-۶- مثالی از زمان‌بند WFQ (ارسال بسته بر اساس زمان پایان)

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 نظارت‌کننده 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 به‌صورت نرم‌افزاری انجام‌شده و نه سخت‌افزاری. این امر کاربرد WFQ را برای واسط‌های با سرعت پایین در لبه‌های شبکه محدود می‌کند. جریان‌های بدرفتاری در یک کلاس سرویس می‌توانند بر روی کارایی سایر جریان‌های موجود در آن کلاس تأثیر 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، هنگامی‌که نیاز به پشتیبانی تعداد زیادی کلاس سرویس است بر روی قیاس‌بندی WFQ آن تأثیر منفی می‌گذارد. و نهایتاً باوجود اینکه محدوده تأخیر پشتیبانی شده توسط WFQ بهتر از زمان‌بندی‌های دیگر است، اما این محدوده هنوز هم بزرگ می‌باشد.

۳-۲-۳ ساختار صف‌بندی GPS

در روش سرویس‌دهی GPS [۲]، در هرلحظه قدرت سرویس‌دهی سرور بین محاوره‌های فعال در آن لحظه توزیع می‌شود. به نحوی که برای هر بازه‌ی زمانی (t_1, t_2) که در طی آن $B(t)$ تغییر ننماید، داشته باشیم:

$$\frac{w^i(t_1, t_2)}{\varphi^i} = \frac{w(t_1, t_2)}{\sum_{j \in B(t_1,t_2)} \varphi^j} \qquad (۱۴-۳)$$

که در رابطه‌ی فوق، $w(t_1, t_2)$ کل کار انجام شده توسط سرور در بازه‌ی (t_1, t_2) و $w^i(t_1, t_2)$ کار ارائه‌شده به محاوره‌ی i در این بازه است. به علاوه، $B(t_1, t_2)$ بیانگر مجموعه محاوره‌های فعال در سرتاسر بازه‌ی زمانی (t_1, t_2) است، به‌طوریکه مجموعه محاوره‌های فعال $B(t)$ در هر لحظه‌ی $t \in (t_1, t_2)$ ثابت است.

بر اساس رابطه‌ی (۳-۱۴)، در یک سیستم با سرویس‌دهی GPS مقدار کار نرمالیزه شده به سهم از دید هر محاوره‌ی فعال، مقدار ثابت و یکسانی دارد و برابر مقدار کار نرمالیزه‌ی انجام شده بر روی ترافیک محاوره‌های فعال می‌باشد. به علاوه، از آنجا که با گذر زمان مقدار کار نرمالیزه‌ی انجام شده در سیستم را اصطلاحاً زمان مجازی[1]، $V(t)$ ، می‌نامند. برای هر بازه‌ی زمانی (t_1, t_2) که در طی آن $B(t)$ تغییر ننماید، داریم:

$$v(t_1, t_2) \triangleq \frac{w(t_1, t_2)}{\sum_{j \in B(t_1, t_2)} \varphi^j} \qquad (۳-۱۵)$$

۳.۲.۳.۱ روش زمان‌بندی *PGPS*

در سرویس‌دهی GPS، جریان ترافیک به‌صورت یک جریان سیال فرض می‌شود و ماهیت بسته‌ای بودن ترافیک نادیده گرفته می‌شود. لذا جهت پیاده‌سازی عملی یک سیستم مبتنی بر ارسال و دریافت بسته که در آن بسته‌ها به‌صورت non-preemptive سرویس داده می‌شوند؛ لازم است سیستم فرضی فوق به‌صورت تقریبی پیاده‌سازی شود.

یک روش برای پیاده‌سازی تقریبی این سیستم با فرض بسته‌ای بودن ترافیک، آن است که بسته-هایی که در یک لحظه داخل سیستم هستند، به همان ترتیبی سرویس داده شوند که در سیستم GPS معادل سرویس آن‌ها تمام می‌شود. لم زیر بیانگر مشخص بودن ترتیب بسته‌ها در سیتم GPS معادل است. در اینجا منظور از سیستم GPS معادل، سیستم فرضی است که همان جریان‌های ترافیکی عیناً به آن وارد شود و در آن به جریان‌های ترافیکی مختلف با نادیده گرفتن ماهیت بسته‌ای بودن آن‌ها سرویس داده شود [۲].

لم ۳-۴ : ترتیب پایان سرویس بسته‌های حاضر در سیستم در لحظه‌ی τ در سیستم GPS معادل، مستقل از بسته‌هایی است که بعد از لحظه‌ی τ وارد می‌شوند. (اثبات: به مرجع [۲] مراجعه شود)

[1] virtual time

باتوجه به اینکه مقدار کار نرمالیزه شده به سهم هر محاوره فعال در سیستم GPS معادل برابر تابع زمان مجازی است و با توجه به خاصیت یکنوایی $v(t)$ بر حسب زمان، می‌توان ترتیب زمان پایان سرویس هر بسته در سیستم GPS معادل را با مرتب کردن زمان مجازی اتمام سرویس بسته‌ها مشخص نمود.

$$F_p^i = S_p^i + \frac{L_p^i}{\varphi^i} \qquad (۱۶-۳)$$

$$S_p^i = \max\{v(a_p^i), F_{p-1}^i\}$$

که در روابط فوق S_p^i ، بیانگر زمان مجازی سیستم در زمان شروع سرویس بسته pام از محاوره iام است. بدین ترتیب، پس از پایان سرویس بسته‌ی فعلی موجود در سرور، از بین بسته‌های موجود در صف بسته‌ای انتخاب می‌شود که کوچک‌ترین مقدار F_p^i را داشته باشد [۲].

لازم به ذکر است که جهت محاسبه‌ی S_p^i ، در شرایطی که محاوره‌ی iم فعال باشد، زمان مجازی شروع سرویس بسته‌ی pام از محاوره‌ی i ، برابر زمان مجازی خاتمه‌ی سرویس بسته‌ی قبلی از این محاوره خواهد بود. در شرایطی که محاوره‌ی iم برای مدتی غیر فعال باشد، سهم این محاوره از سرویس این بازه، به محاوره‌های دیگر اختصاص می‌یابد. بدین ترتیب، با ورود بسته‌ی Pام و فعال شدن محاوره‌ی i، S_p^i ، $v(a_p^i)$ خواهد شد.

از آنجا که WFQ در ابتدا در سال ۱۹۸۹ ارائه‌شده است، توسعه‌های زیادی بر آن ایجاد شده است که هریک سعی در ایجاد تعادل بین پیچیدگی، دقت و کارایی آن دارند. برخی از این توسعه‌ها در ادامه توضیح داده شده‌اند.

۳-۲-۴ روش زمان‌بندی SCFQ

جهت پیاده‌سازی روش زمان‌بندی PGPS نیاز است که تابع زمان مجازی متناظر با سیستم GPS معادل، محاسبه شود. به‌عبارت‌دیگر، برای اجرای روش زمان‌بندی PGPS لازم است

سیستم GPS معادل شبیه‌سازی شود. گلستانی در [۳] روشی را پیشنهاد کرده است که بدون نیاز به هیچ گونه محاسبه‌ی اضافی، تابع زمان مجازی تقریب زده می‌شود. در ادامه‌ی بحث سعی داریم که این الگوریتم را به عنوان اصلی‌ترین الگوریتم دسته اول که در بخش ۱-۷ بیان شد، به‌طور دقیق‌تر مورد بررسی قرار دهیم.

۳.۲.۴.۱ بررسی دقیق‌تر الگوریتم *SCFQ*

همان‌طور که در بخش‌های پیشین ذکر شد، SCFQ از یک تابع پله‌ای به عنوان تابع زمان مجازی سیستم استفاده می‌کند. مقدار این تابع در هر لحظه، برابر برچسب بسته در حال سرویس در آن لحظه است. به‌طور خلاصه روند اجرای الگوریتم SCFQ را می‌توان به‌صورت زیر بیان کرد:

۱- هربسته هنگام ورود به صف به وسیله زمان مجازی پایان سرویس محاسبه شده برای آن برچسب می‌خورد. برچسب متناظر با زمان خاتمه سرویس بسته‌ی موجود در داخل سرور قرار داده می‌شود. درنتیجه، $\hat{v}(a_p^i)$ به عنوان تقریبی از $v(a_p^i)$ به دست می‌آید.

۲- بدین ترتیب محاسبه‌ی برچسب بسته‌ها مشابه روش PGPS ، به‌صورت زیر انجام می‌شود [۳]:

$$F_p^i = S_p^i + \frac{L_p^i}{\varphi^i} \qquad (۱۷-۳)$$

$$S_p^i = \max\{\hat{v}(a_p^i), F_{p-1}^i\}$$

۳- در هنگام انتخاب بسته برای سرویس، بسته‌ای که برچسب کوچکتری دارد انتخاب می‌شود.

۴- تابع زمان مجازی سیستم در هر زمان t برابر برچسب سرویس بسته‌ای است که در لحظه t در حال سرویس است. به‌عبارت‌دیگر،

$$v(t) = F_l^j, \quad S_l^j < t \leq d_l^j \qquad (۱۸-۳)$$

که S_l^j و d_l^j به ترتیب زمان‌هایی هستند که بسته p_l^j (بسته در حال سرویس) سرویسش را آغاز و به پایان می‌برد.

۵- پس از پایان هر دوره کار، الگوریتم شروع مجدد[1] خواهد شد و $v(t)$ برابر صفر قرار می‌گیرد تا دوره کار بعدی آغاز شود و همین مراحل برای آن هم اجرا گردد.

در ادامه به بررسی رفتار عدالت و تأخیر الگوریتم پرداخته می‌شود و معیار عدالت SCFQ و پارامتر تعویق آن محاسبه می‌شود.

۳.۲.۴.۲ تحلیل عدالت الگوریتم *SCFQ*

ابتدا متغیرها و پارامترهایی را که در اثبات موردنیاز است، معرفی می‌کنیم. سیستمی را که با آن مواجه هستیم، یک سیستم با چند صف ورودی و یک سرویس‌دهنده است. نرخ سرویس سرویس‌دهنده برابر مقدار ثابت C است. تعداد بیشترین صف‌های ممکن در ورودی برابر K فرض می‌شود. L_k^i اندازه بسته iام صف kام و a_k^i زمان حقیقی ورود آن و d_k^i زمان حقیقی خروج آن از سیستم است. $Q_K(t)$ طول صف kام در لحظه t و $v(t)$ تابع زمان مجازی سیستم است. همچنین فرض می‌کنیم در نقاط پایان سرویس بسته، ورود بسته نداریم و سرویس یک بسته زمانی پایان می‌یابد که آخرین بیت از آن بسته سرویس گرفته باشد. در طی یک دوره کار که در لحظه $t=0$ آغاز شده، $u_K(t)$ به عنوان سرویس نرمالیزه شده از دست رفته برای صف kام به‌صورت زیر تعریف می‌شود:

$$u_k(0) = 0 \qquad , \qquad k \in K \qquad (۱۹-۳)$$

[1] Restart

$$u_k(t_1, t_2) = \begin{cases} 0 & k \in B(t_1, t_2) \\ v(t_1, t_2) & k \in A(t_1, t_2) \end{cases}$$

در رابطه بالا $k \in B(t_1, t_2)$ نشان‌دهنده غیر خالی بودن صف k و $k \in A(t_1, t_2)$ بیان‌کننده خالی بودن صف k در بازه (t_1, t_2) است. به‌عبارت‌دیگر اگر صف غیر خالی باشد، سرویسی را از دست نمی‌دهد. اگرچه ممکن است سرویس آن نسبت به سایر صف‌ها کمی عقب یا جلو بیفتد. ولی در زمانی که خالی است به اندازه مقدار پیشرفت زمان مجازی سیستم، سرویس از دست می‌دهد. اکنون تابع زمان مجازی سیستم را به‌صورت زیر تعریف می‌کنیم و آن را به عنوان معیاری برای تعیین سطح عدالت سیستم به کار میگیریم. فرض کنید که $w_K(t)$ سرویس نرمالیزه شده‌ای (نرمالیزه نسبت به نرخ تخصیص داده شده به آن) باشد که به صف Kم داده شده است.

$$v_k(t) = u_k(t) + w_k(t) \quad , \quad k \in K \qquad (۲۰-۳)$$

تابع زمان مجازی صف k به‌صورت مجموع سرویس نرمالیزه شده از دست رفته و دریافت شده محاسبه می‌شود. درواقع هدف از این کار این است که نشان دهیم در تعیین عدالت یک سیستم تنها سرویس دریافت شده نباید مورد توجه قرار گیرد، بلکه سرویس از دست رفته در بازه‌های خالی بودن صف نیز باید به آن‌ها اضافه شود. درواقع $v_K(t)$ بیانگر پیشرفت کار یک صف است که برابر مقدار سرویس نرمالیزه دریافت شده در بازه‌های غیر خالی بودن آن است و در بازه‌های خالی بودنش برابر مقدار سرویس نرمالیزه‌ای است که از دست می‌دهد، یعنی همان پیشرفت تابع زمان مجازی سیستم در آن بازه. بنابراین عقب ماندگی سرویس[1] یک صف را می‌توانیم به‌صورت زیر تعریف کنیم:

[1] Service lag

$$\delta_k(t) = v(t) - v_k(t) \quad , \quad k \in K \tag{۲۱-۳}$$

با توجه به مطالب مذکور می‌توان نتیجه گرفت که:

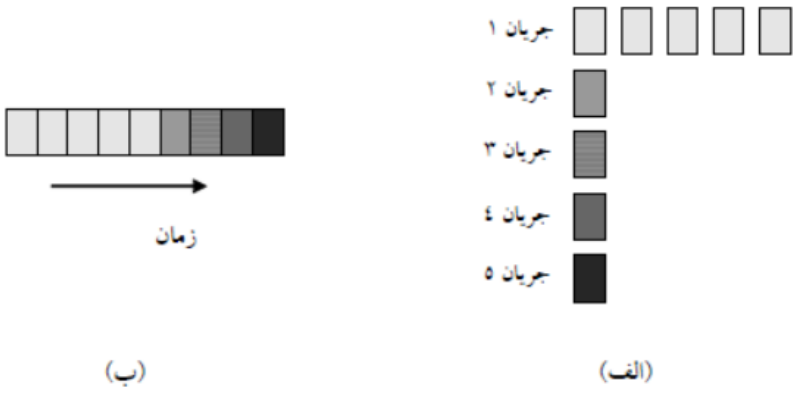
$$v_k(0) = 0 \quad , \quad k \in K \tag{۲۲-۳}$$

$$v_k(t_1, t_2) = \begin{cases} w_k(t_1, t_2) & k \in B(t_1, t_2) \\ v(t_1, t_2) & k \in A(t_1, t_2) \end{cases}$$

۳-۲-۵ روش زمان‌بندی WF^2Q

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

در شکل (۳-۷) (الف)، ۵ جریان از یک اتصال به‌صورت 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. درحالی‌که ۴ جریان دیگر فقط یک بسته در زمان صفر می‌فرستند. همان‌طور که در شکل (۳-۷) (ب) نشان داده‌شده است، اگر سرویس گر WFQ باشد، ابتدا ۵ بسته جریان اول پشت سرهم ارسال می‌شوند و سپس نوبت به بسته‌های جریان‌های بعد می‌رسد.



شکل ۳-۷- الف) ترتیب ورود بسته‌ها ب) ترتیب سرویس‌دهی در WFQ

این نحوه عملکرد ممکن است منجر به بروز ازدحام در شبکه شود و با رفتار ایده آل GPS که در شکل (۳-۸)(ب) نمایش داده‌شده است تفاوت دارد.

الگوریتم WF^2Q که در سال ۱۹۹۶ توسط بنت و ژانگ[1] ارائه گردید، سعی در رفع این مشکل و ارائه تخمینی دقیق‌تر برای GPS دارد. در این الگوریتم، هنگامی‌که بسته‌ای برای سرویس گیری در زمان T انتخاب می‌شود، به‌جای انتخاب بسته از بین همه بسته‌های موجود در سرویس گر به‌مانند WFQ ، سرویس گر تنها بسته‌هایی را در نظر می‌گیرد که در سیستم GPS مربوط، در زمان T سرویس گیری خود را آغاز کرده باشند. به‌عبارت دیگر $\left\{P_I^K \middle| B_{I,GPS}^K < T < B_{I,WFQ}^K\right\}$ که $B_{I,S}^K$ زمان شروع سرویس بسته K ام از جریان Iام تحت سرویس گر S است و آنگاه از میان این بسته‌ها، بسته با کوچک‌ترین زمان پایان را برای سرویس‌دهی انتقال می‌دهد.

بر اساس این عملکرد، ترتیب سرویس گیری بسته‌ها در مثال فوق تحت سرویس‌گر WF^2Q به‌صورت شکل (۳-۸) (الف) است.

WF^2Q تخمین بهتری برای GPS می‌باشد این روش، سرویس تقریباً 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 هست. به‌منظور 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.

[1] Bennett &Zhang.h

در روش WF^2Q در هر لحظه از زمان از بین بسته‌های موجود در داخل سیستم، انتخاب از بین مجموعه‌ای از بسته‌ها انجام می‌شود که در سیستم GPS معادل، سرویس آن‌ها شروع شده است. این مجموعه از بسته‌ها را اصطلاحاً بسته‌های مجاز[1] می‌نامند. از بین بسته‌های مجاز، بسته‌ای جهت سرویس‌دهی انتخاب می‌شود که کوچک‌ترین زمان خاتمه سرویس را داشته باشد [۵].

لم ۳-۵ : در روش زمان‌بندی WF^2Q به‌صورت توصیف شده در بالا، کار پایستار خواهد بود.

اثبات: به [۵] مراجعه شود. ■

شکل ۳-۸- الف) ترتیب سرویس‌گیری در WF^2Q ب) ترتیب سرویس‌گیری در GPS

جهت پیاده‌سازی روش زمان‌بندی WF^2Q لازم است که تابع زمان مجازی در سیستم GPS معادل محاسبه شود. در الگوریتم WF^2Q+ جهت کاهش پیچیدگی موجود در الگوریتم زمان-بندی WF^2Q ، تابع زمان مجازی در هر لحظه به‌صورت تقریبی زیر محاسبه می‌شود [۵]:

$$v_{WF^2Q+}(t+\tau) \qquad (۲۳-۳)$$
$$= \max\left\{v_{WF^2Q+}(t) + w(t,t+\tau), \min_{l\in B(t)} S^i_{h_{i(t)}}\right\}$$

[1] Eligible

که در رابطه‌ی فوق، $W(t, t+\tau)$ مقدار کار انجام شده توسط سرور در بازه‌ی $(t, t+\tau)$ ، $B(t)$ مجموعه محاوره‌های فعال در زمان t و $h_i(t)$ بسته‌ی موجود در ابتدای صف محاوره‌ی iام است. در این روش جهت ساده‌تر کردن پیاده‌سازی، نظیر هر بسته‌ی P که به ابتدای صف می‌رسد، برچسب‌های S^i و F^i به‌صورت زیر محاسبه شده و به روز می‌شوند. بدین ترتیب، نظیر هر محاوره تنها یک دسته از مقادیر S^i و F^i تعریف می‌شود.

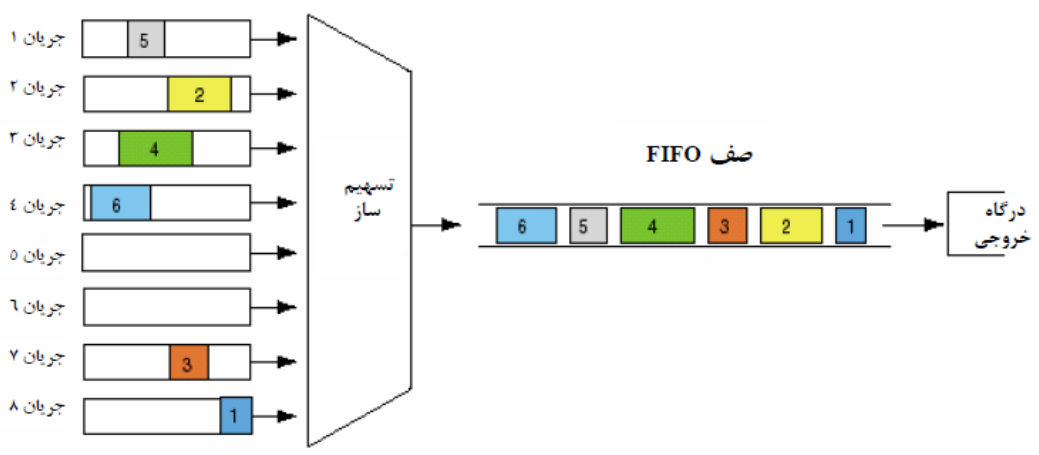
$$S^i = \begin{cases} F^i & \text{if } Q_i(a_p^i) \neq 0 \\ \max\{F^i, v(a_p^i)\} & \text{if} Q_i(a_p^i) = 0 \end{cases} \quad (۲۴-۳)$$

$$F^i = S^i + \frac{L_p^i}{C_i} \quad (۲۵-۳)$$

در ادامه به بیان ۴ الگوریتم دیگر از الگوریتم‌های نظم‌بندی ترافیک می‌پردازیم که اساس تحقیق در این پایان‌نامه می‌باشند.

۳-۲-۶ روش زمان‌بندی FCFS

این الگوریتم، یکی از ساده‌ترین ساختارهای زمانبندی صف است. در صف‌بندی FCFS ، 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) نیز گفته می‌شود. در شکل (۳-۹) مثالی از صف‌بندی FCFS آورده شده است.



شکل ۳-۹- مثالی از صف‌بندی FCFS

ساختار FCFS دارای مزایای زیر است:

- در مقایسه با سایر روش‌ها این صف‌بندی بار محاسباتی بسیار کمی در مسیریاب‌های مبتنی بر نرم افزار دارد.
- در یک صف FCFS ترتیب بسته‌ها به هم نمی‌خورد و حداکثر تأخیر، با حداکثر اندازه صف تعیین می‌شود.
- تا زمانی که اندازه صف کوتاه باشد، صف FCFS عملکرد خوبی دارد چراکه در هر نود شبکه، تأخیر صف‌بندی زیادی وجود ندارد.

همچنین این مکانیزم دارای محدودیت‌های زیر است:

- نمی‌تواند یک کلاس ترافیک را به‌طور جداگانه‌ای از سایر کلاس‌های ترافیکی سرویس داد.
- با اینکه تأخیر صف‌بندی متوسط برای تمام جریان‌ها با افزایش ازدحام، زیاد می‌شود، تأخیر مشاهده شده برای تمام جریان‌ها با هم مساوی است. درنتیجه صف‌بندی FCFS می‌تواند برای کاربردهای بلادرنگی که از یک صف FCFS عبور می‌کنند، موجب افزایش اتلاف، تأخیر و تغییرات تأخیر بسته‌ها شود و از آنجا که این کاربردها به این مسئله، حساس هستند، موجب مشکل می‌شود.

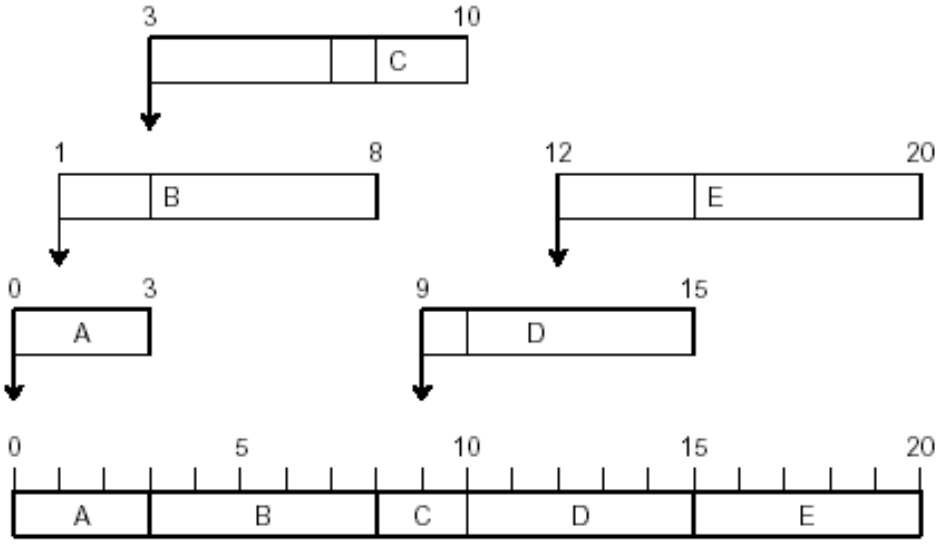
- در زمان‌هایی که شبکه با ازدحام مواجه می‌شود، صف‌بندی FCFS بیشتر به نفع جریان-های UDP عمل می‌کند تا جریان‌های TCP. وقتی که بخاطر ازدحام، تلف بسته‌ای مشاهده شد، کاربردهای بر مبنای TCP نرخ ارسالشان را کاهش می‌دهند، اما کاربردهای بر مبنای UDP به تلف بسته‌ها بی‌توجه بوده و با نرخ همیشگی و معمول خود، به ارسال بسته‌ها ادامه می‌دهند.
- یک جریان انفجاری می‌تواند کل فضای بافر یک صف FCFS را اشغال کند و باعث عدم سرویس‌دهی مناسب به سایر جریان‌ها شود.

وقتی که از این الگو استفاده می‌شود، اغلب دو صف روی درگاه خروجی استفاده می‌شود. این صف-ها عبارتند از:

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

در شکل (۳-۱۰) یک مثال عددی از نحوه‌ی سرویس‌دهی این الگوریتم بیان می‌شود.

Process name	Arrival time	Service required	Start time	Finish time	T	M	P
A	0	3	0	3	3	0	1.0
B	1	5	3	8	7	2	1.4
C	3	2	8	10	7	5	3.5
D	9	5	10	15	6	1	1.2
E	12	5	15	20	8	3	1.6
Mean					6.2	2.2	1.74



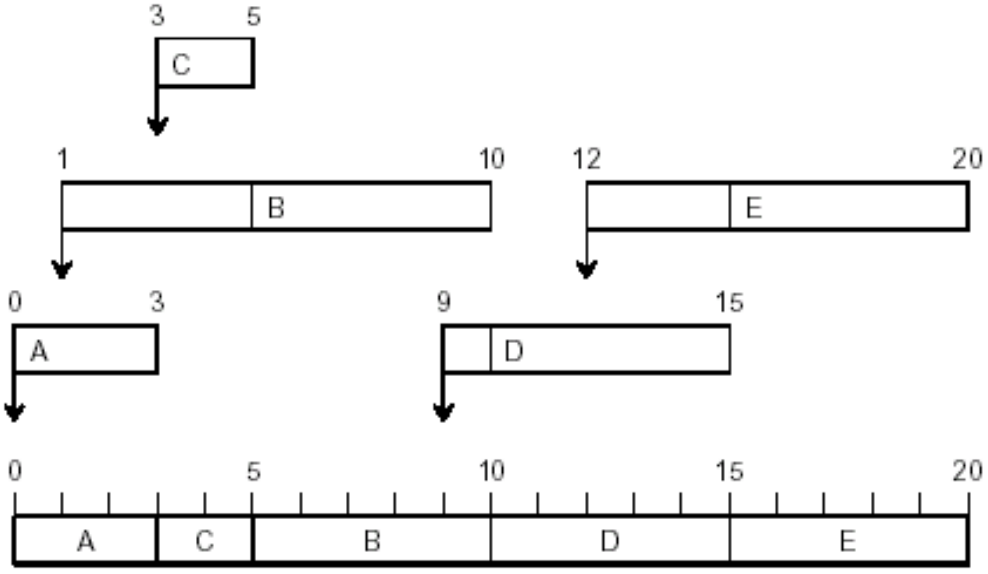
شکل ۳-۱۰- مثال عددی از نحوه‌ی سرویس‌دهی FCFS

۳-۲-۷ روش زمان‌بندی SPN

یکی دیگر از چهار الگوریتم زمان‌بندی مورد نظر ما در این پایان نامه، الگوریتم SPN است. SPN نیز مانند FCFS یک الگوریتم انحصاری است که در آن جریان‌ها با توجه به کمترین زمان سرویس‌گیری، مرتب می‌شوند. به‌بیان‌دیگر، جریانی که مدت زمان سرویسش از بقیه کمتر است، در هر دوره در اولویت سرویس‌گیری می‌باشد. یکی از سختی‌های الگوریتم SPN ، دانستن و یا تخمین مدت زمان موردنیاز برای سرویس می‌باشد. خطری که در الگوریتم SPN جریان‌های داده را تحریک می‌کند، احتمال گرسنگی[1] برای بسته‌هایی با طول بزرگتر، هنگامی که بسته‌های با طول کم زیاد است، می‌باشد.

[1] starvation

Process name	Arrival time	Service required	Start time	Finish time	T	M	P
A	0	3	0	3	3	0	1.0
B	1	5	5	10	9	4	1.8
C	3	2	3	5	2	0	1.0
D	9	5	10	15	6	1	1.2
E	12	5	15	20	8	3	1.6
Mean					5.6	1.6	1.32



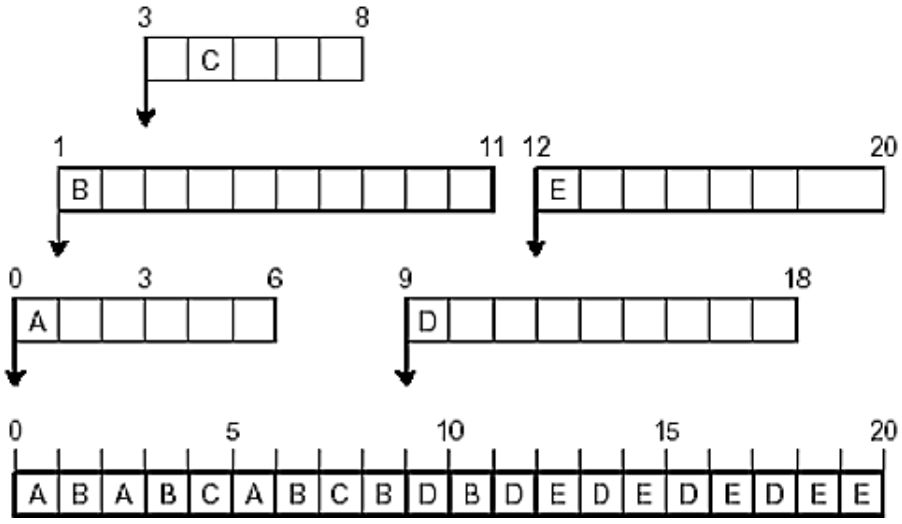
شکل ۳-۱۱- مثال عددی از الگوریتم SPN

۳-۲-۸ روش زمان‌بندی RR

در این روش[۴۴]، محور زمان به چارچوب‌هایی با طول حداکثر F تقسیم می‌شود. بسته‌های مربوط به جریان‌های مختلف به‌صورت چرخشی سرویس می‌گیرند. به‌علاوه ازآنجایی‌که 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. درصورتی‌که شمارنده یک ارتباط صفر شود، ارتباط، دیگر قادر به سرویس گیری نمی‌باشد. سادگی این الگوریتم پیاده‌سازی سریع آن را ممکن ساخته است. در شکل زیر یک مثال عددی از این الگوریتم با **q=1 و q=4 آورده شده است.**

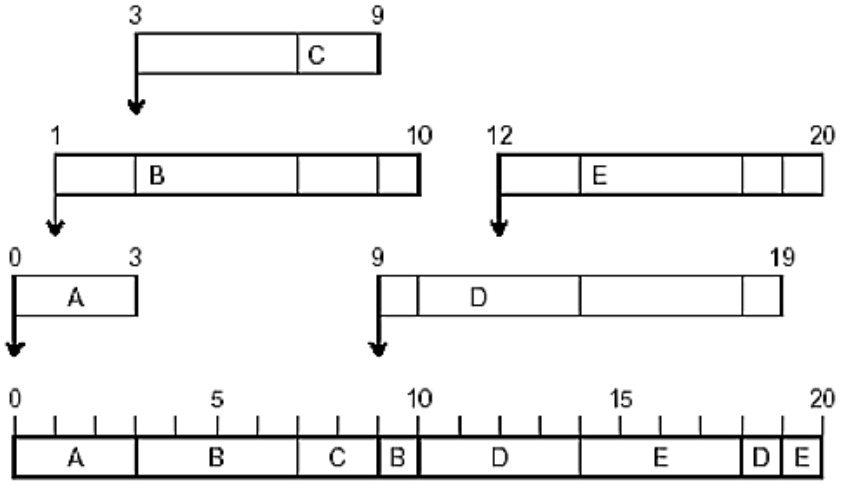
Process name	Arrival time	Service required	Finish time	*T*	*M*	*P*
A	0	3	6	6	3	2.0
B	1	5	11	10	5	2.0
C	3	2	8	5	3	2.5
D	9	5	18	9	4	1.8
E	12	5	20	8	3	1.6
Mean				7.6	3.6	1.98



شکل۳-۱۲- مثال عددی از الگوریتم RR با q=1

حل این مثال با q=4 به‌صورت زیر خواهد بود:

Process name	Arrival time	Service required	Finish time	*T*	*M*	*P*
A	0	3	3	3	0	1.0
B	1	5	10	9	4	1.8
C	3	2	9	6	4	3.0
D	9	5	19	10	5	2.0
E	12	5	20	8	3	1.6
Mean				7.2	3.2	1.88



شکل۳-۱۳- مثال عددی از الگوریتم RR با q=4

۳-۲-۹ روش زمان‌بندی DRR

DRR [۸]، یکی دیگر از الگوریتم‌های زمانبندی تک سروره می‌باشد که تنها با درجه پیچیدگی O(1) برای سرویس یک بسته کار می‌کند. در DRR ، مقدار نقصانی[1] به تعداد بایت‌هایی می‌گویند که صف اجازه دارد در هر زمانی که توسط زمان‌بند ملاقات می‌شود، ارسال کند. درواقع در پایان هر دوره، مقدار نقصانی اختلاف بین تعداد بایت‌هایی که می‌توانست در آن دوره ارسال شود و تعداد بایت‌هایی که در آن دوره ارسال شد، می‌باشد.

در الگوریتم DRR به هر صف i یک مقدار کوانتوم به اندازه Q_i در هر دوره داده می‌شود. یک نظم DRR با ظرفیت سرویس C در نظر بگیرید که می‌تواند حداکثر به F بایت اجازه دهد تا در یک دوره تقسیم شوند. آنگاه $\sum_{i=1}^{n} Q_i \leq F$. Q_i به‌طور غیر مستقیم میانگین نرخ سرویس در دراز مدت که جریان i می‌تواند بگیرد را نشان می‌دهد. یعنی:

$$\rho_i = {Qi \cdot C}/{F} \qquad (۲۶-۳)$$

[1] Deficit

مقدار اولیه D_i ، صفر در نظر گرفته می‌شود. $D_i + Q_i$ تعداد کل بایت‌هایی که صف i می‌تواند در یک دوره ارسال کند را محدود می‌کند. صف i ی که در سرویس قرار دارد تنها وقتی می‌تواند بسته‌ای ارسال کند که خالی نباشد و یا اینکه طول بسته‌ی بعدی آن بیشتر از $D_i + Q_i - S_i$ نباشد، که S_i تعداد بایت‌های قبلی است که در یک دوره از صف i ارسال شده است. این باعث می‌شود که مقدار نقصانی هیچ‌گاه منفی نباشد. هنگامی که صف به دلیل خالی بودن، قادر به ارسال بسته‌ای نیست، مقدار D_i صفر می‌شود؛ در غیر اینصورت D_i به مقدار $D_i + Q_i - S_i$ به روز می‌شود. آنگاه زمانبند به صف بعدی یعنی $i+1$ می‌رود.

از توضیحات فوق آشکار است که در پایان یک دوره، $0 < D_i < L_{max}$. چون در غیر اینصورت جریان i ، هنوز مجاز به ارسال یک بسته می‌باشد. برای جریان i ، $T_i[k,l]$ ، را مقدار سرویسی در نظر بگیرید که آن جریان از شروع دوره kم تا پایان دوره lم ($l \geq k$) می‌تواند دریافت کند. آنگاه $T_i[k,l]$ به قرار زیر می‌باشد[۱۶]:

$$T_i[k,l] = (l-k+1)Q_i + D_i^{k-1} - D_i^l \qquad (۲۷-۳)$$

که D_i^x مقدار شمارش‌گر نقصانی جریان i در پایان زمان دوره xم می‌باشد. برای ترافیک کلی که الگوریتم DRR می‌تواند سرویس دهد(برای همه جریان‌ها) از شروع دوره kم تا پایان دوره lم با $T[k,l]$ نمایش داده می‌شود و از رابطه‌ی زیر به دست می‌آید [۴۵]:

$$T[k,l] \leq (l-k+1)F + \sum_{j=1}^{n} D_j^{k-1} - \sum_{j=1}^{n} D_j^l \qquad (۲۸-۳)$$

که n تعداد جریان‌ها در سرور می‌باشد.

۳-۳ ضرورت وجود الگوریتم‌های چندسروره

مباحثی که تا کنون بیان شد، تاریخچه‌ای از به وجود آمدن الگوریتم‌های تک سروره و شامل مفاهیمی بود که برای مطالعه در این زمینه دارای اهمیت است. اما همان‌طور که می‌دانیم، امروزه شبکه‌های کامپیوتری دارای چندین رابط[1] یا همان سرور هستند. برای مثال تلفن‌های هوشمند معمولاً دارای رابط wifi,4G,3G می‌باشند و تعداد این رابط‌ها روز به روز در حال افزایش است. اگر تلفن بتواند همزمان از چندین رابط استفاده کند، مقدار بازده آن برای کاربرهایی مانند رشته-های ویدئویی و VOIP افزایش می‌یابد، اتصال آن مداوم شده و تأخیرش کم می‌شود. به‌طور مشابه سرورهای CDN اغلب دارای رابط‌های شبکه چندتایی و یا چندین مسیر مختلف به سرور می-باشند. بنابراین با روی کارآمدن این دسته از نیازها، بیان مطالب زیر را به عنوان مقدمه‌ای برای فهم بهتر مسائل فصل ۴ و ارائه سیستم مدل پیشنهادی ضروری می‌دانیم.

مؤسسه تحقیقات اخیراً سازوکاری را ارائه داده که در آن چندین سرور را با هم ادغام می‌کند تا یک سرور با بازده اتصال بالا به دست آورد [۴۶]،[۴۷]. باوجوداینکه این کار می‌تواند شروع خوبی باشد، اما بهتر است به جای بهره‌گیری ساده از ترکیب منطقی همه‌ی سرورهای در دسترس، به قابلیت انعطاف‌پذیری بیشتری دست پیدا کنیم. برای مثال یک تلفن هوشمند را در نظر بگیرید که از برنامه ارتباط تصویری در هنگام استفاده از مرورگر وب استفاده می‌کند. کانال صوتی ترجیح می‌دهد از سرورهای شبکه‌ای استفاده کند که تعویق پایین و اتصال مداوم‌تری دارند. در حالیکه ویدئو سرورهایی را ترجیح می‌دهد که هزینه پایین به همراه بازده بالا داشته باشند. این ایده منجر به پیشنهاد مدلی می‌شود که در آن برنامه‌های مختلف از زیر مجموعه‌های متفاوتی از سرورهای در دسترس استفاده کنند. علاوه بر این ممکن است سیاست‌های اولویتی را در استفاده از شبکه به کار گیرد. مثلاً از اتصال 4G تنها برای VOIP استفاده کند، در حالیکه از اتصال wifi برای ارسال اطلاعات بزرگ اما کم اهمیت بهره ببرد. چنین اولویت‌های مشابهی در سرور هم رخ می‌دهد. مثلا

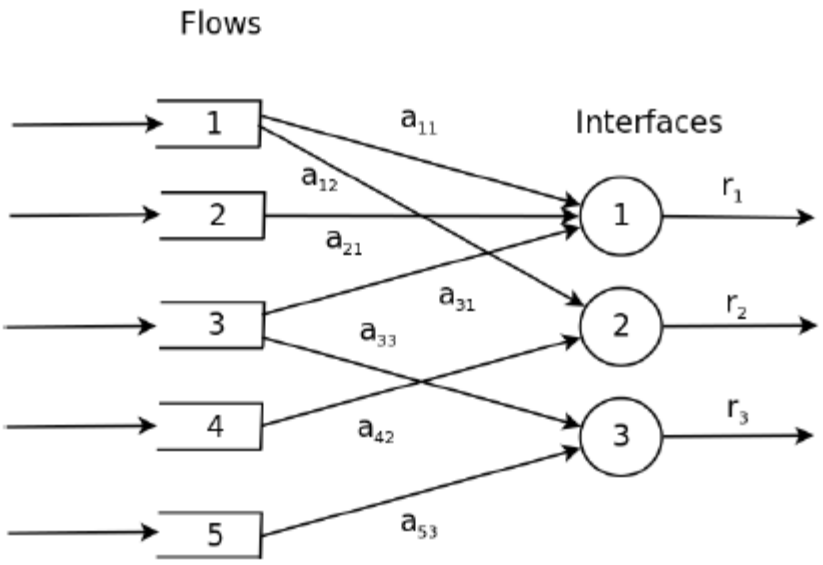
[1] Interface

سرور ممکن است از اتصالات پرهزینه تنها برای ترافیک‌های مهم بهره ببرد. بنابراین یک سوال طبیعی این است: چطور کاربرها می‌توانند مجموعه‌ای از سرورهای شبکه را قسمت‌بندی کنند که نیازهای تکی هر کاربر را برآورده سازند و محدودیت‌های سیاستی را روی مصرف منبع به کار گیرند؟

تکنیک‌های اخیر مانند TCP های چند مسیره، اولویت‌های برابری را به هر جریان و زیر جریان اختصاص می‌دهد و کاربرهای با اولویت‌های مختلف را تطبیق نمی‌دهد. در این پژوهش ما به پاسخ این سؤال می‌پردازیم که چطور چندین جریان کاربری می‌توانند به‌طور منصفانه چندین سرور شبکه را قسمت کنند که در خلال آن نرخ و تأخیر محدود را نیز برای جریان‌ها تضمین نمایند.

یک نقطه‌ی شروع، GPS است که نشان می‌دهد چندین جریان رقابت کننده می‌توانند به عدالت بیشینه‌ی حداقل پهنای باند تخصیص یافته برسند. اما همه‌ی کارهای سابق GPS برای یک سرور تکی انجام شده بود و یا تنظیمات آن برای حالتی بود که جریان‌ها بدون محدودیت از تمامی سرورها استفاده می‌کردند. توجه اصلی این پژوهش این است تا این نتایج را به سیستمی با چندین سرور، تعمیم دهد که هرجریان توسط زیرمجموعه‌ای از سرورها، سروریس داده می‌شود.

شکل (۳-۱۴) سیستم مدل ما را نشان می‌دهد که در آن هر کاربر صف ارسالی خودش را دارد که در آن بسته‌ها تا زمانی که یک سرور در دسترس باشد، منتظر می‌ماند. هر جریان مجاز به استفاده از زیر مجموعه‌ای از سرورها، می‌باشد. ماتریس π ، ماتریس اولویت‌های بین هر جریان و هر سرور می‌باشد. طوری که m جریان را به n سرور ربط می‌دهد. $\pi_{ij} = 1$ می‌باشد، اگر و تنها اگر جریان i مایل به استفاده از سرور j باشد. GPS برای حالت چندسروره به وسیله‌ی وزن‌های $\varphi_1, \varphi_2, \dots\dots, \varphi_m$ هر جریان و نرخ هر سرور $r_1, r_2, \dots\dots, r_n$ و ماتریس مسیریابی π معرفی می‌شود.



شکل ۳-۱۴- سیستم مدل چند سروره با در نظرگرفتن اولویت سرورها

می‌دانیم برای GPS با یک سرور با نرخ r ، نرخی که جریان i می‌گیرد $a_i(t) \geq \varphi_i/\sum_i \varphi_i * r$ است. با انتخاب φ_i مناسب، می‌توانیم یک نرخ کمینه را تضمین کنیم که هر جریان می‌تواند در آن سرویس بگیرد. برای مثال اگر می‌خواهیم جریان ۱ حداقل ۱۰ درصد از نرخ ارتباط را کسب کند، آنگاه به سادگی $\varphi_1 = 0/1$ می‌گیریم و البته اطمینان حاصل می‌کنیم که $\sum_i \varphi_i \leq 1$. حال که به دنبال عدالت بیشینه‌سازی حداقل پهنای باند تخصیص یافته وزن‌دار، برای حالت چند سروره هستیم، شایسته است که بدانیم آیا می‌توانیم هنوز هم یک نرخ تضمین شده‌ای را به هر جریان بدهیم؟

قضیه۳-۱ درستی این ادعا را ثابت می‌کند:

قضیه ۳-۱: تحت عدالت بیشینه سازی حداقل پهنای باند تخصیص یافته در شبکه، نرخی که جریان i به دست می‌آورد حداقل برابر است با سهم منصفانه وزن‌دارش در بین همه‌ی جریان‌هایی که مایل هستند که یک یا چند سرور را با جریان i قسمت کنند[۴۸].

$$a_i(t) \geq \frac{\varphi_i}{\sum_{j,\exists k,\pi_{ik}=1,\pi_{jk}=1} \varphi_j} \sum_{j,\pi_{ij}=1} r_j(t) \qquad (۲۹-۳)$$

اثبات: فرض کنید همه‌ی جریان‌ها مایل هستند تا یک یا چند سرور را با i قسمت کنند و دقیقاً از مجموعه‌ی مشابهی از سرورها استفاده کنند. آنگاه نامساوی بالا به تساوی تبدیل می‌شود، چون a_i ، یک نرخ عدالت بیشینه شده‌ی حداقل پهنای باند تخصیص یافته‌ی وزن‌دار است. اگر هر جریان کمتر از این نرخ استفاده کند(به هر دلیلی. مثلاً ممکن است جریان دیگر بسته‌ای برای ارسال نداشته باشد یا اینکه دیگر جریان تمایل به استفاده از یک سرور که i استفاده می‌کند، ندارد و یا جریان از سروری استفاده می‌کند که i تمایلی به استفاه از آن ندارد) آنگاه نرخ تخصیص سرویس به جریان‌های باقیمانده افزایش می‌یابد که شامل جریان i نیز می‌شود.

این به این معناست که نرخ عدالت بیشینه‌سازی حداقل پهنای باند تخصیص یافته که جریان i دریافت خواهد کرد، حداقل سهم منصفانه از سرورهایی است که مایل به استفاده از آنهاست، به این دلیل که این مقدار کمتر از تساوی سمت راست معادله (۳-۲۹) است.

$$\mathrm{a_i(t)} \geq \frac{\varphi_\mathrm{i}}{\sum_\mathrm{j} \varphi_\mathrm{j}} \sum_{\mathrm{j},\pi_{\mathrm{ij}}=1} \mathrm{r_j(t)} \qquad (۳۰-۳)$$

در اینجا برای سادگی، نرخ تضمین شده برای جریان i را g_i می‌نامیم که به ازای هر t ، $a_i(t) \geq g_i$.

برای مثال، اگر در شکل (۳-۱۴) بخواهیم جریان ۱ حداقل ۲۰ درصد از $r_1 + r_2$ را بگیرد، کافی است، $\varphi_1 = 0.2$ بگیریم و $\varphi_1 + \varphi_2 + \varphi_3 + \varphi_4 \leq 1$. چون تنها جریان ۱ تا ۴ سرور را با جریان ۱ قسمت می‌کنند. وقتی الگوریتم را اجرا می‌کنیم تا تخصیص منصفانه وزن‌دار را به دست آوریم، جریان ۱ حداقل نرخ سرویس تقاضا شده‌اش را دریافت خواهد کرد.

همان‌طور که در ابتدای فصل بیان شد،یکی از ویژگی‌های GPS، محدود کردن تأخیر یک بسته در داخل سیستم است و روش معمول برای کنترل آن سطل نشتی می‌باشد. اگر $A_i(t_1, t_2)$ تعداد بسته‌های ورودی برای جریان i در بازه‌ی $(t_1, t_2]$ باشد، آنگاه A_i را با مشخصه‌های (σ_i, ρ_i) محدود می‌کنیم.

$$A(t_1, t_2) \leq \sigma_i + \rho_i(t_2 - t_1) \quad , \quad \forall t_2 \geq t_1 \geq 0. \tag{۳۱-۳}$$

انباشتگی ورودی توسط σ_i و نرخ ارسال پیوسته توسط ρ_i محدود می‌شود.

حال اثبات می‌کنیم که یک سیستم GPS چند سروره ویژگی مشابهی دارد و تأخیر بسته در جریان i بیشتر از $\frac{\sigma_i}{\rho_i}$ نیست. (قضیه ۳-۲) . اما مراحل تصمیم‌گیری این موضوع که یک جریان جدید چه هنگام اجازه‌ی ورود به سیستم را دارد، بسیار پیچیده تر از حالت تک سروره است. ناچاراً باید بدانیم جریان i مایل به استفاده از کدام سرور است؟ و اینکه آیا نرخ سرویس خواسته‌شده‌ی ρ_i می‌تواند به دست آید؟ این بدان معنی است که سیستم مجبور است مقدار پیکی را برای φ_j به ازای همه مقادیر j برگزیند، طوری که نرخ توسط رابطه (۳-۲۹) تضمین شود.

$a_i(t) \geq \rho_i$

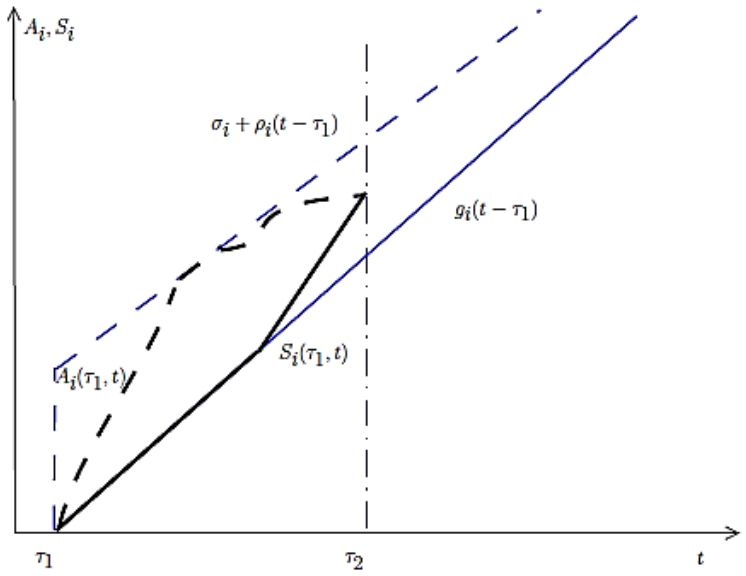
اگر این شرایط ایجاد شود آنگاه ضمانت تأخیر حاصل می‌شود و زمان مرخصی از سرور توسط (σ_i, ρ_i) محدود می‌شود.

قضیه ۳-۲ : تصور کنید می‌خواهیم جریان i که با پارامترهای (σ_i, ρ_i) محدود شده را در یک شبکه چند سروره وارد کنیم و جریان مایل به استفاده از سرور π_{ij} است. اگر $\sum_j \sigma_j \leq B$ باشد و اگر بتوانیم مقادیر φ_j را پیدا کنیم طوری که $a_i(t) > \rho_i$ ، آنگاه تأخیر هر بسته در جریان i توسط حد بالای σ_i/ρ_i محدود می‌شود.

اثبات: $S_i(t_1, t_2)$ سرویسی است که توسط جریان i ، در بازه زمانی $(t_1, t_2]$ به دست آمده است. جریان i را در نظر بگیرید که در τ_1 می‌رسد (یعنی دچار انباشتگی شده است) و در زمان τ_2 تمام می‌شود (یعنی از انباشتگی خارج می‌شود). مشاهده می‌کنیم که $A_i(t_1, t_2)$از بالا توسط رابطه(۳-۳۱) و $S_i(t_1, t_2)$ از پایین توسط رابطه زیر محدود می‌شود.

$$S_i(\tau_1, t) \geq g_i(t, \tau_1) \quad , \forall \tau_1 \leq t \leq \tau_2 \qquad (۳۲-۳)$$

که $a_i(t) \geq g_i$ در شکل (۳-۱۵) نشان داده شده است. فاصله‌ی افقی بین $S_i(\tau_1, t)$ و $A_i(\tau_1, t)$ در شکل نشان‌دهنده‌ی تأخیر و فاصله‌ی عمودی انباشتگی در لحظه t را نشان می‌دهد.



شکل ۳-۱۵- نمایش $S_i(\tau_1,t)$ و $A_i(\tau_1,t)$ و حد بالا و پایین قابل قبول آن‌ها

از آنجائیکه تأخیر طول خط افقی بین A_iو S_i است، می‌توانیم ماکزیمم آن را از طریق محاسبات ساده به دست آوریم. با به دست آوردن تابع معکوس حدها شروع می‌کنیم.

$$y = \sigma_i + \rho_i(t_a - \tau_1) \longrightarrow \ t_a = \frac{y - \sigma_i}{\rho_i} + \tau_1 \qquad (۳۳-۳)$$

$$y = g_i(t_s - \tau_1) \ \longrightarrow \ t_s = \frac{y}{g_i} + \tau_1 \qquad (۳۴-۳)$$

تأخیر بسته‌ها آنگاه از بالا باید اینطور محدود شود:

$$D = t_s - t_a = \frac{y}{g_i} - \frac{y - \sigma_i}{\rho_i}. \qquad (۳۵-۳)$$

مشاهده کنید که D یک تابع نسبی با گرادیان زیر است:

$$\frac{dD}{dy} = \frac{1}{g_i} - \frac{1}{\rho_i}, \qquad (۳۶-۳)$$

که همیشه منفی است، چون $\rho_i < g_i$ است. این به این معنی است که D به‌طور یکنواخت نزولی است. در آنالیزها ما مشتاق به دامنه‌های $t > \tau_1$ هستیم. بنابراین D در نرخ $y = \sigma_i$ ماکزیمم می‌شود و این بدان معنی است که تأخیر بسته :

$$D \leq \frac{\sigma_i}{g_i} < \frac{\sigma_i}{\rho_i} \qquad (۳۷-۳)$$

چون $\rho_i < g_i$ است. این حداکثر تأخیر برای آخرین بیتی که در زمان τ_1 برای آخرین بیتی که در آن توده می‌رسد، اتفاق می‌افتد.

فصل ۴

الگوریتم پیشنهادی

۴-۱ مقدمه

استفاده بهینه از منابع در یک شبکه‌ی داده، مستلزم طراحی یکپارچه‌ی الگوریتم‌های تخصیص امکانات می‌باشد. برای این منظور، افزون بر لزوم حداکثری از منابع شبکه، لازم است قیود کیفیت سرویس مربوط به هر محاوره لحاظ شود. اکنون که تلفن‌های هوشمند ما دارای چندین سرور از قبیل (WiFi، 3G، 4G و...) هستند، ما یک‌سری اولویت‌هایی داریم که مشخص می‌کند هر کاربر از چه سرورهایی برای سرویس می‌تواند استفاده کند. به‌طور مثال اگر رشته‌های ویدئویی را از طریق WiFi ارسال کنیم سریع‌تر خواهد بود، اما برای تماس‌های VOIP از 3G که اتصال پایدارتری دارد، بهره ببریم. اگرچه تاکنون تحقیقات گسترده‌ای در زمینه‌ی طراحی توام الگوریتم-های تخصیص امکانات انجام شده است، با این وجود هیچ‌یک از نظم‌های صف‌بندی نمی‌توانند اولویت‌های یک کاربر در استفاده از بعضی سرورها را در برابر بقیه سرورها درنظر بگیرند.

۴-۲ شرح مسئله

وسایل موبایل دارای سرورهای شبکه چندتایی مانند WiFi، 3G، 4G و ... هستند، که به آن‌ها اجازه می‌دهد به شبکه های متنوعی وصل شوند، و این برای کاربرها معمول است که دو یا چند

سرور فعال در یک زمان داشته باشند. همچنین می‌آموزیم که کاربران در نحوه‌ی استفاده از شبکه-های گوناگون باویژگی‌های متفاوت دارای اولویت هستند. برای مثال برای برقراری تماس VOIP ازطریق skype، از WiFi استفاده می‌کنیم به این دلیل که تأخیر[1] در 3G بالاتر است، و یا درهنگام حرکت، شبکه های سلولار را ترجیح می‌دهیم، تا مطمئن شویم به‌صورت پایدارتری متصل هستیم. هرچند در آینده ممکن است با اولویت‌هایی مواجه شویم که درحال حاضر مورد حمایت قرار نمی‌گیرند، برای مثال ممکن است که بخواهیم از همه‌ی سرورها در یک زمان استفاده کنیم، تا همه‌ی پهنای باند در دسترس را به یک کاربر خاص اختصاص دهیم. می‌توان تنوع عظیمی از اولویت‌ها را تصور کرد که همگی با سبک سنگین کردن پهنای باند، تأخیر، هزینه و اتصال مداوم ناشی می‌شوند، که می‌توانند سرورهای کاربر را باتوجه به نیاز کاربران تراز کنند. روش ما برپایه ی دو نوع متفاوت از اولویت‌هاست: اول، اولویت سرور، که نشان می‌دهد یک کاربر مایل به استفاده از یک سرور خاص می‌باشد یا خیر. این اولویت، یک اولویت باینری است. دوم، اولویت نرخ که نشان می‌دهد یک کاربر چه میزان از یک سرور را باید در اختیار گیرد. برای مثال حداقل نصف پهنای باند سرور WiFi را به Netflix اختصاص می‌دهد.

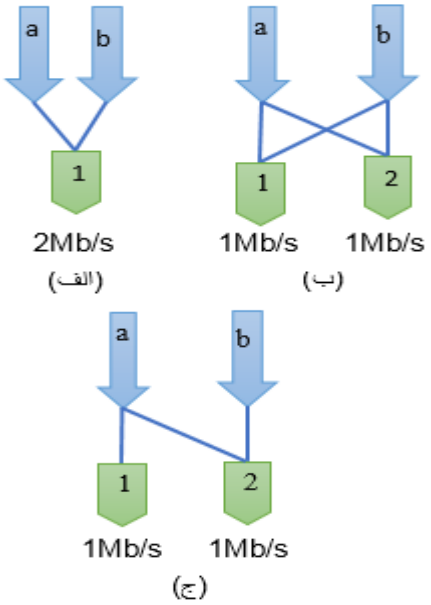
هدف ما روبرویی با هر دو نوع اولویت است، در هنگامی که استفاده از همه‌ی شبکه‌های قابل دسترس را ماکزیمم می‌کنیم. درواقع بیان مسئله برای اشتراک‌گذاری چندین سرور با توجه به اولویت کاربران می‌باشد و اینکه چطور هر سرور باید توسط هرکاربر استفاده شود. خوشبختانه هیچ‌یک از کارهای گذشته این مشکل را آدرس‌دهی نکرده‌اند. ممکن است تصور شود که کارهای بسیاری روی صف‌بندی منصفانه، بر روی سرورهای چندگانه انجام شده است، اما کارهای گذشته [۲۰]،[۴]،[۴۹] شامل مفهوم اولویت سرور نمی‌شوند، بلکه در آنجا همه‌ی کاربران اجازه دارند تا از همه‌ی سرورها در همه‌ی زمان‌ها استفاده کنند. کار قبلی برای فراهم کردن عدالت، با استفاده از صف‌بندی منصفانه وزن‌دار تنها بر روی اولویت نرخ تمرکز دارد[۵۰]. چنین کارهایی در چندین

[1] latency

حوزه می‌تواند یافت شود، از محدوده‌ی شبکه های چند اتصاله[1] گرفته تا شبکه‌های مش وایرلس [۲۰]،[۵۰]. اولویت سرور به‌طور روشنی مسئله را پیچیده‌تر می‌کند و کارهای قبلی را نامناسب می‌داند.

برای فهم بهتر موضوع، مثال آورده شده در شکل (۴-۱) را ببینید. دو کاربر سرورهای در دسترس را به اشتراک می‌گذارند و هیچ اولویت نرخی وجود ندارد (هر جریان دارای وزن برابر است). همان‌طور که در کارهای گذشته پیشنهاد شده فرض کنید WFQ را بر روی هر سرور به کار بریم. اگر تنها یک سرور وجود داشته باشد، شکل (۴-۱) (الف)، WFQ یک تخصیص مساوی ۱ Mb/s را به هر جریان می‌دهد. حال فرض کنید دو سرور ۱ Mb/s و ظرفیت کل ۲ Mb/s داریم. اگر جریان‌ها هیچ اولویت سروری نداشته باشند و مایل به استفاده از هر دو سرور باشند شکل (۴-۱) (ب)، تخصیص عدالت برای هر جریان ۱ Mb/s باقی می‌ماند، که می‌تواند با بکارگیری WFQ روی هر سرور به دست آید. اما اگر اولویت سرور را معرفی کنیم که در آن جریان a می‌تواند از هر دو سرور و جریان b تنها می‌تواند از سرور ۲ استفاده کند، شکل (۴-۱) (ج)، آنگاه اجرای WFQ روی هر سرور برای فراهم آوردن یک تخصیص منصفانه مردود می‌شود: جریان a ، ۱/۵ Mb/s و جریان b تنها۰/۵ Mb/s از ظرفیت را می‌گیرد. این اتفاق به‌این خاطر است که سرور ۱ همه‌ی ظرفیتش را به جریان a می‌دهد، چون تنها جریان مایل به استفاده از این سرور است و سرور ۲ ظرفیتش را به‌طور مساوی بین دو جریان تقسیم می‌کند. به‌بیان‌دیگر WFQ نمی‌تواند اولویت نرخ را در حضور اولویت سرور فراهم کند.

[1] Multi-homing



شکل ۴-۱- مثال‌های زمان‌بند بسته‌ای الف) حالت استاندارد(دوجریان و یک سرور). ب)دو جریان و دو سرور بدون اولویت باینری. ج) دو جریان و دو سرور با اولویت باینری

در ادامه الگوریتمی را معرفی می‌کنیم که (در این وضعیت که اشتراک منصفانه غیر وزن‌دار داریم) با در نظر گرفتن اولویت سرور با اولویت نرخ روبرو می‌شود و این موضوع در مثال ما یعنی اینکه به هر جریان Mb/s ۱ بدهیم. جریان a همه‌ی ظرفیت سرور ۱ را می‌گیرد و جریان b همه‌ی ظرفیت سرور ۲ را به خود اختصاص خواهد داد. درواقع تلاش می‌کنیم تا جایی که ممکن است به هر جریان سهم منصفانه وزن‌دارش را بدهیم، درحالی‌که تضمین می‌کنیم اولویت سرور نقض نمی‌شود. درواقع اطمینان حاصل می‌کنیم که ظرفیت شبکه هیچ‌گاه تلف نمی‌شود و همه‌ی سرورها به‌صورت کار مداوم عمل می‌کنند.

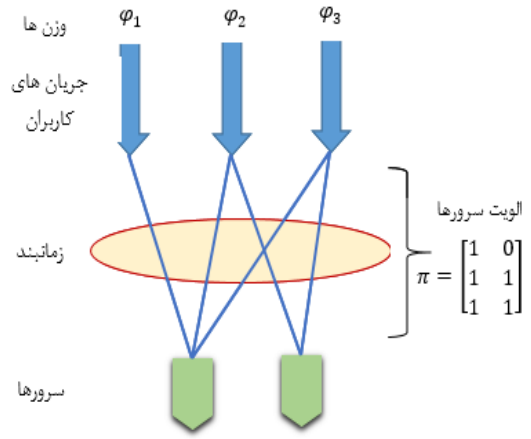
حالت‌هایی وجود دارد که اولویت سرور مقابل اولویت نرخ قرار می‌گیرد. مثال قبل را در نظر بگیرید. اگر کاربر یک اولویت نرخ را اعلام کند که در آن جریان b دو برابر جریان a را بگیرد، با یک مشکل مواجه می‌شویم. در حالتی که هیچ اولویت سروری نبود، جریان a ، Mb/s ۶۷/. و جریان b ، تنها Mb/s ۱/۳۳را دریافت می‌کردند. اما چون جریان b تنها می‌تواند از سرور ۲ استفاده کند، ما می‌توانیم حداکثر Mb/s ۱ را به آن بدهیم. آیا باید به جریان a تنها Mb/s ۰/۵

را بدهیم تا اولویت نرخ را رعایت کرده باشیم؟ پاسخ تصمیم‌گیری طرح ما خیر است. هیچ‌گاه نمی-خواهیم که ظرفیت را تلف کنیم. پس همه‌ی ظرفیت باقیمانده را به a می‌دهیم. یک نقطه طبیعی برای شروع جستجوی یک زمان‌بند بسته‌ای، الگوریتم‌های مشابه WFQ می‌باشد که زمان پایان یک بسته را در لحظه‌ای که یک بسته می‌رسد، محاسبه می‌کند و سپس زمان خاتمه سرویسش را بر اساس زودترین زمان پایان برنامه‌ریزی می‌کند. در کمال تعجب، همان‌طور که در قضیه ۴-۱ بررسی خواهیم کرد، متوجه خواهیم شد که این کلاس از الگوریتم‌ها، نیاز ما را برآورده نمی‌کنند؛ درواقع اثبات کردیم که تنها الگوریتم‌هایی می‌توانند این کار را انجام دهند که اطلاعاتی راجع به ورود بسته‌ها در زمان آینده داشته باشند.

در این پایان‌نامه ما یک روش زمان‌بندی عملی و کارآمد را معرفی می‌کنیم که بسته‌ها را معرفی کند و هرکجا که لازم است، در حضور اولویت نرخ به اولویت سرور هم توجه می‌کند. این الگوریتم برای پیاده‌سازی، عملی است. درعین‌حال می‌توانیم به‌صورت فرمولی صحت آن را اثبات کنیم. در وهله‌ی نخست طراحی یک الگوریتم کمی مشکل به نظر می‌رسد، اما تنها یک پرچم تک‌بیتی به ما این امکان را می‌دهد که زمان‌بند تک‌سروره را به یک زمان‌بند چندسروره‌ی عملی تبدیل کنیم.

مسئله‌ی زمان‌بندی ما چکیده‌ای از مدل شکل (۴-۲) می‌باشد. با ۳ جریان که توسط ۲ رابط خارجی سرویس داده می‌شوند. در این مدل هر جریان i دارای وزن φ_i برای نشان دادن تقدم مربوطه، یا درواقع همان اولویت نرخ خودش هست. برای مثال اگر $\varphi_1 = 2\varphi_2$ باشد، یعنی کاربر ۱، دو برابر بیشتر از کاربر ۲ پهنای باند دریافت می‌کند (هنگامی‌که هر دو ترافیک انباشته دارند). همچنین هر جریان برای معرفی زیرمجموعه‌ای از سرورهایی که مایل به استفاده از آن می‌باشند، دارای اولویت سرور نیز می‌باشد. اگر جریان a مایل به استفاده از سرور ۱ باشد، آن را این‌طور نماش می‌دهیم که $\pi_{a1} = 1$. اولویت سرور توسط ماتریس اتصال $\pi = [\pi_{ij}]$ ضبط می‌شود. همان‌گونه که در شکل (۴-۲) نشان داده‌شده است، این ماتریس یک گراف دوقسمتی است که اولین گره آن از جریان a و سرور ۱ شروع می‌شود. $\pi_{a1} = 1$ اگر و تنها اگر جریان a مایل به

استفاده از سرور ۱ باشد. ممکن است همیشه همه‌ی درایه‌های ماتریس ۱ نباشد. یعنی همه‌ی جریان‌ها مایل به استفاده از همه‌ی سرورها نیستند.



شکل ۴-۲: سیستم مدل فرضی برای زمان‌بند بسته‌ای در حالت چند سروره

۴-۳ عدم کار آیی الگوریتم‌های منصفانه تک سروره

برای حالت تک‌سروره، صف‌بندی منصفانه وزن‌دار، توسط الگوریتم‌هایی مانند PGPS، همه‌ی ویژگی‌های یک زمان‌بند بسته‌ای را برآورده می‌کند و نرخ منصفانه r_i/φ_i را برای هر جریان فراهم می‌کند. PGPS به‌عنوان یک معیار عدالت بیشینه‌سازی حداقل پهنای باند تخصیص‌یافته، برای حالت تک‌سروره شناخته‌شده است.

تعریف : بیشینه‌سازی حداقل نرخ تخصیص‌یافته، نرخی است که در آن هیچ جریانی نمی‌تواند یک نرخ بیشتری را بگیرد، بدون اینکه نرخ جریان دیگری را که تخصیص کمتر یا مساوی دارد، کاهش دهد.

الگوریتم PGPS به‌وسیله‌ی طراحی یک زمان پایان، برای هر بسته در لحظه‌ی ورود و با استفاده از یک استراتژی ساده، همه‌ی ویژگی‌های مدنظر مارا برای حالت تک‌سروره برآورده می‌کند، که در آن بسته‌هایی با زمان پایان کمتر دارای اولویت سرویس‌گیری هستند. در حضور یک سرور،

PGPS به‌صورت کار مداوم[1] عمل می‌کند. اگر کاربر ادعا کند که به‌طور مثال جریان a باید دو برابر جریان b دریافت کند، آنگاه ما به‌سادگی وزن a را دو برابر وزن b انتخاب می‌کنیم و PGPS تخصیص صحیح را فراهم می‌کند. حال اثبات می‌کنیم که در حالت ذکرشده در اینجا، گرچه درک مسئله ساده است، اما این استراتژی با یک الگوریتم علی ممکن نیست. برخلاف حالت تک سروره، الگوریتم باید از ورود بسته‌های آینده هم اطلاع داشته باشد.

برای درک بهتر مسئله، با یک مثال ساده آن‌ها شرح می‌دهیم. شکل (۴-۱) (ج) را درنظربگیرید، که در آن جریان a و b ، سرور ۲ را قسمت می‌کنند. فرض کنید در لحظه $t=0$ ، P_a و P_b ، به ترتیب با طول‌های $L/2$ و L می‌رسند و همه‌ی جریان‌ها تقدم برابر دارند. هر دو سرور در نرخ ۱ Mb/s عمل می‌کنند. در لحظه‌ی $t=0$ ، که سرور ۲ در دسترس می‌باشد، باید تصمیم‌گیری کرد که آیا P_a تحت PGPS زودتر کارش به اتمام می‌رسد، یا P_b .این موضوع ما را به سمت قضیه ۴-۱ هدایت می‌کند.

قضیه ۴-۱ : در حضور اولویت سرور، یک زمان‌بند بسته‌ای نمی‌تواند همیشه ترتیب مربوطه‌ی زمان پایان بسته‌ها را تعیین کند.

اثبات: دو سناریو آورده شده در زیر را در نظر بگیرید.

۱) اگر هیچ جریان جدیدی بعد از زمان $t=0$ نرسد، هر جریان نرخ ۱ می‌گیرد. بنابراین زمان پایان جریان a وb به ترتیب $f_a=L$ و $f_b=L/2$ ، خواهد بود. بنابراین P_b زودتر سرویسش تمام می‌شود.

۲) فرض کنید سه جریان جدید در مدت کوتاهی بعد از $t=0$ می‌رسند و آن‌ها تنها تمایل دارند تا از سرور ۲ استفاده کنند. جریان a بجای رقابت کردن بر سر سرور ۲، به استفاده

[1] Work-conserving

از سرور ۱ ادامه می‌دهد و نرخش در Mb/s ۱ باقی می‌ماند، درحالی‌که نرخ جریان b به Mb/s ۱/۴ کاهش می‌یابد. در این حالت P_a زودتر سرویسش تمام می‌شود.

ازآنجایی‌که نظم پایان بسته‌ها در این دو سناریو متفاوت است و زمان‌بند بسته‌ای نمی‌تواند به‌صورت علی تشخیص دهد که کدام سناریو اتفاق خواهد افتاد، نمی‌تواند تعیین کند که ترتیب پایان بسته‌ها چگونه خواهد بود.■

۴-۴ روش پیشنهادی

در اینجا یک الگوریتم کار مداوم جدید و ساده را توصیف می‌کنیم تا بسته‌ها را برای سرورهای چندگانه با اولویت سرور زمان‌بندی کنیم. همان‌طور که در شکل (۴-۲) مشاهده می‌شود، زمان‌بند، ورودی را از کاربر یا به‌صورت خیلی تخصصی از اولویت‌های کاربر مدیریت کننده‌ی سیستم، دریافت می‌کند. اولویت سرور، سروری را که هر کاربر می‌تواند استفاده کند را مشخص می‌کند و اولویت نرخ، نرخ‌های جریان مربوطه را به ما می‌دهد. ماتریس π و بردار φ که به ترتیب نشان‌دهنده‌ی این دو اولویت هستند، ورودی‌های الگوریتم زمان‌بندی می‌باشند. با علم براینکه نمی‌توان از یک زمان‌بند استفاده کرد که زمان پایان را محاسبه کند، آیا می‌توان به مکانیزم‌های عکس‌العملی مانند DRR رجوع کرد؟ ایده‌ی پشت DRR که در ادامه به بیان آن می‌پردازیم، این است که برای فراهم کردن اولویت نرخ، تعداد بیت‌هایی که در مدت هر نوبت سرویس داده می‌شوند و تحت عنوان کوانتوم شناخته می‌شوند را باحالت برابر با تعداد RR بگیریم. مسئله‌ی اساسی در این حالت این است که اگر یک جریان تمایل به استفاده از بیش از یک سرور را داشته باشد، وقتی یک سرور بسته‌ای را زمان‌بندی می‌کند، هیچ راهی وجود ندارد که بفهمد جریان‌ها چه نرخ‌هایی را از بقیه سرورها می‌گیرند. داشتن این اطلاعات برای اطمینان بخشیدن به مسئله‌ی عدالت بیشینه‌سازی حداقل پهنای باند، مهم است. یک راه حل واضح این است که سرورها اطلاعات راجع به نرخ‌هایی که جریان‌ها از هر سرور می‌گیرند را مبادله کنند. به‌این‌ترتیب که زمان‌بند بسته‌ای نرخ-

های فراهم‌شده برای هر جریان را موردتوجه قرار دهد. همان‌طور که می‌توان حدس زد، این طرح به یک مقدار غیرعملی از اطلاعات حالت نیاز دارد تا بتواند به‌خوبی سرورها قابل مبادله باشد و نرخ لحظه‌ای آن‌ها را بداند، که البته همه این‌ها نیازهای احتمالی و غامضی هستند، مخصوصاً در شبکه‌های متحرک. پس چطور می‌توان به یک نرخ عدالت بیشینه‌سازی حداقل پهنای باند تخصیص‌یافته دست یافت، تا بتواند به‌طور مستقل بر روی هر سرور به دست بیاید، بدون اینکه حتی مجبور باشیم نرخ‌های واقعی گرفته شده بین همه‌ی سرورها را محاسبه و مبادله کنیم.

سهم کلیدی این پایان‌نامه عمومیت بخشیدن به الگوریتم زمان‌بندی DRR برای سرورهای چندگانه با اولویت نرخ، هنگامی‌که تقریباً هیچ نیازی به هماهنگی بین سرورها نیست، می‌باشد. خصوصاً اینکه به هیچ محاسبه‌ی نرخی نیاز ندارد و حداکثر یک بیت از سیگنال هماهنگی از هر سرور، برای هر جریان نیاز است. این یک بیت برای پرچم سرویس بولی است و روی هر سرور یک پرچم برای هر جریان وجود دارد. پرچم بیان می‌دارد که آیا اخیراً جریانی توسط سروری دیگر سرویس داده شده است یا خیر. وقتی یک سرور به سرویس دادن به یک جریان رسیدگی می‌کند، اگر پرچم سرویسش، ست(یک) شده باشد از آن می‌گذرد. این ساختار ساده به تخصیص عدالت بیشینه‌سازی حداقل پهنای باند، منجر می‌شود. علاوه بر این هر سرور تنها نیاز دارد تا نرخ‌های مربوطه به‌دست‌آمده بین جریان‌هایی را که اجازه دارد بر اساس اولویت سرور، سرویس‌دهی کند را بداند. زمان‌بند بسته‌ای تنها باید چک کند که آیا نرخ یک جریان بخصوص از حداقل یک جریان دیگر که در حال سرویس‌گیری روی سرور مشابه است، بیشتر می‌باشد. اگر چنین است، تصمیم زمان‌بند ساده است: نباید به جریانی با نرخ مربوطه‌ی بالاتر سرویس دهد.

اگر الگوریتم زمان‌بندی این شرایط را به‌صورت تکراری، برای همه‌ی جریان‌ها اعمال کند، نهایتاً به جریانی سرویس می‌دهد که آن را به سمت عدالت بیشینه‌سازی حداقل پهنای باند تخصیص‌یافته، هدایت می‌کند.

در ادامه به بررسی دقیق‌تر چگونگی کارکرد این الگوریتم می‌پردازیم.

سرور j برای هر جریان i ، یک پرچم سرویس را دربر دارد: [1] SF_{ij}. پرچم برای بقیه‌ی سرورهاست تا به سرور j نشان دهد که به جریان i اخیراً سرویس داده شده است. برقرار ساختن این پرچم سرویس بولی نیازمند انجام دو کار است:

۱. وقتی تداخل k جریان i را سرویس می‌دهد، پرچم‌های سرویس بقیه یعنی $SFij\ ,\ \forall j \neq k$ ، ست می‌شود. به این معنی که به بقیه سرورها اطلاع دهد که به جریان i سرویس داده شده است.

۲. وقتی تداخل j جریان i را برای سرویس انتخاب می‌کند، پرچم سرویس SF_{ij} را ریست می‌کند.

درواقع در این روش، هر سرور DRR تکی خودش را انجام می‌دهد، با این تغییر جزئی که قبل از سرویس دادن جریان، پرچم سرویس را چک می‌کند. برای فهم بهتر الگوریتم، به مثال قبلی که در شکل (۴-۱) (ج) آمده برمی‌گردیم. فرض کنید که سرور ۲ بیکار است و الان در حال آمادگی است که جریان b را سرویس دهد. سرور ۲ پرچم سرویسش را با جریان b ست نمی‌کند، چون هیچ سرور دیگری جریان b را سرویس نمی‌دهد. بنابراین اقدام به سرویس‌دهی به جریان b می‌کند. همان‌طور که در DRR بیان می‌شود، شمارشگر جریان b، DC_b، به‌اندازه Q_b افزایش خواهد یافت و به‌صورت مؤثر یک یا بیش از یک بسته از جریان b فرستاده خواهد شد. وقتی بسته‌ای ارسال می‌شود، طولش از DC کم می‌شود. جریان b تاز مانی که DC آن برای بسته‌ی بعدی کافی باشد، سرویس می‌گیرد. در غیر این صورت، سرور ۱ اقدام به سرویس‌دهی به جریان a می‌کند. به دلیل اینکه سرور۱ در یک نرخ یکسان به جریان b سرویس می‌دهد، سرور ۲ پرچم سرویسش را با جریان a به‌وسیله‌ی سرور۱ ست می‌کند. در عوض بعدازاینکه پرچم سرویسش را

[1] Service flag

برای جریان a ریست کرد، به جریان b برمی‌گردد. با ادامه‌ی الگوریتم سرور ۱ تنها به جریان a و سرور۲ تنها به جریان b سرویس می‌دهد. درواقع موضوع این است که هر سرور درحالی‌که توسط پرچم‌های سرویس چک و مخابره می‌شود، زمان‌بند بسته‌ای خودش را انجام می‌دهد.

۴-۵ بررسی الگوریتم‌ها

الگوریتمDRR و یا RR نسبت به الگوریتم‌های پیشین همچون SPN، در برخورد با جریان‌هایی با طول بسته‌های کاملاً متفاوت بسیار مقاوم‌تر عمل می‌کند. به‌طور مثال اگر در جریانی در حال انتقال ویدئو باشیم و در جریان دیگر انتقال داده‌ی صدا را داشته باشیم، در الگوریتم SPN، تا زمانی که انتقال بسته‌های کوتاه صدا، اتمام نیافته است، جریان ویدئو باید انتظار بکشد. باوجود رسیدن نوبت به جریان ویدئو بعد از زمان انتظار، در صورت آمدن داده در جریان صدا، مجدداً ادامه‌ی پردازش جریان ویدئو به‌صف انتظار می‌رود. این تسهیم بسیار ناعادلانه است و در دنیای رسانه‌ای جدید باوجود انواع طول داده‌ها، این نوع الگوریتم‌ها فاقد کارایی مناسب می‌باشند. الگوریتم‌های RR برای مقابله با مشکلات گرسنگی شدید جریان‌های طولانی ابداع شدند. این الگوریتم‌ها با دادن تسهیم زمانی برابر بر روی تمامی جریان‌ها، تعادلی را در پردازش داده‌های شبکه ایجاد کرده‌اند. ولی باز هم در شبکه‌هایی که جریان‌های با طول بسته‌ی به‌شدت متفاوت دارند، استفاده از RR ساده، تقسیم باند کاملاً منصفانه‌ای را ارائه نمی‌دهد.

در مثال مطرح‌شده در مورد جریان ویدئو و جریان صدا، با اعمال تقسیم زمانی برابر از پردازش سرور برای هر جریان، میزان تأخیر در جریان ویدئو بیشتر خواهد بود. درنتیجه دریافت ویدئو و صدا برای کاربر هم‌زمان صورت نخواهد گرفت و داده‌ی ویدئو تأخیر خواهد داشت. برای مقابله با این مشکل باید الگوریتمی پیاده‌سازی شود که طول بسته‌های داده را در نظر بگیرد و با توجه به آن پهنای باند سرور را بین جریان‌هایی که به آن دسترسی دارد، تقسیم نماید. از سوی دیگر سرعت سرورها نیز برای این تقسیم‌بندی مؤثر می‌باشد. به‌طور مثال اگر جریانی توانایی سرویس

داده شدن توسط دو سرور با سرعت‌های مختلف را دارد، قاعدتاً باید سهم بیشتری از داده‌های خود را، از طریق سرور قوی‌تر ارسال نماید.

حال در شرایط برابری کامل برای دو جریان ازنظر طول داده و دسترسی به سرورهای مشابه اگر جریانی سرعت تولید داده‌ی بالاتری داشته باشد، در صورت عدم کفایت سرعت پاسخ‌دهی سرورها و انباره شدن داده در بافر، نیاز است تا جریان سریع‌تر سهم بیشتری از پهنای باند سرور مشترک با جریان آهسته‌تر را دریافت نماید.

درنتیجه در تعیین نسبت موردنیاز پهنای باند سه عامل سرعت جریان، طول بسته‌های جریان و سرعت سرور دخیل می‌باشند. عامل اول و دوم چون مربوط به خود جریان است رابطه‌ی مستقیم با متغیرهای خود جریان و جریان‌های هم‌سرور دارد، اما پارامتر سوم یک پارامتر مشترک و در ارتباط با سایر جریان‌های غیر هم سرور می‌باشد. برای محاسبه ضرایب پهنای باند روابط زیر را بر اساس مطالب بیان‌شده، پیاده‌سازی می‌نماییم.

$$SP_{FN}(i) = \frac{SP_F(i)}{\sum_{n=0}^{NF} SP_F(n)} \quad (۱-۴)$$

که در آن i جریان ، SP_F سرعت جریان و SP_{FN} سرعت جریان نرمالیزه شده می‌باشد.

$$SP_{SN}(j) = \frac{SP_S(j)}{\sum_{n=0}^{NS} SP_S(n)} \quad (۲-۴)$$

که در آن j سرور، SP_S سرعت سرور و SP_{SN} سرعت سرور نرمالیزه شده می‌باشد.

$$L_{FN}(i) = \frac{L_F(i)}{\sum_{n=0}^{NF} L_F(n)} \quad (۳-۴)$$

که در آن L_F بیشترین طول داده هر جریان و L_{FN} بیشترین طول داده هر جریان نرمالیزه شده می‌باشد. آنگاه ماتریس نسبت ضرایب وابستگی A به‌صورت زیر به دست می‌آید.

$$A_{N_F * N_S =}$$

$$\begin{bmatrix} \frac{\pi(1,1) * L_{FN}(1) * SP_{FN}(1)}{\sum_{n=0}^{NF} L_F(n) * SP_{FN}(n) * \pi(n,1)} & \frac{\pi(1,2) * L_{FN}(1) * SP_{FN}(1)}{\sum_{n=0}^{NF} L_F(n) * SP_{FN}(n) * \pi(n,2)} & \dots \\ \frac{\pi(2,1) * L_{FN}(2) * SP_{FN}(21)}{\sum_{n=0}^{NF} L_F(n) * SP_{FN}(n) * \pi(n,1)} & \ddots & \vdots \\ \vdots & \dots & \frac{\pi(N_F, N_S) * L_{FN}(N_F) *}{\sum_{n=0}^{NF} L_F(n) * SP_{FN}(n} \end{bmatrix}$$

(۴-۴)

در رابطه‌ی (۴-۴)، $\pi(i,j)$ ماتریس رابطه‌ها بین جریان i و سرور j می‌باشد. آنگاه از حاصل‌ضرب ماتریس A در ماتریس سرعت سرورها (B)، مقدار ضرایب φ برای هر جریان محاسبه می‌شود.

$$B_{N_S \times 1} = \begin{bmatrix} SP_{SN}(1) \\ SP_{SN}(2) \\ \vdots \\ SP_{SN}(N_S) \end{bmatrix} \quad (۵-۴)$$

$$\varphi_{N_F \times 1} = A_{N_F * N_S} \times B_{N_S \times 1} \quad (۶-۴)$$

ضرایب φ که نشان‌دهنده‌ی میزان پهنای باند تقسیم‌شده می‌باشد را می‌توان با گسترش جدیدترین الگوریتم صف‌بندی با DRR چندسروره پیاده‌سازی کرد.

در الگوریتم پیشنهادی برای اعمال پهنای باند، به‌جای توقف تک‌مرحله‌ای روی هر جریان، متناسب با ضرایب φ ، تعداد مراتب بیشتری که باید روی یک جریان بمانیم را تعیین می‌کنیم. درنتیجه φ باید به‌صورت عددی صحیح که نشان‌دهنده تعداد است، تبدیل گردد و همچنین کمترین مقدار آن ۱ باشد. جهت رسیدن به این منظور φ را بر کوچک‌ترین مقدار آن تقسیم می‌کنیم و به نزدیک-ترین عدد صحیح گرد می‌نماییم.

$$(\varphi_i)_{Normalized} = \text{round}(\varphi_i / \min(\varphi)) \quad (۷-۴)$$

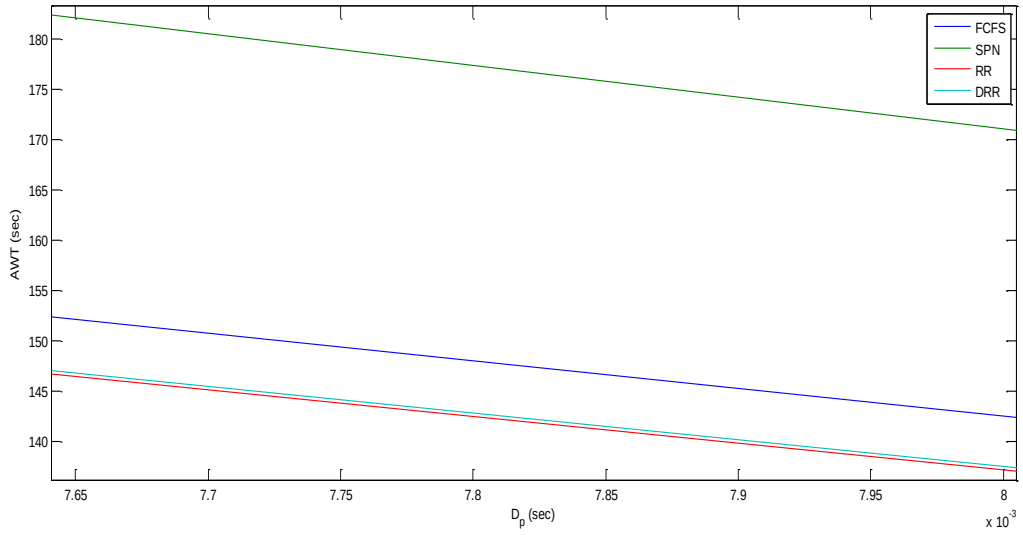
درنهایت مقدار q برای هر صف به‌صورت زیر به دست می‌آید.

$$q(i) = q \times \varphi(i) \qquad (۸-۴)$$

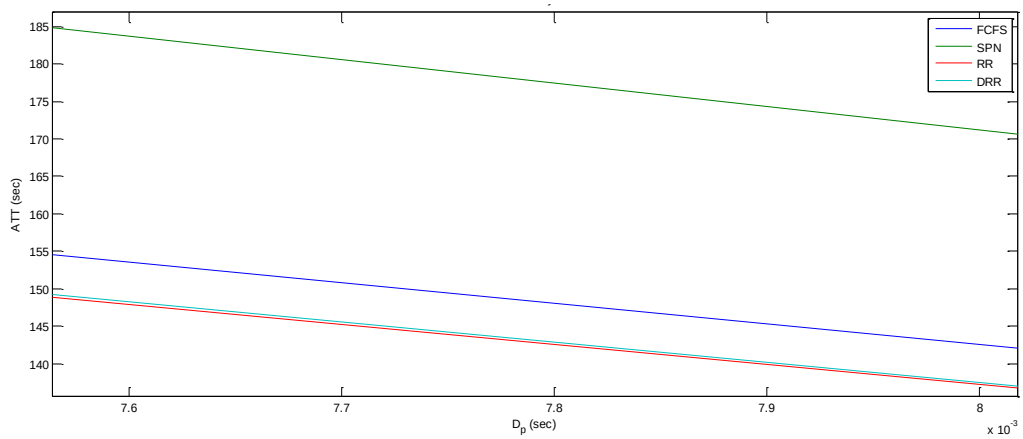
این الگوریتم با نرم‌افزار MATLAB تحت عنوان DRRNQ پیاده‌سازی شده است و با روش-های پیشین در شرایط مختلف مقایسه گردیده است.

باید در نظر داشت که در این الگوریتم، مقدار q، متناسب با بیشترین طول داده در کل جریان‌ها به دست می‌آید. در نگاه اول شاید تصور شود که بتوان ضریب q را برابر با بیشترین طول داده در هر جریان به دست آورد و به تعداد q ثانیه روی آن جریان سرویس‌دهی کرد، اما همان‌طور که در نتایج شبیه‌سازی نشان داده‌شده است این استدلال با استدلال آورده شده برای DRRNQ کاملاً متفاوت بوده و نتایج کاملاً متفاوتی نیز می‌دهد، زیرا در این حالت مقدار q فقط به طول داده بستگی دارد و مقدار سرعت سرور و سرعت جریان در آن بی‌تأثیر است. ما این الگوریتم را DRRN نام‌گذاری کرده و نتایج آن را با دیگر الگوریتم‌های از پیش ذکرشده مقایسه کرده‌ایم. همان‌طور که انتظار می‌رفت DRRN نه‌تنها مانند DRRNQ عمل نکرده است، بلکه مدت زمان تأخیر و انتظار آن از RRو DRR هم بدتر شده است.

در شکل‌های (۴-۳) و (۴-۴) نتایج میانگین زمان تأخیر و انتظار برای ۴ الگوریتم FCFS، SPN، RR و DRR در حالت چندسروره بر مبنای شرایطی که قبلاً بیان شد، رسم شده است. سپس الگوریتم‌هایDRRNو DRRNQ را با اعمال ضرایب φ در شکل (۴-۵) برای مقایسه با عملکرد الگوریتم‌های پیشین رسم کرده‌ایم. پارامتر D_p، مقدار تأخیر در تولید داده‌ی پوآسن است که به‌منظور وجود امکان ارسال نشانه از سطل نشانه با بیشینه سرعت ممکن در زمان‌هایی که نشانه در سطل وجود دارد، در نظر گرفته‌شده است.



شکل ۴-۳- میانگین زمان انتظار برای ۴ الگوریتم در مدت T=0-100(s)



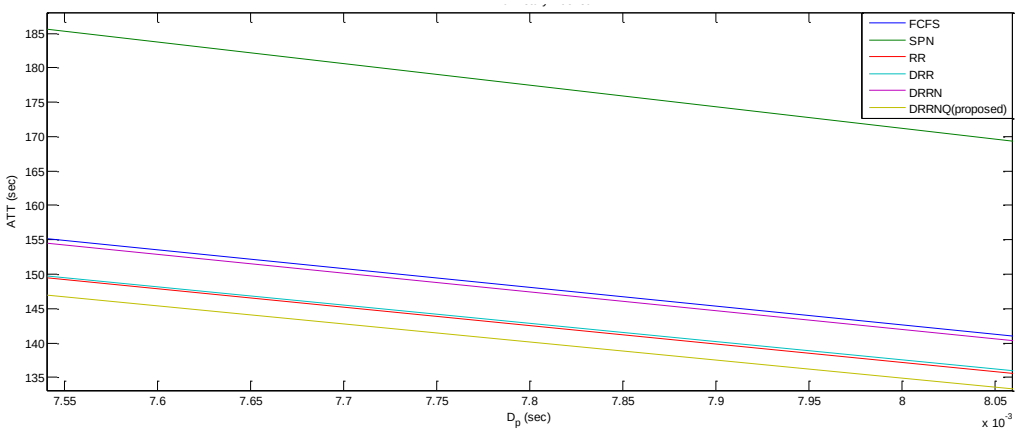
شکل ۴-۴ میانگین زمان تأخیر برای ۴ الگوریتم در مدت T=0-100(s)

برای مقایسه عملکرد الگوریتم‌های فوق با و بدون در نظر گرفتن سطل نشانه در حالت چند سروره، جدول(۴-۱) تهیه‌شده است.

جدول ۴-۱- مقایسه تأثیر سطل نشانه در حالت چندسروره

d-p=0.05	AWT		ATT	
	With leaky bucket	Without leaky bucket	With leaky bucket	Without leaky bucket
FCFS	224.9	227.32	226.1	228.86
SPN	264.8	268.81	267.3	269.43
RR	217.28	220.21	219.8	222.9
DRR	217.66	220.89	219.98	223.01
DRRN	224.24	226.7	226.4	228.5
DRRNQ	213.78	217.33	215.83	217.9

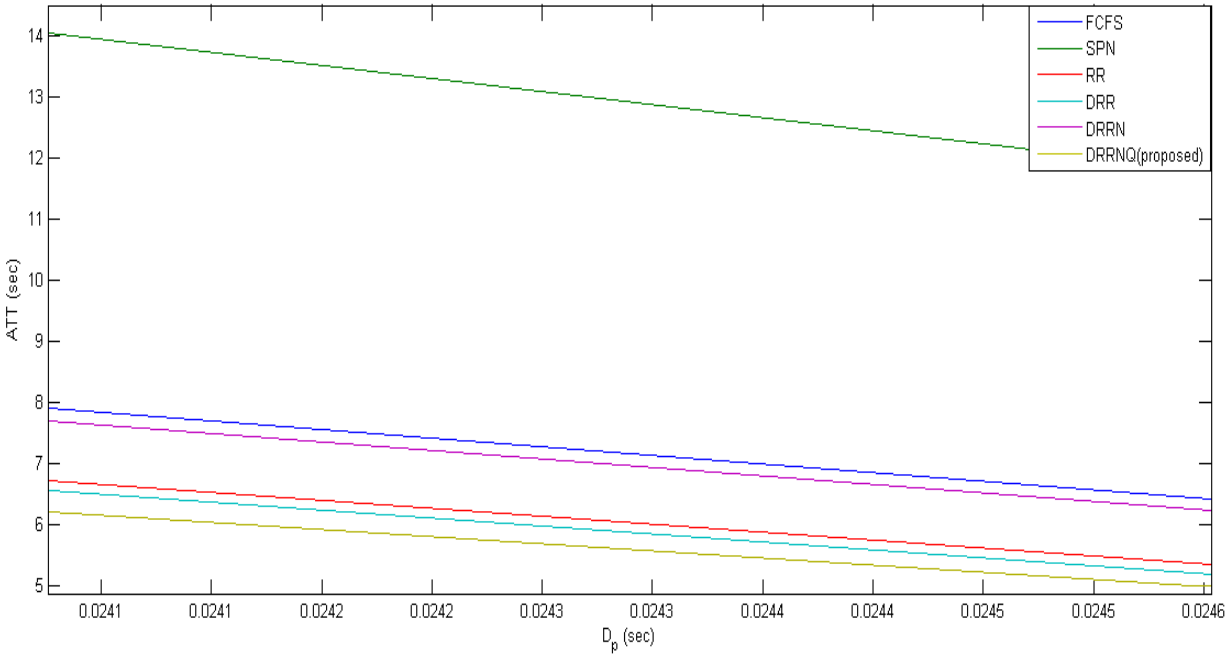
در ادامه به دلیل مشابه بودن ترتیب قرارگرفتن الگوریتم‌ها در نمودارهای میانگین زمان تأخیر و میانگین زمان انتظار، تنها به مقایسه نمودار ATT در حالت‌های مختلف بسنده می‌کنیم. شکل (۴-۵) با در نظرگرفتن دو الگوریتم جدید رسم شده است.



شکل ۴- ۵- مقایسه میانگین زمان تأخیر الگوریتم‌های DRRN و DRRNQ با ۴ الگوریتم موردنظر

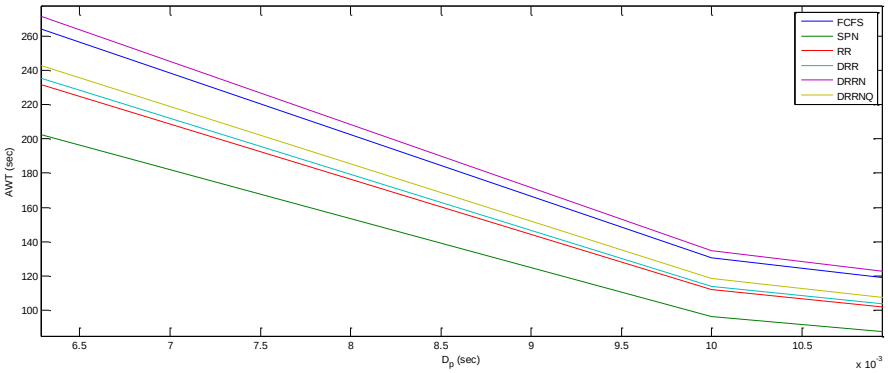
نتیجه‌ی مهم دیگری که از شبیه‌سازی‌های فوق به دست می‌آید، این است که الگوریتم DRRو RR در D_pهای مختلف رفتار متفاوتی دارند. در D_pهای بسیار نزدیک به صفر و کمتر از آستانه ۰/۰۲۴ که درواقع نشان‌دهنده حجم زیاد ورودی است، الگوریتم RR دارای پارامترهای کیفیت سرویس بهتری نسبت به DRR می‌باشد و در D_pهای بیشتر از این مقدار نتایج برای این دو الگوریتم دقیقاً برعکس یکدیگر می‌باشد. اما الگوریتم DRRNQ در هردوی این حالت‌ها دارای ثبات رفتاری بوده و مستقل از حجم فشار ترافیک ورودی، مقدار آن از هر ۴ الگوریتم قبلی بهتر

است. این نشان می‌دهد که با انتخاب q مناسب‌تر می‌توان بهبود بهتری در شبکه، در حالت‌های حجم ترافیک زیاد، برای سرویس‌دهی به دست آورد و استقلال بیشتری را برای الگوریتم در حالت‌های شلوغ بودن ترافیک تأمین کرد که این مهم در الگوریتم DRRNQ به‌دست‌آمده است. نتایج در شکل‌های (۴-۵) و (۴-۶) آورده شده است.

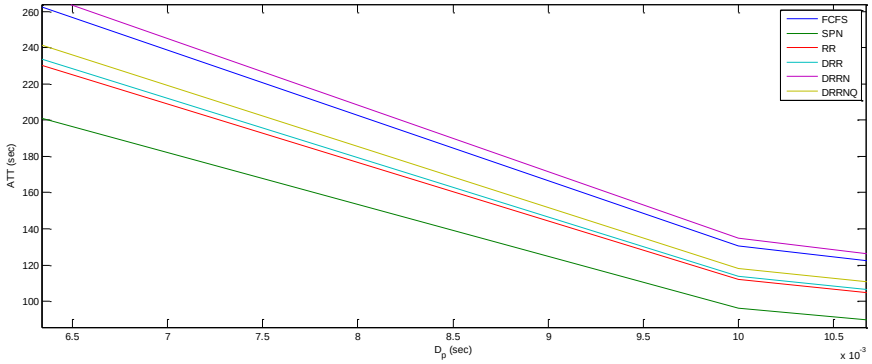


شکل ۴- ۶- مقایسه میانگین زمان تأخیر الگوریتم‌ها در D_pهای بیشتر از ۰/۰۲۴

نکته‌ای که در آخر باید ذکر کرد، این است که این الگوریتم‌ها با این شرایط در حالت تک سروره صرفاً مانند حالت چندسروره عمل نمی‌کنند و هیچ لزومی به مشاهده رفتار مشابه در دو حالت تک سروره و چند سروره وجود ندارد. بنابراین سیستم مدل انتخابی و نتایج به‌دست‌آمده تنها برای حالت چندسروره بوده و قابل‌تعمیم به حالت تک سروره نیستند. شکل(۴-۷) و (۴-۸) بیانگر این مطلب است.



شکل ۴-۷- نتایج میانگین زمان انتظار در حالت تک سروره در T=0-100



شکل ۴-۸- نتایج میانگین زمان تأخیر در حالت تک سروره در T=0-100

برای مقایسه در حالت تک سروره بدون سطل نشانه مقادیر را در جدول زیر جهت مقایسه‌ی بهتر نمایش می‌دهیم:

جدول ۴-۲- مقایسه اثر سطل نشتی در حالت تک سروره

d-p=0.01	AWT		ATT	
	With leaky bucket	Without leaky bucket	With leaky bucket	Without leaky bucket
FCFS	129.9	133.4	132.5	135.34
SPN	95.66	98.45	97.43	101.5
RR	111.43	114	113.52	116.01
DRR	113.42	116.1	116.2	119.32
DRRN	134.1	137.5	135.98	138.8
DRRNQ	117.83	120.4	119.01	122

فصل ۵

نتیجه گیری و پیشنهادها

۵-۱ نتیجه‌گیری

در این پایان‌نامه روش‌های کنترل جریان در حالت تک‌سروره و چندسروره موردبررسی قرار گرفت. ابتدا مراحل شکل‌گیری الگوریتم‌های زمان‌بندی در حالت تک‌سروره از سال‌های ابتدائی بوجودآمدن آن بیان شد و ضمن معرفی سیر تکاملی روش‌های زمان‌بندی تک‌سروره، حالت ایده آل و سیال آن مورد ارزیابی دقیق قرار گرفت. سپس دلیل ضعف الگوریتم‌های بر پایه‌ی زمان مجازی در پاسخگویی به نیازهای حالت چندسروره مطرح گردید و برای بررسی در این حالت، ۴ الگوریتم FCFS، SPN، RR، DRR، انتخاب گردید.

الگوریتم‌ها دارای ورودی پوآسن بوده و ازدحام داده‌های ورودی توسط الگوریتم سطل نشانه کنترل گردیده است و سپس مقدار پارامترهای کیفیت سرویس از قبیل میانگین زمان تأخیر و میانگین زمان انتظار برای آن‌ها موردبررسی قرار گرفت. درنهایت با ارائه الگوریتم پیشنهادی بر مبنای الگوریتم DRR چندسروره، مشاهده شد که می‌توان با انتخاب ضریب کوانتوم بهینه‌تر، مقدار پارامترهای کیفیت سرویس را تا حد چشمگیری بهبود بخشید.

درنهایت نتیجه کلی که می‌توان از این پایان‌نامه گرفت این است که داشتن پارامترهای کیفیت سرویس مطلوب در یک الگوریتم زمان‌بندی دستخوش عوامل زیادی است. در اینجا دو عامل سطل نشانه و انتخاب ضریب کوانتوم تا حد امکان بهینه، در الگوریتم‌های بر مبنای RR ، می‌توان نتایج مطلوب‌تری از کارایی سرویس‌دهی شبکه حاصل نمود.

۵-۲ پیشنهادها

۱- استفاده از ورودی خودسان به‌جای ورودی پوآسن در الگوریتم پیشنهادی برای بررسی دقیق‌تر نتایج شبیه‌سازی.

۲- به دست آوردن خطای انتها به انتهای الگوریتم‌های موردبررسی.

۳- پیاده‌سازی عملی تئوری مطرح‌شده.

۴- بررسی عوامل مختلف در مشاهده‌ی نتیجه‌ی شبیه‌سازی و بهینه‌سازی پارامترهای مختلف، ازجمله بررسی سرعت سرورها و جریان‌ها، سرعت مطلوب تولید نشانه در سطل نشتی و ماکزیمم سرعت خروج از سطل.

۵- فراهم آوردن قابلیتی که در آن یک جریان بتواند از همه‌ی سرورهای در دسترس به‌طور هم‌زمان استفاده کند تا میزان بازده در شبکه به‌صورت چشمگیری افزایش یابد و تأخیر تا حد امکان کم شود.

فصل ۶

مراجع

[۱] حلبیان، حسن، (۲۰۰۷) ، پایان‌نامه دکتری، " مطالعه و طراحی الگوریتم‌های زمان‌بندی ترافیک مبتنی بر اولویت مرتب‌شده در شبکه‌های باند وسیع"، دانشکده برق، دانشگاه صنعتی اصفهان.

[2] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the single-node case", *IEEE/ACM Transactions on Networking*, vol. 1, pp. 344-357, 1993.

[3] S. J. Golestani, "A self-clocked fair queueing scheme for broadband applications", *13th Proceedings IEEE INFOCOM '94*, pp. 636-646 vol.2, 1994.

[4] A. Demers, S. Keshav, and S. Shenker, “Analysis and simulation of a fair queueing algorithm,” *SIGCOMM Comput. Commun*. Rev., vol. 19, pp. 1–12, August 1989.

[5] J. C. R. Bennett and Z. Hui, "WF^2Q: worst-case fair weighted fair queueing", *INFOCOM '96, Proceedings IEEE*, pp. 120-128 vol.1, 1996.

[6] M. J. Neely, et al., "Dynamic power allocation and routing for time-varying wireless networks", *IEEE Journal on Selected Areas in Communications*, vol. 23, pp. 89-103, 2005.

[7] N. Matsufuru and R. Aibara “Efficient fair queueing for ATM networks using uniform round robin", *IEICE Trans. Commun*., vol. E83-B, No. 6, pp. 1330-1341, Jun. 2000.

[8] S. Shreedhar and G. Varghese, “Efficient fair queueing using deficit round robin", *IEEE/ACM Trans. Networking*, vol. 4, no. 3,pp. 375-385, Jun. 1996.

[9] D. Verma, H. Zhang, and D. Ferrari, “Guaranteeing delay jitter bounds in packet switching networks," *in Proc. Tricomm '91*, pp. 35-46, Chapel Hill, NC, April. 1991.

[10] H. Zhang and D. Ferrari, “Rate-controlled static priority queueing," *in Proc. IEEE INFOCOM '93*, pp. 227-236, San Francisco, CA, Apr. 1993.

[11] M.Katevenis, S.Sidiropoulos, and C.Courcoubetis, “Weighted round-robin cell multiplexing in a general-purpose ATM switch chip", *IEEE J. Select. Areas Commun*, vol. 9, no. 8, pp. 1265-1279, Oct. 1991.

[12] D. P. Bertsekas and R. G. Gallager, “Data networks: Prentice Hall”, 1992.

[13] S. J. Golestani, "A unified theory of flow control and routing in data communication networks", *PhD, Electrical Engineering, MIT*, 1980.

[14] S. H. Low and D. E. Lapsley, "Optimization flow control. I. Basic algorithm and convergence", *IEEE/ACM Transactions on Networking*, vol. 7, pp. 861 874, 1999.

[15] R. L. Cruz, "A calculus for network delay. I. Network elements in isolation", *IEEE Transactions on Information Theory*, vol. 37, pp. 114-131, 1991.

[16] R. L. Cruz, "A calculus for network delay. II. Network analysis", *IEEE Transactions on Information Theory*, vol. 37, pp. 132-141, 1991.

[17] L. Georgiadis, et al., "Efficient network QoS provisioning based on per node traffic shaping", *IEEE/ACM Transactions on Networking*, vol. 4, pp. 482-501, 1996.

[18] S. J. Golestani, "Network delay analysis of a class of fair queueing algorithms", *IEEE Journal on Selected Areas in Communications*, vol. 13, pp. 1057-1070, 1995.

[19] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks", *IEEE Transactions on Automatic Control*, vol. 37, pp. 1936-1948, 1992.

[20] J.M. Blanquer and B.Ozden, "Fair queuing for aggregated multiple links", *ACM SIGCOMM'2001*, pp. 189-197, Aug.2001.

[21] J. Xu,R.j.Lipton "On Fundamental Tradeoffs between Delay Bounds and Computational Complexity in Packet Scheduling Algorithms" , *Proc. ACM SIGCOMM'02*, pp.279-292, 2002

[22] J. A. Cobb, "A theory of multi-channel schedulers for quality of service", *Journal of High Speed Networks*, vol. 12, no. 1,2, pp. 61-86, 2003.

[23] Y. Jiang, "Relationship between guaranteed rate server andlatency rate server", *Computer Networks*, vol. 43, no. 3, pp. 307-315, Oct. 2003.

[24] S. J. Golestani, "Congestion-free transmission of real-time traffic in packet networks, "*in Prvc. lEEE lNFOCOM 'W*, pp. 527-536, San Fransisco, CA, 1990.

[25] H.Halabian, H. Saidi, R.Changiz. "LVT-SCFQ-A Modfied Self Clocked Fair Queueing Algorithm for Broadband Networks" *Third International Conference on Broadband Communications, IEEE*, p.p 175-180, 2008.

[26] L. Zhang. “Virtual clock: A new traffic control algorithm for packet switching”. *ACM Transactions on Computer Systems*, 9(2):101-124, may 1991.

[27] A. K. Parekh and R. G. Gallager, "A generalized processor sharing approach to flow control in integrated services networks: the multiple node case", *IEEE/ACM Transactions on Networking*, vol. 2, pp. 137-150, 1994.

[28] P. Goyal, H. M.Vin, H. Chen, “Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks,” *SIGCOMM’96,* pp.157–169, September 1996.

[29] P. Goyal, H. M. Vin, H. Chen, “Start-Time Fair Queueing: A Scheduling Algorithm for Integrated Services Packet Switching Networks,” *IEEE/ACM Transaction on networking*, Volume: 5 , Issue:5, pp.690-704, August 2002.

[30] D. Stiliadis, A. Varma, "Latency-rate servers: a general model for analysis of traffic scheduling algorithms"; *IEEE/ACM Transactions on Networking,* Volume 6, Issue 5, pp.611 – 624, October 1998

[31] S.Suri, G.Varghese, G.P.Chandranmenon, “Leap forward virtual clock: a new fair queuing scheme with guaranteed delays and throughput fairness” *IEEE/ACM conference*, volume 2, May 1997.

[32] D.Kwak, N.S.Ko, B.Kim, and H.S.Park.” A new starting potential fair queuing algorithm with O (1) virtual time” *IEEE*, volume 25, pp.475-488, Aug 2003.

[33] F.M. Chiussi and A. Francini, “Minimum-Delay Self-Clocked Fair Queuing Algorithm for Packet-Switched Networks,” *Proc. Of IEEE INFOCOM ’98*, vol. 3, pp. 1112-1121, Mar. 1998.

[34] Ch. Wang; K. Long; Xi. Gong; Sh. Cheng, "SWFQ: a simple weighted fair queueing scheduling algorithm forhigh-speed packet switched network" *Proc. Of ICC 2001*, vol. 8, pp. 2343 - 2347, 2001.

[35] N.ko, "Emulated weighted fair queuing algorithm for high-speed packet-switched networks"*15th IEEE international conference*, pp.52-58, Jan 2001.

[36] J. M. Blanquer and B. Ozden. "Fair queuing for aggregated multiple links". *In Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications, SIGCOMM '01*, p.p 189-197, New York, NY, USA, ACM, 2001.

[37] Y. Bejerano, S.-J. Han, and L. E. Li, "Fairness and load balancingin wireless lans using association control," *in Proceedings of the 10th annual international conference on Mobile computing and networking, ser. MobiCom '04. New York, NY*, pp. 315–329, USA: ACM, 2004.

[38] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, "Dominant resource fairness: fair allocation of multiple resource types," *in Proceedings of the 8th USENIX conference on Networked systems design and implementation, ser. NSDI'11. Berkeley, CA*, pp. 24–24, , USA: USENIX Association,2011.

[39] W.wang, B.liang, B.baochun Li, "Multi-Resource Generalized Processor Sharing for Packet Processing" *IEEE/ACM Trans,Montreal QC*, pp.1-10, June 2013.

[40] J. Vanlerberghe, T. Maertens, J. Walraevens, S. De Vuyst, and H.Bruneel, "A hybrid analytical/simulation optimization of generalized processor sharing," *in Proceedings of The 25th*

International Teletraffic Congress (ITC 25), Shanghai, September 2013.

[41] J. Vanlerberghe, J.walraevens, T. Maertens, H. Bruneel "Approximating the optimal weights for discrete-time generalized processor sharing" *IEEE/ACM Trans, in networking conference, Trondheim,* pp.1-9, June 2014.

[42] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley. "Design, implementation and evaluation of congestion control for multipath tcp". *In Proceedings of the 8th USENIX conference on Networked systems, design and implementation, NSDI'11*, pp. 1-14. Berkeley, USENIX Association, CA, USA, 2011.

[43] H. Xiao. "Analysis of multi-server round robin service disciplines", *Master Thesis at National University of Singapore*, 2005.

[44] E. L. Hahne, "Round-robin scheduling for max-min fairness in datanetworks," *IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS*, vol. 9, pp. 1024–1039, 1991.

[45] Dimitrios Stidilias, "Traffic scheduling in Packet-switched networks: Analysis, Design, and Implementation", *PhD dissertation, University of California, Santa Cruz,* Jun. 1996.

[46] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and D. Towsley, "Multi-path tcp: a joint congestion control and routing scheme to exploit path diversity in the internet," *IEEE/ACM Trans. Netw.*, vol. 14, pp. 1260–1271, December 2006.

[47] E. Nordstrom, D. Shue, P. Gopalan, R. Kiefer, M. Arye, S. Y. Ko, J. Rexford, and M. J. Freedman, "Serval: An end-host stack for service-centric networking," *in Proc. Networked Systems Design and Implementation*, April 2012.

[48] Y. Kok-Kiong, T. Huang, Y. Yiakoumis, S. Chinchali, N. McKeown, and S. Katti. "Scheduling packets over multiple interfaces while respecting user preferences." In *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*, pp. 109-120. ACM, 2013.

[49] A. Ghodsi, V. Sekar, M. Zaharia, and I. Stoica. "Multi-resource fair queueing for packet processing." *In Proceedings of the ACM SIGCOMM 2012 conference on Applications, technologies, architectures, and protocols for computer communication, SIGCOMM '12*, pp.1-12. New York, NY, USA, 2012.

[50] C. Zhou and N. F. Maxemchuk. "Scalable max-min fairness in wireless ad hoc networks." *Ad Hoc Network*, 9(2):112-119, Mar. 2011.

Abstract

The problem of traffic scheduling and scheduling algorithms for servicing data networks have long been investigated from different aspects of quality of service parameters. Nowadays our smartphones have different servers such as WiFi, 3G, 4G, etc. and the number of these servers is growing over time. For each server we may have some preferences. If smartphones use several servers at the same time, it can increase throughput and connectivity for applications like video stream or VOIP, and it can also reduce delay in the network. This means our mobile devices need to schedule packets in keeping with our preferences while making use of all the capacity available. Traditional fair queuing algorithms couldn’t use an application preferences for some servers over others. In this thesis, with vertifying DRR, RR, SPN and FCFS as multiserver and identifying proposed algorithm based on DRR algorithm, average delay and waiting time parameters are obtained, And by using poison distribution as input and leaky bucket as input controller, show that our proposed algorithm has better implementation than other 4 algorithms in quality of service parameters.

Key words: multi server fair queuing, packet scheduling algorithms, virtual time.

Shahrood University of Technology
Department of Electrical and Robotic Engineering

Title

Network flow control based on general processor sharing (GPS) method

Presented for Master of Science in Electrical Engineering

Written by

Azade Asghari

Supervisor

Dr. O.R. Marouzi

February 2016