

روشی برای تضمین تأخیر بسته های موجود در صف RED با استفاده از اتوماتاهای یادگیر

^۱محسن جهانشاهی، ^۲محمد رضا میبیدی

^۱دانشکده مهندسی کامپیوتر، دانشگاه آزاد اسلامی، تهران مرکز، تهران، ایران

^۲دانشکده مهندسی کامپیوتر، دانشگاه صنعتی امیرکبیر، تهران، ایران

E-mail: Mjahanshahi@iauctb.ac.ir

E-mail: Mmeybodi@aut.ac.ir

چکیده - یکی از پارامترهای مهم کیفیت سرویس در شبکه های کامپیوتری، تأخیر بسته های موجود در صف مسیریابها می باشد. برای تضمین این تأخیر، می توان از آلوگوریتمهای کنترل ازدحام در صف مسیریابها استفاده کرد. آلوگوریتم RED مشهورترین و کاربردی ترین آلوگوریتم کنترل ازدحام می باشد. در این مقاله برای اولین بار روشی با استفاده از اتوماتاهای یادگیر برای تضمین تأخیر بسته های موجود در صف مبتنی بر آلوگوریتم RED ارائه می گردد. در این روش حدود آستانه آلوگوریتم RED با استفاده از اتوماتاهای یادگیر بطور پویا با هدف تضمین بیشینه تأخیر تنظیم می گردد. نتایج شبیه سازی نشان می دهد که روش پیشنهادی علاوه بر اینکه بیشینه تأخیر را تضمین می کند، بهره وری درگاه خروجی مسیریاب را نیز افزایش می دهد و در نتیجه کیفیت سرویس بهتری را ارائه می نماید.

کلید واژه - آلوگوریتم RED، اتوماتاهای یادگیر، تضمین تأخیر، کنترل ازدحام

۱- مقدمه

صف در زمان ورود بسته بین دو مقدار دو آستانه بیشینه و کمینه باشد بسته با احتمال p_g (مطابق شکل ۱) دور ریخته می شود. در غیر اینصورت یعنی اگر در زمان ورود بسته طول متوسط صف بیشتر از مقدار آستانه بیشینه باشد بسته با احتمال یک دور ریخته می شود. در این آلوگوریتم احتمال از بین رفتن بسته ها با افزایش طول صف افزایش می یابد. کارایی آلوگوریتم RED بسیار حساس به انتخاب مقدار پارامترهای آن می باشد. برای رسیدن به یک کارایی بهینه پارامترهای RED باید بصورت پویا تنظیم شوند. بنابراین پیکره بندی RED برای دستیابی به یک کارایی قابل پیش بینی کاری پیچیده و دشوار می باشد. مهمترین هدف آلوگوریتم RED کاهش تأخیر بسته های موجود در صف و افزایش میزان بهره وری از لینک متصل به درگاه خروجی مسیریاب می باشد. محدود کردن طول صف باعث کاهش در تأخیر و میزان بهره وری از لینک متصل به درگاه خروجی مسیریاب می شود.

یکی از مهمترین موارد کیفیت سرویس تأخیر بسته های موجود در صف مسیریابها می باشد. مسیریابها به منظور تضمین تأخیر و اعمال موارد ذکر شده در توافق سطح سرویس از آلوگوریتمهای کنترل ازدحام در داخل صفها استفاده می کنند. امروزه آلوگوریتم RED به عنوان مکانیزم پیش فرض کنترل ازدحام در مسیریابها بکار برده می شود [۳۶]. آلوگوریتم RED اولین نشانه های ازدحام را تشخیص داده و با دور ریختن احتمالی بسته ها نسبت به آن عکس العمل نشان می دهد. این آلوگوریتم حجم ترافیک صف را در حد متوسط نگه می دارد [۲۵]. آلوگوریتم RED از دو حد آستانه برای تصمیم گیری در مورد از بین بردن یا وارد کردن بسته به صف استفاده می کند. بدین ترتیب که اگر در زمان ورود بسته طول متوسط صف کمتر از آستانه کمینه باشد بسته وارد صف می شود و اگر طول متوسط

```

Initialization:
avg = 0;
count = -1;
for each packet arrival
calculate the new average queue size avg:
    if the queue is nonempty
         $avg = (1 - w_q) * avg + w_q * queue\_len$ 
    else
         $m = f(current\_time - empty\_queue\_time)$ 
         $avg = (1 - w_q)^m * avg$ 
    if  $min_{th} \leq avg < max_{th}$  {
        increment count
        calculate probability  $p_a$ :
         $p_b = max_p ( avg - min_{th} ) / (max_{th} - min_{th} )$ 
         $p_a = p_b / (1 - count * p_b)$ 
        loss the arriving packet with probability  $p_a$  and then    set count = 0;
    }
    else if  $max_{th} \leq avg$  {
        loss the arriving packet
        count = 0;
    }
    else count = -1;

```

شکل ۱: آلوگوریتم RED

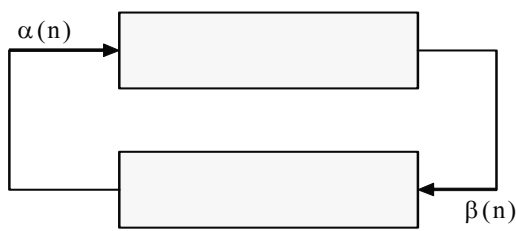
فرض شده است و نمی تواند دارای تغییرات سریع باشد. این فرض با توجه به ترافیکهای انفجاری اینترنت فرض مناسبی نمی باشد. دوم اینکه زمان شبیه سازی تنها ۴۰ ثانیه در نظر گرفته شده است که زمان بسیار کوتاهی می باشد. سوم اینکه روش ارایه شده با RED استاندارد با یک پارامتر خاص مقایسه شده است. مشکل چهارم این است که تعیین طول صف توسط راهبر مشکل می باشد و با توجه به نوع ترافیک و میزان بیشینه بافر صف متغیر می باشد [۱۴]. بهمین دلیل باید به ازای هر شبکه پیکره بندی جداگانه ای انجام شود. در [۱۸] نیز زمان شبیه سازی بسیار کوتاه و برابر ۱۰۰ ثانیه در نظر گرفته شده است. در این مرجع از منطق فازی برای کاهش میزان از دست رفتن بسته های موجود در صف RED بدلیل کمبود فضا در صف استفاده شده است. در این روش میزان بهره وری از لینک متصل به پورت خروجی ۰,۹۹۷ می باشد. اما بطور کلی

شکل ۱ آلوگوریتم RED را نشان می دهد. در این آلوگوریتم پارامتر max_{th} آستانه بیشینه، min_{th} آستانه کمینه، max_p بیشینه احتمال از بین رفتن بسته و p_a احتمال از بین رفتن بسته می باشد. در این آلوگوریتم با استفاده از یک فیلتر پایین گذر و مقدار دهی به w_q بعنوان وزن صف مقدار avg بعنوان متوسط طول صف محاسبه می شود. در هر زمان ورود بسته اگر طول متوسط صف بین دو آستانه باشد متغیر Count یکی افزایش می یابد. در نتیجه مقدار p_a افزایش می یابد که باعث افزایش تعداد دور ریختن بسته ها می گردد.

تحقیقات وسیعی بر روی آلوگوریتم RED صورت گرفته است که در ادامه به ذکر آنها می پردازیم. در [۱۵] سعی شده است تا بطور پویا طول صف، نزدیک به مقداری که راهبر شبکه تعیین می کند نگاه داشته شود. اما این تحقیق دارای ضعفهایی می باشد. اول اینکه، ترافیک "شبه ایستا"^a

۲- اتوماتاهای یادگیر

اتوماتای یادگیر یک مدل انتزاعی است که بطور تصادفی یک عمل از مجموعه متناهی عمل‌های خود را انتخاب کرده و بر محیط اعمال می‌کند. محیط عمل انتخاب شده توسط اتوماتا را ارزیابی کرده و نتیجه ارزیابی خود را توسط سیگنال تقویتی به اتوماتای یادگیر اعلام می‌کند. اتوماتا با استفاده از عمل انتخاب شده و سیگنال تقویتی وضعیت داخلی خود را تغییر داده و سپس عمل بعدی خود را انتخاب می‌کند. شکل ۲ ارتباط بین اتوماتای یادگیر و محیط را نشان می‌دهد [۳-۱].



شکل ۲- ارتباط اتوماتای یادگیر با محیط

محیط را می‌توان توسط سه‌تایی $E = \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودیه‌ها، $\beta = \{\beta_1, \beta_2, \dots, \beta_r\}$ مجموعه خروجیه‌ها و $c = \{c_1, c_2, \dots, c_r\}$ مجموعه احتمالهای جریمه می‌باشد. هرگاه β مجموعه دو عضوی باشد، محیط از نوع P می‌باشد. در چنین محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته می‌شود. در محیط از نوع Q، مجموعه β دارای تعداد متناهی عضو می‌باشد و در محیط از نوع S، مجموعه β دارای تعداد نامتناهی عضو می‌باشد. c_i احتمال جریمه شدن عمل α_i است. اتوماتاهای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم‌بندی می‌گردند. در ادامه اتوماتای یادگیر با ساختار متغیر معرفی می‌شود. اتوماتای یادگیر با ساختار متغیر^b توسط چهارتایی $\{\alpha, \beta, p, T\}$ نشان داده می‌شود که در آن $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عملهای اتوماتا، $\beta = \{\beta_1, \beta_2, \dots, \beta_r\}$ مجموعه ورودیه‌های اتوماتا، $p = \{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب هر یک از عملها و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می‌باشد. نحوه فعالیت این اتوماتا به صورت زیر است. اتوماتا یک عمل از مجموعه عمل‌های خود را به صورت تصادفی و مطابق بردارهای احتمال p_i انتخاب کرده

روشهای مبتنی بر منطق فازی که از یک سری قانون فازی استفاده می‌کنند دارای دو ضعف عمده می‌باشند؛ اول اینکه ممکن است در دنیای واقعی وضعیتی پیش آید که توسط طراح پیش بینی نشده باشد. لذا ممکن است روش با شکست مواجه شود. مشکل دوم این است که اگر در فضای مسأله تغییری هر چند کوچک رخ دهد، طراح می‌بایستی کلیه قوانین را تغییر کند که این باعث کاهش قابلیت گسترش روش می‌شود. در [۲۵] مقدار Max_p برای جریانهای مختلف بصورت پویا تنظیم می‌شود. در [۱۹] [مقادیر استاندارد بیشینه و استاندارد کمینه بصورت دستی تعیین می‌گردد. با این فرض که مقدار استاندارد بیشینه، سه برابر مقدار استاندارد کمینه در نظر گرفته می‌شود. روشهای ارایه شده در [۱۷، ۱۶] نیز مقادیر استاندارد را بطور دستی تنظیم می‌کنند. در این روشها تنظیم چندین پارامتر به طور همزمان کاری بسیار دشوار است. در [۱۴] مقدار حد استاندارد بیشینه حداقل دو برابر مقدار حد استاندارد کمینه در نظر گرفته شده است. در این مرجع اشاره شده است که مقدار W_q بایستی حداقل برابر با ۰،۰۰۱ در نظر گرفته شود. تحقیقات دیگری در این زمینه انجام گرفته است که برای اطلاع از آنها می‌توان به [۲۱، ۲۲ و ۳۲] مراجعه نمود.

گرچه تحقیقات وسیعی بر روی الگوریتم RED صورت گرفته است ولی تا آنجائیکه نویسندگان این مقاله اطلاع دارند تاکنون هیچ تحقیقی در زمینه تضمین تأخیر بسته‌های موجود در صف مبتنی بر الگوریتم RED صورت نگرفته است. در این مقاله روشی مبتنی بر اتوماتاهای یادگیر ارایه می‌گردد. این روش با تنظیم پویای حدود استاندارد الگوریتم RED علاوه بر اینکه بیشینه تأخیر را تضمین می‌کند، بهره‌وری درگاه خروجی مسیریاب را نیز افزایش می‌دهد و در نتیجه کیفیت سرویس بهتری را ارایه می‌نماید.

ادامه این مقاله بدین صورت سازماندهی شده است. در بخش دوم اتوماتاهای یادگیر بطور اختصار معرفی می‌شود. در بخش سوم روش پیشنهادی ارایه می‌گردد. مدل شبیه سازی در بخش چهارم و نتایج شبیه سازیهای انجام گرفته در بخش پنجم آمده است. بخش ششم مقاله نتیجه گیری می‌باشد.

و بر محیط اعمال می‌کند. اگر عمل انتخاب شده α_i باشد، پس از دریافت پاسخ محیط، اتوماتا بردار احتمال عملهای خود را در صورت دریافت پاسخ مطلوب بر اساس رابطه (۱) و در صورت دریافت پاسخ نامطلوب طبق رابطه (۲) بروز می‌کند.

$$\begin{aligned} p_i(n+1) &= p_i(n) + a.(1-p_i(n)) \\ p_j(n+1) &= p_j(n) - a.p_j(n) \quad \forall j \neq i \end{aligned} \quad (1)$$

$$\begin{aligned} p_i(n+1) &= (1-b).p_i(n) \\ p_j(n+1) &= \frac{b}{r-1} + (1-b)p_j(n) \quad \forall j \neq i \end{aligned} \quad (2)$$

در روابط (۱) و (۲)، a پارامتر پاداش و b پارامتر جریمه می‌باشد. اگر a و b با هم برابر باشند، الگوریتم L_{R-P} ،^c و اگر a از b خیلی کوچکتر باشد، الگوریتم L_{REP} ^d و اگر b مساوی صفر باشد، الگوریتم L_{R-I} ^e نام دارد.

۳- روش پیشنهادی

در این بخش روشی مبتنی بر اتوماتای یادگیر بمنظور تضمین تأخیر سه میلی ثانیه برای بسته های موجود در صف مبتنی بر الگوریتم RED ارایه می شود. در این روش که آنرا LARED می نامیم از یک اتوماتای یادگیر با ساختار متغیر از نوع L_{R-P} استفاده شده است. در ابتدای شبیه سازی احتمال انتخاب همه عملها با هم مساوی و برابر $\frac{1}{7}$ در نظر گرفته شده است. اتوماتای یادگیر دارای ۷ عمل می‌باشد که در جدول ۱ شرح داده شده اند. در جدول ۱ مقدار value0 صفر می باشد. یعنی تغییری در مقدار آستانه مربوطه ایجاد نمی شود. همچنین مقدار value1 عدد دو کیلو بیت در نظر گرفته شده است. بعنوان مثال اگر اتوماتای یادگیر عمل شماره چهار را انتخاب کند آنگاه به میزان دو کیلو بیت مقدار آستانه کمینه کاهش می یابد و مقدار آستانه بیشینه تغییری نمی کند.

در روش پیشنهادی دو الگوریتم بصورت موازی توسط مسیریاب اجرا می‌شوند؛ الگوریتم RED معمولی با ورود هر بسته و الگوریتم اتوماتای یادگیر هر دو ثانیه یکبار اجرا می‌شوند. اتوماتای یادگیر مقادیر آستانه الگوریتم RED معمولی را محاسبه می کند و الگوریتم RED معمولی نیز از آخرین مقادیر آستانه تنظیم شده توسط اتوماتای یادگیر

استفاده می کند. عملکرد اتوماتای یادگیر برای تنظیم مقادیر آستانه در ادامه ذکر می گردد؛ در ابتدا اتوماتای یادگیر یکی از اعمال خود را بر طبق بردار احتمال اعمال انتخاب می کند. عمل انتخاب شده توسط مسیریاب اعمال می شود. پس از گذشت دو ثانیه از انجام این عمل تأخیر صف RED مجدداً محاسبه می شود. اگر این تأخیر پایین تر از حد بیشینه تأخیر باشد یا تفاضل تأخیر فعلی با حد بیشینه کوچکتر از تفاضل تأخیر قبلی با حد بیشینه باشد سیگنال تقویتی β برابر با صفر و در غیر اینصورت برابر با یک در نظر گرفته می شود. باتوجه به مقدار β بردار احتمال اعمال بر طبق الگوریتم یادگیری L_{R-P} بروز می-شود. پس از بروز رسانی بردار احتمالات اتوماتای یادگیر مجدداً یکی از اعمال خود را انتخاب می کند.

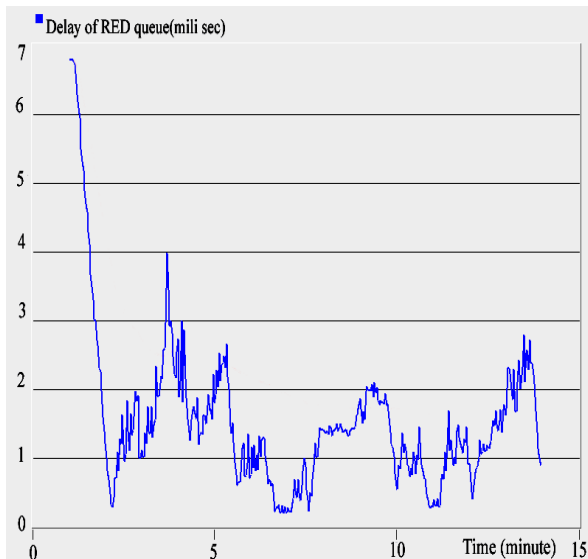
جدول ۱: اعمال اتوماتای یادگیر

اثر هر عمل	اعمال اتوماتای یادگیر
MaxTh = MaxTh + value1	۰
MaxTh = MaxTh - value1	۱
MaxTh = MaxTh + value0	۲
MaxTh = MaxTh + value1	۳
MaxTh = MaxTh + value0	۴
MaxTh = MaxTh - value1	۵
MaxTh = MaxTh + value0	۶

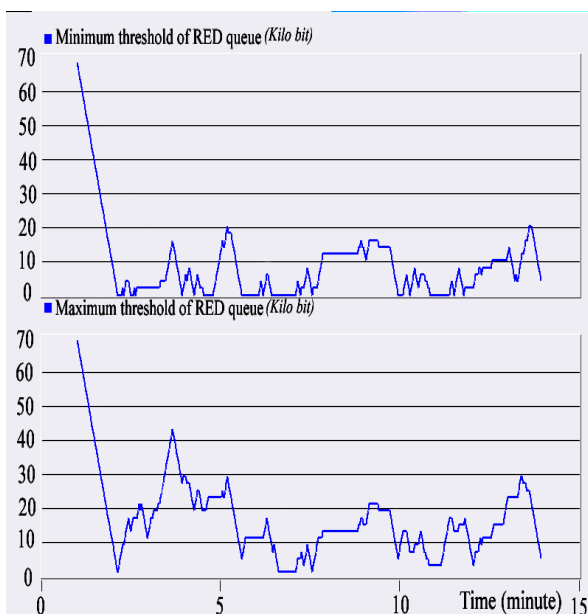
۴- مدل شبیه سازی

برای ارزیابی روش پیشنهادی از مدل شبکه شکل ۳ که با استفاده از نرم افزار شبیه ساز OpNet پیاده سازی شده است استفاده می گردد. در این مدل یک سرور بنام self-similar traffic generator وجود دارد که مولد ترافیک انفجاری به مقصد مسیریاب مبتنی بر الگوریتم RED می‌باشد.

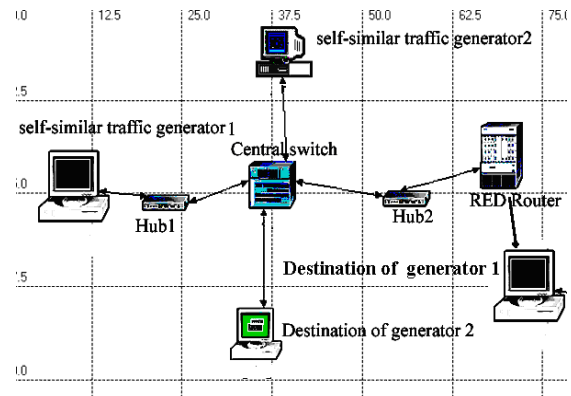
شده در [۲۵ و ۱۵، ۱۸] مقدار آستانه بیشینه نبایستی همیشه سه یا چهار برابر مقدار آستانه کمینه در نظر گرفته شود. همانطور که در بخش اول از این مقاله اشاره شد پارامتر *avg* متوسط طول صف می باشد. این مقدار تخمینی از طول واقعی صف می باشد. نمودار بالا از شکل ۶ مربوط به تغییرات *avg* و نمودار پایین از این شکل مربوط به تغییرات طول واقعی صف در این شبیه سازی می باشد.



شکل ۴: اتوماتای یادگیر تصادفی تأخیر سه میلی ثانیه را در صف RED تضمین می کند.



شکل ۵: تغییرات آستانه بیشینه و کمینه الگوریتم RED



شکل ۳: مدل شبکه روش LARED

از آنجاییکه توزیع پواسن برای مدل کردن ترافیک شبکه مناسب نیست در شبیه سازیهای انجام شده در این تحقیق از ترافیک خود همانند به منظور تولید ترافیک انفجاری استفاده شده است [۳۵]. مقدار پارامتر آشفستگی^f که شدت انفجاری بودن ترافیک خود همانند را تعیین می کند ۰,۷ لحاظ شده است. به منظور ایجاد بار ترافیکی بر روی سوئیچ مرکزی یک سرور بنام *self-similar traffic generator2* مولد ترافیک انفجاری به مقصد *Destination of generator2* می باشد. در این مدل بافر صف RED، ۷۰ کیلو بیت و نرخ سرویس، سرویس دهنده ۱۰ کیلو بیت بر ثانیه در نظر گرفته شده است. همچنین طول بسته ها نیز ۱ کیلو بیت لحاظ شده است. در این تحقیق شبیه سازیها بر روی یک کامپیوتر شخصی پنتیوم چهار ۱,۷ گیگا هرتز با ۲۵۶ مگا بایت حافظه اصلی انجام شده است. سیستم عامل مورد استفاده Advance Server 2000 می باشد. همچنین در این تحقیق زمان شبیه سازی پانزده دقیقه در نظر گرفته شده است.

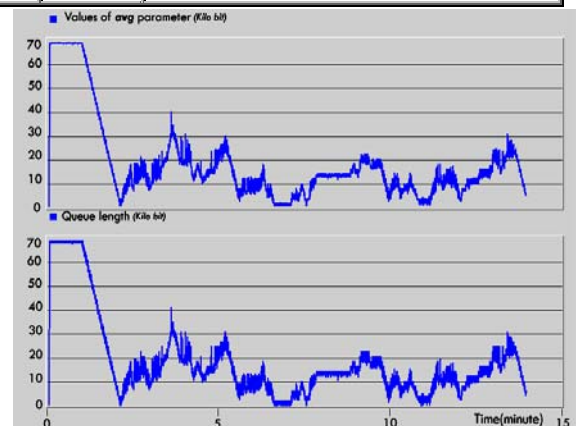
۵- نتایج شبیه سازی

شکل ۴ نتیجه شبیه سازی انجام شده را نشان می دهد. در این شبیه سازی هدف تضمین بیشینه تأخیر سه میلی ثانیه برای صف RED می باشد. همانطور که در این شکل ملاحظه می شود اتوماتای یادگیر تأخیر سه میلی ثانیه را با متوسط ۱,۵۱ میلی ثانیه در طول پانزده دقیقه شبیه سازی بخوبی تضمین می کند. شکل ۵ تغییرات آستانه بیشینه و کمینه الگوریتم RED را در این شبیه سازی نشان می دهد. متوسط آستانه بیشینه برابر با ۱۶۶۱۵ بیت و متوسط آستانه کمینه برابر با ۸۵۷۴ بیت (تقریباً دو برابر) می باشد. از این آزمایش می توان نتیجه گرفت که بر خلاف روشهای رایج

جدول ۲: تأثیر تغییرات W_q در تفاوت بین میانگین طول واقعی و متوسط صف (avg)

W_q	تغییرات طول متوسط صف RED (avg)			تغییرات طول واقعی صف RED			تفاوت بین مقادیر میانگین طول واقعی و متوسط صف (avg)
	کمینه	متوسط	بیشینه	کمینه	متوسط	بیشینه	
۰/۱	۰/۸۱	۲۷۲۰۵	۶۸۵۵۸	۰	۲۷۲۰۴	۶۸۶۰۸	۱
۰/۳	۰/۴۹	۲۲۸۰۷	۶۸۶۰۵	۰	۲۲۸۱۳	۶۸۶۰۸	۶
۰/۴	۰/۳۶	۱۶۷۱۵	۶۸۶۰۷	۰	۱۶۷۱۳	۶۸۶۰۸	۲
۰/۵	۰/۲۵	۲۰۵۸۸	۶۸۶۰۷	۰	۲۰۵۸۴	۶۸۶۰۸	۴
۰/۷	۰/۰۹	۱۹۶۲۹	۶۸۶۰۸	۰	۱۹۶۳۱	۶۸۶۰۸	۳
۰/۹	۰/۰۱	۳۶۳۴۱	۶۸۶۰۸	۰	۳۶۳۴۱	۶۸۶۰۸	۰

به عبارت دیگر مقدار پارامتر W_q بایستی با توجه به طول بافر صف و شرایط ترافیکی تعیین شود. جدول ۴ تأثیر تغییرات پارامتر Max_p بر روی میزان تأخیر بسته های موجود در صف RED را نشان می دهد. در این جدول ستون زمان یادگیری مشخص می کند که پس از چند ثانیه اتوماتای یادگیر توانسته است میزان تأخیر بسته های موجود در صف RED را پایین تر از حد بیشینه تعیین شده آورده و در حدود پانزده دقیقه زمان شبیه سازی این تأخیر را زیر این حد نگاه دارد.



شکل ۶: تغییرات طول واقعی و متوسط صف (avg)

جدول ۳: تأثیر تغییرات W_q بر روی میزان تأخیر بسته های موجود در صف RED

α	W_q	Max_p	ارزیابی
۰,۵	۰,۱	۰,۷	نا مطلوب
۰,۵	۰,۳	۰,۷	نا مطلوب
۰,۵	۰,۴	۰,۷	مطلوب
۰,۵	۰,۵	۰,۷	نا مطلوب
۰,۵	۰,۷	۰,۷	نا مطلوب

در جدول ۴ نرخ خرابی بسته ها تا زمان یادگیری اتوماتا نیز ارایه شده است. بطور مثال اگر مقدار پارامتر Max_p برابر با ۰,۹ لحاظ شود تعداد ۹۹۷ بسته از بین می رود تا تأخیر بسته های موجود در صف RED پایین تر از حد بیشینه تعیین شده آورده شود. جدول ۵ نیز تأثیر تغییرات پارامتر α بر روی میزان تأخیر بسته های موجود در صف RED را

جدول ۲ تأثیر تغییرات W_q در مقادیر کمینه، متوسط و بیشینه طول واقعی صف و طول متوسط صف (avg) و همچنین تفاوت بین مقادیر متوسط این دو را در طول این شبیه سازی نشان می دهد. مقداری برای W_q مناسب است که تفاوت بین طول واقعی صف و طول متوسط صف (avg) را نه خیلی زیاد و نه خیلی کم کند. همانطور که در جدول ۲ ملاحظه می شود اگر مقدار پارامتر W_q برابر با ۰/۴ اختیار شود تأخیر سه میلی ثانیه تضمین می شود. جدول ۳ تأثیر تغییرات W_q بر روی میزان تأخیر بسته های موجود در صف RED را نشان می دهد. منظور از کلمه مطلوب در این جدول این است که روش پیشنهادی توانسته است بیشینه تأخیر را تضمین کند و منظور از کلمه نا مطلوب این است که روش پیشنهادی نتوانسته است بیشینه تأخیر را تضمین کند. بنابراین نتایج نشان می دهند که برخلاف روشهای ارایه شده در [۱۹ و ۱۵، ۱۴] مقدار پارامتر W_q نبایستی همواره یکی از مقادیر ۰,۰۰۱، ۰,۰۰۲ و یا ۰,۰۰۵ اختیار شود.

جدول ۴: تأثیر تغییرات پارامتر Max_p بر روی میزان تأخیر بسته های موجود در صف RED

α	w_q	زمان یادگیری (ثانیه)	نرخ خرابی بسته ها تا زمان پایین آوردن تأخیر	Max_p	ارزیابی
۰,۵	۰,۴	~	۹۹۴	۰,۱	نا مطلوب
۰,۵	۰,۴	~	۹۹۱	۰,۳	نا مطلوب
۰,۵	۰,۴	~	۹۹۹	۰,۵	نا مطلوب
۰,۵	۰,۴	۲۶	۹۹۳	۰,۶	نسبتا مطلوب
۰,۵	۰,۴	۲۶	۹۹۴	۰,۷	مطلوب
۰,۵	۰,۴	۲۶	۱۰۵۹	۰,۸	نسبتا مطلوب
۰,۵	۰,۴	۲۶	۹۹۷	۰,۹	مطلوب

۶- نتیجه

در این مقاله آلفوریتمی مبتنی بر اتوماتاهای یادگیر برای تضمین کیفیت سرویس در صف RED ارائه گردید. در این روش حدود آستانه آلفوریتم RED با استفاده از اتوماتاهای یادگیر بطور پویا با هدف تضمین بیشینه تأخیر تنظیم می گردد. برای این منظور اتوماتای یادگیر نیازی به اطلاعات اولیه درباره مشخصات ترافیکی ندارد. نتایج شبیه سازی نشان داد که روش پیشنهادی علاوه بر اینکه بیشینه تأخیر را تضمین می کند، بهره وری درگاه خروجی مسیریاب را نیز افزایش می دهد و در نتیجه کیفیت سرویس بهتری را ارائه می نماید.

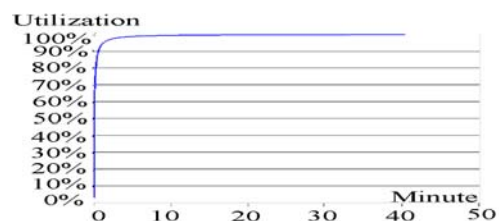
مراجع

- [1] K. S. Narendra, M. L. Thathachar, "Learning Automata: An Introduction", Englewood Cliffs, NJ: Prentice Hall, 1989.
- [2] V. I. Varshavski, I. P. Vorontsova, "On the Behavior of Stochastic Automata with Variable Structure", Automata, Telemekh, Vol. 24, pp. 253-360, 1963.
- [3] K. S. Fu, "Stochastic Automata as Models of Learning Systems", Computer And Information Sciences II, J.T. Lou, Editor, New York: Academic, 1967.
- [4] K. Najim, A. S. Poznyak, "Learning Automata: Theory and Application", Tarrytown, NY: Elsevier Science Ltd., 1994.
- [5] P. Bhoj, S. Singhal, S. Chutani, "SLA Management In Federated Environments", The Sixth IFIP/IEEE International Symposium On Integrated Network Management (IM'99), Boston, MA, 1999.
- [6] E. Bouillet, D. Mitra, K. G. Ramakrishnan, "The Structure and Management of Service Level Agreements in Networks", IEEE Journal On Selected Areas In Communications, Vol. 20, No. 4, 2002.
- [7] S. Frolund, M. Jain, J. Pruyne, "Monitoring End-User Service Levels", Integrated Network Management VI (Edited By M. Sloman et al), IEEE Publication, 1999.

نشان می دهد. شکل ۷ میزان بهره وری از درگاه خروجی مسیریاب را نشان می دهد. بهره وری به معنی درصد ارسال بسته ها از طریق لینک متصل به درگاه خروجی نسبت به ظرفیت کل لینک می باشد. همانطور که در این نمودار ملاحظه می شود، در روش پیشنهادی میزان بهره وری لینک متصل به درگاه خروجی مسیریاب ۰,۹۹۸۵ می باشد و با گذشت زمان به ۱ نزدیکتر می شود در حالیکه در [۱۸] میزان بهره وری از لینک متصل به پورت خروجی حداکثر ۰,۹۹۷ می باشد.

جدول ۵: تأثیر تغییرات α بر روی میزان تأخیر بسته های موجود در صف RED

α	w_q	Max_p	ارزیابی
۰,۱	۰,۴	۰,۷	نا مطلوب
۰,۳	۰,۴	۰,۷	نا مطلوب
۰,۴	۰,۴	۰,۷	نا مطلوب
۰,۵	۰,۴	۰,۷	مطلوب
۰,۶	۰,۴	۰,۷	نا مطلوب
۰,۷	۰,۴	۰,۷	نا مطلوب
۰,۹	۰,۴	۰,۷	نا مطلوب



شکل ۷: میزان بهره وری از لینک متصل به درگاه خروجی مسیریاب

- [28] T. C. Wan, S. K. Joo, "Random Early Drop With In & Out (RIO) For Asymmetrical Geostationary Satellite Links", Proceedings of Joint Int'l Conf. IEEE MICC 2001, LiSLO 2001, ISCE 2001, Kuala Lumpur, Malaysia, 2001.
- [29] F. Anjum, L. Tassiulas, "Balanced-RED: An Algorithm to Achieve Fairness in the Internet", http://WWW.Isr.Umd.Edu/Techreports/ISR/1999/TR_99-17/TR_99-17.Phtml.
- [30] <http://WWW.Nanog.Org/Mtg-0102/ppt/Diot/Sld005.Htm>.
- [31] Cisco IOS Release 12.2(8)T .
- [32] W. Feng, D. Kandlur, D. Saha, K. Shin, "Blue: A New Class of Active Queue Management Algorithms", University Of Michigan Technical Report CSE-TR-387-99, April 1999.
- [33] D. Lin, R. Morris, "Dynamics of Random Early Detection", Proceedings of SIGCOMM 97, pp. 127-138, September 1997.
- [34] M. L. Tsetlin, "Automaton Theory and Modeling of Biological Systems", Vol, 102 in Mathematics in Science and Engineering, New York: Academic Press, 1973.
- [35] V. Paxson, S. Floyd, "Wide-Area Traffic: The Failure of Poisson Modeling", Sigcomm 94. September 1994.
- [36] B. Braden et al, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998.

زیر نویس ها

- a Quasi-stationary
- b Variable structure
- c Linear Reward-Penalty
- d Linear Reward epsilon Penalty
- e Linear Reward Inaction
- f Hurst

- [8] J. L. Hellerstein, F. Zhang, P. Shahabuddin, "An Approach to Predictive Detection for Service Management", Integrated Network Management VI (Edited By M. Sloman Et Al). IEEE Publication, 1999.
- [9] A. Kuepper, C. Popien, B. Meyer, "Service Management Using Up-To-Date Quality Properties", The IFIP/IEEE International Conference on Distributed Platforms: Client/Server And Beyond: DCE, CORBA, ODP And Advanced Distributed Applications. Chapman and Hall, 1996.
- [10] M. Sloman (Editor), "Network and Distributed System Management", Addison-Wesley, 1994.
- [11] L. Lewis, "Service Level Management for Enterprise Networks", Artech House, 1999.
- [12] E. Wustenhoff, "Service Level Agreement in the Data Center", Sun Professional Services 2002, <http://WWW.Sun.Com/Blueprints>.
- [13] D. Goderis, et al, "Service Level Specification Semantics, Parameters and Negotiation Requirements", IETF Internet Draft, Jun 2001.
- [14] S. Floyd, V. Jacobson, "Random Early Detection Gateways for Congestion Avoidance", IEEE/ACM Transactions on Networking, Vol. 1, pp. 397-413, August 1993.
- [15] T. Ye and S. Kalyanaraman, "Adaptive Tuning Of RED Using On-Line Simulation", GLOBECOM, Vol. 3, pp. 2210-2214, 2002.
- [16] C. Chien, W. Liao, "A Self-Configuring RED Gateway for Quality of Service (QoS) Networks", Proceedings of IEEE ICME, July 2003.
- [17] J. Orozco, D. Ros, "An Adaptive RIO (A-RIO) Queue Management Algorithm", Lecture Notes in Computer Science, Springer, 2003.
- [18] L. Rossides, A. Sekercioglu, A. Pitsillides, A. Vasilakos, S. Kohler, P. Tran-Gia, "Fuzzy RED: Congestion Control For TCP/IP Diff-Serv", Electrotechnical Conference, MELECON 2000, Vol.1, pp. 19-22, vol.1, 2000.
- [19] M. Christiansen, K. Jeffay, D. Ott and F. Donelson, "Tuning RED for Web Traffic", Smith Chapel Hill, NC 27599-3175 USA, <http://WWW.Cs.Unc.Edu/Research/Dirt>.
- [20] F. Anjum, L. Tassiulas, "A Self-Configuring RED Gateway", Proceedings of INFOCOM '99, March 1999.
- [21] T. Ott, T. Lakshman, L. Wong, "SRED: Stabilized RED", Proceedings of IEEE INFOCOM '99, pp. 1346-1355, 1999.
- [22] M. Maowidzki, "Simulation-Based Study of ECN Performance in RED Networks", Military Communication Institute 05-130 Zegrze, Poland, http://www.republika.pl/maom_onet/papers/ecn/ecn-spects03.pdf.
- [23] B. Prabhakar, K. Psounis, "CHOKe: A Stateless Active Queue Management Scheme for Approximating Fair Bandwidth Allocation", Proceedings of IEEE INFOCOM'00, Tel-Aviv, Israel, pp. 942-951, Apr. 2000.
- [24] V. Jacobson, "Congestion Avoidance and Control", SIGCOMM '88PP, 314-329, August 1988.
- [25] V. Vukadinovic, L. Trajkovic, "RED with Dynamic Thresholds for Improved Fairness", School of Engineering Science Simon Fraser University Vancouver, BC, Canada, 2000.
- [26] W. Leland, M. Taqqu, W. Willinger, D. Wilson, "On The Self-Similar Nature Of Ethernet Traffic (Extended version)", Proceedings of IEEE/ACM Transactions On Networking, 2(1), PP. 1-15, February 1994.
- [27] W. C. Feng, D. D. Kandlur, D. Saha, K.G. Shin, "A Self-Configuring RED Gateway", Proceedings of INFOCOM '99, Eighteenth Annual Joint Conference Of The IEEE Computer And Communications Societies. Proceedings. Volume 3, pp. 1320 - 1328, Vol. 3, 1999.