

«بناه آنکه جان را فکرت آموخت»

مجموعه مقالات

هشتمین کنفرانس مهندسی برق ایران

۱۳۷۹-۳۰ اردیبهشت ماه

جلد چهارم خبرات

دانشگاه صنعتی اصفهان

دانشکده مهندسی برق و کامپیوتر



کنترل درخواست ورود در شبکه‌های ATM با استفاده از اتماتانهای یادگیر با ساختار متغیر

حمید رضا شهریاری محمد رضا میبدی

دانشگاه صنعتی امیرکبیر

دانشکده مهندسی کامپیوتر

(shahriar@ce.aku.ac.ir)

چکیده

یکی از مسائل مهم در شبکه‌های ATM، کنترل درخواست ورود است، که سعی در جلوگیری از بروز تراکم در شبکه با استفاده از قبول یا رد درخواستهای اتصال می‌کند. روش‌های متعددی از جمله روش پهنه‌ای باند معادل و روش تقریب تراکم سنگین و یا تخمین حد بالای اتلاف سلول ارائه شده‌اند. در این مقاله کنترل درخواست ورود با استفاده از اتماتانهای یادگیر با ساختار متغیر بررسی شده است. اتماتان یادگیر می‌تواند با استفاده از تجربیات سابق خود، خود را با وضعیت موجود شبکه تطبیق داده و در مورد قبول درخواستها تصمیم گیری نماید. در این روش از اتماتان یادگیر برای تخمین ظرفیت معادل برای تعدادی منبع ترافیکی مشابه استفاده می‌شود. در این مقاله رفتار الگوریتم‌های مختلف یادگیری از جمله الگوریتم‌های یادگیری خطی استاندارد، سلسله‌مراتبی و یک الگوریتم یادگیری جدید برای کنترل درخواست ورود بررسی می‌شوند. برای این منظور با استفاده از شبیه‌سازی محیط شبکه ATM، الگوریتم‌های مورد نظر از لحاظ بهره‌وری خط، تعداد درخواستهای پذیرفته شده و میزان اتلاف سلول مقایسه شده‌اند.

کلمات کلیدی: شبکه‌های ATM، کنترل تراکم، کنترل درخواست ورود، اتماتان یادگیر

۱ مقدمه

ATM به عنوان راهکاری برای پیاده‌سازی B-ISDN پیشنهاد شده است^[۱]. یکی از مباحث مهم در شبکه‌های ATM کنترل تراکم است. وظيفة اصلی روال کنترل تراکم حفاظت از شبکه و کاربران در مقابل کاهش کارایی شبکه و استفاده بهینه کردن از منابع موجود است. روش‌های کنترل تراکم می‌تواند به دو دسته بازدارنده و انفعالی تقسیم شود. در روش‌های بازدارنده از بروز تراکم جلوگیری می‌شود، در حالی که در روش‌های انفعالی از اطلاعات پس خور شبکه برای کنترل میزان تراکم استفاده می‌شود. یکی از روش‌های مهم در روش‌های بازدارنده کنترل درخواست ورود (CAC) است. از آنجا که ATM یک سرویس مبتنی بر اتصال است، قبل از آنکه کاربر شروع به انتقال داده روی شبکه ATM کند، یک اتصال باید ایجاد شود. CAC با این سوال در رویه‌رو است که آیا سویچ می‌تواند یک اتصال جدید را قبول کند یا خیر. به علت وجود مشکلاتی نظری مشکل تعیین و توصیف فرآیند ترافیک ورودی از کاربر و لزوم پاسخ و تصمیم گیری سریع در مورد پذیرش یا عدم پذیرش ارتباط، پیاده‌سازی CAC بر مبنای تخصیص آماری دشوار است^[۲].

¹ Congestion Control

² Call Admission Control

البته روش‌های متعددی ارائه شده است که در آنها با اتخاذ تقریب‌های لازم و بدون حل دقیق معادلات فوق از پیچیدگی مسئله کاسته شده است. از جمله روش‌هایی که سعی در تقریب یکی از پارامترهای سیستم می‌کنند، می‌توان به روش پهنه‌ای باند معادل^[۳] و تقریب ترافیک‌های سنگین^[۴] اشاره کرد. در این روشها با استفاده از مدل کردن منبع ترافیکی به یک توزیع شناخته شده و تبدیل مسئله به یک مسئله صفر، سعی در تقریب یک پارامتر تصمیم‌گیری مانند پهنه‌ای باند لازم یا احتمال اتلاف سلول می‌کنند. این روشها به یک توزیع خاص برای منابع ترافیکی متنکی هستند. اگر منابع از این مدل پیروی نکنند، این روشها تخمین درستی نخواهد داشت. از همین رو روش‌های پویا با قابلیتها یادگیری و استفاده از تجربیات سابق خود مورد توجه هستند. در این روشها یک الگوریتم یادگیر با مشاهده رفتار سیستم خود را با آن وفق می‌دهد و بر اساس رفتار مشاهده شده منابع، در مورد درخواستهای جدید تصمیم می‌گیرد. روش‌های متعددی برای کنترل درخواست ورود با قابلیت یادگیری ارائه شده است. از جمله می‌توان به کنترل ترافیک با استفاده از شبکه عصبی^[۵]، یا استفاده از منطق فازی^[۶]، یا استفاده از اتوماتان یادگیر^[۷] و یا با استفاده از تخمین پویای آستانه^[۸] اشاره کرد.

در [۷] از اتوماتان یادگیر برای کنترل درخواست ورود استفاده شده است. در این روش از یک اتوماتان یادگیر تصادفی تخمین زننده (SELA) استفاده شده است. برای استفاده در کنترل درخواست ورود دو عمل $\{\alpha_0, \alpha_1\}$ برای اتوماتان تعریف شده است که α_0 نشان دهنده قبول درخواست جدید و α_1 نشان دهنده رد درخواست است. این اتوماتان با هر درخواست جدید یکی از اعمال α_0 یا α_1 را انتخاب کرده و سپس با استفاده از تخمین پهنه‌ای باند معادل پاسخ محیط را بدست آورده و بردار احتمالهای متناظر با اعمال α_0 را بهنگام می‌کند.

در این مقاله از اتوماتانهای یادگیر برای کنترل درخواست ورود استفاده می‌شود. بدین منظور برای هر اتصال تخمینی از میزان پهنه‌ای باند لازم می‌زند و با استفاده از این تخمین درباره پذیرش درخواستها تصمیم می‌گیرد. اتوماتان به مرور زمان با دریافت عکس‌عملهای محیط، خود را با محیط وفق داده و تخمین بهتری ارائه می‌دهد. در ادامه این مقاله در بخش ۲ اتوماتان یادگیر با ساختار متغیر توضیح داده شده است. در بخش ۳ چگونگی کنترل درخواست ورود با استفاده از اتوماتانهای یادگیر با ساختار متغیر ارائه شده است. در انتها در بخش ۴ با استفاده از شبیه‌سازی شبکه ATM کارایی الگوریتمهای مختلف یادگیری از جمله الگوریتمهای یادگیری خطی استاندارد، الگوریتم پیشنهادی و اتوماتان سلسله‌مراتبی از جنبه‌های گوناگون همچون حداکثر اتصالهای پذیرفته شده، میزان کارایی خط، درصد سلولهای از دست رفته و حداقل تأخیر سلولها مقایسه شده است.

۲ اتوماتان یادگیر

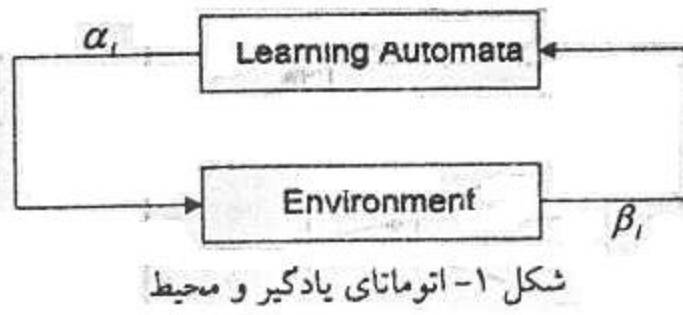
اتوماتان یادگیر یک اتوماتان است که در یک محیط تصادفی عمل می‌کند و احتمالهای هر یک از اعمال را بر حسب ورودیهایی که از محیط دریافت می‌کند، بهنگام می‌کند. روش عملکرد یک اتوماتان یادگیر به این صورت است که یک عمل از مجموعه مجاز اعمال انتخاب کرده و عمل انتخاب شده را به یک محیط تصادفی اعمال می‌کند. در مقابل محیط با یک سینگال پس‌خور پاسخ می‌دهد. این پاسخ باعث می‌شود که وضعیت داخلی اتوماتان برای عمل بعد بهنگام شود. اتوماتانهای یادگیر به دو گروه ساختار ثابت و متغیر دسته‌بندی می‌شوند. شکل ۱ نحوه ارتباط یک اتوماتان یادگیر با محیط نشان می‌دهد.

یک اتوماتان با ساختار متغیر را می‌توان با شش تابی $\{\beta, \phi, \alpha, P, G, T\}$ تعریف کرد که β مجموعه عملهای ورودی، ϕ مجموعه حالت‌های داخلی، α مجموعه خروجیها، P بردار احتمالهای انتخاب عملها است که در هر مرحله یکی انتخاب می‌شوند، G نگاشت خروجی و T الگوریتم یادگیری است^[۹].

¹ Equivalent Capacity

² Heavy Traffic Approximation

³ Stochastic Estimator Learning Algorithm



شکل ۱- اتوماتای یادگیر و محیط

واضح است که عامل مؤثر در کارایی اتوماتان یادگیر با ساختار متغیر الگوریتم یادگیری است که برای بهنگام اکردن احتمالهای هر یک از عملها استفاده می‌شود. الگوریتمهای یادگیری متعددی در [۹] و [۱۰] ارایه شده است. فرض کنید α_i در زمان k مطابق توزیع $p(k)$ انتخاب شده است. الگوریتم خطی زیر یکی از متدائلترین الگوریتمهای یادگیری است:

اگر پاسخ محیط مطلوب باشد:

$$p_j(k+1) = \begin{cases} p_j(k) + a[1 - p_i(k)] & j = i \\ (1-a)p_j(k) & j \neq i \end{cases} \quad (1)$$

$$p_j(k+1) = \begin{cases} (1-b)p_j(k) & j = i \\ \frac{b}{r-1} + (1-b)p_j(k) & j \neq i \end{cases} \quad (2)$$

اگر پاسخ محیط نامطلوب باشد:

که a پارامتر پاداش و b پارامتر جریمه است ($a, b < 0$). a میزان پاداش (افزایش احتمال) و b میزان جریمه (کاهش احتمال) را تعیین می‌کند. در صورتی که $a=b=0$ الگوریتم linear reward-inaction ($L_{R,I}$) و در صورتی که $a=b$ باشد، الگوریتم linear reward-penalty ($L_{R,P}$) بدست می‌آید و اگر $a > b$ باشد، الگوریتم L_{R-P} بدست می‌آید.

از اتوماتان یادگیر در کاربردهای بسیاری از جمله در مسیریابی بسته‌های داده‌ای [۱۱]، کنترل جریان در شبکه [۱۲] و کنترل ترافیک با شکل دادن ترافیک شبکه [۱۳]، کنترل صف [۱۴]، مهندسی شبکه‌های عصبی [۱۵]، [۱۶] و تناظر گراف [۱۷] استفاده شده است. برای مطالعه بیشتر درباره اتوماتانهای یادگیر و کاربردهای آن به [۹]، [۱۰]، [۱۸]، [۱۹]، [۲۰] و [۲۱] مراجعه نمایید.

۳ کنترل درخواست ورود با استفاده از اتوماتانهای یادگیر

در این قسمت نحوه کار الگوریتم توضیح داده می‌شود. اگر N منبع ترافیکی مشابه به یک سونیج متصل شده باشند، در صورتی که ظرفیت خط خروجی سونیج C باشد، باید $c < N < C$ که c ظرفیت واقعی مورد نیاز برای هر اتصال است. c باید با توجه به خصوصیات منبع ترافیکی و کیفیت سرویس مورد نظر تعیین شود.

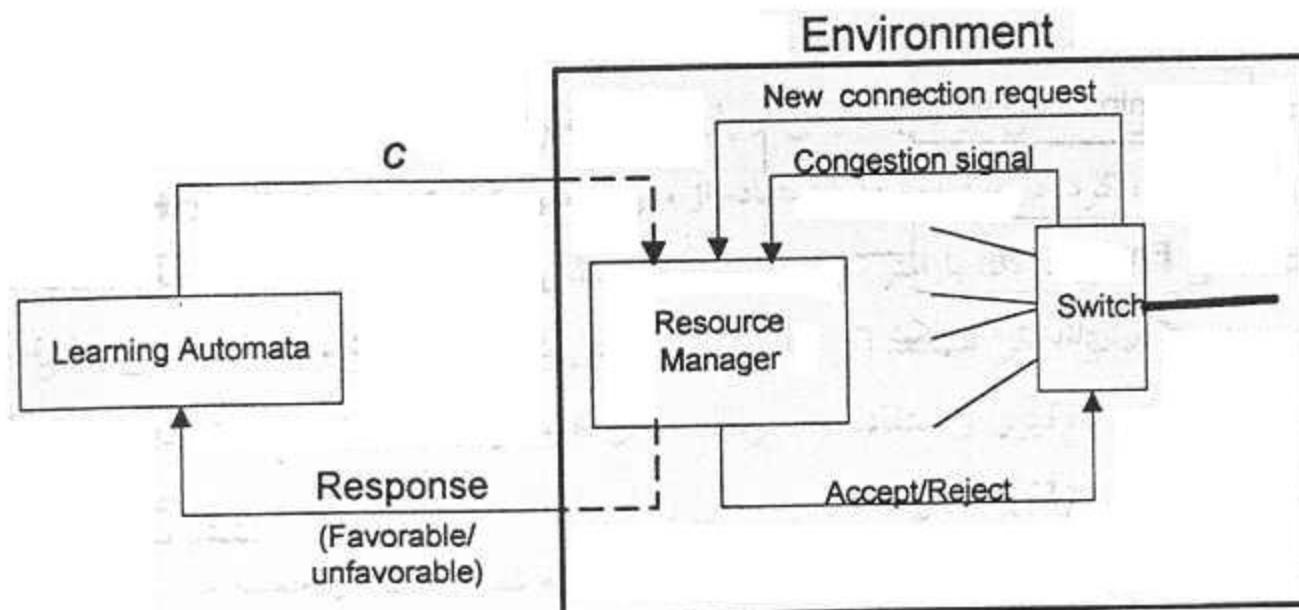
اگر محدوده یک پارامتر مشخص باشد، می‌توان مقدار آن را با اتوماتان یادگیر تخمین زد. بدین صورت که بازه بین حداقل و حداکثر مقدار ممکن برای پارامتر را به تعدادی بازه مساوی تقسیم کرده و به هر بازه یک عمل اتوماتان نسبت داده می‌شود. اتوماتان در هر بار یک عمل را با توجه به احتمال انتخاب آن انتخاب می‌کند و به محیط می‌دهد. محیط پس از اعمال کردن عمل اتوماتان، پاسخی مبنی بر مطلوب یا نامطلوب بودن عمل به اتوماتان داده و اتوماتان با استفاده از الگوریتم یادگیری بردار احتمال را بهنگام می‌کند. در شکل ۲ نحوه تعامل اتوماتان با محیط برای کنترل درخواست ورود مشاهده می‌شود.

در رابطه $c < N < C$ با توجه به اینکه منابع از نوع VBR هستند، $R < c < C$ خواهد بود که R حداقل نرخ بیت و C میانگین نرخ بیت یک منبع ترافیکی است. برای تخمین پارامتر c ، فاصله $[R, C]$ به $\frac{1}{c}$ بازه مساوی تقسیم می‌شود. در ابتدا همه عملها دارای احتمال مساوی $1/c$ هستند. اگر N منبع به سونیج متصل باشند و درخواست اتصال جدیدی وارد شود، در این صورت مقدار تخمین ظرفیت واقعی برابر $c_{N+1} = \rho + \frac{(i-1)(R-\rho)}{c}$ خواهد شد. اگر c_{N+1} صدق کند، درخواست جدید پذیرفته می‌شود، در غیراین صورت رد می‌شود.

انتخاب معیاری مناسب برای پاسخ محیط و نیز الگوریتم یادگیری نقش اساسی در دقت تخمین خواهند داشت. نحوه پاسخ محیط به پذیرش یا رد درخواست قبلی و رخ دادن تراکم در اثر عمل قبلی بستگی دارد و به صورت زیر تعریف می‌شود:

عمل α_i مطلوب است اگر: ۱) تخمین c_i منجر به قبول درخواست شود و تراکم در سیستم رخ ندهد.
۲) تخمین c_i منجر به رد درخواست شود و تراکم در سیستم رخ دهد.

عمل α_i نامطلوب است اگر: ۱) تخمین c_i منجر به قبول درخواست شود و تراکم در سیستم رخ ندهد.
۲) تخمین c_i منجر به رد درخواست شود و تراکم در سیستم رخ دهد.



شکل ۲- نحوه کنترل درخواست ورود با اتوماتان یادگیر

نکته دیگر آنکه در ابتدا درخواستها با استفاده از روش تخصیص حداقل نرخ بیت پذیرفته می‌شوند و هنگامی که دیگر درخواستها با این روش قابل پذیرش نباشند، از الگوریتم ارائه شده استفاده می‌شود. نرخ از دست رفتن سلولها به عنوان معیار اصلی برای پاسخگویی محیط برگزیده شده است. برای آنکه پیش از بروز تراکم از وقوع احتمالی آن آگاه شد، دو آستانه بالا و پایین برای طول صفت در نظر گرفته شده است. اگر طول صفت از آستانه بالا بیشتر شد، فرض می‌شود که تراکم رخ داده و تا هنگامی که طول صفت از آستانه پایین کمتر نشده، این فرض باقی خواهد ماند.

در این الگوریتم بر معیار حداقل نرخ اتلاف سلول تأکید شده است، زیرا با حفظ حداقل نرخ اتلاف سلول، می‌توان حداقل تأخیر سلولها را با میزان کردن ظرفیت بافر یا حداقل طول صفت اعمال نمود. اگر حداقل تأخیر سلولها T باشد، با فرض روش سرویس FCFS، ظرفیت بافر $K = TC/L$ سلول خواهد بود که C ظرفیت خط و L طول هر سلول است.

۴ شبیه‌سازی و بررسی نتایج

برای شبیه‌سازی از شبیه‌ساز ATM که توسط موسسه NIST¹ فراهم شده است استفاده شده است. پس از برطرف کردن خطاهای برنامه نویسی شبیه‌ساز، امکان کنترل درخواست ورود به آن اضافه شده است.

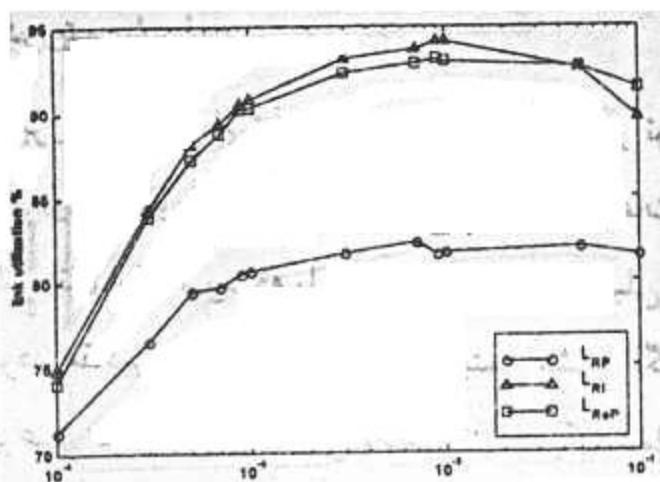
شبکه مورد استفاده شامل ۲۰۰ منبع ترافیکی اتفاقی از نوع ON-OFF است که همگی به یک سوئیچ ATM متصل هستند. هر منبع دارای بازه‌های ON و OFF است که در بازه ON با حداقل نرخ بیت $R=10 \text{ Mbps}$ سلول تولید می‌کند و در بازه OFF سلولی تولید نمی‌شود. طول بازه‌های ON و OFF دارای توزیع نمایی با میانگین T_{on} و T_{off} است. میانگین زمان انفجار و میانگین ناحیه غیرفعال به ترتیب برابر $T_{on}=0.4 \text{ ms}$ و $T_{off}=1.6 \text{ ms}$ در نظر گرفته شده‌اند. درخواستهای اتصال به صورت فرآیند پواسون با میانگین $\lambda=20 \text{ sec}^{-1}$ از یک منبع ترافیکی در زمان بیکاری وارد می‌شوند. زمان اتصال هم دارای توزیع نمایی با میانگین $h=100 \text{ ms}$ است. فرض می‌شود که حداقل نرخ از دست رفتن سلولها 10^4 است. همچنین آستانه بالای طول صفت، برابر ۱۰٪ و آستانه پایین صفت برابر ۵٪ طول صفت فرض شده‌اند. طول زمان اجرا برابر با 7000 ms است و هر یک از اتوماتانها دارای $L_{R,P}=100$ عمل هستند. به علت لگاریتمی بودن نمودارهای نرخ اتلاف سلول، عدد صفر با 10^{10} نشان داده شده است.

۴-۱- استفاده از الگوریتمهای یادگیری خطی استاندارد

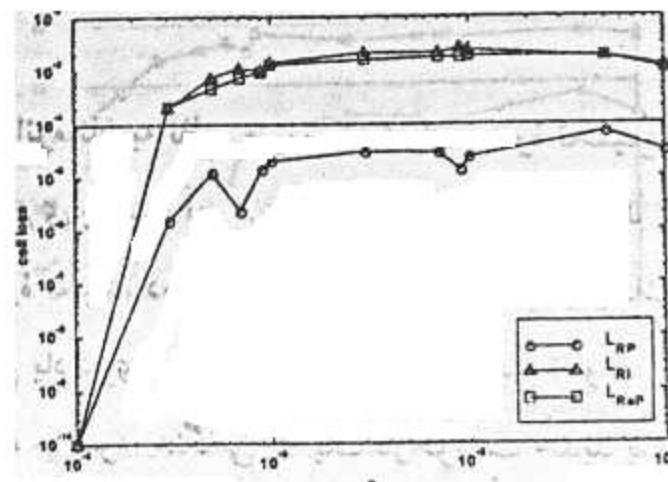
در این بخش با توجه به روش گفته شده از اتوماتانهای یادگیر $L_{R,I}$ و $L_{R,P}$ برای کنترل درخواست ورود استفاده می‌شود. الگوریتم یادگیری $L_{R,P}$: در الگوریتم یادگیری $L_{R,P}$ پارامترهای پاداش و جریمه برابر هستند. بررسیهای انجام شده نشان می‌دهد که الگوریتم $L_{R,P}$ در یک محیط ایستا، از هر نقطه اولیه، اتوماتان را به یک رفتار مصلحت‌گرا سوق می‌دهد [۹]. در آزمایش انجام شده محدوده مقادیر a و b ، 0.0001 تا 0.1 در نظر گرفته شد. نتایج شبیه‌سازی با الگوریتم $L_{R,P}$ نشان می‌دهد که این الگوریتم برای این حالت به خوبی عمل کرده است. شکل ۳-الف نرخ اتلاف سلول را به ازای مقادیر مختلف a نشان می‌دهد. در این نمودار

¹ National Institute of Standards and Technology

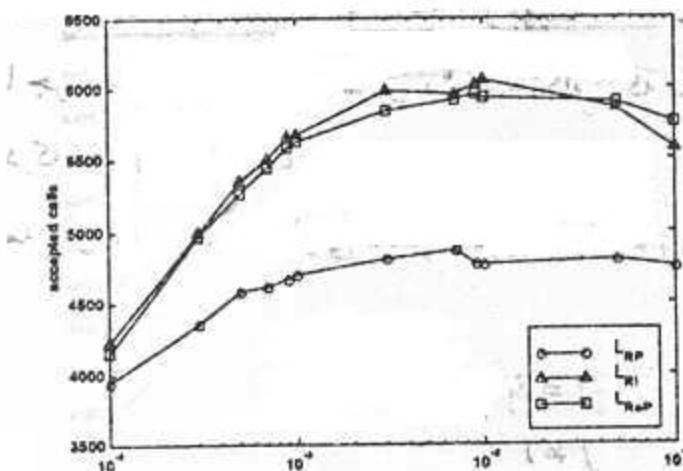
² Holding time



(ب) درصد بهره‌وری خط



(الف) نرخ اتلاف سلول



(ج) تعداد درخواستهای پذیرفته شده

شکل ۳- نتایج شبیه‌سازی کنترل درخواست ورود با استفاده از الگوریتمهای یادگیری خطی به ازای مقادیر مختلف پارامتر پاداش (a)

نرخ اتلاف سلول همواره کمتر از میزان تعیین شده $a = 10^{-3}$ است؛ نمودار نشان می‌دهد که با زیاد شدن مقادیر a و b به تدریج میزان اتلاف سلول افزایش می‌یابد. شکل ۳-ج تعداد درخواستهای پذیرفته شده را به ازای مقادیر مختلف a و b نشان می‌دهد. در این نمودار با افزایش مقادیر a و b تعداد درخواستهای پذیرفته شده افزایش یافته تا به ازای $a = b = 0.007$ تعداد اتصالها تقریباً ثابت می‌ماند.

الگوریتم یادگیری L_{R-I} : در این الگوریتم تنها پاداش داده می‌شود و پارامتر جریمه صفر است. تحقیقات انجام شده نشان می‌دهد که در یک محیط ایستا الگوریتم یادگیری L_{R-I} باعث نزدیک به بهینه شدن رفتار اتوماتان می‌شود [۹]. نتایج شبیه‌سازی با الگوریتم L_{R-I} نشان می‌دهد که این الگوریتم برای این حالت به خوبی عمل نکرده است. نرخ اتلاف سلول بیش از حد تعیین شده است. البته به ازای $a = 0.001$ نرخ اتلاف سلول صفر شده است ولی در عوض میزان بهره‌وری خط 75% شده است. در حالیکه با توجه به نتایج الگوریتمهای دیگر از جمله L_{R-P} ، با حداقل اتلاف سلول 0.001 بهره‌وری 81% قابل دسترسی است. از آنجا که محیط غیر ایستا بوده و احتمالهای جریمه هر از چندگاهی تغییر می‌کنند، الگوریتم L_{R-I} مناسب به نظر نمی‌رسد. زیرا الگوریتم L_{R-I} توزیع احتمالها را به یک عمل سوق می‌دهد. به عنوان مثال در ابتدای شبیه‌سازی عمل a_0 مطلوب است، ولی پس از بروز تراکم به یک عمل نامطلوب بدل می‌شود. اگر در این الگوریتم، اتوماتان به یک عمل همگرا شد و احتمال انتخاب آن عمل یک شد، دیگر تا به انتهای الگوریتم در آن عمل گیر کرده و عمل دیگری انتخاب نمی‌شود، هر چند آن عمل پس از مدتی نامطلوب شود.

الگوریتم یادگیری L_{R-EP} : در الگوریتم یادگیری L_{R-EP} پارامتر پاداش چندین برابر از پارامتر جریمه بزرگتر است. این الگوریتم یک الگوریتم نزدیک به بهینه است [۱۵]. در اینجا پارامتر جریمه برابر $b/a = 10$ در نظر گرفته شده است. در شکل ۳-الف نرخ اتلاف سلول بر حسب پارامتر پاداش در مقایسه با دیگر الگوریتمهای یادگیری نشان داده شده است. در این نمودار به ازای تمام مقادیر به جز $a = 0.001$ الگوریتم نتوانسته است حداقل نرخ اتلاف سلول را حفظ نماید. به ازای مقادیر کم a تعداد اتصالهای پذیرفته شده و در نتیجه نرخ اتلاف سلول، کمتر است، تا آنجا که به ازای $a = 0.001$ نرخ اتلاف سلول به صفر می‌رسد. در مقابل میزان بهره‌وری خط در این نقطه برابر به 74% می‌رسد. رفتار این الگوریتم در کنترل تراکم بسیار مشابه الگوریتم L_{R-I} است.

به طور کلی الگوریتمهای یادگیری L_{R-I} و L_{R-EP} به خوبی عمل نکرده‌اند؛ این به علت غیر ایستا بودن محیط و تغییر شرایط شبکه است. در حالی که الگوریتم یادگیری L_{R-P} توانسته است به ازای تمامی مقادیر پاداش کیفیت سرویس را حفظ نماید.

با توجه به بررسیهای انجام شده در مورد تخمین پارامتر توسط اتوماتان به نظر می‌رسد بتوان نحوه تخمین اتوماتان را بهبود بخشد. به ویژه در مواردی که می‌توان از پاسخ محیط اطلاعات بیشتری بدست آورد. به همین منظور سعی شد الگوریتم یادگیری دیگری برای اتوماتان ارائه شود که با استفاده از اطلاعات قبلی درباره محیط، در هر بار تعداد بیشتری عمل را پاداش یا جریمه دهد. برای بهنگام کردن مقادیر بردار احتمال از الگوریتم مشابه الگوریتم یادگیری L.R.P استفاده می‌شود. با این تفاوت که در این الگوریتم جریمه به چند عمل تعلق می‌گیرد، ولی پاداش تنها به یک عمل داده می‌شود. هنگامی که یک درخواست جدید پذیرفته می‌شود ولی این درخواست منجر به بروز تراکم می‌شود، مشخص می‌شود که مقدار تخمین c_i کمتر از حد واقعی بوده که $C < c_i$ شده است. از آنجا که $c_1 < c_2 < \dots < c_r$ ، خواهیم داشت: $c^* < c_1 < c_2 < \dots < c_r$ که c^* مقدار بهینه است. بنابراین هیچ یک از مقادیر c_1, c_2, \dots, c_r هم مطلوب نخواهد بود و باید به همه آنها جریمه تعلق بگیرد. به طریق مشابه هنگامی که یک درخواست جدید رد می‌شود ولی تراکمی در شبکه رخ نمی‌دهد، مشخص می‌شود که مقدار تخمین c_i بیش از حد واقعی بوده و باید جریمه شود. مقادیر $c_{i+1}, c_{i+2}, \dots, c_r$ هم که بزرگتر از c_i هستند، مستوجب جریمه نخواهند بود. به عبارتی دیگر الگوریتم یادگیری به صورت زیر خواهد بود:

اگر c_i مطلوب باشد:

$$p_j(k+1) = \begin{cases} p_j(k) + a[1 - p_i(k)] & j = i \\ (1-a)p_j(k) & j \neq i \end{cases} \quad (3)$$

اگر c_i نامطلوب باشد و درخواست قبلی پذیرفته شده باشد:

$$p_j(k+1) = \begin{cases} (1-b)p_j(k) & j \leq i \\ \frac{b}{r-i} + (1-b)p_j(k) & j > i \end{cases} \quad (4)$$

اگر c_i نامطلوب باشد و درخواست قبلی رد شده باشد:

$$p_j(k+1) = \begin{cases} (1-b)p_j(k) & j \geq i \\ \frac{b}{i-1} + (1-b)p_j(k) & j < i \end{cases} \quad (5)$$

که a و b پارامترهای پاداش و جریمه هستند.

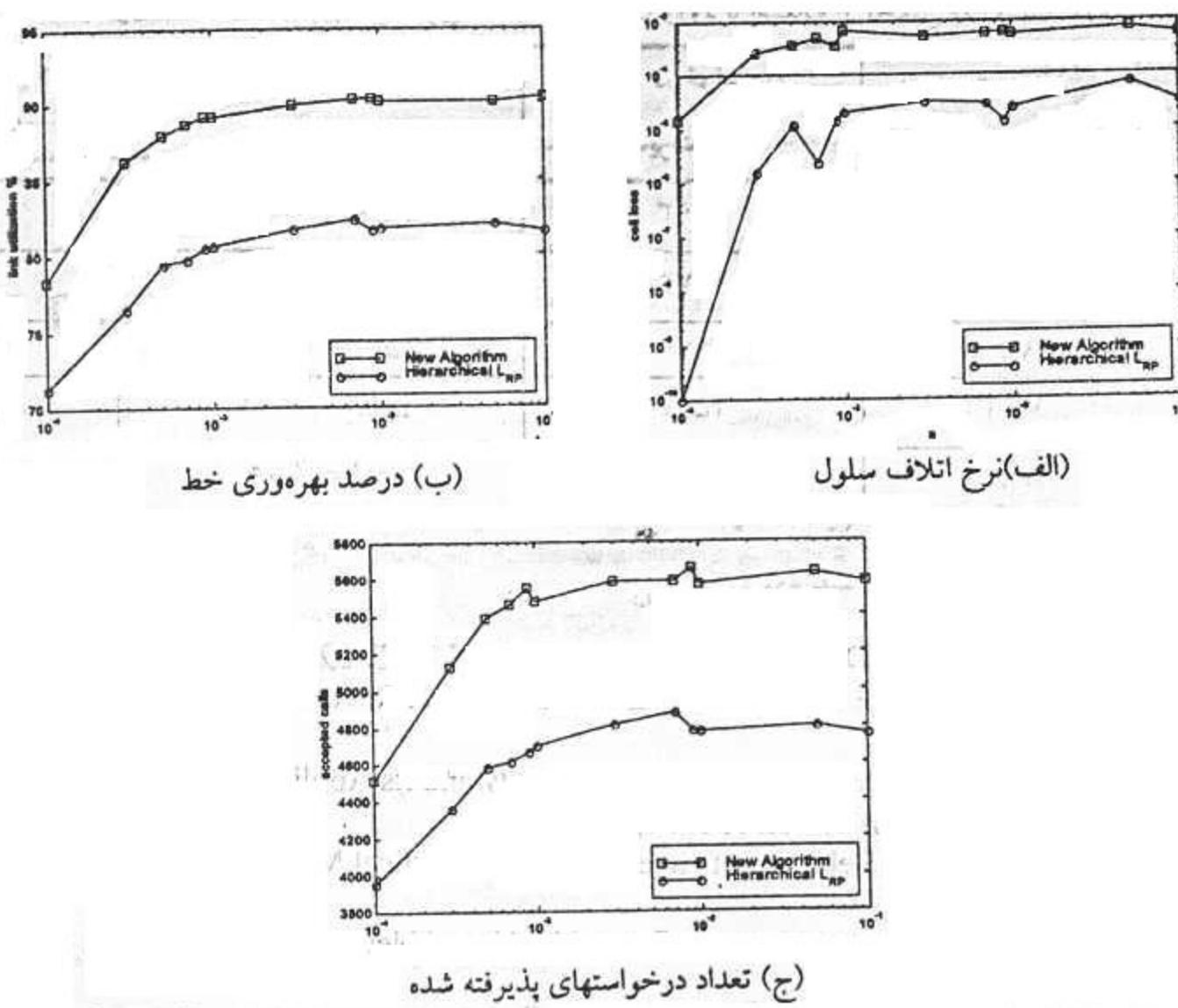
اگر انتخاب اتوماتان منجر به قبول درخواست می‌گردد، در صورت وجود تراکم در شبکه، انتخاب به عنوان یک عمل نامطلوب به اتوماتان پس داده می‌شود. یعنی در واقع پذیرش درخواست به سوئیچ ابلاغ نمی‌شود. این مرحله تا انتخاب عملی که منجر به رد درخواست شود، تکرار می‌شود. این تغییر باعث تصحیح تصمیمهای اتوماتان می‌شود، بدون آنکه این تصمیمهای اشتباہ در محیط اعمال گردند.

شکل ۴ نتایج شبیه‌سازی کنترل درخواست ورود با الگوریتم جدید را نشان می‌دهد. نمودار ۵-الف نرخ اتلاف سلوول را برای مقادیر مختلف a نشان می‌دهد. به ازای $a=0.0001$ کمترین نرخ اتلاف سلوول برابر با 4.1×10^{-5} است. در این نقطه تعداد ۴۵۱۷ درخواست پذیرفته شده است که میزان بهره‌وری خط به ۷۸۳۲٪ رسیده است. با افزایش پارامتر a تعداد درخواستهای پذیرفته شده افزایش یافته و در نتیجه نرخ اتلاف سلوول و میزان بهره‌وری خط هم افزایش می‌یابد. یعنی در دیگر نقاط نرخ اتلاف سلوول از آستانه تعیین شده فراتر رفته است.

این الگوریتم درواقع حالت خاصی از اتوماتانهای یادگیر سلسله مراتبی است. در بخش بعد به بررسی کنترل درخواست ورود با استفاده از اتوماتان یادگیر سلسله مراتبی می‌پردازم.

۴-۳ اتوماتان یادگیر سلسله مراتبی

یکی از پارامترهای مهم در کارایی تخمین اتوماتان، انتخاب تعداد مقادیر ممکن برای پارامتر و نیز محدودهای که مقادیر در نظر گرفته شده در آن قرار دارند، است. برای بالا بردن دقت تخمین پارامتر نیاز به افزایش تعداد عملهای اتوماتان است که این کار می‌تواند باعث کاهش چشمگیر در همگرایی اتوماتان شود. یکی از روشایی که برای حل مشکل فوق پیشنهاد شده است، استفاده از ساختارهای سلسله مراتبی یا ساختار درختی است [۹]. در ریشه این درخت یک اتوماتان وجود دارد که هر عمل آن متناظر با یک زیر



شکل ۴- نتایج شبیه‌سازی با اتماتان ارائه شده و سلسه مراتبی بر حسب پارامتر پاداش ($a=b$).

درخت است. برگهای درخت متناظر با عملهای نهایی انتخاب شده توسط اتماتان هستند که به محیط داده می‌شود. برای انتخاب یک عمل نخست در ریشه یک عمل با توجه به بردار احتمال عملها انتخاب می‌شود. سپس در زیر درخت متناظر با این عمل این کار تکرار می‌شود. این کار تکرار شده تا یکی از برگهای درخت انتخاب می‌شود. پاداش یا جریمه دادن هم از ریشه به سمت پایین در هر سطح با توجه به عمل انتخاب شده در آن سطح انجام می‌شود.

برای تخمین پهنای باند معادل می‌توان از یک اتماتان یادگیر سلسله‌مراتبی استفاده کرد. در اینجا از یک درخت دو سطحی که هر گره آن یک اتماتان L_{RP} با 10^0 عمل است، استفاده شده است. نسبت جریمه یا پاداش دادن به گره پدر و فرزند یکسان است. برای شبیه‌سازی محدوده 0.0001 تا 10^0 برای a و نیز b انتخاب شد. شکل ۴ نتایج شبیه‌سازی بر حسب a را نشان می‌دهد. این روش به خوبی توانسته است حتی به ازای تمام مقادیر a نرخ اتلاف سلوک را کمتر از حد تعیین شده نگاه دارد.

۵ خلاصه و نتیجه‌گیری

در این مقاله کنترل درخواست ورود با استفاده از اتماتانهای یادگیر با ساختار متغیر بررسی شد. در این روش از اتماتانهای یادگیر برای تخمین ظرفیت واقعی مورد نیاز برای منابع مشابه متصل به سویچ ATM استفاده شد. با استفاده از این تخمین برای پذیرش یا رد درخواستها تصمیم گیری می‌شود. اتماتان یادگیر با استفاده از پاسخهای محیط به انتخابهای گذشته مبنی بر مطلوب یا نامطلوب بودن آنها، تصمیم گیری‌های بعدی را تصحیح می‌کند. محیط با توجه به قبول یا رد درخواست قبلی و با توجه به آخرین نمونه‌برداری از نرخ از دست رفتن سلوک و طول صفحه اتماتان پاسخ می‌دهد. شبیه‌سازی‌ها نشان دادند که الگوریتم‌های با رفتار نزدیک به بهینه مانند L_{RP} و L_{RP-I} برای کنترل درخواست ورود مناسب نیستند، زیرا محیط غیر ایستا بوده و عمل بهینه تغییر می‌کند. طبیعی است که در سویچ‌های سریع، عملیات CAC باید سریع انجام گردد و واژ این روند حتی الامکان ساختار CAC باید ساده باشد. CAC در حقیقت یک عملیات نرم افزاری است که در داخل سویچ ATM قرار گرفته و موظف به تصمیم گیری در مورد پذیرش یا رد درخواست ارتباط ورودی می‌باشد. الگوریتم‌های مبنی بر اتماتان یادگیر می‌توانند تصمیم گیری را سریع انجام دهند. در جدول ۱ بهترین نتیجه هر الگوریتم مشاهده می‌شود. بهترین نتیجه هر الگوریتم آن است که با حفظ حداقل نرخ اتلاف سلوک (یا حداقل نرخ اتلاف سلوک) بیشترین نرخ بهره‌وری حاصل شده باشد اتماتان یادگیر سلسله‌مراتبی و اتماتان L_{RP} بهترین نتیجه را در میان اتماتانهای بررسی شده داشته‌اند.

جدول ۱ - خلاصه نتایج شبیه‌سازی کنترل درخواست ورود با استفاده از تخمین پهنه‌ای باند معادل.

نوع اتوماتان	پارامتر	تعداد درخواستهای پذیرفته شده	بهره‌وری خط (%)	اتلاف سلول
L_{R-I}	$a=0.10003$	۴۹۹۳	۸۴/۲۴	2×10^{-4}
L_{R-eP}	$a=0.10003, b=a/10$	۴۹۷۳	۸۳/۹	2×10^{-4}
L_{R-P}	$a=b=0.1009$	۴۷۷۸	۸۱/۶۲	$1/3 \times 10^{-5}$
الگوریتم ارائه شده	$a=b=0.1001$	۴۵۱۷	۷۸/۳۲	$1/4 \times 10^{-5}$
سلسله‌مراتبی با	$a=b=0.1009$	۴۸۱۳	۸۱/۸۲	$1/8 \times 10^{-5}$

مراجع

- [1]. ITU-T, "Broadband Aspects of ISDN ITU-T Recommendation I.121", 1991.
- [2]. H. G. Perros, K. M. Elsayed, "Call Admission Control Schemes: A Review", IEEE Commun. Magazine, Nov. 1996, pp. 82-91.
- [3]. R. Guerin, H. Ahmadi, M. Naghshineh, "Equivalent Capacity and Its Application to Bandwidth Allocation in High-Speed Networks", IEEE JSAC, Vol. 9, 1991, pp. 968-81.
- [4]. K. Sohraby, "On the Asymptotic Behavior of Heterogeneous Statistical Multiplexer With Applications", Proc. INFOCOM'92, pp. 839-47.
- [5]. A. Hiramatsu, "ATM Communication Network Control by Neural Networks", IEEE Trans. on Neural Networks, Vol. 1, No. 1, March 1990.
- [6]. Chang-Ju Chang, Ray-Guang Cheng, "Traffic Control in an ATM Network Using Fuzzy Set Theory", Proceedings IEEE INFOCOM, Vol. 3, 1994.
- [7]. A. V. Vasilakos, N. H. Loukas and A. F. Atlasis, "The Use of Learning Algorithms in ATM Networks Call Admission Control Problem: A Methodology", Proc. IEEE Local Computer Networks, 1995, pp. 407-412.
- [8]. A. T. Burrel, P. Papantoni-Kazakos, "Real Time Learning and Traffic Monitoring for the Traffic Management of ATM Networks", International Conference on Telecommunications, Jun. 1998, Vol. 2, pp. 21-25.
- [9]. P. Mars, J. R. Chen, R. Nambiar, "Learning Algorithms, Theory and Applications in Signal Processing, Control and Communications", CRC Press, 1996.
- [10]. K. S. Narendra and M. A. L. Thathachar, "Learning Automata: An Introduction", Prentice-Hall, Englewood cliffs, 1989.
- [11]. A. V. Vasilakos, S. A. Koubias, "On Routing and Performance Comparison of Techniques for Packet-Switched Networks Using Learning Automata", IEEE Int. Symposium on Circuits and Systems 1988, pp. 109-113.
- [12]. P. Mars, K. S. Narendra, "Routing, Flow Control and Learning Algorithms", In Proc. 1st IEE Int. Conf. on Communication Networks, 1987.
- [13]. A. V. Vasilakos and A. F. Atlasis, "LB-SELA: Rate based Access Control for ATM Networks", In Proc. IEEE INFOCOM'94, 1994, pp. 1552-1559.
- [14]. M. R. Meybodi and S. Lakshminarayanan, "A Learning Approach to Assignment in a Two class M/M/1 Queuing System with Unknown Parameters", Proc. of Third Yale Workshop on Applications of Adaptive Systems Theory, Yale University, pp. 106-109, 1983.
- [15]. M. R. Meybodi and H. Beigy, "Neural Network Engineering Using Learning Automata: Determination of Desired Size for Three Layer Feed-forward Neural Networks", Proc. of 5th Annual Int. Computer Society of Iran Computer Conf. CSICC'2000, Tehran, Iran, (to appear).
- [16]. P. Bahr and M. R. Meybodi, "A Method for Adaptation of Vigilance Factor and Choice Parameter in Fuzzy-ART Systems", Proc. of 7th Iranian Conf. on Electrical Engineering, ICEE'99, Tehran, Iran, Vol. 3, pp. 12-27.
- [17]. H. Beigy and M. R. Meybodi, "Graph Isomorphism Using Learning Automata", Proc. of 5th Annual Int. Computer Society of Iran Computer Conf. CSICC'2000, Tehran, Iran, (to appear).
- [18]. K. S. Narendra and M. A. L. Thathachar, "Learning Automata: An Introduction", Prentice-Hall, Englewood cliffs, 1989.
- [19]. M. R. Meybodi and S. Lakshminarayanan, "Optimality of a General Class of Learning Algorithms", Information Science, Vol. 28, pp. 1-20, 1982.
- [20]. M. R. Meybodi, "Results on Strongly Absolutely Expedient Learning Automata", Proc. of OU Inference Conf. 86, ed. D. R. Moots and R. Butrick, Athens, Ohio: Ohio Univ. Press, pp. 197-204, 1984.
- [21]. S. Lakshminarayanan, "Learning Algorithms: Theory and Applications", New York, Springer-Verlag, 1991.