

تنظیم پارامتر چکه در ساماندهنده ترافیک ATM توسط اتوماتای یادگیر

یونس سیفی محمدرضا میبدی

آزمایشگاه سیستم های نرم افزاری

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی امیرکبیر

چکیده: این مقاله یک مدل سطل چکه ای با تنظیم پارامتر چکه متغیر را ارائه نموده و اقدام به بررسی کارایی آن در هنگام اجرای UPC در شبکه ATM می نماید. از آنجائیکه قرارداد اولیه صورت پذیرفته در CAC برای کاربردهای مهم دقیق نمی باشد و از سوی دیگر در انواع گوناگونی از منابع مانند منابع انفجاری تخمین مشخصات ترافیک مشکل و دارای پیچیدگی بسیار می باشد آنگونه که باید روشهای ساماندهی شناخته شده قادر نیستند بصورتی کارآمد پارامترهای توافق شده را کنترل نمایند. لذا سعی شده است از تنظیم پویای این پارامترها استفاده شود. نتایج شبیه سازی های صورت گرفته نشان می دهند که روش جدید از جنبه های گوناگونی مانند درصد سلولهای دور ریخته شده و بهره وری استفاده از منابع بهتر از نوع قدیمی سطل چکه ای عمل می نماید.

کلمات کلیدی: انتقال حالت ناهمگام، اتوماتای یادگیر، سطل چکه ای، ساماندهی ترافیک، کنترل پارامتر کاربری

(۱) مقدمه

ATM یکی از راهکارهای عمده در شبکه های ارتباطی پیشرفته است و توسط CCITT بعنوان بهترین راه حل پیاده سازی سرویس های B-ISDN پیشنهاد شده است [۱]. یکی از عمده ترین قابلیت های این شبکه پشتیبانی انواع سرویسها با نرخهای گوناگون، مشخصات و کیفیت های سرویس^۱ متنوع می باشد. ATM با استفاده از تکنولوژی مولتی پلکس سازی آماری میزان بهره وری از منابع شبکه را بهبود می بخشد. با این وجود اگر شبکه دارای مکانیزم ساماندهی مناسب نباشد، کیفیت سرویس به دلیل تداخلهای تصادفی پیش آمده در ترافیک و همچنین بروز مشکل در گره های سوئیچ شبکه تنزل خواهد یافت. از آنجائیکه کنترل ترافیک در تکنولوژی ATM مبحثی مهم می باشد، CCITT در توصیه های I.371 پیشنهادات و قراردادهای لازم را عرضه نموده است. بدلیل اینکه در شبکه های ATM ساختار کانال مشخصی در نظر گرفته نشده است، ترافیک کاربر می تواند براحتی منابع شبکه را اشغال نماید و منجر به بروز تراکم گردد. مکانیزمهای کنترل درخواست ورود^۲ می توانند تعیین نمایند که آیا پذیرفته شدن یک اتصال با کیفیت سرویس معین می تواند منجر به بروز تداخل در سایر اتصالات گردد یا خیر. حال پس از پذیرفته شدن یک اتصال در صورت بروز هر گونه تخلف از پارامترهای اولیه مکانیزم کنترل پارامتر کاربری^۳ اقدام به انجام عمل متناسب با تخلف صورت گرفته می نماید. در مجموع هر اندازه که برقراری قرارداد با جزئیات بیشتری صورت پذیرد، رفتار سرویس را به شکل کارآمدتری می توان ساماندهی نمود. ولی از سوی دیگر هنگامیکه قرارداد بسیار پیچیده می شود اعمال UPC بصورتی کارآمد مشکل می گردد و مهمتر آنکه بدست آوردن چنین مشخصاتی برای سرویس های خاص، مانند ویدئو بسیار مشکل می شود. الگوریتم سطل چکه ای^۴ و انواع مختلف الگوریتم های پنجره^۵ بعنوان دو روش کارآمد برای اعمال UPC در شبکه ATM در نظر گرفته شده اند. هر چند که الگوریتم های پنجره ای در شبکه های بسته ای عملکرد بسیار خوبی دارند، اما آنها مناسب برای شبکه های با سرعت بالای ATM نیستند. روشهای UPC بسیاری وجود دارند که کنترل نرخ میانگین و حداکثر را انجام

۱ - Quality Of Service (QOS)

۲ - Call Admission Control (CAC)

۳ - Usage Parameter Control (UPC)

۴ - Leaky Bucket (LB)

۵ - Windowing Algorithms

می‌دهند. در [۲] چندین مکانیزم ساماندهی پیشنهاد شده که سطل چکه‌ای بعنوان موثرترین آنها معرفی گردیده است. این الگوریتم می‌تواند با تغییر، ساماندهی و شکل‌دهی نرخ ترافیک ورودی به شبکه اقدام به حفاظت از کیفیت سرویس موجود نماید. پیاده‌سازی سخت‌افزاری این مکانیزم بسیار سریع و مقرون به صرفه می‌باشد.

اما از سوی دیگر هنگامیکه سطل چکه‌ای برای ساماندهی منابع انفجاری مورد استفاده قرار می‌گیرد تخمین مشخصات یک منبع انفجاری مانند نرخ میانگین، بعثت طبیعت آماری انفجار منجر به بروز خطاهایی می‌شود که با کاهش دوره‌های تخمین افزایش می‌یابند. درضمن افزایش فواصل تخمین منجر به بروز عدم پویایی مورد نیاز شبکه ATM می‌شود [۳].

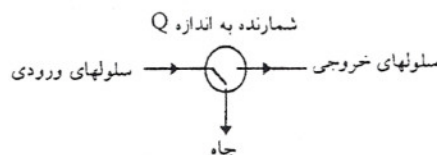
از زمان ارائه الگوریتم سطل چکه‌ای تابحال، روشهای بهبودیافته بسیاری ارائه شده است. در [۴] کارایی سطل چکه‌ای در شبکه ATM مورد مطالعه و بررسی قرار گرفته است و در [۲] نیز عملکرد کنترل جریان چهار نوع الگوریتم پنجره را در شبکه ATM مورد مطالعه و بررسی قرار داده است. در [۵] براساس سطل چکه‌ای قدیمی مکانیزم جدیدی ارائه شده است که دارای قابلیت تطبیق بیشتری نسبت به مشخصات ترافیک می‌باشد. در حال حاضر نیاز مبرمی به یک مکانیزم کنترل پویای UPC در شبکه‌های سرعت بالا که در ضمن تضمین کیفیت سرویس، استفاده بهینه را از منابع شبکه بعمل آورد [۶] احساس می‌شود.

در این راستا مکانیزمهای گوناگونی ارائه شده که عمده آنها سعی دارند به گونه‌ای تشخیص دهند رفتار منبع چگونه است و براساس این تشخیص اگر منبع بصورت غیرعادی رفتار نمود سلولهای آنرا علامتگذاری نموده و یا دور بریزند. ساماندهنده پیشنهادی در [۷] از این نوع می‌باشد. البته در بعضی روشهای پیشنهاد شده مانند [۶] نیز سعی گردیده به اشکال گوناگونی مانند تنظیم نرخ تولید توکن عملکرد ساماندهنده بهبود یابد. در مکانیزم ساماندهی پیشنهادی در این مقاله سعی می‌گردد پارامتر چکه گونه‌ای تنظیم شود که میزان بهره‌وری از منابع شبکه افزایش یابد. در این روش همواره از بین چندین پارامتر چکه از پیش تعیین شده یکی بعنوان نرخ چکه جاری در نظر گرفته می‌شود و انتخاب آن نیز بگونه‌ای است که تعداد سلولهای دور ریخته شده توسط ساماندهنده کاهش یابد.

این مقاله به صورت زیر سازماندهی شده است: در بخشهای ۲، ۳ و ۴ توضیحاتی در مورد سطل چکه‌ای ساده، مدل ترافیک انفجاری و اتوماتای یادگیر ارائه می‌گردد. سپس در بخشهای ۵ و ۶ ساماندهنده‌های جدیدی با استفاده از اتوماتای یادگیر و منطق فازی پیشنهاد می‌گردد و در نهایت در بخشهای ۷ و ۸ نتایج شبیه‌سازی و مقایسات صورت پذیرفته عرضه می‌شود.

۲) مکانیزم سطل چکه‌ای

ساده‌ترین نوع مکانیزم سطل چکه‌ای یک شمارنده می‌باشد که بهنگام تولید هر سلول یک واحد افزایش می‌یابد و دائماً با نرخ ثابت α تا زمانیکه شمارنده مثبت است، کاهش می‌یابد (شکل ۱). زمانی که شمارنده به یک مقدار آستانه Q برسد سلول ورودی دور ریخته می‌شود (یا آنکه بعنوان سلول اضافه مطابق با [۸] علامت زده می‌شود).



شکل (۱) مکانیزم سطل چکه‌ای

نرخ چکه r معمولاً بصورت $r = c \cdot m_0$ با نرخ میانگین مرتبط می‌باشد که در این رابطه c ثابتی بزرگتر یا مساوی یک است و m_0 نیز نرخ میانگین مورد توافق می‌باشد، اگر انحراف بصورت $d = m/m_0$ تعریف شود (m نرخ میانگین واقعی منبع می‌باشد) در آنصورت اثبات گردیده است بازای هر مقدار d که از c کوچکتر باشد، سطل چکه‌ای قادر به عمل نمی‌باشد [۹]. مقدار c (و در نتیجه مقدار r) دارای اهمیت بسیاری است زیرا توسط آن شدت کنترل مشخص می‌گردد و اهمیت این موضوع بدین دلیل است که هر اندازه کنترل اعمال شده توسط رویه ساماندهی قویتر باشد پهنای باند بیشتری حفظ می‌شود [۱۰].

۳) مدل ترافیک یک منبع انفجاری

یک منبع وقتی انفجاری نامیده می‌شود که دارای حالات فعال و بیکار باشد. منابع صوتی و انتقال داده با سرعت بالا نمونه‌هایی از منابع انفجاری هستند. جهت مدل نمودن این منابع می‌توان از زنجیره مارکوف دو حالت استفاده نمود. حالات ON و OFF دوره‌های فعال و بیکار را نشان می‌دهند. مشخصات منبع را می‌توان با مجموعه پارامترهای زیر بیان نمود:

m : نرخ بیت میانگین	p : نرخ بیت حداکثر
b : میزان انفجاری بودن منبع	L : طول متوسط انفجار (برحسب سلول)
T_{OFF} : میانگین مدت زمان بیکاری	T_{ON} : میانگین مدت زمان انفجار

در اینجا فرض شده است که دوره‌های فعالیت و بیکاری هر دو دارای توزیع نمایی با میانگین‌های T_{ON} و T_{OFF} هستند. بر طبق [۹]:

$$b = p/m, \quad T_{ON} = (L \cdot n_{cell})/p, \quad T_{OFF} = T_{ON}(b-1)$$

در اینجا n_{cell} نشاندهنده طول سلول برحسب بایت می باشد، که برطبق استاندارد ATM برابر ۵۳ می باشد.

۴) مروری بر اتوماتای یادگیر

اتوماتای یادگیر یک ماشین حالت متناهی است که با یک محیط تصادفی دائما در حال محاوره می باشد و از طریق یک فرآیند یادگیری سعی در آموختن عمل بهینه مورد نیاز محیط دارد. در هر حالتی اتوماتا برطبق بردار احتمالات یک عمل را انتخاب می نماید و سپس از یک تابع خروجی استفاده می نماید تا تعیین نماید پاسخ محیط به این عمل چیست (جریمه یا پاداش). اتوماتا با در نظر گرفتن این پاسخ و بوسیله یک تابع انتقال، حالت خود را به حالتی جدید که لزوما با حالت قبل متفاوت نیست تغییر می دهد. اتوماتا حتی در زمانی که اطلاعات کمی از سیستم در دسترس باشد، می تواند برای پیشگویی رفتار آن سیستم عمل نماید [۱۱].

اتوماتاهای یادگیر را می توان به لحاظ ساختاری به دو دسته عمده ساختار ثابت و ساختار متغیر تقسیم نمود. در نوع اول هنگام استفاده از اتوماتا بردار احتمالات تغییر نمی نماید و فقط براساس یک ترتیب از پیش تعریف شده انتقال بین اعمال صورت می گیرد. ولی در نوع دوم بردار احتمالات دائما در حال تغییر می باشد و برحسب آنکه یک عمل از محیط جریمه یا پاداش دریافت دارد احتمال مرتبط با آن کاهش و یا افزایش می یابد.

هر الگوریتم یادگیری را می توان بصورت زیر بیان نمود:

$$P(n+1) = T[P(n), \alpha(n), \beta(n)]$$

$\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه خروجی های اتوماتا

$\beta = \{\beta_1, \beta_2, \dots, \beta_r\}$ مجموعه ورودی های اتوماتا

$P = \{p_1, p_2, \dots, p_r\}$ بردار احتمالات

اگر تابع T خطی باشد الگوریتم یادگیری خطی نامیده می شود و در غیر این صورت غیر خطی خوانده می شود. ایده اصلی تمامی الگوریتمهای یادگیر بدین صورت است که اگر الگوریتم در تکرار n -ام عمل α_i را انتخاب نماید و پاسخ مطلوبی برای آن از محیط دریافت نماید در آن صورت احتمال $p_i(n)$ مربوط به عمل α_i افزایش می یابد و احتمالات مربوط به سایر اعمال کاهش می یابد. در صورت دریافت جواب غیرمطلوب، $p_i(n)$ کاهش می یابد و احتمالات مربوط به سایر اعمال افزایش می یابد. بنابراین:

• هنگام دریافت پاسخ مطلوب

$$p_j(n+1) = p_j(n) - f_j[p_j(n)]; \quad \forall j; \quad j \neq i$$

$$p_i(n+1) = p_i(n) + \sum_{j=1, j \neq i}^r f_j[p_j(n)]$$

• هنگام دریافت پاسخ نامطلوب

$$p_j(n+1) = p_j(n) - g_j[p_j(n)]; \quad \forall j; \quad j \neq i$$

$$p_i(n+1) = p_i(n) + \sum_{j=1, j \neq i}^r g_j[p_j(n)]$$

توابع f_j و g_j توابعی غیرمنفی می باشند و از آنها با عنوان تابع پاداش و جریمه نام برده می شود. هم اکنون مطالعات بسیاری بر روی الگوریتمهای یادگیر خطی به علت قابلیت انعطاف بالای آنها انجام می شود. برای این نوع از الگوریتمها توابع پاداش و جریمه بصورت زیر تعریف می شوند:

$$f_j[p_j(n)] = ap_j(n); \quad 0 < a < 1$$

$$g_j[p_j(n)] = \frac{b}{r-1} - bp_j(n); \quad 0 \leq b < 1$$

در روابط فوق r تعداد اعمال اتوماتا، a پارامتر پاداش و b پارامتر جریمه می باشند. براساس مقادیر مختلفی که برای پارامترهای یادگیری a و b در نظر گرفته می شود الگوریتمهای مختلفی بدست می آید که در زیر مواردی از آنها نام برده شده است الگوریتم یادگیری L_{np} نامیده می شود اگر $a=b$ باشد و L_{ri} خوانده می شود اگر $b=0$ باشد.

هر دو نوع اتوماتای فوق از نوع اتوماتاهای با ساختار متغیر هستند. اتوماتاهایی که در ادامه به تشریح آنها خواهیم پرداخت از نوع ساختار ثابت می باشند.

اتوماتای $L_{SN,5}$

اتوماتای $L_{SN,5}$ دارای ۵N حالت و ۵ عمل می باشد. اتوماتای $L_{SN,5}$ اقدام به نگهداری تعداد موفقیتها و شکستهای دریافت شده بازای هر عمل می نماید و فقط زمانی که تعداد شکستها بیش از تعداد موفقیتها گردد (یا از یک مقدار حداکثر N تجاوز نماید) اقدام به تغییر عمل می نماید. اینکار با توسعه دادن فضای حالت به ۵N و تعیین قواعد انتقال از یک حالت به حالات دیگر صورت می پذیرد.

N حافظه مرتبط با هر عمل گفته می‌شود و در مجموع اتوماتا دارای حافظه‌ای برابر $5N$ می‌باشد. هنگام دریافت هر پاسخ مطلوب، حالت اتوماتا به صورت عمیق‌تری در حافظه آن قرار می‌گیرد و اگر پاسخ نامطلوبی دریافت گردد آن حالت بیشتر از حافظه اتوماتا خارج می‌شود.

اتوماتای $L_{5N,5}$ دارای $5N$ حالت $\varphi_1, \varphi_2, \dots, \varphi_{5N}$ و پنج عمل $\alpha_1, \dots, \alpha_5$ می‌باشد. حالت‌های $\varphi_{(i-1)N+K}$ ($K=1 \dots N$) مربوط به عمل α_i ($i=1 \dots 5$) می‌باشند و هنگام قرار گرفتن در هر یکی از این حالات عمل α_i اجرا می‌گردد. انتقال حالت بهنگام دریافت پاسخ مطلوب بصورت زیر می‌باشد:

$$\varphi_i \rightarrow \varphi_{i+1} \text{ if } i \neq (K-1)N+1, K=1 \dots 4 \quad \varphi_i \rightarrow \varphi_i \text{ if } i=(K-1)N+1, K=1 \dots 4$$

از آنجائیکه این اتوماتا دارای پنج عمل می‌باشد هنگام تعویض اعمال نیاز به وجود یک راهکار از پیش تعریف شده است تا عمل بعدی بدون بروز هیچگونه ابهامی انتخاب گردد. در اینجا اعمال بصورت ترتیبی عوض می‌شوند. بدین صورت که پس از عمل اول عمل دوم انتخاب می‌گردد و در نهایت پس از آخرین عمل مجدداً عمل اول اجرا می‌گردد. با توجه به این توضیحات، بهنگام دریافت پاسخ نامطلوب نیز عملکرد اتوماتا بصورت زیر است:

$$\varphi_i \rightarrow \varphi_{i+1} \text{ if } i \neq KN, K=0 \dots 5 \quad \varphi_i \rightarrow \varphi_{(i+1)N} \text{ if } i=KN, K=0 \dots 4$$

$$\varphi_{5N} \rightarrow \varphi_N$$

$$\alpha_i \rightarrow \alpha_{(i \bmod 5)+1}$$

اتوماتای $G_{5N,5}$

این نوع از اتوماتا دارای ۵ عمل می‌باشد و با آموزش رفتار گذشته سیستم سعی در تغییر اعمال می‌نماید. در این اتوماتا هنگام دریافت پاسخ مطلوب، انتقال حالات بصورت زیر می‌باشد:

$$\varphi_i \rightarrow \varphi_{i+1} \text{ if } i \neq KN+1, K=0 \dots 4$$

$$\varphi_i \rightarrow \varphi_i \text{ if } i=KN+1, K=0 \dots 4$$

برخلاف اتوماتای $L_{5N,5}$ اتوماتای $G_{5N,5}$ پیش از آنکه از عملی به عمل دیگر انتقال یابد عمل جاری را حداقل N بار انجام می‌دهد و در نهایت هنگام دریافت پاسخ نامطلوب از محیط، انتقال حالات بشکل زیر صورت می‌پذیرد:

$$\varphi_i \rightarrow \varphi_{i+1} \text{ if } i \neq 5N$$

$$\varphi_{5N} \rightarrow \varphi_1$$

در این نوع از اتوماتا نیز مانند اتوماتای $L_{5N,5}$ تعویض اعمال بشکل ترتیبی صورت می‌گیرد. و پس از تعویض هر عمل، عمل بعدی انتخاب می‌گردد.

اتوماتای کرینسکی

این نوع از اتوماتا نیز همانند موارد پیشین دارای پنج عمل می‌باشد. عملکرد این اتوماتا هنگام دریافت پاسخ مطلوب، بصورت زیر است:

$$\varphi_i \rightarrow \varphi_{i \bmod N+1} \text{ if } i \neq KN+1$$

$$\varphi_i \rightarrow \varphi_{(i \bmod N-1)} \text{ if } i=KN$$

$$\varphi_i \rightarrow \varphi_i \text{ if } i=KN+1, K=0 \dots 4$$

عملکرد این نوع از اتوماتا بهنگام دریافت پاسخ نامطلوب از محیط مانند اتوماتای $L_{5N,5}$ می‌باشد. و می‌توان آنرا بصورت زیر تشریح نمود:

$$\varphi_i \rightarrow \varphi_{i+1} \text{ if } i \neq KN, K=1 \dots 4$$

$$\varphi_i \rightarrow \varphi_{(i+1)N} \text{ if } i=KN, K=1 \dots 4$$

$$\varphi_{5N} \rightarrow \varphi_N$$

$$\alpha_i \rightarrow \alpha_{(i \bmod 5)+1}$$

در اینجا مجموعاً جهت رفتن از یک عمل به عمل دیگر نیاز به دریافت N پاسخ نامطلوب متوالی می‌باشد. و در ضمن تعویض اعمال نیز مانند روشهای پیشین بصورت ترتیبی صورت می‌پذیرد.

۵) ساماندهی ترافیک با استفاده از سطل چکه‌ای دارنده اتوماتای یادگیر

کارهای گوناگونی در زمینه استفاده از اتوماتای یادگیر به منظور ساماندهی ترافیک در طول سالهای ۱۹۹۴ الی ۱۹۹۷ توسط واسیلیاکوس^۱ و اتلسیس^۲ صورت پذیرفته است. در اولین کار انجام شده از نوعی اتوماتای یادگیر بنام SELA در جهت بهبود عملکرد سطل چکه‌ای استفاده گردیده است. عملکرد این ساماندهنده آنگونه است که با استفاده از تابع توزیع منبع ON-OFF مقدار ایده‌آل شمارنده LB محاسبه می‌شود و پس از آن این مقدار با مقدار واقعی شمارنده مقایسه می‌شود. هر اندازه اختلاف بیشتری بین این دو مقدار وجود داشته باشد منبع از پارامترهای مورد توافق خود بیشتر تخطی داشته است. با توجه به این تعریف دو عمل برای اتوماتا در نظر گرفته شده است که برطبق یکی از آنها منبع عملکردی عادی (عمل ۱) و برطبق دیگری عملکردی غیرعادی (عمل ۲) دارد. اگر میزان اختلاف گفته شده از یک مقدار آستانه‌ای تجاوز نماید احتمال عمل ۲ افزایش می‌یابد و اگر به یک مقدار آستانه برسد در آنصورت تمامی سلولهای منبع دور ریخته می‌شوند. حال هر زمان که عمل ۱ انتخاب شود سلولهای ورودی به سطل چکه‌ای بدون هیچ تغییری از آن عبور می‌نمایند و اگر احتمال این عمل به یک مقدار مشخص برسد در آنصورت کلیه احتمالات اتوماتا مجدداً مقداردهی اولیه می‌شوند.

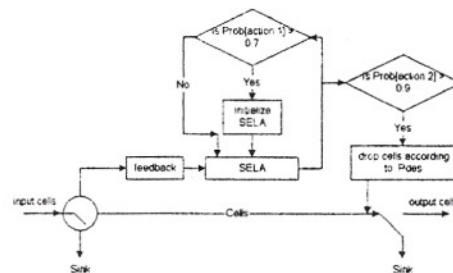
1

- A. V. Vasilakos

2

- A. F. Atlasis

با توجه به شبیه‌سازی‌های صورت گرفته در [۱۱] روش جدید منجر به کاهش قابل توجهی در میزان سلولهای دورریخته شده و کارایی سطل می‌شود. لازم است در اینجا یادآوری شود که این روش در گره‌های ابتدایی شبکه و بعبارت واضح‌تر در محل تولید ترافیک بکار گرفته شده است. عملکرد این ساماندهنده در شکل ۲ نمایش داده شده است.

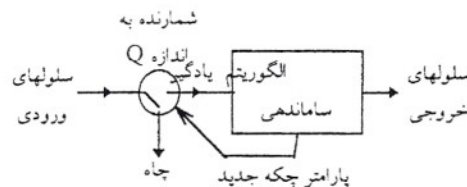


شکل ۲) مکانیزم ساماندهی LB-SELA

کار بعدی صورت گرفته در این راستا توسط واسیلاکوس و اتلیسیس در سال ۱۹۹۵ صورت گرفت [۱۰] و در آن ساماندهنده فوق‌الذکر بجای گره‌های ابتدایی در گره‌های میانی شبکه جای گرفت. شبیه‌سازیهای صورت گرفته با استفاده از این روش نیز نشان می‌دهد که عملکرد سطل چکه‌ای جدید به مراتب بهتر از انواع رایج می‌باشد.

آخرین کار صورت گرفته در این زمینه در سال ۱۹۹۷ با تعیین تابع توزیع شمارنده سطل چکه‌ای منابع ترافیک ویدیویی توسط اتلیسیس و واسیلاکوس صورت پذیرفت [۱۲]. در اینحالت نیز با محاسبه مقدار واقعی و ایده‌آل شمارنده LB و اختلاف آنها تعیین می‌شود که آیا عملکرد منبع برطبق قرارداد بوده است یا نه. و در صورتیکه هرگونه تخلفی مشاهده شود سلولهای منبع خطای دور ریخته می‌شوند. در اینحالت نیز شبیه‌سازیها حاکی از بهبود عملکرد و کارایی بهتری نسبت به سطل چکه‌ای ساده می‌باشد.

در نوع ساده سطل چکه‌ای پارامتر چکه ثابت است. این امر باعث می‌شود هنگامیکه سطل چکه‌ای برای ساماندهی منابع انفجاری مورد استفاده قرار می‌گیرد و بعلاطف طبیعت انفجاری منبع تخمین مشخصات آن مانند میانگین بسادگی انجام نمی‌گیرد و همیشه متضمن خطا می‌باشد و باعث دورریزی سلولها بیش از مقدار مورد توافق در قرارداد می‌شود. در روش پیشنهاد شده از سطل چکه‌ای، نرخهای چکه مختلفی در یک محدوده مجاز در نظر گرفته شده است. هر یک از این نرخها متناظر با یکی از اعمال اتوماتا می‌باشد. اتوماتا دائما با شناسایی مشخصات منبع سعی می‌نماید بهترین نرخ را برای پارامتر چکه به گونه‌ای انتخاب نماید که میزان سلولهای دورریخته شده کاهش یابد و عملکرد ساماندهنده ترافیک بهبود یابد. با توجه به توضیحات فوق نحوه عملکرد سطل چکه‌ای جدید مطابق با شکل ۳ می‌باشد.

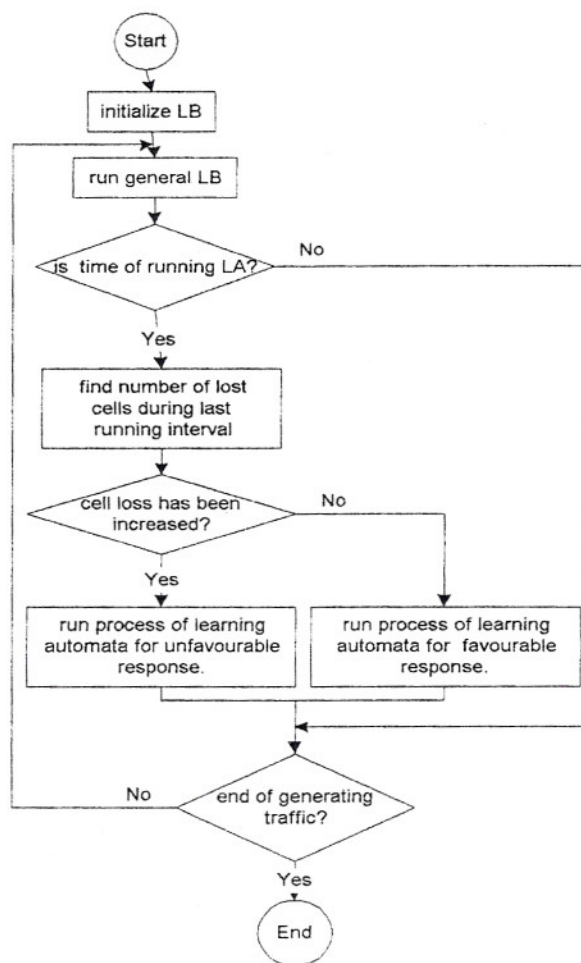


شکل ۳) سطل چکه‌ای با پارامتر چکه متغیر

جزئیات بیشتر این روش بصورت نمودار جریان در شکل ۴ نمایش داده شده است. مطابق این نمودار اتوماتا ابتدا مدت زمانی مشخص را صرف آموزش اولیه می‌نماید. پس از آن در فواصل زمانی مشخص اتوماتا اجرا می‌شود و در هر بار اجرا، تعداد سلولهای دورریخته شده در فاصله زمانی گذشته محاسبه می‌گردد. اگر این تعداد نسبت به دفعه ماقبل آن بهبود یافته باشد به معنای پاسخ مطلوب ارزیابی می‌شود. حال برحسب آنکه چه نوعی از اتوماتا (ساختار ثابت یا متغیر) مورد استفاده قرار گرفته باشد این پاسخ مطلوب در احتمال و یا حالت اعمال تاثیر می‌گذارد. اما اگر تعداد سلولهای دورریخته شده نسبت به دفعه ماقبل آن افزایش یافته باشد به منزله پاسخ نامطلوب در نظر گرفته می‌شود و متناسب با ساختار اتوماتا تاثیر خود را بر احتمالات و حالات اعمال می‌نماید و در صورت تعویض عمل اتوماتا اندازه سطل جدیدی را انتخاب می‌نماید. مدت زمانی که برای آموزش اولیه اتوماتا بکار می‌رود و همچنین فواصل زمانی اجرای اتوماتا مقداری ثابت هستند که می‌توان آنها را متناسب با منابع ترافیک گوناگون تنظیم نمود. روش جدید در مقایسه با روشهای مبتنی بر اتوماتای یادگیر [۷] و [۱۰-۱۲] از دو جنبه متفاوت است. اولاً معیار عملکرد مطلوب و یا نامطلوب منبع در روشهای پیشین میزان انحراف از مقدار ایده‌آل شمارنده سطل چکه‌ای می‌باشد ولی در روشهای پیشنهادی معیار جدیدی بر مبنای تعداد سلولهای گم‌شده در بازه‌های زمانی پیشین تعریف شده است. با استفاده از این معیار تصمیم‌گیری‌ها بیشتر بر واقعیت منطبق است. و عملکرد اتوماتا و اینکه آیا از قرارداد خود تخطی داشته است را می‌توان براساس یک معیار عینی (مانند میانگین سلولهای دورریخته شده در یک بازه زمانی) بیان نمود. ثانیاً در این روش اتوماتاهایی کاملاً متفاوت با انواع طراحی شده توسط اتلیسیس و واسیلاکوس بکار گرفته شده است.

در روش پیشنهادی، ATM به عنوان یک سرویس مورد توجه قرار گرفته است تا یک بستر ارتباطی. اگر ATM بعنوان یک بستر در نظر گرفته شده باشد روشهای پیشنهادی که از اتوماتاهای با ساختار ثابت استفاده می‌کنند مناسبتر می‌باشد. بدلیل اینکه پیاده‌سازی اتوماتاهای با ساختار

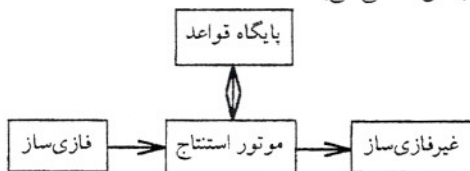
متغیر در سخت‌افزار بسیار کند عمل می‌نماید استفاده از آنها در شبکه‌های پرسرعتی مانند ATM مناسب نمی‌باشد. یکی از مزایای پیاده‌سازی سخت‌افزاری روشهای پیشنهادی بیشتر شدن قابلیت انعطاف ساماندهنده جهت استفاده به همراه کاربردهای گوناگون می‌باشد. زیرا در این حالت ساماندهنده خود را متناسب با عملکرد کاربرد جدید تنظیم می‌نماید و نیازی به تغییر ساماندهنده بدلیل تغییر کاربرد نمی‌باشد.



شکل ۴) نمودار عملکرد سطل چکه‌ای جدید

۶) ساماندهی ترافیک با استفاده از منطق فازی

همانگونه که در مباحث پیشین بیان شد هدف از بکاربردن اتوماتای یادگیر تنظیم پارامترهای گوناگون سطل چکه‌ای در جهت بهبود عملکرد ساماندهی منابعی که شناخت کاملی از آنها در دسترس نیست می‌باشد. علاوه بر استفاده از اتوماتای یادگیر به کمک سایر روشها نیز می‌توان اقدام به تنظیم این پارامترها نمود که از مهمترین آنها می‌توان به منطق فازی اشاره نمود. با استفاده از یک کنترل‌کننده فازی می‌توان با بهره‌گیری از مجموعه‌ای از متغیرهای زبانی ورودی و تعدادی قواعد کنترل فازی اقدام به تعیین یک عمل کنترلی نمود. در ادامه به بیان چگونگی طراحی یک ساماندهنده ترافیک فازی پرداخته و نحوه کاربرد آنرا در شبکه‌های ATM مورد مطالعه قرار می‌دهیم. ساماندهنده فازی مطابق شکل ۵ دارای چهار بخش اساسی می‌باشد:



شکل ۵) اجزای مختلف ساماندهنده ترافیک فازی

- فازی‌ساز: پارامتر ورودی سیستم را به متغیر زبانی مناسب جهت استفاده در موتور استنتاج تبدیل می‌کند.
- پایگاه قواعد فازی: در بردارنده تعدادی قواعد کنترل فازی است که بصورت زبانی جهت تشریح رویه کنترل نوشته شده است.

• موتور استنتاج : اقدام به استنتاج عمل کنترل فازی براساس قواعد کنترل فازی و پارامترهای زبانی ورودی می‌نماید.

• غیرفازی‌سازی: عمل کنترل فازی استنتاج شده را به یک عمل کنترل غیرفازی، و براساس یک استراتژی غیرفازی‌سازی تبدیل می‌نماید.

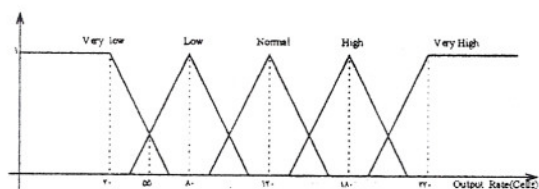
اساس عملکرد این ساماندهنده به آن صورت است که هر اندازه نرخ ورودی بیشتر شود نرخ خروجی نیز افزایش می‌یابد و بالعکس با کاهش نرخ ورودی نرخ خروجی نیز کاهش می‌یابد. این روش کنترل علاوه بر کاهش تعداد سلولهای دورریخته شده امکان بهره‌وری هر چه بیشتر منابع را نیز فراهم می‌کند. جهت طراحی این کنترل کننده پنج نوع متغیر زبانی برای بیان چگونگی تغییرات نرخ ورودی و یک متغیر زبانی خروجی جهت نرخ خروجی تعریف شده است.

با بکار بردن هریک از متغیرهای زبانی ورودی به همراه متغیر زبانی خروجی، پنج نوع روش فازی طراحی شده است که هریک از آنها بصورت مجزا مورد آزمایش قرار گرفته‌اند و درصد سلولهای دورریخته شده توسط آنها محاسبه گردیده و نتایج حاصله در نمودار ۶ نمایش داده شده است.

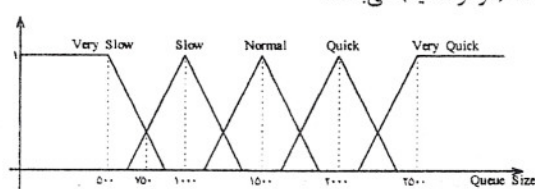
قواعد فازی بکار رفته توسط موتور استنتاج بصورت زیر است:

Rule 1) IF input rate is very slow	THEN output rate	is very low.
Rule 2) IF input rate is Slow	THEN output rate is	low.
Rule 3) IF input rate is Normal	THEN output rate is	Normal..
Rule 4) IF input rate is Quick	THEN output rate	is High..
Rule 5) IF input rate is Very Quick	THEN output rate is	Very High.

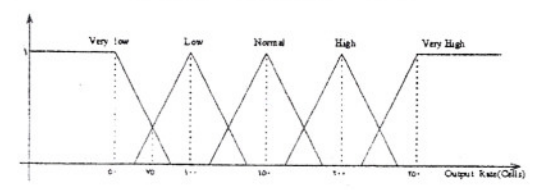
توابع عضویت برای متغیرهای زبانی ورودی و خروجی در اشکال ۶ تا ۱۱ نمایش داده شده است. در ضمن موتور استنتاج بکاررفته از نوع COA (مرکز ناحیه) می‌باشد.



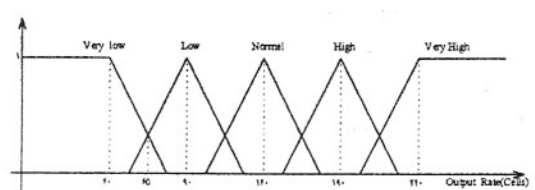
شکل ۷) متغیر زبانی نرخ خروجی نوع ۱



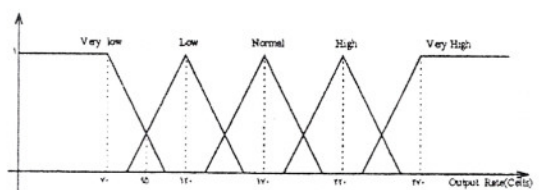
شکل ۶) متغیر زبانی نرخ ورودی



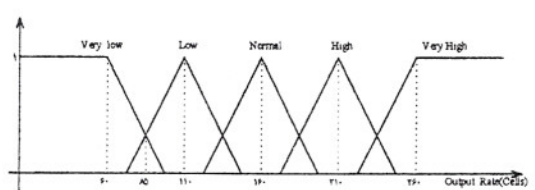
شکل ۹) متغیر زبانی نرخ خروجی نوع ۳



شکل ۸) متغیر زبانی نرخ خروجی نوع ۲



شکل ۱۱) متغیر زبانی نرخ خروجی نوع ۵



شکل ۱۰) متغیر زبانی نرخ خروجی نوع ۴

۷) شبیه‌سازی

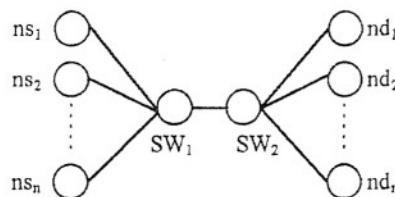
جهت انجام شبیه‌سازیها با استفاده از روشهای پیشنهادی مبتنی بر اتوماتاهای یادگیر و همچنین روشهای پیشنهادی مبتنی بر منطق فازی، از شبیه‌ساز ATM/HFC تولید شده در موسسه NIST استفاده گردیده است. این شبیه‌ساز دارای قابلیت مطالعه و ارزیابی شبکه‌های ATM و HFC می‌باشد و یک محیط مدلسازی محاوره‌ای به همراه واسط کاربر گرافیکی برای کاربر فراهم می‌کند. موسسه NIST این ابزار را با استفاده از زبان C و سیستم XWindows که در بستر یونیکس اجرا می‌گردد تولید نموده است. با استفاده از این ابزار کاربر می‌تواند اقدام به ایجاد همبندی‌های گوناگون شبکه، کنترل پارامترهای مختلف اجزاء، مشاهده فعالیت شبکه و ثبت نمودن داده‌های حاصل از شبیه‌سازیها نماید [۱۳]. در شبیه‌سازیهای صورت گرفته از منابع ON-OFF استفاده شده است که مشخصات آن در جدول ۱ آورده شده است.

1 - Center Of Area
2 - Hybrid Fiber Coax

پارامتر منبع	مقدار
مدت زمان میثگین ON	4240 usec
مدت زمان میثگین OFF	3816000 usec
نرخ حداکثر ترافیک ارسالی	10 Mbit/s
نرخ میثگین ترافیک ارسالی	1 Mbit/s
مدت زمان اجرا	500 sec

جدول (۱) مشخصات منابع ON-OFF مورد استفاده

همبندی استفاده شده در شبیه‌سازیها مطابق با شکل ۱۱ می‌باشد.



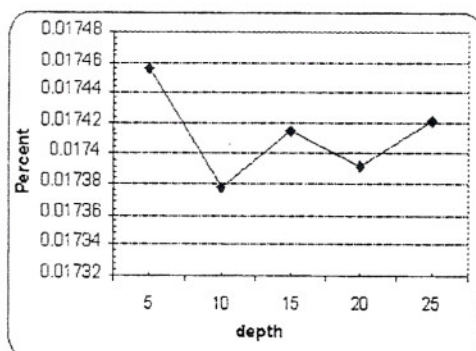
شکل (۱۱) همبندی شبکه بکار رفته

در این همبندی تعداد ۲۰۰ عدد منبع اقدام به ارسال ترافیک به مقصد می‌نمایند. از آنجائیکه ساماندهی ترافیک فقط در مبدا انجام می‌گیرد تعداد سلولهای دورریخته شده تنها در مبدا محاسبه شده و مورد بررسی قرار گرفته است. این بررسی برای انواع مختلف اتوماتاها صورت پذیرفته است و نتایج آن در انتهای این قسمت آورده شده است. جهت مقایسه ابتدا عملکرد سطل چکه‌ای ساده با مشخصات جدول ۲، توسط شبیه‌سازی مورد مطالعه قرار گرفت که برای آن متوسط سلولهای دور ریخته شده ۰/۲۷۰۸ درصد بوده است که در مقایسه با نتایجی که از روش سطل چکه‌ای که در آن اتوماتاهای یادگیر استفاده شده است بمراتب بالاتر بوده است.

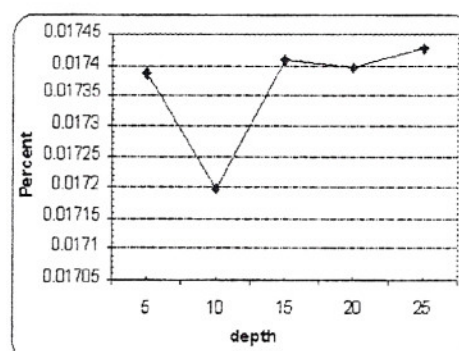
پارامتر سطل چکه‌ای	مقدار
نرخ چکه	1465000 bit/s
اندازه سطل	2500 cell
درصد متوسط سلولهای دور ریخته شده	0.27087%

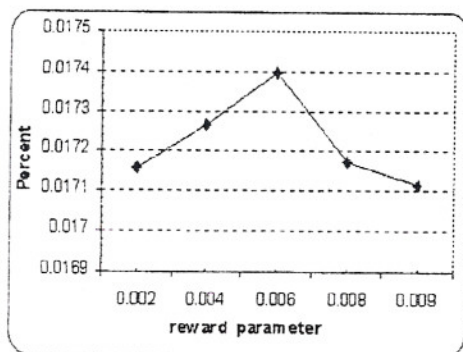
جدول (۲) مشخصات سطل چکه‌ای بکار رفته

شبیه‌سازیهای روش پیشنهادی برای اتوماتاهای L_{pi} ، L_{np} ، کرینسکی، $L_{SN,S}$ و $G_{SN,S}$ و ساماندهنده‌های فازی انجام گرفت که نتایج حاصل در نمودارهای ۱ تا ۶ ارائه گردیده است.

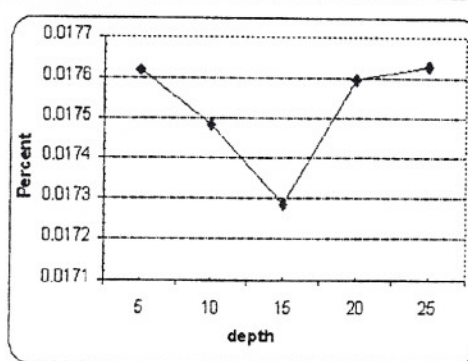


نمودار (۲) تغییرات درصد سلولهای گم‌شده برحسب عمق‌های مختلف اتوماتای کرینسکی

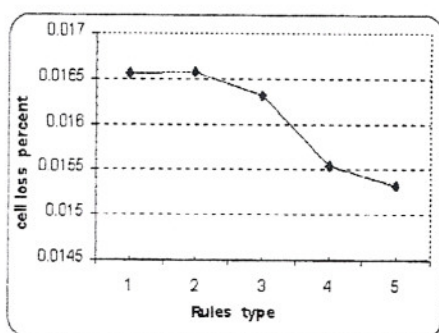

 نمودار (۱) تغییرات درصد سلولهای گم‌شده برحسب عمق‌های مختلف اتوماتای $L_{SN,S}$



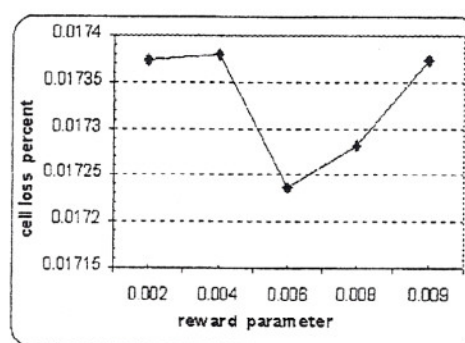
نمودار ۴) چگونگی تغییر درصد سلولهای گمشده بر حسب مقادیر مختلف پارامتر پاداش و جریمه در اتوماتای I_{np}



نمودار ۳) تغییرات درصد سلولهای گمشده بر حسب عمق های مختلف اتوماتای $G_{NN,S}$



نمودار ۶) تغییرات درصد سلولهای گمشده بازی هریک از قواعد ساماندهنده فازی



نمودار ۵) تغییرات درصد سلولهای گمشده بر حسب پارامتر پاداش در اتوماتای I_{np}

برطبق نمودار ۱ اتوماتای $L_{NN,S}$ با عمق حافظه ۱۰ منجر به کمترین میزان دورریختگی سلول می شود. برای عمقهای ۵، ۱۵ و ۲۰ عملکرد اتوماتا تقریباً یکسان است. با افزایش عمق درصد سلولهای گمشده روبه فزونی می گذارد. اتوماتای کرینسکی برطبق نمودار ۲ بدترین عملکرد را برای عمق ۵ و بهترین عملکرد را برای عمق ۱۰ تولید می کند. با افزایش بیشتر عمق درصد سلولهای دورریخته شده به تدریج روبه ازدیاد می گذارد. با توجه به نمودار ۳ اتوماتای $G_{NN,S}$ در عمق ۱۵ بهترین عملکرد را دارا می باشد. بازی عمق های دیگر عملکرد بدتر شده و از عملکرد بهینه خود دور می شود. برطبق نمودار ۴ اتوماتای I_{np} زمانیکه پارامتر پاداش ۰/۰۰۶ است بدترین عملکرد را عرضه می دارد اما هر اندازه که بر میزان پارامتر پاداش افزوده می شود مشاهده می گردد که عملکرد آن بهبود می یابد. بدیهی است که این روند صعودی نمی تواند دائماً ادامه یابد و در جایی متوقف خواهد شد که جهت بدست آوردن آن نیاز به شبیه سازیهای بیشتری است.

اتوماتای I_{np} بازی پارامتر پاداش ۰/۰۰۶ دارای عملکرد بهینه است و در سایر نقاط منجر به دورریزی سلولهای زیادتری می شود که استفاده از آن مقادیر را غیر ضروری می کند. با مقایسه نمودار ۶ با نمودارهای شماره ۱ الی ۵ مشاهده می شود که عملکرد روش فازی بهتر از روش های یادگیر پیشنهادی می باشد. البته نوعی اتوماتای ۱۵-عملی جهت ساماندهی ترافیک به منظور تنظیم اندازه سطل نیز پیشنهاد گردیده است که عملکرد آن بهتر از انواع ۵-عملی مشابه می باشد. در این اتوماتا فواصلی که بین اعمال اتوماتا وجود دارد کاهش یافته و از سوی دیگر تعداد آنها نیز افزایش یافته است این امر منجر به آن گردیده است که عملکرد اتوماتا و تغییر اعمال آن تا حد زیادی به عملکرد روشهای فازی که در آنها متغیرهای زبانی ورودی و خروجی حول یک نقطه در تغییر هستند نزدیکتر شود و از سوی دیگر کارایی اتوماتا در جهت کاهش تعداد سلولهای دورریخته شده افزایش یابد.

پس از بررسی کارایی اتوماتای یادگیر در ساماندهی ترافیک در ادامه به مقایسه این روشها با ساماندهنده ترافیک پیشنهاد شده در [۶] پرداخته می شود. مطابق نمودار ارائه شده در شکل A این مقاله هنگامیکه نرخ توکنها در محدوده ۰/۵ تا ۰/۵۶ برابر نرخ میانگین منبع تغییر می کند عملکرد اتوماتای یادگیر بهتر از روش فازی می باشد زیرا در این حالت احتمال دور ریختن سلولها در ساماندهنده دارای اتوماتای یادگیر در محدوده ۰/۰۱۷ الی ۰/۱۷۵ تغییر می کند اما محدوده تغییر این احتمال برای روش فازی محدوده ۰/۰۱۷ تا ۰/۰۳ است. درضمن با تنظیم مناسب نسبت پارامتر چکه و نرخ میانگین منبع می توان این بهبود را در تمامی محدوده ها بدست آورد.

۸) نتیجه گیری و کارهای آتی

در این مقاله روش سطل چکه ای معمولی و سطل چکه ای با پارامتر چکه متغیر با یکدیگر مقایسه شده اند. جهت تنظیم پارامتر چکه از اتوماتاهای یادگیر با ساختار ثابت مانند کرینسکی و ساختار متغیر نظیر I_{np} و منطق فازی استفاده گردیده است. نتایج حاصل از شبیه سازیها بیانگر بهبود قابل توجهی در عملکرد سطل چکه ای می باشد. با توجه به انواع گوناگون اتوماتای یادگیر و تنوع بسیاری که می توان از جهت تعداد اعمال،

ساختار اتوماتا و یا حتی ترکیب انواع گوناگون از اتوماتا با یکدیگر بدست آورد می‌توان بهبودهای بیشتری در عملکرد اتوماتاها ایجاد نمود. از جمله این کارها می‌توان تغییر تعداد اعمال اتوماتا، تنظیم مناسب فواصل زمانی اجرای اتوماتا، بررسی چگونگی تنظیم اتوماتا برای انواع منابع ترافیک و ترکیب انواع اتوماتا را نام برد.

۹) فهرست مراجع

- 1- CCITT, Recommendation I.121 : "Broadband aspects of ISDN", Blue Book, Vol. III. 7, Geneva, Switzerland, 1989.
- 2- E. P. Rathgeb, "Modeling and performance comparison of policing mechanisms for ATM networks", IEEE J-SAC April 1991 pp 325-334.
- 3- A.I. Elwalid, D. Mitra, "Analysis and design of rate-based congestion control of high speed networks, I: stochastic fluid models, access regulation", Queueing Systems 9 1991, pp. 29-63.
- 4- Milena Butto, Elisa Cavallero and Alberto Tonietti, "Effectiveness of the leaky bucket policing mechanism in ATM networks", IEEE. J. Select. Areas. Commun., vol.9, no.3, pp. 335-342 April 1991.
- 5- Irfan Khan, et al, "Traffic control in ATM networks", Computer Networks and ISDN System 27(1994)85-100.
- 6- Z. Jiang, Z. Liu, "An improved algorithm of usage parameter control in ATM networks", ICCT'96, vol.1 pp. 24-27.
- 7- Y. Seifi, M.R. Meybodi, "Traffic policing in ATM networks using two action I_{rep} learning automata", Technical report, Computer eng. dep., Amirkabir University of Technology, Tehran, Iran, 2001.
- 8- G. Gallassi, G. Rigolio, L. Fratta, "ATM: Bandwidth assignment and bandwidth enforcement policies", IEEE GLOBECOM 1989, pp.178801793.
- 9- J. A. S. Monteiro, M Gerla, L. Fratta, "Leaky bucket input rate control in ATM networks", ICC 1990, New Delhi, India, 1990, pp. 370-376.
- 10- A. V. Vasilakos, A. F. Atlasis, "Effectiveness of the LB-SELA policing mechanism in an ATM network node", IEEE GLOBECOM'95, November 1995, pp. 627-631.
- 11- A. V. Vasilakos, A. F. Atlasis, "LB-SELA: rate-based access protocol for ATM networks", IEEE INFOCOM June 1994, pp. 1552-1559.
- 12- A. F. Atlasis, G.I. Stassinopoulos, A. V. Vasilakos, "Leaky bucket mechanism with learning algorithm for ATM traffic policing", second IEEE symposium on computer and communications, July 1997, pp 68-72.
- 13- N. Golmie, F. Mouveaux, L. Hester, Y. Saintillan, A. Koenig, D. Su, "The NIST ATM/HFC network simulator version 4.0", high-speed networks technology group, advanced networks technology division, information technology laboratory. NIST, December 1998.