

TCP

*

^۱ دانش‌آموخته کارشناسی ارشد برق - کنترل - دانشکده فنی - دانشگاه تربیت مدرس

آستادیار گروه برق - کنترل - دانشکده فنی - دانشگاه تربیت مدرس
(// // //)

مدیریت فعال صف (AQM) با کلاس دور ریز/علامت‌گذاری کردن بسته‌ها در صف مسیریاب‌ها بعنوان یک پیشنهاد جدید در کنترل تراکم شبکه‌های TCP/IP با پروتکل انتها به انتها مطرح شده است. مدیریت فعال صف در مسیریاب‌های شبکه TCP/IP به منظور بهره‌برداری بیشتر از پهنای باند موجود و کاهش تاخیر انتقال صورت می‌گیرد و یک مسئله کنترل پسخوردی است. با علم به این موضوع که دینامیک شبکه‌های TCP/IP متغیر با زمان و غیرخطی است در این مقاله از یک مدل CARIMV و یک کنترلگر پسخورد تطبیقی به منظور کنترل ازدحام استفاده شده است. استراتژی کنترل به این صورت است که ابتدا یک پیش‌بینی از اندازه بافر گلوگاه صورت گرفته سپس پارامترهای کنترلگر طوری تنظیم می‌شوند که تابع هزینه بکار گرفته شده مینیمال شود. در پایان نتایج بدست آمده به کمک نرم افزار MATLAB شبیه‌سازی و با نتایج حاصل از کارهای قبلی نظیر الگوریتم RED و کنترلگرهای PI و PID مقایسه شده است.

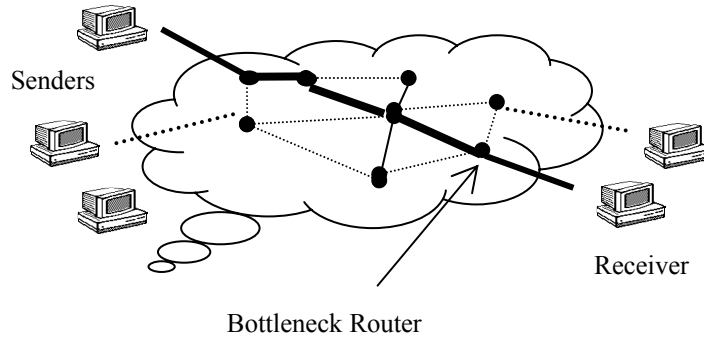
: کنترل پیش‌بین تطبیقی - مدیریت فعال صف - الگوریتم RED - مدل CARIMV

مقدمه

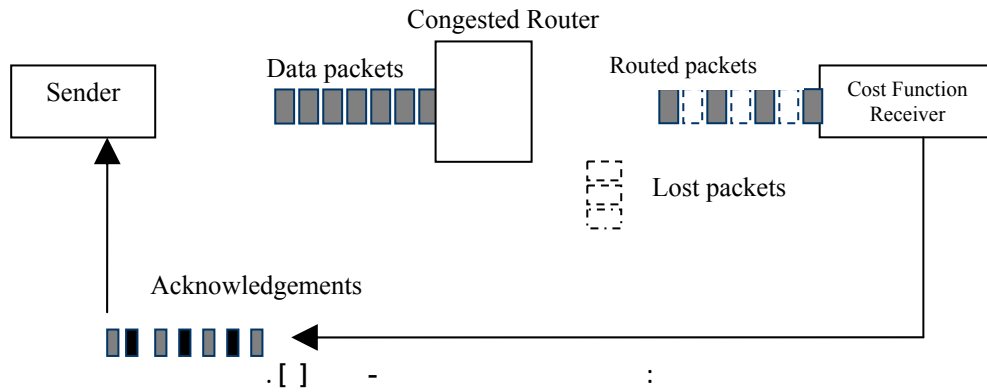
می‌شود. با توجه به پروتکل انتها به انتهای^۱ TCP، در الگوریتم RED برای انجام پسخورد به فرستنده‌ها به طور غیرمستقیم عمل می‌شود به این صورت که در حالت ازدحام، بسته‌ها به صورت تصادفی علامت‌گذاری/دور ریز می‌شوند. گیرنده‌ها اطلاعات ناشی از علامت‌گذاری و یا دور ریز بسته‌ها را به فرستنده‌ها ارسال داشته و پسخورد کامل می‌گردد. فرستنده‌ها به محض دریافت چنین اطلاعاتی نرخ ارسال خود را کاهش و یا افزایش خواهند داد.

روشن است که رفع ازدحام^۲ به محض کشف علائم آن بسیار کارآمدتر است تا اینکه بگذاریم اوضاع به هم بریزد سپس برای رفع آن اقدام کنیم. علامت‌گذاری بسته‌ها در RED بخاطر حذف جریان‌های همزمانسازی است، تصادفی بودن به این خاطر است که مسیریاب نمی‌تواند تشخیص دهد کدام منبع سبب بروز مشکلات بیشتری شده و استفاده از میانگین صف برای کم کردن ترافیک تهاجمی^۳ است. در هر حال الگوریتم RED مشکلاتی هم دارد. یکی از مشکلات آن اینست که تنظیم پارامترهایش بسیار مشکل است (به مراجع [۸] و [۹]

توسعه مدیریت فعال صف^۱ نقش کلیدی در افزایش تقاضا برای خدمات اینترنت دارد. از این خدمات به ارسال صدا از طریق اینترنت^۲ (VOIP) و کلاس‌بندی سرویس^۳ (CoS) می‌توان اشاره نمود. برای شروع یک مسیر ارتباطی بین فرستنده و گیرنده از طریق یک مسیریاب گلوگاه بصورت شکل(۱) در نظر می‌گیریم. این ارتباط با شماتیک کنترلی در شکل(۲) نشان داده شده است. فرستنده نرخ ارسال خود را به صورت خطی افزایش می‌دهد تا اینکه بسته‌ها مفقود شوند به محض اینکه بسته‌ای مفقود گردید، گیرنده به فرستنده سیگنالی ارسال می‌کند تا نرخ ارسالش را کاهش دهد. بعضی از مشکلات این روش شامل جریان‌های همزمانسازی^۴ و تضعیف عملکرد (آغاز مجدد^۵ و انقضای مهلت^۶ فراوان) ناشی از دور ریز فراوان بسته‌هاست. در مرجع [۱]، الگوریتم RED^۷ به منظور بهبود عملکرد شبکه معرفی شد تا مسیریاب‌ها در اداره شبکه به TCP کمک کنند. نسبت به حالت دور ریز بسته‌ها^۸، در الگوریتم RED اینطور عمل می‌شود که ابتدا طول صف مسیریاب اندازه‌گیری شده سپس بر اساس مقادیر اندازه‌گیری شده نرخ ارسال فرستنده‌ها تعیین



: - [۱].



: - [۱].

این مرجع، [۱۳]، دقت مدل را نشان داده است. این مدل مقادیر میانگین متغیرهای کلیدی شبکه را در نظر گرفته و با جفت معادله تفاضلی غیرخطی زیر تشریح می‌شود:

$$\dot{w}(t) = \frac{1}{R(t)} - \frac{w(t)}{2} \frac{w(t) - R(t)}{R(t) - R(t)} p(t - R(t)) \quad (1)$$

در این معادله‌ها:

w : میانگین اندازه پنجره ازدحام TCP (packets)

q : میانگین طول صف (packets)

$R(t)$: زمان تاخیر حلقه (secs)، $R(t) = T_p + \frac{q(t)}{C}$

C : ظرفیت لینک (packets/sec)

T_p : تاخیر انتشار (secs)

N : تعداد منابع ارسال کننده داده

p : احتمال علامت دار شدن بسته‌ها

است.

از این پس p (احتمال علامت دار شدن بسته‌ها) به عنوان سیگنال کنترل و $q(t)$ (طول صف بافر) به عنوان خروجی سیستم در نظر گرفته شده است.

مراجعه شود). در مراجع [۱۰-۱۲] کارهایی به منظور بهبود عملکرد این الگوریتم شده است. در مرجع [۲] از یک کنترلگر کلاسیک PI استفاده شده است پارامترهای این کنترلگر براساس مدل خطی شده دینامیک بافر حول نقطه کار طراحی شده‌اند.

در هر حال کنترلگرهای فوق برای یک منطقه عملکردی خاص طراحی شده‌اند و این در حالیست که پارامترهای شبکه دائماً در حال تغییر می‌باشند، لذا استفاده از کنترلگر تطبیقی مناسب‌تر به نظر می‌رسد.

با توجه به مطالب فوق و با توجه به اینکه هم در مسیر پیشرو و هم در مسیر پسخور تاخیر داریم در این مقاله از یک کنترلگر پیش‌بین تطبیقی که ذاتاً زمان‌های مرده را جبران‌سازی می‌کند، استفاده شده است.

TCP

در مرجع [۱۳] یک مدل دینامیکی از رفتار TCP براساس آنالیز معادله تفاضلی اتفافی^{۱۳} و جریان سیال بدست آمده است. نتایج شبیه‌سازی‌های انجام گرفته در

و $C(s)e^{-sR_0}$ را پایدار کند سیستم کنترل AQM خطی شده پایدار خواهد بود.

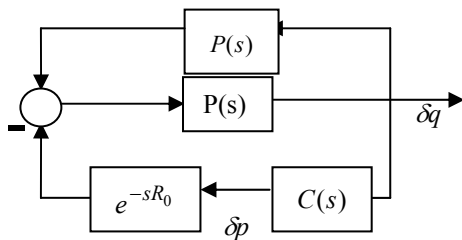
یعنی برای طراحی پارامترهای کنترلگر تنها تابع انتقال $p(s)e^{-sR_0}$ در نظر گرفته می‌شود. در نهایت پارامترهای RED به کمک این قضیه و با استفاده از قوانین ساده طراحی کنترل خطی برای یک حدفاز و حدبهره مطلوب طراحی شده است.

برای مثال برای $N = 60$ و $C = 3750 \text{ packets/sec}$ و نقطه کار $W_0 = 15$ ، $p_0 = 0.008$ و $R_0 = 0.246s$:

$$p(s) = \frac{1.17126 * 10^5}{(s + 4.1)(s + 0.53)}$$

با $\Delta(s) = 2.24 * 10^{-6} s(1 - \exp(-0.246s))$ خواهد بود که با توجه به این مقادیر، تابع انتقال الگوریتم RED به صورت زیر بدست آمده است [۲]:

$$C(s) = \frac{(5 * 10^{-3})(1.86 * 10^{-4})}{(s + 0.005)} \quad (4)$$



شکل ۴: بلوک دیاگرام AQM سیستم خطی شده [۲].

PID

به منظور بهبود عملکرد RED در مرجع [۲] کنترلگر کلاسیک PI و در مرجع [۱۵] کنترلگر کلاسیک PID بکار گرفته شده است. در این مقالات به منظور تعیین ضرایب کنترلگر از مدل خطی شده (۱) برای $N = 60$ ، $p_0 = 0.008$ ، $W_0 = 15$ و نقطه کار $C = 3750 \text{ packets/sec}$ و $R_0 = 0.246s$ و قضیه گفته شده در پاراگراف قبل استفاده شده است. نتایج حاصل از طراحی کنترلگرها به صورت زیر می‌باشد:

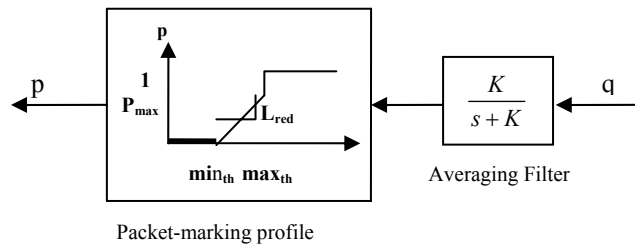
$$PI: C(s) = 9.64 * 10^{-6} \frac{\frac{s}{s} + 1}{s} \quad (5)$$

$$PID: C(s) = 5.1 * 10^{-5} \frac{s^2 + 1.22s + 0.61}{s} \quad (6)$$

RED

ساده‌ترین راهی که در کنترل تراکم وجود دارد اینست که منابع TCP به محض سرریز شدن بافر گلوگاه اندازه پنجره‌شان را کوچکتر کنند. این روش بنام Tail-Drop معروف است و قانون کنترل آن به صورت زیر می‌باشد:

$$\begin{cases} p = 1 & \text{when } q \geq \bar{q} \\ p = 0 & \text{otherwise} \end{cases} \quad (2)$$



RED [۲].

این مکانیزم on-off سبب نوسانات طول صف، جریانهای همزمانسازی و تنزیل عملکرد ناشی از انقضای مهلت و آغاز مجدد فراوان می‌شود [۵ و ۱۴]. مدیریت فعال صف در الگوریتم RED به این صورت است که احتمال دور ریز بسته‌ها، p ، به عنوان تابعی از طول صف محاسبه می‌شود.

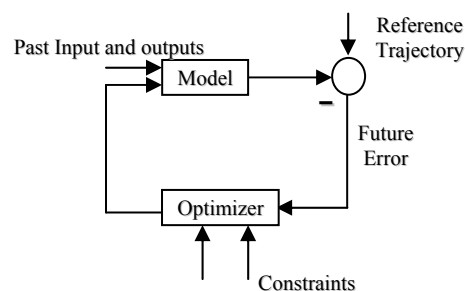
همانطور که از شکل (۳) پیداست الگوریتم RED شامل یک فیلتر پایین‌گذر برای میانگین‌گیری از طول صف و یک فرایند علامت‌گذاری بسته‌هاست. تنظیم RED شامل تعیین قطب فیلتر پایین‌گذر K ، آستانه \min_{th} ، سطح p_{max} و بهره L_{red} است.

یکی از مشکلات RED این است که تنظیم پارامترهای آن برای یک عملکرد بهینه بسیار مشکل است. در مرجع [۲] فرض شده که N و C ثابت هستند. p بعنوان ورودی، (w, q) متغیرهای حالت و q خروجی است. مدل (۱) حول نقطه کار W_0 ، R_0 و p_0 خطی‌سازی شده و در نهایت بلوک دیاگرام AQM سیستم خطی شده به صورت شکل (۴) بدست آمده است. تابع انتقال سیستم خطی $H(s) = \frac{\delta q(t)}{\delta p(t)}$ و کنترلگر $C(s)$ است. در این مقاله بصورت یک قضیه ثابت شده است که اگر:

$$|\Delta(j\omega)w(j\omega)| < 1 \quad (3)$$

[] MPC

به منظور بررسی این استراتژی یک ساختار پایه در شکل (۵) نشان داده شده است. خروجی کنترلر بوسیله بهینه‌ساز با در نظر گرفتن تابع هزینه (خطاهای آینده در نظر گرفته می‌شود) و قیدها محاسبه می‌شود.



.MPC :

مدل باید طوری انتخاب شود که توانایی در برگرفتن دینامیک‌های فرایند را داشته باشد و بتواند خروجی‌های آینده فرایند را با دقت خوبی پیش‌بینی کند در ضمن بکارگیری آن ساده و قابل فهم باشد. MPC روش یکتایی ندارد و مجموعه‌ای از روش‌های متفاوت است که با انواع مدل‌های مختلف و فرمولسازهای متفاوت بکار برده می‌شود [۶].

در اینجا از یک مدل $CARIMV^{15}$ به منظور پیش‌بینی طول صف بافر استفاده شده است. به منظور تنظیم تطبیقی پارامترهای مدل از الگوریتم معروف $NLMS^{16}$ به صورت بازگشتی استفاده گردیده و قانون کنترل با مینیمم سازی تابع هزینه بدست آمده است، شکل (۶).

[] CARIMV

مدل $CARIMV$ که برای مدل‌سازی ورودی-خروجی سیستم بکار می‌رود به صورت زیر است:

$$A(q^{-1})y(k) = q^{-d}B(q^{-1})u(k-1) + C(q^{-1})\frac{w(k)}{\Delta} \quad (7)$$

که

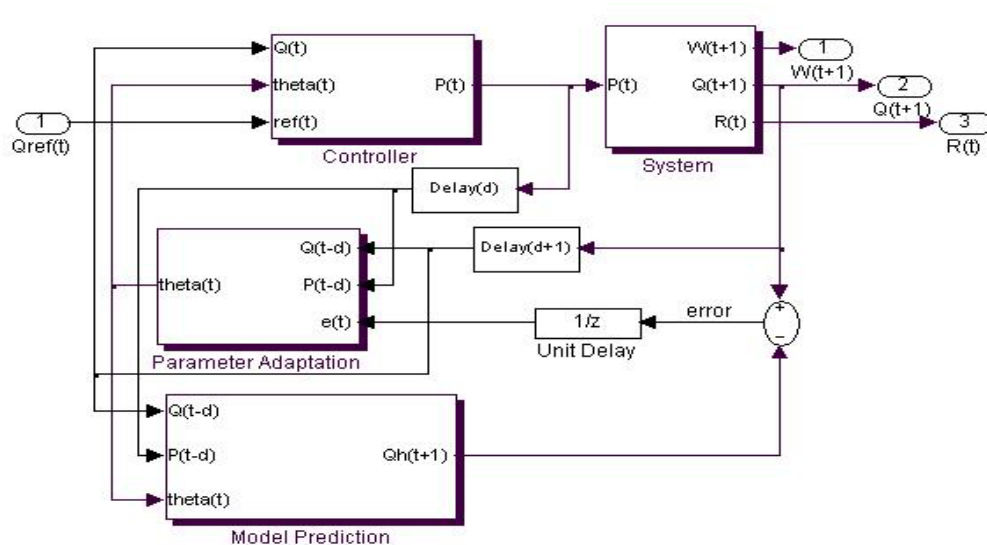
$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + \dots + a_nq^{-na}$$

$$B(q^{-1}) = 1 + b_1q^{-1} + b_2q^{-2} + \dots + b_nq^{-nb}$$

$$C(q^{-1}) = 1 + c_1q^{-1} + c_2q^{-2} + \dots + c_nq^{-nc}$$

$$\Delta = 1 - q^{-1}$$

q^{-1} بیانگر اپراتور تاخیر می‌باشد و $w(k)$ را نویز سفید در نظر می‌گیریم.



هدف از طراحی کنترلرگر بهینه جریان ترافیک بدست آوردن مقادیر $u(k)$ به گونه‌ای که تابع هزینه

$$J = E\left\{\frac{1}{2}(\hat{y}(t+k|t) - y_r(t+k))^2 + \frac{\lambda}{2}\Delta u^2(t)\right\} \quad (13)$$

حداقل شود، است. y_r مقدار مرجع می‌باشد. با استفاده از معادله (12) داریم:

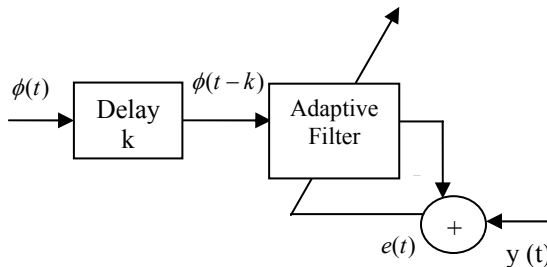
$$\frac{\partial \hat{y}(t+k|t)}{\partial \Delta u(t)} = \beta_0 \quad (14)$$

آنگاه به کمک معادله‌های (12 تا 14) خواهیم داشت:

$$\frac{dJ}{d\Delta u(t)} = (\lambda + \beta_0^2)\Delta u(t) + \beta_0(-y_r(t+k) + \sum_{i=0}^p \alpha_i y(t-i) + \sum_{i=1}^q \beta_i \Delta u(t-i)) \quad (15)$$

با صفر قرار دادن معادله (15) بدست خواهیم آورد:

$$u(t) = u(t-1) + \frac{\beta_0}{\lambda + \beta_0^2}(y_r(t+k) - \sum_{i=0}^p \alpha_i y(t-i) - \sum_{i=1}^q \beta_i \Delta u(t-i)) \quad (16)$$



[] :

پارامترهای $(\alpha_0, \alpha_1, \dots, \alpha_p)$ و $(\beta_0, \beta_1, \dots, \beta_q)$ برای محاسبه $\hat{y}(t+k|t)$ و سیگنال کنترل $u(t)$ لازم هستند. بلوک دیاگرام تنظیم تطبیقی پارامترها در شکل (7) نشان داده شده است. با توجه به تئوری پردازش سیگنال تطبیقی ما می‌توانیم از تفاضل مقدار تخمین $\hat{y}(t)$ و مقدار واقعی طول بافر $y(t)$ ، $e(t)$ ، به منظور تخمین پارامترهای نامعلوم $(\alpha_0, \alpha_1, \dots, \alpha_p)$ و $(\beta_0, \beta_1, \dots, \beta_q)$ استفاده کنیم. برای اینکار از الگوریتم معروف NLMS، به صورت زیر استفاده می‌کنیم.

به منظور حذف ترم نویز $w(k)$ در معادله (7) از معادله دیوفانتین به صورت زیر استفاده می‌شود:

$$C(q^{-1}) = E_k(q^{-1})\hat{A}(q^{-1}) + z^{-k}F_k(q^{-1}) \quad (8)$$

$\hat{A} = \Delta A$ است و $E_k(q^{-1})$ و $F_k(q^{-1})$ به ترتیب از درجه n_a و $K-1$ هستند.

با ضرب $\Delta E_k(q^{-1})q^k$ در دو طرف معادله (7) و با استفاده از معادله (8) خواهیم داشت:

$$C(q^{-1})(y(t+k) - E_k(q^{-1})e(t+k)) = E_k(q^{-1})B(q^{-1})\Delta u(t+k-d-1) + F_k(q^{-1})y(t) \quad (9)$$

با توجه به اینکه ترم‌های نویز همگی در آینده هستند مقدار مورد انتظار برای سمت چپ معادله بالا به صورت زیر خواهد بود:

$$E[C(q^{-1})(y(t+k) - E_k(q^{-1})e(t+k))] = C(q^{-1})\hat{y}(t+k|t) \quad (10)$$

$\hat{y}(t+k|t)$ پیش‌بینی k پله جلوتر $y(t+k)$ در لحظه t است. از اینرو:

$$C(q^{-1})\hat{y}(t+k|t) = E_k(q^{-1})B(q^{-1})\Delta u(t+k-d-1) + F_k(q^{-1})y(t) \quad (11)$$

با فرض اینکه تمام صفرهای $C(q^{-1})$ داخل دایره واحد قرار دارند خواهیم داشت:

$$\begin{aligned} \hat{y}(t+k|t) &= \frac{\gamma(q^{-1})}{C(q^{-1})}y(t) + \frac{\delta(q^{-1})}{C(q^{-1})}\Delta u(t+k-d-1) \\ &\cong \sum_{i=0}^p \alpha_i y(t-i) + \sum_{i=0}^q \beta_i \Delta u(t+k-d-1-i) \end{aligned}$$

با انتخاب $k = d+1$ خواهیم داشت:

$$\hat{y}(t+k|t) \cong \sum_{i=0}^p \alpha_i y(t-i) + \sum_{i=0}^q \beta_i \Delta u(t-i) \quad (12)$$

$\hat{y}(t+k|t)$ پیش‌بین k پله جلوتر بهینه از مدل CARIMA در (7) است و $\sum_{j=1}^p \alpha_j q^{-j}$ و $\sum_{j=0}^q \beta_j q^{-j}$ به منظور رهیافت به توابع انتقال، به ترتیب، $\frac{\gamma(q^{-1})}{C(q^{-1})}$ و $\frac{\delta(q^{-1})}{C(q^{-1})}$ هستند.

اگر $Q_{ref}(t)$ بزرگ انتخاب شود بهره‌برداری از پهنای باند بیشتر و تاخیر نیز بیشتر خواهد بود و برعکس.

حال اگر فرض کنیم $Q_{ref}(t)$ برای یک عملکرد بهینه انتخاب شده است، عملکرد کنترلی سیستم در پاسخ به این مرجع می‌تواند به عنوان معیار کارایی عملکرد الگوریتم AQM قرار بگیرد. از اینرو هر چه پاسخ سیستم (اندازه بافر گلوگاه) به مقدار مرجع ($Q_{ref}(t)$) سریعتر، با فراجاهش کمتر، خطای حالت پایایی کمتر و نوسانات کمتر باشد، عملکرد کنترلی مطلوب‌تر است. در [۲] از دامنه نوسانات حالت پایایی خروجی و سرعت پاسخ‌دهی به سیگنال مرجع و در [۱۵] از مجذور میانگین مربعی سیگنال خطای خروجی برای بررسی کارایی الگوریتم کنترلی استفاده شده است. در این مقاله از میانگین قدرمطلق خطای خروجی $(E = \frac{1}{n} \sum_{t=1}^n |Q(t) - Q_{ref}(t)|)$ به منظور ارزیابی عملکرد الگوریتم کنترلی استفاده شده است. به این معنی که هر چه $E(\text{packets})$ کوچکتر باشد عملکرد سیستم مطلوب‌تر است.

به منظور تشریح الگوریتم کنترلی فوق از شبیه سازی با نرم‌افزار مطلب استفاده شده است. برای این منظور یک شبکه ساده به صورت شکل (۸) در نظر گرفته شده است. این شبکه شامل دو مسیر یاب R_0 و R_1 ، منبع TCP/Reno و مقصد TCP/Reno است. کنترل بروی مسیر یاب R_0 قرار دارد. ظرفیت لینک $C = 30Mbps = 3750 \text{ packets}$ است. طول هر بسته ۱۰۰۰ بایت بوده و ظرفیت گلوگاه ۸۰۰ بسته است. به منظور لحاظ شدن تاخیر منابع مختلف فرض می‌شود که تاخیر انتشار بصورت تصادفی در بازه $T_p \in [160ms, 240ms]$ تغییر کند لذا تاخیر حلقه در بازه $R(t) \in [160ms, 453.3ms]$ تغییر خواهد کرد.

$$\begin{aligned} \hat{y}(t) &= [\hat{\alpha}_0 y(t-k) + \hat{\alpha}_1 y(t-k-1) + \dots + \hat{\alpha}_p y(t-k-p) \\ &+ \hat{\beta}_0 \Delta u(t-k) + \hat{\beta}_1 \Delta u(t-k-1) + \dots + \hat{\beta}_q \Delta u(t-k-q)] \\ &= \hat{\theta}(t)^T \phi(t-k) \end{aligned} \quad (17)$$

که

$$\begin{aligned} \hat{\theta}(t) &= [\hat{\alpha}_1(t), \dots, \hat{\alpha}_p(t), \hat{\beta}_1(t), \dots, \hat{\beta}_q(t)]^T \\ \phi(t-k) &= [y(t-k), \dots, y(t-k-p), \Delta u(t-k) \\ &\dots, \Delta u(t-k-q)]^T \end{aligned}$$

برای حالت اولیه فرض می‌کنیم $\hat{\theta}(0) = 0$ است.

$$e(t) = y(t) - \hat{y}(t)$$

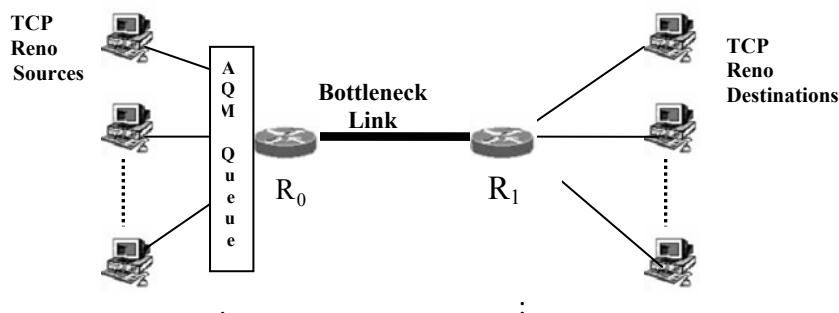
آنگاه:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + \frac{\mu}{a + \|\phi(t-k)\|^2} \phi(t-k) e(t) \quad (18)$$

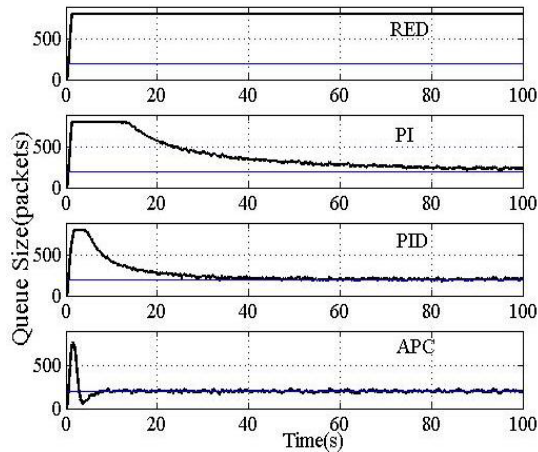
a یک مقدار مثبت اختیاری برای ممانعت از تقسیم بر صفر است. μ یک پارامتر طراحی است و همانطور که در [۷] آمده است برای پایداری الگوریتم NLMS باید بین ۰ و ۲ انتخاب شود.

باید توجه شود که در مد کنترل پیش‌بین تطبیقی لازم نیست پارامترهای مدل $(\gamma), (a_j, b_j, c_j)$ ، تخمین زده شوند فقط لازم است پارامترهای (α_j, β_j) در (۱۲) پیش‌بینی شوند.

از اهدافی که در AQM دنبال می‌شود کاهش تاخیر در انتقال داده‌ها و افزایش بهره‌برداری از پهنای باند موجود است و این در حالیست که این دو هدف با هم در تضادند. کاهش تاخیر موجب کاهش بهره‌برداری از پهنای باند و افزایش بهره‌برداری از پهنای باند موجب افزایش تاخیر می‌شود لذا باید بین آنها مصالحه‌ای برقرار کرد. یکی از فاکتورهایی که بر روی این دو هدف تاثیر دارد انتخاب مقدار مرجع برای طول بافر صف، $Q_{ref}(t)$ ، است.



میانگین قدرمطلق سیگنال خطا، میانگین تاخیر صف و بهره‌برداری از پهنای باند در جدول (۲) نشان داده شده است. همانطور که دیده می‌شود اگر چه بهره‌برداری از پهنای باند کنترلگرهای دیگر نسبت به APC بیشتر است ولیکن تاخیر بیشتری دارند اما با توجه به معیار بهینه که هر دو فاکتور تاخیر صف و بهره‌برداری از پهنای باند را در بر دارد و در بخش معیار کارایی شبکه توضیح داده شد از آنجایی که APC میانگین قدرمطلق خطای (E) کمتری دارد لذا عملکرد کنترلی مطلوب‌تری نیز خواهد داشت.



APC PID PI RED :

$$. N = 300 \quad Q_{ref} = 200$$

$$. N = 300 \quad Q_{ref} = 200$$

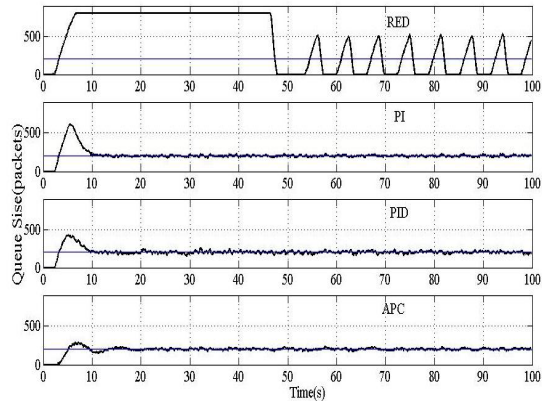
	APC	PID	PI	RED
E(packets)	13.7	38.2	105.7	574.6
Dq(msec)	53.4	61.8	80.7	206.4
U (%)	99.9	100	100	100

در اینجا فرض می‌کنیم که مقدار مرجع همان $Q_{ref} = 200$ است، تاخیر انتشار بصورت تصادفی در بازه $Tp \in (0.16 \text{ sec}, 0.24 \text{ sec})$ تغییر می‌کند و تعداد منابع ارسال کننده داده بصورت زیر است:

$$N(t) = 100 \times 1(t) + 50 \times 1(t - 15) - 20 \times 1(t - 25) + 40 \times 1(t - 40) + 20 \times 1(t - 50) - 30 \times 1(t - 75)$$

نتایج حاصل از شبیه سازی در شکل (۱۱) نشان داده شده است. همانطور که دیده می‌شود کنترلگر APC پاسخ سریع‌تری دارد و این در حالیست که بافر کنترلگرهای PI و RED دچار سرریز شده‌اند.

تعداد منابع ارسال کننده $N = 60$ و $Q_{ref} = 200$ است. نتایج شبیه‌سازی برای کنترلگرهای RED، PI، PID و ^{۱۷} APC در شکل (۹) نشان داده شده است. همانطور که از شکل پیداست کنترلگرهای PI و PID عملکرد خوبی در حوالی نقطه کار دارند.



شکل ۹: طول صف برای کنترلگرهای RED، PI، PID و APC

$$. N = 60 \quad Q_{ref} = 200$$

همانطور که در بخش معیار کارایی شبکه توضیح داده شد معیار مورد نظر در این مقاله میانگین قدرمطلق خطای خروجی (E)، است که در جدول (۱) نشان داده شده است. در این جدول به منظور مقایسه بیشتر میانگین تاخیر صف (Dq)، و بهره‌برداری از پهنای باند (U)، نیز آورده شده است.

$$. N = 60 \quad Q_{ref} = 200$$

	APC	PID	PI	RED
E(packets)	19.6	24.3	25.9	348.4
Dq(msec)	51.3	53.8	55.4	110.1
U (%)	98.7	98.8	98.8	100

با توجه به اینکه پارامترهای شبکه دائما در حال تغییر هستند لذا برای بررسی بیشتر الگوریتم‌ها کمی از نقطه کار طراحی کنترلگرهای RED، PI و PID فاصله می‌گیریم. برای این منظور $N = 300$ انتخاب کرده و سایر پارامترها را بدون تغییر در نظر می‌گیریم. نتایج شبیه‌سازی‌ها در شکل (۱۰) نشان داده شده است. همانطور که از شکل پیداست عملکرد کنترلگرهای RED، PI، PID با دور شدن از نقطه کار تضعیف شده است (بافر گلوگاه‌هایشان دچار سرریز شده و پاسخ آنها کندتر شده است) و این در حالی است که عملکرد کنترلگر APC چندان تغییر پیدا نکرده است.

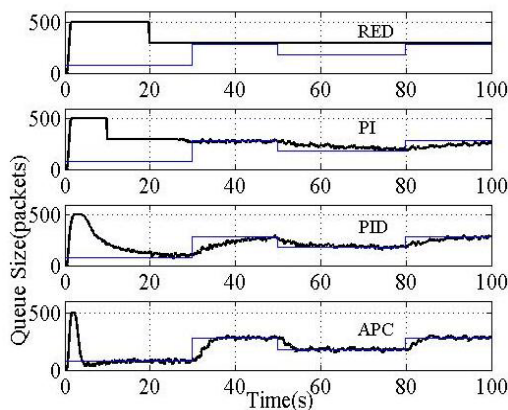
میانگین قدرمطلق سیگنال خطا در جدول (۴) نشان داده شده است همانطور که دیده می‌شود APC باز هم کمترین خطا را دارد.

$$N = 200$$

$$Q(t) = 80 \times 1(t) + 200 \times 1(t - 30) - 100 \times 1(t - 50) + 100 \times 1(t - 80)$$

	APC	PID	PI	RED
E(packet)	28.4	61.0	126.7	516.7
Dq(msec)	50.4	57.1	72.8	188.1
U (%)	99.7	99.7	100	100

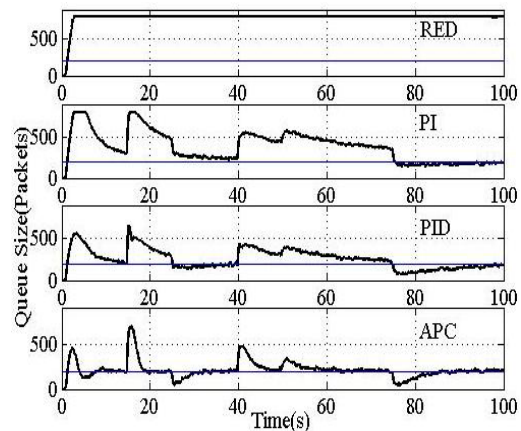
به منظور بررسی اثر تغییر ظرفیت بافر فرض می‌کنیم که در شبیه‌سازی قبلی ظرفیت گلوگاه به صورت $C(t) = 500 \times 1(t) - 200 \times 1(t - 20)$ تغییر کند نتایج شبیه‌سازی‌ها در شکل (۱۳) نشان داده شده است. همانطور که دیده می‌شود هر چهار کنترلگر دچار سرریز شده‌اند ولیکن APC کمترین سرریز، و سریعترین پاسخ را دارد. میانگین قدرمطلق سیگنال خطا در جدول (۵) نشان داده شده است همانطور که دیده می‌شود APC باز هم کمترین خطا را دارد.



	APC	PID	PI	RED
E(packet)	27.3	61.0	119.14	145.3
Dq(msec)	50.5	57.2	72.8	149
U (%)	99.8	99.9	100	100

به منظور بررسی اثر تعداد منابع ارسالی بر روی عملکرد سیستم از لحاظ میانگین تاخیر حلقه، میانگین تاخیر حلقه به ازای تغییرات تعداد منابع ارسال کننده در شکل (۱۴) رسم شده است. همانطور که دیده می‌شود با استفاده از کنترلگر APC، تاخیر حلقه حساسیت کمتری نسبت به تغییرات تعداد منابع ارسالی دارد.

میانگین قدرمطلق سیگنال خطا در جدول (۳) نشان داده شده است همانطور که دیده می‌شود APC باز هم کمترین خطا را دارد.



APC PID PI RED

$$N(t) = 100 \times 1(t) + 50 \times 1(t - 15) \quad Q_{ref} = 200$$

$$-20 \times 1(t - 25) + 40 \times 1(t - 40) + 20 \times 1(t - 50) - 30 \times 1(t - 75)$$

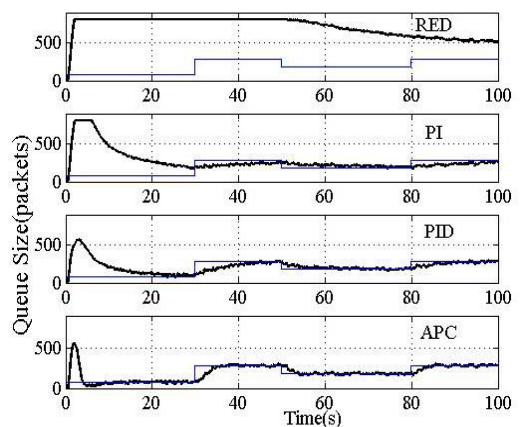
$$Q_{ref} = 200$$

$$N(t) = 100 \times 1(t) + 25 \times 1(t - 15) - 20 \times 1(t - 25)$$

$$+ 40 \times 1(t - 40) + 20 \times 1(t - 50) - 30 \times 1(t - 75)$$

	APC	PID	PI	RED
E(packet)	29.0	55.4	103.4	528.9
Dq(msec)	54.6	60.1	75.7	193.7
U (%)	58.7	58.6	58.7	96.5

به منظور بررسی عملکرد کنترلی نسبت به تغییرات سیگنال مرجع، $N = 200$ و $Q(t) = 80 \times 1(t) + 200 \times 1(t - 30) - 100 \times 1(t - 50) + 100 \times 1(t - 80)$ انتخاب می‌کنیم. نتایج شبیه‌سازی در شکل (۱۲) نشان داده شده است. همانطور که از شکل پیداست APC سریعترین پاسخ را داراست، در اینجا هم کنترلگرهای RED و PI دچار سرریز شده‌اند.



APC PID PI RED

$$Q(t) = 80 \times 1(t) + 200 \times 1(t - 30) \quad N = 200$$

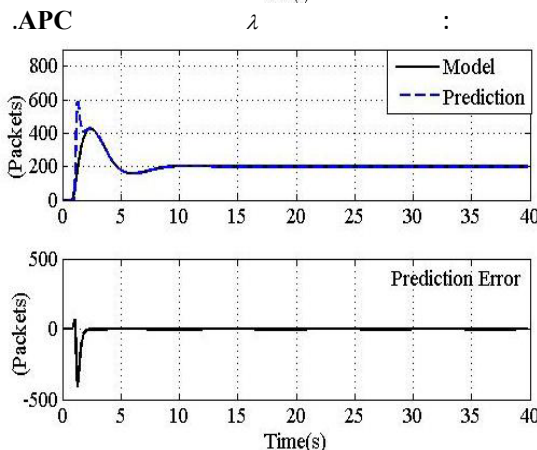
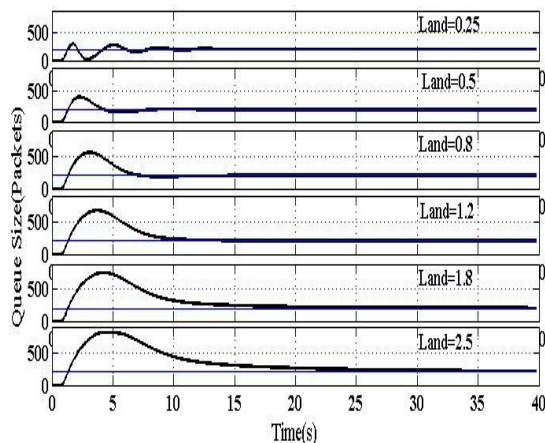
$$-100 \times 1(t - 50) + 100 \times 1(t - 80)$$

میانگین قدرمطلق سیگنال خطا در جدول (۶) نشان داده شده است همانطور که دیده می‌شود APC باز هم کمترین خطا را دارد.

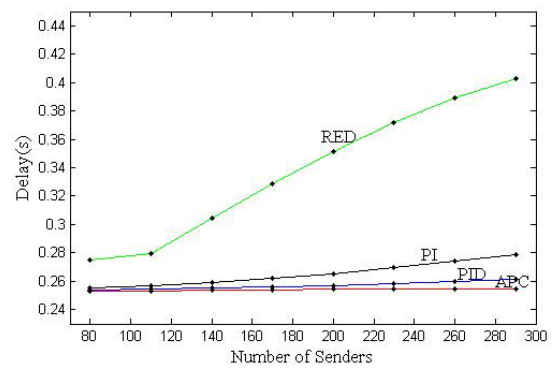
$$\lambda$$

در تابع هزینه (۱۳)، λ یک پارامتر طراحی است و باید در تعیین آن دقت لازم را به خرج داد. در اینجا به منظور تاثیر پارامتر λ در عملکرد کنترلی کنترلر APC فرض می‌کنیم که $N = 150$ و $Q_{ref} = 200$ باشد.

شکل (۱۶) عملکرد کنترلی الگوریتم APC را به ازای تغییرات λ نشان می‌دهد. همانطور که از شکل پیداست برای مقادیر کوچک λ سیستم نوسانی است در حالی که برای مقادیر بزرگ آن سیستم کند است و حتی دچار سرریز شده است.



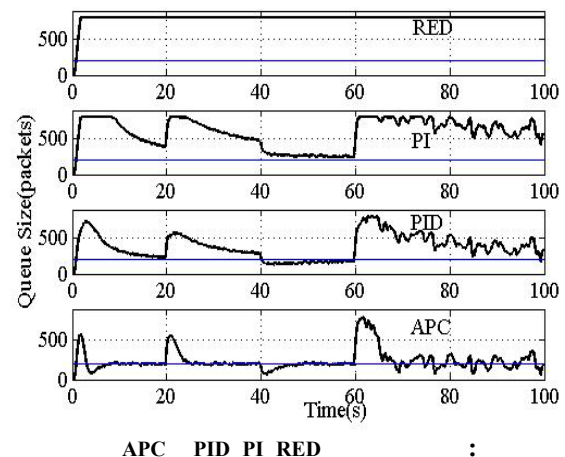
در شکل (۱۷) عملکرد سیستم و خطای مدل پیش‌بین به ازای $\lambda = 0.9$ نشان داده شده است. همانطور که از شکل پیداست مدل پیش‌بین جز در لحظات اولیه عملکرد خوبی دارد.



به منظور بررسی اثر اغتشاش فرض می‌کنیم که یک اغتشاش به صورت زیر به بافر گلوگاه وارد شود:

$$d(t) = 10 \times 1(t) + 30 \times 1(t - 30) - 10 \times 1(t - 50) + n \times 1(t - 80)$$

n یک سیگنال تصادفی با میانگین ۳۰، واریانس ۳۰ و زمان نمونه‌برداری ۰.۵ است. نتایج شبیه‌سازی در شکل (۱۵) نشان داده شده است. همانطور که از شکل پیداست الگوریتم APC پاسخ سریع‌تری در دفع اغتشاش دارد.



APC PID PI RED

APC PID PI RED

	APC	PID	PI	RED
E(packets)	70.0	194.9	381.9	592.4
Dq(msec)	62.4	100.0	154.3	210.5
U (%)	61.1	60.6	62.2	100

فاصله گرفتن از نقطه کار عملکردشان تضعیف می شود و این در حالیست که عملکرد APC چندان تغییر نکرده است و حساسیت کمتری نسبت به تغییرات پارامترهای شبکه دارد.

با توجه به اینکه پارامترهای شبکه دائما در حال تغییر می باشند لذا می توان گفت APC عملکرد بهتری از لحاظ مدیریت فعال صف نسبت به RED، PI و PID دارد.

در این مقاله از یک الگوریتم پیش بین تطبیقی به منظور کنترل بهینه جریان ترافیک در یک مسیریاب شبکه TCP که دارای دینامیک غیرخطی، تاخیر و مشخصات غیراستاتیکی است، استفاده شد. نتایج شبیه سازی ها نشان می دهد اگر چه الگوریتم های RED و PID عملکرد نسبتا خوبی در حوالی نقطه کار دارند ولی با

مراجع

- 1 - Floy, S. and Jacobson, V. (1993). "Random early detection gateways for congestion avoidance." *IEEE/ACM Trans. Networking*, Vol.1, PP.397-413.
- 2 - Holot, C. V. and Vishal Misra, (2002). "Analysis and design of controllers for AQM routers supporting TCP flows." *IEEE Trans. Automatic Control.*, Vol. 47, No. 6.
- 3 - Bor-Sen Chen and Yu-Shuang Yang, (2003). "Fuzzy predictive flow control of ATM network traffic." *IEEE Trans.*, Vol. 11, No. 4, PP.568-581.
- 4 - Jangannathan, S. and Jayasree, (2003). "Adaptive predictive congestion control of high-speed ATM networks." *IEEE Trans.*, Vol. 48, No.2, PP.129-139.
- 5 - Åström, K. J. (1995). "Oscillations in systems with relay feedback." *Adapt. Control Filtering Signal Processing, IMA Volumes Math. Appl.*, Vol. 74, PP. 1-25.
- 6 - Camach, E. F. and Bordon, (1999). *Model Predictive Control*, Springer.
- 7 - Haykin, S. (1996). *Adaptive Filter Theory, 2nd ed. Upper Saddle River, NJ:Prentice-Hall.*
- 8 - Christiansen, M., Jeffay, K. Ott, D. and Smith, F. (2000). "Tuning red for web traffic." *In Proc. ACM/SIGCOM.*
- 9 - May, M., Bonald, T. and Bolot, J. C. (2000). "Analytic evaluation of RED performance." *In Proc. IEEE INFOCOM.*
- 10 - Lin, D. and Morris, R. (1997). " Dynamics of random early detection." *In Proc. ACM/SIGCOMM.*
- 11 - Ott, T.J., Lakshman, T. V. and Wong, L.H. (1999). "SRED: Stabilized RED." *In Proc. IEEE INFOCOM.*
- 12 - Feng, W., Kandlur, D., Saha, D. and Shin, K. (1999). "Blue: A new class of active queue management algorithms." *Tech. Rep. UM CSE-TR=387-99.*
- 13 - Misra, V., Gong, W. B. and Towsley, D. (2000). "Fluid-based analysis of a network of AQM routers supporting TCP flows with an application to RED." *Presented at the Proc. ACM/SIGCOMM.*
- 14 - Veras, A. and Boda, M. (2000). "The chaotic nature of TCP congestion control." *In Proc. IEEE INFOCOM.*
- 15 - Ryu, S., Rump, C. and Qiao, C. (2003). "A predictive and robust active queue management for internet congestion control." *Proceedings of the Eighth IEEE International symposium on Computers and Communication.*

1 - Active Queue Management
2 - Voice Over IP
3 - Class of Service
4 - Flow Synchronization
5 - Restart
6 - Time-out
7 - Random Early Detection
8 - Packet Loss
9 - End-to-End

10 - Congestion
11 - Bursty Traffic
12 - Fluid-Flow Model
13 - Stochastic Differential Equation
14 - Model Predictive Control
15 - Controlled Autoregressive Integrated Moving Average
16 - Normalized Least Mean Square
17 - Adaptive Predictive Control
