ارائه یک الگوریتم بهبود یافته جهت حل مشکل ناسازگاری TCP-Vegas با اتصالات دیگر در شبکه‌های ناهمگن

شیرین اسدالهی[1]، فضل‌الله ادیب‌نیا[2]

۱.کارشناسی ارشد فناوری اطلاعات، دانشگاه یزد،kaktus6093@yahoo.com

۲. استادیار دانشکده برق و کامپیوتر، دانشگاه یزد

تاریخ دریافت: ۹۲/۸/۱۸ تاریخ پذیرش: ۹۳/۱۱/۱۴

چکیده

جهت اداره کردن ازدحام در یک شبکه کامپیوتری از الگوریتم‌های کنترل ازدحام استفاده می‌شود. مطالعات تاکنون، نشان می‌دهد که در شبکه‌های همگن، TCP-Vegas در بسیاری از موارد، بهتر از TCP-Reno عمل می‌کند. اما هنگامی‌که در اینترنت از هر دو الگوریتم Vegas و Reno با هم استفاده می‌شود، Reno بهتر از Vegas عمل می‌کند. به همین دلیل، هیچ انگیزه‌ای برای سیستم عامل‌ها وجود ندارد که از Vegas بعنوان پروتکل پیش‌فرض لایه انتقال استفاده کنند. در این مقاله، یک Vegas بهبود یافته پیشنهاد شده است که با Reno سازگار است و زمانی که با Reno رقابت می‌کند، می‌تواند یک سهم عادلانه از منابع شبکه را بدست آورد. برخلاف TCP-Vegas که از پارامترهای αوβ ثابت استفاده می‌کند، الگوریتم پیشنهادی، این پارامترها را با توجه به شرایط شبکه و به صورت پویا تعیین می‌کند. الگوریتم پیشنهادی با Vegas اصلی و سایر الگوریتم‌های بهبود یافته Vegas، نظیر Vegas-A و CODETCP مقایسه شد. نتایج شبیه‌سازی نشان می‌دهد که الگوریتم پیشنهادی در رقابت با Reno، نسبت به سایر الگوریتم‌های بهبود یافته، سهم منصفانه‌تری از پهنای باند را بدست می‌آورد و دارای توان عملیاتی بالاتری است. بعلاوه الگوریتم پیشنهادی خصوصیات خوب الگوریتم Vegas را نیز حفظ می‌کند.

کلید واژه

TCP-Vegas، TCP-Reno، توان عملیاتی، کنترل ازدحام، عدالت

مقدمه

TCP یک پروتکل اتصالگرای انتها به انتها[1] و قابل اطمینان است. امروزه اکثر ترافیک اینترنت به وسیله TCP منتقل می‌شود. بنابراین رفتار TCP با کارایی اینترنت عمومی بهم پیوند داده می‌شوند. بمنظور بهبود کارایی شبکه، انواع مختلفی از پروتکل‌های TCP پیشنهاد شده است. دو نوعی که امروزه استفاده می‌شود، یکی Reno[۱] است که به طور گسترده در اینترنت به کار گرفته می‌شود و دیگری Vegas[۲]است که ادعا می‌کند، توان عملیاتی آن بیش‌تر از Reno است. TCP-Vegas یک مکانیسم کنترل ازدحام مبتنی بر تاخیر است. برخلاف TCP-Reno که برای تنظیم اندازه پنجره‌اش وابسته به Packet Loss است، TCP-Vegasاز تاخیر صف به منظور اجتناب از ازدحام استفاده می‌کند. Vegas می‌تواند ازدحام شبکه را در مرحله اولیه تشخیص دهد و به طور متناوب از PacketLoss جلوگیری کند. در حالی که در Reno، معمولا Packet Loss اتفاق می‌افتد. Vegas در مقایسه با Reno از نظر بهره‌مندی شبکه[2] [۲]، پایداری[3] [۳و۴]، عدالت[4] [۳و۴]، توان عملیاتی[5] و PacketLoss [۲]در محیط‌های مشابه (مقاربت کننده با هم جنس خود) بهتر عمل می‌کند. براساس مطالعات انجام شده قبلی، هنگامی که اتصالات Reno و Vegas در یک محیط شبکه با هم رقابت می‌کنند، Reno در مقایسه با Vegasسهم بیش‌تری از پهنای باند را بدست می‌آورد [۴، ۵ و۶] به دلیل این ناسازگاری، Vegas به طور وسیع به کار گرفته نمی‌شود. الگوریتم‌های زیادی [۷، ۸، ۱۳ و ۱۴]جهت بهبود کارایی Vegas ارائه شده است. در این مقاله نیز، یک مکانیسم جدید از Vegas پیشنهاد شده است که در مقایسه با سایر الگوریتم‌های بهبود یافته دارای عملکرد بهتری است. هنگامی که الگوریتم پیشنهادی، وقوع ازدحام را احساس می‌کند، به جای اینکه نرخ جریانش را کاهش دهد، پارامترهای α و β را افزایش می‌دهد. این باعث می‌شود که الگوریتم پیشنهادی بیش‌تر شبیه Reno رفتار کند. بنابراین الگوریتم پیشنهادی در رقابت با Reno طوری عمل می‌کند که کارایی آن کاهش نمی‌یابد. در ادامه این مقاله، در بخش دوم،

1 . End-to-End
2 . Network Utilization
3 . Stability
4 . Fairness
5 . Throughput

کارهای مرتبط معرفی می‌شود. در بخش سوم، الگوریتم پیشنهادی توضیح داده می‌شود. در بخش چهارم، نتایج شبیه‌سازی در ns2 نشان داده می‌شود. در بخش پنجم هم نتیجه‌گیری ارائه می‌گردد.

کارهای مرتبط

در این بخش، الگوریتم های کنترل ازدحام TCP Renoو TCP-Vegas، دلیل ناسازگاری این دو الگوریتم در رقابت با هم و الگوریتم‌های بهبود یافته Vegas-A[۷] و CODE TCP[۸] را که تلاش می‌کنند تا مشکل ناسازگاری Vegas با Reno را حل کنند، توضیح داده خواهد شد.

الگوریتم TCP-Reno

رویه کنترل ازدحام Reno، به سه فاز شروع آهسته[۶] ، اجتناب از ازدحام[۷] و انتقال مجدد سریع و بازیابی سریع[۸] تقسیم می‌شود[۱]. این الگوریتم به ازای گم شدن هر بسته به فاز شروع آهسته نمی‌رود بلکه با دریافت سه اعلام وصول تکراری، فرستنده بسته گم شده را دوباره می‌فرستد. اندازه پنجره اطلاعات را نصف می‌کند و به فاز بازیابی سریع می‌رود. هنگامی که فرستنده اولین اعلام وصول غیرتکراری را دریافت کرد، از فاز بازیابی سریع خارج شده و به فاز اجتناب از ازدحام می‌رود. استفاده از بازیابی سریع در Reno، تنها در صورتی بهینه است که فقط یک بسته از فرستنده گم شود. در صورتی که چند بسته از یک پنجره گم شود، چون این الگوریتم تا تمام شدن مهلت هر بسته صبر می‌کند، به این ترتیب از پهنای باند موجود استفاده کافی نخواهد شد.

الگوریتم TCP-Vegas

TCP-Vegas[۲]از یک رویه پیچیده‌تری برای تخمین پهنای باند استفاده می‌کند. TCP-Vegas تلاش می‌کند، از وقوع ازدحام جلوگیری کند تا اینکه بخواهد بعد از وقوع ازدحام، از ادامه آن پیشگیری کند. Vegas از RTT اندازه‌گیری شده برای محاسبه تعداد بسته‌های داده‌ای که یک مبدأ می‌تواند بفرستد، استفاده می‌کند. همچنین یک الگوریتم شروع آهسته اصلاح شده، که از ازدحام شبکه جلوگیری می‌کند را پیشنهاد می‌کند. الگوریتم TCP-Vegas از توان عملیاتی واقعی و توان عملیاتی مورد انتظار برای ارسال داده استفاده می‌کند. توان عملیاتی واقعی و توان عملیاتی مورد انتظار طبق روابط (۱) و (۲) محاسبه می‌شود[۷].

$$\text{Actual Throughput} = \frac{cwnd(t)}{RTT} \quad (۱)$$

$$\text{Expected Throughput} = \frac{cwnd(t)}{BaseRTT} \quad (۲)$$

cwnd(t) اندازه پنجره ازدحام جاری، RTT زمان رفت و برگشت واقعی و Base RTT حداقل RTT اتصال است. Δ، اختلاف بین توان عملیاتی واقعی و توان عملیاتی مورد انتظار است وW بیانگر طول پنجره ازدحام است. در فاز اجتناب از ازدحام، طول پنجره ازدحام طبق رابطه (۳) تغییر می‌کند[۷].

$$w = \begin{cases} w+1 & if \Delta < \alpha \\ w-1 & if \Delta > \beta \\ w & if \alpha \le \Delta \le \beta \end{cases} \quad (۳)$$

Vegas از مقادیر ثابت ۱ و ۳ به ترتیب برای پارامترهای α و β استفاده می‌کند و اندازه پنجره ازدحام را طوری تنظیم می‌کند که تعداد کمی بسته در مسیریاب بافر شود. در این الگوریتم، میانگین تعداد بسته‌های بافر شده در مسیریاب بین α و β نگه داشته می‌شود. در این الگوریتم، زمانی که توان عملیاتی واقعی به حداکثر توان عملیاتی مورد انتظار نزدیک می‌شود، ممکن است از فضای بافر مسیریاب‌های میانی بطور مؤثر استفاده نشود، در نتیجه نرخ جریان باید افزایش یابد. از طرف دیگر زمانی که توان عملیاتی واقعی خیلی کمتر از توان عملیاتی مورد انتظار است، احتمال دارد که شبکه دچار ازدحام شده باشد، در نتیجه اتصال باید نرخ جریان را کاهش دهد.

ناسازگاری TCP-Vegas وTCP-Reno

بعضی از مطالعات انجام شده نشان داده است که TCP-Vegas در بسیاری از موارد از TCP-Reno جلو می‌افتد[۲]. اما زمانی که یک اتصال Vegas با اتصالات دیگری از Reno رقابت می‌کند، Vegas در این رقابت سهم منصفانه‌ای از پهنای باند را بدست نمی‌آورد. دلیل این امر آن است که Reno به طور متوسط بسته‌های خیلی بیش‌تری را در بافر نگه‌داری می‌کند. بنابراین پهنای باند بزرگ‌تری را بدست می‌آورد.

رویه اجتناب از ازدحام Reno مهاجم است بدلیل اینکه فضای کوچکی از بافر را برای اتصالات دیگر می‌گذارد. در حالی که Vegas محافظه‌کار است و تلاش می‌کند تا فضای کمی از بافر را اشغال کند به همین دلیل، با وجود خصوصیات مطلوب زیادی که Vegas دارد اما در شبکه‌های ناهمگن و در رقابت با Reno کاربرد وسیعی ندارد. چندین رویه برای کاهش این ناسازگاری پیشنهاد شده است. این رویه‌ها به دو نوع تقسیم می‌شوند. نوع اول، αو β را با مقادیر بهینه بر اساس خصوصیات محیط شبکه مانند اندازه بافر در مسیریاب و تعداد اتصالاتی که از مسیریاب عبور می‌کنند [۵، ۶، ۹، ۱۰ و ۱۱]، مقداردهی می‌کند. اگرچه این رویه‌ها، بدلیل وجود برخی از مسائل، نمی‌تواند به طور وسیع در اینترنت بکار گرفته

6. Slow Start
7. Congestion Avoidance
8. Fast Retransmission and Fast Recovery

شود. رویه‌های نوع دوم تلاش می‌کنند تا مشکل ناسازگاری را با تنظیم α و β به طور پویا و براساس بازخورد شبکه حل نمایند [۷ و ۱۲].

الگوریتم TCP Vegas-A

به دلیل ناسازگار بودن Vegas با اتصالات دیگر در شبکه‌های ناهمگن، الگوریتم‌های دیگری جهت بهبود کارایی Vegas و سازگارتر کردن آن با اتصالات دیگر ارائه شده است. یکی از این الگوریتم‌های بهبود یافته، Vegas-A است. ایده اصلی Vegas-A [۷] تعیین مقادیرα و β به طور پویا و بر اساس بازخورد شبکه است. البته در آغاز یک اتصال، پارامترهایα و β به ترتیب با مقادیر پیش فرض ۱و۳ مقداردهی می‌شوند و بعد در طول الگوریتم متناسب با شرایط شبکه تغییر می‌کنند. عملکرد Vegas-A در شبه برنامه زیر آورده شده است:

```
if (α< Δ <β) {
   if (Th(t)> Th(t-rtt)) {
   Cwnd=cwnd+1
  α= α+1, β= β+1
   }
 else if (Th(t)<= Th(t-rtt))  {
   No update of cwnd, α, β
  }
 }
elseif (Δ<α) {
 if (α>1 and Th(t)>Th(t-rtt)) {
    Cwnd=cwnd+1
}
elseif (α>1 and Th(t)<Th(t-rtt)) {
cwnd=cwnd -1, α=α- 1, β=β- 1
}
elseif (α==1) {
 Cwnd=cwnd+1
}
elseif (Δ>β) {
if (α>1) {α=α- 1, β=β- 1}
cwnd=cwnd -1
}
else {
no update of cwnd, α, β
    }
```

در الگوریتم فوق، $Th_{(t)}$ نرخ توان عملیاتی واقعی در زمان t و $Th_{(t-rtt)}$ نرخ توان عملیاتی واقعی اندازه‌گیری شده در یک RTT قبل از t است. این الگوریتم، در صورتی که احتمال وقوع ازدحام در شبکه وجود داشته باشد($\Delta>\beta$) و یا توان عملیاتی اتصال کاهش یابد ($Th_{(t)}<Th_{(t-rtt)}$)، نرخ جریان را کاهش می‌دهد و در صورتی که از فضای بافر مسیریاب‌های میانی به طور موثر استفاده نشود ($\Delta<\alpha$) و یا توان عملیاتی اتصال افزایش یابد ($Th_{(t)}> Th_{(t-rtt)}$) نرخ جریان را افزایش می‌دهد.

الگوریتم CODE TCP

الگوریتم CODE TCP[۷]، از دیگر الگوریتم‌های بهبود یافته Vegasدر شبکه‌های ناهمگن است که جهت بهبود کارایی Vegas و سازگارتر کردن آن با اتصالات دیگر ارائه شده است. ایده این الگوریتم، از الگوریتم بهبود یافته TCP NewVegas[۱۲]گرفته شده است. در الگوریتم‌های Vegasو Vegas-Aدر آغاز یک اتصال، پارامترهایα و β به ترتیب با مقادیر پیش فرض ۱و۳ مقداردهی می‌شوند در حالی که در الگوریتم CODE TCP این دو پارامتر در عامل ε ضرب می‌شوند که بر اساس محاسبات انجام شده، بهترین مقدار برای ε، ۰٫۶۸ بدست آمده است. در الگوریتم Vegasپارامترهای α و β همواره ثابت می‌مانند در حالی که در الگوریتم‌های Vegas-A و CODE TCP، این پارامترها در طول الگوریتم متناسب با شرایط شبکه تغییر می‌کنند.

عملکرد CODE در شبه برنامه زیر آورده شده است:

```
if (receive_dupack)
if (loss_event_is_true)
fast_retransmission_and_recovery
α=ε × α, β=ε × β
else {
if(Δ>β)
w=w -1
elseif(Δ<α)
w=w+1
else
w=w
if(RTT > RTTold and w ≤ wold) {
α= α+2
β= β+2}
elseif (RTT ≤ RTTold and α > α0) {
α= α-1
β= β-1}
}
```

در شبه برنامه فوق، RTT تاخیر رفت و برگشت بسته و RTT_{old} همان RTT قبلی است. w اندازه پنجره ازدحام جاری و w_{old} اندازه پنجره ازدحام قبلی است.

الگوریتم پیشنهادی

الگوریتم پیشنهادی جهت بهبود کارایی Vegasو سازگارتر کردن آن با اتصالات دیگر در شبکه‌های ناهمگن ارائه شده است. ایده الگوریتم پیشنهادی با مطالعه الگوریتم‌های بهبود یافته TCP-Vegas در [۷ و ۸] شکل گرفت. بر خلاف Vegasکه از پارامترهای ثابت α و β استفاده می‌کند، در الگوریتم‌های Vegas-A، CODE TCPو پیشنهادی این پارامترها در طول الگوریتم، به طور پویا و متناسب با شرایط شبکه تغییر می‌کنند. الگوریتم‌های Vegasو Vegas-Aدر آغاز اتصال، از مقادیر پیش فرض α و β استفاده می‌کنند، در حالی که، در الگوریتم CODE TCP، این پارامترها در ۰/۶۸ و در الگوریتم پیشنهادی در ۲ ضرب می‌شوند. شبه کد الگوریتم پیشنهادی در فاز اجتناب از ازدحام به شکل زیر است:

```
if (receive_dupack)
if (loss_event_is_true)
fast_retransmission_and_recovery
α=2×α, β=2×β
else {
if(Δ>β) {
if(α>1) {
α=α-1, β=β-1
}
if((Δ-β)<cwnd) {
cwnd=cwnd-(Δ-β)
}
else {
cwnd =2
}
} elseif(Δ<α){
cwnd= cwnd + 1/cwnd
if(v_actual(t) > v_actual(t-rtt)){
α=α+2, β=β+2
}
elseif (α>1 and v_actual(t) <= v_actual(t-rtt)){
α=α-1, β=β-1
}
}
elseif(α<Δ<β){
no update of cwnd, α, β
}
}
```

در الگوریتم فوق $v_actual_{(t)}$توان عملیاتی واقعی در زمان t و$v_actual_{(t-rtt)}$ توان عملیاتی اندازه‌گیری شده در یک RTT قبل از t است. cwnd هم طول پنجره ازدحام است. در این الگوریتم، هنگامی که زمان منقضی می‌شود (timeout) یا سه عدد ACK تکراری توسط مبدا دریافت می‌شود، بعد از انتقال مجدد سریع و بازیابی سریع، پارامترهای α و β دو برابر شده و الگوریتم دوباره از فاز اجتناب از ازدحام شروع می‌کند.

در ادامه به توضیح فاز اجتناب از ازدحام الگوریتم پیشنهادی پرداخته خواهد شد:

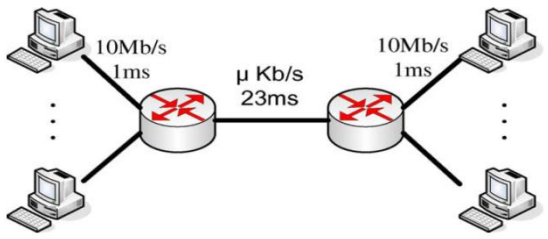
- هنگامی که Δ بزرگتر از β است، یعنی هنگامی که توان عملیاتی واقعی خیلی کمتر از توان عملیاتی مورد انتظار است، احتمال دارد که شبکه دچار ازدحام شده باشد، در نتیجه اتصال باید نرخ جریان را کاهش دهد. از طرفی چون ممکن است تخمین مبدا از α و β اشتباه باشد، بنابراین اگر α بزرگتر از یک باشد، مقادیر α و β نیز کاهش می‌یابند.
- هنگامی که Δکوچکتر از α است، یعنی هنگامی که توان عملیاتی واقعی به حداکثر توان عملیاتی مورد انتظار نزدیک می‌شود، ممکن است از فضای بافرهای مسیریاب‌های میانی بطور مؤثر استفاده نشود، در نتیجه اتصال باید نرخ جریان را افزایش دهد. در این حالت، چنانچه توان عملیاتی افزایش یابد یعنی توان عملیاتی در زمان t بزرگتر از توان عملیاتی در یک RTT قبل از t باشد، مقادیر کوچک α و β جلوگیری می‌کند که اتصال از پهنای باند در دسترس شبکه استفاده کند. بنابراین پارامترهای α و β افزایش داده می‌شود تا به رشد پنجره ازدحام کمک کند. چنانچه توان عملیاتی کاهش یابد یعنی توان عملیاتی در زمان t کوچکتر از توان عملیاتی در یک RTT قبل از t باشد، مقادیر بزرگ α و β می‌تواند باعث ایجاد ازدحام در شبکه شود. بنابراین پارامترهای α و β کاهش داده می‌شود.
- هنگامی که Δ بین α و β است، جهت جلوگیری از ازدحام مقادیر cwndو α و β بدون تغییر باقی می‌مانند.

تغییرات صورت گرفته در الگوریتم پیشنهادی، تنها محدود به مکانیسم اجتناب از ازدحام می‌شود. هدف از ایجاد این تغییرات، سازگارتر کردن Vegas با Reno در شبکه‌های ناهمگن است.

شبیه‌سازی

در این بخش نتایج شبیه‌سازی در محیط ns2 [۱۵]نشان داده شده است. توپولوژی استفاده شده در همه شبیه‌سازی‌ها مطابق شکل(۱) است که در آن یک اتصال،Reno و اتصال دیگر Vegas یا Vegas-A یا CODE-TCP یا الگوریتم پیشنهادی است. در همه شبیه‌سازی‌ها اندازه بسته TCP، ۱۰۰۰ بایت است و فرض می‌شود که پنجره ازدحام به وسیله گیرنده محدود نباشد. در غیر این صورت فرض می‌کنیم که مبدا همیشه داده‌ای برای ارسال دارد. الگوریتم مسیریاب هم Drop Tailاست. توان عملیاتی در همه شبیه‌سازی‌ها طبق رابطه (۴) محاسبه می‌شود:

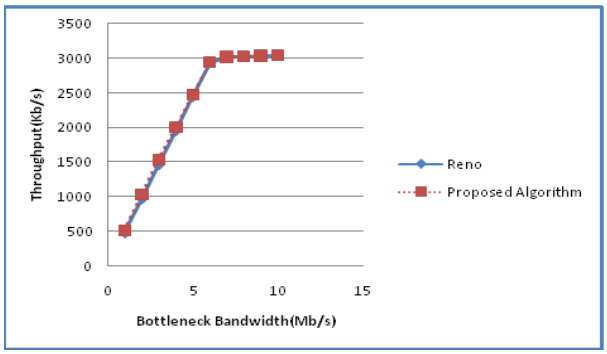
$$\text{Throughput}=\frac{(numberofreceivedpackets)\times(packetsize)\times 8}{totaltime(\text{sec})\times 1000} \quad (۴)$$



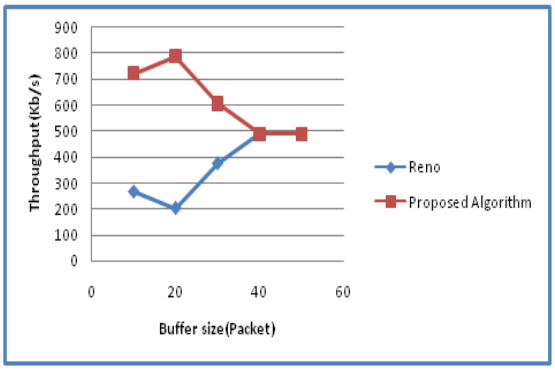
شکل ۱. توپولوژی شبکه

پیوند گلوگاه با پهنای باندهای متفاوت

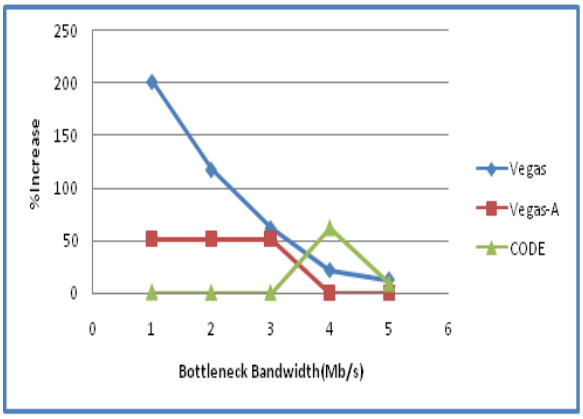
در این قسمت، یک اتصال Reno با یک اتصال Vegas، Vegas-A، CODETCP و یا الگوریتم پیشنهادی رقابت می‌کند. در این شبیه‌سازی، اتصال Reno در ثانیه ۱۰ و اتصال دیگر در ثانیه صفر شروع می‌شود. اندازه بافر ۱۰۰ بسته است. میانگین توان عملیاتی هر اتصال از ثانیه ۱۰ تا ۲۰۰ محاسبه شده است.



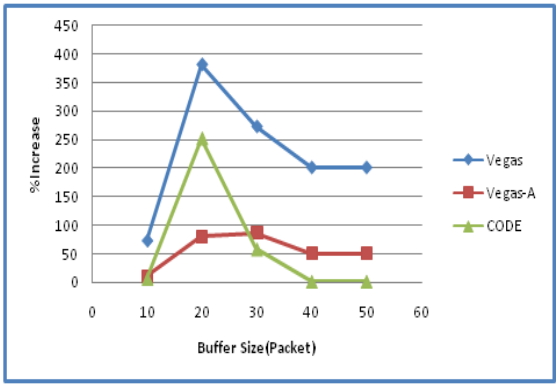
شکل۲. توان عملیاتی Reno و الگوریتم پیشنهادی با پهنای باند متفاوت



شکل۴. توان عملیاتی Reno و الگوریتم پیشنهادی با اندازه‌های بافر متفاوت



شکل۳. میزان بهبود الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌ها



شکل۵. میزان بهبود الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌ها

- در شکل(۲) مشاهده می‌شود، هنگامی که Reno با الگوریتم پیشنهادی رقابت می‌کند، همواره توان عملیاتی الگوریتم پیشنهادی بیش‌تر و نزدیک به توان عملیاتی Reno است. در واقع کارایی الگوریتم پیشنهادی در رقابت با Reno کاهش نمی‌یابد و سهم منصفانه‌ای از پهنای باند را دریافت می‌کند. در شکل (۳) مشاهده می‌شود، الگوریتم پیشنهادی در مقایسه با الگوریتم‌های Vegas، Vegas-A و CODE دارای توان عملیاتی بالاتری است که نشان‌دهنده بهبود کامل الگوریتم پیشنهادی است.

پیوند گلوگاه با اندازه‌های بافر متفاوت

در این قسمت، یک اتصال Reno با یک اتصال Vegas، Vegas-A، CODETCP و یا الگوریتم پیشنهادی رقابت می‌کند. اتصال Reno در ثانیه ۱۰ و اتصال دیگری در ثانیه صفر شروع می‌شود. پهنای باند پیوند گلوگاه هم ۱ مگابیت بر ثانیه است. میانگین توان عملیاتی هر اتصال از ثانیه ۱۰ تا ۲۰۰ محاسبه شده است.

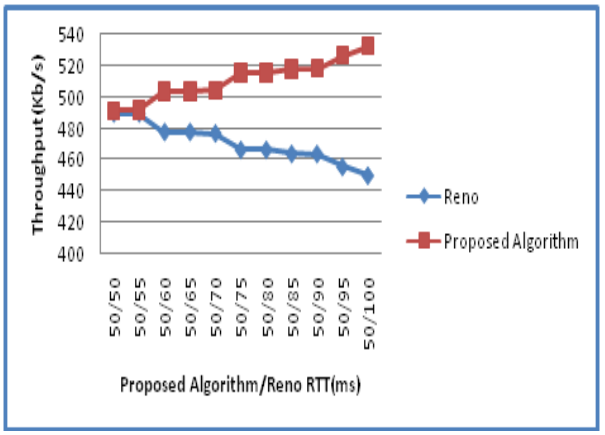
در شکل (۴) مشاهده می‌شود:

- هنگامی که Reno با الگوریتم پیشنهادی رقابت می‌کند، همواره توان عملیاتی الگوریتم پیشنهادی بیش‌تر و نزدیک به توان عملیاتی Reno است. در واقع کارایی الگوریتم پیشنهادی در رقابت با Reno کاهش نمی‌یابد و سهم منصفانه‌ای از پهنای باند را دریافت می‌کند.
- هنگامی که اندازه بافر کوچک است، چون اتصال‌ها برای بدست آوردن فضای کافی از بافر با هم رقابت می‌کنند، هر اتصالی که بتواند فضای بیش‌تری بدست آورد، توان عملیاتی بالاتری خواهد داشت. ولی هنگامی که اندازه بافر بزرگ است، چون دو اتصال می‌توانند فضای مورد نیاز خود را بدست آورند، توان عملیاتی ثابت می‌شود و اتصال‌ها به حالت پایدار می‌رسند.
- بین نمودارها تقارن وجود دارد چون اندازه بافر در محاسبه نرخ جریان واقعی تاثیر ندارد. به همین دلیل است که مجموع توان عملیاتی دو اتصال در اندازه های بافر متفاوت ثابت است.

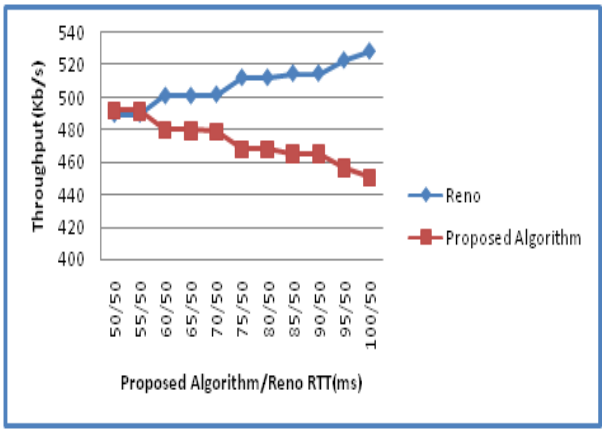
در شکل (۵) مشاهده می‌شود، الگوریتم پیشنهادی در مقایسه با الگوریتم‌های Vegas، Vegas-A و CODE دارای توان عملیاتی بالاتری است که نشان‌دهنده بهبود کامل الگوریتم پیشنهادی است.

اتصالات با RTTهای متفاوت

در این قسمت، یک اتصال Reno با یک اتصال Vegas، Vegas-A، CODETCP و یا الگوریتم پیشنهادی رقابت می‌کند. حداقل تاخیر رفت و برگشت برای یک اتصال ثابت و برای اتصال دیگر متغیر است. اتصال Reno در ثانیه ۱۰ شروع می‌شود و اتصال دیگری در ثانیه صفر شروع می‌شود. پهنای باند پیوند گلوگاه ۱ مگابیت بر ثانیه و اندازه بافر ۱۰۰ بسته است. میانگین توان عملیاتی هر اتصال از ثانیه ۱۰ تا ۲۰۰ محاسبه شده است.

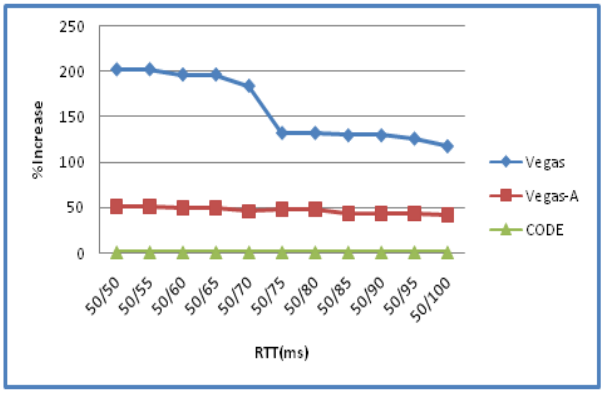


الف.Reno با RTT بزرگتر

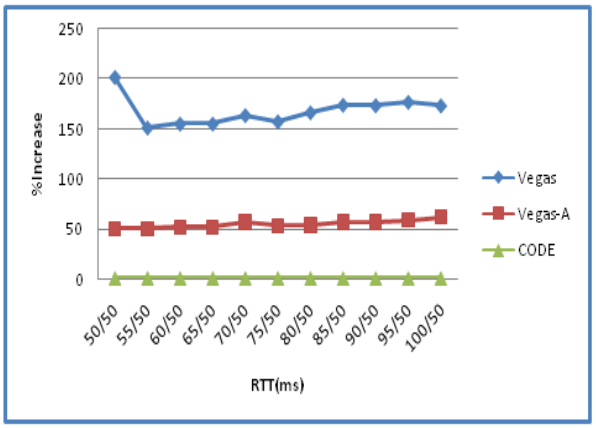


ب.الگوریتم پیشنهادی با RTT بزرگتر

شکل۶. توان عملیاتی دو اتصال Reno و الگوریتم پیشنهادی با RTT متفاوت



الف.Reno با RTT بزرگتر



ب.Reno با RTT کوچکتر

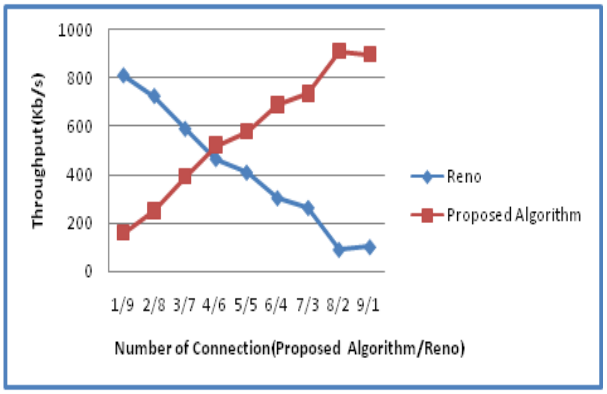
شکل۷. میزان بهبود الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌ها

در شکل‌های (۶)-الف و ب مشاهده می‌شود، هنگامی که الگوریتم پیشنهادی با Reno رقابت می‌کند، اتصالی که RTT کوچکتری دارد، دارای توان عملیاتی بیش‌تری است. چون اتصالی که RTT کمتری دارد، دارای تاخیر کمتر و سرعت بیش‌تری است و می‌تواند پهنای باند بیش‌تری را اشغال کند.

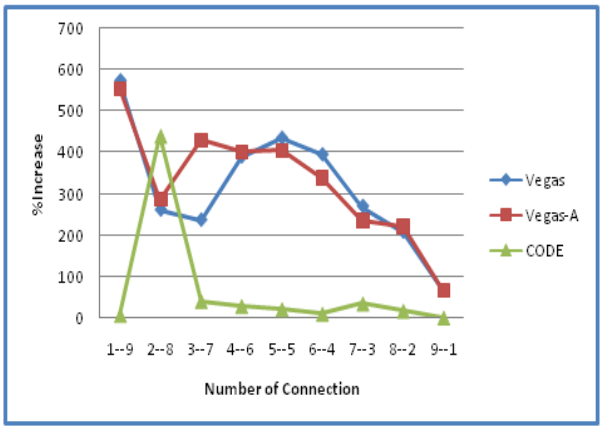
در شکل‌های (۷)-الف و ب مشاهده می‌شود، الگوریتم پیشنهادی در مقایسه با الگوریتم‌های Vegas، Vegas-A و CODE دارای توان عملیاتی بالاتری است که نشان‌دهنده بهبود کامل الگوریتم پیشنهادی است.

نسبت متفاوت تعداد اتصالات

در این شبیه‌سازی، ۱۰ اتصال یک گلوگاه را به اشتراک می‌گذارند. بعضی از مبدأها Reno و بعضی دیگر Vegas یا Vegas-AیاCODE یا الگوریتم پیشنهادی هستند. اتصالات Reno در ثانیه ۱۰ و اتصالات دیگر در ثانیه ۰ شروع می‌شود. پهنای باند لینک گلوگاه ۱ مگابیت بر ثانیه و اندازه بافر هم ۱۰۰ بسته است. میانگین توان عملیاتی هر نوع اتصال از ثانیه ۱۰ تا ۲۰۰ محاسبه شده است.



شکل۸. توان عملیاتی الگوریتم پیشنهادی و Renoبا تعداد اتصالات متفاوت



شکل۹. میزان بهبود الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌ها

در شکل (۸) مشاهده می‌شود، هنگامی که الگوریتم پیشنهادی با Reno رقابت می‌کند، الگوریتمی که تعداد اتصالاتش بیش‌تر است، دارای توان عملیاتی بالاتری است. در برخی از نقاط هم، الگوریتم پیشنهادی با این‌که تعداد اتصالات کمتری دارد ولی دارای توان عملیاتی بالاتری است.

در شکل (۹) مشاهده می‌شود، الگوریتم پیشنهادی در مقایسه با الگوریتم‌های Vegas، Vegas-Aو CODE دارای توان عملیاتی بالاتری است که نشان‌دهنده بهبود کامل الگوریتم پیشنهادی است.

عدالت در سناریوهای همگن

در این بخش، به وسیله سناریوهای شبیه‌سازی شده بررسی می‌شود، زمانی که فقط یک نوع اتصال در شبکه وجود دارد، آیا الگوریتم پیشنهادی، یک عدالت خوب را مشابه با Vegas حفظ می‌کند یا نه؟ بمنظور بررسی عدالت میان اتصالات، از شاخص عدالت[9] پیشنهاد شده در [۱۶]استفاده می‌شود. اگر یک مجموعه از توان‌های عملیاتی(x_1، x_2، ... و x_n) را داشته باشیم، آن‌گاه شاخص عدالت این مجموعه به صورت زیر تعریف می‌شود:

$$F = \frac{(\sum_{i=1}^{n} xi)^2}{n.\sum_{i=1}^{n} xi^2} \qquad (۵)$$

شاخص عدالت عددی بین ۰ و ۱ است. اگر توان‌های عملیاتی همه اتصالات یکسان باشد، شاخص عدالت ۱ خواهد بود. توپولوژی استفاده شده در این بخش، همان توپولوژی شکل(۱) است. در این شبیه‌سازی، ۱۰ اتصال هم نوع (Vegas یا الگوریتم پیشنهادی) وجود دارد که همزمان شروع می‌شوند و یک گلوگاه را به اشتراک می‌گذارند. پهنای باند گلوگاه متغیر است و اندازه بافر هم ۱۰۰ بسته است.

9 . Fairness Index

براساس نتایج شبیه‌سازی نشان داده شده در جدول(۱)، مشاهده می‌شود که الگوریتم پیشنهادی، خصوصیت عدالت را حفظ می‌کند و در یک سناریو همگن (مقاربت کننده با هم‌جنس خود) مشابه Vegas رفتار می‌کند.

جدول۱. مقدار شاخص عدالت ۱۰ اتصال در حال رقابت با مقادیر متغیر برای پهنای باند گلوگاه

شاخص عدالت الگوریتم پیشنهادی	شاخص عدالت Vegas	پهنای باند گلوگاه(Mb/s)
۰/۹۹	۰/۹۸	۵
۰/۹۷	۰/۹۴	۱۰
۰/۹۹	۰/۹۶	۱۵
۰/۹۹	۰/۹۲	۲۰
۰/۹۹	۰/۹۸	۲۵

نتیجه‌گیری

هنگامی که دو اتصال Vegas و Reno باهم رقابت می‌کنند، به دلیل ناسازگار بودن این دو الگوریتم با هم، Reno بهتر از Vegas عمل می‌کند. به همین دلیل، هیچ انگیزه‌ای برای سیستم عامل‌ها وجود ندارد که از Vegas بعنوان پروتکل پیش‌فرض لایه انتقال استفاده کنند.

برای حل این مشکل، در این مقاله، یک الگوریتم جدید جهت بهبود الگوریتم Vegas پیشنهاد شده است.

الگوریتم پیشنهادی دارای ویژگی‌های زیر است:

۱. سازگاری با الگوریتم کنترل ازدحام Reno

هنگامی که الگوریتم پیشنهادی، وقوع ازدحام را احساس می‌کند، به جای اینکه نرخ جریانش را کاهش دهد، پارامترهای α و β را افزایش می‌دهد. این باعث می‌شود که الگوریتم پیشنهادی بیش‌تر شبیه Reno رفتار کند. بنابراین الگوریتم پیشنهادی در رقابت با Reno طوری عمل می‌کند که کارایی آن کاهش نمی‌یابد. در واقع الگوریتم پیشنهادی می‌تواند یک سهم عادلانه از منابع شبکه را بدست آورد. بنابراین می‌توان نتیجه گرفت که الگوریتم پیشنهادی با Reno سازگار است.

۲. افزایش توان عملیاتی

الگوریتم پیشنهادی با Vegasاصلی و سایر الگوریتم‌های بهبود یافته Vegas، نظیر Vegas-A و CODETCP مقایسه شد. بر طبق نتایج شبیه‌سازی، الگوریتم پیشنهادی در رقابت با Reno، نسبت به Vegas اصلی، Vegas-A و CODETCP، می‌تواند سهم منصفانه‌تری از پهنای باند را بدست آورد بنابراین دارای توان عملیاتی بالاتری است.

۳. حفظ خصوصیات Vegas

براساس نتایج شبیه‌سازی، مشاهده شد که الگوریتم پیشنهادی، خصوصیت عدالت را حفظ می‌کند و در یک سناریو همگن(مقاربت کننده با هم‌جنس خود) مشابه Vegas رفتار می‌کند. بنابراین الگوریتم پیشنهادی، خصوصیات خوب Vegas را حفظ می‌کند و می‌تواند در یک محیط شبکه همگن(مقاربت کننده با هم جنس خود) پایدار باشد.

مراجع

[1] W. Stevens, "TCP Slow Start, Congestion Avoidance, Fast Retransmit, and Fast Recovery Algorithms", in RFC 2001, January 1997.

[2] L.S. Brakmo, L.L. Peterson, "TCP Vegas: end to end congestion avoidance on a global internet", in: IEEE Journal on Selected Areas in Communications, vol. 13, issue 8,pp. 1465-1480, August 1995.

[3] G. Hasegawa, H. Miyahara, "Fairness and stability of congestion control mechanisms of TCP", in: Proceedings of IEEE INFOCOM'99, vol. 3, pp. 1329-1336, 1999.

[4] J. Mo, R.L.V. Anantharam, J. Walrand, "Analysis and comparison of TCP Reno and Vegas", in: Proc. Of IEEE INFOCOM'99, vol. 3, pp. 1556-1563, March 1999.

[5] W. Feng, S. Vanichpun, "Enabling compatibility between TCP Reno and TCP Vegas", in: Proceedings of the IEEE Symposium on Applications and the Internet, pp. 301-308, January 2003.

[6] Y.C. Lai, C.L. Yao, "Performance comparison between TCP Reno and TCP Vegas, in: Computer Communications", vol. 25, issue 18, pp. 1765- 1773, December 2002.

[7] K.N. Srijith*, L. Jacob1, A.L. Ananda, "TCP Vegas-A: Improving the Performance of TCP Vegas", pp. 429-440, accepted 23 August 2004.

[8] Y.C.Chan[a] , C.L. Lin[a], C.T.Chan[b], C.Y. Ho[c], "CODE TCP: A competitive delay-based TCP", pp. 1013-1029, Accepted 11 January 2010.

[9] G. Hasegawa, K. Kurata, M. Murata, "Analysis and improvement of fairness between TCP Reno and Vegas for deployment of TCP Vegas to the internet", Proceedings of the IEEE International Conference on Network Protocols (ICNP 2000), pp. 177-186, November 2000.

[10] Y.C. Lai, "Improving the performance of TCP Vegas in a heterogeneous environment", in: Proceedings of the IEEE ICPADS, pp. 581-587, June 2001.

[11] E. Weigle, W. Feng, "A case for TCP Vegas in high-performance computational grids", in: High Performance Distributed Computing, pp. 158-167, August 2001.

[12] A. DeVendictis, A. Baiocchi, M. Bonacci, "Analysis and enhancement of TCP Vegas congestion control in a mixed TCP Vegas and TCP Reno network scenario, in: Performance Evaluation", vol. 5, issue 3-4, pp. 225-253, August 2003.

[13] S. Jamali, A.Eftekhari, "PSO-Vegas: PSO-based enhanced Vegas", ISSN 0033-2097, R. 87 NR 1/2011

[14] T.S. Pan, M.F.Horng, C.C. Shih, B.H. Su ,"An Intelligent TCP Vegastosupportlong-distancehighspeednetworks", pp. 3059-3071,June 2011.

[15] Network Simulator 2 (NS-2), Available from <http:// www.isi.edu/nsnam/ns>.

[16] R. Jain, A. Durresi, G. Babic, "Throughput fairness index: an explanation", Available from <http:// www.cis.ohio-state.edu/-jain/atmf/a99-0045.htm>, February 1999.