علوم رایانش و فناوری اطلاعات
نشریه علمی انجمن کامپیوتر ایران
مجلد ۱۳، شماره ۱، ۱۳۹۴
صفحات ۴۳-۵۳
مقاله عادی

نشریه علوم رایانش و فناوری اطلاعات

مشکلات و راهکارهای بهبود پروتکل TCP در شبکه‌های بی‌سیم

مریم شفیعی ناصر یزدانی مسعود رهگذر

دانشکده برق و کامپیوتر، دانشگاه تهران، تهران، ایران

چکیده

پروتکل TCP رایج‌ترین پروتکل انتقال در اینترنت است. از آنجا که این پروتکل برای شبکه‌های سیمی طراحی و پیاده‌سازی شده است، کارایی آن در محیط‌های بی‌سیم دچار افت و کاهش شدیدی می‌شود. از طرفی این پروتکل ارسال مطمئن داده‌ها را تضمین می‌کند و حدود ۸۵٪ کاربردهای اینترنت از این استاندارد استفاده کرده‌اند، بنابراین حذف این پروتکل یا نادیده گرفتن مشکلات آن در شبکه‌های بی‌سیم عملا غیرممکن است. در سال‌های اخیر، طرح‌ها و پیشنهادهای بسیاری به‌منظور حل مشکل ناکارآمد بودن TCP در شبکه‌های بی‌سیم ارائه شده است. ما در این مقاله سعی کرده‌ایم تا تمام مشکلاتی که در شبکه‌های بی‌سیم بر کارایی TCP تاثیر منفی گذاشته‌اند را مورد بررسی قرار دهیم. ما این مشکلات را به دو دسته، مشکلات ناشی از ساختار پروتکل TCP و مشکلات ناشی از ویژگی‌های محیط بی‌سیم بر کارایی پروتکل TCP، تقسیم کرده‌ایم. همچنین متناسب با این مشکلات، روش‌های بهبود کارایی TCP را از نقطه نظر کاربردها و از نقطه نظر پیاده‌سازی به دو دسته تقسیم کرده‌ایم. از نقطه نظر کاربردها، شبکه‌های بی‌سیم رایج شامل شبکه‌های ماهواره‌ای، Ad hoc و سلولی هستند. در دسته دوم نیز روش‌های بهبود TCP را به پنج دسته تقسیم کرده‌ایم: اتصال تقسیم شده، کنترل ازدحام انفعالی، کنترل ازدحام فعالانه، ارتقاء کارکرد لایه پیوند داده و روش‌های همکاری بین لایه‌ای. ما درباره ماهیت هر یک از روش‌ها بحث کرده و برای هرکدام از آنها پروتکل‌هایی را برای مثال آورده‌ایم و درباره عملکرد و نقاط ضعف و قوتشان بحث کرده‌ایم. در ادامه یک پروتکل جدید انتها به انتها به نام VECU که در برخورد با اتلاف بسته‌های تصادفی موثر و کارامد است معرفی می‌نماییم. عنصرکلیدی VECU این است که این پروتکل سطح ازدحام شبکه را مانیتور کرده و از این اطلاعات برای تشخیص علت اتلاف بسته‌ها (ازدحام و یا خطای بیتی) استفاده می‌کند. در این روش زمانی که اندازه پنجره از نقطه اشباع دور است، اندازه پنجره را به صورت تهاجمی‌افزایش می‌دهد و وقتی که نزدیک به نقطه اشباع است به کندی افزایش می‌دهد. VECU با رفع وابستگی تابع افزایش پنجره از RTT، پهنای باند را بصورت عادلانه بین جریان‌ها با RTT متفاوت به اشتراک می‌گذارد. این پروتکل پایدار و مقیاس‌پذیر است و از لینک به صورت کامل استفاده می‌کند. این روش تنها نیاز به تغییر در سمت فرستنده دارد.

کلمات کلیدی: شبکه‌های بی‌سیم، روش‌های بهبود کارایی TCP، VECU، مشکلات پروتکل TCP.

۱- مقدمه

در طول دهه‌های گذشته ما شاهد رشد بی‌نظیری در شبکه‌های بی‌سیم بوده‌ایم ولی کیفیت سرویس‌دهی و بهره‌وری شبکه‌های بی‌سیم امروزی هنوز فاصله زیادی از نقطه مطلوب و ایده‌آل خود دارد. اتصالات بی‌سیم از نرخ خطای بیت بالا، پهنای باند پایین و تأخیرهای طولانی رنج می‌برند. TCP یک پروتکل عملی است که سرویس انتقال با قابلیت اطمینان، انتها به انتها و اتصال‌گرا را روی اینترنت فراهم می‌کند [۲، ۷]. همزمان که اینترنت به سمت شبکه‌های خیلی سریع و با فاصله زیاد حرکت می‌کند، کارایی TCP نیز به چالش کشیده شده است. در TCP استاندارد با اتلاف بسته به عنوان ازدحام در شبکه رفتار می‌شود در حالیکه این فرض در شبکه‌هایی با کانال‌های بی‌سیم همواره صحیح نمی‌باشد چون اتلاف بسته‌ها به دلایلی مثل نویز، خطای لینک و یا دلایلی غیر از ازدحام است و این تفسیر غلط باعث کاهش قابل توجه کارایی می‌شود [۳، ۴]. در مساله مورد نظر پرسش اصلی مربوط به بالابردن کارایی TCP در شبکه‌های بی‌سیم سرعت بالا است. از آنجا که بسیاری از کاربردها و سرویس‌های اینترنت تحت قوانین TCP عرضه می‌شوند و از طرفی استفاده از شبکه‌های بی‌سیم سرعت بالا بسیار ضروری و مفید است، و با توجه به مشکلات TCP در این محیط، هدف ما بالا بردن کارایی TCP را در این محیط‌ها است که تاثیر قابل توجهی بر بروی کارایی کاربردها و سرویس‌ها

خواهد داشت. در این مقاله ابتدا راه حل‌های موجود برای بالا بردن کارایی TCP در شبکه بی‌سیم و نقاط ضعف و قوت آنها مورد مطالعه قرار گرفته، سپس پروتکلی را به عنوان راه‌حل برای بالا بردن کارایی TCP در شبکه‌های بی‌سیم با سرعت‌های بالا ارائه داده‌ایم.

در بخش بعدی به معرفی ویژگی‌های محیط‌های بی‌سیم و سیار و مسائل و مشکلاتی که شبکه‌های بی‌سیم کنونی در زمینه TCP با آنها مواجه هستند می‌پردازیم. در بخش سوم و چهارم نیز به دسته‌بندی و مقایسه روش‌های بهبود کارایی پروتکل TCP در این شبکه‌ها پرداخته‌ایم. در ادامه یک پروتکل جدید ارئه داده‌ایم که جزئیات این پروتکل در بخش پنجم آمده است. به‌منظور ارزیابی کمی این روش و مشخص شدن مقدار بهبودی حاصله در این روش در مقایسه با سایر راهکارها، ما روش خود را در نرم NS شبیه‌سازی کردیم. پارامترهای محیط شبیه‌سازی و نتایج حاصل از آزمایشات در قالب نمودارها در فصل ششم آمده است. سرانجام در بخش هفتم نتیجه‌گیری از تحقیق بیان شده است.

۲- دلایل کارایی ضعیف TCP در شبکه‌های بی‌سیم

TCP الگوهای رفتاری نامطلوبی در شبکه‌های بی‌سیم از خود نشان می‌دهد و فاقد استراتژی‌هایی برای مصرف مقرون به صرفه انرژی است تا ما را به بازدهی بالایی برساند. TCP قادر به تشخیص ماهیت خطا نمی‌باشد بلکه تنها وقوع خطا را اعلام می‌کند. بنابراین مکانیزم بازیابی خطا همیشه کارایی ندارد، بخصوص وقتی که الگوی خطا مرتب عوض می‌شود، زیرا گم شدن بسته همیشه از نظر پروتکل ناشی از ازدحام است [۸].

۲-۱- مشکلات ناشی از ساختار پروتکل TCP

TCP یک پروتکل عملی است که سرویس انتقال با قابلیت اطمینان، انتها به انتها و اتصال‌گرا را روی اینترنت فراهم می‌کند. TCP از پنجره‌ای به نام پنجره ازدحام (cwnd) برای رسیدگی به ازدحام در شبکه استفاده می‌کند.پیاده‌سازی عملی TCP اغلب بر مبنای مفروضاتی بهینه‌سازی می‌شوند که فقط برای شبکه‌های سیمی صادق هستند و برای شبکه‌های بی‌سیم با شکست مواجه می‌شود و در لینک‌های سرعت بالا باعث درست مصرف نشدن منابع شبکه می‌شود [۶، ۷، ۸، ۹].

۳-۱-۱- عدم تشخیص خطای بیتی از ازدحام:

- افزایش تهاجمی cwnd۱
- به روزرسانی قطعی cwnd۲
- اقدامات قطعی۳
- ناعادلانه بودن TCP
- انقضای زمان متوالی
- متغیر بودن اندازه بسته‌ها
- افزایش پنجره ارسال به اندازه یک بسته در هر RTT
- زمان رفت و برگشت طولانی

۲-۲- مشکلات ناشی از لینک‌های بی‌سیم بر کارایی TCP

تفاوت بین شبکه‌های بی‌سیم و سیمی، همانطور که از نام آن برمی‌آید در این است که گره‌ها بدون اختیار داشتن هیچ زیرساختی بتوانند با هم ارتباط برقرار کنند. با وجود مزایای بسیار زیاد شبکه‌های بی‌سیم، برخی ویژگی‌های آن تاثیر منفی بر کارایی TCP دارد که شامل موارد زیر می‌باشند [۱، ۲۰، ۲۱].

- نرخ خطای بیتی بالا
- قطع شدن ارتباط
- پهنای باند محدود و متغیر
- اندازه سلول
- محدودیت انرژی
- توپولوژی پویای شبکه
- عدم تقارن لینک‌ها

۳- رویکردهای بهبود کارایی TCP

از آنجا که TCP برای یک نوع شبکه خاص یا برنامه‌های کاربردی بخصوص طراحی نشده است، پس می‌توان اصلاحات و بهبود هایی در مکانیزم های آن انجام داد تا کارایی آنرا افزایش داد. محققان سعی کردند تا ساختارهایی را پیشنهاد کنند که ضمن اینکه عملکرد پروتکل را در این شبکه های خاص بهبود می‌بخشد، در عین حال کمترین تغییر ممکن را در خود پروتکل ایجاد کند.

۳-۱- TCP برای کاربردهای مختلف بی‌سیم

اگر از جنبه کاربردها به این مسئله نگاه کنیم، ما چند محیط بی‌سیم داریم و TCP را می‌توان متناسب با مشکلات هر محیط طراحی کرد. رایج‌ترین شبکه‌های بی‌سیم، شبکه‌های ماهواره‌ای، شبکه‌های ad hoc و زیرساخت‌های بی‌سیم عمومی مانند شبکه‌های محلی بی‌سیم و سلولی هستند. طراحی TCP می‌تواند براساس ویژگی‌ها و نیازهای این محیط‌ها انجام شود. با این وجود مشکل اصلی همه شبکه‌های بی‌سیم، نرخ خطای بیتی بالا در آنها می‌باشد و هدف اصلی در طراحی این است که فرستنده از علت اصلی اتلاف بسته‌ها مطلع شود. در این صورت فرستنده قادر خواهد بود که تصمیم مناسبی برای تنظیم پنجره ازدحام بگیرد.

۳-۱-۱- شبکه‌های ماهواره‌ای

در شبکه‌های ماهواره‌ای، فاز شروع آهسته استاندارد TCP باعث می‌شود که فرستنده زمان زیادی را برای رسیدن به نرخ ارسال بالا صرف کند. از آنجا که اکثر کاربردها مثل HTTP فایل‌های کوچک را انتقال می‌دهند، ممکن است در کل زمان انتقال فایل، TCP در فاز شروع آهسته بماند و نتواند به صورت کامل از پهنای باند شبکه استفاده کند.

TCP-Peach به پروتکل TCP و مکانیسم انتقال مجدد سریع دو مکانیسم اضافه می‌کند، شروع ناگهانی و بازیابی سریع. در فاز شروع ناگهانی، در بازه زمانی $0 \le t < RTT$، فرستنده ابتدا یک بسته داده می‌فرستند و سپس به اندازه (rwnd-1) بسته ساختگی با اولویت کم می‌فرستند و بعد در بازه $RTT \le t < 2RTT$، ACKهای مربوط به داده و بسته‌های ساختگی که در بازه $0 \le t < RTT$ فرستاده شده بودند به فرستنده می‌رسند. پس به ازای هر ACK مربوط به بسته‌های ساختگی، فرستنده اندازه cwnd را یکی اضافه می‌کند و یک بسته داده ارسال می‌کند. بنابراین نرخ ارسال فرستنده به صورت ناگهانی افزایش می‌یابد. بازیابی سریع هم جایگزین مکانیسم بازیابی سریع در استاندارد TCP می‌شود تا بازدهی را در لینک‌هایی با نرخ خطای بالا، افزایش دهد [۱۳].

۳-۱-۲- شبکه‌های ad hoc

عبارت است از یک شبکه بدون زیرساخت ثابت که از تعدادی گره بی‌سیم تشکیل شده است که به‌طور کاملا پویا شبکه را بدون هیچ مدیریت مرکزی تشکیل می‌دهند. ATCP [۱۴] به عنوان یک راه حل انتها به انتها برای بهبود کارایی این شبکه‌ها طراحی شده است که به صورت یک لایه بین لایه‌های TCP و IP پیاده‌سازی شده است، پس تغییری در کدهای TCP نمی‌دهد، و از ECN برای تشخیص ازدحام و علت وقوع خطا استفاده می‌کند. همچنین این الگوریتم از پیام "عدم دسترسی به مقصد" برای تشخیص تغییر در مسیرها استفاده می‌کند. فرستنده بر اساس این دو بازخورد شبکه، در حالت کنترل ازدحام یا انتقال مجدد قرار می‌گیرد و اگر تغییر مسیر رخ دهد، در فاز پافشاری برای مسیری برای ارسال مجدد می‌ماند.

۳-۱-۳- شبکه‌های سلولی

این شبکه‌ها امکان اینکه تعداد زیادی عامل بی سیم در هر نقطه از شبکه با یکدیگر از طریق ایستگاه‌های پایه ارتباط برقرار کنند را بوجود می‌آورد. تغییرات TCP در این شبکه‌ها براساس ویژگی‌های شبکه بی‌سیم مانند نرخ خطای بیتی بالا و handoff صورت می‌گیرد. Freeze-TCP یک راه حل انتها به انتها برای بهبود کارایی TCP در این شبکه‌ها است.

این الگوریتم علت اصلی کاهش کارایی را قطع شدن‌های مکرر می‌داند و فرض می‌کند که واحد متحرک از قدرت سیگنالش اطلاع دارد و می‌تواند قطعی اتصال قریب الوقوع را پیش‌بینی کند. پس گیرنده متحرک در این شرایط اندازه پنجره‌اش را در ACK مقدار صفراعلام می‌کند و فرستنده هم مجبور می‌شود که ارسال بسته‌ها را تا تغییر اندازه پنجره متوقف کند [۱۵].

۳-۲- پیاده‌سازی‌های TCP در شبکه‌های بی‌سیم

از نقطه نظر پیاده‌سازی و اجرا، الگوریتم‌های TCP بی‌سیم می‌توانند به پنج دسته تقسیم شود. حالت تقسیم، کنترل ازدحام انفعالی، کنترل ازدحام فعالانه، لایه پیوند داده و همکاری بین لایه‌ای.

۳-۲-۱- تقسیم اتصال

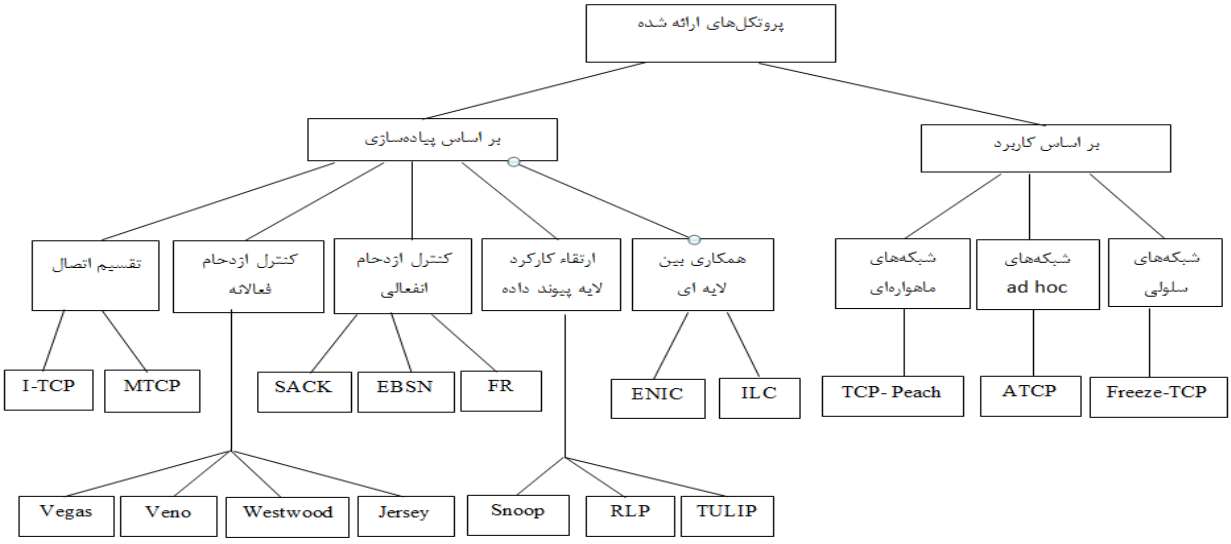
قسمت سیمی در یک شبکه ناهمگن از جهت ظرفیت اتصال و نرخ خطا از قسمت بی‌سیم بسیار قابل اطمینان‌تر است و اتصالات پراتلاف بی‌سیم در انتقال، تبدیل به یک گلوگاه می‌شود. در این دسته روش‌ها می‌توان هر اتصال TCP را در نقطه تماس دو زیر شبکه (مثلا ایستگاه پایه) به دو اتصال مجزا تقسیم کرد. بزرگترین نقطه ضعف این روش مشکل حفظ مفاهیم انتها به انتها بخصوص در مکانیزم اعلام وصول‌ها (ACK) می‌باشد. ضعف دیگر این است که فرآیند handover بسیار پیچیده و زمان‌بر می‌شود که سربار زیادی به شبکه تحمیل می‌کند.

I-TCP: در این روش [۱۶] ایستگاه ثابت از یک اتصال TCP با میزبان ثابت بهره می‌برد، در حالی که در همان زمان از پروتکل دیگری که برای کار روی پیوندهای بی‌سیم بهینه شده است، برای اتصال با میزبان سیار استفاده می‌کند. داده‌های ارسال شده به میزبان سیار توسط ایستگاه ثابت دریافت شده و بافر می‌شود و بلافاصله به فرستنده اعلام وصول می‌شود. به این ترتیب امکان دارد که قبل از آنکه یک بخش عملا توسط گیرنده (میزبان سیار) دریافت شود توسط ایستگاه پایه به فرستنده اعلام وصول شود. پس از انتقال ACK از ایستگاه ثابت به فرستنده، وظیفه انتقال و تحویل داده‌ها به میزبان سیار به عهده ایستگاه پایه گذاشته می‌شود.

MTCP: روشی مشابه با I-TCP پیشنهاد کرده است با این تفاوت که تنها زمانی رسیدن داده به فرستنده اعلام وصول می‌شود که آخرین بایت از داده به میزبان سیار رسیده باشد. این کار به منظور حفظ مفاهیم انتها- به- انتهای TCP انجام می‌شود. برتری MTCP به I-TCP در این است که اولا مفهوم انتها به انتها را حفظ می‌کند و دیگر اینکه در صورت قطعی اتصال مکرر و کوتاه، یا طولانی و نادر که به علت جابجایی کاربر یا علل دیگر پدید می‌آید، بهتر عمل می‌کند [۱۷].

۳-۲-۲- کنترل ازدحام انفعالی و انتها به انتها

کنترل جریان استاندارد Reno بصورت انفعالی است؛ TCP با افزایش مداوم و تدریجی پنجره ازدحام تا رسیدن شبکه به وضعیت ازدحام، به دنبال فهمیدن ظرفیت دردسترس شبکه است و در این صورت، ازدحام غیرقابل اجتناب است. پس از آن، TCP به نرخ انتقال بسیار کندتر سقوط خواهد کرد که این برای اتلاف بسته تصادفی در شبکه‌های بی‌سیم غیر ضروری است.



شکل ۱- روش‌های بهبود کارایی پرونکل TCP

SACK: این الگوریتم [۱۸] در واقع راهی برای تهیه اطلاعات بیشتر در مورد بسته‌های دریافت شده است. گزینه SACK یک فیلد اختیاری در سرآیند TCP است. هرگاه داده‌ای خارج از ترتیب دریافت شود، SACK فرستاده می‌شود. زمانی که فرستنده یک ACK با گزینه SACK دریافت می‌کند، همه بخش‌هایی که در بلاک‌های SACK مشخص شده‌اند به‌عنوان SACKed مارک می‌شوند و سطر مربوط به هر بخش تا وقتیکه آن بخش ACK شود، در جدول باقی می‌ماند. در طول مدت انتقال مجدد سریع، فرستنده ابتدا تلاش می‌کند تا حفره‌های موجود در بلاک‌های SACK را قبل از فرستادن هر بسته جدید انتقال مجدد دهد. هر وقت که فرستنده یک بخش را دوباره انتقال می دهد، آن بخش در جدول به عنوان انتقال مجدد شده مارک می‌شود. اگر یک بخش انتقال مجدد شده گم شود، فرستنده تایم اوت داده و شروع آهسته را اجرا می‌کند. زمانی که تایم اوت رخ می‌دهد، فرستنده جدول SACK را دوباره تنظیم می‌کند.

FR: این مکانیسم [۱۲] از ساده‌ترین روش‌ها برای بهبود کارایی TCP است که در آن از مکانیسم بازیابی سریع در TCP استاندارد استفاده شده است تا مشکل اتلاف بسته‌ها بر اثر handoff را حل کند. تغییرات کمی در نرم‌افزار IP میزبان متحرک نیز صورت می‌گیرد تا برای handoff بتواند به میزبان ثابت سیگنال بفرستد. این روش بر مشکل handoff تمرکز می‌کند و توجهی به دیگر مشکلات ندارد.

ECN: این راهکار بر پایه این واقعیت کار می‌کند که مسیریاب‌ها هیچ اطلاعاتی به فرستنده فیدبک نمی‌کنند و از مکانیزم صف‌بندی drop tail استفاده می‌کنند. TCP با اعلان صریح ازدحام، وظایف بیشتری به روترها می‌دهد. این روش برای تحویل فیدبک و انتقال قابلیت ECN از دو بیت در سرآیند IP و یک بیت در سرآیند TCP استفاده می‌کند. زمانی که طول صف از یک سطح آستانه تجاوز کند، مسیریاب بسته‌ها را با علامت «دچار ازدحام شده» نشانه‌گذاری می‌کند. در مقصد، بیت دچار ازدحام شده به بیت ECN_echo در اعلام وصول TCP کپی شده و با ACK به فرستنده TCP بازگردانده می‌شود. فرستنده TCP پس از دریافت ECN_echo از پنجره ازدحام خود می‌کاهد تا ازدحام را کاهش دهد [۱۹].

Congestion Coherence (CC): این روش [۲۰] انتقال مجدد محلی روی پیوند بی سیم را در کنار یک راهکار با عنوان «انسجام ازدحام» به خدمت می‌گیرد. روش کار الگوریتم به این صورت است که در صورت گم شدن بسته‌ها یا نامرتب رسیدن آن‌ها، گیرنده به بسته‌های مجاور آن نگاه می‌کند، اگر بعضی از بسته‌ها با ECN نشانه‌گذاری شده باشد پس علت وقوع خطا، ازدحام در شبکه است. در واکنش به ازدحام، با استفاده از اعلام وصول تکراری یک انتقال مجدد انتها به انتها را آغاز می‌کند. حال اگر بسته‌های مجاور با ECN نشانه‌گذاری نشده باشند، نتیجه‌گیری می‌کند که گم شدن بسته ناشی از خطای انتقال در پیوند بی‌سیم بوده است و باید انتقال مجدد محلی روی لایه پیوند بی‌سیم برای بازیابی آن اجرا شود.

۳-۲-۳- کنترل ازدحام فعالانه

در کنترل ازدحام فعالانه، فرستنده به صورت هوشمندانه به وضعیت شبکه یا دلیل اتلاف بسته‌ها واکنش نشان می‌دهد، بنابراین از وارد شدن شبکه به وضعیت نامطلوب جلوگیری می‌کند. استراتژی‌های متفاوتی را در طراحی می‌توان در نظر گرفت تا فرستنده بتواند اطلاعات صریح و روشنی از وضعیت شبکه داشته باشد.

TCPVegas: این الگوریتم [۲۱] بسته‌های جمع شده در بافر لینک گلوگاه را تخمین زده و سرعت واقعی ارسال را ثبت می‌کند تا بازده واقعی را محاسبه کند. تفاوت بین بازدهی بهینه و سرعت ارسال واقعی برای محاسبه تعداد بسته‌های جمع شده در شکه به کار می‌رود. همچنین با مقایسه توان عملیاتی اندازه‌گیری شده و توان عملیاتی مورد انتظار که بوسیله اندازه پنجره محاسبه می‌شود، تغییرات توان عملیاتی را در نظر می گیرد. الگوریتم پرهیز از ازدحام از این اطلاعات برای حفظ مقدار بهینه داده در شبکه استفاده می‌کند.

TCPVeno: متدولوژی آن شبیه Vegas است به علاوه راهی را برای تشخیص علت اتلاف بسته‌ها پیشنهاد می‌دهد. اگر تعداد بسته‌های انباشته شده کمتر از یک مقدار آستانه باشد، اتلاف به دلیل خطای تصادفی است و در غیر این صورت به دلیل ازدحام است. اگر اتلاف به دلیل ازدحام باشد، رویکرد Reno را اتخاذ می‌کند و اگر به دلیل خطای تصادفی باشد، پنجره ازدحام را با رفتاری محافظه کارانه افزایش می‌دهد [۲۲].

TCP Westwood: یک رویکرد مبتنی بر سرعت و انتها به انتها می‌باشد که فرستنده با اندازه‌گیری و میانگین‌گیری نرخ بازگشت ACKها، پهنای باند قابل دسترسی را به صورت پویا تخمین می‌زند و از آن برای تعیین نرخ ارسال استفاده می‌کند. وقتی که TCP Westwood با رسیدن سه ACK تکراری تشخیص می‌دهدکه اتلاف رخ داده است، مقدار آستانه را به معکوس ضریب پهنای باند- تاخیر تنظیم می‌کند [۲۳].

TCP-Jersey: نرخ ارسال را متناسب با وضعیت شبکه تنظیم می‌کند که دارای دو جزء کلیدی است. الگوریتم تشخیص پهنای باند در دسترس (ABE) که در سمت فرستنده پیاده‌سازی می‌شود و هشدار ازدحام (CW) در مسیریاب پیاده می‌شود [۲۴].

۳-۲-۴- ارتقاء کارکرد لایه پیوند داده

همان‌طورکه قبلا نیز اشاره شد، اکثر مشکلات پروتکل‌های انتقال مطمئن مانند TCP از ماهیت متفاوت رسانه بی‌سیم ناشی می‌شود. بنابراین عاقلانه آن است که تلاش شود تا این مشکل را در ریشه خود حل کنیم. پروتکل لایه پیوند (LL) که بالای لایه فیزیکی اجرا می‌شود زودتر از دیگر لایه‌ها از حذف شدن بسته‌ها مطلع می‌شود و بنابراین می‌تواند سریع‌تر به آن پاسخ دهد. همچنین کنترل بیشتری روی پروتکل لایه فیزیکی دارد. لذا انتظار داریم با کاهش ناکارآمدی رسانه بی‌سیم در لایه پیوند داده‌ها، کانال ارتباطی قابل اطمینانی- همانند آنچه در کانال سیمی داریم- برای پروتکل لایه انتقال فراهم شود. با این روش ما مجبور نیستیم پیاده سازی‌های کنونی TCP را تغییر بدهیم و براحتی می‌توانیم با ساختار نرم‌افزاری و سخت‌افزاری موجود کار کنیم. علیرغم تمام محاسنی که برای روش‌های مرتبط با لایه پیوند داده‌ها ذکرشده، ولی متأسفانه شبیه کردن رسانه بی‌سیم به کانال سیمی کار سادهای نیست، پروتکل LL باید توزیع و تحویل نسبتاً مطمئن بسته‌ها را تضمین کند. اینکار معمولا با استفاده از راهکار ARQ یا FEC انجام می‌شود. ولی این روش‌ها نیز مشکلات و کاستی‌های خاص خود را دارند [۲۵].

Snoop: مشهورترین روش از گروه لایه پیوند داده‌ها پروتکل snoop می‌باشد [۲۶]. در این روش یک عامل Snoop به ایستگاه پایه اضافه می‌شود که بسته‌های ارسالی ولی اعلام وصول نشده را در حافظه خود نگاه می‌دارد تا بتواند بر طبق تایم اوت‌ها و سیاست بسته اعلام وصول نشده، انتقال مجددهای محلی را روی این بسته‌ها اجرا کند. ایستگاه ثابت بی‌آنکه نیازی به افزایش کارکرد لایه انتقال داشته باشد، تمام بسته‌های مبادله شده روی اتصال‌های سیمی و بی‌سیم را مانیتور می‌کند. همچنین Snoop اعلام وصول‌های دریافتی از میزبان سیار را مانیتور و پردازش می‌کند تا بدین‌وسیله عملیات‌های مورد نیاز را اجرا نماید. پس از دریافت یک ACK جدید، Snoop فقط بسته‌های اعلام وصول شده را از حافظه پاک می‌کند.

RLP: این الگوریتم [۱۲] سگمنت‌های TCP را به فریم‌هایی تقسیم می‌کند و از یک کد تصحیح خطای قوی (ARQ) و مکانیسم ارسال سریع برای پنهان کردن

خطاهای محیط بی‌سیم از دید فرستنده، استفاده می‌کند و مانع کاهش کارایی TCP می‌شود.

TULIP: این روش [۲۵] برای کانال‌های بی‌سیم نیمه دوطرفه با پهنای باند محدود طراحی شده است. با وجود اینکه TULIP در گروه روش‌های ناآگاه از TCP قرار می‌گیرد، اما باید حداقل از نوع سرویس ارائه شده در لایه انتقال آگاه باشد. TULIP همانند Snoop بسته‌ها را بطور محلی بافر می‌کند و با کمک روش ARQ سعی می‌کند تا پیش از این که فرستنده TCP تایم اوت بدهد، خطاهای روی پیوند بی‌سیم را بازیابی نماید. نکته جالب توجه این است در TULIP تحویل قابل اعتماد اعلام وصول‌های TCP پیش‌بینی نشده است.

DDA: یک راهکار مبتنی بر پروتکل لایه پیوند و ناآگاه از TCP است که اساس کار آن با Snoop یکسان است [۲۷]. طراحان این روش خاطرنشان کرده‌اند که بخش‌های داده TCP باید در فریم LL منفرد و هر بخش ACK نیز در ACK مربوط به LL منفرد خود قرار داده شود. DDA از شماره ترتیب‌های متفاوت برای فریم‌ها استفاده می‌کند، همچنین پروتکل هیچ کوششی برای تحویل مرتب بسته‌ها به لایه‌های بالاتر نمی‌کند. رسیدن اعلام وصول تکراری به BS نشان دهنده این است که خطایی در LL بوقوع پیوسته است.

۳-۲-۵- روش‌های همکاری بین لایه‌ای

در شبکه‌های بی‌سیم علت‌های کاهش کارایی به تنهایی از یک لایه نشأت نمی‌گیرد، بلکه تأثیرات متقابل فاکتورهای مختلف در لایه‌های متفاوت منجر به افت کارایی TCP می‌شود. به‌منظور رسیدن به یک انتقال مطمئن‌تر و کاراتر، باید فعل وانفعالات میان لایه‌های مختلف را به دقت مورد توجه و بررسی قرار دهیم. افزایش ارتباطات عمودی لایه به لایه می‌تواند باعث نزدیک‌تر شدن پروتکل‌های لایه بالاتر به لایه‌های پایین‌تر شود که به فعالیت‌های بین لایه‌ای درست‌تر و مؤثرتر منتهی خواهدشد. بدین‌ترتیب نه تنها سربار ناشی از ارسال پیام‌های کنترلی در روی پهنای باند کم می‌شود، بلکه سرعت واکنش به خطاهای شبکه در موقع جابجایی کاربر و خطای انتقال بالا بسیار افزایشمی یابد [۲۸].

ENIC: یک راهکار مبتنی بر اعلان صریح به همراه یک مکانیزم کنترل بین لایه‌ای پیشرفته می‌باشد [۲۸] که درآن سه پروتکل MAC (لایه پیوند)، مسیریابی (لایه شبکه) و TCP (لایه انتقال) بطورگروهی عمل می‌کنند. روال کار به این صورت است که پس از شکسته شدن یک مسیر، پروتکل مسیریابی در لایهٔ شبکه یک پیغام مبنی بر خرابی مسیر (EREN)[۴] به TCP فرستاده و عملیات انتقال داده تا بازیابی مجدد مسیر متوقف می‌شود. ضمنا متغیرهایی مثل اندازه پنجره ازدحام، مهلت زمان سنج انتقال مجدد و غیره دست نمی‌خورند. بعد از اینکه یک مسیر جدید توسط الگوریتم مسیریابی باز سازی شد، پیغامی داخلی به نشانه بازیابی مسیر (ERRN)[۵] به TCP ارسال می‌شود.

ILC-TCP: در این روش [۲۹] یک لایه مدیریتی جدید به نام SM موازی با پشته شبکه اضافه می‌شود. مطابق با شکل زیر هر لایه مقداری از اطلاعات خود را با این لایه به اشتراک می‌گذارد. در زمانی که کانال ارتباطی در حال نوسان یا ضعیف شدن باشد یا قطعی اتصال در شرف وقوع باشد، لایه پیوند اطلاعات مربوط به وضعیت خود را به SM می‌فرستد و لایه IP وضع اتصال‌پذیری به شبکه را به SM اطلاع می‌دهد. این روش، یک مکانیزم انتها به انتهای واقعی است و نیازی به کمک گرفتن از هیچ گره میانی مانند ایستگاه پایه ندارد.

۴- مقایسه روش‌ها

در دو جدول زیر خلاصه‌ای از ویژگی‌های پروتکل‌های ارئه شده و مشکلاتی که توسط آن‌ها برطرف شده‌اند را آورده‌ایم. به علاوه می‌توان گفت که مزیت اصلی پروتکل‌های لایه لینک و انفعالی و فعالانه این است که مفهوم انتها به انتهای TCP را حفظ می‌کنند. پروتکل‌های لایه لینک و فعالانه در محیط‌هایی با نرخ خطای بیتی بالا بسیار خوب عمل می‌کنند اما پروتکل‌های لایه لینک ساختار لایه‌ای پروتکل‌های شبکه را حفظ نمی‌کند. راهکار تقسیم اتصال، مفاهیم انتها به انتها را نقض می‌کند. پروتکل‌های فعالانه می‌توانند ماهیت خطا را تشخیص دهند و متناسب با نوع خطا تصمیم مناسب را برای تنظیم پنجره ازدحام بگیرند و از کاهش بیهوده پنجره ازدحام و در نتیجه کاهش سرعت انتقال جلوگیری کنند. پروتکل‌های لایه پیوند بشدت به خصوصیات پیوند بی‌سیم وابسته هستند و نمی‌توانند در مواردی که قطعی طولانی مدت یا مکرر داریم کمک کنند. روش‌های فعالانه و انفعالی این مشکل را دارند که نیاز به تغییر در کد TCP میزبان ثابت دارند و در بعضی مواقع نمی‌توانند به سازگاری برسند. مزیت پروتکل‌های تقسیم این است که با پروتکل‌های شبکه‌های سیمی سازگاری دارند، بنابراین برای تطبیق با میزبان‌های سیار نیاز به تغییری در میزبان‌های ثابت ندارد. آنها مشکلات قابلیت حرکت و شبکه بی‌سیم را از دید میزبان ثالث پنهان نگه می‌دارند. این دسته پروتکل‌ها می‌توانند مشکل قطعی اتصال را به صورت کارآمدی مدیریت کنند.

جدول ۱- مقایسه پروتکل‌ها ی ارائه شده

	نام روش	تشخیص علت خطا	مفهوم TCP	پشتیبانی از قابلیت حرکت	ایجاد تغییرات	کاربرد
روش‌های مربوط به کاربرد	TCP-Peach	نه	انتها به انتها	بالا	مسیریاب و فرستنده	ماهواره‌ای
	ATCP	بله	انتها به انتها	بالا	پشته TCP	Ad hoc
	Freeze-TCP	بله	انتها به انتها	بالا	ایستگاه ثابت	سلولی
روش‌های انفعالی	TCP-New Reno	نه	انتها به انتها	کم	فرستنده	ناهمگن
	TCP-SACK	نه	انتها به انتها	کم	فرستنده	ناهمگن
روش‌های فعالانه	TCP-Vegas	بله	انتها به انتها	کم	فرستنده	ناهمگن
	TCP-Veno	بله	انتها به انتها	کم	فرستنده	ناهمگن
	TCP-Westwood	بله	انتها به انتها	بالا	فرستنده	ناهمگن
	TCP-Jersey	بله	انتها به انتها	بالا	مسیریاب و فرستنده	ناهمگن
همکاری بین لایه‌ای	ENIC	بله	انتها به انتها	کم	فرستنده و گیرنده	ناهمگن
	ILC-TCP	بله	انتها به انتها	بالا	فرستنده	ناهمگن

جدول ۲- مقایسه پروتکل‌های ارائه شده

	I-TCP	MTCP	WTCP	EBSN	FR	ECN	CC	Snoop	RLP	DDA	Westwood	Veno	Santa Cruz	ILC-TCP
نرخ خطای بیتی بالا	√	√	√	√			√	√	√	√	√	√	√	√
خطای توده‌ای		√		√				√		√			√	
Handoff	√				√	√		√		√	√			√
قطعی اتصال طولانی	√	√												√
قطعی اتصال مکرر	√	√												√
پهنای باند			√						√		√	√	√	
اندازه سلول					√									
محدودیت انرژی			√	√		√								√
انقضای زمان متوالی	√	√	√	√		√	√			√				
متغیر بودن اندازه بسته‌ها		√		√						√				
سازگاری	√	√						√	√	√	√	√		
دوطرفه بودن	√	√				√	√	√		√				√

۵- راهکار پیشنهادی برای بهبود کارایی TCP

تفسیر غلط اتلاف تصادفی به عنوان ازدحام باعث می‌شود که TCP نرخ داده را کم کند که این باعث کاهش قابل توجه کارایی می‌شود. برای غلبه بر این مشکل ما می‌توانیم آن را دو قسمت کنیم:

۱) چگونه اتلاف بسته تصادفی و اتلاف ناشی از ازدحام را از هم تشخیص دهیم.

۲) چگونه از این اطلاعات برای تصحیح فرایند تنظیم پنجره ازدحام استفاده کنیم.

ایده‌ای در TCP Vegas [۲۱] وجود دارد که از یک مکانیسم ساده برای اندازه‌گیری شرایط شبکه استفاده می‌کند و از اتلاف بسته‌ها به صورت فعال جلوگیری می‌کند. ما از این ایده برا تعیین علت اتلاف بسته به دلیل ازدحام و یا اتلاف تصادفی استفاده می‌کنیم. بنابراین ما می‌توانیم در زمانی که اتلاف تصادفی رخ می‌دهد با سرعت کمتری اندازه پنجره ازدحام را کاهش دهیم و از کاهش غیرضروری کارایی جلوگیری کنیم. این روش تاثیر منفی بر روی دیگر اتصالات TCP همزمان در شبکه ندارد.

VECU بهبودهایش را از کارایی در بهره‌برداری از پهنای باند استفاده نشده بدست می‌آورد نه از گرفتن پهنای باند از دیگر اتصالات TCP. همچنین این روش فقط نیاز به تغییر در سمت فرستنده دارد، بدون نیاز به هیچ گونه تغییری در سمت گیرنده یا مداخله نودهای شبکه میانی. بنابراین می‌تواند به سرعت در سرورهای برنامه‌های کاربردی اینترنت پیاده‌سازی شود.

۵-۱- تشخیص علت اتلاف بسته‌ها

تشخیص بین اتلاف حاصل از ازدحام و اتلاف تصادفی یک مشکل بنیادی در TCP است. ما سعی می‌کنیم که وضعیت شبکه را مانیتور کنیم. اگر اتلاف بسته در حالی که اتصال در وضعیت ازدحام است تشخیص داده شود، فرض می‌کنیم که اتلاف به دلیل ازدحام است و در غیر این صورت فرض می‌کنیم که اتلاف تصادفی است. ابتدا مانند آنچه که در TCP Vegas است، نرخ واقعی و نرخ مورد انتظار را اندازه‌گیری می‌کنیم:

$$Expected= cwnd/BaseRTT \quad (۱)$$

$$Actual=cwnd/RTT \quad (۲)$$

که در معادلات بالا cwnd برابر اندازه فعلی پنجره، BaseRTT کمترین RTT اندازه‌گیری شده و RTT زمان رفت و برگشت اندازه‌گیری شده است. اختلاف بین این نرخ‌ها برابر است با

$$Diff= Expected- Actual \quad (۳)$$

زمانی که RTT>BaseRTT، به این معنی است که یک لینک گلوگاه وجود دارد که بسته‌های اتصال در آن جمع شده‌اند. به انباشتگی در این صف، N می‌گوییم.

$$RTT=BaseRTT+ N/Actual \quad (۴)$$

قسمت دوم سمت راست معادله بیانگر تاخیر اضافی در لینک گلوگاه است. با توجه به فرمول‌های بالا داریم:

$$N=Actual*(RTT-BaseRTT)=Diff*BaseRTT \quad (۵)$$

از N برای تعیین اینکه اتصال در وضعیت ازدحام است یا نه، استفاده می‌کنیم.به عبارت دیگر در زمانی که یک اتلاف رخ می‌دهد، اگر$N<\gamma$ فرض می‌کنیم که اتلاف رندم بوده است و به دلیل ازدحام نبوده است و کاهش را با سرعت کمتری انجام می‌دهیم. در غیر این صورت فرض می‌کنیم که اتلاف به دلیل ازدحام بوده است. آزمایشات نشان می‌دهد که $\gamma=3$ یک مقدار خوب است.

۵-۲- مکانیسم افزایش پنجره ازدحام

برای تابع افزایش اندازه پنجره از یک تابع چند جمله‌ای درجه فرد استفاده می‌کنیم. این الگوریتم با دریافت یک رخداد اتلاف بسته، اندازه پنجره‌اش را با پارامتر ضربی β کاهش می‌دهد. اندازه پنجره دقیقا قبل از کاهش را W_{max} و اندازه پنجره دقیقا بعد از کاهش را W_{min} می‌نامیم. از آنجا که اتلاف بسته در W_{max} رخ می‌دهد، اندازه پنجره‌ای که در حال حاضر شبکه می‌تواند بدون اتلاف مدیریت کند باید جایی بین این دو باشد. در حالیکه اکثر الگوریتم‌های استاندارد TCP از تابع افزایش محدب بعد از رخداد اتلاف استفاده می‌کنند، ما از تابع مقعر و سپس محدب که تابعی از زمان سپری شده از آخرین ازدحام است، برای افزایش پنجره استفاده می‌کنیم. در نتیجه رشد پنجره فقط به زمان واقعی بین دو رویداد ازدحام پشت سر هم وابسته می‌باشد و این باعث می‌شود که رشد پنجره مستقل از RTT شود. این ویژگی به جریان‌های VECU رقیب این امکان را می‌دهد که مستقل از RTT آنها، تقریبا اندازه پنجره مشابه داشته باشند و به RTT-fairness بهتر دست

یابد. توجه داشته باشید که پروتکل‌ها بـا تـابع افـزایش محـدب بیشـترین افـزایش پنجره را در اطراف نقطه اشباع دارند که این باعث اتلاف بسته‌های توده‌ای می‌شود.

پس این الگوریتم بعد از دریافت رخداد اتلاف بسته، مقدار W_{max} را که انـدازه فعلی پنجره است ثبت می‌کند و شروع به کاهش‌اندازه پنجره می‌کند. بعد از اینکـه بعد از مرحله بازیابی سریع وارد پیشگیری از ازدحام شد، شروع به افـزایش پنجـره ازدحام با منحنی مقعر تابع افزایش می‌کند. تابع افزایش به این صورت تنظیم شده است که فلاتش را در W_{max} داشته باشد، پس رشد اندازه پنجره با تابع مقعر ادامه می‌یابد تا اندازه پنجره برابر W_{max} شود. تاثیر این مسئله این است کـه زمـانی کـه اندازه فعلی پنجره از ظرفیت در دسترس مسیر فاصله زیادی دارد، رشد پنجره زیاد است و اگر نزدیک ظرفیت در دسترس باشد (جایی که ما اتلاف قبلی را داشته‌ایـم، W_{max}) رشد پنجره را به آرامی‌کاهش می‌دهد. پس کمترین رشد پنجره را در نقطه اشباع دارد و بیشترین مقدار آن در زمانی است که از نقطه اشباع فاصله دارد که در این ناحیه اتلاف‌هایی که رخ می‌دهد بسیار کم است. این تابع مقعر، پنجره ازدحـام را مدت طولانی‌تری در نقطه اشباع نگه می‌دارد. همچنین از تابع محدب یـا خطـی تعادل و توازن بیشتری دارد زیرا این توابـع بیشـترین افـزایش پنجـره را در نقطـه اشباع دارند، پس بیشترین افزایش را در زمانی که اتلاف بسته رخ مـی‌دهـد دارنـد. این ویژگی باعث پایداری و در عین حال مقیاس‌پذیری شدن VECU می‌شود.

اگر رشد پنجره از مقدار ماکزیمم عبور کند، این به این معنی است کـه انـدازه پنجره باید از مقدار ماکزیمم فعلی بزرگ‌تر شده باشد و مقدار ماکزیمم جدید بایـد پیدا شود که در این شرایط تابع افـزایش وارد منحنـی محـدب مـی‌شـود. در ایـن مرحله، پنجره ابتدا به آرامی افزایش می‌یابد (آهسـته‌تـر از تـابع خطـی) تـا مقـدار ماکزیمم جدید را در همین نزدیکی پیدا کنـد. ایـن ویژگـی بـه پایـداری پروتکـل می‌افزاید زیرا اگر پروتکل در پیدا کردن پنجره ماکزیمم جدید دچار اشـتباه شـود، پنجره ماکزیمم بعدی را در نزدیکی مقدار ماکزیمم قبلی پیدا می‌کند، پـس مـدت طولانی‌تری در نقطه اشباع قبلی می‌ماند. اما تابع نمـایی بعـد از گذشـت مـدتی از شروع آهسته، اگر اتلافی رخ ندهد و مقدار ماکزیمم جدید را پیدا نکند، شـروع بـه افزایش سریع‌اندازه پنجره می‌کند.

تابع افزایش پنجره VECU از تابع زیر استفاده می‌کند:

$$W_t = C(t-K)3 + W_{max} \quad (۶)$$

C یک پارامتر ثابت و t زمان سپری شده از آخرین کاهش پنجره است. K نیـز برابر است با $[(W_{max}*\beta)/C]1/3$. زیرا زمانی که تابع افزایش پنجره شروع می‌شود، اندازه پنجره برابر $W_{max}*\beta$است. به عبارت دیگر ما می‌توانیم بگوییم کـه K مـدت زمانی است که تابع بالا می‌گیرد تا W را به W_{max} افزایش دهد در حالی که اتلاف بسته وجود نداشته باشد.

ممکن است شرایطی پیش بیاید که اندازه پنجره کمتـر از مقـداری باشـد کـه استاندارد TCP در زمان t بعد از آخرین اتـلاف بـه آن مـی‌رسـد. در ایـن شـرایط کارایی VECU کمتر از استاندارد TCP خواهد شـد. بـرای حـل ایـن مسـئله، در چنین شرایطی اندازه پنجره را به اندازه پنجره اسـتاندارد TCP تنظـیم مـی‌کنـیم. روند کار در این مرحله به صورت زیر می‌باشد:

زمانی که در فاز پرهیز از ازدحام یک ACK می‌رسد، اندازه پنجـره اسـتاندارد TCP در زمان سـپری شـده t را محاسـبه مـی‌کنـیم. انـدازه پنجـره در مکانیسـم AIMD با پارامتر افزایش α و پارامتر کاهش β از فرمول زیر بدست می‌آید [۳۰].

$$\frac{1}{RTT}\sqrt{\frac{\alpha}{2}\frac{2-\beta}{\beta}\frac{1}{p}} \quad (۷)$$

میانگین اندزه پنجره TCP با α=1 و β=0.5 برابر است با

$$\frac{1}{RTT}\sqrt{\frac{3}{2}\frac{1}{p}} \quad (۸)$$

پس برای اینکه فرمول اول مشابه فرمول‌اندازه پنجره TCP شود، α بایـد برابـر $\frac{3\beta}{2-\beta}$ باشد. اگر TCP اندازه پنجـره‌اش را در هـر RTT بـه‌انـدازه α افـزایش دهـد، می‌توانیم اندازه پنجره TCP در زمان سپری شده t را از فرمول زیر بدست آوریم:

$$W_{TCP}(t) = W_{max}(1-\beta) + 3\frac{\beta}{2-\beta}\frac{t}{RTT} \quad (۹)$$

بنابراین بر اساس مقدار cwnd تابع افزایش در سه حالت متفاوت خواهد بود:

۱) اگـر cwnd کـم‌تـر از $W_{TCP}(t)$ باشـد، از انـدازه پنجـره اسـتاندارد TCP استفاده می‌کنیم و cwnd با دریافت هر ACK به $W_{TCP}(t)$ تنظیم می‌شود.

۲) اگر cwnd از $W_{TCP}(t)$ بیشتر و از W_{max} کمتر باشد، پس در ناحیه مقعـر از تابع افزایش قـرار دارد. پـس در زمـانی کـه انـدازه فعلـی پنجـره از ظرفیـت در دسترس فاصله زیادی دارد، اندازه پنجره را سریع افزایش می‌دهد و زمانی که اندازه فعلی پنجره به نقطه اشباع نزدیک می‌شود، سرعت افزایش را کاهش می‌دهد.

۳) اگر cwnd از W_{max} بیشتر باشد، پس در ناحیه محدب از تابع افزایش قرار دارد که به‌این معنی است که ظرفیت در دسترس افزایش یافته اسـت و بـه دنبـال W_{max} جدید خواهیم بود.

۵-۳- رویکرد کاهش اندازه پنجره

یکی از اهداف این پروتکل تشخیص علت اتلاف بسـته‌هـا مـی‌باشـد. مـا مکانیسـم تشخیص علت اتلاف را در بخش‌های قبلی توضیح دادیم. اگر شبکه در زمـانی کـه یک رخداد اتلاف بسته دریافت کند در وضعیت عدم ازدحام باشـد، VECU فـرض می‌کند که اتلاف تصادفی رخ داده است و به آرامی اندازه پنجره را کاهش می‌دهد. در نتیجه از کاهش غیرضروری کارایی جلـوگیری مـی‌کنـد. بنـابراین مـا براسـاس تخمین وضعیت شبکه و علت اتلاف بسته‌ها از دو پارامتر کاهش ضربی (β) متفاوت استفاده می‌کنیم:

۱) اگر $N<\gamma$، این بدان معنی اسـت کـه اتـلاف تصـادفی رخ داده اسـت. پـس cwnd را با ضریب 4/5 کاهش می‌دهد.

۲) اگر $N>\gamma$، یعنی اتلاف به دلیل ازدحام رخ داده است و cwnd را بـا ضـریب 1/2 کاهش می‌دهد.

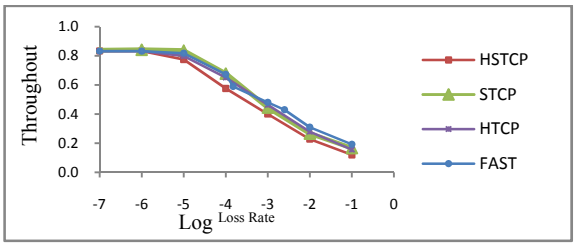
۶- پیاده‌سازی و نتایج آزمایشات

در این بخش به بررسی عملکرد الگوریتم پیشـنهادی در ایـن پـروژه مـی‌پـردازیم. تحلیل‌های انجام شده با استفاده از اسـتاندارد IEEE 802.11n صـورت گرفتـه و برای شبیه‌سازی از نرم‌افزار NS-2 استفاده شده است و معیارهای کارایی از جملـه میزان گذردهی، استفاده از لینک، پایـداری و عـدالت VECU را بـا پروتکـل‌هـای HTCP [۳۱]، HSTCP [۳۲]، FastTCP [۳۳] و STCP [۳۴] مقایسه کرده‌ایم.

ابعاد محیط شبیه‌سازی ۱۰۰۰×۱۰۰۰ بـا ۵۰ نـود مـی‌باشـد و نودهـا حرکـت می‌کنند و ثابت نیستند. پروتکل‌های لایه کـاربرد، مسـیریابی و MAC بـه ترتیـب FTP، AODV و 802.11n می‌باشند. مدت شـبیه‌سـازی ۵۰۰ ثانیـه اسـت و هـر آزمایش را ۴۰ بار تکرار کرده‌ایم. مدل قرارگیری نودها بـه صـورت تصـادفی اسـت ومیزان توقف گره‌ها پس از هر بار تحرک برابر ۱۰ ثانیه در نظر گرفته شـده اسـت. نودها از صف droptail با اندازه بافر ۲۴ استفاده مـی‌کننـد و سـایز بسـته‌هـا برابـر ۱۴۶۰ بایت می‌باشد.

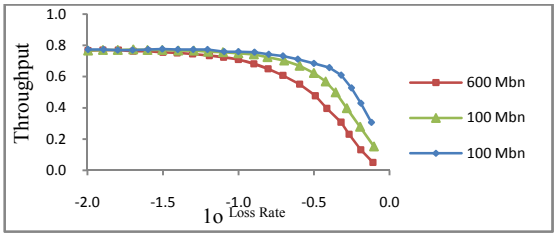
۶-۱- ارزیابی الگوریتم پیشنهادی در نرخ خطای بیتی متفاوت و سرعت متفاوت

همانطور که می‌دانیم یکی از ویژگی‌های شبکه‌های بی‌سیم، نرخ خطای بیتی بالا در این شبکه‌ها است. ما در شکل ۲ گذردهی پروتکل‌های TCP را در نرخ اتلاف بسته متفاوت مقایسه کرده‌ایم. همانطور که در شکل مشخص است، این پروتکل‌ها در نرخ اتلاف بسته بالا افت شدید گذردهی را تجربه می‌کنند. در نمودار بعدی گذردهی پروتکل ارائه شده در نرخ اتلاف بسته متفاوت و در سرعت‌های متفاوت نشان داده شده است.



شکل ۲- نمودار گذردهی در نرخ خطای بیتی متفاوت

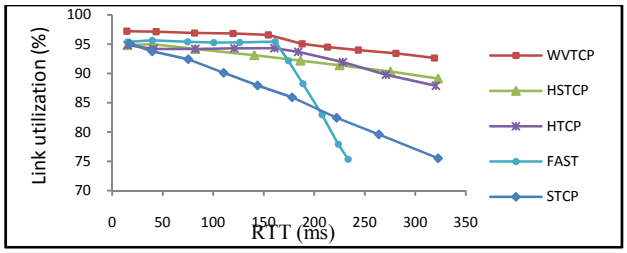
آنچه که از شکل ۲ و ۳ مشخص می‌شود این است که پروتکل پیشنهادی در شبکه‌های بی‌سیم با نرخ اتلاف بسته بالا گذردهی بهتری دارد. شکل ۳ همچنین نشان می‌دهد که در سرعت‌های بالاتر، اتلاف بسته، کارایی را بیشتر کاهش می‌دهد.



شکل ۳- نمودار گذردهیVECUدر سرعت‌های متفاوت و نرخ خطای بیتی متفاوت

۶-۲- ارزیابی الگوریتم پیشنهادی از نظر بکارگیری لینک

همانطور که انتظار داشتیم، VECU درصد بکارگیری لینک خوبی را نشان می‌دهد. از آنجا که این پروتکل دارای ویژگی TCPFairness نیز می‌باشد، ما می‌توانیم بگوییم که بهبودهایش را با گرفتن پهنای باند از دیگر اتصالات TCP بدست نمی‌آورد بلکه بهبودها نتیجه‌ی استفاده بهینه از پهنای باند استفاده نشده است.

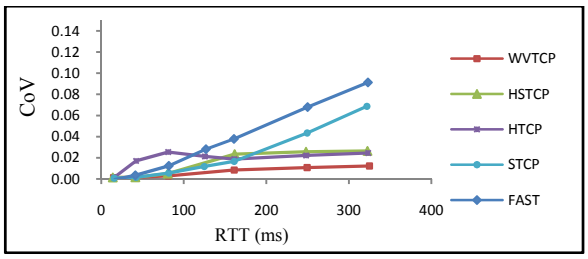


شکل۴- نمودار بکارگیری لینک در RTTهای متفاوت

همانطور که در شکل ۴ دیده می‌شود، VECU بکارگیری لینک خوبی دارد. همچنین در اکثر پروتکل‌ها، بکارگیری لینک در RTTهای بیشتر از 150ms کاهش می‌یابد. البته این مسئله در VECU ناچیز است.

۶-۳- ارزیابی الگوریتم پیشنهادی از نظر پایداری

پایداری پروتکل‌ها را براساس اندازه‌گیری میزان CoV نرخ انتقال بررسی کرده‌ایم. هرچه میزان CoV کمتر باشد، پایداری بیشتر خواهد بود. همانطور که در شکل ۵ دیده می‌شود، پروتکل VECU نسبت به دیگر پروتکل‌ها دارای پایداری بهتری است.

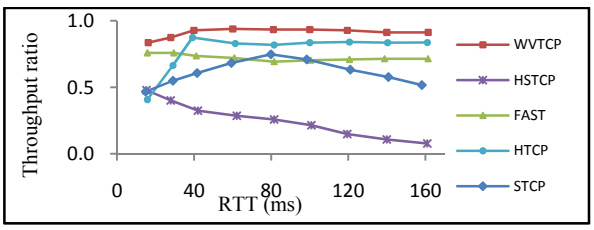


شکل ۵- نمودار ضریب تغییرات در RTTهای متفاوت

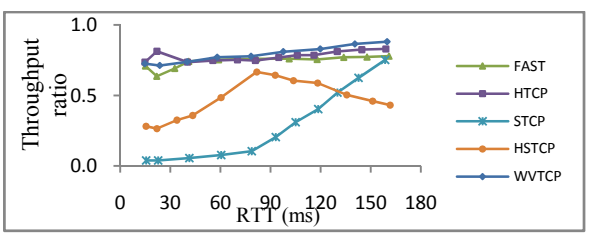
۶-۴- بررسی رفتار عادلانه پروتکل پیشنهادی

برای بررسی رفتار عادلانه پروتکل پیشنهادی، پارامترهای IntraProtocolFairness، Fairness RTT و TCP-Friendliness را اندازه گرفته‌ایم. Intra Protocol Fairness معیاری برای اندازه‌گیری به اشتراک‌گذاری عادلانه پهنای باند بین جریان‌هایی با پروتکل مشابه می‌باشد. بنابراین برای‌اندازه گیری این معیار در مورد هریک از پروتکل‌های TCP در هر بار اجرا دو جریان با پروتکل مشابه را اجرا کردیم.

RTT fairness معیاری برای‌اندازه گیری به اشتراک‌گذاری عادلانه پهنای باند بین جریان‌هایی با RTT متفاوت است. برای اندازه‌گیری این معیار، دو جریان با پروتکل مشابه را اجرا می‌کنیم که مقدار RTT یکی از این جریان‌ها را ثابت در نظر گرفته و مقدار RTT جریان دیگر را بین مقادیر صفر تا ۱۷۰ متغیر در نظر می‌گیریم.



شکل ۶- Intra Protocol Fairness



شکل ۷- RTT Fairness

همانطور که در شکل ۶ دیده می‌شود این پروتکل از جهت Intra ProtocolFairness خوب عمل می‌کند و پهنای باند را عادلانه بین جریان‌ها به اشتراک می‌گذارد. البته پروتکل HTCP نیز در RTTهای بالای ۴۰ms نیز خوب عمل می‌کند. پروتکل HSTCP نیز کمترین مقدار را دارد که این به علت سرعت همگرایی پایین آن است.

شکل ۷ بررسی معیار RTT Fairness را بررسی می‌کند. VECUو HTCP از این جهت خوب عمل می‌کنند و ظرفیت را به صورت عادلانه بین جریان‌ها با RTT متفاوت به اشتراک می‌گذارند. البته پروتکل FAST نیز خوب عمل می‌کند.

۷- نتیجه‌گیری

شبکه‌های بی‌سیم و سیار در دهه‌های گذشته رشد فوق‌العاده‌ای داشته‌اند و هنوز هم با سرعت زیادی در حال گسترش هستند .علیرغم آنکه شبکه‌های بی‌سیم امکانات فراوانی چون سهولت دسترسی و جابجایی در اختیار کاربران قرار داده‌اند، ولی کارایی و بهره‌وری این شبکه‌ها با نقطه مطلوب فاصله زیادی دارد. علت این امر این است که بیشتر پروتکل‌های کنونی شبکه، خاص شبکه‌های ثابت و سیمی طراحی شده‌اند و فرضیات آنها بر مبنای خصوصیات رسانه‌های سیمی است. زمانی که پیوندهای بی‌سیم به شبکه اضافه می‌شوند، رسانه بی‌سیم با خصوصیات خاص خود مثل عدم‌تقارن، پهنای باند کم، خطاپذیری و ...، موجب کاهش کارایی می‌شود. هدف روش‌های بهبود کارایی در شبکه‌های بی‌سیم و سیار، شناسایی این ویژگی‌ها و بررسی تاثیر آن‌ها در پروتکل‌های کنونی شبکه است و سپس تغییر این پروتکل‌ها بهگونه‌ای که شبکه‌های بی‌سیم و سیار بتوانند به کارایی بالائی دست یابند و سرویس‌هایی باکیفیت بهتر به کاربران سیار ارائه دهند. از طرفی پروتکل TCP نیز در این شبکه‌ها کارایی پایینی دارد. TCP یک پروتکل قابل اطمینان است و در صورتی که بسته‌های اطلاعاتی به‌درستی در اختیار فرستنده قرار نگیرند، فرستنده مجدد اقدام به ارسال اطلاعات می‌نماید. همچنین اتصال‌گراست و مکانیزم‌هایی برای کنترل جریان و کنترل ازدحام دارد. با این ویژگی‌ها و با توجه به اینکه اکثر کاربردهای موجود در اینترنت از این پروتکل استفاده می‌کنند، حذف این پروتکل و یا نادیده گرفتن مشکلات آن در شبکه‌های بی‌سیم امکان‌پذیر نمی‌باشد. ما در این مقاله سعی کردیم جمع‌بندی نسبتا کاملی از دلایل کارایی ضعیف پروتکل TCPدر شبکه‌های بی‌سیم بدست آوریم. ما این مشکلات را به دو دسته تقسیم کردیم؛ مشکلاتی که به دلیل ساختار پروتکل TCP باعث کارایی ضعیف آن می‌شوند و مشکلاتی که به دلیل خصوصیات و ویژگی‌های محیط بی‌سیم بر کارایی TCP تاثیر منفی می‌گذارند. در ادامه نیز روش‌های بهبود کارایی TCP را مورد مطالعه قرار داده و در سطح اول آنها را از نقطه نظر کابرد و پیاده‌سازی TCP به دو دسته تقسیم کردیم. اگر از جنبه کاربردها به این مسئله نگاه کنیم، ما چند محیط بی‌سیم داریم و TCP را می‌توان متناسب با مشکلات هر محیط طراحی کرد. رایج‌ترین شبکه‌های بی‌سیم، شبکه‌های ماهواره‌ای، شبکه‌های ad hoc و زیرساخت‌های بی‌سیم عمومی مانند شبکه‌های محلی بی‌سیم و سلولی هستند. از نقطه نظر پیاده‌سازی و اجرا نیز الگوریتم‌های TCP بی‌سیم را به پنج دسته تقسیم کردیم. حالت تقسیم، کنترل ازدحام انفعالی و کنترل ازدحام فعالانه، لایه پیوند داده و همکاری بین لایه‌ای. همچنین برای هر کدام از دسته‌ها نیز مثال‌هایی را آورده و عملکرد و مزایا و معایب هر کدام را بررسی کرده و متناسب با مشکلاتی که برای TCP مطرح کردیم به مقایسه این روش‌ها پرداختیم. در پایان یک پروتکل جدید به نام VECU ارائه دادیم که این پروتکل متناسب با ویژگی‌های شبکه بی‌سیم طراحی شده است. یک مشکل اساسی در شبکه‌های بی‌سیم عدم تشخیص علت ازدحام بسته‌ها است. این پروتکل با مانیتور کردن وضعیت شبکه، علت اتلاف بسته‌ها را تشخیص می‌دهد. VECU براساس وضعیت شبکه، ضریب کاهش اندازه پنجره را تعیین می‌کند. همچنین برای افزایش پنجره از یک تابع چند جمله‌ای درجه سه بر حسب زمان گذشته از آخرین اتلاف استفاده کرده است. این تابع باعث می‌شود که رشد پنجره ازدحام مستقل از RTT شود و به جریان‌های رقیب با RTTهای متفاوت این امکان را می‌دهد که مستقل از RTT، رشد یکسانی داشته باشند. همچنین کمترین رشد پنجره را در نقطه اشباع دارد و بیشترین مقدار آن در زمانی است که از نقطه اشباع فاصله دارد و این ویژگی اتلاف‌های توده‌ای را نیز کم می‌کند. به‌علاوه این پروتکل پایدار و مقیاس‌پذیر است و بکارگیری لینک آن بالا می‌باشد.

مراجع

[1] J. Postel, "Transmission control protocol," RFC 793, 1981.

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

[3] C. Siva Ram Murthy, and B. S. Manoj, Ad Hoc Wireless Networks: Architectures and Protocols, New Jersey: Prentice Hall, 2004.

[4] H. Hari Balakrishnan, "Challenges to Reliable Data Transportover Heterogeneous Wireless Networks," Ph.D. dissertation, Univ. California, Berkeley, CA, 1998.

[5] M. Moustafa, "TCP over Mobile Ad-Hoc Access Networks," Master Thesis, Carleton University, Ottawa, December 2002.

[6] V. Badarla, and C. Siva Ram Murthyb, "Learning-TCP: A stochastic approach for efficient update in TCP congestion window in ad hoc wireless networks," J. Parallel Distrib. Comput. vol. 71, pp. 863–878, 2011.

[7] M. C. Chan, and R. Ramjee, "Improving TCP/IP Performance over Third-Generation Wireless Networks," IEEE Trans. Mobile Computing, vol. 7, no. 4, 2008.

[8] X. Chen, H. Zhai, J. Wang, and Y. Fang, "TCP Performance over Mobile Ad-hoc Networks," Canadian Journal of Electrical and Computer Engineering, vol. 29, pp. 129–134, 2004.

[9] V. Anantharaman, S. J. Park, K. Sundaresan, and R. Sivakumar, "TCP Performance over Mobile Ad-hoc Networks: a Quantitative Study," Wireless Communication and Mobile Computing Journal, vol. 4, pp. 203–222, 2004.

[10] H. Balakrishnan, V. N. Padmanabhan, and R. H. Katz, "The Effects of Asymmetry on TCP Performance," Proc. ACM/IEEE Mobicom, pp. 77–89, 1997.

[11] Y. Tian. K. Xu, and N. Ansari, "TCP in Wireless Environments: Problems and Solutions," IEEE Radio Communications, pp. 0163-6804, 2005.

[12] X. Chen, H. Zhai, J. Wang, and Y. Fang, A Survey on Improving TCP Performance over Wireless Networks, in Resource Management in Wireless Networking, Springer, 2003, sec. 6, pp 657-695.

[13] I. F. Akyildiz, G. Morabito, and S. Palazzo, "TCP-Peach: A New Congestion Control Scheme for Satellite IP Networks," IEEE/ACM Trans. Net., vol. 9, no. 3, pp. 307–21, 2001.

[14] J. Liu, and S. Singh, "ATCP: TCP for Mobile Ad Hoc Networks," IEEE JSAC, vol. 19, no. 7, pp. 1300–1315, 2001.

[15] T. Goff, and et. al., "Freeze-TCP: A True End-to-end Enhancement Mechanism for Mobile Environments," Proc. IEEE INFOCOM, pp. 1537–45, 2000.

[16] A. Bakre, and B. R. Badrinath, "I-TCP: Indirect TCP for Mobile Hosts," Proc. ICDCS, pp. 136–43, 1995.

[17] R. Yavatkar, and N. Bhagawat, "Improving End to End Performance of TCP over Mobile Networks," Proc. Workshop on Mobile Computing Systems and Applications, 1994.

[18] M. Mathis, and et. al., "TCP Selective Acknowledgment Options," RFC 2018, 1996.

[19] K. Pentikousis, H. Badr, and B. Kharmah, "TCP with ECN: Performance Gains for Large TCP Transfers," Department of Computer Science, SUNY at Stony Brook, 2001.

[20] Ch. Liu, and R. Jain, "Approaches of Wireless TCP Enhancement and a New Proposal Based on Congestion Coherence," Proc. Annual Hawaii International Conference on System Sciences, 2003.

[21] L. Brakmo, and L. Peterson, "TCP Vegas: End to End Congestion Avoidance on a Global Internet," IEEE JSAC, vol. 13, no. 8, pp. 1465-80, 1995.

[22] C. P. Fu, and S. C. Liew, "TCP Veno: TCP Enhancement for Transmission over Wireless Access Networks," IEEE JSAC, vol. 21, no. 2, pp.216–28, 2004.

[23] C. Casetti, and et. al., "TCP Westwood: Bandwidth Estimation for Enhanced Transport over Wireless Links," Proc. ACM Mobicom, pp. 287–97, 2001.

[24] K. Xu, Y. Tian, and N. Ansari, "TCP-Jersey for Wireless IP Communications," IEEE JSAC, vol. 22, no. 4, pp. 747–56, 2004.

[25] C. Parsa, and J. J. Garcia-Luna-Aceves, "Improving TCP Performance over Wireless Networks at the Link Layer," Mobile Networks and Applications, 1999.

[26] S. Vangala, and M. Labrador, "Performance of TCP over wireless networks with the Snoop protocol," Proc. Local Computer Networks, 2002.

[27] N. H. Vaidya, M. Mehta, C. Perkins, and G. Montenegro, "Delayed Duplicate Acknowledgement: a TCP unaware approach to improve performance of TCP over wireless," Technical Report 99-003, Computer Science Dept., Texas A&M University, 1999.

[28] D. Sun, and H. Man, "ENIC – An Improved Reliable Scheme for Mobile Ad Hoc Networks," Proc. IEEE Globe COM Conference, 2001.

[29] M. Chinta, A. Helal, and Ch. Lee, "ILC-TCP: An Interlayer Collaboration Protocol for TCP Performance Improvement in Mobile and Wireless Environments," IEEE Wireless Communications and Networking, vol. 2, pp. 1004-1010, 2003.

[30] S. Floyd, M. Handley, and J. Padhye, "A Comparison of Equation-Based and AIMD Congestion Control," 2000.

[31] D. Leith, and R. Shorten, "H-TCP protocol for high-speed long distance networks," Proc. International Workshop on Protocols for Fast Long- Distance Networks, pp. 16–17, 2004.

[32] S. Floyd, "Highspeed TCP for large congestion windows," Internet draft, draft-floyd-tcp-highspeed-01.txt, Available at: http://www.icir.org/floyd/hstcp.html, 2003.

[33] D. X. Wei, C. Jin, S. H. Low, and S. Hedge, "FAST TCP: Motivation architecture, algorithm and performance," IEEE/ACM Trans. Netw., vol. 14, no. 6, pp. 1246–1259, 2006.

[34] T. Kelly, "Scalable TCP: Improving Performance in High Speed Wide Area Networks," Computer Communication Review, vol. 33, No. 2, pp. 83-91, 2003.

مریم شفیعی فارغ‌التحصیل رشته مهندسی فناوری اطلاعات از دانشکده مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران در سال ۱۳۹۲ است. ایشان دوره کارشناسی خود را در سال ۱۳۸۸ در رشته مهندسی فناوری اطلاعات دانشگاه صنعتی اصفهان به اتمام رسانیده‌اند. شبکه‌های بی‌سیم سرعت بالا، شبکه‌های حسگر بی‌سیم، داده‌کاوی و طراحی و پیاده‌سازی روتر از زمینه‌های تحقیقاتی مورد علاقه ایشان است.

آدرس پست‌الکترونیکی ایشان عبارت است از:

m.shafiei@ut.ac.ir

ناصر یزدانی در سال ۱۳۸۸ در یکی از روستاهای آذربایجان شرقی متولد گردید. ایشان در سال ۱۳۶۴ بعد از بازگشائی دانشگاه‌ها، در رشته کارشناسی مهندسی کامپیوتر، گرایش نرم‌افزار فارغ التحصیل شد. در آذرماه ۱۳۶۸ با بورس وزارت علوم و تحقیقات و فن‌آوری جهت ادامه تحصیل در رشته کارشناسی ارشد کامپیوتر در دانشگاه کیس وسترن رزرو (Case Western Reserve University) آمریکا عازم ایالت اوهایو، شهر کلیولند شد. وی از این دانشگاه موفق به اخذ درجه دکتری در رشته کامپیوتر، در زمینه پایگاه داده‌ها شد.

دکتر یزدانی سپس در دانشگاه واشنگتن در ایالت میسوری و چند شرکت تازه تاسیس عمدتاً در زمینه طراحی سوئیچ و شبکه در کشور امریکا مشغول کار و تحقیق شد و دو مورد مالکیت معنوی در زمینه سریعتر کردن اینترنت به ثبت رساند. در سال ۱۳۷۹ ایشان به کشور بازگشت و همزمان در دانشگاه‌های تهران و امیرکبیر مشغول کار گردید. دکتر یزدانی هم‌اکنون به‌عنوان استاد تمام رشته کامپیوتر در دانشکده برق و کامپیوتر دانشگاه تهران مشغول تدریس و تحقیق

می‌باشد. در این مدت ایشان بیش از ۵۰ دانشجوی کارشناسی ارشد و ۶ دانشجوی دکتری را سرپرستی کرده و بیش از ۲۰۰ مقاله علمی در مجلات علمی و کنفرانس‌های بین‌المللی به چاپ رسانده است. وی چند پروژه موفق در زمینه طراحی و پیاده سازی سوئیچ و روتر در دانشگاه تهران سرپرستی کرده و اخیرا یک ثبت اختراع هم در دست اقدام دارد.

آدرس پست‌الکترونیکی ایشان عبارت است از:

yazdani@ut.ac.ir

مسعود رهگذر در سال ۱۳۸۵ مدرک کارشناسی را در رشته مهندسی برق از دانشگاه صنعتی شریف، و در سال‌های ۱۳۶۱ و ۱۳۶۵ به ترتیب مدارک کارشناسی ارشد و دکتری خود را در رشته مهندسی کامپیوتر گرایش نرم‌افزار از دانشگاه PARIS-VI دریافت نموده است. ایشان در سال ۱۳۸۷ در دانشگاه تهران مشغول کار گردید و هم‌اکنون دانشیار دانشکده مهندسی برق و کامپیوتر دانشکده فنی دانشگاه تهران است. زمینه‌های تحقیقاتی ایشان شامل پیاده‌سازی و بهبود پایگاه داده‌ها XML، داده‌کاوی و سیستم‌های ذخیره‌سازی و بازیابی اطلاعات، تعامل انسان- کامپیوتر، نرمال‌سازی سیستم‌های پایگاه داده (مهندسی معکوس و بهینه‌سازی) و مدرن‌سازی سیستم‌های قدیمی هستند.

آدرس پست‌الکترونیکی ایشان عبارت است از:

rahgozar@ut.ac.ir

اطلاعات بررسی مقاله:

تاریخ ارسال: ۱۳۹۳/۱۲/۲۰
تاریخ اصلاح: ۱۳۹۴/۰۲/۱۳
تاریخ قبول شدن: ۱۳۹۴/۰۵/۲۲
نویسنده مرتبط: مریم شفیعی، دانشکده برق و کامپیوتر، دانشگاه تهران، تهران، ایران.

[1] Aggressive
[2] Deterministic
[3] Deterministic Actions
[4] Explicit Route Error Notification
[5] Explicit Route Recovery Notification