یک بررسی بر پیشرفت های اخیر در پروتکل های لایه انتقال

چکیده

در طول سال‌ها، اینترنت با تکنولوژی‌های ارتباطی در دسترس جدید، برای شبکه‌ها و دستگاه‌های ثابت و سیار، رشد چشمگیر از نظر عملکرد، افزایش پایدار نرخ داده‌های دسترس غنی شده است. جامعه پژوهش اینترنت هنوز هم برای تکامل پروتکل‌های لایه انتقال از نظر تطبیق با قابلیت‌های شبکه‌های مدرن، به منظور بهره برداری کامل از مزایای تکنولوژی‌های ارتباطی جدید تلاش می‌کند. این مقاله بر نوآوری‌های اصلی مربوط به پروتکل‌های انتقال که اخیرا مطرح شده اند تمرکز میکند، و سه روند پژوهشی اصلی را شناسایی می‌کند: (1) تکامل الگوریتم‌های کنترل ازدحام، برای هدف قراردادن عملکرد بهینه در سناریوهای چالش برانگیز، با استفاده از تکنیک‌های یادگیری ماشین؛ (2) پیشنهاد پروتکل‌های انتقال جدید برند، که جایگزینی برای پروتکل کنترل انتقال (TCP) است و در یک فضای کاربری اجرا می‌شود؛ و (3) مقدمه ای بر قابلیت‌های چند مسیری در لایه انتقال.

1. مقدمه

تکنولوژی‌های ارتباطی که دسترسی و ارتباطات بک هال[1] را به اینترنت فراهم می‌کنند از دهه 1980 به بعد، از زمانی که پروتکل‌هایی که بخشی از پشته TCP/IP بوده اند برای اولین بار معرفی شدند، به صورت چشمگیری تغییر کردند [1]–[3]. انتظار می‌رود که در سال 2021 ترافیک تولید شده از دستگاه‌های موبایل از ترافیک تولید شده توسط دستگاه‌های دسکتاب و سرورها پیشی بگیرد [4]. استانداردهای ارتباطی جدید مطرح شدند، و در طی چند سال گذشته به بازار وارد شدند. با افزایش تحریک آمیز تقاضای ترافیک وب و چند رسانه ای ها، شبکه‌های موبایل و ثابت

[1] backhaul

به سرعت در حال تکامل هستند. 3GPP NR [5]، با نرخ داده به شدت بالا با تاخیر بسیار کم برای دستگاه‌های 5G آینده همراه است، و به طور مشابه، استاندارد IEEE 802.11 استقرارهای به شدت متراکم و توان عملیاتی چند گیگابایت در ثانیه را هدف قرار می‌دهد [8]. دستگاه‌های مدرن قابلیت ارتباط با شبکه‌های ناهمگن را دارا هستند [9]، و شبکه‌های ثابت از تکنولوژی‌های نوری و شبکه‌های نرم افزاری تعریف شده[2] (SDN) برای نرخ‌های بی سابقه و تاخیر کم استفاده می‌کنند [10].

قابلیت‌های در حال افزایش شبکه ایجاد نوع جدیدی از اپلیکیشن‌ها را امکان پذیر کرده است، رشد نمایی ترافیک چند رسانه ای یا بلادرنگ [4] بدون پیشرفت‌های تکنولوژیکال اخیر غیر ممکن بوده است. با پیشرفت شبکه به سمت 5G، نوع جدیدی از اپلیکیشن ها، مانند واقعیت افزوده[3] (AR) و واقعیت مجازی[4] (VR) یا رانندگی خودکار، نیازمند شبکه‌های بیشتر و محدودیت‌های کیفیت خدمات (QoS) سفت و سخت تر هستند. این، با افزایش ناهمگونی شبکه ها، نقش پروتکل انتقال را مهم تر، و چالش برانگیز تر کرده است. براستی، عملکرد انتها به انتها و کیفیت تجربه[5] (QoE) کاربران تا حد زیادی به تعامل میان اپلیکیشن ها، لایه انتقال و شبکه اصلی مربوط است [11]. به خصوص، لایه انتقال، که مسئول مدیریت ارتباطات انتها به انتها بر روی چند هاب شبکه است، باید به منظور بهره مندی کامل کاربران از نوآوری‌های فوق الذکر تکامل یابد. به هر حال، تعدادی از فاکتورها یا عامل‌ها مانع از این می‌شوند که راه حل‌های جدید در لایه انتقال، در سال‌های اخیر به صورت گسترده پذیرفته شوند، جامعه پژوهشی قصد دارد که با این محدودیت‌ها مقابله کند و راه حل‌های نوآورانه را به منظور تاثیر گذاری قابل توجه بر عملکرد اینترنت شناسایی کند.

به طور خاص، استقرار لایه‌های انتقال جایگزین، مانند پروتکل انتقال کنترل جریان[6] (SCTP) [12]، با استفاده گسترده از جعبه‌های میانی[7] کاهش یافته است [13]، که اغلب بسته‌هایی که از بسته‌های پروتکل کنترل انتقال (TCP) و یا پروتکل دیتاگرام کاربر (UDP) [15] متفاوت هستند را دور می‌اندازند. علاوه بر این، سوکت واسط برنامه

[2] Software-Defined Networking
[3] Augmented Reality
[4] Virtual Reality
[5] Quality of Experience
[6] Stream Control Transmission Protocol
[7] middleboxes

نویسی اپلیکیشن[8] (API) به صورت جهانی استفاده شده است [16]، بنابراین گزینه‌های رابط بین اپلیکیشن و لایه‌های انتقال را برای آنچه API پشتیبانی می‌کند تغییر میدهد. سرانجام، گسترده ترین سیستم‌های عملیاتی توابع انتقال (TCP&UDP) را در کرنل اجرا می‌کنند، و استقرار راه حل‌های جدید را سخت می‌سازد. این عناصر تعریف می‌کنند که چه چیزی مبنای لایه انتقال گفته می‌شود [16]، پدیده‌ای که محققین و توسعه دهندگان را به سمت استفاده از TCP قدیمی (برای ترافیک قابل اعتماد و کنترل ازدحام) سوق میدهد، حتی اگر این بهترین پروتکل عملکردی برای مورد استفاده مطلوب نباشد. TCP در برخی از سناریوهای خاص با برخی از مشکلات عملکردی مواجه است، مانند لینک‌های بی سیم با تغییرپذیری بالا [17]، بلاکینگ Head of Line (HoL) با ترافیک وب، و bufferbloat [20].

این بررسی بر سه جهت پژوهش لایه انتقال (پروتکل انتقال جدید، نوآوری‌های کنترل ازدحام، و رویکردهای چند مسیری) که در 15 سال اخیر برای کل مسائل فوق الذکر ظهور کرده اند، تمرکز میکند. پروتکل‌های ازدحام جدید برای هدف قراردادن تاخیر کم و استفاده کامل از پهنای باند پیشنهاد شدند. پروتکل‌های انتقال جدید توسط نیروی کار مهندسی اینترنت (IETF)، با نوآوری‌های فنی مانند قابلیت‌های چند مسیری، برای بررسی واسط‌های متعدد موجود در تلفن‌های هوشمند مدرن، کامپیوترها، و سرورها و اجرای فضای کاربر، برای غلبه بر آنچه که مانع از پذیرش کامل الگوریتم‌های جدید در لایه انتقال می‌شود، بحث کردند. بنابراین، در این بررسی، ابتدا پروتکل‌های انتقال جدید اصلی را که توسط IETF از سال 2006 پیشنهاد شدند و استاندارد سازی شدند را بررسی کردیم؛ ما یک بررسی مختصر بر SCTP و پروتکل کنترل ازدحام دیتاگرام (DCCP) ارائه دادیم، و سپس به مبحث‌های دیگر پرداختیم، مانند پروتکل ارتباطات اینترنتی UDP سریع (QUIC). سپس، ما بر پژوهش‌های انجام شده بر کنترل ازدحام نگاهی انداختیم. هر دو مکانیزم‌های جدید را با استفاده از رویکردهای کلاسیک تشریح می‌کنیم، مانند پهنای باند گلوگاه و زمان انتشار دور چرخشی[9] (BBR) و تاخیر کم (LoLa) برای TCP، و برخی از پیشنهادات جدید برای استفاده از تکنیک‌های یادگیری ماشین برای کنترل ازدحام. سرانجام، روند سومی که در این بررسی بحث شده است به پذیرش

[8] Application Programming Interface
[9] Round-trip propagation time

راه حل‌های چند مسیری در لایه انتقال، اساسا با TCP چند مسیره (MPTCP) [26] ,[25]، مربوط است اما به بسط‌های چند مسیره SCTP [27]، QUIC [28]، و با پروتکل تجمعی انتها به انتهای کنترل تاخیر (LEAP) مربوط است.

هدف ما در این بررسی ارائه نقطه نظرهای جامع در پژوهش‌های به روز پروتکل‌های انتقال برای خواننده‌های علاقمند است، نقطه نظرهایی که در دیگر بررسی‌های اخیر که بر استخراج، انتقال‌های چند مسیره، یا الگوهای کنترل ازدحام برای MPTCP یا در مراکز داده تمرکز می‌کنند، دیده نمی‌شود.

باقی مقاله به شرح زیر سازمان یافته است. در بخش 2 مسائل و محدودیت‌های اصلی پروتکل‌های انتقال را در شبکه‌های مدرن تشریح کردیم. در بخش 3 سه پروتکل انتقال به تازگی پیشنهاد شده را تشریح کردیم. در بخش 4 آخرین پروپوزال‌ها از نظر الگوریتم‌های کنترل ازدحام را برای TCP و دیگر پروتکل‌های انتقال گزارش کردیم، و آن‌ها را به به عنوان الگوریتم‌های مبتنی بر گم شدن بسته، تاخیر یا ظرفیت دسته بندی کردیم. ما بر مکانیزم‌های ترکیبی، الگوریتم‌های مبتنی بر یادگیری ماشین و رویکردهای بین لایه‌ای بحث کردیم. در بخش 5 بر رویکردهای متفاوت برای پروتکل‌های انتقال چند مسیره اخیرا پیشنهاد شده بررسی‌هایی انجام دادیم. سرانجام بخش 6 نتیجه گیری مقاله است و جهات پژوهشی آینده را مطرح میکند. یک لیست جامع از کلمات اختصاری نیز در پایان مقاله آورده شده است.

2. محدودیت‌های لایه انتقال در شبکه‌های مدرن

همانطور که در بخش 1 گفته شد، پروتکل‌های لایه انتقال یک دیدگاه ارتباطی انتها به انتها دارند؛ آن‌ها هر هاب را به صورت انفرادی در نظر نمی‌گیرند، اما تنها یک پیوند منطقی بین دو نقطه انتهایی در نظر می‌گیرند. به همین دلیل، مهم ترین لینک‌ها در ارتباط آهسته ترین لینک‌ها هستند، که اصطلاحا به آن‌ها گلوگاه گفته می‌شود. سرویس ارائه شده توسط TCP، پروتکل انتقال استاندارد بالفعل، می‌تواند به عنوان یک لوله واحد با ظرفیت لینک گلوگاه و یک زمان دور چرخشی (RTT) اصلی مدل شود، مانند زمان از لحظه ای که فرستنده یک بسته را ارسال میکند تا زمانی که تایید متناظر آن را دریافت کند است.

به هر حال، ویژگی‌های خاص لینک‌های منفرد و رفتار لایه‌های پایین تر بر رفتار TCP، و موارد دیگر پروتکل انتقال تاثیر دارد: چندین ویژگی لینک شامل ارتباطات انتها به انتها (مانند تاخیر، گم شدن بسته، وضعیت بافر و اندازه، نوسانات ظرفیت است) که می‌تواند بر عملکرد لایه انتقال تاثیر گذارد [34]. به منظور غلبه بر این مسئله، مکانیزم‌های کنترل ازدحام پیچیده تری پیشنهاد شدند، که فراتر از اصل افزایش جمعی کاهش ضربی[10] (AIMD) ساده هستند. ما فلسفه‌های طرح اصلی را در بخش 4 به صورت دقیق تر توضیح می‌دهیم. در حقیقت، محققین در مورد استفاده گسترده از TCP به عنوان یک راه حل مناسب برای همه سوال پرسیدند، چرا که همه این فاکتورها باعث شوند که مسائل عملکردی بیشتری بروز کنند. برخی از این مسائل مسئله‌های اصلی انتزاع لایه انتقال هستند، در حالی که موارد دیگر به ویژگی‌های خاص پروتکل مربوط هستند و می‌توانند با طراحی درست پروتکل اجتناب شوند. این بخش یک دیدگاه مختصر بر مهم ترین مسائل ارائه داده است، باقی مقاله نیز به بحث بر راه حل‌های متعددی که برای بررسی این چالش‌ها پیشنهاد شدند تخصیص یافته است.

A. Bufferbloat

همانطور که در بخش 4 بحث خواهیم کرد، مکانیزم‌های کنترل ازدحام برای تنظیم مقدار داده ارسال شده از یک دیدگاه انتزاعی بر شبکه اصلی استفاده میکنند. همانطور که در [20] نشان داده شد، این انتزاع در برخی از نمونه در ارائه اطلاعات دقیق در مورد لینک‌های ارتباط دهنده دو هاست با شکست مواجه می‌شود و به کاهش عملکرد منجر می‌شود. به خصوص، زمانی که بافرهای بزرگ قبل از گلوگاه قرار گیرند، تا از گم شدن بسته‌ها جلوگیری شود، سپس مکانیزم‌های پروب TCP مبتنی بر گم شدن اشغال شدن صف را افزایش می‌دهند، بنابراین باعث تاخیر می‌شود. علاوه بر این، از انجا که نسخه‌های اجرایی فعلی TCP نرخ ارسال را تا زمانی که اولین بسته گم شود افزایش می‌دهند، آن‌ها اغلب ظرفیت کانال را تحت شعاع قرار میدهند، ازدحام را افزایش میدهند و در زمانی که صف پر می‌شود باعث

[10] Additive Increase Multiplicative Decrease

انتقال‌های مجدد می‌شوند. دیگر پروتکل‌ها مانند QUIC, SCTP, and DCCP نیز با مسئله مشابهی مواجه هستند، چرا که مسئله کنترل ازدحام یک مشکل شایع بافرهای بزرگ است و مسئله ای نیست که به پروتکل مربوط باشد.

این پدیده، که به عنوان Bufferbloat شناخته شده است، QoS اپلیکیشن را کاهش می‌دهد، به خصوص زمانی که جریان انتقال ویدیو یا فایل بافر را با جریان‌های مرور وب به اشتراک بگذارند، و این مسئله در سال‌های اخیر با توجه به استراتژی‌های طراحی منع گم شدن بسته بدتر شده است، چرا که این استراتژی‌های بافرهای بزرگی را در ابتدای لینک‌های با ظرفیت دسترسی کم قرار میدهند [35].

پژوهش در این عرصه حل این مسئله را با تکنیک‌های مدیریت صف فعال[11] (AQM) محلی یا جریان انتها به انتها و کنترل ازدحام برای پروتکل‌های انتقال هدف قرار می‌دهد.

مسئله، در حالی که تشخیص آن سخت نیست، بدون یک هزینه سربار قابل توجه به سختی حل می‌شود [36]. پروتکل کنترل ازدحام در نقطه انتهایی ممکن است متفاوت باشد، و AQM را پیچیده تر کند؛ این اطلاعات اغلب برای روترها هم در دسترس نیستند، به همین دلیل روترها هم ممکن است قادر به پیش بینی دور انداختن بسته‌ها در شرایط ازدحام نباشند، و همین الگوریتم‌ها را به شدت پیچیده تر میکند و نسبت به تنظیم پارامترها حساس تر میکند.

AQM یک ایده تازه نیست: تکنیک‌های بیشماری پیشنهاد شدند، مانند دور انداختن زود هنگام تصادفی[12] (RED)، و ما به خواننده پیشنهاد می‌کنیم که برای کسب اطلاعات بیشتر در مورد این موضوع به [37][36] مراجعه کند. با توجه به این پروپوزال‌های بسیار، آن‌ها با پذیرش محدودی مواجه شده اند، که تا حدی با توجه به مسائل ذکر گفته شده بالا در تنظیم پارامترها و هزینه محاسباتی الگوریتم‌ها بوده است [35]. اخیرا، روش‌های جدید و آسان تری برای تنظیم و استقرار الگوهای AQM، مانند مدیریت تاخیر کنترل شده[13](CoDel)، در محصولات تجاری متعدد پذیرفته شده است. CoDel یک الگوریتم AQM است که تاخیر بافرینگ را با مانیتورینگ تاخیر صف D در یک بازه (معمولا

[11] Active Queue Management
[12] Random Early Dropping
[13] Controlled Delay Management

100ms) و گم شدن بسته‌ها در زمانی که حداقل مقدار D بزرگتر از 5ms ثانیه است، محدود می‌کند. با این وجود همانطور که در بخش D قسمت 3 بحث می‌کنیم، مشابه Bufferbloat به دامنه‌های بی سیم مربوط است.

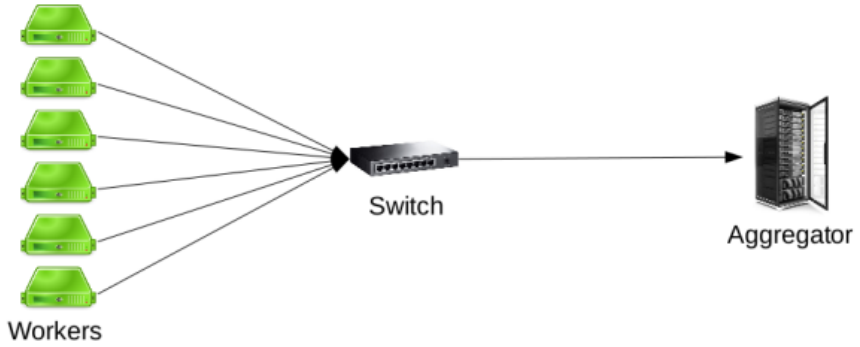
B. مسئله Incast

مراکز داده نواحی محدودی هستند که شامل سرورها و سیسم‌های نظارت کننده فعالیت‌های سرور، ترافیک و عملکرد وب هستند. تبادل داده بین سرورها معمولا بر اساس APIهای مبتنی بر پروتکل انتقال ابرمتن (HTTP) است، که از TCP به عنوان پر کاربردترین پروتکل انتقال در مراکز داده استفاده می‌شود. برخی از فعالیت ها، مانند مهاجرت‌های ماشین مجازی، نیز حجم بالایی از ترافیک را بین سرورها ایجاد میکند. بنابراین، لینک‌ها در مراکز داده معمولا پهنای باند بالا و تاخیر کمی دارند، در عین حال که سوئیچ‌ها بافرهای کوچکی دارند، برخلاف آنچه که در لینک‌های دسترسی، که در بخش قبل گفته شد، رخ می‌دهد.

فریم ورک‌های رایانش ابری به صورت گسترده در مراکز داده بزرگ قرار دادند و بارهای ترافیک بسیار بالایی را تولید می‌کند. برای مثال، MapReduce (که از الگوی طراحی تجمعی/بخش بندی استفاده می‌کند) [40] یا PageRank (استفاده شده برای جستجوی وب) [41] اغلب شامل الگوهای ترافیک چند به یک هستند، که در آن‌ها کارگران متعدد داده‌هایی را همزمان به یک گره تجمعی واحد ارسال میکنند، که در شکل 1 نشان داده شده است. در این سناریوی چند به یک، اگر همه جریان‌های incast متعدد از یک سوئیچ عبور کنند، بافر آن ممکن است کافی نباشد، و به ازدحام منجر شود. مکانیزم بازیابی بسته‌های گم شده TCP کمتر به درد می‌خورد، و با تایم اوت‌های متعدد مواجه می‌شود و باعث تنزل توان عملیاتی و تاخیرهای طولانی می‌شود [42].

تلاش‌های بسیاری برای تحلیل و حل این مسئله، که مسئله Incast گفته می‌شود، و عملکرد شبکه و تجربه کاربر را تضعیف می‌کند، صورت گرفته است [43]. تحلیل تخمین دقیق توان عملیاتی را می‌توانید در [42] و [44] ببینید. راه حل‌ها ممکن است به چهار گروه ذکر شده در [42] دسته بندی شوند: (1) تنظیم پارامترهای سیستم، مانند

غیرفعال کردن شروع آهسته[14] برای اجتناب از انباشتگی و سرریز ناگهانی بافر که باعث تایم اوت انتقال‌های مجدد می‌شود؛ (2) ارتقای طراحی الگوریتم سمت کاربر و داخلی شبکه، برای کاهش هدر رفت پهنای باند، حداقل سازی تایم اوت‌های انتقال مجدد و کاهش تعداد بسته‌های گم شده، و بهبود بازیابی سریع بسته‌های گم شده [39]؛ جایگزینی الگوریتم‌های کنترل ازدحام مبتنی بر گم شدن بسته با پیاده سازی‌های بهتر که اندازه پنجره ازدحام را برطبق تاخیر اندازه گرفته شده از RTTها تنظیم می‌کنند؛ مانندVegas؛ و (4) طراحی الگوریتم‌های کاملا جدید برای این محیط خاص، مانند Data Center TCP (DCTCP) [45]، که از اعلان ازدحام صریح[15] (ECN) برای ارائه متدهای کنترلی مبتنی بر پنجره، یا IATCP [46]، یک رویکرد مبتنی بر نرخ استفاده می‌کند که مجموع تعداد بسته‌های تزریق شده برای برآورده کردن محصول تاخیر پهنای باند[16](BDP) شبکه را محاسبه می‌کند [42].



شکل 1: یک سناریوی معمولی که در آن مسئله TCP Incast بروز می‌کند.

اطلاعات بیشتر در مور کنترل پروتکل انتقال در مراکز داده را می‌توانید در [47] بیابید.

C. تاخیر و بلاکینگ Head of Line

HoL پدیده ای است که زمانی رخ میدهد که دو جریان داده مستقل یا بیشتر یک ارتباط TCP را به اشتراک گذارند، برای مثال با ترافیک وب بر روی TCP؛ چرا که این پروتکل انتقال داده ای که دریافت میکند را به عنوان یک جریان مستمر واحد از بایت‌ها تفسیر می‌کند، و نیازمند تحویل ترتیبی است، با گم شدن یا تاخیر یک بسته برای هر جریانی،

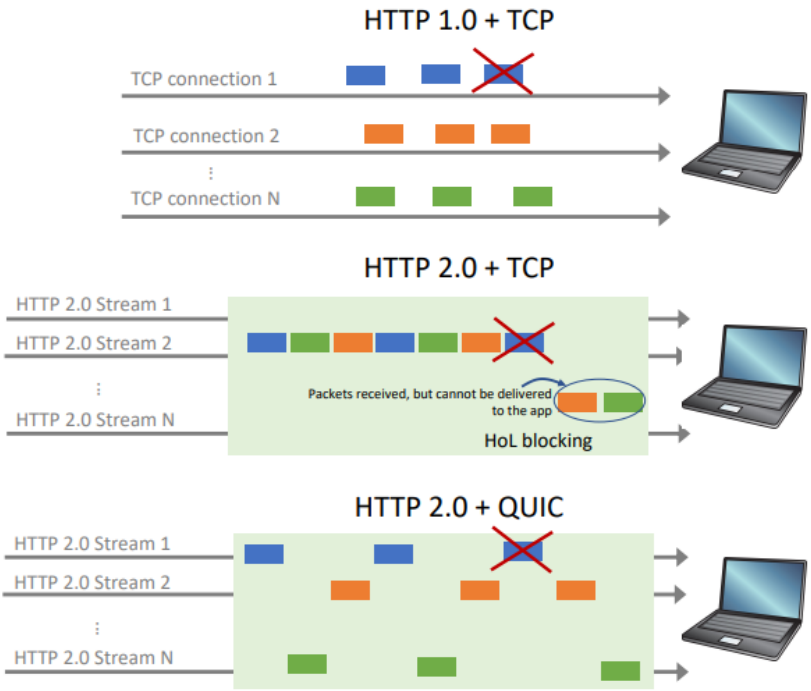
[14] slow start
[15] Explicit Congestion Notification
[16] Bandwidth-Delay Product

باعث تاخیر قابل توجه می‌شود. HoL در هر پروتکلی که نیازمند تحویل ترتیبی است یک مسله شناخته شده است، و راه حل‌های آشکار آن بازکردن یک ارتباط برای هر جریان داده‌ای است که از سربار قابل توجه در راه اندازی ارتباطی و بازیابی خطا رنج می‌برد. علاوه بر این، با این گزینه، کنترل ازدحام کمتر پایدار است، چرا که هر ارتباط به صورت مستقلی اجرا می‌شود.

ترافیک وب یک مثال از مسئله HoL است: معمولا، صفحات وب شامل اشیای متعددی هستند، مانند متن، تصویر، رسانه و اسکریپت‌های ثالث؛ زمانی که یک کلاینت یک صفحه را از سرور درخواست می‌کند، هریک از این اشیا با یک درخواست HTTP GET واحد دانلود می‌شوند، اما نیازی نیست که در همان لحظه نمایش داده شوند.HTTP/1.1 مالتی پلکسینگ را اجازه نمیدهد، لذا کلاینت مجبور است که برای هر شی یک ارتباط TCP باز کند، که مسائل گفته شده در بالا پیش می آید. نسخه 2 پروتکل، که در سال 2015 تحت عنوان RFC 7540 معرفی شد، قرار بود که این مسئله را با استفاده از یک ارتباط TCP واحد برای مدیریت همه درخواست ها، با کاهش زمان بارگذاری صفحه حل کند. به هر حال، این مزیت با HoL در شبکه‌هایی که بسته‌ها گم می‌شدند لغو شد: با توجه به اینکه همه بسته‌ها برای جریان‌های HTTP 2.0 متعدد در طول زمان بر یک ارتباط TCP مالتی پلکس می‌شوند، که نیازمند تحویل ترتیبی است، سپس گم شدن یک بسته باعث توقف همه جریان‌ها می‌شود، نه فقط جریان مربوط به بسته، که در شکل 2 نیز نشان داده شده است. برای مثال، سنجش‌هایی در شبکه‌های سلولی نشان میدهد که HTTP/2 مزیت عملکردی قابل توجهی نسبت به HTTP/1.1 ندارد [19].

در اصل، تحویل ترتیبی برای قابلیت اعتماد ضروری نیست، اما پروتکل‌های انتقال مانند TCP نیازمند هردوی آن‌ها هستند، اگر چه پروپوزالی برای بسط TCP برای اجازه دادن انتقال بدون ترتیب برای سرویس‌های چند رسانه ای برای اجتناب از مسئله HoL وجود دارد، این به صورت گسترده پذیرفته نشده است. شکل 2 نشان می‌دهد که پروتکل‌ها از جریان‌های داده متعدد مانند SCTP یا QUIC حمایت می‌کنند، که در بخش 3 تشریح شده است، و می‌توان بر مسئله HoL به دلیل اینکه هیچ خط واحدی برای مسدود سازی وجود ندارد غلبه کند: هر جریان، بافر خود را دارد و تحویل ترتیبی در هر جریان تضمین می‌شود [58].

سناریوهای چند مسیره مورد دیگری است که در آن TCP به شدت از مسئله HoL رنج می‌برد؛ که این مسئله را به صورت دقیق در بخش 5 تشریح می‌کند.



شکل 2: مسئله HoL با TCP و HTTP/2, که به لطف QUIC حمایت از جریان‌های مستقل متعدد به ازای هر ارتباط حل شده است.

D. عملکرد در کانال‌های بی سیم

عملکرد TCP در کانال‌های بیسیم از زمان معرفی سرویس‌های بی سیم تجاری در دهه 1990 بررسی شده است [59]. الگوهای کنترل ازدحام سنتی معمولا به گم شدن بسته ها، که فرض میکنند ناشی از ازدحام است، با کاهش نرخ ارسال واکنش نشان می‌دهند. در لینک‌های بی سیم، بسته‌ها ممکن است به دلیل کاهش کیفیت کانال گم شوند. بنابراین، انتزاع ارتباطات کلی ضعیف است، و عملکرد انتها به انتها تضعیف میشود. راه حل‌های متداول شامل ارائه مکانیزم‌های انتقال مجدد و حفاظت در لینک‌های بی سیم [60] (حتی با افزایش نوسانات تاخیر)، و پروکسی‌های ارتقای عملکرد درون شبکه است. چندین تکنیک در طول سال‌ها برای تکنولوژی‌های متفاوت پیشنهاد شده است مانند

LTE [62]، و WiFi [63]. خوانندگان را به برای بحث‌های گسترده در مورد عملکرد TCP در لینک‌های بی سیم به [64][63] ارجاع می‌دهیم.

اخیرا، پذیرش فرکانس‌های mmWave برای نسل بعد شبکه‌های سلولی علاقه ای نسبت به عملکرد TCP در لینک‌های بی سیم بوجود آورده است. به خصوص، در این رسانه بی سیم نوسانات کانال بسیار بیشتر از فرکانس‌های زیر 6GHz است، که با توجه به حساسیت نسبت به مسدود سازی موارد معمولی، و جهت گیری ارتباطات است. این محدودیت‌ها بر طرح لایه‌های پایینی پشته پروتکل تاثیری ندارد، اما بر عملکرد لایه انتقال، که در [67][66] بحث شده است تاثیر می‌گذارد. حلقه کنترل TCP براستی برای واکنش مناسب به پویایی کیفیت کانال خیلی آهسته است و نرخ داده ای را ارائه میدهد که بر لینک‌های mmWave تاثیر می‌گذارد. برای مثال، حتی اگر Bufferbloat در شبکه‌های سلول زیر 6GHz ارائه شوند، این جدید در mmWaves مخرب تر است، جایی که بافرهای بزرگ برای حفاظت از تغییرات زمانی و ناگهانی ظرفیت لینک نیاز است، با توجه به انتقال خط دید[۱۷] (LOS) به غیرخط دید[۱۸] (NLOS). علاوه بر این، این نشان میدهد که TCP استفاده زیربهینه ای از نرخ داده mmWave بسیار بالا دارد، چرا که زمان زیادی می‌برد که بعد از گم شدن بسته یا در آغاز ارتباط به ظرفیت کامل دست یابد. از آنجایی که این مسئله در منطق کنترل ازدحام ذاتی است، هر پروتکلی که کنترل ازدحام را اجرا می‌کند نیازمند کار کارآمدی در یک سناریوی بی سیم است.

3. پروتکل‌های انتقال اخیر

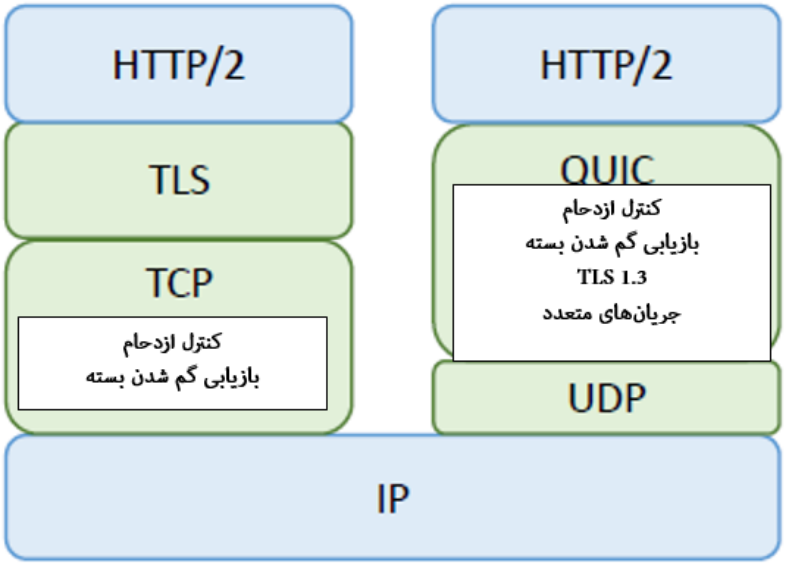
در این بخش، بر تشریح سه پروتکل انتقال به تازگی پیشنهاد شده تمرکز میکنیم: QUIC, SCTP and DCCP. به طور خاص، ویژگی‌های اصلی هر پروتکل را تشریح میکنیم، و با توجه به TCP نوآوری‌های آن را مطرح می‌کنیم. جدول 1 مشخصه‌های اصلی این پروتکل‌ها را، همراه با اسناد IEFT مرتبط و برخی از جزییات اجرایی آن‌ها خلاصه

[17] Line of Sight
[18] Non Line of Sight

میکند. بحث بر مکانیزم‌های کنترل ازدحام می‌تواند توسط این پروتکل‌ها استفاده شود و بسط‌های چند مسیره اخیر در زیر بخش بعدی ارائه شده است.

جدول 1: خلاصه ای از ویژگی‌های اصلی پروتکل‌های انتقال اخیر که در این مقاله بررسی شده اند (SCTP, QUIC and DCCP). ما همچنین یک بررسی بر پروتکل‌های TCP و UDP قبلی به عنوان مقایسه آورده ایم.

پروتکل	RFC یا پیش نویس	ویژگی‌ها	اجراها	بسط‌های چند مسیر
QUIC	[24], [48]	جریان بایتی، کنترل جریان و ازدحام، قابلیت اعتماد، چند جریانی، ادغام با TLS، ایجاد ارتباط RTT صفر	فضای کاربر، کتابخانه‌های متعدد در دسترس هستند	QUIC چند مسیری، تشریح شده در V-B
SCTP	[12]	جریان بایتی، کنترل جریان و ازدحام، قابلیت اعتماد، چند جریانی ، multihoming	کرنل سیستم عامل	CMT-SCTP تشریح شده در V-B
DCCP	[23], [53], [54]	مبتنی بر دیتاگرام، کنترل ازدحام	کرنل سیستم عامل	MP-DCCP تشریح شده در V-B
TCP	[1], [55]	جریان بایت، کنترل جریان و ازدحام، قابلیت اعتماد	کرنل سیستم عامل	MPTCP تشریح شده در V-A
UDP	[3]	مبتنی بر دیتاگرام، بهترین تلاش	کرنل سیستم عامل	خیر

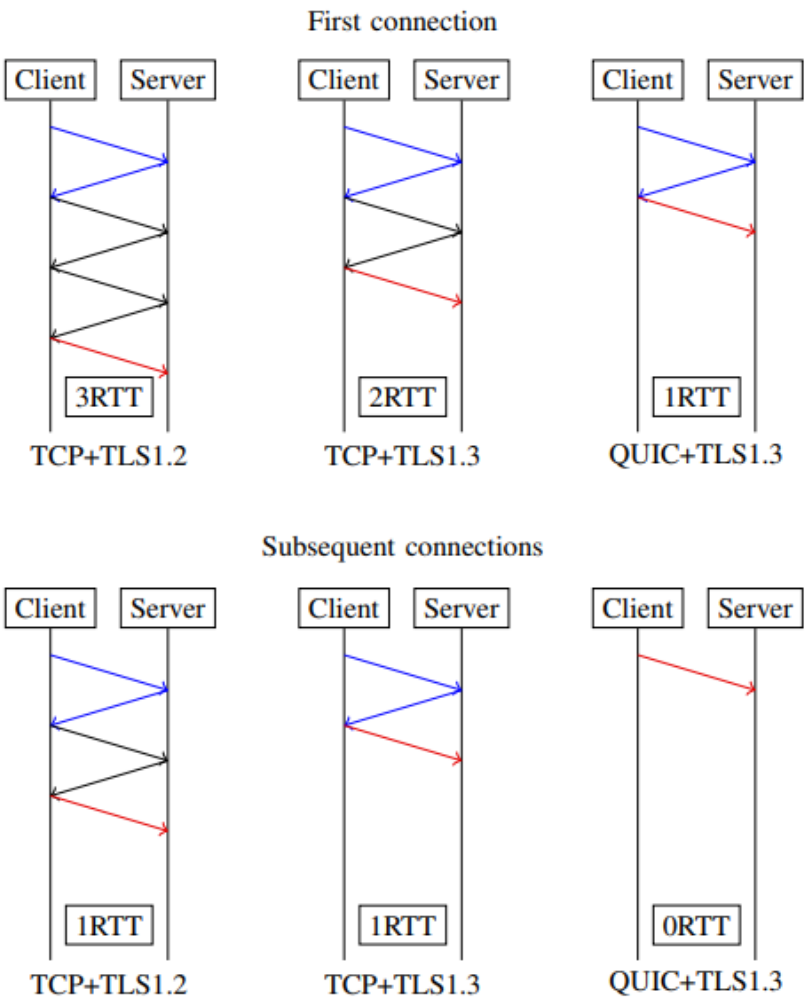


شکل 3: پشته پروتکلی با QUIC و TCP، که در [58] تشریح شده است.

A . یک ارتباط اینترنتی UDP سریع (QUIC)

QUIC یک مکانیزم انتقال لایه کاربرد یا اپلیکیشن است [69] که توسط گوگل در سال 2013 معرفی شده و در حال حاضر برای استانداردسازی توسط IEFT در نظر گرفته شده است. QUIC در کرنل اجرا نمی‌شود، اما به عنوان یک کتابخانه فضای کاربر یا در خود اپلیکیشن اجرا می‌شود، و بر UDP به عنوان پروتکل انتقال اصلی تکیه می‌کند. انگیزه‌های اصلی مربوط به معرفی QUIC در [70] بحث شده است. در حالی که گزینه‌های متعددی برای TCP در لایه انتقال پیشنهاد شده اند، پذیرش آن‌ها خیلی گسترده نیست، با توجه به موارد استخراج شده ناشی شده از جعبه‌های میانی در شبکه ای که بسته‌ها از پروتکل‌های مجهول را دور می اندازد، برای مثال پروتکل‌هایی که از TCP و یا UDP متفاوت هستند. QUIC بسته‌های خود را در دیتاگرام‌های UDP کپسوله میکند، بنابراین از بیشتر ناسازگاری‌ها با جعبه‌های میانی اجتناب میکند. دوم، همانطور که در بخش 1 گفتیم، TCP در کرنل اجرا می‌شود، و هر تغییر یا الگوریتم کنترل ازدحام جدید نیازمند یک به روز رسانی سیستم عامل است، که همیشه امکان پذیر نیست. سرانجام، از نظر طراحی پروتکل، QUIC حل مسئله TCP HoL را که در بخش II-C تشریح شد را هدف قرار میدهد، و تاخیر دست تکان دادن[۱۹] در آغاز ارتباط را کاهش می‌دهد.

[19] handshake



شکل 4: رویه های ایجاد اتصال. فلش های آبی نشان دهنده دست دادن TCP یا QUIC هستند، فلش های سیاه نشان دهنده دست دادن رمزنگاری هستند. فلش قرمز اولین انتقال داده است.

QUIC از (1) برخی از ویژگی‌های TCP، شامل مکانیزم تایید، کنترل ازدحام، و بازیابی بسته‌های گم شده، (2) مذاکرات کلیدی TLS 1.3 که نیازمند ارتباطات رمز شده است، و (3) ویژگی‌هایی HTTP/2، مانند چند جریانی تشکیل شده است [58]. شکل 3 تفاوت‌هایی را در پشته پروتکلی برای یک ترکیب HTTP/2 و TCP و HTTP/2 و QUIC نشان می‌دهد. ترکیب این عناصر به QUIC اجازه میدهد که چند عرصه کلیدی را بهینه سازی کند. برای مثال، یکپارچه سازی دست تکان دادن‌های مربوط به انتقال و رمزنگاری در همان پیام کاهش زمان را برای دست تکان دادن اولیه ممکن میسازد. علاوه بر این، رمزگذاری هدر بسته بررسی و ردیابی هدر بسته را برای جعبه‌های میانی مشکل می‌کند. علاوه بر این، این بازیابی بسته‌های گم شده TCP را با تمایز قائل شدن در ACKها بین بسته‌های

گم شده و خارج از ترتیب بهبود می‌بخشد. یک ارتباط واحد از چند جریان مستقل تشکیل شده است (که می‌تواند به جریان‌های HTTP/2 نگاشت یابد) لذا گم شدن یک بسته از یک جریان واحد دیگر را مسدود نمیکند، که مسئله HoL را به صورت موثری کاهش میدهد. سرانجام، شناسه‌های اضافی برای برچسب گذاری یک ارتباط معرفی شدند، که می‌توانند حتی در مورد تغییرات در آدرس پروتکل اینترنت (IP) حفظ شوند.

1) ایجاد ارتباط: ایجاد ارتباط یکی از نوآوری‌های اصلی معرفی شده در QUIC است، به خصوص با گزینه 0-RTT. TCP حداقل به یک RTT برای دست تکان دادن در زمان استفاده از TLS 1.3، یک یا دو RTT اضافی برای دست تکان دادن رمزنگاری در زمان استفاده از نسخه 1.2 قبلی تر احتیاج دارد.

QUIC حداقل از TLS 1.3 استفاده میکند، و در مورد ایجاد ارتباط برای اولین بار، همه پارامترهای راه اندازی انتقال QUIC در اولین بسته حمل می‌شوند، برای یک پیام Client Hello، یک RTT واحد کافی است. پارامترها در طول اولین ارتباط و مقادیر Diffie-Hellman، برای محاسبه کلید رمزگذاری، که در کلاینت ذخیره شده است، مذاکره می‌شوند. در مورد ارتباطات بعدی، این اطلاعات توسط کلاینت همراه با اولین بسته داده به سرور ارسال می‌شود، و سرور از این برای شناسایی کلاینت و رمزگشایی داده استفاده می‌کند، بنابراین دست تکان دادن 0-RTT نشان داده شده در شکل 4 رخ میدهد.

2) بسته‌های QUIC : استفاده از TLS 1.3 به بسته‌های QUIC اجازه میدهد که جعبه‌های میانی را بدون اینکه اطلاعات داخلی آن‌ها را دستکاری کنند بگردند. در حقیقت، تنها بایتی که رمزنگاری نمی‌شود سرآیند UDP و یک بخش سرآیند QUIC است، که در شکل 5 نشان داده شده است: این اطلاعات کلیدی در جریان‌های QUIC را از تجهیزات شبکه و اپراتورها مخفی میکند. بعد از ایجاد ارتباط، تنها مقادیری که در متن واضح در سرآیند QUIC ارسال می‌شوند IDهای ارتباطی مبدا و مقصد، نوع بسته، شماره ورژن و طول بسته است. شماره بسته با همان رمزگذاری که بر بار مفید اعمال شده است حفاظت نمی‌شود، اما در برابر مشاهدات گاه به گاه حفاظت در برابر محرمانگی دارد. شماره بسته، مانند TCP، برای جریان‌های ارسال و دریافت متفاوت است، با توجه به این محدودیت که نباید در همان ارتباط مجدد، حتی برای انتقال مجدد، استفاده شود [24].

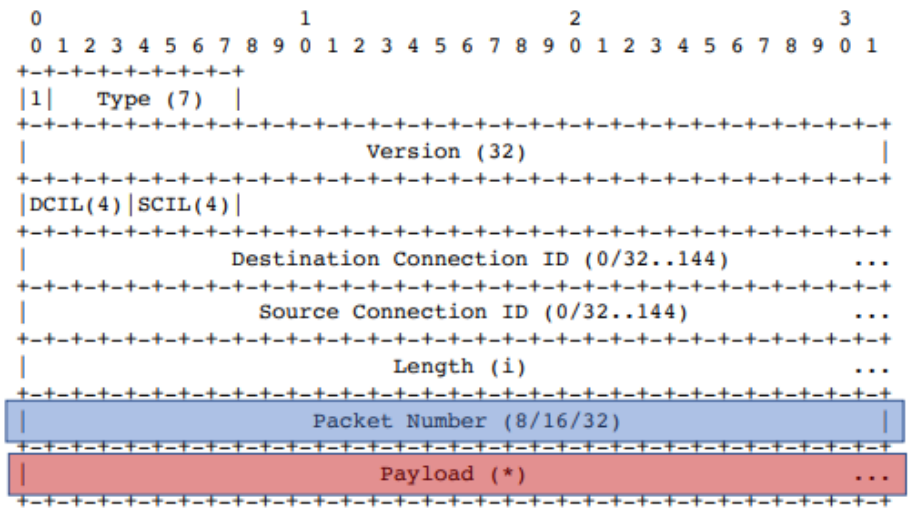
بار مفید بسته شامل یک فریم یا بیشتر است. هر فریک یک نوع دارد، مانند ack, ping or stream، و یک سری فیلدهایی دارد که وابسته به نوع هستند مانند stream ID و data length [24]. بنابراین یک بسته می‌تواند در چند جریان مالتی پلکس شود.

3) عملکرد QUIC : با توجه به اهداف طراحی امیدوار کننده QUIC، عملکرد ان خیرا تحت نظر قرار گرفته است، حتی اگر پروتکل هنوز کاملا توسط IEFT مشخص نشده باشد. تا به امروز، پروتکل QUIC در سرورهای گوگل اجرا شده است، و در سمت کلاینت، در Google Chrome و در برخی از اپلیکیشن‌های موبایل مانند YouTube or Google Search اجرا شده است. برطبق [70]، در سال 2017 این 7٪ ترافیک اینترنت را به خود تخصیص میدهد، و بیش از 30٪ Google's egress traffic را نیز به خود تخصیص میدهد. یک تحلیل دقیق تر از استقرار QUIC در [74] آورده شده است، که نشان میدهد که تعداد نقاط انتهایی شامل QUIC در حال افزایش است، به لطف استقرارهای Google و Akamai، در حالی که ترافیک به شدت توسط گوگل و خدمات ان پوشش داده می‌شود، با توجه به TCP/HTTPS در سپتامبر 2017 همراه با ستون فقرات MAWI به سهم 6.7٪ می‌رسد، که در شکل 6 نشان داده شده است و در یک ISP موبایل به 9.1٪ می رسد. نویسندگان به یک مشکل ممکن در پایداری بلند مدت با توجه به به روزرسانی‌های سازگار پروتکل اشاره کردند.

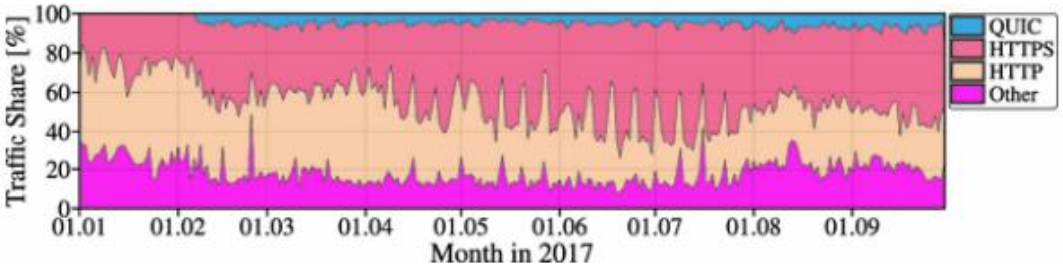
کار دیگر [70] بر برخی از معیارهای اندازه گرفته شده از ترافیک زنده گوگول بحث می‌کند. شکل 7 کاهش در تاخیر جستجوی تجربه شده توسط کلاینت‌ها با استفاده از QUIC نسبت به کلاینت‌هایی که از TCP/TLS استفاده می‌کنند را گزارش میکند. برخی از مقاله‌ها ادعا کرده اند که یک کاهش در نرخ بافرینگ مجدد صورت گرفته که 18٪ آن در ترافیک یوتیوب مشاهده شده است. سرانجام نویسندگان تاکید کردند که، با اجرا در زمان نوشتن، بار CPU به صورت میانگین 3.5 برابر بیشتر از QUIC نسبت با TCP/TLS است [70].

به تازگی، کار دیگری عملکرد 13 نسخه متفاوت از QUIC را، در سناریوهای دسکتاب و موبایل تست کرده است. با توجه به مطالعات قبلی [73][72]، این مقاله تحقیقات دقیق تری را با استفاده از نسخه‌های تست شده پیشرفته و اخیرتر پروتکل ارائه می‌دهد. یافته‌های اصلی (استفاده از CUBIC به عنوان کنترل ازدحام) تایید میکند که QUIC

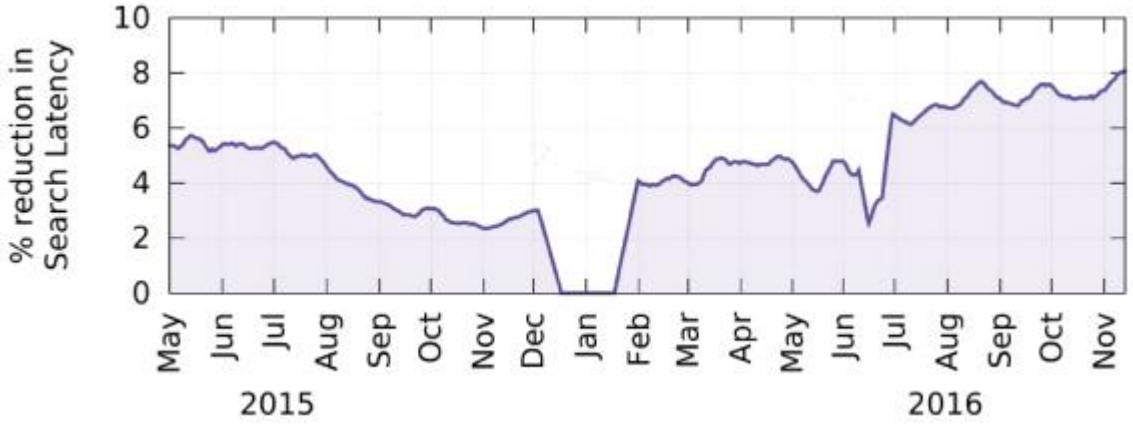
در محیط دسکتاب بهتر از TCP/TLS اجرا می‌شود، اما از یک مسئله مرتب سازی دوباره بسته رنج میبرد (که ممکن است در یک محیط موبایل بروز کند). با توجه به تغییرات در نرخ داده اصلی، QUIC به لطف مکانیزم ACK پیشرفته بهتر از TCP کار می‌کند. سرانجام QUIC با TCP یکی نیست، از پهنای باند گلوگاه بیشتری نسبت به سهم منصفانه استفاده میکند، و مشکلات عملکردی در سخت افزارهای قدیمی تر دارد، که با توجه به ان پردازش بسته و عملیات رمزنگاری در فضای کاربر به دلیل اینکه در کرنل هستند، بهینه نشده است. یک اجرای QUIC برای شبیه سازی شبکه منبع باز ns-3 اخیر منتشر شده است [75].



شکل 5: مثال ساختاری از یک بسته QUIC با سرآیند طولانی برای IETF QUIC (نسخه 14)، که از [24] پذیرفته شده است. فیلدهای قرمز رمزگذاری شده اند، در حالی که موارد آبی در برابر مشاهدات گاه و بی گاه حفاظت می‌شوند.



شکل 6: درصد ترافیک بر QUIC نسبت به پروتکل‌های دیگر در ستون فقرات MAWI در طول 2017.



شکل 7: درصد کاهش تاخیر در جستجو.

B. پروتکل انتقال کنترل جریان (SCTP)

SCTP، که در [12] تعریف شده است، یک پروتکل سیگنالینگ استاندارد است که توسط گروه کاری IETF Signaling Transport توسعه داده شده است که بعدها در یک پروتکل با هدف عمومی در لایه انتقال تکامل یافت. با این حال در گروه کاری عرصه انتقال IEFT (TSVWG)حفظ شده است. مانند TCP، در کرنل سیستم عامل اجرا می‌شود، و یک سرویس انتقال داده قابل اعتماد، نقطه به نقطه، ارتباط گرا را بر روی شبکه‌های IP فراهم می‌کند. SCTP به خصوص به صورت TCP دوستانه طراحی شده است، و کنترل ازدحام مبتنی بر پنجره خود، تشخیص خطا و مکانیزم‌های انتقال مجدد را به اشتراک می‌گذارد. با این وجود، SCTP نیز از چند ویژگی جدید تشکیل شده است، مانند:

• Multihoming – تا زمانی که یک ارتباط TCP بین دو سوکت (که توسط آدرس IP و شماره پورت شناخته می¬شود) در دو هاست برقرار شود، یک ارتباط SCTP می¬تواند در چند آدرس IP گسترده شود و احتمالا چندین نوع لینک را در شبکه شامل می¬شود. برای هر کلاینت، یک آدرس اولیه برای تبادل داده استفاده می¬شود، و سرورهای دیگر در مورد شکست لینک یا خطای شبکه به عنوان بک آپ استفاده می¬شوند، که دسترس پذیری بالاتر را بدون

وقفه در انتقال داده فراهم میکند. بسته‌های دوره ای به آدرس بک آپ ارسال می¬شوند تا وضعیت لینک را بررسی کنند، و انتقال مجدد بسته‌های گم شده می¬تواند با استفاده از آدرس بک آپ صورت گیرد.

• چند جریانی(multistreaming): مانند QUIC، SCTP قابلیت‌های چند جریانی را مطرح می¬کند. با استفاده از اصطلاحاتی از [12]، تعریف یک جریان SCTP به عنوان جریان داده منطقی در یک رابطه SCTP امکان پذیر است [79]. SCTP سپس جداسازی داده لایه اپلیکیشن در زیرجریان‌های متعدد را، برطبق اپلیکیشن‌های خاصی که بسته‌هایی را تولید میکنند اجازه میدهد. توالی و تحویل به صورت مستقلی در هر زیر جریان اجرا می¬شود، بنابراین بر مسئله HoL blocking که قبلا توضیح داده شد غلبه می¬کند. علاوه بر این، تحویل در هر جریان به شدت به صورت پیش فرض به ترتیب صورت میگیرد، مانند TCP، اما در صورت لزوم تحویل غیرترتیبی نیز امکان پذیر است. برای بهبود عملیات و امنیت ایجاد و قطع ارتباط کمی از TCP متفاوت است. یک مکانیزم کوکی در طول دست تکان دادن 4 باره اولیه مانع از حملات جعل هویت و ایجاد سیل می¬شود؛ علاوه بر این ارتباطات نیمه بسته در جایی که یک نقطه انتهایی مجاز به ارسال داده است هنوز مجاز نیست.

عملکرد SCTP در چند سناریو ارزیابی شده است: گزارش شده است که ترتیب جزیی و قابلیت اعتماد جزیی عملکرد را نسبت به پروتکل‌های انتقال سنتی مانند TCP افزایش می‌دهد. Jungmaier و همکاران [81] اثبات کرده اند که ترافیک SCTP همان تاثیر ترافیک TCP استاندارد را دارد، به همین ترتیب، دو پروتکل می‌توانند همزمان در یک شبکه حضور داشته باشند. علاوه بر این، الگوی multihoming می‌تواند توان عملیاتی بهتری را کسب کند و در زمان اعمال به شبکه‌های LAN بی سیم بازیابی حوادث شبکه سریعتر رخ می‌دهد [82].

سرانجام، برخی از مقالات عملکرد SCTP را با توجه به مسئله HoL blocking تست کردند. ارزیابی‌ها بر اساس پروتکل شروع جلسه[۲۰](SIP) در لایه کاربرد، و SCTP، TCP و UDP به عنوان انتقال است، که نشان میدهد که چگونه مکانیزم اجتناب از SCTP HoL blocking عملکرد بهتری در سناریوهای نرمال ترافیک همزمان در شبکه ندارد، و تنها کمی تاخیر انتها به انتها را با سطح بالای گم شدن بسته‌ها را کاهش می‌دهد [83]. علاوه بر این، زمانی

[20] Session Initiation Protocol

که مقدار حداکثر واحد انتقال[21](MTU) ناگهان کاهش می‌یابد، SCTP قادر به اجتناب از بخش بندی IP نیست و NATها ممکن است قادر نباشند که به درستی دیتاگرام‌های تقسیم شده را مسیریابی کنند.

C. پروتکل کنترل ازدحام دیتاگرام (DCCP)

DCCP، که در [23], [53], [54], معرفی شده است پروتکلی است که UDP را با توابع اتصال گرا یا ارتباط گرا مانند کنترل ازدحام تقویت میکند، و در عین حال ساختار پیام گرای آن را حفظ می‌کند. مانند UDP نرمال، این تحویل قابل اعتماد را تضمین نمی‌کند، و بسته‌ها می‌توانند بدون ترتیب تحویل داده شوند. بنابراین، یک گزینه مناسب برای انتقال در اپلیکیشن‌هایی است که به قابلیت اعتماد نیازی ندارند، و نیاز به اجرای مکانیزم کنترل ازدحام هم ندارند [84]. دلیل اصلی اینکه چرا این پروتکل معرفی شده است رشد سریع اپلیکیشن‌های تاخیر گرا با تکیه بر ارتباطات UDP است، مانند بازی‌های آنلاین یا تلفن‌های IP. این اپلیکیشن‌های بلادرنگ نمی‌توانند تاخیرهای ایجاد شده توسط مکانیزم‌های قابلیت اعتماد را تحمل کنند، و اما هنوز برای مطابقت و اجتناب از جریان‌های choking TCP در زمان اشتراک گذاری یک گلوگاه با آن‌ها نیازمند اجرای کنترل ازدحام هستند. DCCP یک فریم ورک متداول را ارائه میدهد که می‌تواند کنترل ازدحام را برای این اپلیکیشن‌ها اجرا کند، اما به مکانیزم‌های جدید اجازه میدهد که به یک روش سرراست اجرا شوند.

DCCP یک ارتباط دو طرفه را ایجاد می‌کند، که به لحاظ منطقی از دو ارتباط یک طرفه تشکیل شده است [23]. با توجه به اینکه این یک پروتکل انتقال غیرقابل اعتماد است، DCCP انتقال مجدد را اجرا نمی‌کند. با این وجود، نیاز است که برای تشخیص بسته‌های گم شده کنترل ازدحام را اجرا کند [84]. بنابراین، پروتکل ACKهای تجمعی ندارد، اما شماره توالی هر بسته اجازه تشخیص اینکه آیا دیتاگرامی گم شده است یا نه را میدهد. ACKها نیز تایید می‌شوند، با استفاده از همان مکانیزم، و این مسئله اجرای کنترل ازدحام را در این بسته‌های بازخورد اجازه میدهد. ویژگی ارتباطی (مانند مکانیزم‌های امنیتی، یا الگوریتم‌های کنترل ازدحام)، باید در زمان ایجاد ارتباط، با استفاده از

[21] Maximum Transmission Unit

فیلدهای کنترلی تعبیه شده در سرآیندهای DCCP بررسی شوند [23]. بنابراین، تنظیم کنترل ازدحام در هر ارتباط انتها به انتهای واحد امکان پذیر است. DCCP ارائه ECN را اجازه میدهد، اما از اعلان ازدحام شبکه بدون لزوما ثبت بسته‌های از دست رفته حمایت میکند.

D. چالش‌ها و جهات پژوهشی

پروتکل‌های ارائه شده در این بخش در تلاش برای حل تعدادی از مسائل و چالش‌های مربوط به TCP مطرح شده اند. با این وجود، هنوز مشکلات اصلی مربوط به پذیرش گسترده در اینترنت وجود دارد، که همانطور که بحث شد، با تغییر در تجهیزات شبکه برطرف می‌شود. QUIC یک جهت امیدوار کننده را به لطف یک پیاده سازی فضای کاربر، و پشتیبانی از ارائه دهندگان محتوای وب اصلی، مانند گوگل در پیش گرفته است.

دیگر روند مهم در QUIC و SCTP پشتیبانی از به روزرسانی‌ها در لایه IP و یا multi-homing است، لذا ارتباط انتها به انتها تا زمانی که نقطه انتهایی آدرس IP خود را تغییر نداده است نیازی نیست شکسته شود. این یک ویژگی سودمند در یک شبکه بی سیم به شدت سیار است، که در آن کاربر ممکن است به صورت مکرر در شبکه‌های مختلف گذر کند. علاوه بر این، این پروتکل‌ها طبیعتا پشتیبانی می‌شوند یا می‌تواند برای حمایت از مسیرهای متعدد بسط یابند، که در بخش 5 در مورد آن‌ها بحث شد.

علاوه بر این، QUIC، SCTP و DCCP به عنوان یک مشخصه اصلی طراحی با انعطاف پذیری همراه هستند، که یک مسئله جدید نسبت به TCP است. برای مثال، در QUIC و SCTP غیرفعال کردن انتقال‌های قابل اعتماد برای جریان‌های اصلی ممکن است، و QUIC و DCCP مکانیزم‌هایی برای ویژگی‌های و پارامترهای مذاکره شده در طول دست تکان دادن ارتباط دارد. این تنظیم مجدد باید با تکامل مناسب APIها با لایه کاربرد منطبق باشد، لذا توسعه دهندگان می‌توانند از ان بهره ببرند [107]. سیگنالینگ QoS صریح در این جهت یک مرحله فراتر می‌رود: به منظور پشتیبانی کامل از اپلیکیشن‌ها وسرویس‌های جدید، نقاط انتهایی نیاز دارند که قادر باشند که پیش نیازهای QoS خود را مشخص کنند، همانطور که الگوهای آگاه از QoS از ابزارهایی مانند SDN استفاده می‌کنند [109].

جدول 2: خلاصه ای از الگوهای کنترل ازدحام ارائه شده، دسته بندی بر اساس فلسفه طراحی

معایب	مزایا	مکانیزم کنترل ازدحام	الگوریتم	نوع
ناکارآمدی در LFN بسیار تهاجمی تعداد بالای انتقال مجدد RTT به شدت نوسان دار	همگرایی اثبات شده، عدالت عدالت بالاتر عدالت مستقل از RTT عدالت و بهره وری	AIMD تابع افزایش جستجوی باینری تابع Cubic CWND نطبیق مبتنی بر توالی	NewReno [88] BIC [89] CUBIC [90] Wave [91]	مبتنی بر گم شدن بسته
سرکوب توسط جریان مبتنی بر گم شدن بسته بار CPU سمت ارسال کننده بالا ناعادلانه نسبت به Vegas و BBR محدود نسبت به ترافیک اولویت پایین	انتقال مجدد کمتر، تاخیر کمتر تطبیق پذیری نسبت به کانال-های نوسان دار تهاجمی بودن مقیاس پذیر در مقابله با CUBIC بر جریان‌های دیگر تاثیر ندارد	تغییر RTT با سیگنال ازدحام AIMD مبتنی بر مشخصه تاخیر صف صریح و مدلسازی ترافیک تاخیر اضافی برای جریان‌ها با اولویت بالا	Vegas [92], [93] Verus [94] Nimbus [95] LEDBAT [96]	مبتنی بر تاخیر
بدون کنترل تاخیر نیاز به یک بافر به ازای هر جریان	در وایرلس و لینک‌های سست خوب است تاخیر کم، قابل سفارشی سازی	تخمین پهنای باند با کاهش CWND مدل ظرفیت HMM	Westwood [97] Sprout [98]	مبتنی بر ظرفیت
بدون کنترل تاخیر بدون کنترل تاخیر بدون کنترل تاخیر مسائل عدالت و حرکت	سریع در LFN، عادل نسبت به CUBIC سریع در LFN، عادل نسبت به CUBIC سریع در LFN، عادل نسبت به CUBIC توان عملیاتی بالا، تاخیر کم	مجموع پنجره های Reno و Vegas تاخیر استفاده شده برای تعیین تغییر CWND مدل صریح بافر ظرفیت و سنجش RTT	Compound [99] Illinois [100] Veno [101] BBR [21]	ترکیبی
مسائل عدالت با دیگر TCPها نیازمند دانش شبکه تست نشده bufferbloat تست نشده در محیط‌های به شدت پویا ارزیابی عملکرد محدود	رسیدن به ظرفیتی با تاخیر کم مسئله عدالت برطرف شده است. خوبی در شبکه‌های RTT بالا قابلیت‌های خود سازمان ده توان عملیاتی بالاتر نسبت به NewReno	سیاست مبتنی بر مونت کارلو پیشرفت در Remy آزمایشات آنلاین برای تعیین CWND تقویت یادگیری برای تعیین الگوریتم CC تقویت یادگیری برای انتخاب CWND	Remy [102] TAO [103] PCC [104] TCP-RL [105] QTCP [106]	مبتنی بر یادگیری

4. الگوریتم‌های کنترل ازدحام

کنترل ازدحام یک تابع ضروری در شبکه‌های مدرن است [88]، زمانی که حجم بالایی از ترافیک نیاز است که به صورت قابل اعتمادی ارائه شود می‌تواند حمایت شود اگر ارسال کننده نرخ خود را قبل از ارسال سیل آسای ارتباطات خود محدود کند و بر دیگر جریان‌ها تاثیر گذارد. این می‌تواند مستقیما توسط پروتکل‌های لایه انتقال مانند DCCP, SCTP, QUIC و TCP اجرا شود.

اپلیکیشن‌هایی که مستقیما بر UDP ارسال می‌شوند معمولا کنترل ازدحام خود را اجرا میکند، که توسط IEFT توصیه شده است.

منطق پشت کنترل ازدحام به شرح زیر است: از آنجایی که لینک‌ها با ظرفیت محدود نیاز دارند که توسط جریان‌های متعدد به اشتراک گذاشته شوند، آن‌ها می‌توانند با جریان‌های تجمعی تولید شده توسط منابع متعدد دچار سرریز شوند، که یک گلوگاه ارتباطی را ایجاد میکنند. اگر هیچ محدودیتی در نرخ ارسال نباشد؛ ارسال کننده‌ها انتقال مجدد بسته‌ها را در زمانی که یک تایید دریافت مثبت (ACK) را از مقصد در یک بازه به نام تایم اوت انتقال مجدد[22](RTO) دریافت نکرده اند، آغاز میکنند. این، به نوبه خود، بار شبکه را افزایش میدهد، و یک مارپیچ مخرب به نام فروپاشی ازدحام را هدف قرار میدهد [110]. کنترل ازدحام اجتناب از این شرایط را هدف قرار میدهد: ارسال کننده‌های به علائم ازدحام با کاهش فعال نرخ خود واکنش نشان میدهند. چند طرح کنترل ازدحام وجود دارد، که راه‌های متفاوتی برای تشخیص ازدحام و واکنش به ان دارند، اما همه آنها یک هسته مشترک دارند.

مسائلی که در بخش 2 ذکر کردیم طراحی مکانیزم‌های کنترل ازدحام خوب را سخت می‌سازد: ویژگی‌های خاص ارتباطی و هریک از لینک‌ها می‌توانند بر عملکرد، به روش‌های غیرقابل پیش بینی تاثیر گذارند. مدل‌های ریاضی پروتکل‌ها و فرضیات ان‌ها به ندرت واقع بینانه هستند، و اغلب به لحاظ نظری الگوریتم‌های بهینه آن طور که در شبکه‌های واقعی انتظار می‌رود کار نمی‌کنند. علاوه بر این، در بیشتر موارد (برای TCP)، الگوریتم کنترل ازدحام

[22] Retransmission Timeout

برای یک پروتکل انتقال خاص در یک پایگاه کد کرنل از آن پروتکل اجرا می‌شود، و در نتیجه، برای هر ارتباط انتها به انتها یکی است. بنابراین، سفارشی سازی پاسخ‌های الگوریتم ازدحام به مشخصه‌های هر ارتباط ممکن نیست.

علاوه بر این، تشخیص زمانی که ازدحام صورت میگیرد و اینکه چگونه اندازه پنجره ازدحام باید حفظ شود مسائل پیش پا افتاده ای نیستد. راه حل‌های اولیه گم شدن بسته‌ها را به عنوان نشانه ازدحام می‌دیدند و از یک استراتژی AIMD استفاده کردند، که منصف بودن و ثبات را تضمین می‌کند [112]. الگوی AIMID ساده است، چرا که نیازمند دو دستور برای به روزرسانی پنجره ازدحام است (که اغلب به صورت مختصر CWND گفته می‌شود):

$$\mathrm{CWND} \leftarrow \mathrm{CWND} + \frac{\alpha}{\mathrm{CWND}} \qquad (1)$$

در ACKهای دریافت شده و

$$\mathrm{CWND} \leftarrow \frac{\mathrm{CWND}}{\beta} \qquad (2)$$

در بسته‌های گم شده، که در این روابط α و β پارامترهای تنظیم مکانیزم هستند. پروتکل‌های جدی تر اغلب شامل اطلاعات تاخیر به عنوان یک سیگنال ازدحام بالقوه هستند، و پذیرش پنجره ازدحام متناسب با نیازهای عملیاتی است. ACK انتخابی (SACK) دیگر گزینه ای است که اطلاعات در دسترس برای مکانیزم کنترل ازدحام را بهبود می‌بخشد [55]. حال فلسفه‌های طراحی کنترل ازدحام اصلی و مزایا و نقص‌های آن‌ها را لیست می‌کنیم، و الگوریتم‌های قدیمی تر را به منظور ارائه اطلاعات پیش زمینه لازم برای تناسب بهتر با استقرارهای جدید بررسی می‌کنیم؛ مکانیزم‌های اصلی ارائه شده در این بخش در جدول 2 خلاصه شده اند. همچنین تاکید کردیم که همه مکانیزم‌های کنترل ازدحام TCP بحث شده در زیر می‌توانند برای ارسال کننده‌های QUIC و SCTP بدون هیچ اصلاحی اجرا شوند. DCCP نیازمند تلاش بیشتری است، چرا که شامل مکانیزم انتقال مجدد نیست، اما اصول پایه و فلسفه‌های طراحی یکی هستند. DCCP فریم ورکی را برای مکانیزم‌های کنترل ازدحام ارائه میدهد که می‌توانند توسط اپلیکیشن با استفاده از IDهای کنترل ازدحام (CCIDs) انتخاب شوند. مکانیزم‌های کنترل ازدحام از هسته پروتکل‌های فعال کننده ماژولاریته و بسته‌های ممکن جدا هستند. هریک از دو ارتباط می‌توانند یک CCID متفاوت را انتخاب کنند چرا که

آن‌ها به لحاظ منطقی جدا هستند. در زیر، به الگوهای کنترل ازدحام DCCP استاندارد شده توسط CCID آن اشاره می‌کنیم. دیگر طبقه بندی مکانیزم‌های کنترل ازدحام برای TCP در [113] ارائه شده است، اما فاقد توسعه‌های اخیری است که در این مقاله بررسی کردیم.

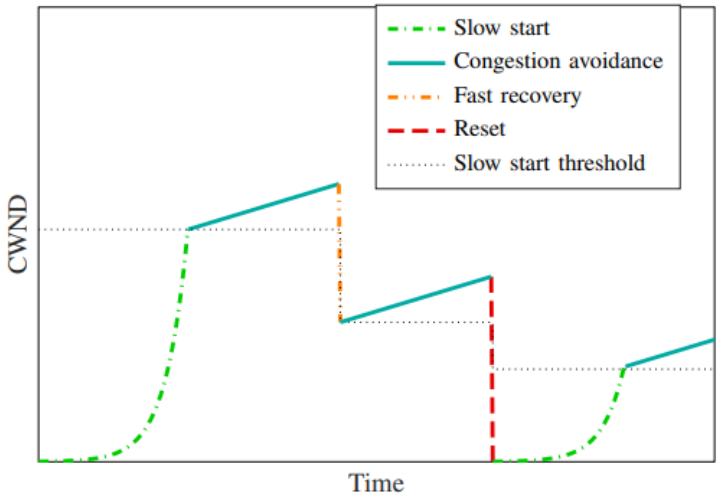
A. مکانیزم‌های مبتنی بر گم شدن بسته

الگوریتم‌های کنترل ازدحام TCP کلاسیک مانند Tahoe [1]، Reno [114] و New Reno [88] از گم شدن برای تشخیص ازدحام استفاده می‌کنند. اگر یک بسته مشخص شود که گم شده است، چه زمانی که تایمر RTO منقضی شده است یا بعد از سه بار ACK تکراری پی در پی، مکانیزم AIMD به صورت همواری پنجره ازدحام را کاهش می‌دهد. در حالی که Tahoe بسیار محافظه کار است، از یک CWND یک بسته ای شروع میکند و به فاز slow start وارد می‌شود، Reno و New Reno تنها CWND را با نیمی از موارد سه ACK تکراری کاهش می‌دهند، که در شکل 8 نشان داده شده است. کنترل ازدحام QUIC پیش فرض نیز بر اساس New Reno است، حتی اگر دیگر الگوریتم‌های کنترل ازدحام استفاده شوند، مانند TCP [48]. تعدادی از اصلاحات که برای New Reno پیشنهاد شده است مستقیما در پیش نویس‌های پروتکل IETF QUIC آورده می‌شود. برای مثال، شماره بسته به صورت یکنواختی افزایش می‌یابد، لذا ارسال کننده QUIC می‌داند که آیا یک ACK دریافت شده برای بسته اصلی است یا برای بسته ای است که مجدد انتقال داده شده است. علاوه بر این، در حالی که با ویژگی TCP SACK تنها سه دامنه ACK مشخص می‌شود [115]، QUIC تعداد دامنه‌های ACK را در یک بسته بازخورد محدود نمی‌کند. پیش نویس [48] نیز استفاده از pacing، را برای اجتناب از انتقال‌های انبوه بسته ها، و مکانیزم کاوش بسته‌های گم شده برای تشخیص موثر بسته‌های گم شده در پایان انتقال داده را توصیه می‌کند.

BDP یک محصول ظرفیت گلوگاه و حداقل RTT است؛ شبکه‌ها با یک BDP بر Long Fat Networks 100kb (LEN) در نظر گرفته می‌شود. مکانیزم‌های مبتنی بر گم شدن بسته کلاسیک به شدت در LENها ناکارآمد هستند،

چرا که فاز بازیابی برای یک ارتباط با یک حداقل 100 RTT میکروثانیه و یک ظرفیت 100 Mb/s کمتر از یک دقیقه است، و حدود ده دقیقه است اگر ظرفیت به 1 Gb/s افزایش یابد.

کنترل افزایش باینری[۲۳] (BIC) [89] و CUBIC [90] دو الگوریتم کنترل ازدحام است که برای حل مسائل مکانیزم‌های اولیه با LENها توسعه یافته است.



شکل 8: ارزیابی TCP New Reno's CWND در طول زمان

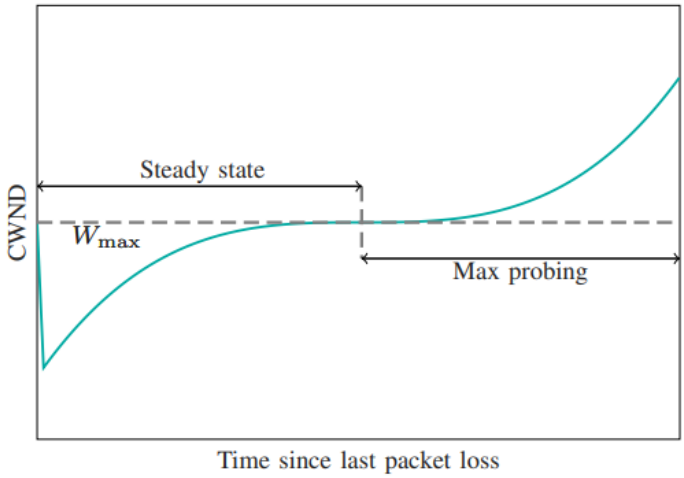
آن‌ها AIMD خالص را برای یک تابع پیچیده تر (یک جستجوی باینری در مورد BIC، و تابع CUBIC زمانی از زمان گم شدن آخرین بسته برای CUBIC، مانند شکل 2) به منظور کسب عدالت بین جریان‌ها با RTTهای متفاوت رها می‌کنند. به هر حال، BIC تهاجمی و نسبت به جریان‌های قبلی ناعادلانه تلقی می‌شود، در حالی که CUBIC به صورت گسترده توسعه یافته است و در حال حاضر یک IETF Internet-Draft است چرا که می‌تواند عملکرد بالایی را بدون انصاف نسبت به جریان‌ها با استفاده از مکانیزم‌های کنترل ازدحام قبلی تر کسب کند. اجرای QUIC در پایگاه کد Chromium کنترل ازدحام CUBIC را برجسته می‌سازد [90] ,[70]. تفاوت اصلی بین CUBIC اعمال شده به TCP و QUIC این است که $\beta_{\text{TCP}} = 0.3$ برقرار است (فاکتور کاهش برای پنجره ازدحام بعد از رویداد گم شدن بسته)، در حالی که $\beta_{\text{QUIC}} = \beta_{\text{TCP}}/2$ نیز برقرار است [72].

[23] Binary Increase Control

DCCP سه مکانیزم مبتنی بر گم شدن بسته را اجرا می‌کنند: CCID 2 یک مکانیزم شبه New Reno است که از همان اصل AIMD استفاده می‌کند، اما به دلیل ماهیت دیتاگرام بودن سوکت UDP اصلی از بسته‌ها به عنوان واحد داده به جای بایت استفاده می‌کند. CCID 2 نیز از کنترل ازدحام برای تایید استفاده میکند، و هیچ مکانیزم انتقال مجددی ندارد چرا که برای اپلیکیشن‌هایی است که نیازمند تحویل قابل اعتماد نیستند، و ممکن است حتی با ویژگی تحویل قابل اعتماد آسیب ببینند. CCID 3، یا کنترل نرخ TCP دوستانه (TFRC)، از معادله توان عملیاتی TCP صریحا برای محاسبه نرخ ارسال به روشی منصفانه برای مقایسه جریان‌های TCP و پایداری نسبی استفاده کرده اند. این برای اپلیکیشن‌هایی توصیه می‌شود که نیازمند توان عملیاتی پایدارتر هستند و در عین حال دوستدار TCP هستند. CCID 4 یک تغییر از TFRC طراحی شده برای اپلیکیشن‌هایی است که بسته‌های کوچک را ارسال می‌کنند، و پهنای باندی معادل با جریان‌های TCP ارسال کننده بسته‌هایی با سایز کامل بدست می آورند. دیگر بسط TFRC، که در زمانی که جریان‌های DCCP یک گلوگاه را به اشتراک میگذارند بهتر رفتار میکند، در [123] پیشنهاد شده است.

این روزها، بیشتر جامعه پژوهشی از تخمین ازدحام مبتنی بر گم شدن بسته فاصله گرفته اند. چندین مطالعه شبیه سازی نشان داده اند که هیچ یک از این مکانیزم‌ها در شبکه‌های بی سیم، در شبکه‌های اد هاک موبایل، و در شبکه‌های پهن باند، با تاخیر بالا خوب اجرا نمی‌شوند. TCP، QUIC، و SCTP همه از مسائل یکسانی در رنج هستند، چرا که مسئله از فلسفه کنترل ازدحام مبتنی بر گم شدن بسته نشات می‌گیرد. به هر حال، این می‌تواند یک اصل طراحی سودمند برای برخی از سناریوهای شبکه باشد، به خصوص زمانی که RTTهای بسیار بالا دخیل هستند. یک مثال TCP Noordwijk [126] است، یک نسخه میان لایه‌ای TCPکه از ویژگی‌های لینک‌های ماهواره ای بهره برداری میکند. ویژگی اصلی TCP Noordwijk این است که داده را در شلوغی و ازدحام با حداکثر ظرفیت گلوگاه انتقال میدهد، و یک بافر بزرگ در گلوگاه دارد و جریان‌های رقیب را در نظر نمی‌گیرد. عدالت توسط برنامه ریزی انتقال مناسب تضمین می‌شود؛ یک پیشنهاد اخیر، به نام TCP Wave [91]، که پروتکل را با حذف جنبه‌های میان لایه‌ای و پذیرش الگوریتم برای هر نوع لینک بسط می‌دهد. پروتکل سریعا عدالت را با هر نوع دیگر جریان مبتنی بر گم شدن

بسته به دست میآورد، و سریعا به محیطی که جریان‌های رقیب متوقف شوند به استفاده کامل از پهنای باند باز میگردد. به هر حال، اگر پروتکل بسته‌هایی را پشت سر هم ارسال کند، بسته‌ها در ارسال کننده تا زمان توالی بعدی صف می‌شوند، که باعث یک زمان صف طولانی میشود. مکانیزم‌های مبتنی بر توالی به شدت برای جیتر و نوسانات ظرفیت ارزشمند است چرا که آن‌ها نیاز است که به یک نرخ ارسال متعهد باشند و فرصت کمتری برای تصمیم اشتباه، نسبت به الگوهایی که تعهدات کمتری دارند دارا می‌باشند: اگر لینک گلوگاه با زمان تغییر کند، واکنش آن‌ها زمان میبرد و می‌تواند خطاهای بزرگتری ایجاد کند. بسیاری از اپلیکیشن‌های متداول نسبت به تاخیر و جیتر حساس هستند، و سودمندی Wave را در شبکه‌های استاندارد کاهش میدهند.



شکل 9: رشد CWND الگوریتم CUBIC

B. مکانیزم‌های مبتنی بر تاخیر

مکانیزم‌های مبتنی بر گم شدن بسته، مانند CUBIC می‌تواند به صورت کارآمدی از بیشتر ارتباطات استفاده میکند؛ به هر حال، تنها سیگنال ازدحامی که درک میکنند گم شدن بسته است. اگر بافر در لینک گلوگاه بزرگ باشد، بسته‌ها تا زمان پر شدن بافر جمع میشوند، در حالی که ارسال کننده نرخ ارسال را افزایش می‌دهند و تاخیر صف نیز افزایش می یابد. حتی در شرایط وخیم تاخیر تا ده ثانیه هم میرسد، و پر شدن بافر کاملا می‌تواند باعث ناپایداری در توان عملیاتی مشاهده شده توسط جریان‌های دیگر شوند.

یک راه حل ممکن برای این کنترل ازدحام مبتنی بر تاخیر است: گم شدن بسته‌ها ممکن است نشانه ای از ازدحام باشد، اما ازدحام ممکن است خیلی قبل تر از پرشدن بافر رخ دهد، و تشخیص زودتر آن می‌تواند واکنش کنترل ازدحام را افزایش دهد، که می‌تواند به محض افزایش RTT از کاهش شدید در توان و تاخیر زیاد خودداری کند.

TCP Vegas [92] اولین اجرا برای استفاده از تاخیر به عنوان سیگنال ازدحام است: بعد از یک فاز slow start ملایم تر [127]، این پنجره ازدحام را برای توان عملیاتی مورد انتظار کانال تنظیم میکند. کانال فرض می‌شود که زیاد استفاده نمی‌شود اگر RTT به حداقل مقدار اندازه گرفته شده نزدیک باشد؛ به محضی که RTT به فراتر از یک آستانه مشخص رشد کند، پروتکل متوقف می‌شود. Vegas منصفانه است و از ظرفیت بهتر از Reno استفاده می‌کند در حالی که یک تاخیر کم را حفظ میکند. به هر حال، اولین نسخه Vegas مسائل شدیدی با مکانیزم Delayed ACK دارد، که بر تخمین RTT آن، با لینک‌هایی با تاخیر بالا تاثیر میگذارد. این مسائل استقرار بهبودهای متدول را ارتقا میدهد، مانند Vegas-A، که مسیریابی مجدد و عدالت در اشتراک گذاری پهنای باند را بررسی میکند، Vegas-V [129]، تهاجمی بودن الگوریتم اصلی را افزایش میدهد، در حالی که همچنان برای جریان‌های دیگر منصف باقی می ماند، و Adaptive Vegas [130]، که تطبیقی در آن به توانایی الگوریتم برای تغییر پارامترهایش بر اساس آماره‌های توان عملیاتی و تغییرات RTT اشاره دارد. برخی از نسخه‌های خاص برای شبکه‌های اد هاک بی سیم و موبایل توسعه یافته‌اند؛ سه اصلاح خاص به بخشی از الگوریتم Vegas جدید تبدیل شده اند. New Vegas [93] از pacing بسته‌ها در فاز شروع آهسته برای اجتناب از مسائل ناشی شده از ,burstiness استفاده میکند، و بسته‌ها را دو تایی برای اجتناب از مسئله Delayed ACK ارسال میکند، در حالی که بیشتر ایده‌های اصلی ’Vegas را حفظ میکند. یک مکانیزم کنترل ازدحام مبتنی بر تاخیر برای DCCP در [134] پیشنهاد شده است، که نرخ ارسال را با نرخی بین حداقل و RTT حاضر کاهش می‌دهد. یک مکانیزم شبه Vegas از تاخیر یک طرفه برای سنجش ازدحام استفاده میکند، که نرخ ارسال را برای حفظ QoS تطبیق میدهد. از انجایی که این الگوها استاندارد نشده اند، آن‌ها به یک CCID خاص نسبت داده می‌شوند.

در طی چند سال گذشته، استقرار پروتکل‌های مبتنی بر تاخیر به لطف افزایش اپلیکیشن‌های تعاملی با الزاماتی بر تاخیر با مقاومت جدیدی مواجه شده اند. Verus [94] یک پروتکل کنترل ازدحام انتها به انتهای مبتنی بر تاخیر است که از سنجش تاخیرها برای تطبیق سریع با اندازه پنجره ازدحام برای ارتقای عملکرد انتقال در شبکه‌های سلولی استفاده میکند. این الگوی AIMD مشابهی مانند TCP سنتی را حفظ میکند، اما بخش افزایش افزودنی را تغییر میدهد: با سنجش پیوسته کانال، اندازه پنجره در هر مرحله برطبق مشاهدات تغییرات تاخیر بسته افزایش می یابد یا کاهش می‌یابد، که نویسندگان آن را " یادگیری مشخصه تاخیر" میگویند. اساسا، پروتکل یک تابع تجربی را با برقراری ارتباط CWND و تاخیر و مشاهده زمانی که نرخ از ظرفیت کانال فراتر میرود، ایجاد میکند. مراحل کاهش چند برابر و شروع کند هنوز اصلاح نشده است. این استراتژی میتواند موثرتر از استفاده از یک مدل صریح ارتباطی باشد، و به صورت خاصی با استفاده از یک پنجره لغزان بر یک دوره برابر با RTT تخمین زده شده حاصل میشود.

تست جهان حقیقی توسط نویسندگان [94] بر شبکه‌های 3G و LTE با در نظر گرفتن جریان‌های رقابتی متعدد بین برخی از دستگاه‌ها و سرورها در یک پهنای باند بالا و شبکه تاخیر کم اجرا میشود، که نتایج آن نشان میدهد که Verus می‌تواند یک توان عملیات را مشابه با TCP CUBIC بدست آورد و در عین حال تاخیر را به ترتیب بزرگی کاهش دهد. نویسندگان همچنان اشاره کردند که چگونه Verus از دیگر پروتکل‌ها بهتر اجرا میشود و سریعتر با تغییرات شبکه و جریان‌های جدید، با توان عملیاتی مستقل از RTT وفق می یابد، و عادلانه بودن پروتکل را تایید میکند. در حالی که نتایج مثبت توسط اقدامات مستقل تایید میشوند، این تست‌ها بر بارهای محاسباتی انبوه که تخمین مشخصه تاخیر Verus بر ارسال کننده وارد میکند تاکید میکند، استفاده ان از سناریوهای لینک بالایی و توان عملیاتی بالا ممکن است به دلیل پیچیدگی پروتکل غیر ممکن باشد.

در کل، مکانیزم‌های مبتنی بر تاخیر با یک توان عملیاتی مشابه در شرایط حقیقی پایدارتر هستند، بسته‌های کمتری را مجدد انتقال میدهند و تاخیر کمتری نسبت به مکانیزم‌های مبتنی بر گم شدن بسته‌ها دارند. به هر حال، به دلیل تهاجمی بودن بالای پروتکل‌های موجود، هنوز پذیرفته نشده اند. اگر یک جریان مبتنی بر تاخیر و گم شدن بسته یک گلوگاه را به اشتراک گذارند، و بافر در گره گلوگاه به اندازه کافی بزرگ باشد، مکانیزم مبتنی بر تاخیر ازدحام را خیلی

قبل تر از رقبای مبتنی بر گم شدن بسته حس می‌کند، که نرخ ارسال آن را بالا نگه خواهد داشت [136]. در نتیجه، توان عملیاتی جریان مبتنی بر تاخیر اساسا به صفر می‌رسد، در حالی که جریان مبتنی بر گم شدن بسته همه ظرفیت موجود را به خود اختصاص میدهد. از آنجایی که اکثر سرورها در اینترنت یک مکانیزم کنترل ازدحام مبتنی بر گم شدن بسته را اجرا میکنند و ترافیک مبتنی بر گم شدن بسته و تاخیر را جدا می‌کنند بسیار هزینه بر هستند و نیازمند تغییرات گسترده هستند، پروتکل‌های مبتنی بر تاخیر هیچ گاه در مقیاس‌های گسترده استقرار نیافته اند. به هر حال، استقرار پروتکل‌های مبتنی بر تاخیر با عملکرد بهتر، که ممکن است از تهاجمی بودن جریان‌های مبتنی بر گم شدن بسته در رنج باشند، اخیرا به لطف تکنیک‌هایی مانند قسمت بندی شبکه[۲۴] و مجازی سازی تابع شبکه[۲۵](NFV) امکان پذیر شده اند، اگر چه هنوز شامل تلاش قابل توجهی بر بخش اپراتور شبکه است.

دو پروتکل مبتنی بر تاخیر می‌توانند به یک حالت تهاجمی تر برای اجتناب از باختن از جریان‌های مبتنی بر گم شدن بسته که اخیرا پیشنهاد شده اند، سوئیچ کنند. مانند Verus، Copa [138] یک پروتکل است که از مشخصه تاخیر برای تعیین نرخ ارسال استفاده می‌کند. این از زنجیره مارکوف برای مدل کردن صریح صف گلوگاه استفاده می‌کند، و به صورت پویا حالت تهاجمی خود را برای مقایسه عادلانه با پروتکل‌های مبتنی بر گم شدن تنظیم می‌کند. Nimbus [95] ارتقای دیگری است که کشش ترافیک متقابل را مدل میکند، و تشخیص میدهد که چگونه جریان‌ها به تغییراتی در پهنای باند درک شده واکنش نشان میدهند و با سناریوی نتیجه تطبیق می‌یابند. مدلسازی ترافیک متقابل با مشاهده تبدیل فوریه ظرفیت و تشخیص رفتار دوره ای اجرا می‌شود؛ این رویکرد با CUBIC یا ترافیک متقابل Copa خوب کار میکند، اما می‌تواند برای ترافیک متقابل BBR با شکست مواجه شود و در حضور جریان‌های Vegan بسیار تهاجمی است، که به عنوان جریان‌های کششی حس می‌شوند و به برنده شدن ادامه میدهد.

دیگر استقرار جالب توجه توسط TCP Low Extra Delay Background Transport (LEDBAT) نشان داده شده است، یک الگوریتم مبتنی بر تاخیر توسعه یافته برای ترافیک BitTorrent است: از آنجایی که ترافیک پیش زمینه باید الویت کمتری نسبت به ترافیک کاربر داشته باشد، تهاجمی بودن پایین مکانیزم‌های مبتنی بر تاخیر

[24] network slicing
[25] Network Function Virtualization

به یک نقطه قوت تبدیل میشود. LEDBAT ظرفیت باقی مانده استفاده نشده توسط جریان‌های پیش زمینه را با سنجش تاخیر اضافی افزوده شده بعد از ارسال یک بسته تخمین می‌زند، و تنها در صورتی که استنتاج کند که اقدام آن هیچ اثری بر جریان‌های مهم تر ندارد، انتقال را انجام میدهد. این در حال حاضر در ماشین Microsoft Windows به عنوان راه حلی برای اپلیکیشن‌های انتقال انبوه مستقر شده است. TIMELY [139] دیگر پروتکل مبتنی بر تاخیر است که پنجره ازدحام را با استفاده از گرادیان RTT تنظیم میکند، که به صورت خاصی برای مراکز داده طراحی شده است. سرانجام، TCP LoLa [22] مکانیزم CUBIC را برای یک متد تشخیص مبتنی بر تاخیر می‌پذیرد، زمانی که تاخیر صف شروع به افزایش می‌کند، LoLa به مکانیزم شبه Vegas، برای حفظ ظرفیت و عدالت نسبت به جریان‌های دیگر، سوئیچ می‌کند.

C. مکانیزم مبتنی بر ظرفیت

TCP Westwood [97] یک مکانیزم کنترل ازدحام است که از سال 2001 معرفی شده است و صراحتا پهنای باند تخمین زده شده را به منظور تغییر تطبیقی پنجره ازدحام بعد از یک گم شدن بسته ردیابی می‌کند، و در عین حال الگوی افزایش افزودنی TCP Reno را حفظ می‌کند. Westwood برای لینک‌های بی سیم و سست طراحی شده است، لینک‌هایی که ظرفیت تغییر سریع دارند، و گم شدن بسته در آن‌ها ناشی از لایه فیزیکی است. هردو این فاکتورها می‌توانند بر عملکرد لایه انتقال تاثیر گذارند، چرا که الگوهای مبتنی بر گم شدن بسته سنتی به گم شدن بسته‌ها با نصف کردن نرخ ارسال حتی در زمانی که نیازی به آن نیست واکنش نشان می‌دهد، و ظرفیت سریعا در حال تغییر میتواند به عنوان نشانه ای از ازدحام تفسیر شود. Westwood پنجره ازدحام را برای سنجش ظرفیت ارتباطی بعد از گم شدن بسته‌ها نمی‌پذیرد و عملکرد بالاتری را نسبت به الگوهای مبتنی بر گم شدن بسته‌ها در این شرایط، مانند شبکه‌های سیم دار، حاصل می‌کند.

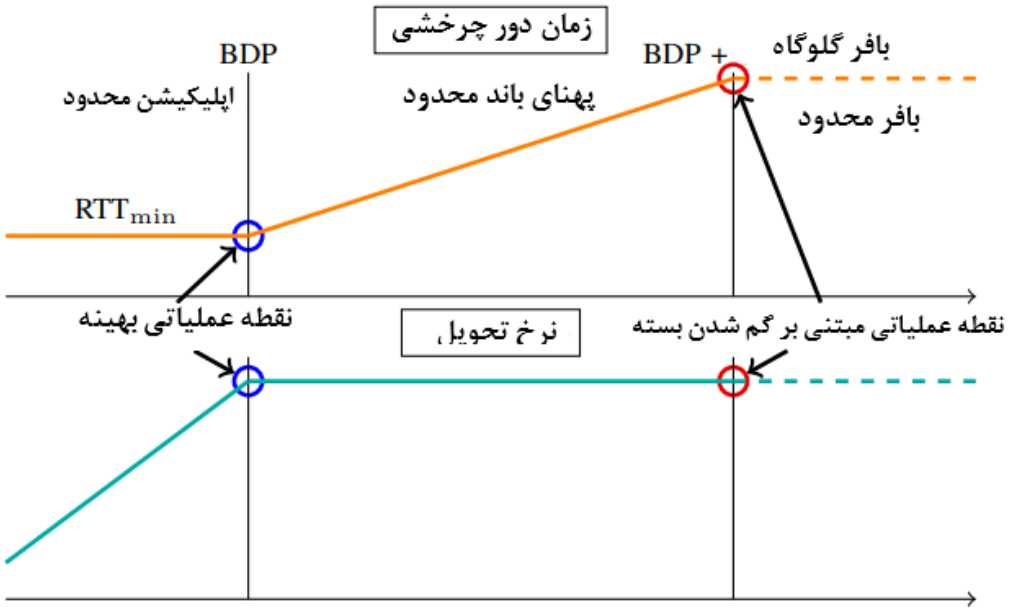
به هر حال، ایده کنترل ازدحام مبتنی بر ظرفیت اخیرا توسط Sprout [98] تجدید نظر شد، یک پروتکل انتها به انتها که برای شبکه‌های بی سیم سلولی طراحی شده است که یک سازش بین کسب بالاترین توان عملیاتی ممکن و

جلوگیری از تاخیرهای طولانی بسته در صف‌های شبکه را هدف قرار میدهد. Sprout از برخی از ویژگی‌های شبکه‌های سلولی استفاده می‌کند، کنترل ازدحام را در سناریوهایی خاص بهبود می‌بخشد. از آنجایی که شبکه‌های سلولی اغلب از bufferbloat رنج می‌برد، از گم شدن بسته‌ها برای سنجش ازدحام‌هایی استفاده میکنند که به تاخیر به شدت بالا منجر می‌شود. علاوه بر این، کاربران شبکه‌های سلولی در معرض صف‌های انباشته شده دیگر کاربران نیستند، چرا که حاملان معمولا صف‌های uplink and downlink مجزایی را برای هر دستگاه متعلق به سلول تنظیم می‌کنند. Sprout از حقیقت سنجش دوره ای ظرفیت ارتباطی و پیش بینی توزیع آینده آن در یک مدل مخفی مارکوف (HMM) استفاده میکند. تخمین ظرفیت موجود در عوض با حساب کردن بایت‌های دریافت شده در یک بازه طولانی و سپس تقسیم نتایج بر مدت زمان آن محاسبه می‌شود. این تخمین برای پیش بینی تعداد بسته‌هایی که برای ارسال بر روی لینک‌ها امن هستند، استفاده می‌شود، به همین ترتیب، تعداد درست بسته‌ها خیلی در صف منتظر نمی‌مانند. این یک پروتکل به شدت قابل سفارشی سازی است: مدل می‌تواند با تغییر یک پارامتر تهاجمی توان علیاتی گرا یا تاخیر گرا باشد. به هر حال، این رویکرد نیازمند یک بافر اختصاصی است، اگر جریان‌های متعدد یک بافر را به اشتراک گذارند، Sprout همه ظرفیت را به جریان‌های رقیب واگذار می‌کند.

D. مکانیزم‌های ترکیبی

برخی از مکانیزم‌های کنترل ازدحام تلاش می‌کنند که دو رویکرد را برای استفاده از مزایای هر دو ترکیب کنند. Compound TCP [99] یک مثال شناخته شده است، که در همه ماشین‌های ویندوز مایکروسافت به صورت پیش فرض در دسترس است. Compound از مجموع پنجره مبتنی بر تاخیر و یک پنجره Reno مبتنی بر گم شدن بسته به عنوان CWNDخود استفاده میکند، و به همین ترتیب می‌توانند با LFNها بهتر از مکانیزم‌های مبتنی بر گم شدن بسته خالص در زمان حفظ معیار عدالت در آن‌ها عمل کند، چرا که پنجره ترکیبی آن سریعتر از مورد مبتنی بر گم شدن بسته خالص رشد میکند، اما رفتار پروتکل زمانی که ازدحام را حس کند به کنترل ازدحام مبتنی بر گم شدن بسته تغییر می‌کند. Illinois [100] الگوریتم دیگری است که هردو تاخیر و گم شدن بسته را برای استنتاج ازدحام

در نظر می‌گیرد: در حالی که از گم شدن برای تشخیص ازدحام استفاده میکند، تغییر در تاخیر را برای تعیین نرخ تغییر در پنجره ازدحام استفاده می‌کند.



شکل 10: اصل پایه BBR: مکانیزم برای کار در نقطه عملیاتی بهینه به جای پر کردن بافر گلوگاه تلاش میکند.

این با CUBIC دوست است، اما، مانند Compound، CWND آن به صورت قابل توجهی در زمانی که خیلی کمتر از ظرفیت است، سریعتر رشد می‌کند؛ Veno [101]، یک ترکیب بین Vegas و Reno، از یک مدل صریح اشغال بافر گلوگاه برای کسب همان نوع رشد CWND استفاده می‌کند.

قابل توجه ترین رشد اخیر در الگوی کنترل ازدحام ترکیبی بدون شک الگوریتم BBR گوگل است، که رویکردهای متفاوتی را با تلاش برای تخمین پهنای باند و RTT اتخاذ میکند. این برای کسب نقطه عملیاتی بهینه با حفظ CWND برابر با BDP تلاش میکند؛ هر افزایش بیشتر در CWND تنها تاخیر را بدون هر مزیت توان عملیاتی افزایش می‌دهد، که در شکل 10 نشان داده شده است. در حالی که همگرایی به BDP با مکانیزم‌های سنتی غیر ممکن است، BBR به روش دیگری حداقل RTT و ظرفیت را با تخلیه دوره ای بافر تخمین می‌زند، سپس RTT را می‌سنجد، و سرانجام به نرخ تحویل اولیه باز میگردد. این مکانیزم در شکل 11 نشان داده شده است.

BBR می‌تواند پهنای باند را به صورت منصفانه با دیگر BBRها یا جریان‌های مبتنی بر گم شدن بسته به اشتراک گذارد [21]، اما تاخیر آن زمانی که ارتباط توسط یگر پروتکل‌های آگاه از تاخیر استفاده می‌شود کم است، چرا که

هیچ راهی برای تصاحب سهم منصفانه ظرفیت از جریان‌های مبتنی بر گم شدن بسته بدون داشتن تاخیرهای صف قابل توجه وجود ندارد. BBR در ارتباطات با ظرفیت بالا خوب کار می‌کند؛ به هر حال، حضور بافرهای کوچک باعث می‌شود که نقطه عملیاتی BBR بیش از حد افزایش یابد، بنابراین به گم شدن انبوه بسته‌ها و عدم رعایت عدالت نسبت به جریان‌های مبتنی بر گم شدن بسته‌ها منجر می‌شود.

عدالت و پایداری در زمان اشترک گذاری یک ارتباط با دیگر الگوریتم‌های کترل ازدحام دو مورد از بزرگترین مسائل BBR هستند. رقابت با دیگر الگوریتم‌ها می‌توانند باعث نوسانات شدید توان عملیاتی شود، و تهاجمی بودن BBR در برابر جریان‌های مبتنی بر گم شدن به شدت به اندازه بافر بستگی دارد: همانطور که گفته شد، در زمانی که بافر بسیار کوچک است مکانیزم بسیار تهاجمی می‌شود و زمانی که شبکه از Bufferbloat رنج می‌برد بسیار محافظه کار می‌شود. مسئله مشابهی توسط Farrow در مطالعه مقایسه ای بسیاری در محیط‌های شبکه ای ناهمگن از الگوریتم‌های کنترل ازدحام یعنی CUBIC، NewReno و BBR آورده شده است. تست‌های مجازی با جریان‌های عملیاتی متعدد هردو با الگوریتم‌های مشابه و متفاوت نویسنده را به سمت این نتیجه گیری سوق می‌دهند که NewReno و CUBIC به اندازه کافی منصفانه هستند، و توان عملیاتی قابل پیش بینی تر و پایدارتری را ارائه میدهند، در عین حال ادعا میشود که BBR ناعادلانه است و برای استفاده عموم آماده نیست.

شک‌هایی در مورد عملکرد BBR در شبکه‌های سلولی و تحت حرکت ظهور کرده اند، اما کار برای پذیرش این نوع محیط در حال حاضر در حال انجام است. BBR نیز برای QUIC پیشنهاد شده است.

E. رویکردهای یادگیری ماشین

در طی چند سال گذشته، یادگیری ماشین یک ابزار مهم برای طراحان شبکه و پروتکل است؛ پژوهش بر کنترل ازدحام مبتنی بر یادگیری ماشین همچنان در حال انجام است، و در حال حاضر بر چند مکانیزم در ادبیات موضوعی کار شده است.

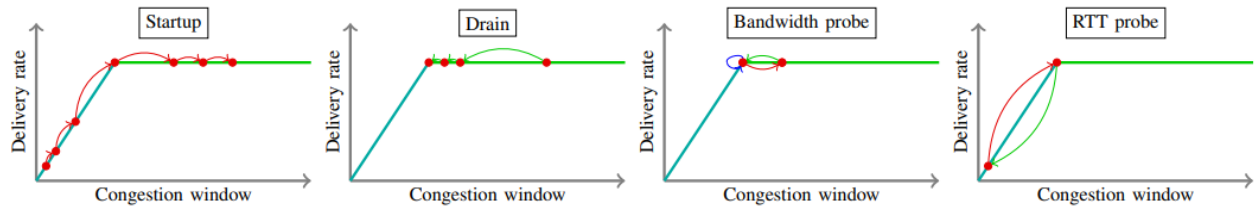
TCP Remy [102] اولین مثال از کنترل ازدحام مبتنی بر یادگیری ماشین است: نویسنده یک مدل مارکوف کانال را و یک تابع هدف را با یک پارامتر α تعریف می‌کند که می‌تواند برای تنظیم تهاجمی بودن پروتکل تنظیم شود، و از یک الگوریتم یادگیری ماشین برای تعریف رفتار مکانیزم کنترل ازدحام استفاده می‌کند. ورودی‌های داده شده به مکانیزم نرخ بین اخیرترین RTT و کمترین RTT سنجیده شده در طول ارتباط است، و میانگین حرکت موزون نمایی[۲۶] (EWMA) زمان بازه آخرین ACKها، و یک EWMA زمان ارسال همان بسته‌ها است. خود مکانیزم ضرورتا یک شبیه سازی مونت کارلو است، و محدودیت‌هایی دارد، چرا که نیازمند دانش قبلی در مورد سناریوی شبکه است و در فرضیات پایه ای عمل میکند که از همه جریان‌های رقیب استفاده میکند. یک نسخه پیشرفته تر Remy، به نام تلاش ردیابی بهینه[۲۷] (TAO)، مسئله آگاهی TCP را حل میکند و در جریان‌های رقابتی ناهمگن به خوبی اجرا می‌شود، اما هنوز نیازمند دانش قبلی گسترده در مورد شبکه‌ها است.

کنترل ازدحام عملکرد گرا (PCC) [104] دیگر مکانیزم یادگیری گرا است که از آزمایشات آنلاین به جای پیش آموزش آفلاین استفاده میکند: این از SACKها برای سنجش سودمندی اقدامات استفاده میکند، نرخ ارسال را منطبق با آن تنظیم میکند. PCC نسبت به نسخه‌های TCP مبتنی بر گم شدن بسته تهاجمی تر است، و با بافرهای کوتاه و RTTهای بالا به خوبی اجرا میشود، اما عملکرد آن در شبکه‌های بی سیم و لینک‌ها تحت تاثیر Bufferbloat تست نشده است. PCC Vivace [12] یک نسخه جدید از مکانیزم کنترل ازدحام با دوستی TCP بهبود یافته و یک مکانیزم یادگیری پیشرفته تر با استفاده از تکنیک‌های رگرسیون خطی است.

یک تلاش در استفاده از تکنیک‌های یادگیری نظارت شده پیشرفته تر برای طراحی مکانیزم‌های کنترل ازدحام در [153] آورده شده است، و یادگیری Q در [155][154] استفاده شده است. به هر حال، همه این الگوریتم‌ها برای توپولوژی‌های بسیار ساده شبکه و شرایط ساده شبکه بهینه سازی شده اند و به شدت به سناریوی در نظر گرفته شده بستگی دارند، لذا تا آنجا که می‌دانیم هیچ مکانیزم کنترل ازدحام به خوبی تست شده و کاملا اجرا شده ای با استفاده

[26] Exponentially Weighted Moving Average
[27] Tractable Attempt at Optimal

از این تکنیک‌های وجود ندارد. در این راستا، نویسندگان [105] از یک رویکرد یادگیری تقویت کننده برای بهبود عملکرد جریان‌های TCP بلند و کوتاه را بررسی کردند.



شکل 11: یک بررسی بر نقاط عملیاتی در وضعیت‌های الگوریتم BBR

به خصوص، رویکرد پیشنهادی، از دو فرآیند یادگیری متفاوت برای پذیرش پنجره اولیه ارتباطی استفاده میکند (تعداد بسته‌هایی که در زمان شروع ارتباط می‌توانند ارسال شوند) و الگوریتم کنترل ازدحام برای هر جریان واحد استفاده می‌شود. ارزیابی عملکرد بر اساس یک اجرای واقعی است، و ترافیک حقیقی یا ردهای ترکیبی آن از یک ارائه دهنده خدمات وب اصلی تولید شده است. سرانجام، QTCP یک رویکرد یادگیری تقویتی را می‌پذیرد اما اقدامات که در دسترس هستند مستقیما به ابعاد پنجره ازدحام مربوط هستند. نویسندگان، عملکرد الگوی آن‌ها با TCP NewReno را مقایسه کردند.

یک بحث دقیق تر بر ملاحظات طراحی برای مکانیزم‌های کنترل ازدحام مبتنی بر یادگیری ماشین جدید را می‌توانید در [156] ببینید.

F. رویکردهای میان لایه‌ای

همانطور گه گفته شد، TCP بر یک انتزاع از هاب‌های متعدد بین دو نقطه انتهایی ارتباطی برای اجرای عملیات کنترل ازدحام متکی است. این، به هر حال، گاهی با توجه به اینکه TCP ممکن است در اتخاذ مشخصه‌های خاص رسانه با شکست مواجه شود، خیلی بهینه نیست. بنابراین، رویکردهای میان لایه‌ای طراحی الگوریتم‌های کنترل ازدحامی را که از اطلاعات دقیق تر و اضافی ارائه شده توسط لایه‌های پایین تر پشته پروتکلی ارائه میدهد را هدف قرار میدهد، و به خصوص در دامنه‌های بی سیم بیشتر عمومیت دارد.

چندین مقاله اخیرا، راه حل‌های متقابلی را برای مقابله با شبکه‌های سلولی فعال در فرکانس‌های mmWave پیشنهاد دادند. در [158] ,[157]، نویسندگان پنجره ازدحام TCP را برای BDP ارتباطی تنظیم کردند، و اطلاعات لایه کنترل دسترسی به رسانه (MAC) و لایه فیزیکی را در پایانه موبایل، برای ارتباطات TCP downlink and uplink ردیابی کردند. مقاله [159] یک مکانیزم مبتنی بر پروکسی را در ایستگاه پایه ای که ACKها را ردیابی میکند پیشنهاد داد و مقدار پنجره اعلان شده را برای اجبار ارسال کننده TCP اصلاح نشده به ردیابی BDP ارتباطات انتها به انتها با یک لینک mmWave تغییر داد.

پارادایم SDN امکانات جدیدی را برای رویکردهای میان لایه‌ای فراهم کرده است، که به لایه انتقال امکان رزرو منابع برای اپلیکیشن‌هایی با تاخیر کم با برقراری ارتباط مستقیم با کنترلر شبکه را می‌دهد.

دیگر رویکر لایه مقابل ممکن برای در نظر گرفتن نیازهای اپلیکیشن: چندین نسخه TCP که اعوجاج نرخ و نوع محتوا را در جریان چند رسانه ای را در نظر می‌گیرند پیشنهاد شدند.

در [164]، نویسندگان مجدد بر یک رویکرد میان لایه‌ای برای شبکه‌های سلولی 3G and 4G اعمال شده به کنترل ازدحام QUIC، با اصلاحاتی در ارسال کننده و گیرنده تجدید نظر کرده اند.

G. شبکه مرکز داده و مسئله Incast

همانطور که در بخش II-B تشریح شده است، مراکز داده یک محیط شبکه ای بسیار عجیب است، که نیازمند چندین تنظیم بر لایه انتقال برای اجتناب از مسائلی مانند مسئله Incast هستند. در طی چد سال گذشته، تلاش‌هایی برای کاهش این مسئله صورت گرفته است. DCTCP [45] یک الگوریتم کنترل ازدحام بر اساس ECN است، که پنجره ازدحام را بر اساس مقدار ازدحام تجربه شده کاهش میدهد، که فضای بافر کافی برای اجتناب از مسئله Incast را حفظ میکند.

گزینه دیگر رها کردن کنترل ازدحام استاندارد و استفاده از کنترل نرخ صریح است، همانطور که در الگوی تحویل با مهلت مقرر[۲۸] (D3)، که می‌تواند مسئل را حل کند اگر پشتیبان کامل شبکه در دسترس باشد، دیده شده است. این می‌تواند با استفاده از راه حل‌های SDN و NFV اجرا شود، اما مقیاس پذیری همچنان مشکل است، و D3 نسبت به انفجار بسته‌ها آسیب پذیر است. TCP مراکز داده آگاه از مهلت مقرر (D2TCP)، یک الگوی توزیع شده است که مهلت‌های مقرر سطح اپلیکیشن را برای بلوک‌های داده مانند D3 در نظر می‌گیرد، در حالی که از نیاز به پشتیبانی کامل شبکه و آسیب پذیری نسبت به ترافیک انفجاری اجتناب میکند. این به DCTCP شبیه است، اما در زمان اجرای اجتناب از ازدحام آگاهی کاملی از مهلت مقرر دارد.

الگوی pFabric [168] براساس دورانداختن زودهنگام دیگر تکنیک پشتیبان شبکه است، سوئیچ‌ها دور انداختن اولویت دار را با بافرهای بسیار کوچک اجرا میکنند، و حتی اگر جریان‌ها تهاجمی باشند و تا زمانی که نرخ گم شدن بسته به صورت سازگاری بالا است back off رخ نمی‌دهد، زمان بندی جریان اجرا می‌شود. در این نوع سناریو، جریان‌ها نیاز دارند تهاجمی باشند، یا مکانیزم دور انداختن زود هنگام؛ توان عملیاتی آن‌ها را تا زیر حد قابل حصول کاهش میدهد. پروتکل pHost [169] جایگزاری عملکرد pFabric's را بدون هر پشتیبانی شبکه هدف قرار میدهد، و از یک الگوی مبتنی بر توکن در گیرنده‌ها برای انتساب اولویت‌ها به ارسال کننده‌ها استفاده میکند. گیرنده‌ها جریان را با کمترین بایت باقی مانده در زمان انتساب توکن‌ها اولویت بندی می‌کنند، عملکرد نزدیک بهینه ای را کسب میکنند. ExpressPass [170] تکنیک دیگری است که از بسته‌های اعتباری سمت گیرنده برای کنترل نرخ سمت ارسال کننده استفاده میکند؛ بسته‌های اعتباری می‌توانند بدون هیچ پیامدی گم شوند، و نرخ گم شدن برای سنجش ظرفیت اعتبار استفاده می‌شود. Homa [171] یک پروتکل بدون غیراتصال گرای جدید است که از pHost و ExpressPass الهام گرفته است، که می‌تواند به صورت قابل توجهی تاخیر را بدون پشتیبانی شبکه کاهش دهد. این به صورت تهاجمی از صف اولویت در سوئیچ، با استفاده از کنترل جریان گیرنده محور برای انتساب اولویت و حد صریح مسئله Incast با حساب درخواست‌های برون سپاری شده و نشانه گذاری بسته‌های جدید برای کاهش اولویت آن‌ها

[28] Deadline- Driven Delivery

اگر با یک طول مشخص باشند، استفاده میکند. این ماهیت غیراتصال گرا HoL blocking را محدود میکند و از تایید صریح اجتناب میکند، و سهم بیشتری در کارآمدی پروتکل دارد.

الویت بندی، داوری، و نقاط انتهایی خود تنظیم (PASE) یک استراتژی است که رویکردهای DCTCP, D3, and pFabric را ترکیب میکند. این از همه سه مکانیزم در مقیاس‌های متفاوت استفاده میکند: کنترل صریح نرخ در مقیاس زمانی و با دقت پایین، چرا که خود نقاط انتهایی میتوانند ک تخصیص بهینه را با کاوش برطبق الویت نسبت داده شده پیدا کنند. سرانجام، اولویت بندی شبه pFabric و دور انداختن بسته‌ها در مقیاس بسیار کوتاه اجرا میشود. این رویکرد ترکیبی میتواند در هر پذیرش بهتر اجرا شود.

برای بحث کامل تر بر شبکه‌های مراکز داده و مسائل لایه انتقال آن ها، خواننده را به [175][173][33] ارجاع می‌دهیم.

H. جهات پژوهشی و چالش‌های باز

در بیشتر پروتکل‌های انتقال کنترل ازدحام یک مولفه مهم در عملیات است. TCP, SCTP, and QUIC این را طبیعی اجرا میکند، و DCCP تکاملی را در طراحی پروتکل‌های انتقال غیرقابل اعتماد که برای ازدحام در نظر گرفته می‌شود نشان میدهد. پژوهش در کنترل ازدحام به عنوان تکامل سریعی در چند جهت دیده می‌شود: در حالی که رویکردهای مبتنی بر گم شدن بسته، تاخیر و ظرفیت همه به عنوان سنگ بنای پروتکل‌های کارآمدتر استفاده می‌شوند، پروتکل‌های میان لایه‌ای و مبتنی بر یادگیری ماشین جدید جذب علایق پژوهشی قابل توجهی را آغاز کردند.

روند اصلی در این عرصه به غلبه بر رویکردهای مناسب برای همه کنترل‌های ازدحام در برابر الگوریتم‌هایی که استفاده از یک مورد خاص را هدف قرار میدهند مربوط است. برای مثال، چند پروتکل مبتنی بر تاخیر صف کم را هدف قرار میدهند، که به تاخیر انتها به انتهای کم تفسیر می‌شود. به هر حال، به روزرسانی‌ها در مکانیزم کنترل ازدحام TCP نیازمند اصلاحاتی در کرنل است، و بنابراین یک به روزرسانی سیستم عامل صورت میگیرد، که یک کار دست و پاگیر است. توسعه پروتکل‌های فضای کاربر، مانند QUIC، به محققین اجازه میدهد که چیزهای بیشتری را آزمایش کنند،

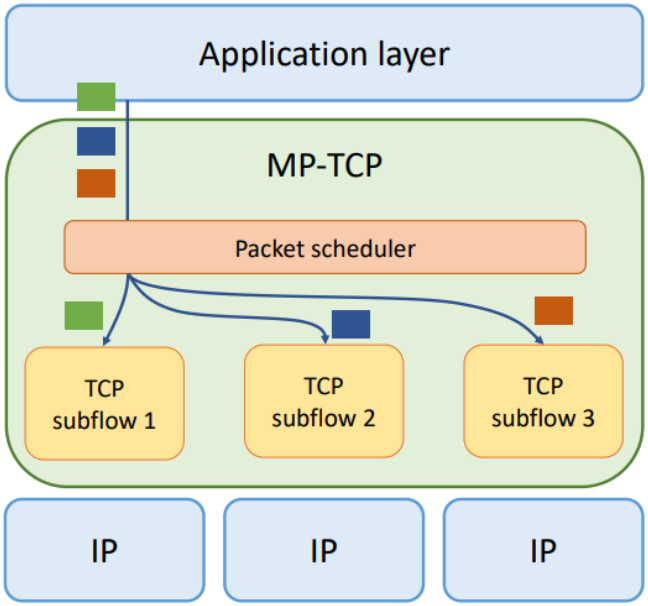
و شبکه از طریق SDN حمایت شود و قطعه بندی شبکه توسعه مکانیزم‌هایی که به اندازه CUBIC تهاجمی نیستند را هدف قرار میدهد، چرا که آن‌ها میتوانند در بافرهای مجزا ایزوله شوند و بنابراین نیاز نیست با آن‌ها مقابله کنیم.

5. پروتکل‌های انتقال چند مسیره

این روزها، بیشتر دستگاه‌ها می‌توانند از تکنولوژی‌های ارتباطی متعدد همزمان استفاده کنند: برای مثال، تلفن‌های هوشمند مدرن می‌توانند هردو WiFi و LTE را به یکدیگر متصل کنند. به همین دلیل، ارتباطات چند مسیره موضوع مورد توجهی در چند سال اخیر بوده است. در لایه انتقال، پروتکل‌های چند مسیره نیاز است که برای بهره برداری موفقیت آمیز از مزایای تنوع چند مسیره طراحی شوند، اما این کار هیچ وقت ساده نبوده است.

A. MPTCP

MPTCP یک بسط سازگار عقب گرد از TCP است، که به عنوان استاندارد آزمایشی در سال 2011 [25], [26], [176] منتشر شده است و حال به صورت گسترده مستقر شده است [177]. این به اپلیکیشن‌ها اجازه میدهد که از ارتباطات متعددی به صورت همزمان بدون هیچ تغییری در سوکت API استفاده کنند.



شکل 12: پشته پروتکلی MPTCP

استفاده از ارتباطات متعدد می‌تواند کل ظرفیت را بهبود بخشد، و افزونگی را در برابر شکست‌های لینک فراهم کند، و بار بر مسیرهای مزدحم را کاهش دهد. به منظور تضمین سازگاری عقب گرد و شفافیت در اپلیکیشن ها، MPTCP نیاز است که در کرنل سیستم عامل اجرا شود.

شکل 12 معماری پایه هاست MPTCP را نشان میدهد: ارتباط ترکیبی از دو جریان TCP مجزا در مسیرهای متفاوت است، هریک دارای کنترل ازدحام خود و مکانیزم ACKing است. استفاده از جریان‌های TCP تک مسیره ضروری است چرا که بسیاری از جعبه‌های میانی ترافیک TCP را بررسی میکنند و بسته‌ها با رفتارهای ناسازگار و با عملیات طبیعی پروتکل به دلایل امنیتی دور انداخته می‌شوند.

یک سشن MPTCP با یک زیرجریان TCP واحد آغاز می‌شود: هاست‌های توانمند MPTCP می‌توانند یک دیگر را با تنظیم گزینه MP_CAPABLE در اولین بسته دست تکان دادن ارتباط شناسایی کنند. دو هاست سپس کلیدهای رمزنگاری را برای افزودن زیرجریان‌های جدید امن تبادل میکنند؛ این میتواند با تنظیم گزینه MP_JOIN و استفاده از هش کلیدهای ارتباطی در طول دست تکان دادن TCP استاندارد ایجاد شود. MPTCP از اضافه و حذف آدرس‌ها در هاست، به صورت صریح و ضمنی حمایت میکند.

MPTCP بسته‌هایی را به دو شماره توالی اختصاص میدهند: شماره تولید استاندارد برای هر زیر جریان و یک شماره توالی داده سطح ارتباط (DSN) نسبت داده شده اند. از آنجایی که کنترل ازدحام و انتقال مجدد نیاز است که در سطح زیر جریان رخ دهد، چرا که هر زیر جریان یک ارتباط full-fledged TCP در نوع خود است، و شماره‌های توالی سطح جریان نیاز است که متوالی باشند و داده در هر زیر جریان باید به ترتیب تحویل داده شود؛ DSN سطح ارتباط نیاز است که دو جریان داده مجزا را به هم پیوند زند و داده اصلی را بسازد. [179]. انتقال مجدد همان داده بر زیرجریان‌های متعدد امکان پذیر است، چرا که DSN سطح ارتباطی می‌تواند بسته را در سطح ارتباط شناسایی کند و HoL را کاهش دهد اگر زیرجریان مسدود شود.

REFERENCES

[1] J. Postel, "Transmission Control Protocol," IETF, RFC 793, Sep. 1981. [Online]. Available: https://rfc-editor.org/rfc/rfc793.txt

[2] ——, "Internet Protocol," IETF, RFC 791, Sep. 1981. [Online]. Available: https://rfc-editor.org/rfc/rfc791.txt

[3] ——, "User Datagram Protocol," RFC 768, Aug. 1980. [Online]. Available: https://rfc-editor.org/rfc/rfc768.txt

[4] Cisco, "Cisco Visual Networking Index, forecast and methodology, 2016–2021," Tech. Rep., June 2017.

[5] 3GPP, "NR and NG-RAN Overall Description - Rel. 15," TS 38.300, Jan. 2018.

[6] S. Parkvall, E. Dahlman, A. Furuskar, and M. Frenne, "NR: The new 5G Radio Access Technology," IEEE Communications Standards Magazine, vol. 1, no. 4, pp. 24–30, Dec 2017.

[7] B. Bellalta, "IEEE 802.11ax: High-efficiency WLANs," IEEE Wireless Communications, vol. 23, no. 1, pp. 38–46, February 2016.

[8] Y. Ghasempour, C. R. C. M. da Silva, C. Cordeiro, and E. W. Knightly, "IEEE 802.11ay: Next-generation 60 GHz communication for 100 Gb/s Wi-Fi," IEEE Communications Magazine, vol. 55, no. 12, pp. 186–192, Dec. 2017.

[9] Q. D. Coninck, M. Baerts, B. Hesmans, and O. Bonaventure, "Observing real smartphone applications over multipath TCP," IEEE Communications Magazine, vol. 54, no. 3, pp. 88–93, Mar. 2016.

[10] R. Alvizu, G. Maier, N. Kukreja, A. Pattavina, R. Morro, A. Capello, and C. Cavazzoni, "Comprehensive survey on T-SDN: SoftwareDefined Networking for transport networks," IEEE Communications Surveys and Tutorials, vol. 19, no. 4, pp. 2232–2283, Jan. 2017.

[11] X. Liu, A. Sridharan, S. Machiraju, M. Seshadri, and H. Zang, "Experiences in a 3G network: Interplay between the wireless channel and applications," in 14th ACM International Conference on Mobile Computing and Networking (MobiCom), San Francisco, California, USA, Sep. 2008, pp. 211–222.

[12] R. Stewart, "Stream Control Transport Protocol," IETF, RFC 6582, Sep. 2007. [Online]. Available: https://rfc-editor.org/rfc/rfc6582.txt

[13] P. Srisuresh, J. Kuthan, J. Rosenberg, A. Molitor, and A. Rayhan, "Middlebox communication architecture and framework," IETF, RFC 3303, Aug. 2002. [Online]. Available: https://rfc-editor.org/rfc/rfc3303. txt

[14] B. Carpenter and S. Brim, "Middleboxes: Taxonomy and issues," IETF, RFC 3234, Feb. 2002. [Online]. Available: https://rfc-editor.org/ rfc/rfc3234.txt

[15] K. Edeline and B. Donnet, "A first look at the prevalence and persistence of middleboxes in the wild," in 29th IEEE International Teletraffic Congress (ITC), vol. 1, Genoa, Italy, Sep. 2017, pp. 161– 168.

[16] G. Papastergiou, G. Fairhurst, D. Ros, A. Brunstrom, K. Grinnemo, P. Hurtig, N. Khademi, M. Tüxen, M. Welzl, D. Damjanovic, and S. Mangiante, "De-ossifying the Internet transport layer: A survey and future perspectives," IEEE Communications Surveys and Tutorials, vol. 19, no. 1, pp. 619–639, Feb. 2017.

[17] H.-S. Park, J.-Y. Lee, and B.-C. Kim, "TCP performance issues in LTE networks," in International Conference on ICT Convergence (ICTC). Seoul, South Korea: IEEE, Sep. 2011, pp. 493–496.

[18] M. Zhang, M. Polese, M. Mezzavilla, J. Zhu, S. Rangan, S. Panwar, and a. M. Zorzi, "Will TCP work in mmWave 5G cellular networks?" IEEE Communications Magazine, vol. 57, no. 1, pp. 65–71, Jan. 2019.

[19] M. Scharf and S. Kiesel, "Head-of-line blocking in TCP and SCTP: Analysis and measurements," in IEEE Global Communications Conference (GLOBECOM), San Francisco, California, USA, Nov. 2006, pp. 1–5.

[20] J. Gettys and K. Nichols, "Bufferbloat: Dark buffers in the Internet," ACM Queue, vol. 9, no. 11, pp. 40:40–40:54, Nov. 2011.

[21] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, "BBR: Congestion-based congestion control," ACM Queue, vol. 14, no. 5, pp. 20–53, Sep. 2016.

[22] M. Hock, F. Neumeister, M. Zitterbart, and R. Bless, "TCP LoLa: Congestion control for low latencies and high throughput," in 42nd IEEE Conference on Local Computer Networks (LCN), Singapore, Oct. 2017, pp. 215–218.

[23] E. Kohler, M. Handley, and S. Floyd, "Datagram Congestion Control Protocol (DCCP)," IETF, RFC 4340, March 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4340.txt
[24] J. Iyengar and M. Thomson, "QUIC: A UDP-based multiplexed and secure transport," IETF, Working Draft: draft-ietf-quic-transport-08, Dec. 2017. [Online]. Available: https://tools.ietf.org/id/draft-ietf-quic-transport-08.txt
[25] A. Ford, C. Raiciu, M. Handley, S. Barre, and J. Iyengar, "Architectural guidelines for multipath TCP development," IETF, RFC 6182, Mar. 2011. [Online]. Available: https://rfc-editor.org/rfc/rfc6182.txt
[26] C. Raiciu, M. J. Handley, and D. Wischik, "Coupled congestion control for multipath transport protocols," IETF, RFC 6356, Oct. 2011. [Online]. Available: https://rfc-editor.org/rfc/rfc6356.txt
[27] T. D. Wallace and A. Shami, "Concurrent multipath transfer using SCTP: Modelling and congestion window management," IEEE Transactions on Mobile Computing, vol. 13, no. 11, pp. 2510–2523, Nov. 2014.
[28] Q. D. Coninck and O. Bonaventure, "Multipath extension for QUIC," IETF, Working Draft: deconinck-multipath-quic-00, Oct. 2017. [Online]. Available: https://datatracker.ietf.org/doc/html/ draft-deconinck-multipath-quic-00
[29] T. Viernickel, A. Froemmgen, A. Rizk, B. Koldehofe, and R. Steinmetz, "Multipath QUIC: A deployable multipath transport protocol," in IEEE International Conference on Communications (ICC), May 2018, pp. 1–7.
[30] F. Chiariotti, S. Kucera, A. Zanella, and H. Claussen, "Analysis and design of a latency control protocol for multi-path data delivery with pre-defined QoS guarantees," IEEE/ACM Transactions on Networking, vol. 27, no. 3, pp. 1165–1178, Jun. 2019.
[31] M. Li, A. Lukyanenko, Z. Ou, A. Ylä-Jääski, S. Tarkoma, M. Coudron, and S. Secci, "Multipath transmission for the internet: A survey," IEEE Communications Surveys and Tutorials, vol. 18, no. 4, pp. 2887–2925, Jun. 2016.
[32] C. Xu, J. Zhao, and G. M. Muntean, "Congestion control design for multipath transport protocols: A survey," IEEE Communications Surveys and Tutorials, vol. 18, no. 4, pp. 2948–2969, Apr. 2016.
[33] R. Rojas-Cessa, Y. Kaymak, and Z. Dong, "Schemes for fast transmission of flows in data center networks," IEEE Communications Surveys and Tutorials, vol. 17, no. 3, pp. 1391–1422, Aug. 2015.
[34] T. Lukaseder, L. Bradatsch, B. Erb, R. W. Van Der Heijden, and F. Kargl, "A comparison of TCP congestion control algorithms in 10G networks," in 41st IEEE Conference on Local Computer Networks (LCN), Dubai, UAE, Nov. 2016, pp. 706–714.
[35] Y. Gong, D. Rossi, C. Testa, S. Valenti, and M. D. Täht, "Fighting the bufferbloat: On the coexistence of AQM and low priority congestion control," in IEEE Conference on Computer Communications (INFOCOM), Turin, Italy, Apr. 2013, pp. 3291–3296.
[36] Z. Liu, J. Sun, S. Hu, and X. Hu, "An adaptive AQM algorithm based on a novel information compression model," IEEE Access, vol. 6, pp. 31 180–31 190, Jun. 2018.
[37] C. A. Grazia, N. Patriciello, M. Klapez, and M. Casoni, "A crosscomparison between TCP and AQM algorithms: Which is the best couple for congestion control?" in 14th IEEE International Conference on Telecommunications (ConTEL), Zagreb, Croatia, Jun. 2017, pp. 75– 82.
[38] K. Nichols and V. Jacobson, "Controlling queue delay," Communications of the ACM, vol. 55, no. 7, pp. 42–50, Jul. 2012.
[39] R. Amrutha and V. Nithya, "Curbing of TCP Incast in data center networks," in 4th IEEE International Conference on Reliability, Infocom Technologies and Optimization (ICRITO) (Trends and Future Directions), Noida, India, Sep. 2015, pp. 1–5.
[40] J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," in Usenix 6th Symposium on Operating System Design and Implementation (OSDI), San Francisco, California, USA, Dec. 2004, pp. 137–150.
[41] L. Page, S. Brin, R. Motwani, and T. Winograd, "The PageRank citation ranking: Bringing order to the web," Stanford InfoLab, Tech. Rep., Nov. 1999.

[42] W. Chen, F. Ren, J. Xie, C. Lin, K. Yin, and F. Baker, "Comprehensive understanding of TCP Incast problem," in IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, Apr. 2015, pp. 1688–1696.
[43] Y. Ren, Y. Zhao, P. Liu, K. Dou, and J. Li, "A survey on TCP Incast in data center networks," International Journal of Communication Systems, vol. 27, no. 8, pp. 1160–1172, Aug. 2014.
[44] J. Zhang, F. Ren, L. Tang, and C. Lin, "Modeling and solving TCP Incast problem in data center networks," IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 2, pp. 478–491, Feb. 2015.
[45] M. Alizadeh, A. Greenberg, D. A. Maltz, J. Padhye, P. Patel, B. Prabhakar, S. Sengupta, and M. Sridharan, "Data Center TCP (DCTCP)," in ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM). New Delhi, India: ACM, 2010, pp. 63–74.
[46] J. Hwang, J. Yoo, and N. Choi, "IA-TCP: A rate based Incast-avoidance algorithm for TCP in data center networks," in IEEE International Conference on Communications (ICC), Ottawa, Canada, Jun. 2012, pp. 1292–1296.
[47] J. Zhang, F. Ren, and C. Lin, "Survey on transport control in data center networks," IEEE Network, vol. 27, no. 4, pp. 22–26, Jul. 2013.
[48] J. Iyengar and I. Swett, "QUIC loss detection and congestion control," IETF, Working Draft: draft-ietf-quic-recovery-08, Dec. 2017. [Online]. Available: https://tools.ietf.org/id/draft-ietf-quic-recovery-08.txt
[49] M. Seemann and L. Clemente. (2019, Apr.) QUIC implementation in Go. [Online]. Available: github.com/lucas-clemente/quic-go
[50] D. Tikhonov and B. Prodoehl. (2017, Sep.) LiteSpeed QUIC. [Online]. Available: github.com/litespeedtech/lsquic-client
[51] J. Lee, B. Hong et al. (2016, Sep.) libquic. [Online]. Available: github.com/devsisters/libquic
[52] R. Shade et al. (2016, Oct.) proto-quic. [Online]. Available: github.com/google/proto-quic
[53] G. Fairhurst, "The Datagram Congestion Control Protocol (DCCP) Service Codes," IETF, RFC 5595, Sep. 2009. [Online]. Available: https://rfc-editor.org/rfc/rfc5595.txt
[54] ——, "Datagram Congestion Control Protocol (DCCP) simultaneousopen technique to facilitate NAT/middlebox traversal," IETF, RFC 5596, Sep. 2009. [Online]. Available: https://rfc-editor.org/rfc/rfc5596.txt
[55] S. Floyd, J. Mahdavi, M. Mathis, and D. A. Romanow, "TCP Selective Acknowledgment options," IETF, RFC 2018, Oct. 1996. [Online]. Available: https://rfc-editor.org/rfc/rfc2018.txt
[56] M. Belshe, M. Thomson, and R. Peon, "Hypertext Transfer Protocol version 2," IETF, RFC 7540, May 2015. [Online]. Available: https://rfc-editor.org/rfc/rfc7540.txt
[57] S. McQuistin, C. Perkins, and M. Fayed, "TCP Hollywood: An unordered, time-lined, TCP for networked multimedia applications," in IEEE/IFIP Networking Conference (IFIP Networking), Vienna, Austria, May 2016, pp. 422–430.
[58] Y. Cui, T. Li, C. Liu, X. Wang, and M. Kühlewind, "Innovating transport with QUIC: Design approaches and research challenges," IEEE Internet Computing, vol. 21, no. 2, pp. 72–76, Mar. 2017.
[59] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," IEEE/ACM Transactions on Networking, vol. 5, no. 6, pp. 756–769, Dec 1997.
[60] G. Xylomenos, G. C. Polyzos, P. Mahonen, and M. Saaranen, "TCP performance issues over wireless links," IEEE Communications Magazine, vol. 39, no. 4, pp. 52–58, April 2001.
[61] J. Griner, J. L. Border, M. Kojo, Z. D. Shelby, and G. Montenegro, "Performance enhancing proxies intended to mitigate link-related degradations," IETF, RFC 3135, Jun. 2001. [Online]. Available: https://rfc-editor.org/rfc/rfc3135.txt
[62] K. Liu and J. Y. Lee, "On improving TCP performance over mobile data networks," IEEE Transactions on Mobile Computing, vol. 15, no. 10, pp. 2522–2536, Oct. 2016.
[63] M. Maity, B. Raman, and M. Vutukuru, "TCP download performance in dense WiFi scenarios: Analysis and solution," IEEE Transactions on Mobile Computing, vol. 16, no. 1, pp. 213–227, Jan. 2017.

[64] B. Sardar and D. Saha, “A survey of TCP enhancements for lasthop wireless networks,” IEEE Communications Surveys and Tutorials, vol. 8, no. 3, pp. 20–34, Jul. 2006.
[65] S. Rangan, T. S. Rappaport, and E. Erkip, “Millimeter-wave cellular wireless networks: Potentials and challenges,” Proceedings of the IEEE, vol. 102, no. 3, pp. 366–385, Mar. 2014.
[66] M. Zhang, M. Mezzavilla, R. Ford, S. Rangan, S. S. Panwar, E. Mellios, D. Kong, A. R. Nix, and M. Zorzi, “Transport Layer Performance in 5G mmWave Cellular,” in IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), San Francisco, California, USA, Apr. 2016, pp. 730–735.
[67] M. Polese, R. Jana, and M. Zorzi, “TCP and MP-TCP in 5G mmWave networks,” IEEE Internet Computing, vol. 21, no. 5, pp. 12–19, Sep. 2017.
[68] H. Jiang, Z. Liu, Y. Wang, K. Lee, and I. Rhee, “Understanding bufferbloat in cellular networks,” in ACM SIGCOMM Workshop on Cellular Networks: Operations, Challenges, and Future Design (CellNet), Helsinki, Finland, Aug. 2012, pp. 1–6.
[69] A. M. Kakhki, S. Jero, D. Choffnes, C. Nita-Rotaru, and A. Mislove, “Taking a long look at QUIC: An approach for rigorous evaluation of rapidly evolving transport protocols,” in ACM Internet Measurement Conference (IMC), London, United Kingdom, Nov. 2017, pp. 290–303.
[70] A. Langley, A. Riddoch, A. Wilk, A. Vicente, C. Krasic, D. Zhang, F. Yang, F. Kouranov, I. Swett, J. Iyengar, J. Bailey, J. Dorfman, J. Roskind, J. Kulik, P. Westin, R. Tenneti, R. Shade, R. Hamilton, V. Vasiliev, W.-T. Chang, and Z. Shi, “The QUIC transport protocol: Design and Internet-scale deployment,” in ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM). Los Angeles, California, USA: ACM, Aug. 2017, pp. 183–196.
[71] E. Rescorla, “The Transport Layer Security (TLS) protocol version 1.3,” IETF, RFC 8446, Aug. 2018. [Online]. Available: https: //rfc-editor.org/rfc/rfc8446.txt
[72] G. Carlucci, L. D. Cicco, and S. Mascolo, “HTTP over UDP: An experimental investigation of QUIC,” in 30th Annual ACM Symposium on Applied Computing (SAC), Salamanca, Spain, Apr. 2015, pp. 609– 614.
[73] P. Megyesi, Z. Krämer, and S. Molnár, “How quick is QUIC?” in IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, May 2016, pp. 1–6.
[74] J. Rüth, I. Poese, C. Dietzel, and O. Hohlfeld, “A first look at QUIC in the wild,” in Passive and Active Measurement Conference (PAM), Berlin, Germany, 2018, pp. 255–268.
[75] A. D. Biasio, F. Chiariotti, M. Polese, A. Zanella, and M. Zorzi, “A QUIC implementation for ns-3,” in Workshop on ns-3 (WNS3), Florence, Italy, Jun. 2019, pp. 1–7.
[76] P. Natarajan, F. Baker, P. D. Amer, and J. T. Leighton, “SCTP: What, why, and how,” IEEE Internet Computing, vol. 13, no. 5, pp. 81–85, Sep. 2009.
[77] R. Stewart and C. Metz, “SCTP: New transport protocol for TCP/IP,” IEEE Internet Computing, vol. 5, no. 6, pp. 64–69, Nov. 2001.
[78] S. Fu and M. Atiquzzaman, “SCTP: state of the art in research, products, and technical challenges,” IEEE Communications Magazine, vol. 42, no. 4, pp. 64–76, April 2004.
[79] A. L. Caro, J. R. Iyengar, P. D. Amer, S. Ladha, G. J. Heinz, and K. C. Shah, “SCTP: A proposed standard for robust Internet data transport,” IEEE Computer, vol. 36, no. 11, pp. 56–63, Nov. 2003.
[80] P. T. Conrad, G. J. Heinz, A. L. Caro, P. D. Amer, and J. Fiore, “SCTP in battlefield networks,” in Military Communications Conference (MILCOM), vol. 1, McLean, Virginia, USA, oct 2001, pp. 289–295.
[81] A. Jungmaier, M. Schopp, and M. Tuxen, “Performance evaluation of the simple control transmission protocol (SCTP),” in IEEE Conference on High Performance Switching and Routing (HPSR), Heidelberg, Germany, Jun. 2000, pp. 141–148.
[82] J. Shi, Y. Jin, H. Huang, and D. Zhang, “Experimental performance studies of SCTP in wireless access networks,” in International Conference on Communication Technology (ICCT), vol. 1, Beijing, China, Apr. 2003, pp. 392–395.
[83] G. Camarillo, R. Kantola, and H. Schulzrinne, “Evaluation of transport protocols for the Session Initiation Protocol,” IEEE Network, vol. 17, no. 5, pp. 40–46, Sep. 2003.

[84] E. Kohler, M. Handley, and S. Floyd, “Designing dccp: Congestion control without reliability,” in ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), Pisa, Italy, 2006, pp. 27–38.
[85] L. Eggert, G. Fairhurst, and G. Shepherd, “UDP usage guidelines,” IETF, RFC 8085, Mar. 2017. [Online]. Available: https://rfc-editor. org/rfc/rfc8085.txt
[86] S. Floyd, M. Handley, and E. Kohler, “Problem Statement for the Datagram Congestion Control Protocol (DCCP),” IETF, RFC 4336, Mar. 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4336. txt
[87] K. Ramakrishnan, S. Floyd, and D. Black, “The Addition of Explicit Congestion Notification (ECN) to IP,” IETF, RFC 3168, Sep. 2001. [Online]. Available: http://www.rfc-editor.org/rfc/rfc3168.txt
[88] A. Gurtov, T. Henderson, S. Floyd, and Y. Nishida, “The NewReno modification to TCP’s Fast Recovery algorithm,” IETF, RFC 6582, Apr. 2012. [Online]. Available: https://rfc-editor.org/rfc/rfc6582.txt
[89] L. Xu, K. Harfoush, and I. Rhee, “Binary increase congestion control (BIC) for fast long-distance networks,” in IEEE Conference on Computer Communications (INFOCOM), vol. 4, Hong Kong, China, Mar. 2004, pp. 2514–2524.
[90] S. Ha, I. Rhee, and L. Xu, “CUBIC: A new TCP-friendly high-speed TCP variant,” ACM Operating Systems Review, vol. 42, no. 5, pp. 64– 74, Jul. 2008.
[91] A. Abdelsalam, M. Luglio, C. Roseti, and F. Zampognaro, “TCP Wave: a new reliable transport approach for future Internet,” Computer Networks, vol. 112, pp. 122–143, Jan. 2017.
[92] L. S. Brakmo, S. W. O’Malley, and L. L. Peterson, “TCP Vegas: New techniques for congestion detection and avoidance,” ACM Computer Communication Review, vol. 24, no. 4, pp. 24–35, Oct. 1994.
[93] J. Sing and B. Soh, “TCP New Vegas: Performance evaluation and validation,” in 11th IEEE Symposium on Computers and Communications (ISCC), Cagliari, Italy, Jun. 2006, pp. 541–546.
[94] Y. Zaki, T. Pötsch, J. Chen, L. Subramanian, and C. Görg, “Adaptive congestion control for unpredictable cellular networks,” in ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM), London, UK, 2015, pp. 509–522.
[95] P. Goyal, A. Narayan, F. Cangialosi, D. Raghavan, S. Narayana, M. Alizadeh, and H. Balakrishnan, “Elasticity Detection: A Building Block for Delay-Sensitive Congestion Control,” arXiv preprint arXiv:1802.08730, pp. 1–17, Jul. 2018. [Online]. Available: https://arxiv.org/abs/1802.08730
[96] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuehlewind, “Low Extra Delay Background Transport (LEDBAT),” IETF, RFC 6817, Dec. 2012. [Online]. Available: https://rfc-editor.org/rfc/rfc6817.txt
[97] S. Mascolo, C. Casetti, M. Gerla, M. Y. Sanadidi, and R. Wang, “TCP Westwood: Bandwidth estimation for enhanced transport over wireless links,” in 7th ACM conference on Mobile Computing and Networking (MobiCom), Rome, Italy, Jul. 2001, pp. 287–297.
[98] K. Winstein, A. Sivaraman, and H. Balakrishnan, “Stochastic forecasts achieve high throughput and low delay over cellular networks,” in 10th USENIX Conference on Networked Systems Design and Implementation (NSDI), Lombard, Illinois, USA, Apr. 2013, pp. 459–472.
[99] K. Tan, J. Song, Q. Zhang, and M. Sridharan, “A Compound TCP approach for high-speed and long distance networks,” in 25th IEEE International Conference on Computer Communications (INFOCOM), Barcelona, Spain, Apr. 2006, pp. 1–12.
[100] S. Liu, T. Ba¸sar, and R. Srikant, “TCP-Illinois: A loss-and delay-based congestion control algorithm for high-speed networks,” Performance Evaluation, vol. 65, no. 6–7, pp. 417–440, Jun. 2008.
[101] C. P. Fu and S. C. Liew, “TCP Veno: TCP enhancement for transmission over wireless access networks,” IEEE Journal on selected areas in communications, vol. 21, no. 2, pp. 216–228, Feb 2003.
[102] K. Winstein and H. Balakrishnan, “TCP Ex Machina: Computergenerated congestion control,” ACM Computer Communication Review, vol. 43, no. 4, pp. 123–134, Aug. 2013.
[103] A. Sivaraman, K. Winstein, P. Thaker, and H. Balakrishnan, “An experimental study of the learnability of congestion control,” ACM Computer Communication Review, vol. 44, no. 4, pp. 479–490, Aug. 2014.

[104] M. Dong, Q. Li, D. Zarchy, P. B. Godfrey, and M. Schapira, “PCC: Re-architecting congestion control for consistent high performance,” in 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI), Oakland, Califonia, USA, May 2015, pp. 395–408.
[105] X. Nie, Y. Zhao, Z. Li, G. Chen, K. Sui, J. Zhang, Z. Ye, and D. Pei, “Dynamic TCP Initial Windows and Congestion Control Schemes Through Reinforcement Learning,” IEEE Journal on Selected Areas in Communications, vol. 37, no. 6, pp. 1231–1247, June 2019.
[106] W. Li, F. Zhou, K. R. Chowdhury, and W. M. Meleis, “QTCP: Adaptive Congestion Control with Reinforcement Learning,” IEEE Transactions on Network Science and Engineering, pp. 1–13, May 2018, Early Access.
[107] V.-H. Tran and O. Bonaventure, “Beyond socket options: making the Linux TCP stack truly extensible,” arXiv preprint arXiv:1901.01863, pp. 1–9, Jan. 2019. [Online]. Available: https://arxiv.org/pdf/1901. 01863.pdf
[108] R. Ando, T. Murase, and M. Oguchi, “Characteristics of QoSguaranteed TCP on real mobile terminal in wireless LAN,” in IEEE International Communications Quality and Reliability Workshop (CQR). Naples, Florida, USA: IEEE, May 2011, pp. 1–6.
[109] P. M. Mohan, D. M. Divakaran, and M. Gurusamy, “Performance study of TCP flows with QoS-supported OpenFlow in data center networks,” in 19th IEEE International Conference on Networks (ICON). Singapore: IEEE, Apr. 2013, pp. 1–6.
[110] J. Nagle, “Congestion control in IP/TCP internetworks,” IETF, RFC 896, Jan. 1984. [Online]. Available: https://rfc-editor.org/rfc/rfc896.txt
[111] L. A. Grieco and S. Mascolo, “Performance evaluation and comparison of Westwood+, New Reno, and Vegas TCP congestion control,” ACM Computer Communication Review, vol. 34, no. 2, pp. 25–38, Apr. 2004.
[112] J. Sing and B. Soh, “TCP New Vegas: Improving the performance of TCP Vegas over high latency links,” in 4th IEEE International Symposium on Network Computing and Application (NCA), Cambridge, Massachusetts, USA, Jul. 2005, pp. 73–82.
[113] S. ullah Lar and X. Liao, “An initiative for a classified bibliography on TCP/IP congestion control,” Journal of Network and Computer Applications, vol. 36, no. 1, pp. 126–133, Jan. 2013.
[114] M. Allman, V. Paxson, and E. Blanton, “TCP congestion control,” IETF, RFC 5681, Sep. 2009. [Online]. Available: https://rfc-editor.org/ rfc/rfc5681.txt
[115] E. Blanton, M. Allman, L. Wang, I. Jarvinen, M. Kojo, and Y. Nishida, “A conservative loss recovery algorithm based on Selective Acknowledgment (SACK) for TCP,” IETF, RFC 6675, Aug. 2012. [Online]. Available: https://rfc-editor.org/rfc/rfc6675.txt
[116] D. Borman, R. T. Braden, V. Jacobson, and R. Scheffenegger, “TCP extensions for high performance,” IETF, RFC 7323, Sep. 2014. [Online]. Available: https://rfc-editor.org/rfc/rfc7323.txt
[117] M. Šošic and V. Stojanovi ´ c, “Resolving poor TCP performance on ´ high-speed long distance links – Overview and comparison of BIC, CUBIC and Hybla,” in 11th IEEE International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, Sep. 2013, pp. 325–330.
[118] M. Ahmad, A. B. Ngadi, A. Nawaz, U. Ahmad, T. Mustafa, and A. Raza, “A survey on TCP CUBIC variant regarding performance,” in 15th International Multitopic Conference (INMIC), Islamabad, Pakistan, Dec. 2012, pp. 409–412.
[119] I. Rhee, L. Xu, S. Ha, A. Zimmermann, L. Eggert, and R. Scheffenegger, “CUBIC for fast long-distance networks,” IETF, draft-ietg-tcpm-cubic-07, Nov. 2017. [Online]. Available: https: //tools.ietf.org/id/draft-ietg-tcpm-cubic-07
[120] S. Floyd and E. Kohler, “Profile for Datagram Congestion Control Protocol (DCCP) Congestion Control ID 2: TCP-like congestion control,” IETF, RFC 4341, Mar. 2006. [Online]. Available: http://www.rfc-editor.org/rfc/rfc4341.txt
[121] S. Floyd, E. Kohler, and J. Padhye, “TCP-Friendly Rate Control (TFRC): Protocol specification,” IETF, RFC 5348, Sep. 2008. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5348.txt

[122] S. Floyd and E. Kohler, “Profile for Datagram Congestion Control Protocol (DCCP) Congestion ID 4: TCP-Friendly Rate Control for Small Packets (TFRC-SP),” IETF, RFC 5622, Aug. 2009. [Online]. Available: http://www.rfc-editor.org/rfc/rfc5622.txt

[123] M. Schier and M. Welzl, “Using DCCP: Issues and improvements,” in 20th IEEE International Conference on Network Protocols (ICNP). Austin, Texas, USA: IEEE, Oct. 2012, pp. 1–9.

[124] A. Kumar, L. Jacob, and A. L. Ananda, “SCTP vs TCP: Performance comparison in MANETs,” in 29th IEEE International Conference on Local Computer Networks (LCN), Tampa, Florida, USA, Nov. 2004, pp. 431–432.

[125] I. Ahmed, O. Yasuo, and K. Masanori, “Improving performance of SCTP over broadband high latency networks,” in 28th IEEE International Conference on Local Computer Networks (LCN), Bonn, Germany, Oct 2003, pp. 644–645.

[126] C. Roseti and E. Kristiansen, “TCP Noordwijk: TCP-based transport optimized for web traffic in satellite networks,” in 26th AIAA International Communications Satellite Systems Conference (ICSSC), San Diego, California, USA, Jun. 2008, pp. 1–12.

[127] L. S. Brakmo and L. L. Peterson, “TCP Vegas: end to end congestion avoidance on a global Internet,” IEEE Journal on Selected Areas in Communications, vol. 13, no. 8, pp. 1465–1480, Oct. 1995.

[128] K. N. Srijith, L. Jacob, and A. L. Ananda, “TCP Vegas-A: solving the fairness and rerouting issues of TCP Vegas,” in IEEE International Performance, Computing, and Communications Conference (IPCC), Phoenix, Arizona, USA, Apr. 2003, pp. 309–316.

[129] W. Zhou, W. Xing, Y. Wang, and J. Zhang, “TCP Vegas-V: Improving the performance of TCP Vegas,” in IEEE International Conference on Automatic Control and Artificial Intelligence (ACAI), Xiamen, China, Mar. 2012, pp. 2034–2039.

[130] Y. Guo, X. Yang, R. Wang, and J. Sun, “TCP Adaptive Vegas: Improving of TCP Vegas algorithm,” in IEEE International Conference on Information Science, Electronics and Electrical Engineering (ISEEE), vol. 1, Sapporo, Japan, Apr. 2014, pp. 126–130.

[131] L. Ding, X. Wang, Y. Xu, W. Zhang, and W. Chen, “Vegas-W: An Enhanced TCP-Vegas for Wireless Ad Hoc Networks,” in IEEE International Conference on Communications (ICC), Beijing, China, May 2008, pp. 2383–2387.

[132] D. Kim, H. Bae, and C. K. Toh, “Improving TCP-Vegas performance over MANET routing protocols,” IEEE Transactions on Vehicular Technology, vol. 56, no. 1, pp. 372–377, Jan. 2007.

[133] J. Sing and B. Soh, “TCP New Vegas revisited,” in 13th IEEE International Conference on Networks, vol. 2, Nov. 2005.

[134] Y.-M. Liu and X.-H. Jiang, “An extended DCCP congestion control in Wireless Sensor Networks,” in IEEE International Workshop on Intelligent Systems and Applications (ISA), Wuhan, China, May 2009, pp. 1–4.

[135] L. Ye and Z. Wang, “A QoS-aware congestion control mechanism for DCCP,” in IEEE Symposium on Computers and Communications (ISCC), Sousse, Tunisia, Jul. 2009, pp. 624–629.

[136] G. Hasegawa, K. Kurata, and M. Murata, “Analysis and improvement of fairness between TCP Reno and Vegas for deployment of TCP Vegas to the Internet,” in 8th IEEE International Conference on Network Protocols (ICNP), Osaka, Japan, Nov. 2000, pp. 177–186.

[137] A. Aijaz, M. Dohler, A. H. Aghvami, V. Friderikos, and M. Frodigh, “Realizing the tactile internet: Haptic communications over next generation 5G cellular networks,” IEEE Wireless Communications, vol. 24, no. 2, pp. 82–89, April 2017.

[138] V. Arun and H. Balakrishnan, “Copa: Practical delay-based congestion control for the Internet,” in 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI). Renton, Washington, USA: USENIX Association, Apr. 2018, pp. 329–342.

[139] R. Mittal, V. T. Lam, N. Dukkipati, E. Blem, H. Wassel, M. Ghobadi, A. Vahdat, Y. Wang, D. Wetherall, and D. Zats, “TIMELY: RTT-based congestion control for the datacenter,” ACM Computer Communication Review, vol. 45, no. 4, pp. 537–550, Aug. 2015.

[140] A. Zanella, G. Procissi, M. Gerla, and M. Sanadidi, “TCP Westwood: Analytic model and performance evaluation,” in IEEE Global Telecommunications Conference (GLOBECOM), vol. 3. San Antonio, Texas, USA: IEEE, 2001, pp. 1703–1707.
[141] S. Mascolo, L. A. Grieco, R. Ferorelli, P. Camarda, and G. Piscitelli, “Performance evaluation of Westwood+ TCP congestion control,” Performance Evaluation, vol. 55, no. 1-2, pp. 93–111, Jan. 2004.
[142] L. Kleinrock, “Power and deterministic rules of thumb for probabilistic problems in computer communications,” in IEEE International Conference on Communications (ICC), Boston, Massachusetts, USA, Jun. 1979, pp. 43.1.1–43.1.10.
[143] J. Jaffe, “Flow control power is nondecentralizable,” IEEE Transactions on Communications, vol. 29, no. 9, pp. 1301–1306, Sep. 1981.
[144] M. Hock, R. Bless, and M. Zitterbart, “Experimental evaluation of BBR congestion control,” in 25th IEEE International Conference on Network Protocols (ICNP), Toronto, Ontario, Canada, Oct. 2017, pp. 1–10.
[145] K. Miyazawa, K. Sasaki, N. Oda, and S. Yamaguchi, “Cyclic performance fluctuation of TCP BBR,” in 42nd IEEE Annual Computer Software and Applications Conference (COMPSAC), vol. 1, Tokyo, Japan, Jul. 2018, pp. 811–812.
[146] P. Farrow, “Performance analysis of heterogeneous TCP congestion control environments,” in IEEE/IFIP International Conference on Performance Evaluation and Modeling in Wired and Wireless Networks (PEMWN), Paris, France, Nov. 2017, pp. 1–6.
[147] Z. Zhong, I. Hamchaoui, R. Khatoun, and A. Serhrouchni, “Performance evaluation of CQIC and TCP BBR in mobile network,” in 21st IEEE Conference on Innovation in Clouds, Internet and Networks and Workshops (ICIN), Paris, France, Feb. 2018, pp. 1–5.
[148] E. Atxutegi, F. Liberal, H. K. Haile, K. Grinnemo, A. Brunstrom, and A. Arvidsson, “On the use of TCP BBR in cellular networks,” IEEE Communications Magazine, vol. 56, no. 3, pp. 172–179, Mar. 2018.
[149] F. Li, J. W. Chung, X. Jiang, and M. Claypool, “TCP CUBIC versus BBR on the Highway,” in International Conference on Passive and Active Network Measurement (PAM). Cleveland, Ohio, USA: Springer, 2018, pp. 269–280.
[150] N. Cardwell, Y. Cheng, C. S. Gunn, S. H. Yeganeh, and V. Jacobson, “A quick update on BBR in shallow buffers,” in IETF 100, Singapore, Nov. 2017. [Online]. Available: https://datatracker.ietf.org/meeting/100/materials/slides-100-iccrg-a-quick-bbr-update-bbr-in-shallow-buffers
[151] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, “Machine learning for networking: Workflow, advances and opportunities,” IEEE Network, vol. 32, no. 2, pp. 92–99, Mar. 2018.
[152] M. Dong, T. Meng, D. Zarchy, E. Arslan, Y. Gilad, B. Godfrey, and M. Schapira, “PCC Vivace: Online-learning congestion control,” in 15th USENIX Symposium on Networked Systems Design and Implementation (NSDI), Renton, Washington, USA, Apr. 2018, pp. 343–356.
[153] Y. Kong, H. Zang, and X. Ma, “Improving TCP congestion control with machine intelligence,” in ACM SIGCOMM Workshop on Network Meets AI&ML (NetAI). Budapest, Hungary: ACM, Aug. 2018, pp. 60–66.
[154] W. Li, F. Zhou, W. Meleis, and K. Chowdhury, “Learning-based and data-driven TCP design for memory-constrained IoT,” in IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS), Washington, DC, USA, May 2016, pp. 199–205.
[155] A. P. Silva, K. Obraczka, S. Burleigh, and C. M. Hirata, “Smart congestion control for delay- and disruption tolerant networks,” in 13th IEEE International Conference on Sensing, Communication, and Networking (SECON), London, UK, Jun. 2016, pp. 1–9.
[156] M. Schapira and K. Winstein, “Congestion-Control Throwdown,” in 16th ACM Workshop on Hot Topics in Networks (HotNets), Palo Alto, California, USA, Nov. 2017, pp. 122–128.
[157] M. Zhang, M. Mezzavilla, J. Zhu, S. Rangan, and S. Panwar, “TCP dynamics over mmWave links,” in 18th IEEE International Workshop on Signal Processing Advances in Wireless Communications (SPAWC), Sapporo, Japan, July 2017, pp. 1–6.

[158] T. Azzino, M. Drago, M. Polese, A. Zanella, and M. Zorzi, “X-TCP: A cross layer approach for TCP uplink flows in mmWave networks,” in 16th Annual Mediterranean Ad Hoc Networking Workshop (MedHoc-Net), Budva, Montenegro, Jun. 2017, pp. 1–6.
[159] M. Polese, M. Mezzavilla, M. Zhang, J. Zhu, S. Rangan, S. Panwar, and M. Zorzi, “milliProxy: A TCP proxy architecture for 5G mmWave cellular systems,” in 51st Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, California, USA, Oct. 2017, pp. 951–957.
[160] A. T. Naman, Y. Wang, H. H. Gharakheili, V. Sivaraman, and D. Taubman, “Responsive high throughput congestion control for interactive applications over SDN-enabled networks,” Computer Networks, vol. 134, pp. 152–166, Apr. 2018.
[161] H. Shiang and M. van der Schaar, “A quality-centric TCP-friendly congestion control for multimedia transmission,” IEEE Transactions on Multimedia, vol. 14, no. 3, pp. 896–909, Jun. 2012.
[162] O. Habachi, H.-P. Shiang, M. van der Schaar, and Y. Hayel, “Online learning based congestion control for adaptive multimedia transmission,” IEEE Transactions on Signal Processing, vol. 61, no. 6, pp. 1460–1469, Mar. 2013.
[163] S. M. Aghdam, M. Khansari, H. R. Rabiee, and M. Salehi, “WCCP: A congestion control protocol for wireless multimedia communication in sensor networks,” Ad Hoc Networks, vol. 13, pp. 516–534, February 2014.
[164] F. Lu, H. Du, A. Jain, G. M. Voelker, A. C. Snoeren, and A. Terzis, “CQIC: Revisiting cross-layer congestion control for cellular networks,” in 16th ACM International Workshop on Mobile Computing Systems and Applications (HotMobile). Santa Fe, New Mexico, USA: ACM, Feb. 2015, pp. 45–50.
[165] C. Wilson, H. Ballani, T. Karagiannis, and A. Rowtron, “Better never than late: Meeting deadlines in datacenter networks,” in ACM Computer Communication Review, vol. 41, no. 4, Aug. 2011, pp. 50–61.
[166] D. Drutskoy, E. Keller, and J. Rexford, “Scalable network virtualization in Software-Defined Networks,” IEEE Internet Computing, vol. 17, no. 2, pp. 20–27, Mar. 2013.
[167] B. Vamanan, J. Hasan, and T. Vijaykumar, “Deadline-aware Datacenter TCP (D2TCP),” ACM Computer Communication Review, vol. 42, no. 4, pp. 115–126, Sep. 2012.
[168] M. Alizadeh, S. Yang, M. Sharif, S. Katti, N. McKeown, B. Prabhakar, and S. Shenker, “pFabric: Minimal near-optimal datacenter transport,” in ACM Computer Communication Review, vol. 43, no. 4. ACM, Aug. 2013, pp. 435–446.
[169] P. X. Gao, A. Narayan, G. Kumar, R. Agarwal, S. Ratnasamy, and S. Shenker, “pHost: Distributed near-optimal datacenter transport over commodity network fabric,” in 11th ACM Conference on Emerging Networking Experiments and Technologies (CoNEXT). Heidelberg, Germany: ACM, Dec. 2015, pp. 1–12.
[170] I. Cho, K. Jang, and D. Han, “Credit-scheduled delay-bounded congestion control for datacenters,” in ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM). Los Angeles, California, USA: ACM, Aug. 2017, pp. 239–252.
[171] B. Montazeri, Y. Li, M. Alizadeh, and J. Ousterhout, “Homa: A receiver-driven low-latency transport protocol using network priorities,” in ACM Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM). Budapest, Hungary: ACM, Aug. 2018, pp. 221–235.
[172] A. Munir, G. Baig, S. M. Irteza, I. A. Qazi, A. X. Liu, and F. R. Dogar, “Friends, not foes: synthesizing existing transport strategies for data center networks,” ACM Computer Communication Review, vol. 44, no. 4, pp. 491–502, Feb. 2015.
[173] M. Noormohammadpour and C. S. Raghavendra, “Datacenter traffic control: Understanding techniques and tradeoffs,” IEEE Communications Surveys and Tutorials, vol. 20, no. 2, pp. 1492–1525, Dec. 2017.
[174] J. Zhang, F. R. Yu, S. Wang, T. Huang, Z. Liu, and Y. Liu, “Load balancing in data center networks: A survey,” IEEE Communications Surveys and Tutorials, vol. 20, no. 3, pp. 2324–2352, Jul. 2018.
[175] T. Hafeez, N. Ahmed, B. Ahmed, and A. W. Malik, “Detection and mitigation of congestion in SDN enabled data center networks: A survey,” IEEE Access, vol. 6, pp. 1730–1740, 2018.
[176] A. Ford, C. Raiciu, M. Handley, O. Bonaventure, and C. Paasch, “TCP extensions for multipath operation with multiple addresses,” IETF, RFC 6824, Jan. 2013. [Online]. Available: https://rfc-editor.org/ rfc/rfc6824.txt

[177] O. Mehani, R. Holz, S. Ferlin, and R. Boreli, “An early look at Multipath TCP deployment in the wild,” in 6th ACM International Workshop on Hot Topics in Planet-Scale Measurement (HotPlanet), Paris, France, 2015, pp. 7–12.
[178] C. Raiciu, C. Paasch, S. Barre, A. Ford, M. Honda, F. Duchene, O. Bonaventure, and M. Handley, “How hard can it be? designing and implementing a deployable Multipath TCP,” in 9th USENIX Conference on Networked Systems Design and Implementation (NSDI), Apr. 2012, pp. 29–29.
[179] M. Scharf and A. Ford, “Multipath TCP (MPTCP) application interface considerations,” IETF, RFC 6897, Mar. 2013. [Online]. Available: https://rfc-editor.org/rfc/rfc6897.txt
[180] M. Bagnulo, “Threat analysis for TCP extensions for multipath operation with multiple addresses,” IETF, RFC 6181, Mar. 2011. [Online]. Available: https://rfc-editor.org/rfc/rfc6181.txt
[181] S. Barré, C. Paasch, and O. Bonaventure, “Multipath TCP: From theory to practice,” in IEEE/IFIP Networking Conference (IFIP Networking), Valencia, Spain, May 2011, pp. 444–457.
[182] O. Bonaventure, C. Paasch, and G. Detal, “Use cases and operational experience with Multipath TCP,” IETF, RFC 8041, Jan. 2017. [Online]. Available: https://rfc-editor.org/rfc/rfc8041.txt
[183] B. Han, F. Qian, and L. Ji, “When should we surf the mobile web using both WiFi and cellular?” in 5th ACM SIGCOMM Workshop on All Things Cellular: Operations, Applications and Challenges (AllThingsCellular). London, UK: ACM, 2016, pp. 7–12.
[184] M. Li, A. Lukyanenko, and Y. Cui, “Network coding based multipath TCP,” in IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Orlando, Florida, USA, Mar. 2012, pp. 25–30.
[185] K. Nguyen, G. P. Villardi, M. G. Kibria, K. Ishizu, F. Kojima, and H. Shinbo, “An enhancement of Multipath TCP performance in lossy wireless networks,” in 41st IEEE Conference on Local Computer Networks Workshops (LCN Workshops), Dubai, UAE, Nov. 2016, pp. 187–191.
[186] M. Becke, T. Dreibholz, H. Adhari, and E. P. Rathgeb, “On the fairness of transport protocols in a multi-path environment,” in IEEE International Conference on Communications (ICC), Ottawa, Ontario, Canada, Jun. 2012, pp. 2666–2672.
[187] H. Han, S. Shakkottai, C. V. Hollot, R. Srikant, and D. Towsley, “Multipath TCP: a joint congestion control and routing scheme to exploit path diversity in the internet,” IEEE/ACM Transactions on Networking, vol. 14, no. 6, pp. 1260–1271, Dec. 2006.
[188] D. Wischik, C. Raiciu, A. Greenhalgh, and M. Handley, “Design, implementation and evaluation of congestion control for Multipath TCP,” in 8th USENIX Conference on Networked Systems Design and Implementation (NSDI), Boston, Massachusetts, USA, 2011, pp. 99– 112.
[189] R. Khalili, N. Gast, M. Popovic, and J.-Y. Le Boudec, “MPTCP is not Pareto-optimal: Performance issues and a possible solution,” IEEE/ACM Transantions on Networking, vol. 21, no. 5, pp. 1651–1665, Oct. 2013.
[190] A. Walid, J. Hwang, Q. Peng, and S. Low, “Balanced Linked Adaptation congestion control algorithm for MPTCP,” IETF, Working Draft draft-walid-mptcp-congestion-control-00, Jul. 2014. [Online]. Available: http://www.ietf.org/internet-drafts/ draft-walid-mptcp-congestion-control-00.txt
[191] M. Zhu, L. Wang, Z. Qin, N. Ding, J. Fang, T. Liu, and Q. Cui, “BELIA: bandwidth estimate-based link increase algorithm for MPTCP,” IET Networks, vol. 6, no. 5, pp. 94–101, September 2017.
[192] Z. Xu, J. Tang, C. Yin, Y. Wang, and G. Xue, “Experience-Driven Congestion Control: When Multi-Path TCP Meets Deep Reinforcement Learning,” IEEE Journal on Selected Areas in Communications, vol. 37, no. 6, pp. 1325–1336, June 2019.
[193] J. R. Iyengar, P. D. Amer, and R. Stewart, “Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths,” IEEE/ACM Transactions on Networking, vol. 14, no. 5, pp. 951–964, Oct. 2006.
[194] M. Becke, H. Adhari, E. P. Rathgeb, F. Fa, X. Yang, and X. Zhou, “Comparison of Multipath TCP and CMT-SCTP based on intercontinental measurements,” in IEEE Global Communications Conference (GLOBECOM). Atlanta, Georgia, USA: IEEE, 2013, pp. 1360–1366.
[195] C.-M. Huang, Y.-C. Chen, and S.-Y. Lin, “Packet scheduling and congestion control schemes for Multipath Datagram Congestion Control Protocol,” The Computer Journal, vol. 58, no. 2, pp. 188–203, Feb. 2015.

[196] B. Y. L. Kimura, D. C. S. F. Lima, and A. A. F. Loureiro, “Alternative scheduling decisions for Multipath TCP,” IEEE Communications Letters, vol. 21, no. 11, pp. 2412–2415, Nov. 2017.
[197] D. Ni, K. Xue, P. Hong, and S. Shen, “Fine-grained forward prediction based dynamic packet scheduling mechanism for multipath TCP in lossy networks,” in 23rd IEEE International Conference on Computer Communication and Networks (ICCCN), Shanghai, China, 2014, pp. 1–7.
[198] K. W. Choi, Y. S. Cho, J. W. Lee, S. M. Cho, J. Choi et al., “Optimal load balancing scheduler for MPTCP-based bandwidth aggregation in heterogeneous wireless environments,” Computer Communications, vol. 112, pp. 116–130, Nov. 2017.
[199] Y.-s. Lim, E. M. Nahum, D. Towsley, and R. J. Gibbens, “ECF: An MPTCP Path Scheduler to Manage Heterogeneous Paths,” in 13th ACM International Conference on Emerging Networking EXperiments and Technologies (CoNEXT), Incheon, South Korea, Dec. 2017, pp. 147– 159.
[200] Y. E. Guo, A. Nikravesh, Z. M. Mao, F. Qian, and S. Sen, “Demo: DEMS: DEcoupled Multipath Scheduler for accelerating multipath transport,” in 23rd ACM International Conference on Mobile Computing and Networking (MobiCom), Snowbird, Utah, USA, Oct. 2017, pp. 477–479.
[201] H. Shi, Y. Cui, X. Wang, Y. Hu, M. Dai, F. Wang, and K. Zheng, “STMS: Improving MPTCP throughput under Heterogeneous Networks,” in USENIX Annual Technical Conference (USENIX ATC), Boston, Massachusetts, USA, Jul. 2018, pp. 719–730.
[202] N. Kuhn, E. Lochin, A. Mifdaoui, G. Sarwar, O. Mehani, and R. Boreli, “DAPS: Intelligent delay-aware packet scheduling for multipath transport,” in IEEE International Conference on Communications (ICC). IEEE, 2014, pp. 1222–1227.
[203] S. Ferlin, O. Alay, O. Mehani, and R. Boreli, “BLEST: Blocking estimation-based MPTCP scheduler for heterogeneous networks,” in IEEE/IFIP Networking Conference (IFIP Networking), Vienna, Austria, May 2016, pp. 431–439.
[204] K. Fahmi, D. Leith, S. Kucera, and H. Claussen, “Low Delay Scheduling of Objects Over Multiple Wireless Paths,” arXiv preprint arXiv:1808.02418, Aug. 2018. [Online]. Available: https: //arxiv.org/abs/1808.02418
[205] T. Shreedhar, N. Mohan, S. Kaul, J. Kangasharju et al., “QAware: a cross-layer approach to MPTCP scheduling,” in IEEE/IFIP Networking Conference (IFIP Networking), Zürich, Switzerland, May 2018.
[206] M. Xu, Y. Cao, and E. Dong, “Delay-based congestion control for MPTCP,” IETF, Working Draft: draft-xu-mptcp-congestion-control-05, Jan. 2017. [Online]. Available: https://datatracker.ietf.org/doc/html/ draft-xu-mptcp-congestion-control-05
[207] T.-A. Le, “Improving the performance of multipath congestion control over wireless networks,” in International Conference on Advanced Technologies for Communications (ATC). Vietri sul Mare, Italy: IEEE, Dec. 2013, pp. 60–65.
[208] T. Shreedhar, N. Mohan, S. K. Kaul, and J. Kangasharju, “More than the sum of its parts: Exploiting cross-layer and joint-flow information in MPTCP,” arXiv preprint arXiv:1711.07565, Nov. 2017. [Online]. Available: https://arxiv.org/abs/1711.07565
[209] K. Noda, Y. Ito, and Y. Muraki, “Study on congestion control of multipath TCP based on web-QoE under heterogeneous environment,” in 6th IEEE Global Conference on Consumer Electronics (GCCE), Nagoya, Japan, Oct. 2017, pp. 1–3.
[210] Y. Muraki and Y. Ito, “Study on effect of congestion control of multipath TCP on Web-QoE,” in 4th IEEE Global Conference on Consumer Electronics (GCCE), Osaka, Japan, Oct. 2015, pp. 52–53.
[211] Y.-C. Chen, Y.-s. Lim, R. J. Gibbens, E. M. Nahum, R. Khalili, and D. Towsley, “A measurement-based study of Multipath TCP performance over wireless networks,” in ACM Conference on Internet Measurement, 2013, pp. 455–468.
[212] M. Li, A. Lukyanenko, S. Tarkoma, Y. Cui, and A. Ylä-Jääski, “Tolerating path heterogeneity in Multipath TCP with bounded receive buffers,” in ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS), Pittsburgh, PA, USA, Jun. 2013, pp. 375–376.
[213] A. Garcia-Saavedra, M. Karzand, and D. J. Leith, “Low Delay Random Linear Coding and Scheduling Over Multiple Interfaces,” IEEE Transactions on Mobile Computing, vol. 16, no. 11, pp. 3100–3114, Nov. 2017.

[214] P. Ageneau, N. Boukhatem, and M. Gerla, “Practical random linear coding for MultiPath TCP: MPC-TCP,” in 24th IEEE International Conference on Telecommunications (ICT), Limassol, Cyprus, May 2017, pp. 1–6.
[215] Y. Cui, L. Wang, X. Wang, H. Wang, and Y. Wang, “FMTCP: A fountain code-based Multipath Transmission Control Protocol,” IEEE/ACM Transactions on Networking, vol. 23, no. 2, pp. 465–478, Apr. 2015.
[216] S. Ferlin, S. Kucera, H. Claussen, and O. Alay, “MPTCP meets FEC: Supporting latency-sensitive applications over Heterogeneous Networks,” IEEE/ACM Transactions on Networking, pp. 1–14, Oct. 2018.
[217] S. Ferlin, T. Dreibholz, and Ö. Alay, “Multi-path transport over heterogeneous wireless networks: Does it really pay off?” in IEEE Global Communications Conference (GLOBECOM). Austin, Texas, USA: IEEE, Dec. 2014, pp. 4807–4813.
[218] M. S. Kim, I. H. Jung, K. M. Han, J. Y. Lee, and B. C. Kim, “Performance enhancement of MPTCP having a bufferbloat path using retransmission of HoL blocking packets,” in IEEE International Conference on Electronics, Information, and Communication (ICEIC), Honolulu, Hawaii, USA, Jan. 2018, pp. 1–2.
[219] J. Hwang and J. Yoo, “Packet scheduling for Multipath TCP,” in 7th International Conference on Ubiquitous and Future Networks, Jul. 2015, pp. 177–179.
[220] E. Dong, M. Xu, X. Fu, and Y. Cao, “LAMPS: A Loss Aware Scheduler for Multipath TCP over Highly Lossy Networks,” in 42nd IEEE Conference on Local Computer Networks (LCN), Singapore, Oct. 2017, pp. 1–9.
[221] C. Paasch, S. Ferlin, O. Alay, and O. Bonaventure, “Experimental evaluation of multipath TCP schedulers,” in ACM SIGCOMM Workshop on Capacity Sharing (CSWS). Chicago, Illinois, USA: ACM, Aug. 2014, pp. 27–32.
[222] C. Raiciu, C. Pluntke, S. Barre, A. Greenhalgh, D. Wischik, and M. Handley, “Data center networking with multipath TCP,” in 9th ACM Workshop on Hot Topics in Networks (HotNets), Monterey, California, USA, Oct. 2010, p. 10.
[223] S. Tariq and M. Bassiouni, “Performance evaluation of MPTCP over optical burst switching in data centers,” in IEEE International Telecommunications Symposium (ITS), São Paulo, Brazil, Aug. 2014, pp. 1–5.
[224] M. Li, A. Lukyanenko, S. Tarkoma, and A. Ylä-Jääski, “MPTCP incast in data center networks,” China Communications, vol. 11, no. 4, pp. 25–37, Apr. 2014.
[225] J. Hwang, A. Walid, and J. Yoo, “Fast coupled retransmission for multipath TCP in data center networks,” IEEE Systems Journal, vol. 12, no. 1, pp. 1056–1059, Mar. 2018.
[226] S. Zannettou, M. Sirivianos, and F. Papadopoulos, “Exploiting path diversity in datacenters using MPTCP-aware SDN,” in 21st IEEE Symposium on Computers and Communication (ISCC), Messina, Italy, Jun. 2016, pp. 539–546.
[227] J. Duan, Z. Wang, and C. Wu, “Responsive multipath TCP in SDNbased datacenters,” in IEEE International Conference on Communications (ICC), London, UK, Jun. 2015, pp. 5296–5301.
[228] T. Dreibholz, M. Becke, H. Adhari, and E. P. Rathgeb, “On the impact of congestion control for concurrent multipath transfer on the transport layer,” in 11th IEEE International Conference on Telecommunications (ConTEL), Graz, Austria, Jun. 2011, pp. 397–404.
[229] Q. De Coninck and O. Bonaventure, “Multipath QUIC: Design and evaluation,” in 13th ACM International Conference on Emerging Networking EXperiments and Technologies (CoNEXT), Incheon, South Korea, Dec. 2017, pp. 160–166.
[230] M. Amend, A. Brunstrom, A. Kassler, and V. Rakocevic, “IP compatible multipath framework for heterogeneous access networks,” IETF, Working Draft draft-amend-tsvwg-multipath-framework-mpdccp-00, Mar. 2019. [Online]. Available: http://www.ietf.org/internet-drafts/ draft-amend-tsvwg-multipath-framework-mpdccp-00.txt