

دانشكده مهندسي کامپیوتر

پخش ویدیو از طریق پروتکل QUIC

پايان‌نامه براي دريافت درجه کارشناسی

در رشته مهندسي کامپیوتر

سیدحسام حسینی

استاد راهنما:

دكتر وصال حکمی

مهرماه 1404



تأييديه‌ي هيأت داوران جلسه‌ي دفاع از پايان‌نامه

نام دانشكده: مهندسی کامپیوتر

نام دانشجو: سیدحسام حسینی

عنوان پايان‌نامه: پخش ویدیو از طریق پروتکل QUIC

تاريخ دفاع:

رشته: مهندسی کامپیوتر

گرايش: -

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **رديف** | **سمت** | **نام و نام خانوادگي** | **مرتبه دانشگاهي** | **دانشگاه يا مؤسسه** | **امضا** |
| 1 | استاد راهنما | وصال حکمی | استادیار | دانشگاه علم و صنعت ایران |  |
| 2 | استاد راهنما |  |  |  |  |
| 3 | استاد مشاور |  |  |  |  |
| 4 | استاد مشاور |  |  |  |  |
| 5 | استاد مدعو خارجي |  |  |  |  |
| 6 | استاد مدعو خارجي |  |  |  |  |
| 7 | استاد مدعو داخلي |  |  |  |  |
| 8 | استاد مدعو داخلي |  |  |  |  |

تأييديه‌ي صحت و اصالت نتايج

**باسمه تعالي**

اينجانب سیدحسام حسینی به شماره دانشجويي 98521144 دانشجوي رشته مهندسی کامپیوتر مقطع تحصيلي کارشناسی تأييد مي‌نمايم كه كليه‌ي نتايج اين پايان‌نامه حاصل كار اينجانب و بدون هرگونه دخل و تصرف است و موارد نسخه‌برداري‌شده از آثار ديگران را با ذكر كامل مشخصات منبع ذكر كرده‌ام. درصورت اثبات خلاف مندرجات فوق، به تشخيص دانشگاه مطابق با ضوابط و مقررات حاكم (قانون حمايت از حقوق مؤلفان و مصنفان و قانون ترجمه و تكثير كتب و نشريات و آثار صوتي، ضوابط و مقررات آموزشي، پژوهشي و انضباطي ...) با اينجانب رفتار خواهد شد و حق هرگونه اعتراض درخصوص احقاق حقوق مكتسب و تشخيص و تعيين تخلف و مجازات را از خويش سلب مي‌نمايم. در ضمن، مسؤوليت هرگونه پاسخگويي به اشخاص اعم از حقيقي و حقوقي و مراجع ذي‌صلاح (اعم از اداري و قضايي) به عهده‌ي اينجانب خواهد بود و دانشگاه هيچ‌گونه مسؤوليتي در اين خصوص نخواهد داشت.

نام و نام خانوادگي:

امضا و تاريخ:

مجوز بهره‌برداري از پايان‌نامه

بهره‌برداري از اين پايان‌نامه در چهارچوب مقررات كتابخانه و با توجه به محدوديتي كه توسط استاد راهنما به شرح زير تعيين مي‌شود، بلامانع است:

🞎 بهره‌برداري از اين پايان‌نامه/ رساله براي همگان بلامانع است.

🞎 بهره‌برداري از اين پايان‌نامه/ رساله با اخذ مجوز از استاد راهنما، بلامانع است.

🞎 بهره‌برداري از اين پايان‌نامه/ رساله تا تاريخ .................................... ممنوع است.

نام استاد يا اساتيد راهنما:

تاريخ:

امضا:

تقديم به:

تقدیم به همسرم، همراه روزهای پرالتهاب و شب‌های بی‌قرارم، و به پدر و مادرم، که ریشه‌های عشق و استقامتم از آنان روییده است.

این ثمره‌ی سال‌ها تلاش، از آنِ شماست.

تشكر و قدرداني:

بر خود لازم می‌دانم از استاد گران‌قدر، دکتر وصال حکمی، که با راهنمایی‌های ارزنده، دلسوزی و حمایت‌های بی‌دریغ خود در تمام مراحل این پژوهش، یار و راهنمای من بودند، صمیمانه تشکر و قدردانی نمایم.

بدون راهنمایی‌های ارزشمند ایشان، انجام این تحقیق ممکن نبود.چکيده

با رشد روزافزون سرویس‌های پخش ویدیو در بستر اینترنت، نیاز به پروتکل‌هایی با کارایی بالا، تأخیر پایین و تحمل‌پذیری در برابر نوسانات شبکه بیش از پیش احساس می‌شود. در این پروژه، پروتکل QUIC به عنوان یکی از پروتکل‌های نوین انتقال داده که بر پایه UDP توسعه یافته است، پیاده‌سازی و مورد ارزیابی عملکردی قرار گرفته است.

هدف اصلی این پژوهش، تحلیل و ارزیابی کارایی پروتکل QUIC در فرآیند پخش ویدیو در شرایط مختلف شبکه، شامل تأخیر، نوسان و از دست رفتن بسته‌ها می‌باشد. برای درک بهتر رفتار QUIC، از پروتکل DASH به عنوان مبنای مقایسه استفاده شده است.

پیاده‌سازی و آزمایش‌ها با استفاده از شبیه‌ساز Mininet در محیط لینوکس و به کمک زبان Python انجام شده‌اند. نتایج به‌دست‌آمده نشان می‌دهد که QUIC با کاهش تأخیر شروع پخش و بهبود نرخ انتقال داده، عملکرد مطلوبی در پخش ویدیوی بلادرنگ از خود نشان می‌دهد. این نتایج حاکی از آن است که QUIC می‌تواند گزینه‌ای مناسب برای بهبود تجربه کاربر در سیستم‌های پخش ویدیوی آینده باشد.

واژه‌هاي كليدي:پخش ویدیو، QUIC، UDP، DASH، Mininet

**فهرست مطالب**

[فصل 1: مقدمه 1](#_Toc211882007)

[1-1- مقدمه 2](#_Toc211882008)

[فصل 2: مروري بر منابع 3](#_Toc211882009)

[2-1- مقدمه 4](#_Toc211882010)

[2-2- مروری بر پروتکل های انتقال 5](#_Toc211882011)

[2-2-1- HTTP/1.0 & HTTP/1.1 5](#_Toc211882012)

[2-2-2- HTTP/2 6](#_Toc211882013)

[2-2-3- HTTP/3 7](#_Toc211882014)

[2-3- مروري بر ادبيات موضوع 8](#_Toc211882015)

[2-4- نتيجه‌گيري 8](#_Toc211882016)

[فصل 3: روش تحقيق 10](#_Toc211882017)

[3-1- مقدمه 11](#_Toc211882018)

[3-2- محتوا 11](#_Toc211882019)

[3-2-1- علت انتخاب روش 11](#_Toc211882020)

[3-2-2- تشريح كامل روش تحقيق 12](#_Toc211882021)

[3-2-3- سناریو های مورد تحقیق 16](#_Toc211882022)

[فصل 4: نتايج و تفسير آنها 18](#_Toc211882023)

[4-1- مقدمه 19](#_Toc211882024)

[4-2- سناریوهای آزمایش پروتکل‌ها: 19](#_Toc211882025)

[4-2-2- سناریو اول(شبکه LAN ایده آل) 20](#_Toc211882026)

[4-2-3- سناریو دوم(WiFi مناسب) 25](#_Toc211882027)

[4-2-4- سناریو سوم(WiFi شلوغ در فاصله دور) 30](#_Toc211882028)

[4-2-5- سناریو چهارم(موبایل 4G) 35](#_Toc211882029)

[4-2-6- سناریو پنجم(موبایل 4G شلوغ) 40](#_Toc211882030)

[4-2-7- سناریو ششم(موبایل 3G نسل قدیمی) 45](#_Toc211882031)

[4-2-8- سناریو هفتم(شبکه ماهواره ایی) 50](#_Toc211882032)

[4-2-9- سناریو هشتم(WiFi شلوغ) 55](#_Toc211882033)

[4-2-10- سناریو نهم(WiFi شلوغ ناپایدار) 60](#_Toc211882034)

[4-2-11- سناریو دهم(شبکه معیوب) 65](#_Toc211882035)

[4-3- بررسی تجمیعی سناریوها برای معیار تاخیر راه‌اندازی(startup delay) 70](#_Toc211882036)

[4-3-1- تفسیر عمومی نمودار CDF برای Startup-Delay 70](#_Toc211882037)

[4-3-2- تفسیر نتایج داخل نمودار 72](#_Toc211882038)

[فصل 5: جمع‌بندي و پيشنهادها 73](#_Toc211882039)

[5-1- مقدمه 74](#_Toc211882040)

[5-2- نتایج و پیشنهاد 74](#_Toc211882041)

[5-2-1- جمع‌بندي 74](#_Toc211882042)

[5-2-2- نوآوري 76](#_Toc211882043)

[5-2-3- پيشنهادها 77](#_Toc211882044)

[مراجع 79](#_Toc211882045)

[پيوست‌ها 81](#_Toc211882046)

فهرست اشکال

[شکل (3-1) توپولوژی شبکه 15](#_Toc211881360)

[شکل (4-1) نمای کلی سناریوهای مورد آزمایش 20](#_Toc211881361)

[شکل (4-2) نمودار میزان Bitrate بر زمان برای سناریو اول 21](#_Toc211881362)

[شکل (4-3) نمودار میزان Buffer-Level بر زمان برای سناریو اول 22](#_Toc211881363)

[شکل (4-4) نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو اول 23](#_Toc211881364)

[شکل (4-5) نمودار میزان Stall-Timeline بر زمان برای سناریو اول 24](#_Toc211881365)

[شکل (4-6) نمودار میزان Throughput بر زمان برای سناریو اول 25](#_Toc211881366)

[شکل (4-7) نمودار میزان Bitrate بر زمان برای سناریو دوم 26](#_Toc211881367)

[شکل (4-8) نمودار میزان Buffer-Level بر زمان برای سناریو دوم 27](#_Toc211881368)

[شکل (4-9) نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو دوم 28](#_Toc211881369)

[شکل (4-10) نمودار میزان Stall-Timeline بر زمان برای سناریو دوم 29](#_Toc211881370)

[شکل (4-11) نمودار میزان Throughput بر زمان برای سناریو دوم 30](#_Toc211881371)

[شکل (4-12) نمودار میزان Bitrate بر زمان برای سناریو سوم 31](#_Toc211881372)

[شکل (4-13) نمودار میزان Buffer-Level بر زمان برای سناریو سوم 32](#_Toc211881373)

[شکل (4-14) نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو سوم 33](#_Toc211881374)

[شکل (4-15) نمودار میزان Stall-Timeline بر زمان برای سناریو سوم 34](#_Toc211881375)

[شکل (4-16) نمودار میزان Throughput بر زمان برای سناریو سوم 35](#_Toc211881376)

[شکل (4-17) نمودار میزان Bitrate بر زمان برای سناریو چهارم 36](#_Toc211881377)

[شکل (4-18) نمودار میزان Buffer-Level بر زمان برای سناریو چهارم 37](#_Toc211881378)

[شکل (4-19) نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو چهارم 38](#_Toc211881379)

[شکل (4-20) نمودار میزان Stall-Timeline بر زمان برای سناریو چهارم 39](#_Toc211881380)

[شکل (4-21) نمودار میزان Throughput بر زمان برای سناریو چهارم 40](#_Toc211881381)

[شکل (4-22) نمودار میزان Bitrate بر زمان برای سناریو پنجم 41](#_Toc211881382)

[شکل (4-23) نمودار میزان Buffer-Level بر زمان برای سناریو پنجم 42](#_Toc211881383)

[شکل (4-24) نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو پنجم 43](#_Toc211881384)

[شکل (4-25) نمودار میزان Stall-Timeline بر زمان برای سناریو پنجم 44](#_Toc211881385)

[شکل (4-26) نمودار میزان Throughput بر زمان برای سناریو پنجم 45](#_Toc211881386)

[شکل (4-27) نمودار میزان Bitrate بر زمان برای سناریو ششم 46](#_Toc211881387)

[شکل (4-28) نمودار میزان Buffer-Level بر زمان برای سناریو ششم 47](#_Toc211881388)

[شکل (4-29) نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو ششم 48](#_Toc211881389)

[شکل (4-30) نمودار میزان Stall-Timeline بر زمان برای سناریو ششم 49](#_Toc211881390)

[شکل (4-31) نمودار میزان Throughput بر زمان برای سناریو ششم 50](#_Toc211881391)

[شکل (4-32) نمودار میزان Bitrate بر زمان برای سناریو هفتم 51](#_Toc211881392)

[شکل (4-33) نمودار میزان Buffer-Level بر زمان برای سناریو هفتم 52](#_Toc211881393)

[شکل (4-34) نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو هفتم 53](#_Toc211881394)

[شکل (4-35) نمودار میزان Stall-Timeline بر زمان برای سناریو هفتم 54](#_Toc211881395)

[شکل (4-36) نمودار میزان Throughput بر زمان برای سناریو هفتم 55](#_Toc211881396)

[شکل (4-37) نمودار میزان Bitrate بر زمان برای سناریو هشتم 56](#_Toc211881397)

[شکل (4-38) نمودار میزان Buffer-Level بر زمان برای سناریو هشتم 57](#_Toc211881398)

[شکل (4-39) نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو هشتم 58](#_Toc211881399)

[شکل (4-40) نمودار میزان Stall-Timeline بر زمان برای سناریو هشتم 59](#_Toc211881400)

[شکل (4-41) نمودار میزان Throughput بر زمان برای سناریو هشتم 60](#_Toc211881401)

[شکل (4-42) نمودار میزان Bitrate بر زمان برای سناریو نهم 61](#_Toc211881402)

[شکل (4-43) نمودار میزان Buffer-Level بر زمان برای سناریو نهم 62](#_Toc211881403)

[شکل (4-44) نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو نهم 63](#_Toc211881404)

[شکل (4-45) نمودار میزان Stall-Timeline بر زمان برای سناریو نهم 64](#_Toc211881405)

[شکل (4-46) نمودار میزان Throughput بر زمان برای سناریو نهم 65](#_Toc211881406)

[شکل (4-47) نمودار میزان Bitrate بر زمان برای سناریو دهم 66](#_Toc211881407)

[شکل (4-48) نمودار میزان Buffer-Level بر زمان برای سناریو دهم 67](#_Toc211881408)

[شکل (4-49) نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو دهم 68](#_Toc211881409)

[شکل (4-50) نمودار میزان Stall-Timeline بر زمان برای سناریو دهم 69](#_Toc211881410)

[شکل (4-51) نمودار میزان Throughput بر زمان برای سناریو دهم 70](#_Toc211881411)

[شکل (4-52) نمودار میزان Buffer-Level بر زمان برای سناریو دهم 71](#_Toc211881412)

فهرست جداول

فهرست علائم اختصاري

**QUIC** ……………………………………………………… Quick UDP Internet Connections

**DASH** …………………………………………….. Dynamic Adaptive Streaming over HTTP

**UDP** …………………………………………………………………. User Datagram Protocol

**TCP** …………………………………………………………... Transmission Control Protocol

**HTTP** …………………………………………………………… Hypertext Transfer Protocol

**RTT** ………………………………………………………………………… Round Trip Time

**CDF** ………………………………………………………. Cumulative Distribution Function

**QoE** ……………………………………………………………………. Quality of Experience

**ABR** …………………………………………………………………………. Adaptive Bitrate

**IP** ……………………………………………………………………………. Internet Protocol

**LAN** ……………………………………………………………………... Local Area Network

**Wi-Fi** ……………………………………………………………………….. Wireless Fidelity

**3G** ……………………………………………………….. Third Generation (mobile network)

**4G** ………………………………………………………. Fourth Generation (mobile network)

**5G** ………………………………………………………... Fifth Generation (mobile network)

1. مقدمه
   1. مقدمه

با رشد سریع فناوری‌های ارتباطی و گسترش سرویس‌های پخش ویدیو به صورت آنلاین، کیفیت تجربه کاربر در شبکه‌های اینترنتی به یک دغدغه مهم تبدیل شده است. کاربران امروزی انتظار دارند ویدیوها با تأخیر کم، پایداری بالا و نرخ انتقال داده مناسب در اختیارشان قرار گیرد. اما ناپایداری شبکه، از جمله تأخیر، نوسان پهنای باند و از دست رفتن بسته‌ها، می‌تواند باعث کاهش کیفیت ویدیو و تجربه نامطلوب کاربر شود. در چنین شرایطی، انتخاب پروتکل مناسب انتقال داده نقش کلیدی در بهبود عملکرد و کیفیت پخش ویدیو دارد.

پروتکل‌های سنتی مانند TCP، با وجود تضمین تحویل داده، به دلیل مکانیزم‌های کنترل جریان و بازفرستادن بسته‌ها، در شرایط شبکه با تأخیر و نوسان قادر به ارائه عملکرد بهینه در پخش ویدیوی بلادرنگ نیستند. پروتکل QUIC، به عنوان یک پروتکل نوین مبتنی بر UDP، با هدف کاهش تأخیر، افزایش بهره‌وری و بهبود تجربه کاربر در انتقال داده‌های حساس به زمان طراحی شده است. این پروتکل امکاناتی مانند کنترل جریان پیشرفته، رمزنگاری داخلی و مدیریت بهتر از دست رفتن بسته‌ها را ارائه می‌دهد که آن را برای سیستم‌های پخش ویدیوی مدرن مناسب می‌سازد.

هدف اصلی این پژوهش، پیاده‌سازی و ارزیابی عملکرد پروتکل QUIC در پخش ویدیو تحت شرایط مختلف شبکه است. برای تحلیل بهتر و ارائه دیدگاه مقایسه‌ای، از پروتکل DASH در برخی سناریوها به عنوان مرجع استفاده شده است. آزمایش‌ها و پیاده‌سازی‌ها با بهره‌گیری از شبیه‌ساز Mininet در محیط لینوکس و به کمک زبان Python انجام شده‌اند. نتایج این تحقیق می‌تواند اطلاعات ارزشمندی درباره کارایی QUIC و کاربردهای آن در بهبود تجربه کاربران سیستم‌های پخش ویدیو ارائه دهد و نقش آن را در توسعه فناوری‌های شبکه‌های آینده روشن سازد.

1. مروري بر منابع
   1. مقدمه

با گسترش روزافزون محتوای چندرسانه‌ای در وب، تقاضا برای پخش ویدئوی روان، پایدار و با کیفیت به یکی از چالش‌های اساسی در حوزه شبکه و اینترنت تبدیل شده است. یکی از عوامل کلیدی در بهبود کارایی سامانه‌های پخش ویدئو، پروتکل انتقال داده است که وظیفه انتقال مؤثر بسته‌های اطلاعاتی میان سرور و کاربر را بر عهده دارد. در این میان، خانواده پروتکل‌های HTTP به دلیل سازگاری بالا با زیرساخت‌های وب، نقش محوری در توسعه و گسترش سیستم‌های پخش ویدئو مانند MPEG-DASH و HLS ایفا کرده‌اند.

با این حال، نسخه‌های اولیه HTTP از جمله HTTP/1.1، به دلیل ساختار ترتیبی و وابستگی کامل به TCP، در شرایطی مانند تأخیر بالا، ازدحام شبکه یا از دست رفتن بسته‌ها دچار کاهش عملکرد می‌شدند. این محدودیت‌ها منجر به توسعه نسخه‌های جدیدتر نظیر HTTP/2 گردید که با معرفی قابلیت‌هایی همچون Multiplexing و فشرده‌سازی هدرها تلاش داشت کارایی ارتباط را افزایش دهد. با وجود این، تکیه‌ی HTTP/2 بر TCP باعث شد همچنان مشکل Head-of-Line Blocking و حساسیت به خطای شبکه باقی بماند.

در پاسخ به این چالش‌ها، نسل سوم یعنی HTTP/3 بر پایه‌ی پروتکل نوینی به نام QUIC طراحی شد که بر بستر UDP عمل می‌کند و بسیاری از محدودیت‌های TCP را رفع می‌نماید. QUIC با فراهم کردن ویژگی‌هایی مانند اتصال سریع‌تر (0-RTT)، مدیریت مستقل جریان‌ها و قابلیت بازیابی بسته‌ها در سطح کاربر، عملکرد بسیار بهتری را در شبکه‌های با تأخیر یا نوسان زیاد از خود نشان داده است. این ویژگی‌ها موجب شده HTTP/3 و QUIC به گزینه‌ای مناسب برای پخش ویدئو در شرایط شبکه ناپایدار، به‌ویژه در دستگاه‌های موبایل و محیط‌های متحرک، تبدیل شوند.

از این رو، بررسی سیر تحول پروتکل‌های HTTP و نقش آن‌ها در بهبود کیفیت تجربه کاربر (QoE) در پخش ویدئو، به‌ویژه در مقایسه با فناوری‌های نوین مبتنی بر QUIC، یکی از محورهای اصلی پژوهش‌های اخیر در حوزه شبکه‌های چندرسانه‌ای است. در ادامه مروری جامع بر نسخه‌های مختلف HTTP، ویژگی‌های آن‌ها و جایگاه پروتکل QUIC در بهینه‌سازی فرایند پخش ویدئو ارائه می‌شود.

* 1. مروری بر پروتکل های انتقال
     1. HTTP/1.0 & HTTP/1.1

پروتکل HTTP به عنوان اصلی‌ترین پروتکل برای انتقال اطلاعات در وب، طی سال‌ها توسعه یافته و نسخه‌های مختلفی از آن معرفی شده است. در این بخش به بررسی نسخه‌های HTTP/1.0 و HTTP/1.1 و تفاوت‌های کلیدی آن‌ها پرداخته می‌شود.

نسخه HTTP/1.0 در سال 1996 معرفی شد و نخستین استاندارد عمومی برای تبادل اطلاعات بین مرورگرها و سرورها محسوب می‌شود. از ویژگی‌های مهم این نسخه می‌توان به موارد زیر اشاره کرد:

1. اتصال هر درخواست: در HTTP/1.0 برای هر درخواست یک اتصال TCP جدید برقرار می‌شود و پس از دریافت پاسخ، اتصال بسته می‌گردد.
2. پشتیبانی محدود از کشینگ: قابلیت‌های کشینگ در این نسخه ابتدایی و محدود است.
3. عدم پشتیبانی از درخواست‌های جزئی: امکان دانلود بخشی از یک فایل یا ادامه دانلود فراهم نبود.
4. هدرهای محدود: تعداد و نوع هدرها در این نسخه بسیار محدود بود.
5. این محدودیت‌ها موجب افزایش زمان بارگذاری صفحات و مصرف بالای منابع شبکه و سرور شد، به ویژه در صفحات وبی که شامل منابع متعدد مانند تصاویر، فایل‌های CSS و JavaScript بودند.

نسخه HTTP/1.1 که در سال 1997 معرفی شد، بسیاری از محدودیت‌های نسخه قبلی را برطرف کرد و استاندارد فعلی برای بسیاری از وب‌سایت‌ها محسوب می‌شود. مهم‌ترین ویژگی‌های این نسخه عبارتند از:

1. Persistent Connections (اتصالات پایدار): امکان نگه داشتن اتصال TCP برای چندین درخواست و پاسخ متوالی (Keep-Alive) باعث کاهش چشمگیر زمان برقراری ارتباط و مصرف منابع شبکه شد.
2. Chunked Transfer Encoding: ارسال داده‌ها در بخش‌های کوچک بدون نیاز به مشخص بودن کل حجم محتوا از ابتدا.
3. مدیریت پیشرفته Caching: با استفاده از هدرهایی مانند Cache-Control و ETag، مدیریت بهینه کشینگ فراهم شد.
4. پشتیبانی از درخواست‌های جزئی (Range Requests): امکان دانلود بخشی از فایل یا ادامه دانلود فایل‌های بزرگ.
5. Host Header و میزبانی مجازی: امکان میزبانی چندین وب‌سایت روی یک آدرس IP فراهم شد.
6. هدرهای بیشتر و استانداردسازی بهتر: هدرهایی مانند Connection, Transfer-Encoding, Expect و TE برای بهبود انعطاف‌پذیری پروتکل اضافه شدند.

HTTP/1.1 به دلیل بهبود عملکرد، کاهش تأخیر و بهینه‌سازی مصرف پهنای باند، به نسخه استاندارد در وب مدرن تبدیل شد.

* + 1. HTTP/2

نسخه HTTP/2 که در سال 2015 معرفی شد، به منظور بهبود عملکرد و کاهش تأخیر در انتقال داده‌ها توسعه یافت و بر اساس نسخه HTTP/1.1 طراحی شده است. این نسخه با حفظ سازگاری با معماری HTTP، تغییرات ساختاری و عملکردی قابل توجهی را ارائه می‌دهد.

ویژگی‌های کلیدی HTTP/2 عبارتند از:

1. Multiplexing (ارسال همزمان چند درخواست): امکان ارسال چندین درخواست و دریافت پاسخ‌ها به صورت همزمان روی یک اتصال TCP واحد، بدون نیاز به ایجاد اتصال‌های مجزا. این ویژگی موجب کاهش قابل توجه تأخیر و افزایش بهره‌وری پهنای باند می‌شود.
2. Header Compression (فشرده‌سازی هدرها): با استفاده از الگوریتم HPACK، هدرهای HTTP فشرده شده و حجم داده‌های منتقل شده کاهش می‌یابد، که مخصوصاً در درخواست‌های متوالی با هدرهای مشابه مؤثر است.
3. Server Push (پیش‌بارگذاری توسط سرور): امکان ارسال منابع مورد نیاز قبل از درخواست مرورگر، مانند فایل‌های CSS و JavaScript، که موجب کاهش زمان بارگذاری صفحات وب می‌شود.
4. بهبود مدیریت جریان داده: HTTP/2 از مفهوم Stream و Stream Priority استفاده می‌کند تا منابع با اولویت مناسب ارسال شوند و تجربه کاربری بهینه شود.

به طور کلی، HTTP/2 با کاهش تعداد اتصال‌های TCP، بهبود فشرده‌سازی هدرها و قابلیت ارسال همزمان چند درخواست، سرعت بارگذاری صفحات وب را به طور چشمگیری افزایش می‌دهد و محدودیت‌های HTTP/1.1 را برطرف می‌کند. این نسخه، زمینه‌ساز توسعه پروتکل‌های مدرن‌تر مانند HTTP/3 و QUIC نیز شده است.

* + 1. HTTP/3

HTTP/3 جدیدترین نسخه پروتکل HTTP است که بر اساس پروتکل QUIC توسعه یافته و هدف اصلی آن بهبود عملکرد و کاهش تأخیر در شبکه‌های اینترنتی با شرایط ناپایدار و تأخیر بالا است. این نسخه اولین بار در سال 2018 توسط IETF استاندارد شد و بر خلاف نسخه‌های قبلی که بر TCP متکی بودند، از پروتکل UDP برای انتقال داده استفاده می‌کند. ویژگی‌های کلیدی HTTP/3 عبارتند از:

1. استفاده از QUIC به جای TCP: QUIC یک پروتکل مبتنی بر UDP است که قابلیت اتصال سریع، بازسازی خودکار بسته‌های از دست رفته و کاهش تأخیر اولیه (0-RTT Connection) را فراهم می‌کند. این ویژگی موجب بهبود عملکرد در شبکه‌های با تاخیر بالا و یا با نرخ از دست دادن بسته‌های داده می‌شود.
2. Multiplexing بدون Head-of-Line Blocking: برخلاف HTTP/2 که روی TCP اجرا می‌شود و بسته‌های از دست رفته می‌توانند باعث تأخیر کل جریان شوند، HTTP/3 با QUIC امکان ارسال همزمان چندین جریان مستقل را بدون ایجاد Head-of-Line Blocking فراهم می‌کند.
3. Encryption پیش‌فرض (TLS 1.3): در HTTP/3 همه ارتباطات به صورت پیش‌فرض رمزنگاری شده است که امنیت داده‌ها و حریم خصوصی کاربران را افزایش می‌دهد.
4. بهبود عملکرد برای پخش ویدیو و برنامه‌های تعاملی: با کاهش تأخیر و افزایش قابلیت بازیابی بسته‌ها، HTTP/3 عملکرد بهتری در پخش زنده ویدیو، بازی‌های آنلاین و برنامه‌های تعاملی ارائه می‌دهد.

به طور خلاصه، HTTP/3 با بهره‌گیری از QUIC و ویژگی‌های پیشرفته انتقال داده، محدودیت‌های نسخه‌های قبلی HTTP را برطرف کرده و امکان انتقال سریع و امن داده‌ها در شبکه‌های اینترنتی مدرن را فراهم می‌کند.

* 1. مروري بر ادبيات موضوع

پروتکل QUIC به عنوان جایگزینی مدرن برای TCP با هدف کاهش تأخیر و بهبود عملکرد در ارتباطات چندرسانه‌ای، در سال‌های اخیر توجه زیادی از سوی محققان حوزه‌ی Video Streaming به خود جلب کرده است. مطالعات متعددی به بررسی عملکرد QUIC در مقایسه با HTTP/TCP و به‌ویژه در کاربردهای مبتنی بر DASH (Dynamic Adaptive Streaming over HTTP) پرداخته‌اند.

در یکی از نخستین پژوهش‌ها، Bhat و همکاران عملکرد DASH بر بستر QUIC را مورد تحلیل قرار دادند و نشان دادند که اگرچه QUIC در کاهش زمان راه‌اندازی پخش و تأخیر اولیه مؤثر است، اما در شرایط شبکه‌ی پرتلفات عملکرد آن به‌طور کامل از TCP برتر نیست[1] . در ادامه، همین گروه در پژوهش دیگری با افزودن مکانیزم‌های بازانتقال در سطح برنامه، توانستند کیفیت تجربه‌ی کاربر (QoE) را در جلسات پخش تطبیقی بهبود دهند و از افت کیفیت در مواجهه با Packet Loss جلوگیری کنند [2].

در پژوهشی دیگر، Mondal و Chakraborty با تمرکز بر هم‌خوانی الگوریتم‌های Adaptive Bitrate (ABR) با QUIC، نشان دادند که در شرایط نوسانی شبکه، رفتار کنترل ازدحام QUIC می‌تواند باعث ناپایداری در نرخ بیت انتخابی شود، هرچند میانگین تأخیر نسبت به DASH/TCP بهبود می‌یابد [3]. همچنین مطالعه‌ی Zhang و همکاران حاکی از آن است که در شبکه‌های پرسرعت، مزیت QUIC در کاهش تأخیر به اندازه‌ی انتظار نیست و بهینه‌سازی‌های بیشتری در سطح زمان‌بندی بسته‌ها مورد نیاز است [4].

در یک پایان‌نامه‌ی آزمایشگاهی دیگر، نتایج شبیه‌سازی نشان داد که پروتکل QUIC در مقایسه با TCP تأخیر انتها به انتهای کمتر، زمان بازانتقال پایین‌تر و کارایی بالاتری در شبکه‌های با توپولوژی‌های مختلف دارد. همچنین، به دلیل استقلال از لایه‌ی انتقال TCP و استفاده از UDP، زمان پاسخ‌دهی چندکاناله در QUIC کاهش یافته و تأثیر ازدحام شبکه کمتر می‌شود. این نتایج بیانگر بهبود قابل توجه در عملکرد QUIC در برابر DASH/TCP در سناریوهای واقعی است [5].

* 1. نتيجه‌گيري

‌با توجه به مطالعات مرور‌شده در این فصل، می‌توان نتیجه گرفت که پروتکل QUIC به عنوان نسل جدیدی از پروتکل‌های انتقال داده، توانسته است بسیاری از محدودیت‌های TCP در حوزه‌ی جریان ویدئویی تطبیقی (Adaptive Streaming) را برطرف کند. تحقیقات پیشین نشان داده‌اند که QUIC به دلیل استفاده از UDP و حذف وابستگی به فرآیندهای سه‌مرحله‌ای برقراری اتصال و کنترل ازدحام سنتی، موجب کاهش محسوس تأخیر انتها به انتها (End-to-End Delay) و زمان راه‌اندازی پخش (Startup Delay) می‌شود [1]-[4].

با این حال، مرور پژوهش‌ها بیانگر آن است که هنوز خلأهای تحقیقاتی قابل‌توجهی در این زمینه وجود دارد. نخست، بسیاری از پژوهش‌ها تنها در محیط‌های آزمایشگاهی با توپولوژی‌های ساده انجام شده‌اند و نیاز به ارزیابی در شبکه‌های پویا و ناپایدار (به‌ویژه در شرایط high-loss و mobile environments) وجود دارد [3]، [5].

دوم، الگوریتم‌های Adaptive Bitrate (ABR) در ترکیب با رفتار کنترل ازدحام QUIC هنوز به‌صورت بهینه تنظیم نشده‌اند و در برخی سناریوها منجر به نوسان نرخ بیت و کاهش QoE می‌شوند.

در نتیجه، با وجود عملکرد بهتر QUIC نسبت به TCP در اغلب سناریوهای استریم ویدیو، هنوز زمینه‌ی قابل‌توجهی برای تحقیق در حوزه‌ی بهینه‌سازی مکانیزم‌های کنترل ازدحام و زمان‌بندی بسته‌ها در چارچوب QUIC وجود دارد. پژوهش‌های آتی می‌توانند با تمرکز بر مدل‌سازی تطبیقی رفتار شبکه، بهبود QoE و ارزیابی در شبکه‌های واقعی با تأخیر و نوسان بالا، مسیر توسعه‌ی آتی این پروتکل را هموار کنند.

1. روش تحقيق
   1. مقدمه

در این فصل، روش‌ها و مراحل انجام پژوهش برای پیاده‌سازی و ارزیابی عملکرد پروتکل QUIC در پخش ویدیو به تفصیل شرح داده می‌شود. هدف از فصل روش تحقیق، ارائه توضیحی روشن از محیط آزمایش، ابزارها، شبیه‌سازی‌ها و فرآیند جمع‌آوری داده‌ها است تا امکان بازتولید نتایج و ارزیابی علمی پژوهش فراهم شود.

برای این منظور، ابتدا در محیط لینوکس و با استفاده از شبیه‌ساز Mininet یک توپولوژی شبکه طراحی شد که شامل سه سرور، یک کلاینت و یک روتر می‌باشد. روتر وظیفه اتصال تمامی دستگاه‌ها را بر عهده دارد و یکی از سرورها برای اجرای ترافیک مصنوعی با استفاده از iperf3، سرور دیگر برای سرویس‌دهی پروتکل QUIC و سرور آخر برای پروتکل DASH اختصاص یافته است.

* 1. محتوا
     1. علت انتخاب روش

Mininet به دلیل قابلیت ایجاد سریع دستگاه‌ها، تنظیم توپولوژی شبکه و شبیه‌سازی دقیق شرایط مختلف شبکه، ابزار مناسبی برای این پژوهش محسوب شد. همچنین، وجود کتابخانه‌ها و قابلیت‌های کاربردی در Python برای پروتکل‌های QUIC و DASH، امکان پیاده‌سازی و مدیریت آسان سرورها و کلاینت‌ها را فراهم کرد. محیط لینوکس نیز به دلیل دسترسی بالا به جزئیات شبکه و امکان اعمال تنظیمات دقیق، انتخاب شد و استفاده از آن برای چنین پژوهشی الزامی بود.

با این روش، امکان ارزیابی دقیق عملکرد پروتکل‌ها در شرایط شبکه متفاوت فراهم شده و داده‌های جمع‌آوری‌شده برای تحلیل تأخیر، نرخ انتقال داده و پایداری عملکرد مورد استفاده قرار گرفته‌اند.

* + 1. تشريح كامل روش تحقيق

در ابتدای کار، لازم بود محیطی مناسب برای پیاده‌سازی و شبیه‌سازی فراهم شود. بدین منظور، از ماشین مجازی Oracle VM VirtualBox استفاده شد و یک ماشین مجازی با سیستم‌عامل Ubuntu بر روی آن ایجاد گردید.

پس از نصب و راه‌اندازی لینوکس، بسته‌های موردنیاز بر روی سیستم نصب شدند که مهم‌ترین آن‌ها Mininet بود. این ابزار امکان ساخت و اجرای توپولوژی شبکه‌های شبیه‌سازی‌شده را فراهم می‌کند و برای ایجاد سناریوهای آزمایشی مورد استفاده قرار گرفت.

در مرحله بعد، برای انجام آزمایش‌ها نیاز به یک ویدیوی ۳۰ ثانیه‌ای نمونه بود. این ویدیو در سه کیفیت ۳۶۰p، ۷۲۰p و ۱۰۸۰p با استفاده از ابزار FFmpeg مجدداً کدگذاری شد. سپس هر ویدیو به سگمنت‌های ۸ ثانیه‌ای تقسیم گردید تا بتوان در فرآیند پخش تطبیقی از آن‌ها استفاده کرد. دستورات مورد استفاده برای انجام این فرآیند به صورت زیر است:

ffmpeg -i sample\_low.mp4 -c copy -map 0 -f segment -segment\_time 4 -reset\_timestamps 1 sample\_low\_seg%d.mp4

ffmpeg -i sample\_medium.mp4 -c copy -map 0 -f segment -segment\_time 4 -reset\_timestamps 1 sample\_medium\_seg%d.mp4

ffmpeg -i sample\_high.mp4 -c copy -map 0 -f segment -segment\_time 4 -reset\_timestamps 1 sample\_high\_seg%d.mp4

در گام بعدی، برای ایجاد ارتباط امن میان کلاینت و سرور، یک جفت کلید خصوصی و عمومی تولید شد. این کلیدها با استفاده از ابزار OpenSSL و دستور زیر ساخته شدند:

openssl req -new -x509 -days 365 -nodes -out cert.pem -keyout key.pem

کلیدهای تولیدشده در پوشه‌ی اصلی پروژه قرار دارند و سگمنت‌های ویدیو نیز در پوشه‌ای با نام video\_segments ذخیره شده‌اند تا سرورها بتوانند از طریق این مسیر به فایل‌ها دسترسی داشته باشند.

در ادامه، چندین اسکریپت به زبان Python توسعه داده شد که هر یک وظیفه‌ی خاصی در فرآیند پیاده‌سازی و تست بر عهده دارد:

1. **quic\_server.py:**

این اسکریپت وظیفه پیکربندی و اجرای سرور QUIC را بر عهده دارد. با اجرای آن، یک سرور پخش ویدیو مبتنی بر پروتکل QUIC بر روی پورت 4433 در نود مربوطه راه‌اندازی می‌شود.

این اسکریپت پس از اجرای توپولوژی در سرور شماره 1 (S1) اجرا میشود.

1. **quic\_client.py:**

این اسکریپت توسط کلاینت اجرا می‌شود و به سرور QUIC در پورت 4433 درخواست ویدیو ارسال می‌کند. در حین دریافت داده، سیستم QLOG فعال می‌گردد تا جزئیات عملکرد پروتکل ثبت شود. در پایان، سگمنت‌های ویدیو به همراه دو فایل گزارش در فرمت‌های CSV و TXT در پوشه‌ی results ذخیره می‌شوند.

1. **dash\_server.py:**

این اسکریپت مشابه سرور QUIC عمل می‌کند و سرور پخش ویدیو مبتنی بر DASH را بر روی پورت 8080 راه‌اندازی می‌کند. اطلاعات مورد نیاز این سرور داخل فولدر dash\_content وجود دارد که شامل فایل manifest.mpd و همچنین سگمنت های ویدیو به تفکیک کیفیت آنها است.

این اسکریپت پس از اجرای توپولوژی در سرور شماره 2 (S2) اجرا میشود.

1. **dash\_client.py:**

این اسکریپت ابتدا فایل manifest را از سرور DASH دانلود کرده و سپس بر اساس شرایط شبکه، سگمنت‌های مورد نیاز را دریافت می‌کند. نتایج و گزارش‌ها نیز در پوشه‌ی results ذخیره می‌شوند.

1. **:random\_traffic.py**

این اسکریپت وظیفه ایجاد ترافیک رقابتی به وسیله ابزار Iperf3 است که به صورت تصادفی عمل میکند. به این علت این ترافیک تصادفی است که در هر زمان تعداد ترافیک درخواستی بین 1 تا 5 متغییر است و به صورت تصادفی انتخاب میشود و همچنین نوع پروتکل این درخواست ها بین TCP و UDP به صورت تصادفی انتخاب میشود.

1. **:worst\_traffic.py**

این اسکریپت وظیفه ایجاد ترافیک رقابتی به وسیله ابزار Iperf3 است که به صورت بدترین حالت ممکن، یعنی شبیه سازی پخش یک ویدیو دیگر به وسیله پروتکل UDP در شبکه است، عمل میکند.

1. **:test\_runner.py**

این اسکریپت برای اجرای سناریو انتخابی برای پژوهش کاربرد دارد که آن را باید پس از اجرای توپولوژی در کلاینت (C1) اجرا بگیریم. ساختار این اسکریپت به گونه ایی است که چندین ورودی برای آن وجود دارد که به وسیله آن میتوان سناریو مورد نظر را تعیین و اجرا گرفت.

برای شبیه سازی وضعیت شبکه لینکی که بین کلاینت (C1) و روتر (R1) وجود دارد که نام آن را c1-eth0 نام گذاری کرده ام به عنوان گلوگاه ما انتخاب شده است و به وسیله ابزار TC مقادیر نرخ پهنای باند، نرخ تاخیر، نرخ جیتر و درصد از دست رفتن بسته ها تنظیم میشود.

همچنین برا شبیه سازی ترافیک رقابتی در شبکه، ورودی برای آن وجود دارد که وضعیت تصادفی یا بدترین حالت را مشخص میکند، که در صورتی که مقداری برای آن وارد نشود به صورت پیش فرض هیچ ترافیک رقابتی برای شبکه در نظر گرفته نمیشود.

ورودی های این اسکریپت باید در زمان اجرای آن در خط فرمان کلاینت وارد شود که بدین صورت است:

* -b: Bandwidth (Mbps)
* -d: Delay (ms)
* -j: Jitter (ms)
* -l: Packet-Loss (%)
* -t: Cross-Traffic

هر یک از ورودی ها یک مقدار پیش فرض دارند که در صورتی که مقداری برای آنها تعیین نشود شبکه حالتی را برای حالت اولیه در نظر بگیرد که این مقادیر پیش فرض برای پهنای باند 20Mbps، نرخ تاخیر 10 ms، و برای جیتر و درصد از دست رفتن بسته ها معادل 0 است.

ابتدای این اسکریپت این مقادیر ورودی را دریافت میکند و سپس کل شبکه را به صورت پیش فرض قرار داده و تمامیه قوانین قبلی روی گلوگاه را توسط TC حذف میکند. در نهایت توسط همین ابزار لینک گلوگاه را تنظیم میکند.

در مرحله بعدی ابتدا بررسی میکند که آیا ترافیک رقابتی تنظیم شده است، که در صورتی که تنظیم شده باشد اسکریپت های مخصوص خود را که random\_traffic.py و یا worst\_traffic.py است را اجرا میکند.

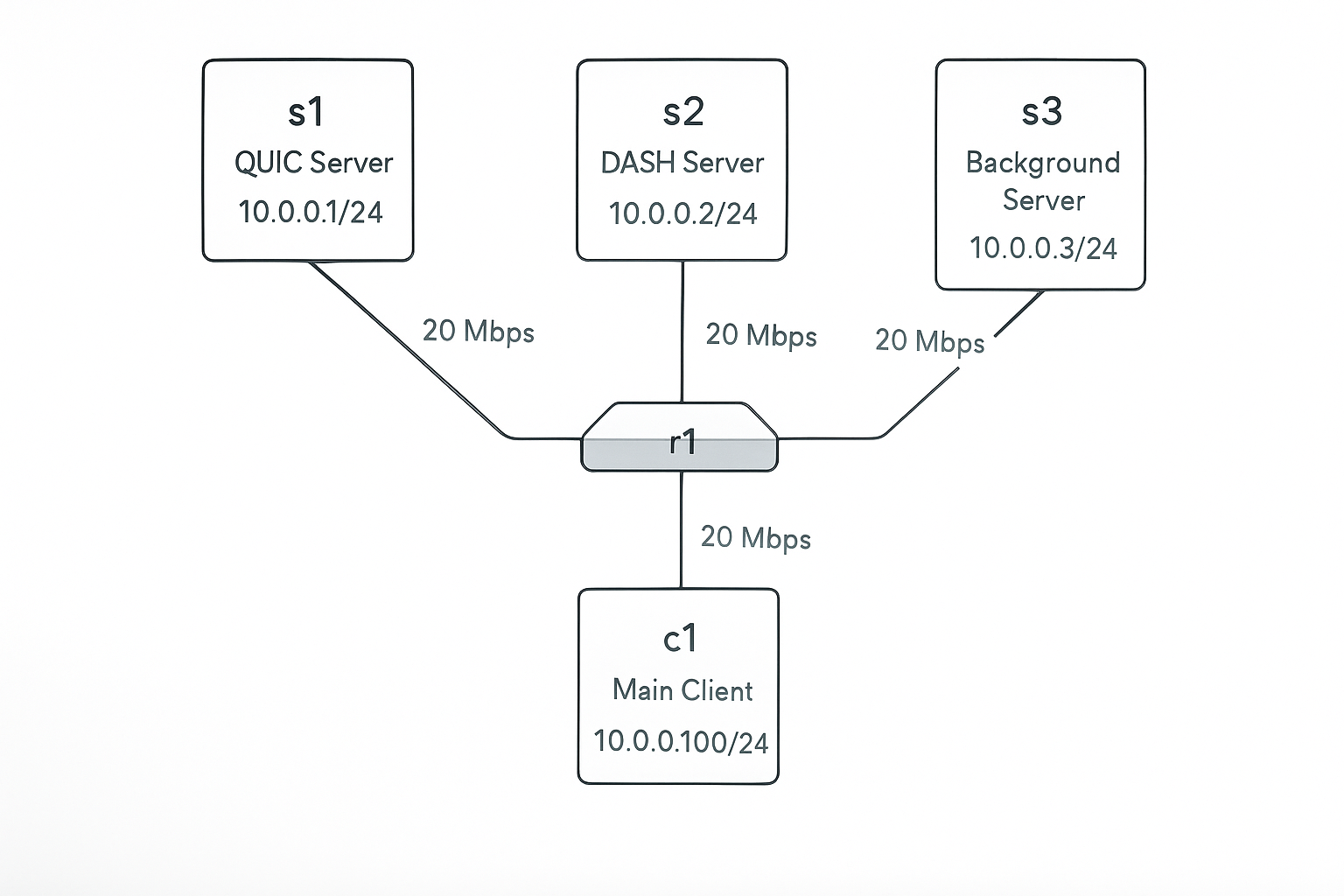
در آخر هم به ترتیب ابتدا اسکریپت quic\_client.py و سپس اسکریپت dash\_clinet.py را اجرا میکند.

1. **:topo.py**

برای اینکه یک توپولوژی برای محیط تست داشته باشیم نیاز است تا این اسکریپت را در ترمینال لینوکس اجرا گرفته که برای این کار نیاز است تا همراه با دستور sudo انجام شود. در این اسکریپت به وسیله ابزار Mininet یک توپولوژی که شامل سه سرور، یک کلاینت، یک روتر و چهار لینک است که وظیفه اتصالات این سرور ها و کلاینت به روتر را دارد.

برای هر سرور و کلاینت به ترتیب مقادیر IP مشخصی تعیین شده است:

* S1: 10.0.0.1/24
* S2: 10.0.0.2/24
* S3: 10.0.0.3/24
* C1: 10.0.0.100/24



توپولوژی شبکه

برای اینکه بتوانیم بر روی هر یک از این host ها کنترل کافی داشته باشیم نیاز است تا محیط خط فرمان برای هر یک وجود داشته باشد که بدین منظور از پکیج MakeTerm برای ایجاد خط فرمان استفاده کردم. زمانی که این اسکریپت اجرا میشود پس از اجرا چهار محیط خط فرمان به صورت مجزا باز میشود که دسترسی کنترلی کاملی به ما میدهد.

پس از اجرا ابتدا باید تمامیه سرور ها را اجرا گرفت که برای این هدف در سرور شماره یک (S1) اسکریپت quic\_server.py، در سرور شماره دو (S2) اسکریپت dash\_server.py و در سرور شماره سه (S3) فرمان مربوط به ابزار Iperf3 برای تعیین سرور است را اجرا میگیریم.

در نهایت برای اجرا سناریو ها در کلاینت (C1) اسکریپت test\_runner.py اجرا میشود.

نمونه کد اجرایی در کلاینت (C1):

python3 test\_runner.py -b 10 -d 40 -j 10 -l 1 -t random

1. **:analise\_results.py**

زمانی که سناریو مورد نظر اجرا گرفته میشود مجموعه ایی از داده ها داخل فولدر های qlog و results ایجاد میشود که برای اینکه بتوانیم به صورت بصری از این داده ها استفاده کنیم تا تحلیل ساده تری برای روی خروجی یک سناریو داشته باشیم باید این اسکریپت را اجرا بگیریم.

با اجرای این اسکریپت داده های سناریو بررسی و تحلیل میشوند و داخل فولدر plots چندین عکس به فرمت png ایجاد میشود که تصویر نمودار های زیر است که توسط کتابخانه هایی مانند pandas، seaborn، matplotlib.pyplot در زبان Python ساخته شده است:

* Bitrate vs Time
* Buffer-Level vs Time
* RTT vs Time
* Stall-Timeline
* Throughput vs Time

1. **:compare\_startup\_delay.py**

برای این یک تحلیل کلی به صورت توزیع تجمیعی بر روی تاخیر ابتدایی تمامیه سناریو ها داشته باشیم ابتدا هر تعداد سناریو که مدنظر است را اجرا گرفته، و سپس فایل های qlog برای پروتکل QUIC و فایل های گزارشات پروتکل DASH را در فولدری به نام CDF\_startup\_delay که داخل فولدری اصلی scenario-result-repo است منتقل میکنیم و سپس این اسکریپت را اجرا میگیریم تا نمودار CDF\_startup\_delay برای ما رسم شود و همچنین یک فایل csv به عنوان گزارش ایجاد میشود.

* + 1. سناریو های مورد تحقیق

در این پژوهش برای اینکه طیف گسترده ایی از حالات مختلف شبکه مورد بررسی قرار گیرد چندین حالت برای هر یک از پارامتر های سناریو شبکه در نظر گرفته شده است که به صورت زیر است:

* Bandwidth: 2, 5, 10, 20 (Mbps)
* Delay: 10, 40, 80 (ms)
* Jitter: 0, 10, 30 (ms)
* Packet-loss: 0, 0.1, 1, 3 (%)
* Traffic-Cross: None, Random, Worst

به علت اینکه در چنین حالتی که یک مجموعه کامل از تمامیه سناریو ها مجموعه ایی 432 تایی است و اجرا تمامیه این سناریو ها زمانگیر و پیچیده بوده است 10 تا از سناریو هایی که به حالت عملیاتی نزدیک هستند و تمام حالت های هر یک از پارامتر ها را پوشش میدهد انتخاب شده اند که به صورت زیر هستند:

* S1: Bandwidth: 20Mbps, Delay: 10ms
* S2: Bandwidth: 10Mbps, Delay: 40ms, Jitter: 10ms, Packet-Loss: 0.1%
* S3: Bandwidth: 5 Mbps, Delay: 40ms, Jitter: 10ms, Packet-Loss: 1%, Cross-Traffic: random
* S4: Bandwidth: 5 Mbps, Delay: 80ms, Jitter: 30ms, Packet-Loss: 1%
* S5: Bandwidth: 2 Mbps, Delay: 80ms, Jitter: 30ms, Packet-Loss: 3%, Cross-Traffic: worst
* S6: Bandwidth: 2 Mbps, Delay: 80ms, Jitter: 30ms, Packet-Loss: 3%
* S7: Bandwidth: 10Mbps, Delay: 80ms, Jitter: 10ms, Packet-Loss: 0.1%
* S8: Bandwidth: 10Mbps, Delay: 40ms, Jitter: 10ms, Packet-Loss: 1%, Cross-Traffic: random
* S9: Bandwidth: 10Mbps, Delay: 40ms, Jitter: 30ms, Packet-Loss: 1%, Cross-Traffic: random
* S10: Bandwidth: 10Mbps, Delay: 40ms, Jitter: 10ms, Packet-Loss: 3%

1. نتايج و تفسير آنها
   1. مقدمه

در این فصل، نتایج حاصل از پیاده‌سازی و شبیه‌سازی سناریوهای مختلف شبکه‌ای ارائه و مورد تحلیل قرار می‌گیرد. همان‌طور که در فصل پیشین تشریح شد، هدف اصلی این پژوهش ارزیابی عملکرد پروتکل‌ QUIC در پخش ویدیو در شرایط مختلف شبکه و مقایسه آن با پروتکل های مشابه مانند MPEG-DASH است. برای این منظور، سناریوهای مختلفی با مقادیر متفاوت عرض باند (Bandwidth)، تأخیر (Delay)، جیتر (Jitter) و نرخ از دست رفتن بسته‌ها (Packet Loss) در محیط Mininet شبیه‌سازی شده‌اند.

در این فصل، ابتدا خروجی‌های حاصل از اجرای هر سناریو شامل نمودارهای تأخیر آغازین (Startup Delay)، نرخ انتقال داده (Throughput)، تعداد و مدت زمان بافر مجدد (Rebuffering) و کیفیت تجربه کاربر (QoE) ارائه می‌گردد. سپس با تحلیل این نتایج، عملکرد هر پروتکل در سناریوهای مختلف مقایسه شده و نقاط قوت و ضعف آن‌ها از منظر کارایی و پایداری ارتباط بررسی می‌شود.

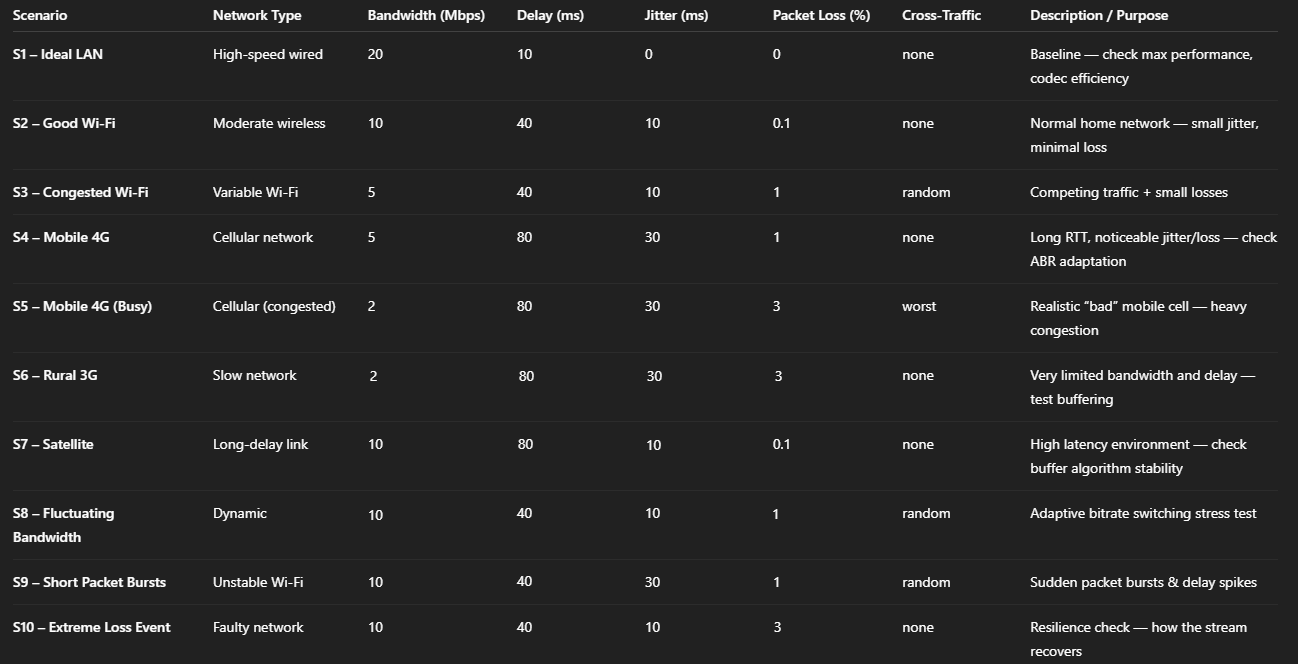
در نهایت، بر اساس داده‌های استخراج‌شده از آزمون‌ها، تفسیر کلی از تأثیر هر پارامتر شبکه بر رفتار پروتکل‌ها ارائه شده و کارایی نسبی هر روش پخش ویدیو در شرایط مختلف شبکه مورد ارزیابی قرار می‌گیرد. نتایج این فصل مبنای نتیجه‌گیری نهایی و پیشنهادهای آتی پژوهش خواهد بود.

* 1. سناریوهای آزمایش پروتکل‌ها:

به منظور ارزیابی عملکرد پروتکل‌های پخش ویدئو شامل QUIC و MPEG-DASH در شرایط مختلف شبکه، مجموعه‌ای از ده سناریوی آزمایشی (S1 تا S10) طراحی و پیاده‌سازی شد.

در این سناریوها، پارامترهای اصلی شبکه شامل پهنای باند (bandwidth)، تأخیر (delay)، نوسان تأخیر یا جیتر (jitter)، نرخ از‌دست‌رفت بسته‌ها (packet loss) و ترافیک متقاطع (cross-traffic) به‌صورت کنترل‌شده تغییر یافتند تا رفتار پروتکل‌ها در محیط‌های با کیفیت‌های متفاوت مورد بررسی قرار گیرد.

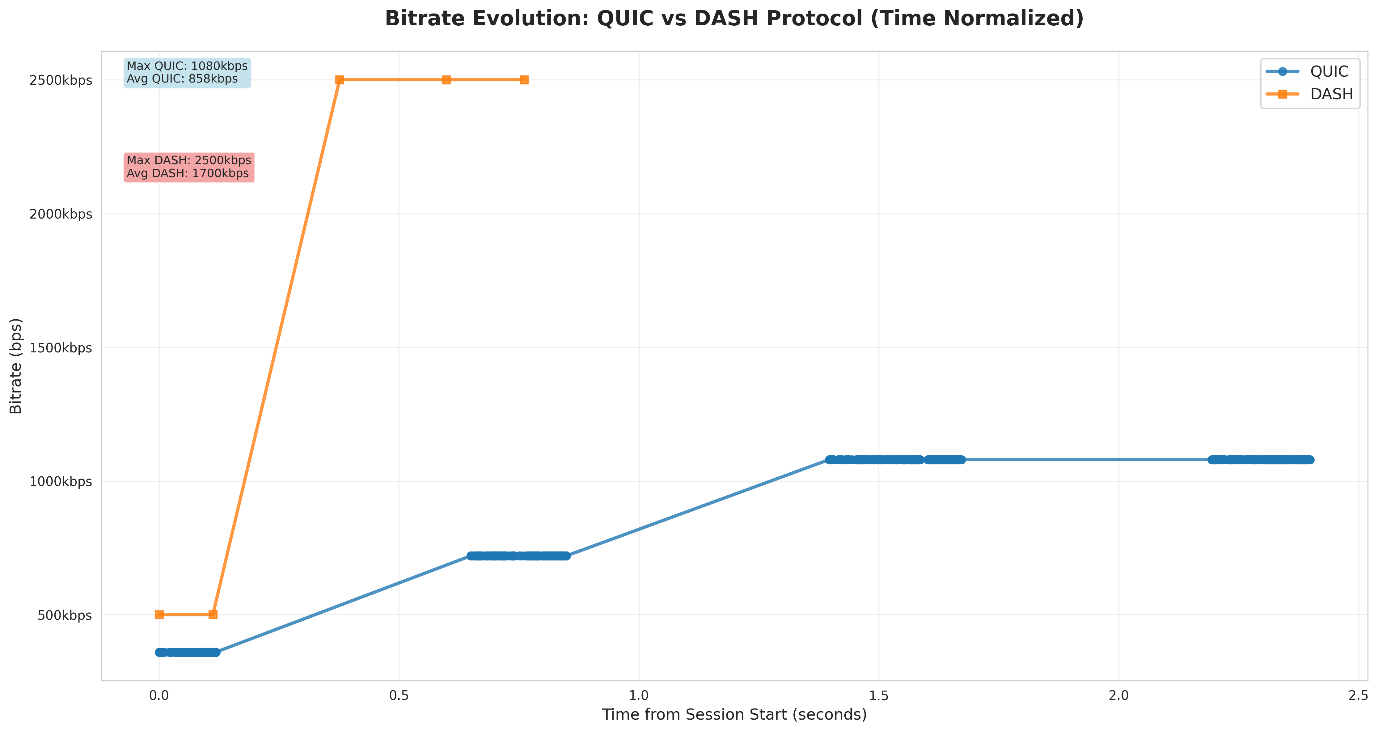
هدف اصلی از این آزمایش‌ها، بررسی پایداری، کیفیت تجربه کاربر (QoE)، و قابلیت سازگاری پروتکل‌ها در شرایط ناپایدار شبکه است.



نمای کلی سناریوهای مورد آزمایش

* + 1. سناریو اول(شبکه LAN ایده آل)

در این سناریو که میتوان گفت خوش بینانه ترین سناریو است، در لینک گلوگاه مقدار 20Mbps پهنای باند و مقدار 10ms تاخیر در نظر گرفته شده است و جیتر و از دستن رفتن بسته ها صفر است. در چنین سناریویی ترافیک های رقابتی وجود ندارد.

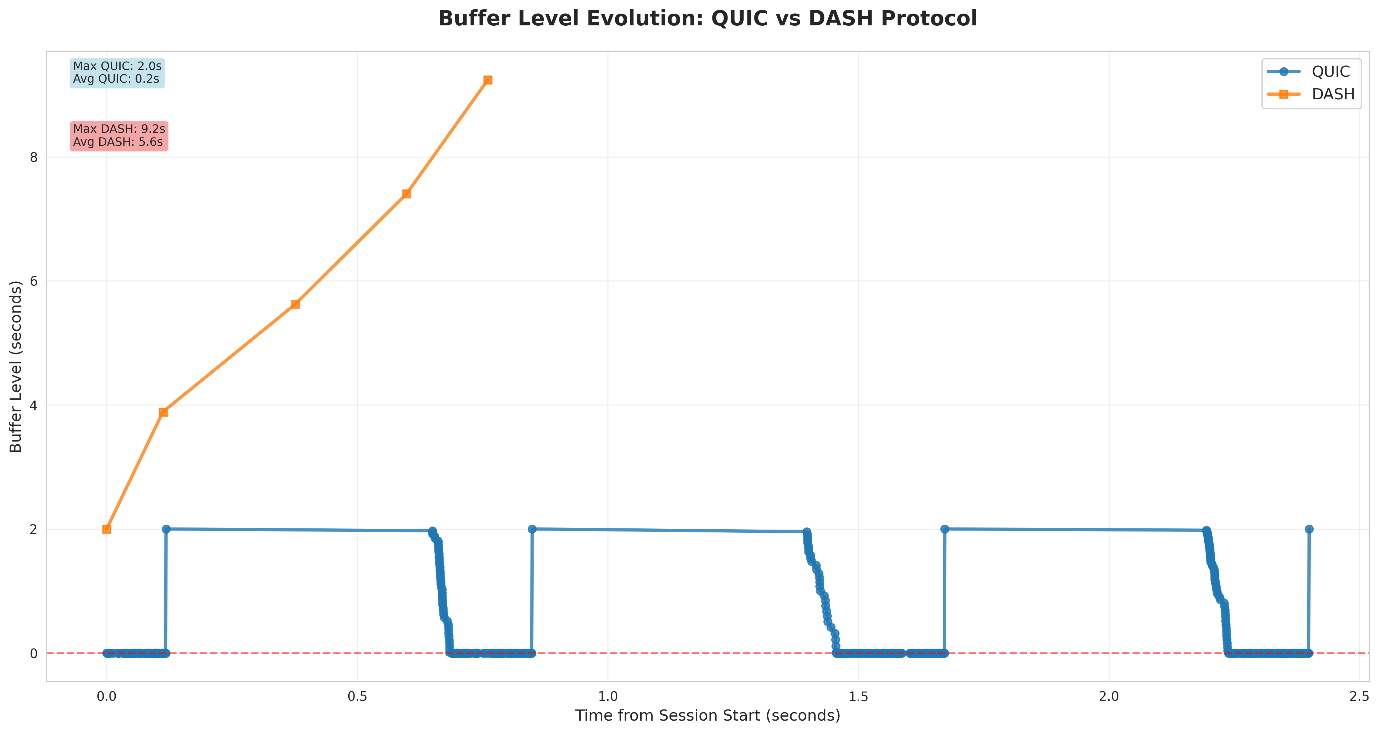


نمودار میزان Bitrate بر زمان برای سناریو اول

در (شکل 2-4) نشان‌دهنده تکامل نرخ بیت‌ریت در طول زمان برای پروتکل‌های QUIC و DASH است.

در پروتکل QUIC نرخ بیت‌ریت از حدود ۳۵۰ kbps شروع شده و پس از حدود ۱.۴ ثانیه به حدود ۱.۱ Mbps می‌رسد. حداکثر نرخ بیت‌ریت به ۱۰۸۰ kbps و میانگین آن ۸۵۸ kbps بوده است. در پروتکل DASH در کمتر از ۰.۴ ثانیه به نرخ ۲.۵ Mbps می‌رسد و پس از آن تقریباً در این نرخ ثابت باقی می‌ماند. حداکثر نرخ بیت‌ریت در DASH ۲۵۰۰ kbps و میانگین آن ۱۷۰۰ kbps بوده است.

با توجه به این اطلاعات به نظر میرسد که DASH به طور سریع‌تری به نرخ‌های بالای بیت‌ریت می‌رسد و در مقایسه با QUIC نرخ میانگین بالاتری دارد. QUIC در ابتدا نرخ بیت‌ریت را به آرامی افزایش می‌دهد.

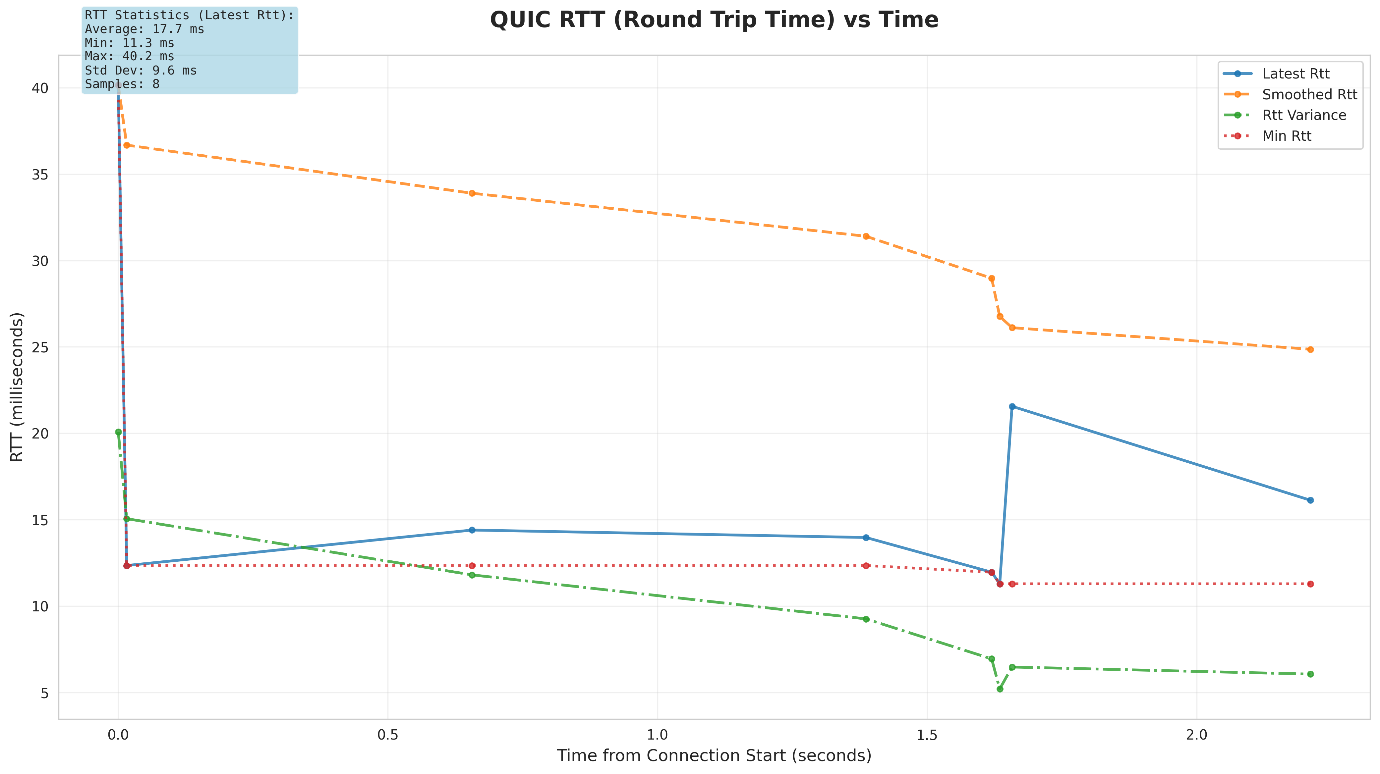


نمودار میزان Buffer-Level بر زمان برای سناریو اول

در (شکل 4-3) نشان‌دهنده تکامل سطح بافر در زمان برای پروتکل‌های QUIC و DASH است. که در QUIC سطح بافر بین ۰ تا ۲ ثانیه نوسان دارد. این نوسان به صورت ناگهانی و به‌طور کوتاه‌مدت مشاهده می‌شود.

اما در DASH سطح بافر به‌طور یکنواخت از ۲ ثانیه به ۹.۲ ثانیه افزایش می‌یابد و در این سطح ثابت باقی می‌ماند. میانگین سطح بافر در DASH ۵.۶ ثانیه بوده است.

با توجه به این اطلاعات به نظر میرسد که DASH با بافر عمیق‌تر و پایدارتر، برای پخش‌های طولانی‌تر و با کیفیت بالاتر مناسب است. QUIC با بافر کوچک‌تر، ممکن است در شرایطی که نوسانات شبکه زیاد باشند، دچار مشکلاتی در حفظ کیفیت پخش شود.

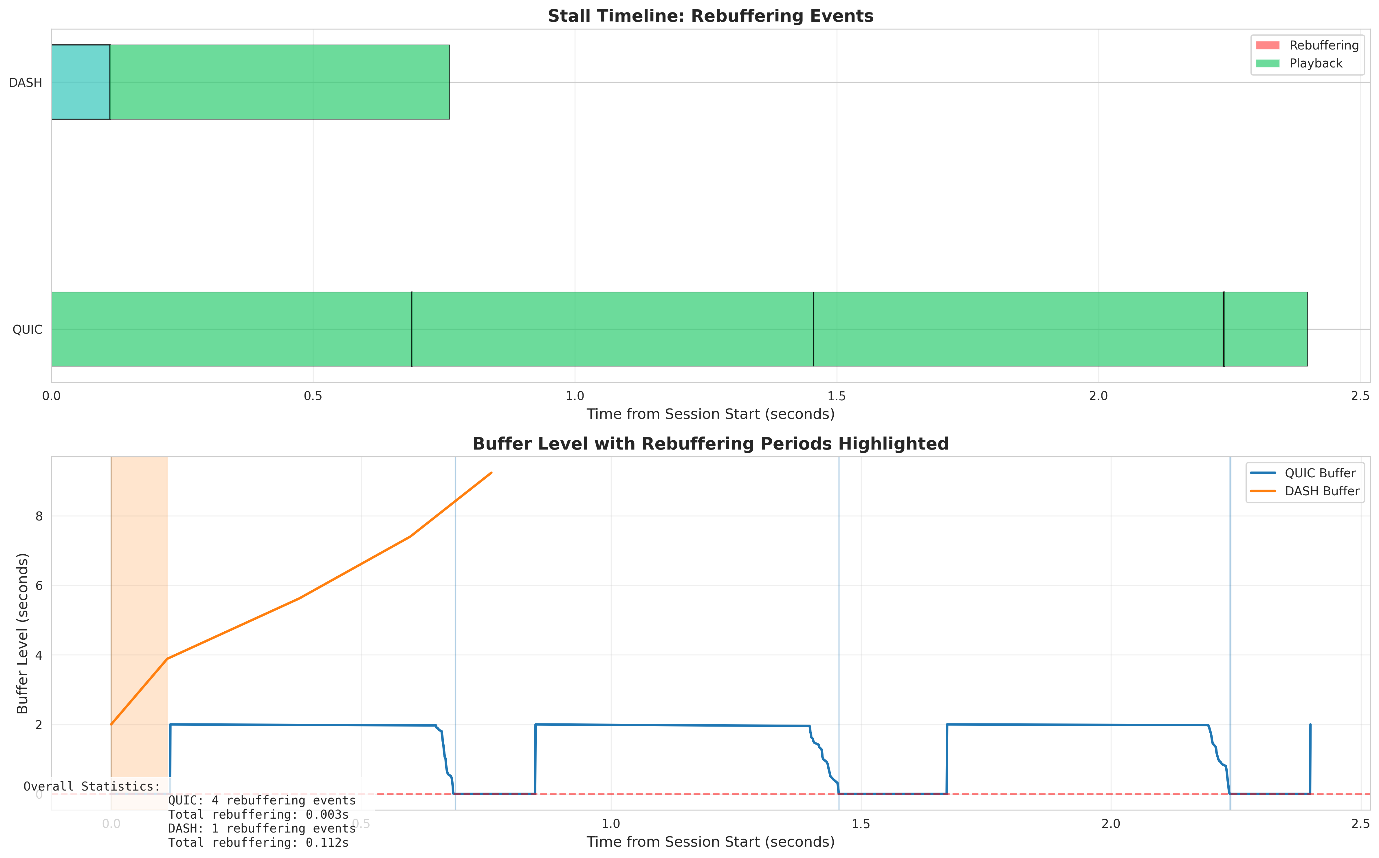


نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو اول

در (شکل 4-4) نمودار زمان رفت‌وبرگشت (RTT) پروتکل QUIC را نشان می‌دهد. RTT جدیدترین (Latest RTT) به طور نوسانی بین ۱۲ تا ۲۲ میلی‌ثانیه قرار دارد. RTT صاف شده (Smoothed RTT) از ۳۷ میلی‌ثانیه به ۲۵ میلی‌ثانیه کاهش می‌یابد. واریانس RTT از حدود ۲۰ میلی‌ثانیه به ۶ میلی‌ثانیه کاهش پیدا می‌کند و حداقل RTT پایدار و حدود ۱۱ میلی‌ثانیه است.

پروتکل QUIC به دلیل سازگاری و بهینه‌سازی‌های مختلف، RTT نسبتاً پایدار و کم دارد.

در آزمایش این سناریو، QUIC توانسته تا حدود زیادی تأخیر را کاهش دهد و شبکه به‌طور یکنواخت‌تر عمل کند.



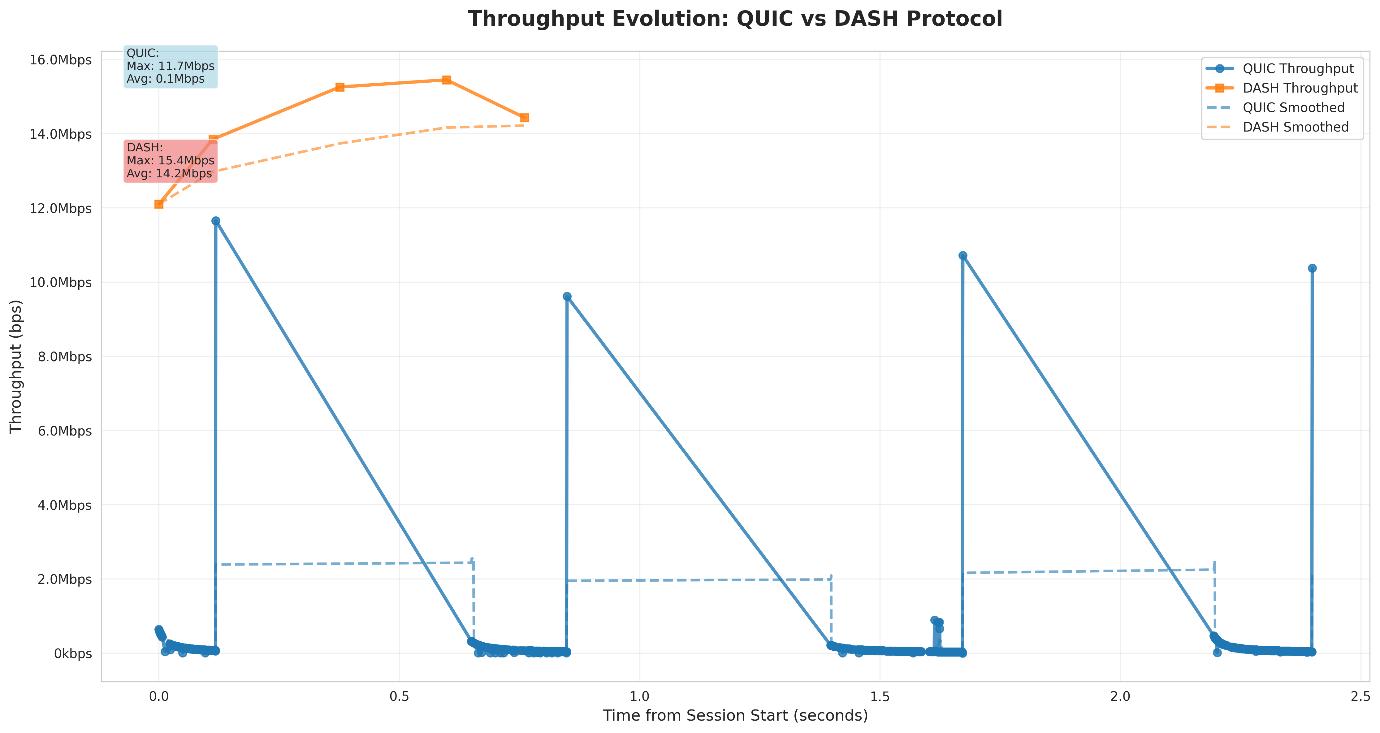
نمودار میزان Stall-Timeline بر زمان برای سناریو اول

در (شکل 4-5) نمایانگر زمان‌بندی توقف‌ها و رویدادهای ریبافر برای هر دو پروتکل QUIC و DASH است.

در QUIC تنها ۴ رویداد ریبافر ثبت شده است که مجموع زمان توقف آن‌ها ۰.۰۰۳ ثانیه بوده است. اما در DASH یک رویداد ریبافر مشاهده شده که زمان توقف آن ۰.۱۱۲ ثانیه بوده است.

QUIC در مقایسه با DASH تقریباً بدون توقف قابل‌مشاهده عمل می‌کند. این ویژگی به دلیل استفاده از بافر کوچک و مدیریت دقیق‌تر زمان‌بندی دانلودها است.

پروتکل DASH، با وجود داشتن بافر بزرگتر، یک توقف اولیه در آغاز کار دارد، اما پس از آن عملکرد پایدارتری دارد.



نمودار میزان Throughput بر زمان برای سناریو اول

در (شکل 4-6) نشان‌دهنده تکامل گذردهی لینک در طول زمان برای پروتکل‌های QUIC و DASH است.

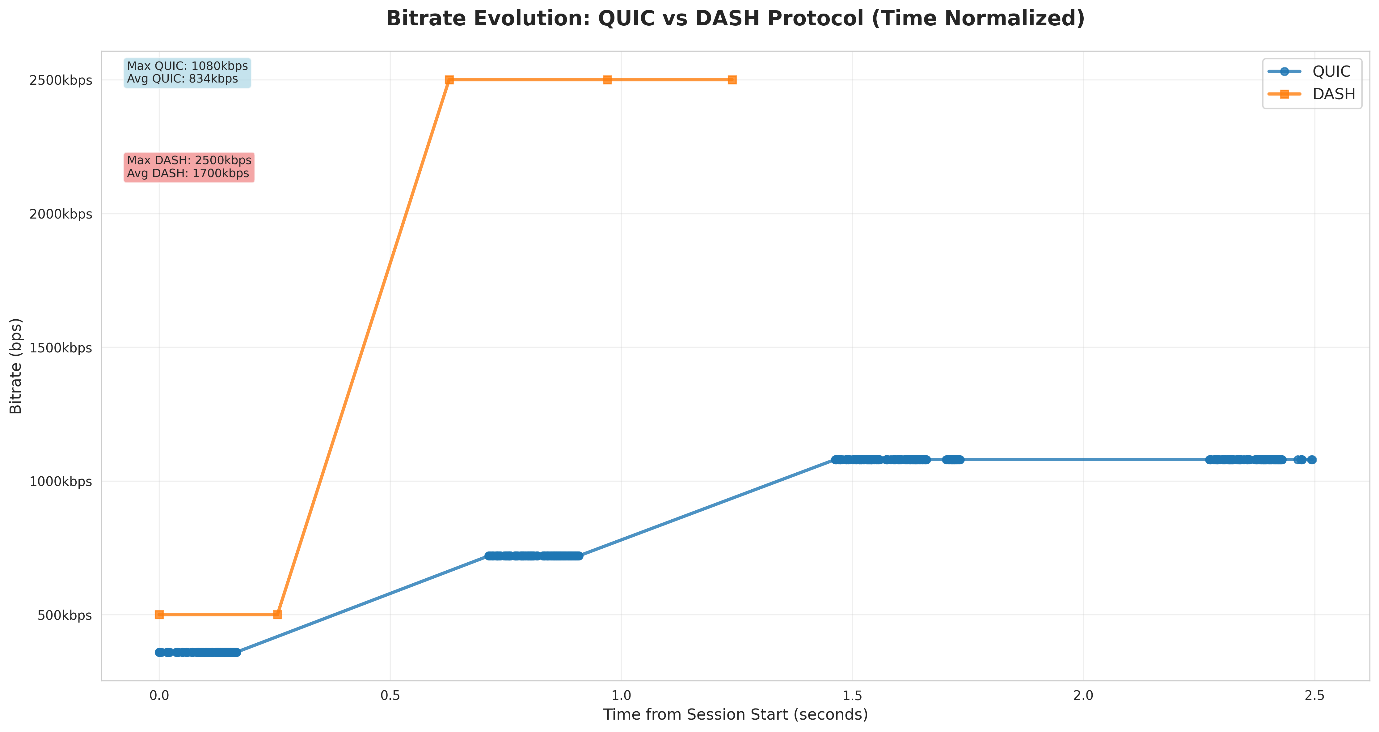
در QUIC قله‌های گذردهی به طور لحظه‌ای تا ۱۱.۷ Mbps می‌رسد، اما بیشتر زمان‌ها در حدود صفر است. میانگین گذردهی QUIC ۰.۱ Mbps و نرخ صاف‌شده آن حدود ۱.۸ تا ۲.۳ Mbps است. در DASH گذردهی DASH در بازه ۱۲ تا ۱۵.۴ Mbps متغیر است و در بیشتر زمان‌ها با یک روند نزولی ملایم عمل می‌کند. میانگین گذردهی DASH ۱۴.۲ Mbps است.

در پروتکل DASH دارای گذردهی پیوسته‌تر و بالاتری است و قادر است از ظرفیت لینک به شکل مؤثرتری استفاده کند.

QUIC، اگرچه در برخی نقاط به گذردهی‌های بالاتری می‌رسد، اما اغلب در فواصل زمانی طولانی با گذردهی نزدیک به صفر مواجه است که ناشی از مدیریت نا‌پیوسته منابع و بافر کوچک است.

* + 1. سناریو دوم(WiFi مناسب)

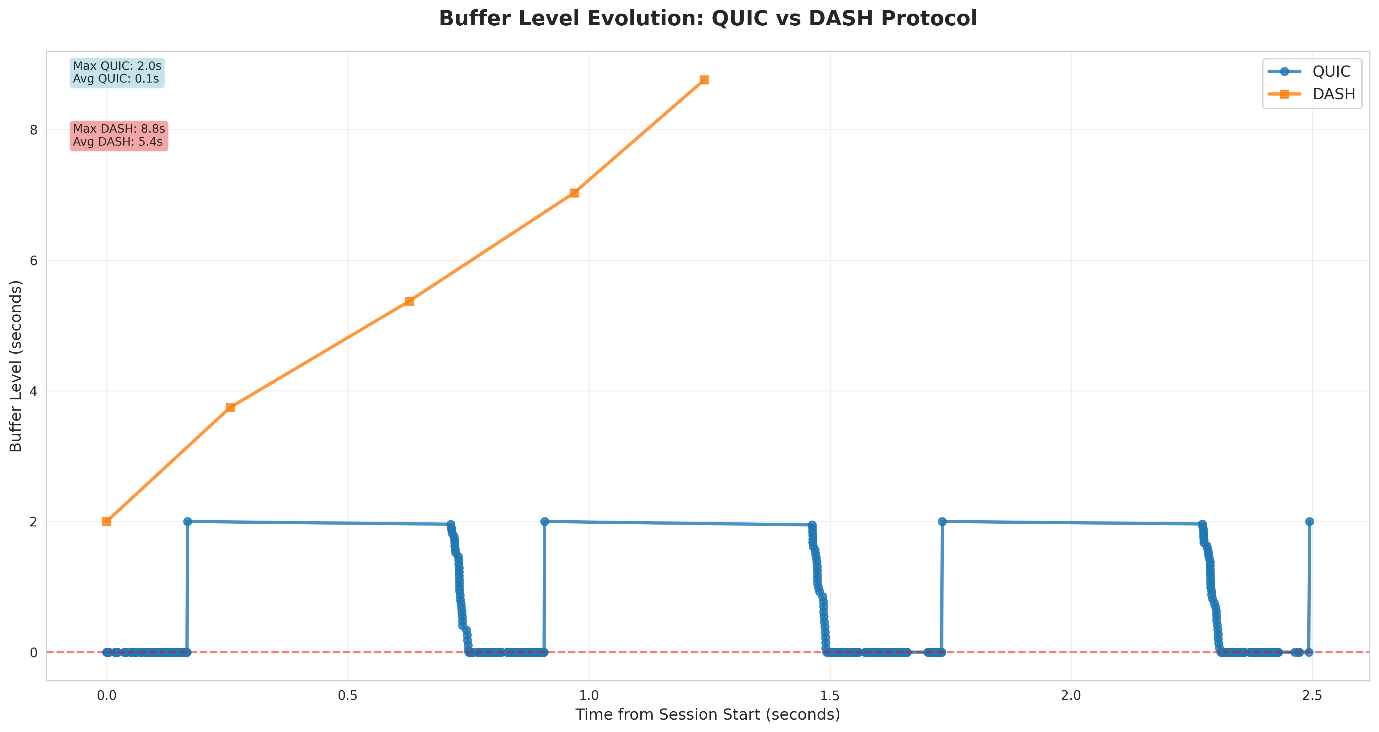
در این سناریو تلاش شده است تا یک شبکه WiFi مناسب شبیه سازی شود که بدین منظور در لینک گلوگاه مقدار 10Mbps پهنای باند و مقدار 40ms تاخیر در نظر گرفته شده است و جیتر معادل 10ms و از دستن رفتن بسته ها 0.1% است. در چنین سناریویی ترافیک های رقابتی وجود ندارد.



نمودار میزان Bitrate بر زمان برای سناریو دوم

در (شکل 4-7) نمودار تکامل نرخ بیت‌ریت در طول زمان برای پروتکل‌های QUIC و DASH را نشان می‌دهد. در QUIC نرخ بیت‌ریت از ۳۵۰ kbps شروع می‌شود و پس از حدود ۱.۵ ثانیه به ۱.۰۸ Mbps می‌رسد. حداکثر نرخ بیت‌ریت برابر با ۱۰۸۰ kbps و میانگین آن ۸۳۴ kbps است. در DASH نرخ بیت‌ریت DASH به سرعت به ۲.۵ Mbps می‌رسد و در آن نقطه ثابت می‌ماند. حداکثر نرخ بیت‌ریت برابر با ۲۵۰۰ kbps و میانگین آن ۱۷۰۰ kbps است.

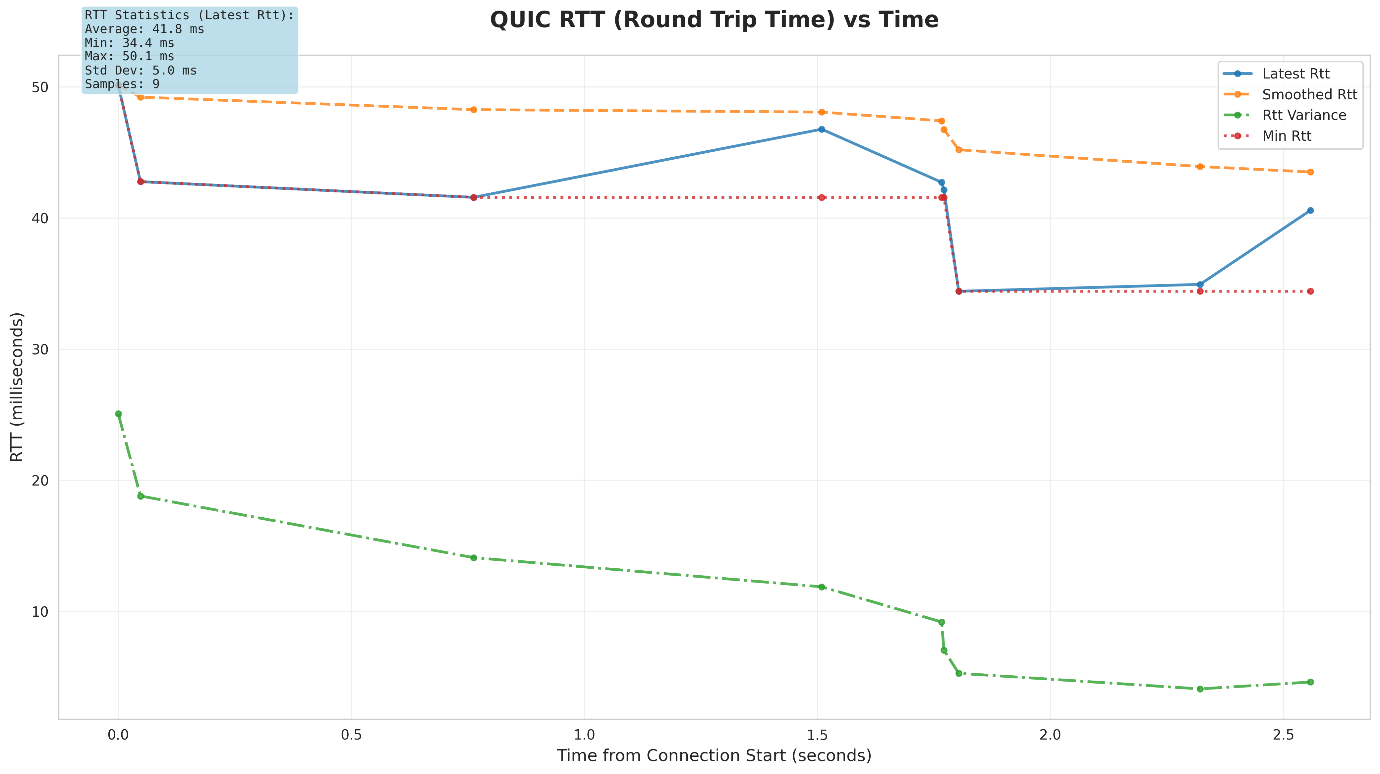
DASH به سرعت به بالاترین نرخ‌های بیت‌ریت می‌رسد و نسبت به QUIC به‌طور میانگین نرخ بیت‌ریت بالاتری دارد. اما در QUIC به‌صورت تدریجی و با شیب ملایم‌تری به افزایش بیت‌ریت می‌پردازد، که این امر ممکن است به دلیل مدیریت محافظه‌کارانه ازدحام و بافر کوچک آن باشد.



نمودار میزان Buffer-Level بر زمان برای سناریو دوم

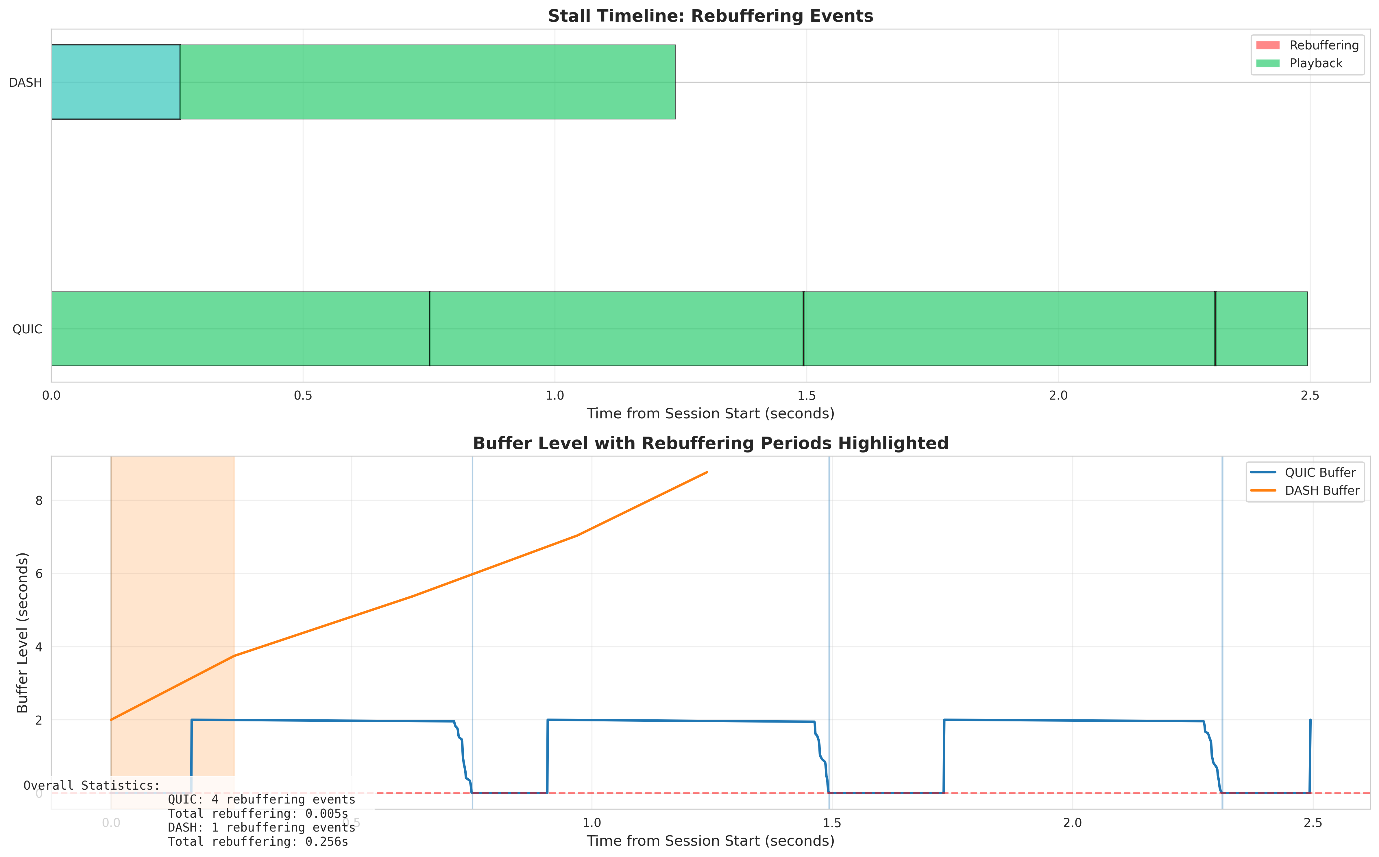
در (شکل 4-8) نمودار تکامل سطح بافر در طول زمان برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC سطح بافر بین ۰ تا ۲ ثانیه نوسان دارد و در ابتدا شاهد تخلیه‌های زیاد و سپس پر شدن‌های سریع آن هستیم. در DASH سطح بافر به‌طور یکنواخت از ۲ تا ۸.۸ ثانیه افزایش می‌یابد و در این سطح باقی می‌ماند. میانگین سطح بافر در DASH ۵.۴ ثانیه است.

DASH با بافر بزرگ‌تر و پایدارتر برای محیط‌هایی که نیاز به پخش پایدار و بدون وقفه دارند، مناسب‌تر است. در مقابل، QUIC با بافر کوچک‌تر و مدیریت زمان‌بندی دقیق‌تر، برای محیط‌هایی که تأخیر کم و واکنش سریع مهم است، عملکرد بهتری دارد.



نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو دوم

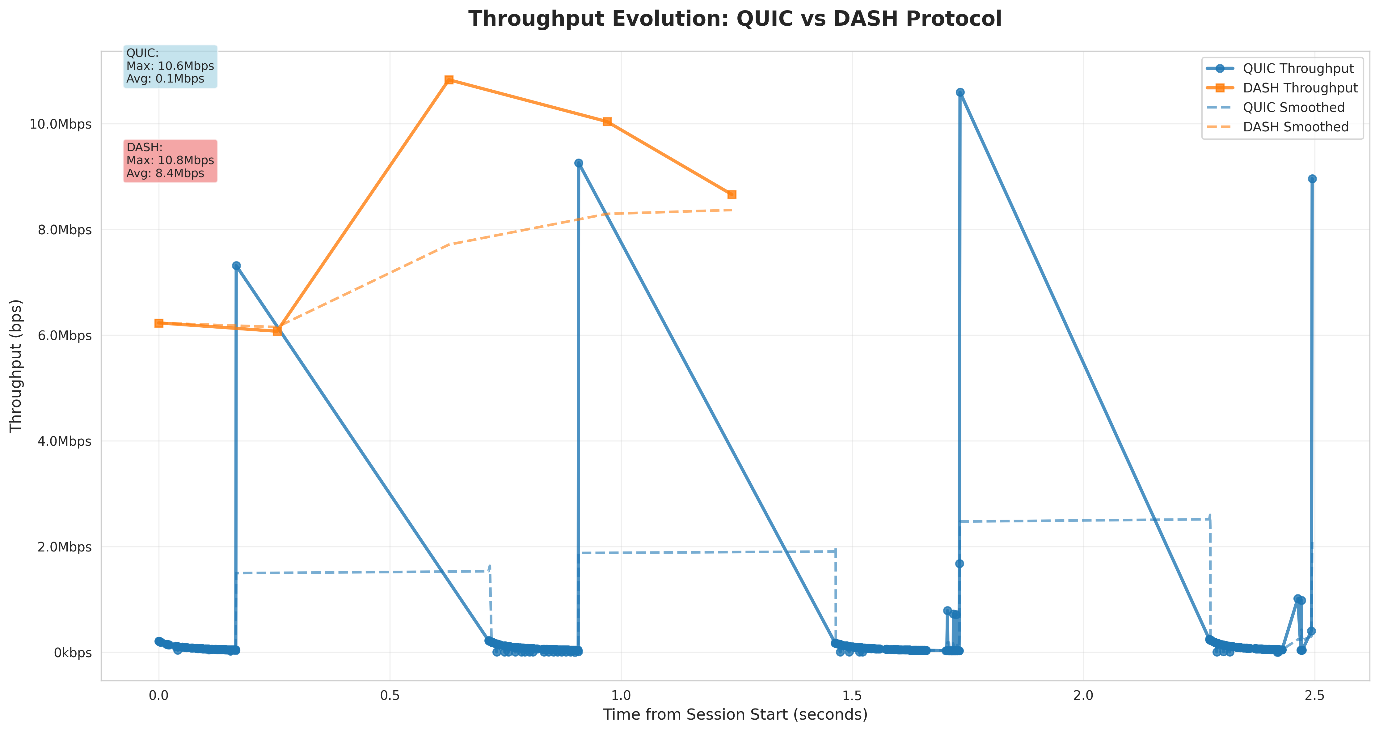
در (شکل 4-9 نمودار زمان رفت‌وبرگشت (RTT) در پروتکل QUIC نشان داده شده است. در این نمودار، RTT جدیدترین (Latest RTT) به طور نوسانی بین ۳۴.۴ تا ۵۰.۱ میلی‌ثانیه قرار دارد. RTT صاف‌شده (Smoothed RTT) از ۴۵ میلی‌ثانیه به ۴۰ میلی‌ثانیه کاهش می‌یابد. همچنین، واریانس RTT به‌طور واضح از ۲۵ میلی‌ثانیه به حدود ۵ میلی‌ثانیه کاهش پیدا می‌کند.

QUIC توانسته RTT را به خوبی مدیریت کند، اما به دلیل نوسانات اولیه، در ابتدا افزایش‌های زیادی را مشاهده می‌کنیم. با گذشت زمان، این نوسانات کاهش یافته و RTT صاف‌شده و واریانس RTT بهبود یافته‌اند.

نمودار میزان Stall-Timeline بر زمان برای سناریو دوم

در (شکل 4-10) نمودار زمان‌بندی توقف‌ها و رویدادهای ریبافر برای پروتکل‌های QUIC و DASH نمایش داده شده است. در QUIC، ۴ رویداد ریبافر ثبت شده‌اند که مجموع زمان توقف آن‌ها تنها ۰.۰۰۵ ثانیه بوده است. در DASH، ۱ رویداد ریبافر ثبت شده که زمان توقف آن ۰.۲۵۶ ثانیه بوده است.

QUIC تقریباً بدون وقفه و به طور پیوسته عمل می‌کند. در حالی که DASH یک وقفه کوتاه در شروع دارد که ممکن است ناشی از مدیریت غیر بهینه بافر در ابتدای پخش باشد، اما پس از آن پخش به‌طور یکنواخت انجام می‌شود.



نمودار میزان Throughput بر زمان برای سناریو دوم

در (شکل 4-11) نمودار تکامل گذردهی لینک برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC، گذردهی لینک در برخی لحظات به حداکثر ۱۰.۶ Mbps می‌رسد، اما در طول آزمایش بیشتر زمان‌ها در حدود صفر باقی می‌ماند. میانگین گذردهی QUIC ۰.۱ Mbps و نرخ صاف‌شده آن بین ۲ تا ۳ Mbps است. در DASH، گذردهی لینک به‌طور پیوسته در بازه ۶–۱۰ Mbps تغییر می‌کند و میانگین آن ۸.۴ Mbps است.

DASH دارای گذردهی پیوسته‌تر و بهتری است و قادر است از ظرفیت لینک به‌صورت مؤثرتری استفاده کند. در حالی که QUIC، اگرچه در برخی لحظات به گذردهی‌های بالاتری می‌رسد، اما به دلیل بافر کوچک و عدم مدیریت مؤثر منابع، گذردهی به طور پیوسته پایین می‌آید.

* + 1. سناریو سوم(WiFi شلوغ در فاصله دور)

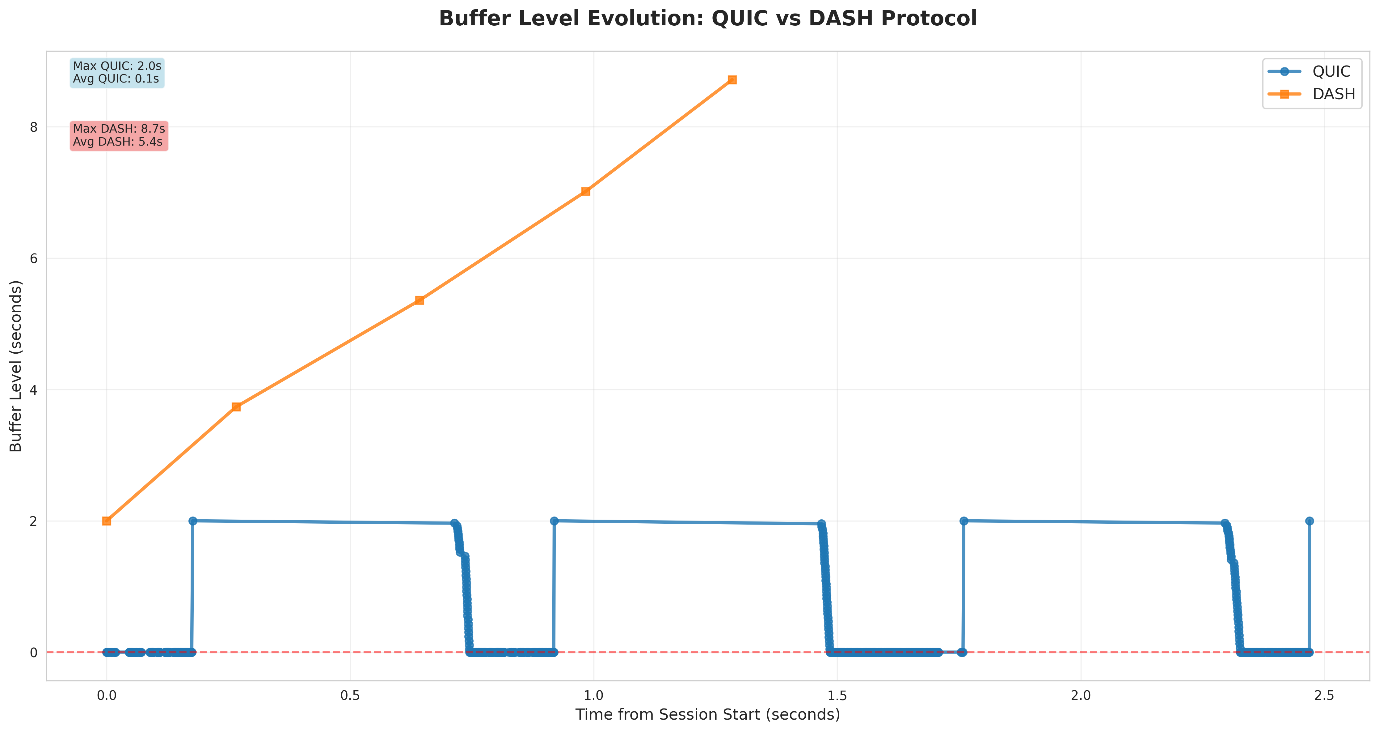
در این سناریو مشابه سناریو دوم تلاش شده تا شبکه یک WiFi که هم در فاصله دوری وجود دارد که دچار تداخل و کاهش پهنای باند شده است و همچنین کاربران دیگری نیز به آن متصل هستند بدین منظور در لینک گلوگاه مقدار 5Mbps پهنای باند و مقدار 40ms تاخیر در نظر گرفته شده است و جیتر معادل 10ms و از دستن رفتن بسته ها 1% است. در چنین سناریویی ترافیک های رقابتی به صورت تصادفی و ترکیبی از 1 تا 5 درخواست TCP و UDP قرار داده شده است.



نمودار میزان Bitrate بر زمان برای سناریو سوم

در (شکل 4-12) نمودار تکامل نرخ بیت‌ریت در طول زمان برای پروتکل‌های QUIC و DASH نمایش داده شده است. در QUIC، نرخ بیت‌ریت از ۵۰۰ kbps شروع می‌شود و پس از حدود ۱.۵ ثانیه به ۱.۰۸ Mbps می‌رسد. حداکثر نرخ بیت‌ریت برابر با ۱۰۸۰ kbps و میانگین آن ۸۷۳ kbps است. در DASH، نرخ بیت‌ریت به سرعت به ۲.۵ Mbps می‌رسد و سپس تقریباً ثابت می‌ماند. حداکثر نرخ بیت‌ریت برابر با ۲۵۰۰ kbps و میانگین آن ۱۷۰۰ kbps است.

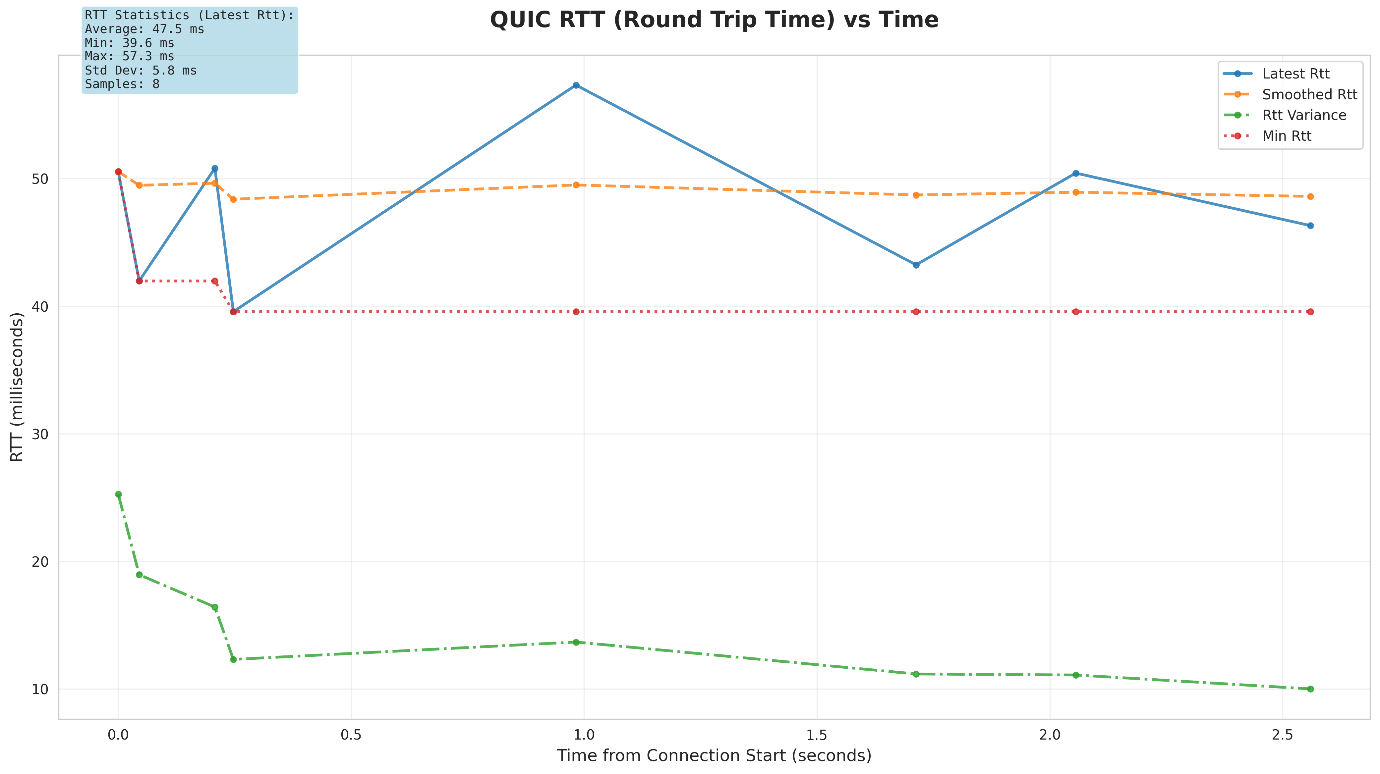
DASH به سرعت به بالاترین نرخ‌های بیت‌ریت می‌رسد و نسبت به QUIC به‌طور میانگین نرخ بیت‌ریت بالاتری دارد. اما در QUIC، نرخ بیت‌ریت به‌صورت تدریجی و با شیب ملایم‌تری افزایش می‌یابد که ممکن است ناشی از مدیریت محافظه‌کارانه ازدحام و بافر کوچک آن باشد.



نمودار میزان Buffer-Level بر زمان برای سناریو سوم

در (شکل 4-13) نمودار تکامل سطح بافر در طول زمان برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC، سطح بافر به طور مداوم بین ۰ تا ۲ ثانیه نوسان دارد و در ابتدا شاهد تخلیه‌های زیادی و سپس پر شدن‌های سریع آن هستیم. در DASH، سطح بافر به‌طور یکنواخت از ۲ تا ۸.۷ ثانیه افزایش می‌یابد و در این سطح ثابت باقی می‌ماند. میانگین سطح بافر در DASH ۵.۴ ثانیه است.

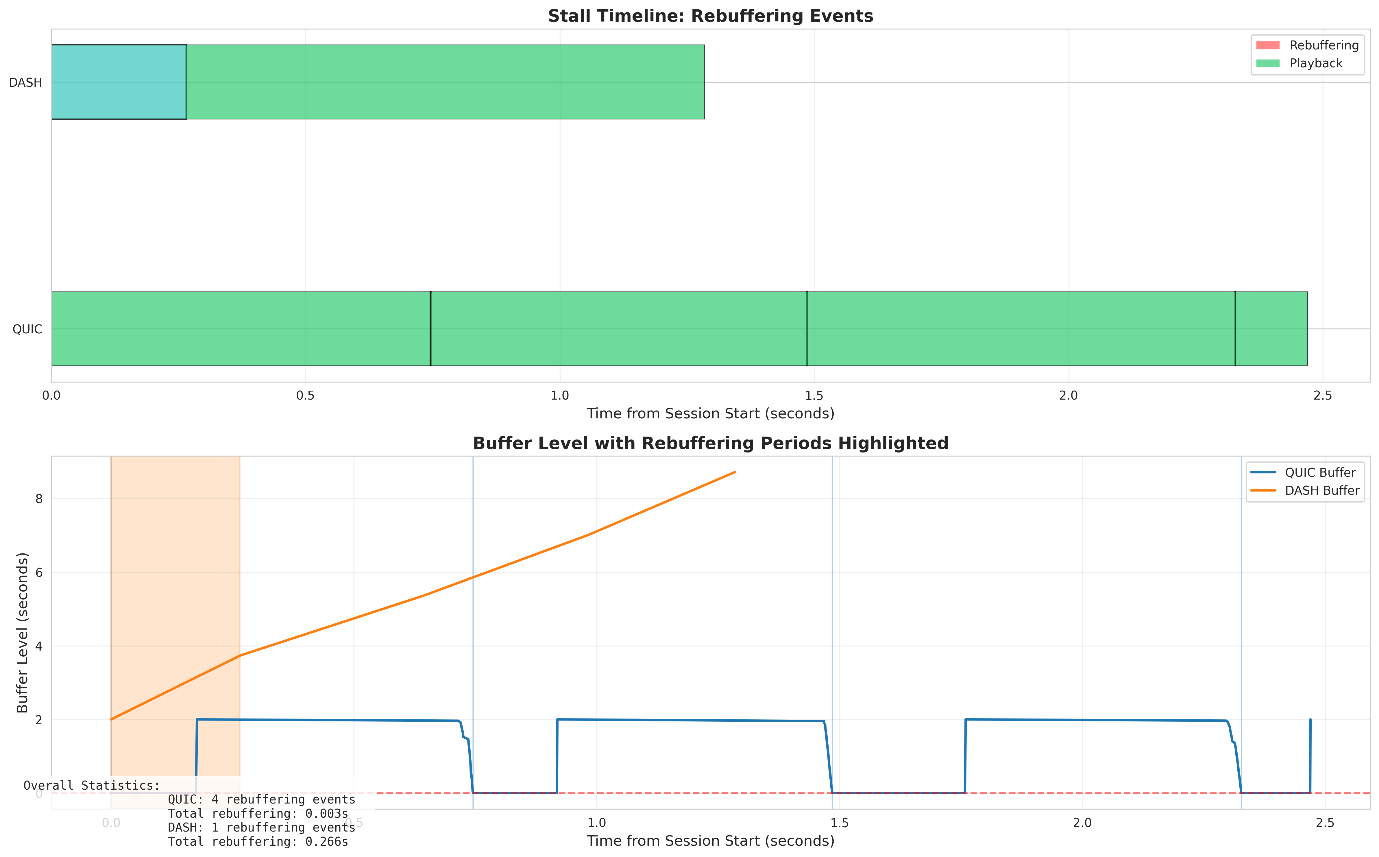
DASH با بافر بزرگ‌تر و پایدارتر برای محیط‌هایی که نیاز به پخش پایدار و بدون وقفه دارند، مناسب‌تر است. در مقابل، QUIC با بافر کوچک‌تر و مدیریت زمان‌بندی دقیق‌تر، برای محیط‌هایی که تأخیر کم و واکنش سریع مهم است، عملکرد بهتری دارد.



نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو سوم

در (شکل 4-14) نمودار زمان رفت‌وبرگشت (RTT) در پروتکل QUIC نمایش داده شده است. در این نمودار، RTT جدیدترین (Latest RTT) به طور نوسانی بین ۳۹.۶ تا ۵۷.۳ میلی‌ثانیه قرار دارد. RTT صاف‌شده (Smoothed RTT) از ۴۷ میلی‌ثانیه به ۴۱ میلی‌ثانیه کاهش می‌یابد. همچنین، واریانس RTT به‌طور واضح از ۲۵ میلی‌ثانیه به حدود ۱۰ میلی‌ثانیه کاهش پیدا می‌کند.

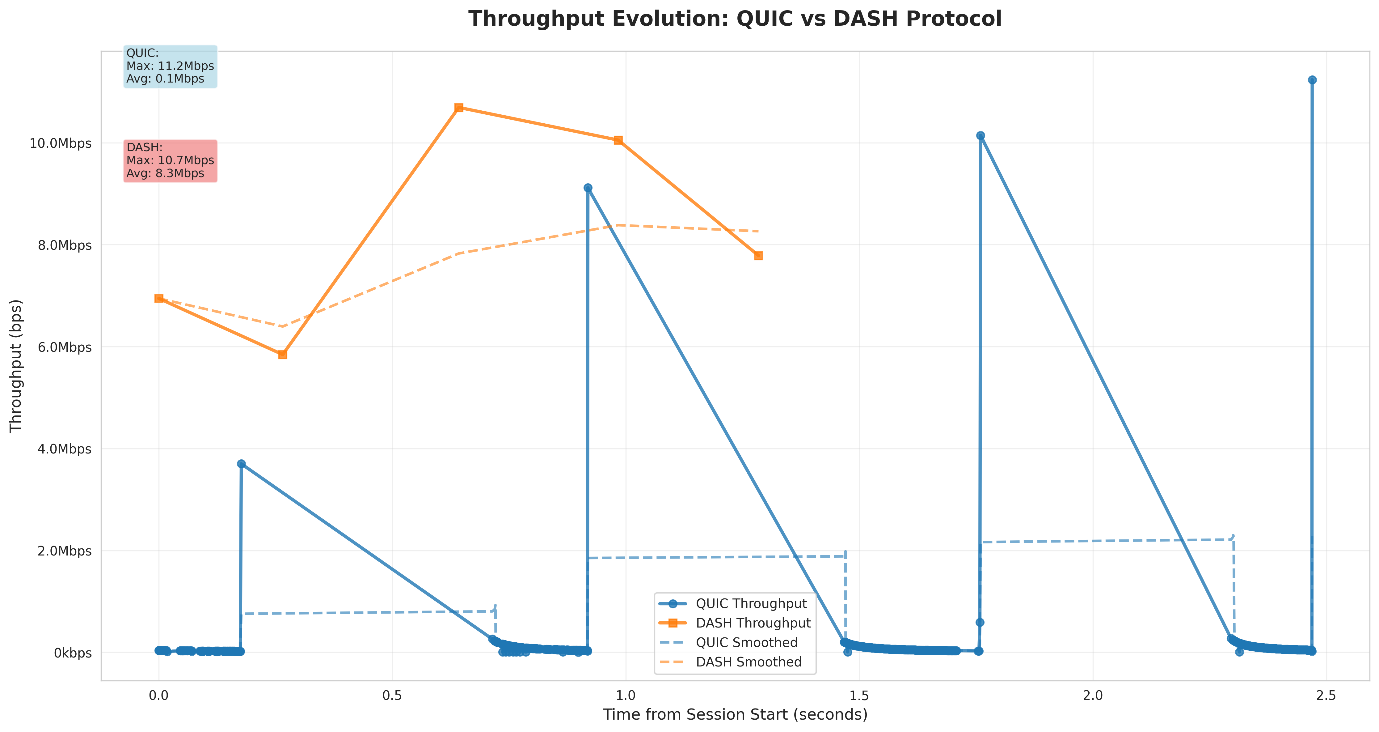
QUIC توانسته RTT را به خوبی مدیریت کند، اما به دلیل نوسانات اولیه، در ابتدا افزایش‌های زیادی را مشاهده می‌کنیم. با گذشت زمان، این نوسانات کاهش یافته و RTT صاف‌شده و واریانس RTT بهبود یافته‌اند.



نمودار میزان Stall-Timeline بر زمان برای سناریو سوم

در (شکل 4-15) نمودار زمان‌بندی توقف‌ها و رویدادهای ریبافر برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC، ۴ رویداد ریبافر ثبت شده‌اند که مجموع زمان توقف آن‌ها تنها ۰.۰۰۳ ثانیه بوده است. در DASH، ۱ رویداد ریبافر ثبت شده که زمان توقف آن ۰.۲۶۶ ثانیه بوده است.

QUIC تقریباً بدون وقفه و به طور پیوسته عمل می‌کند. در حالی که DASH یک وقفه کوتاه در شروع دارد که ممکن است ناشی از مدیریت غیر بهینه بافر در ابتدای پخش باشد، اما پس از آن پخش به‌طور یکنواخت انجام می‌شود.



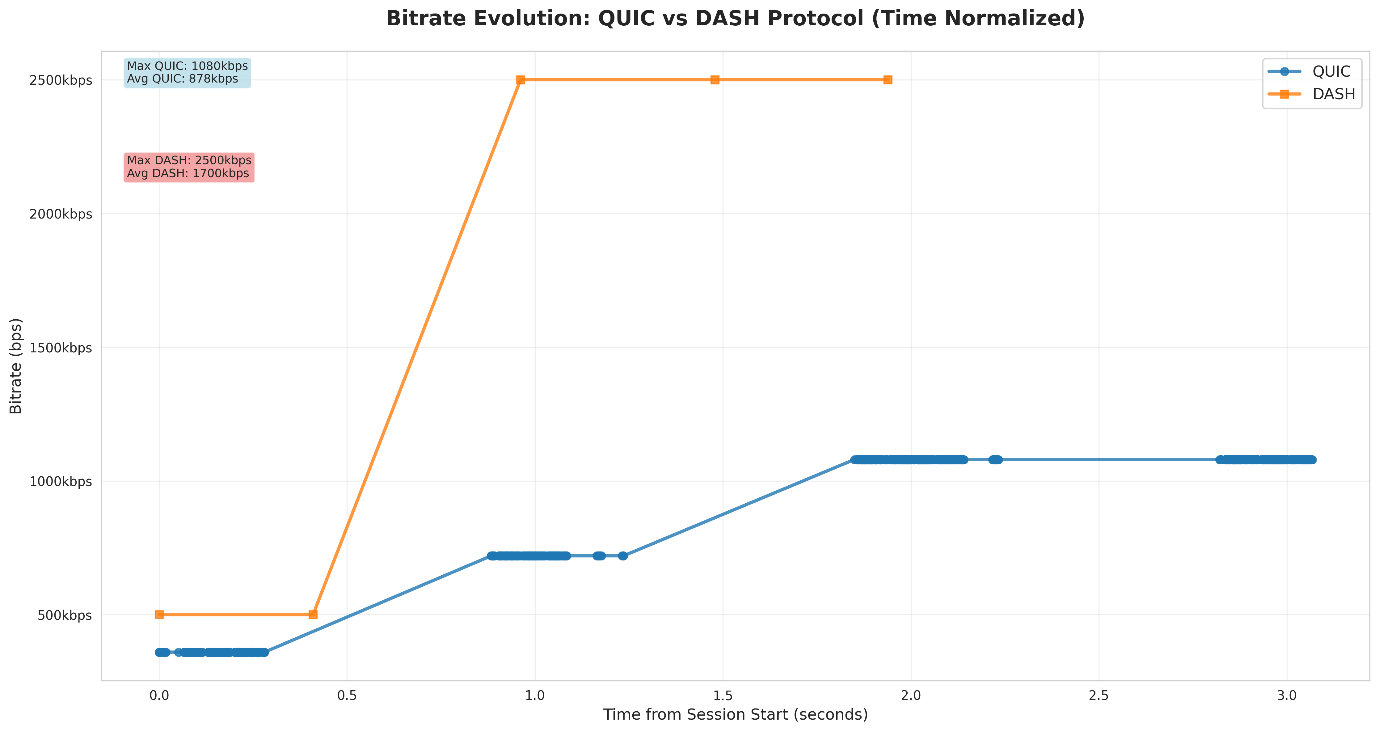
نمودار میزان Throughput بر زمان برای سناریو سوم

در (شکل4-16 ) نمودار تکامل گذردهی لینک برای پروتکل‌های QUIC و DASH نمایش داده شده است. در QUIC، گذردهی لینک در برخی لحظات به حداکثر ۱۱.۲ Mbps می‌رسد، اما در طول آزمایش بیشتر زمان‌ها در حدود صفر باقی می‌ماند. میانگین گذردهی QUIC ۰.۱ Mbps و نرخ صاف‌شده آن بین ۲ تا ۳ Mbps است. در DASH، گذردهی لینک به‌طور پیوسته در بازه ۶–۱۰ Mbps تغییر می‌کند و میانگین آن ۸.۳ Mbps است.

DASH دارای گذردهی پیوسته‌تر و بهتری است و قادر است از ظرفیت لینک به‌صورت مؤثرتری استفاده کند. در حالی که QUIC، اگرچه در برخی لحظات به گذردهی‌های بالاتری می‌رسد، اما به دلیل بافر کوچک و عدم مدیریت مؤثر منابع، گذردهی به طور پیوسته پایین می‌آید.

* + 1. سناریو چهارم(موبایل 4G)

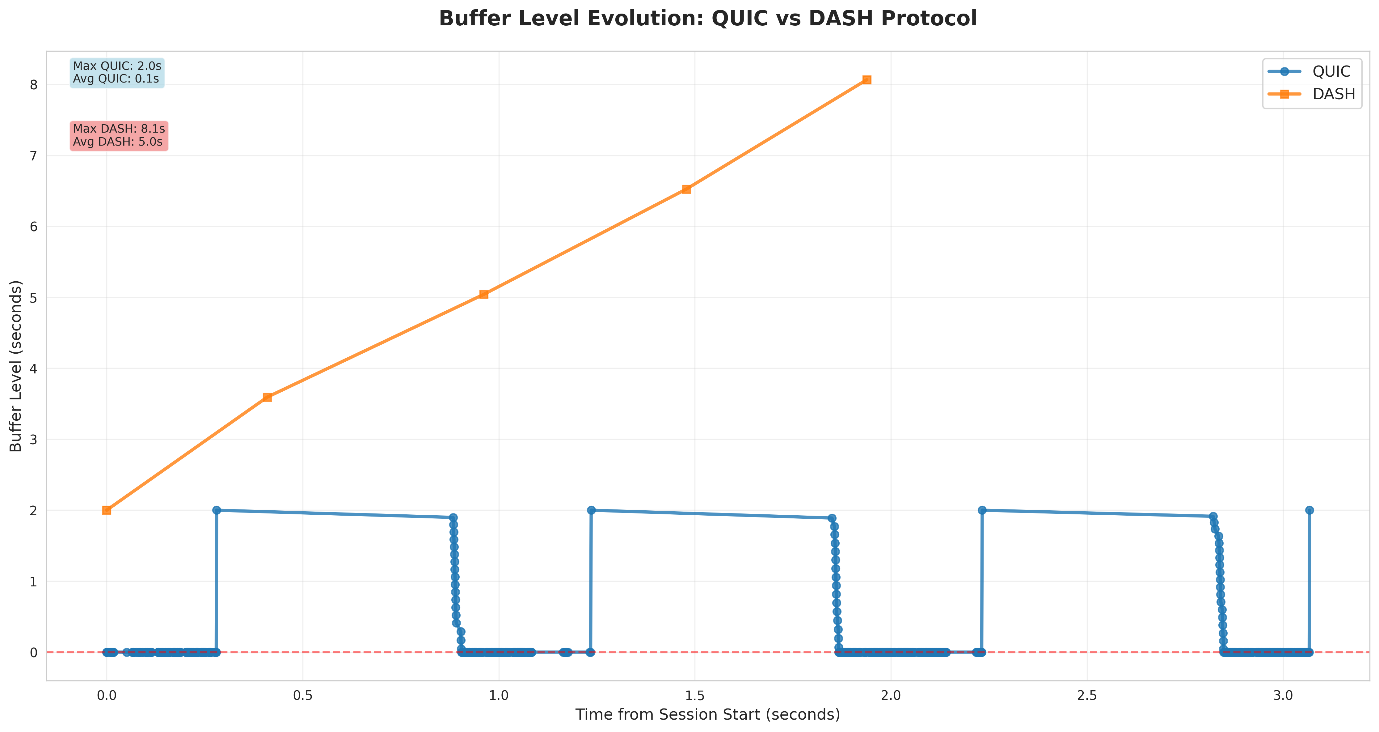
این سناریو تلاش شده است تا شبکه ایی شبیه سازی شود که کلاینت توسط شبکه تلفن همراه 4G در یک مکان خلوت درخواست پخش ویدیو را میدهد، بدین منظور در لینک گلوگاه مقدار 5Mbps پهنای باند و مقدار 80ms تاخیر در نظر گرفته شده است و جیتر معادل 30ms و از دستن رفتن بسته ها 1% است. در چنین سناریویی ترافیک رقابتی وجود ندارد.



نمودار میزان Bitrate بر زمان برای سناریو چهارم

در (شکل 4-17) نمودار تکامل نرخ بیت‌ریت در طول زمان برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC، نرخ بیت‌ریت از ۵۰۰ kbps شروع می‌شود و پس از حدود ۱.۵ ثانیه به ۱.۰۸ Mbps می‌رسد. حداکثر نرخ بیت‌ریت برابر با ۱۰۸۰ kbps و میانگین آن ۸۷۸ kbps است. در DASH، نرخ بیت‌ریت به سرعت به ۲.۵ Mbps می‌رسد و سپس تقریباً ثابت می‌ماند. حداکثر نرخ بیت‌ریت برابر با ۲۵۰۰ kbps و میانگین آن ۱۷۰۰ kbps است.

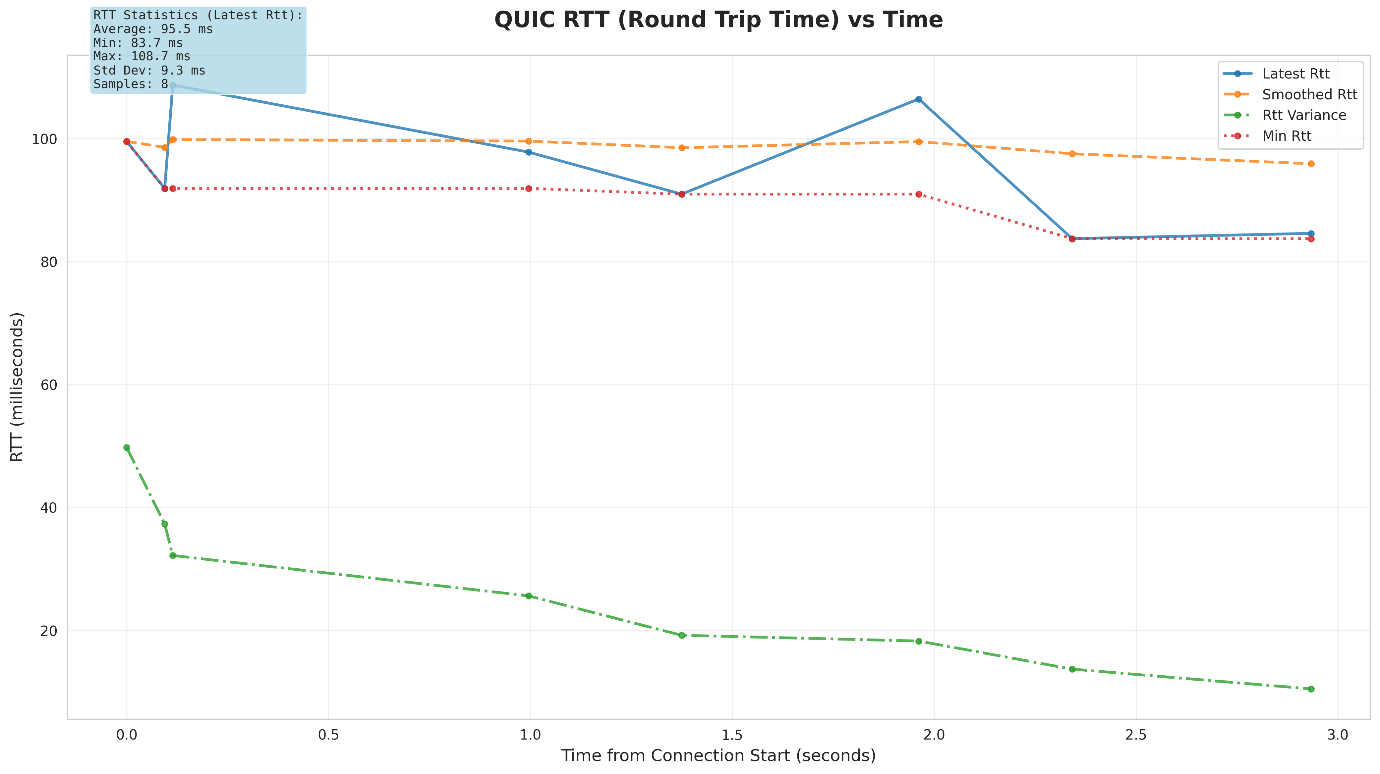
DASH به سرعت به بالاترین نرخ‌های بیت‌ریت می‌رسد و نسبت به QUIC به‌طور میانگین نرخ بیت‌ریت بالاتری دارد. اما در QUIC، نرخ بیت‌ریت به‌صورت تدریجی و با شیب ملایم‌تری افزایش می‌یابد که ممکن است ناشی از مدیریت محافظه‌کارانه ازدحام و بافر کوچک آن باشد.



نمودار میزان Buffer-Level بر زمان برای سناریو چهارم

در (شکل 4-18) نمودار تکامل سطح بافر در طول زمان برای پروتکل‌های QUIC و DASH نمایش داده شده است. در QUIC، سطح بافر به‌طور مداوم بین ۰ تا ۲ ثانیه نوسان دارد و در ابتدا شاهد تخلیه‌های زیادی و سپس پر شدن‌های سریع آن هستیم. در DASH، سطح بافر به‌طور یکنواخت از ۲ تا ۸.۱ ثانیه افزایش می‌یابد و در این سطح باقی می‌ماند. میانگین سطح بافر در DASH ۵.۰ ثانیه است.

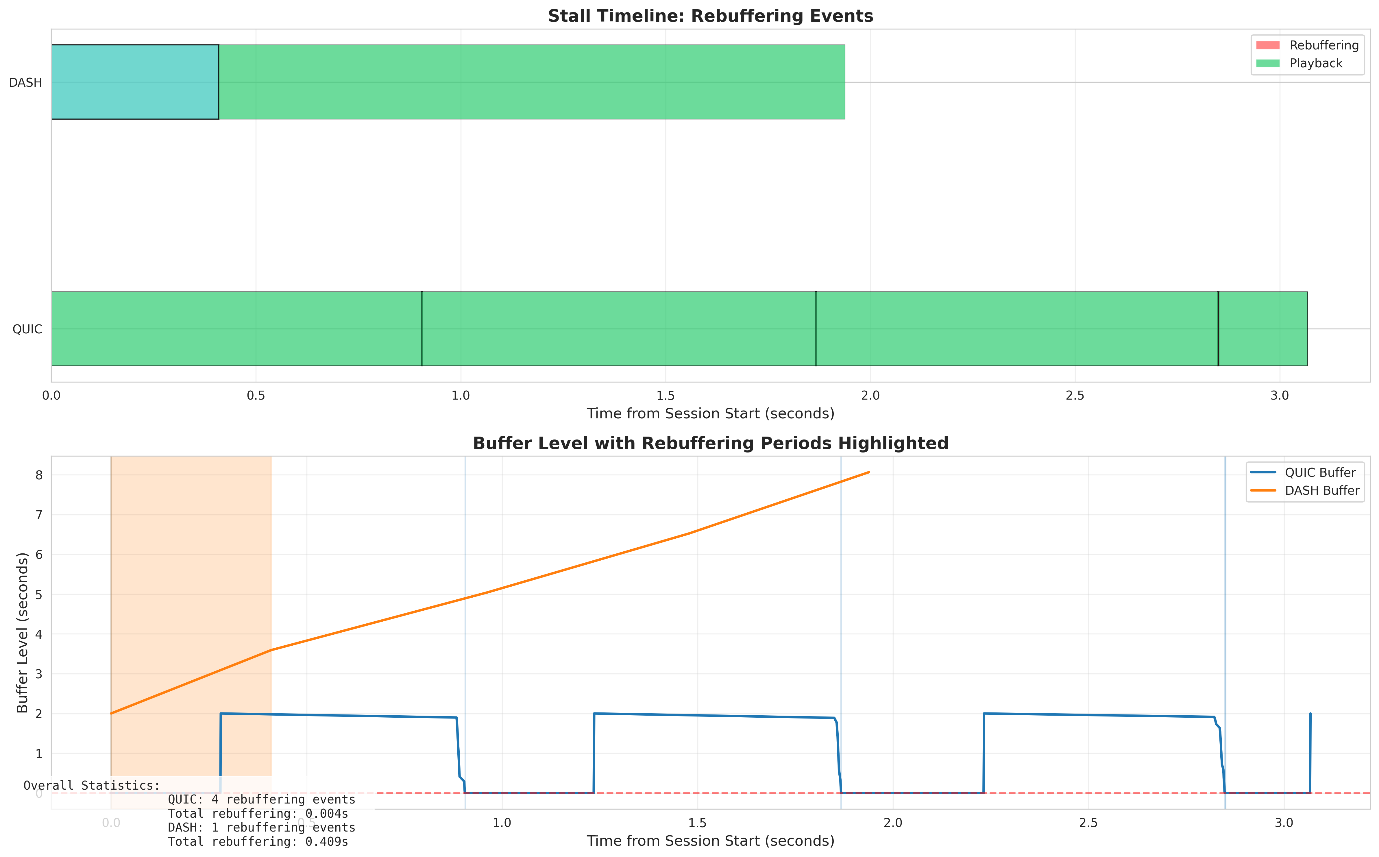
DASH با بافر بزرگ‌تر و پایدارتر برای محیط‌هایی که نیاز به پخش پایدار و بدون وقفه دارند، مناسب‌تر است. در مقابل، QUIC با بافر کوچک‌تر و مدیریت زمان‌بندی دقیق‌تر، برای محیط‌هایی که تأخیر کم و واکنش سریع مهم است، عملکرد بهتری دارد.



نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو چهارم

در (شکل 4-19) نمودار زمان رفت‌وبرگشت (RTT) در پروتکل QUIC نمایش داده شده است. در این نمودار، RTT جدیدترین (Latest RTT) به طور نوسانی بین ۸۳.۷ تا ۱۰۸.۷ میلی‌ثانیه قرار دارد. RTT صاف‌شده (Smoothed RTT) از ۹۵ میلی‌ثانیه به ۴۶ میلی‌ثانیه کاهش می‌یابد. همچنین، واریانس RTT به‌طور واضح از ۵۵ میلی‌ثانیه به حدود ۲۲ میلی‌ثانیه کاهش پیدا می‌کند.

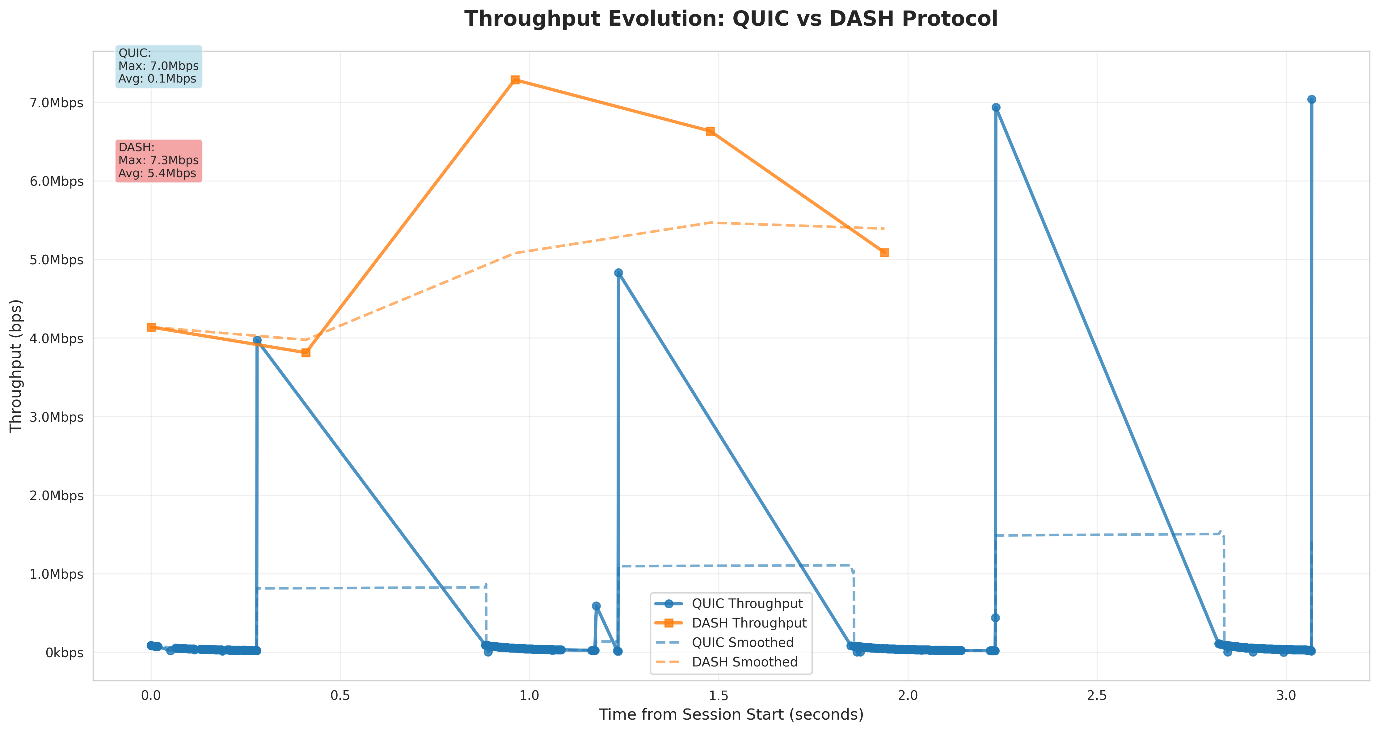
QUIC توانسته RTT را به خوبی مدیریت کند، اما به دلیل نوسانات اولیه، در ابتدا افزایش‌های زیادی را مشاهده می‌کنیم. با گذشت زمان، این نوسانات کاهش یافته و RTT صاف‌شده و واریانس RTT بهبود یافته‌اند.



نمودار میزان Stall-Timeline بر زمان برای سناریو چهارم

در (شکل 4-20) نمودار زمان‌بندی توقف‌ها و رویدادهای ریبافر برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC، ۴ رویداد ریبافر ثبت شده‌اند که مجموع زمان توقف آن‌ها تنها ۰.۰۰۴ ثانیه بوده است. در DASH، ۱ رویداد ریبافر ثبت شده که زمان توقف آن ۰.۴۰۹ ثانیه بوده است.

QUIC تقریباً بدون وقفه و به طور پیوسته عمل می‌کند. در حالی که DASH یک وقفه کوتاه در شروع دارد که ممکن است ناشی از مدیریت غیر بهینه بافر در ابتدای پخش باشد، اما پس از آن پخش به‌طور یکنواخت انجام می‌شود.



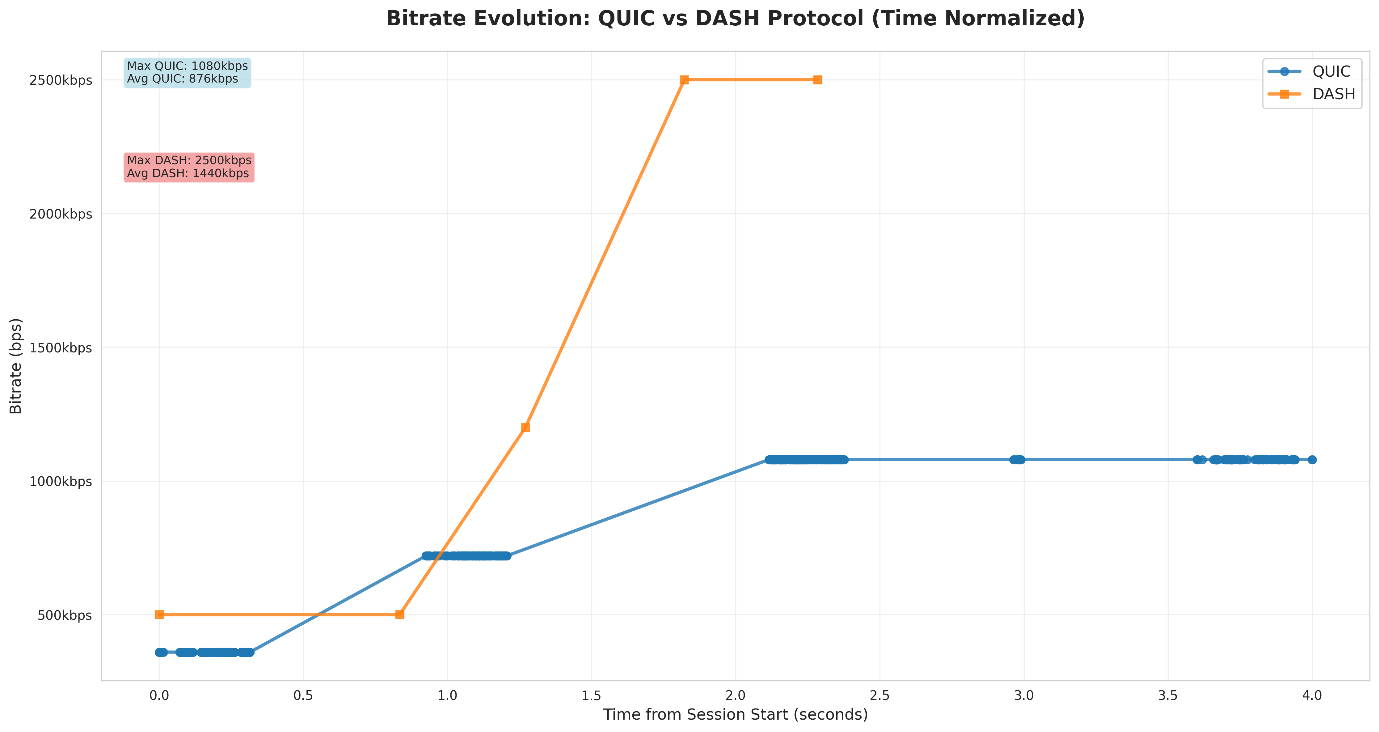
نمودار میزان Throughput بر زمان برای سناریو چهارم

در (شکل 4-21) نمودار تکامل گذردهی لینک برای پروتکل‌های QUIC و DASH نمایش داده شده است. در QUIC، گذردهی لینک در برخی لحظات به حداکثر ۷.۰ Mbps می‌رسد، اما در طول آزمایش بیشتر زمان‌ها در حدود صفر باقی می‌ماند. میانگین گذردهی QUIC ۰.۱ Mbps و نرخ صاف‌شده آن بین ۲ تا ۳ Mbps است. در DASH، گذردهی لینک به‌طور پیوسته در بازه ۵–۷.۳ Mbps تغییر می‌کند و میانگین آن ۵.۴ Mbps است.

DASH دارای گذردهی پیوسته‌تر و بهتری است و قادر است از ظرفیت لینک به‌صورت مؤثرتری استفاده کند. در حالی که QUIC، اگرچه در برخی لحظات به گذردهی‌های بالاتری می‌رسد، اما به دلیل بافر کوچک و عدم مدیریت مؤثر منابع، گذردهی به طور پیوسته پایین می‌آید.

* + 1. سناریو پنجم(موبایل 4G شلوغ)

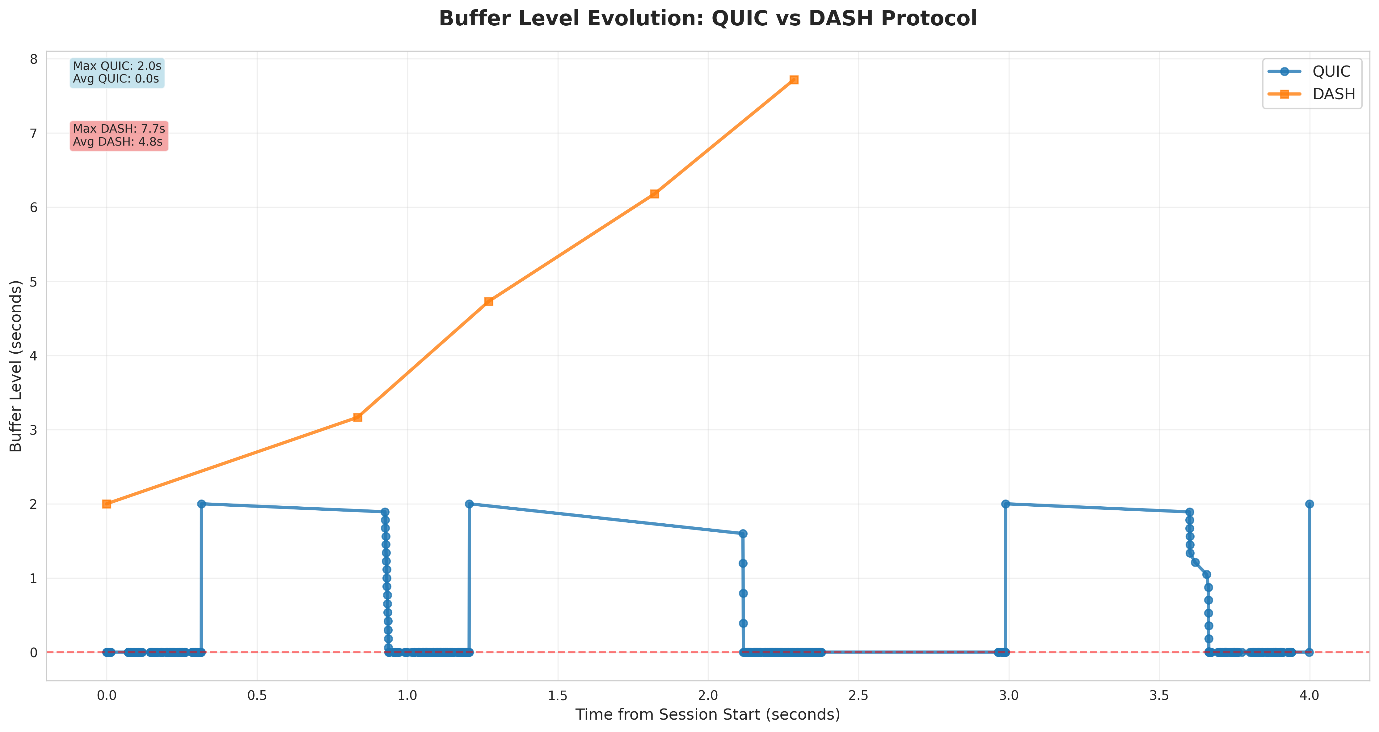
این سناریو تلاش شده است تا شبکه ایی شبیه سازی شود که کلاینت توسط شبکه تلفن همراه 4G در یک مکان به شدت شلوغ و با پوشش ناقص منطقه ایی درخواست پخش ویدیو را میدهد، بدین منظور در لینک گلوگاه مقدار 2Mbps پهنای باند و مقدار 80ms تاخیر در نظر گرفته شده است و جیتر معادل 30ms و از دستن رفتن بسته ها 3% است. در چنین سناریویی ترافیک رقابتی به صورت بدترین حالت تنظیم شده، بدین معنا که در شبکه شخص دیگری هم در حال درخواست و یا دریافت پخش ویدیوی دیگر به صورت UDP است.



نمودار میزان Bitrate بر زمان برای سناریو پنجم

در (شکل 4-22) نمودار تکامل نرخ بیت‌ریت در طول زمان برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC، نرخ بیت‌ریت از ۵۰۰ kbps شروع می‌شود و پس از حدود ۱.۵ ثانیه به ۱.۰۸ Mbps می‌رسد. حداکثر نرخ بیت‌ریت برابر با ۱۰۸۰ kbps و میانگین آن ۸۷۶ kbps است. در DASH، نرخ بیت‌ریت به سرعت به ۲.۵ Mbps می‌رسد و سپس تقریباً ثابت می‌ماند. حداکثر نرخ بیت‌ریت برابر با ۲۵۰۰ kbps و میانگین آن ۱۴۴۰ kbps است.

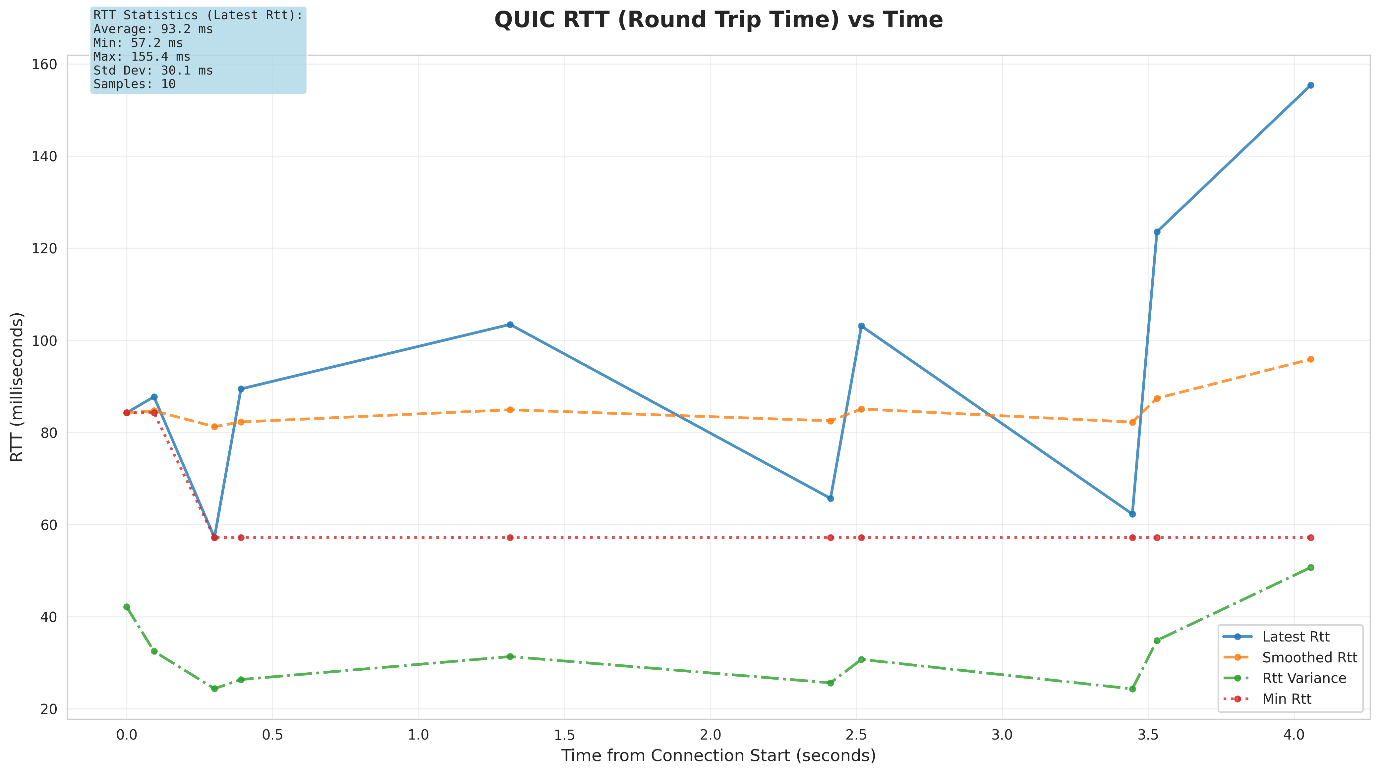
DASH به سرعت به بالاترین نرخ‌های بیت‌ریت می‌رسد و نسبت به QUIC به‌طور میانگین نرخ بیت‌ریت بالاتری دارد. اما در QUIC، نرخ بیت‌ریت به‌صورت تدریجی و با شیب ملایم‌تری افزایش می‌یابد که ممکن است ناشی از مدیریت محافظه‌کارانه ازدحام و بافر کوچک آن باشد.



نمودار میزان Buffer-Level بر زمان برای سناریو پنجم

در (شکل 4-23) نمودار تکامل سطح بافر در طول زمان برای پروتکل‌های QUIC و DASH نمایش داده شده است. در QUIC، سطح بافر به‌طور مداوم بین ۰ تا ۲ ثانیه نوسان دارد و در ابتدا شاهد تخلیه‌های زیادی و سپس پر شدن‌های سریع آن هستیم. در DASH، سطح بافر به‌طور یکنواخت از ۲ تا ۷.۷ ثانیه افزایش می‌یابد و در این سطح باقی می‌ماند. میانگین سطح بافر در DASH ۴.۸ ثانیه است.

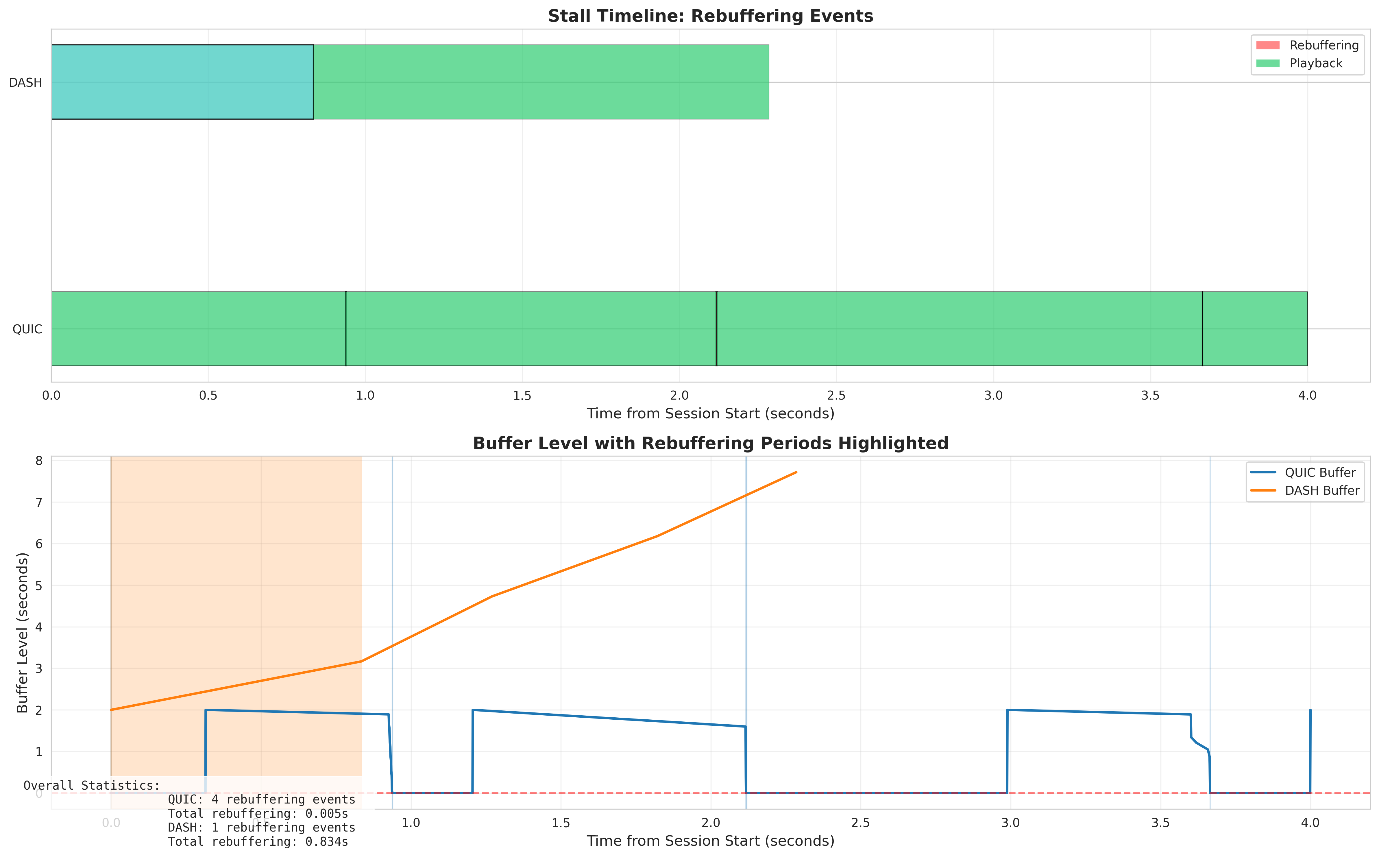
DASH با بافر بزرگ‌تر و پایدارتر برای محیط‌هایی که نیاز به پخش پایدار و بدون وقفه دارند، مناسب‌تر است. در مقابل، QUIC با بافر کوچک‌تر و مدیریت زمان‌بندی دقیق‌تر، برای محیط‌هایی که تأخیر کم و واکنش سریع مهم است، عملکرد بهتری دارد.



نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو پنجم

در (شکل4-24 ) نمودار زمان رفت‌وبرگشت (RTT) در پروتکل QUIC نمایش داده شده است. در این نمودار، RTT جدیدترین (Latest RTT) به طور نوسانی بین ۵۷.۲ تا ۱۵۵.۴ میلی‌ثانیه قرار دارد. RTT صاف‌شده (Smoothed RTT) از ۹۳ میلی‌ثانیه به ۶۲ میلی‌ثانیه کاهش می‌یابد. همچنین، واریانس RTT به‌طور واضح از ۳۰ میلی‌ثانیه به حدود ۱۰ میلی‌ثانیه کاهش پیدا می‌کند.

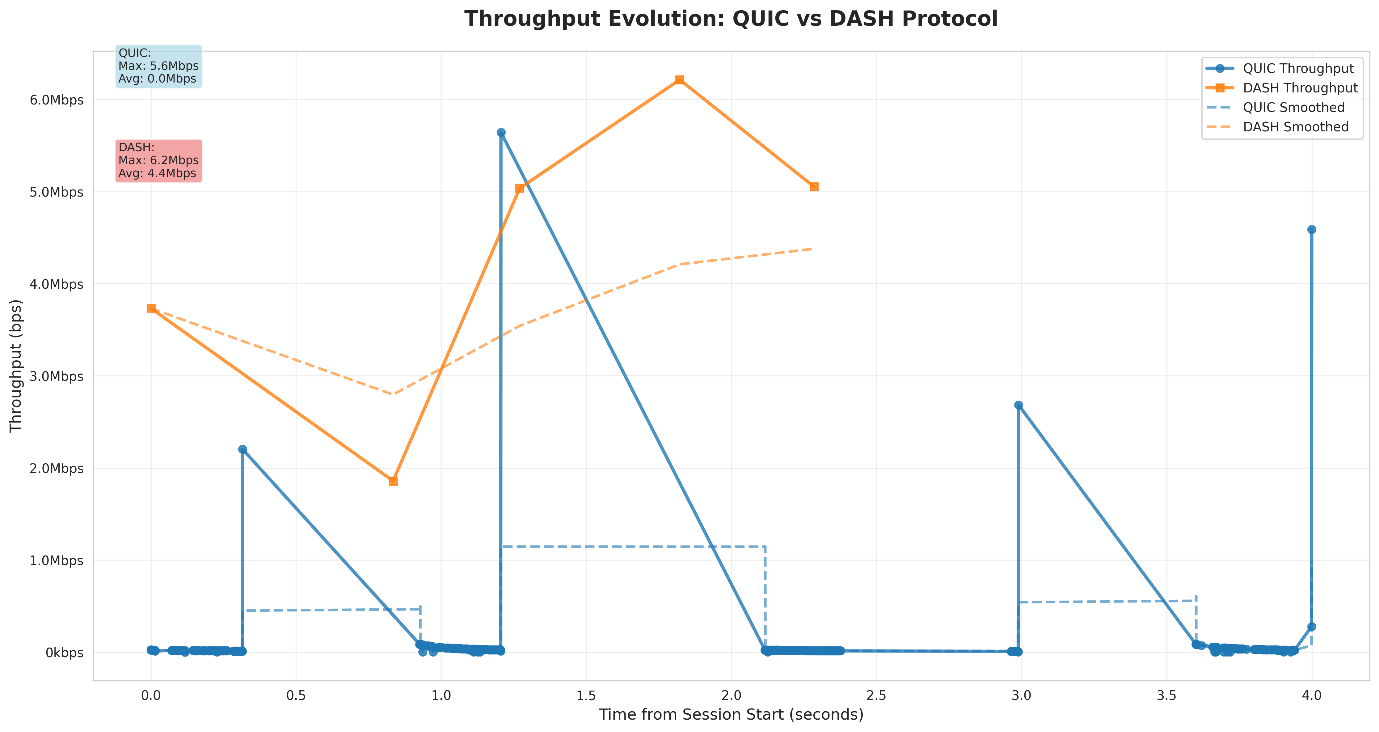
QUIC توانسته RTT را به خوبی مدیریت کند، اما به دلیل نوسانات اولیه، در ابتدا افزایش‌های زیادی را مشاهده می‌کنیم. با گذشت زمان، این نوسانات کاهش یافته و RTT صاف‌شده و واریانس RTT بهبود یافته‌اند.



نمودار میزان Stall-Timeline بر زمان برای سناریو پنجم

در (شکل 4-25) نمودار زمان‌بندی توقف‌ها و رویدادهای ریبافر برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC، ۴ رویداد ریبافر ثبت شده‌اند که مجموع زمان توقف آن‌ها تنها ۰.۰۰۵ ثانیه بوده است. در DASH، ۱ رویداد ریبافر ثبت شده که زمان توقف آن ۰.۸۳۴ ثانیه بوده است.

QUIC تقریباً بدون وقفه و به طور پیوسته عمل می‌کند. در حالی که DASH یک وقفه کوتاه در شروع دارد که ممکن است ناشی از مدیریت غیر بهینه بافر در ابتدای پخش باشد، اما پس از آن پخش به‌طور یکنواخت انجام می‌شود.



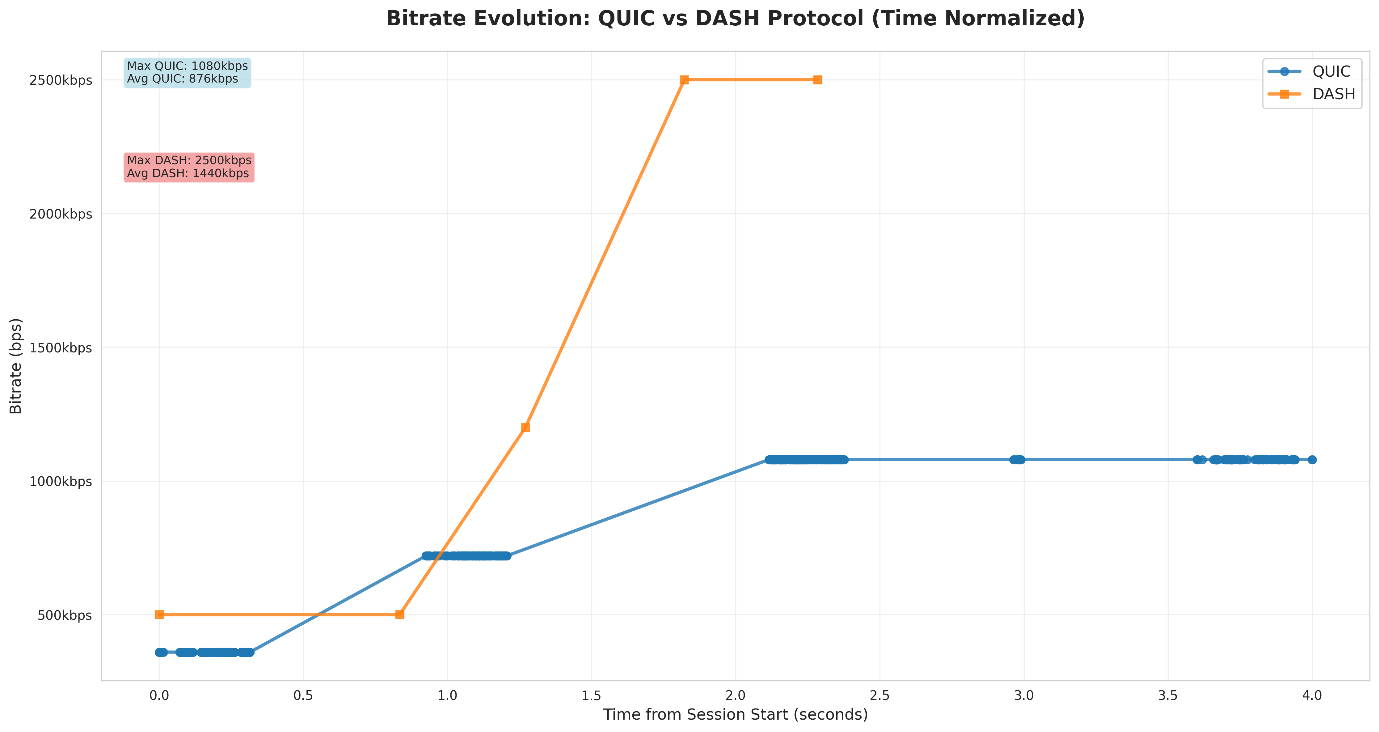
نمودار میزان Throughput بر زمان برای سناریو پنجم

در (شکل 4-26) نمودار تکامل گذردهی لینک برای پروتکل‌های QUIC و DASH نمایش داده شده است. در QUIC، گذردهی لینک در برخی لحظات به حداکثر ۵.۶ Mbps می‌رسد، اما در طول آزمایش بیشتر زمان‌ها در حدود صفر باقی می‌ماند. میانگین گذردهی QUIC ۰.۰ Mbps و نرخ صاف‌شده آن بین ۲ تا ۳ Mbps است. در DASH، گذردهی لینک به‌طور پیوسته در بازه ۴–۶.۲ Mbps تغییر می‌کند و میانگین آن ۴.۴ Mbps است.

DASH دارای گذردهی پیوسته‌تر و بهتری است و قادر است از ظرفیت لینک به‌صورت مؤثرتری استفاده کند. در حالی که QUIC، اگرچه در برخی لحظات به گذردهی‌های بالاتری می‌رسد، اما به دلیل بافر کوچک و عدم مدیریت مؤثر منابع، گذردهی به طور پیوسته پایین می‌آید.

* + 1. سناریو ششم(موبایل 3G نسل قدیمی)

این سناریو تلاش شده است تا شبکه ایی شبیه سازی شود که کلاینت توسط شبکه تلفن همراه 3G در یک منطقه با آنتن های نسل قدیمی، درخواست پخش ویدیو را میدهد، بدین منظور در لینک گلوگاه مقدار 2Mbps پهنای باند و مقدار 80ms تاخیر در نظر گرفته شده است و جیتر معادل 30ms و از دستن رفتن بسته ها 3% است. در چنین سناریویی ترافیک رقابتی وجود ندارد.

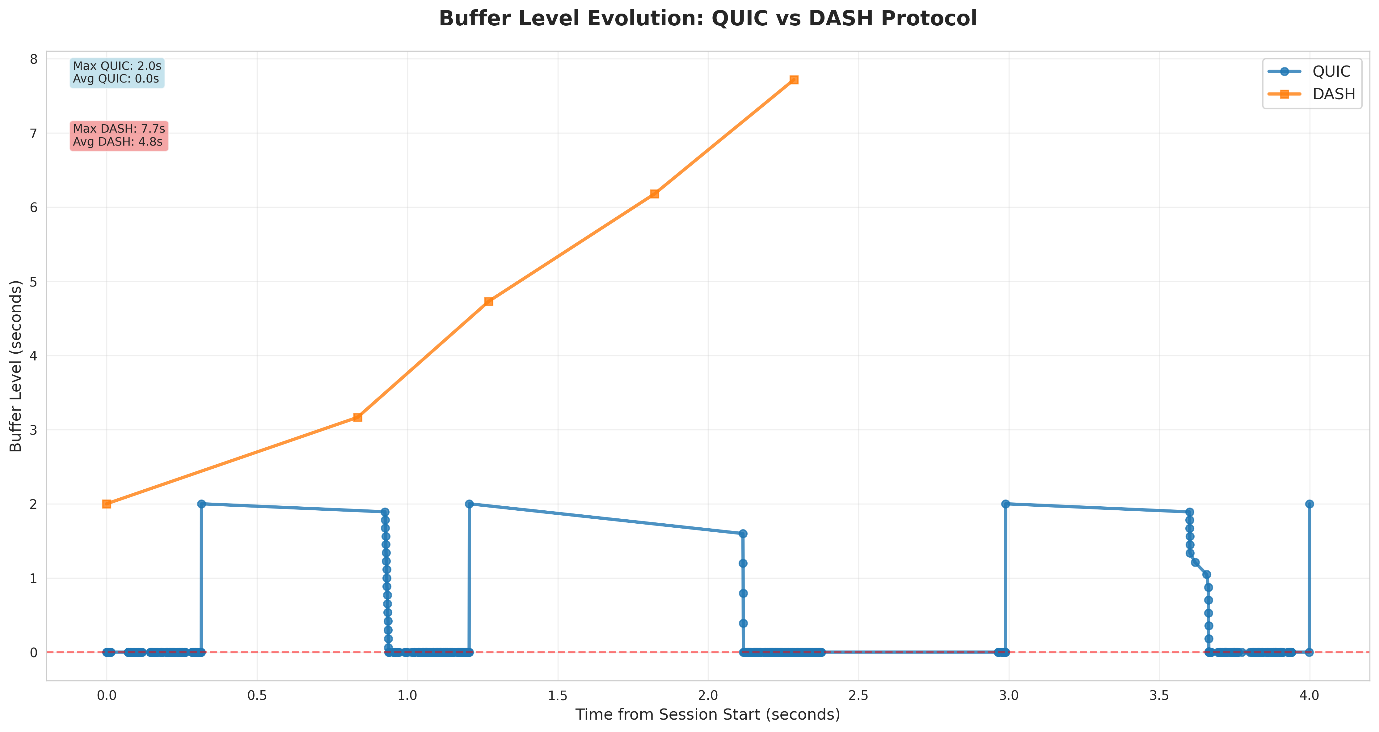


نمودار میزان Bitrate بر زمان برای سناریو ششم

در (شکل 4-27) نمودار تکامل نرخ بیت‌ریت در طول زمان برای پروتکل‌های QUIC و DASH را نشان می‌دهد. در QUIC، نرخ بیت‌ریت از ۵۰۰ kbps شروع می‌شود و پس از حدود ۱.۵ ثانیه به ۱.۰۸ Mbps می‌رسد. حداکثر نرخ بیت‌ریت برابر با ۱۰۸۰ kbps و میانگین آن ۸۷۹ kbps است. در DASH، نرخ بیت‌ریت به سرعت به ۲.۵ Mbps می‌رسد و در آن نقطه ثابت می‌ماند. حداکثر نرخ بیت‌ریت برابر با ۲۵۰۰ kbps و میانگین آن ۱۴۴۰ kbps است.

نتیجه‌گیری: DASH نرخ بیت‌ریت بالاتری نسبت به QUIC دارد و به سرعت به بالاترین مقدار خود می‌رسد، در حالی که QUIC افزایش تدریجی‌تری دارد.

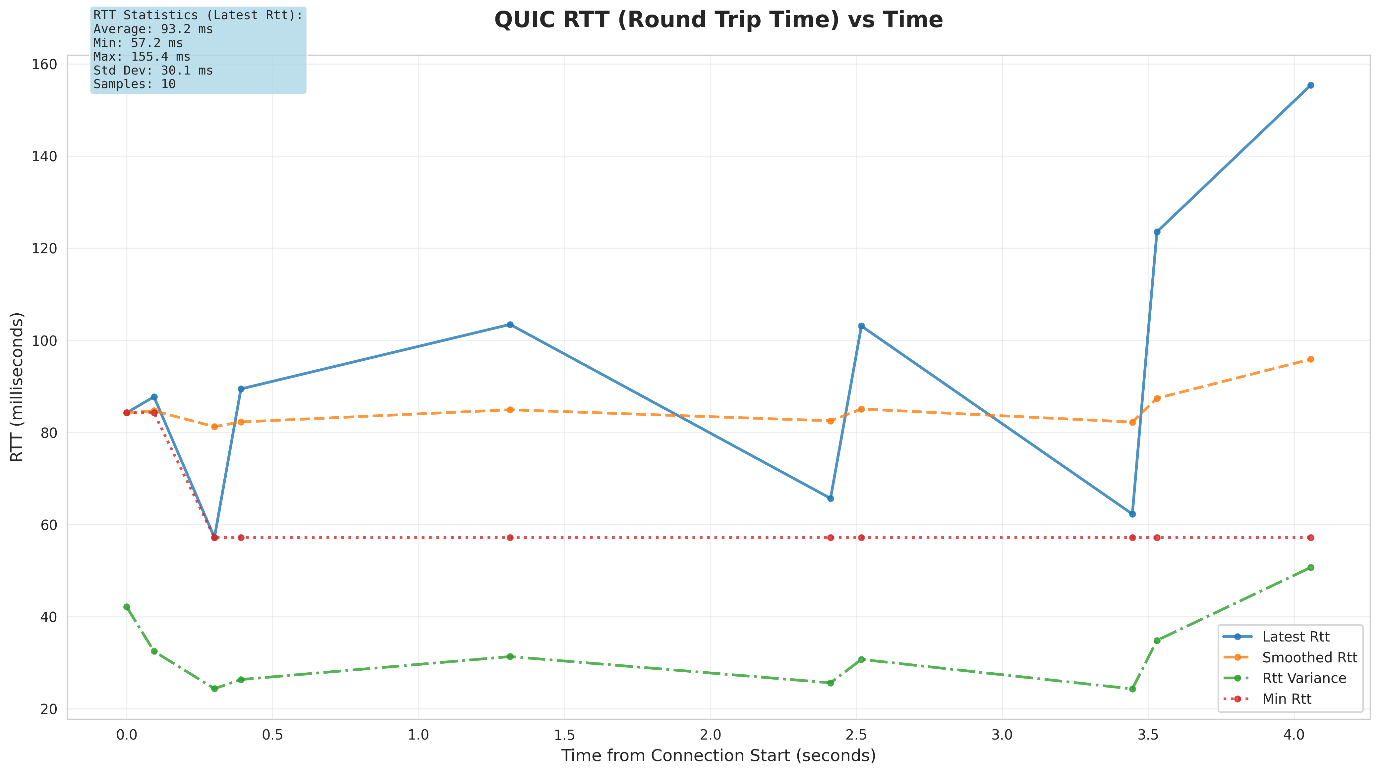
DASH به سرعت به بالاترین نرخ‌های بیت‌ریت می‌رسد و نسبت به QUIC به‌طور میانگین نرخ بیت‌ریت بالاتری دارد. اما در QUIC، نرخ بیت‌ریت به‌صورت تدریجی و با شیب ملایم‌تری افزایش می‌یابد که ممکن است ناشی از مدیریت محافظه‌کارانه ازدحام و بافر کوچک آن باشد.



نمودار میزان Buffer-Level بر زمان برای سناریو ششم

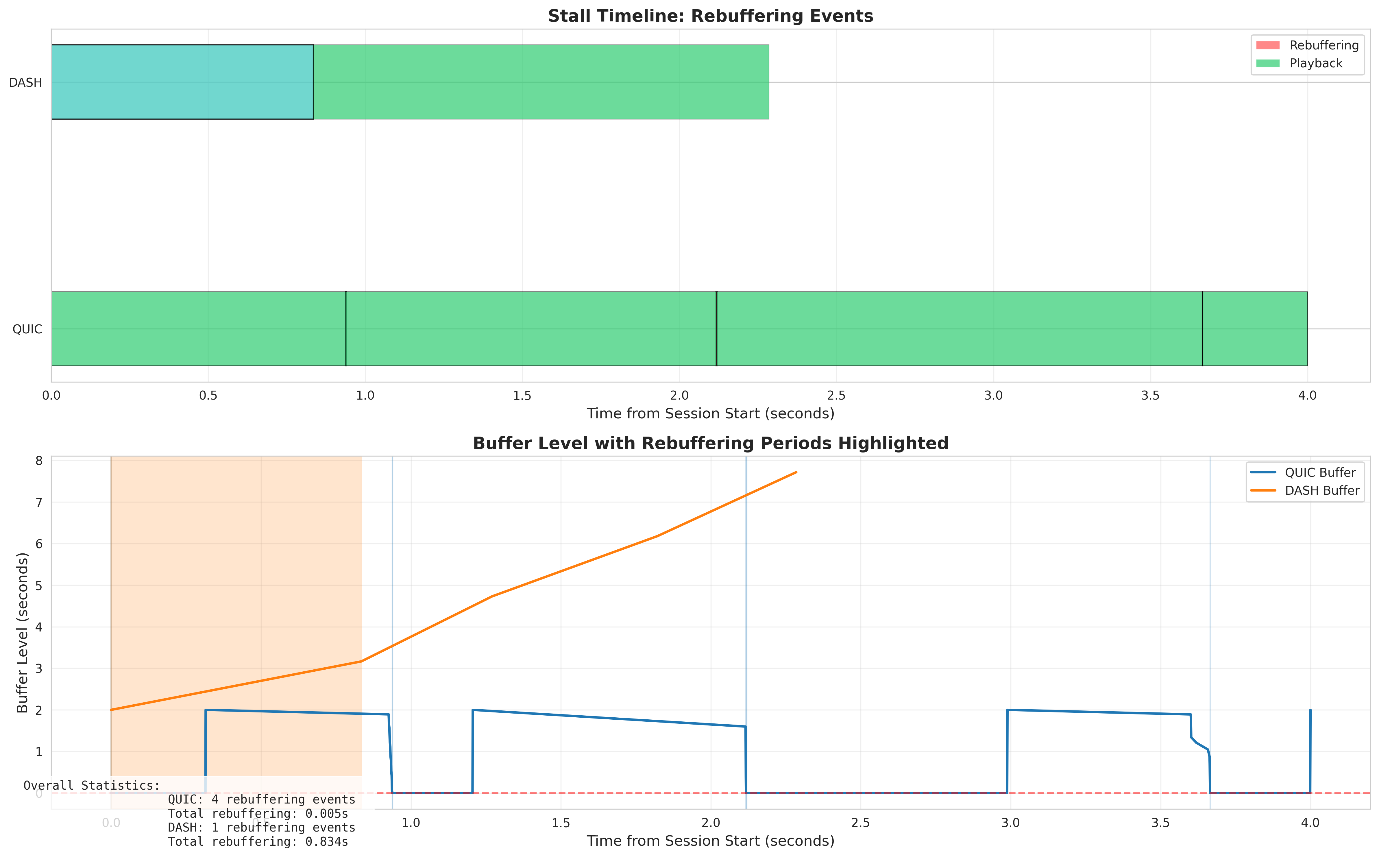
در (شکل 4-28) تکامل سطح بافر برای پروتکل‌های QUIC و DASH به تصویر کشیده شده است. در QUIC، سطح بافر به سرعت کاهش می‌یابد و سپس دوباره بازیابی می‌شود، با حداکثر سطح بافر ۲.۰ ثانیه و میانگین ۰.۱ ثانیه. در DASH، سطح بافر به تدریج افزایش یافته و در سطح بالاتری ثابت می‌ماند، به‌طوری‌که حداکثر سطح بافر ۸.۰ ثانیه و میانگین آن ۵.۱ ثانیه است.

به نظر میرسد که DASH بافر بزرگتری را حفظ می‌کند و به‌طور پیوسته سطح آن را افزایش می‌دهد، در حالی که QUIC بافر کمتری دارد و سطح آن بیشتر نوسان دارد.



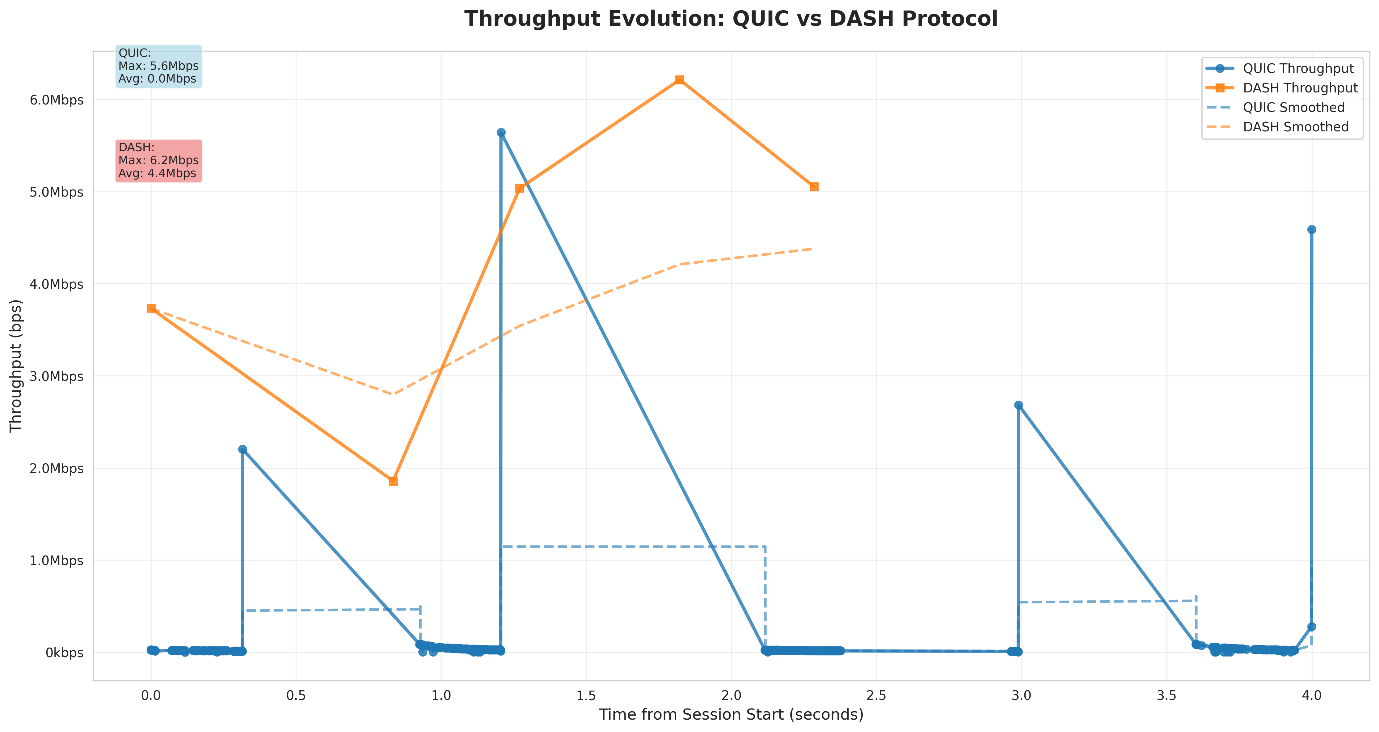
نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو ششم

در (شکل4-29 ) زمان دور رفت و برگشت (RTT) QUIC در طول زمان نشان داده شده است. آخرین مقدار RTTبین ۶۰ ms و ۱۰۰ ms نوسان دارد و حداکثر مقدار آن ۱۰۸.۷ ms است. میانگین RTT برابر با ۸۷.۵ ms است.RTT صاف‌شده (Smoothed RTT) به تدریج کاهش می‌یابد و نشان‌دهنده پایداری کیفیت ارتباط است. تغییرات RTT نیز در ابتدا بیشتر است، اما در نهایت به یک سطح ثابت می‌رسد.  
زمان دور رفت و برگشت QUIC در ابتدا نوسانات زیادی دارد، اما با گذر زمان به یک مقدار ثابت و پایین‌تر می‌رسد، که این نشانه‌ای از بهبود پایداری ارتباط است.



نمودار میزان Stall-Timeline بر زمان برای سناریو ششم

در (شکل 4-30) زمان‌بندی توقف‌ها و سطح بافر با دوره‌های بافر شدن برجسته شده برای پروتکل‌های QUIC و DASH نمایش داده شده است. در QUIC، تعداد ۴ رویداد بافر شدن رخ می‌دهد و زمان کل بافر شدن برابر با ۰.۰۰۵ ثانیه است. در DASH، یک رویداد بافر شدن با زمان کل ۰.۳۹۷ ثانیه اتفاق می‌افتد. سطح بافر در DASH به تدریج افزایش می‌یابد و به حدود ۶ ثانیه می‌رسد، در حالی که در QUIC سطح بافر به سرعت کاهش یافته و پس از آن دوباره بازیابی می‌شود.

DASH به‌طور قابل توجهی کمتر بافر شدن را تجربه می‌کند و سطح بافر بالاتری نسبت به QUIC دارد، که نشان‌دهنده یک استراتژی مدیریت بافر بهتر در DASH است. 

نمودار میزان Throughput بر زمان برای سناریو ششم

در (شکل 4-31) تکامل سرعت انتقال داده برای پروتکل‌های QUIC و DASH نمایش داده شده است.

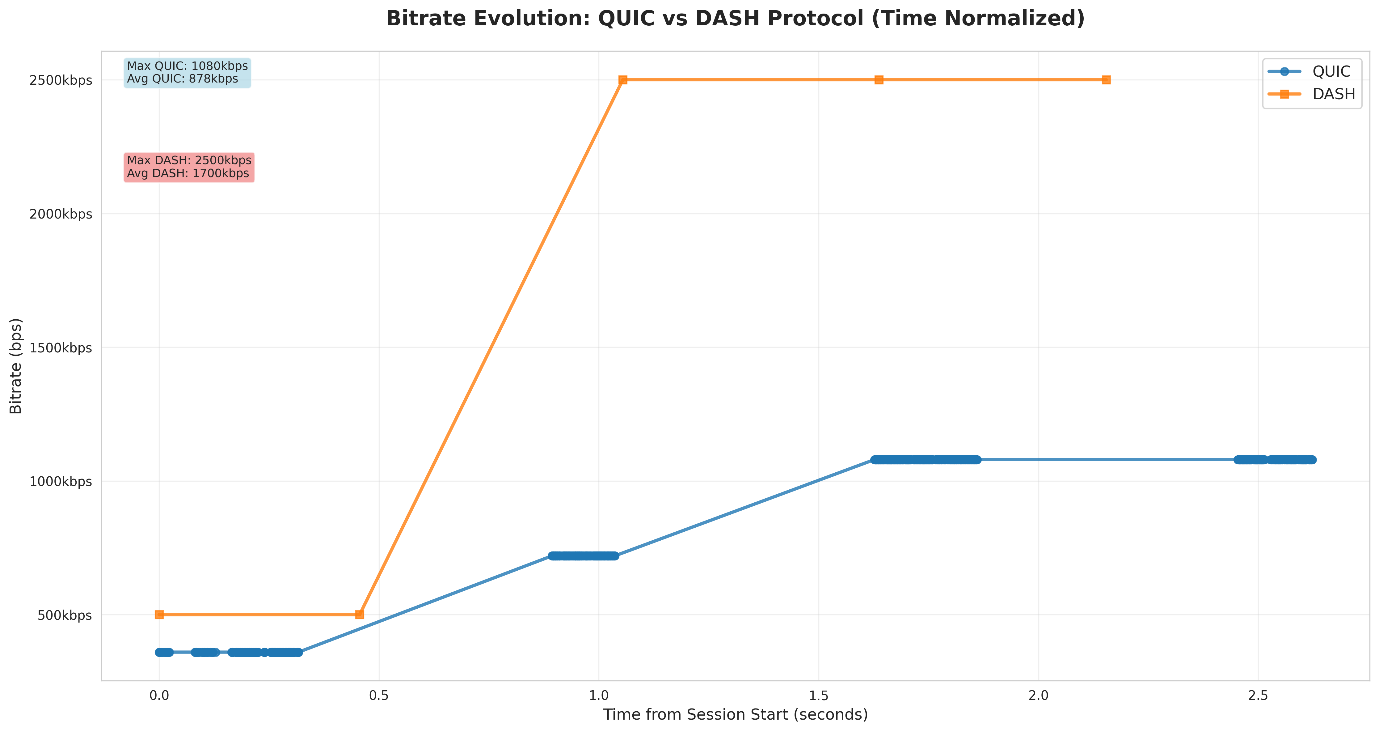
در QUIC، سرعت انتقال داده به‌طور ناگهانی افزایش می‌یابد و به ۵.۶ Mbps می‌رسد. حداکثر سرعت انتقال داده برابر با ۹.۷ Mbps و میانگین آن ۰.۱ Mbps است.

در DASH، سرعت انتقال داده نیز افزایش می‌یابد و به ۸.۱ Mbps می‌رسد. حداکثر سرعت انتقال داده برابر با ۸.۱ Mbps و میانگین آن ۵.۳ Mbps است.

DASH در بیشتر زمان‌ها سرعت انتقال داده بالاتری نسبت به QUIC دارد و این نشان‌دهنده بهره‌برداری بهتر از شبکه توسط DASH است. همچنین، QUIC در ابتدا دارای نوسانات شدید در سرعت است، اما بعد از مدتی به مقدار پایدار و پایین‌تری می‌رسد.

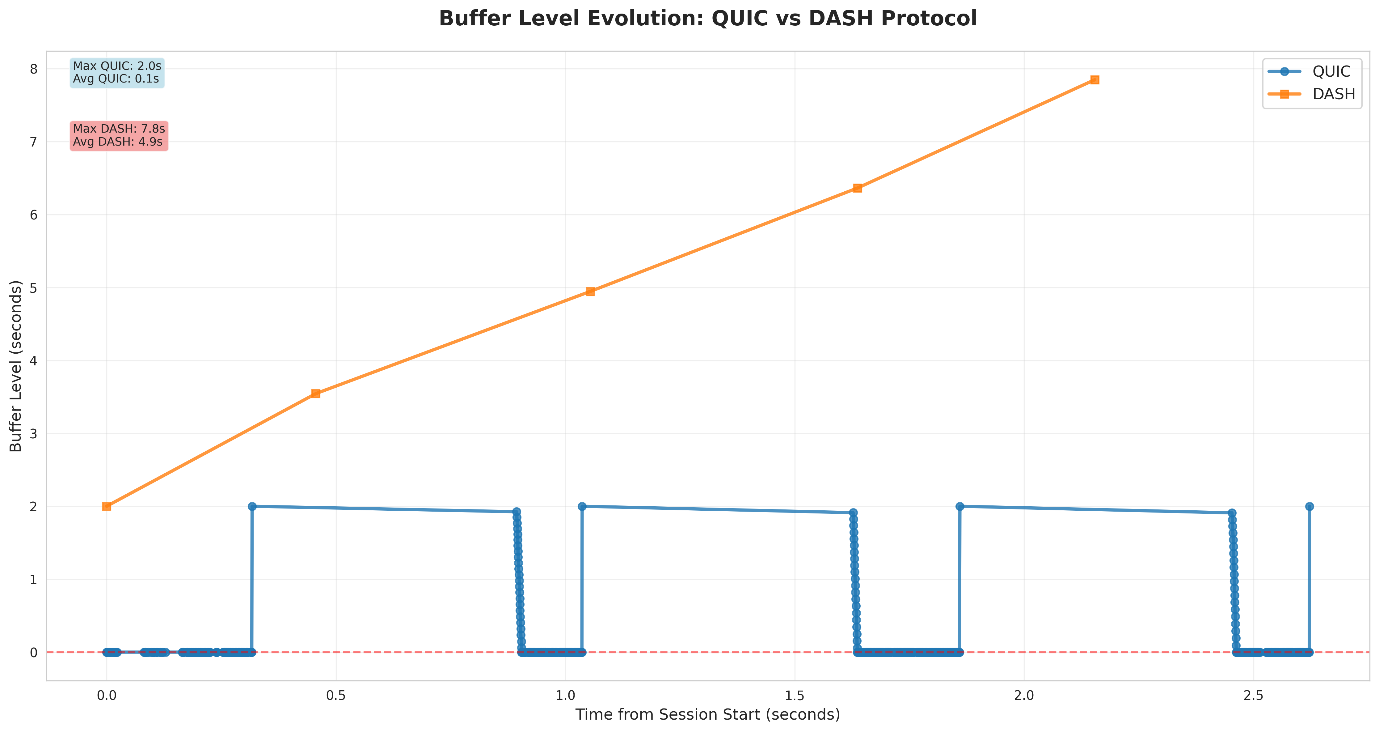
* + 1. سناریو هفتم(شبکه ماهواره ایی)

این سناریو تلاش شده است تا یک شبکه ماهواره ‌ایی شیبه سازی شود که کلاینت در آن درخواست پخش ویدیو را میدهد، بدین منظور در لینک گلوگاه مقدار 10Mbps پهنای باند و مقدار 80ms تاخیر در نظر گرفته شده است و جیتر معادل 10ms و از دستن رفتن بسته ها 0.1% است. در چنین سناریویی ترافیک رقابتی وجود ندارد.



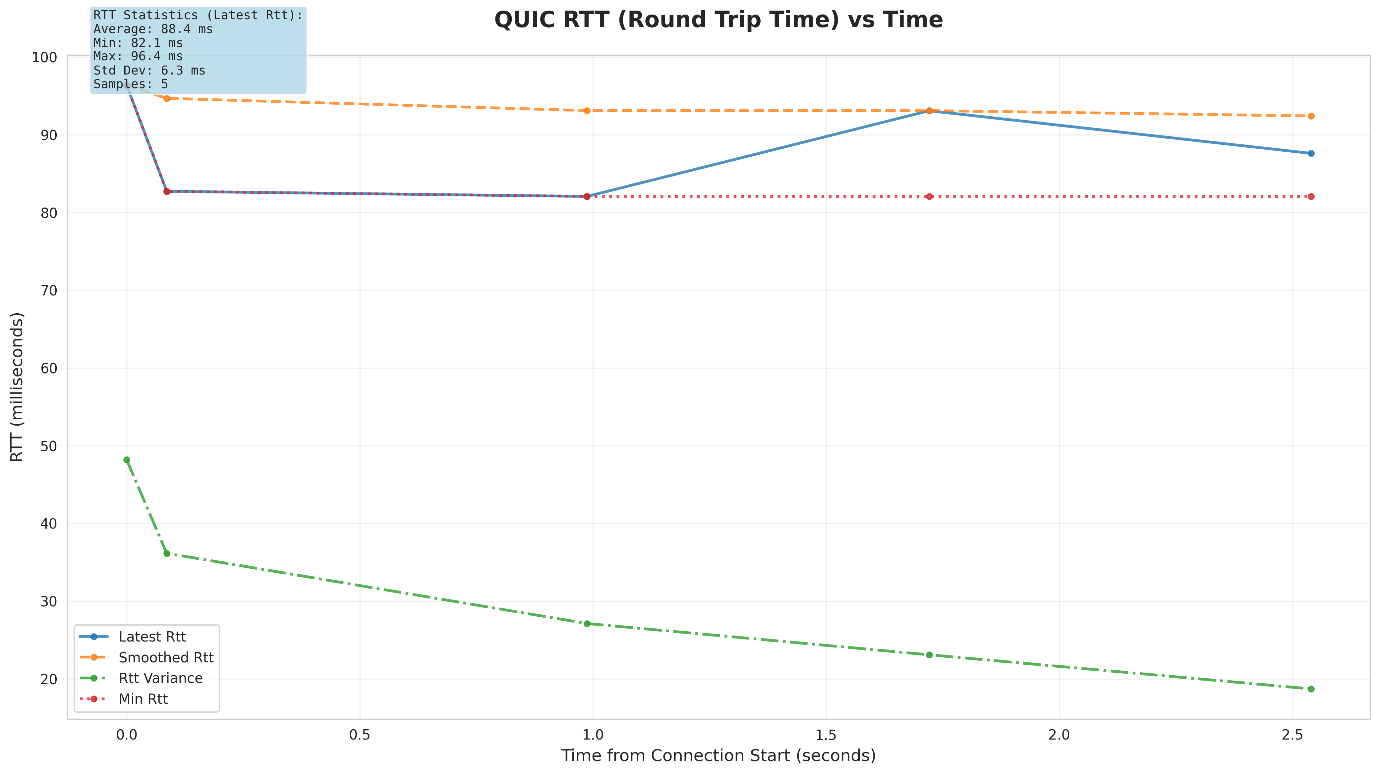
نمودار میزان Bitrate بر زمان برای سناریو هفتم

در (شکل 4-32) نمودار تکامل نرخ بیت‌ریت در طول زمان برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC نرخ بیت‌ریت از ۳۵۰ kbps شروع می‌شود و پس از حدود ۱.۵ ثانیه به ۱.۰۸ Mbps می‌رسد. حداکثر نرخ بیت‌ریت برابر با ۱۰۸۰ kbps و میانگین آن ۸۷۸ kbps است. در DASH نرخ بیت‌ریت به سرعت به ۲.۵ Mbps می‌رسد و در آن نقطه ثابت می‌ماند. حداکثر نرخ بیت‌ریت برابر با ۲۵۰۰ kbps و میانگین آن ۱۷۰۰ kbps است. DASH به سرعت به بالاترین نرخ‌های بیت‌ریت می‌رسد و به‌طور میانگین نرخ بیت‌ریت بالاتری نسبت به QUIC دارد. QUIC به‌طور تدریجی و با شیب ملایم‌تری به افزایش بیت‌ریت می‌پردازد که ممکن است به دلیل مدیریت محافظه‌کارانه ازدحام و بافر کوچک آن باشد.



نمودار میزان Buffer-Level بر زمان برای سناریو هفتم

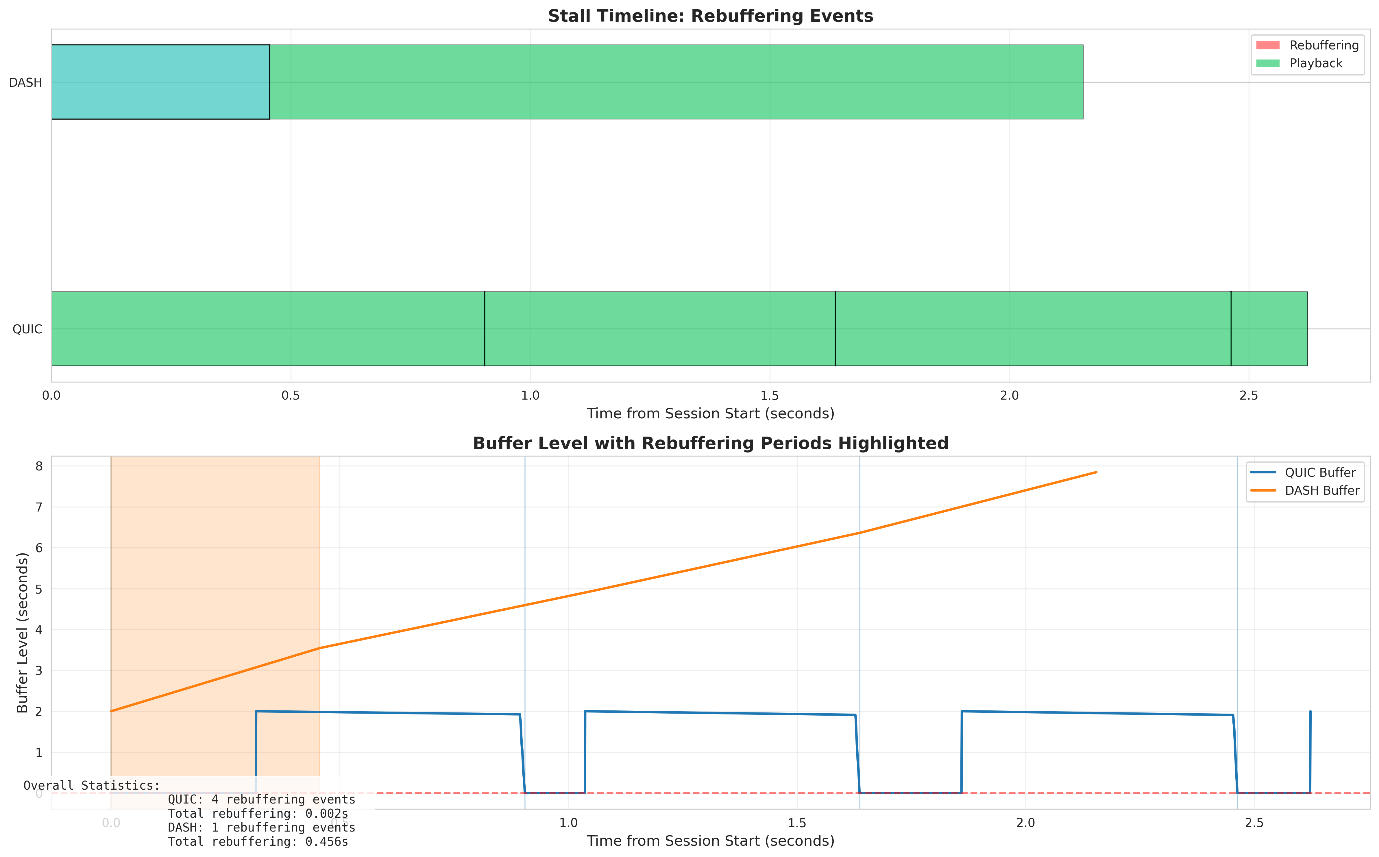
در (شکل 4-33) نمودار تکامل سطح بافر در طول زمان برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC سطح بافر از حدود ۲ ثانیه شروع می‌شود و به تدریج افزایش می‌یابد تا به حداکثر ۲ ثانیه می‌رسد. میانگین سطح بافر برابر با ۰.۱ ثانیه است. در DASH سطح بافر به سرعت افزایش می‌یابد و به حدود ۷.۸ ثانیه می‌رسد. میانگین سطح بافر برابر با ۴.۹ ثانیه است.  
DASH سطح بافر بالاتری نسبت به QUIC دارد و به سرعت افزایش می‌یابد، در حالی که در QUIC سطح بافر به‌طور تدریجی و با شیب ملایم‌تر افزایش پیدا می‌کند.



نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو هفتم

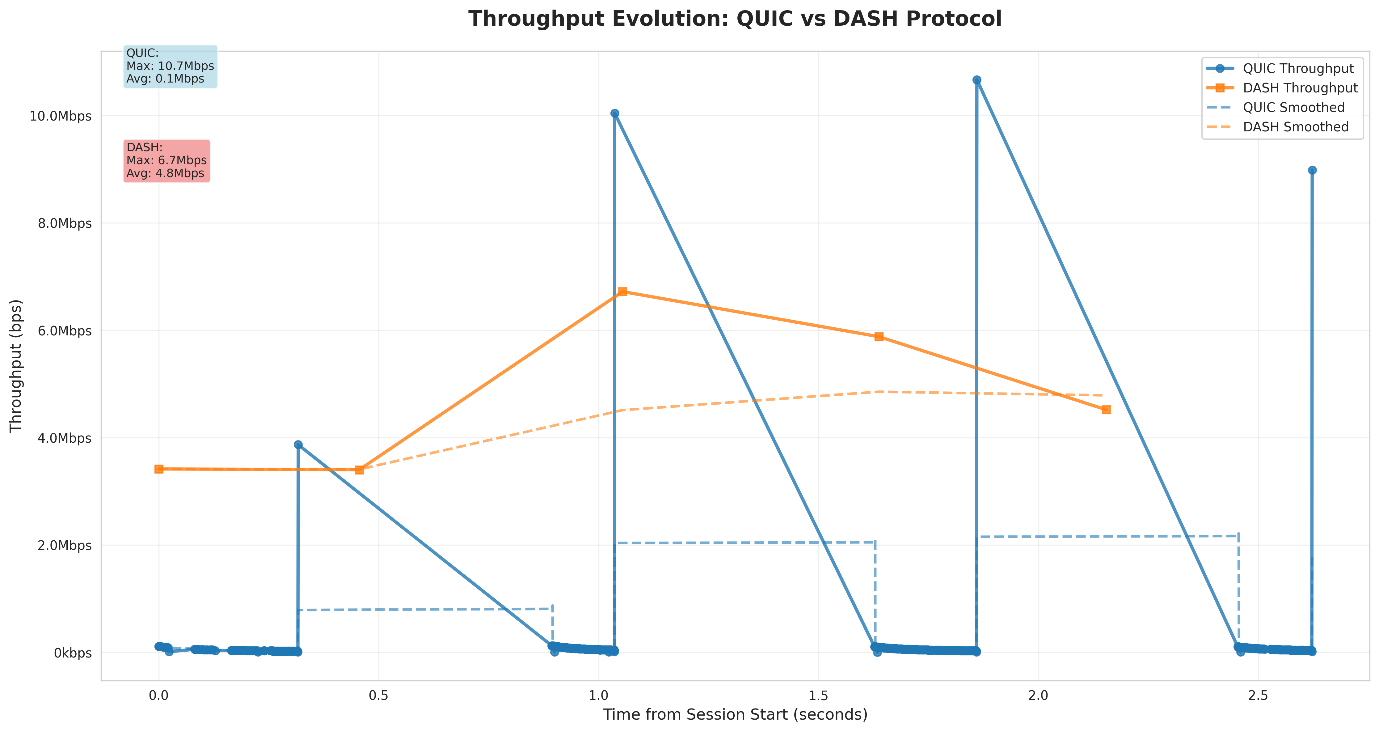
در (شکل4-34 ) نمودار زمان رفت و برگشت (RTT) در طول زمان برای پروتکل QUIC نشان داده شده است. در این نمودار، آخرین RTT در حدود ۸۷.۵ میلی‌ثانیه با حداقل ۷۲.۸ میلی‌ثانیه و حداکثر ۹۶.۴ میلی‌ثانیه قرار دارد. متوسط RTT در حدود ۸۸.۴ میلی‌ثانیه است. همواری RTT به صورت تدریجی کاهش می‌یابد و تنوع RTT به‌طور ثابت پایین می‌آید.

QUIC دارای زمان‌های رفت و برگشت نسبتاً پایدار است که در ابتدا یک افزایش کوتاه مدت دارد، ولی پس از آن به ثبات می‌رسد.



نمودار میزان Stall-Timeline بر زمان برای سناریو هفتم

در (شکل 4-35) نمودار زمان توقف برای پروتکل‌های QUIC و DASH نشان داده شده است. در این نمودار، DASH دارای یک وقفه کوتاه در شروع پخش است که منجر به توقف پخش و سپس ادامه آن می‌شود. در QUIC نیز یک وقفه مشاهده می‌شود، اما این وقفه طولانی‌تر است.

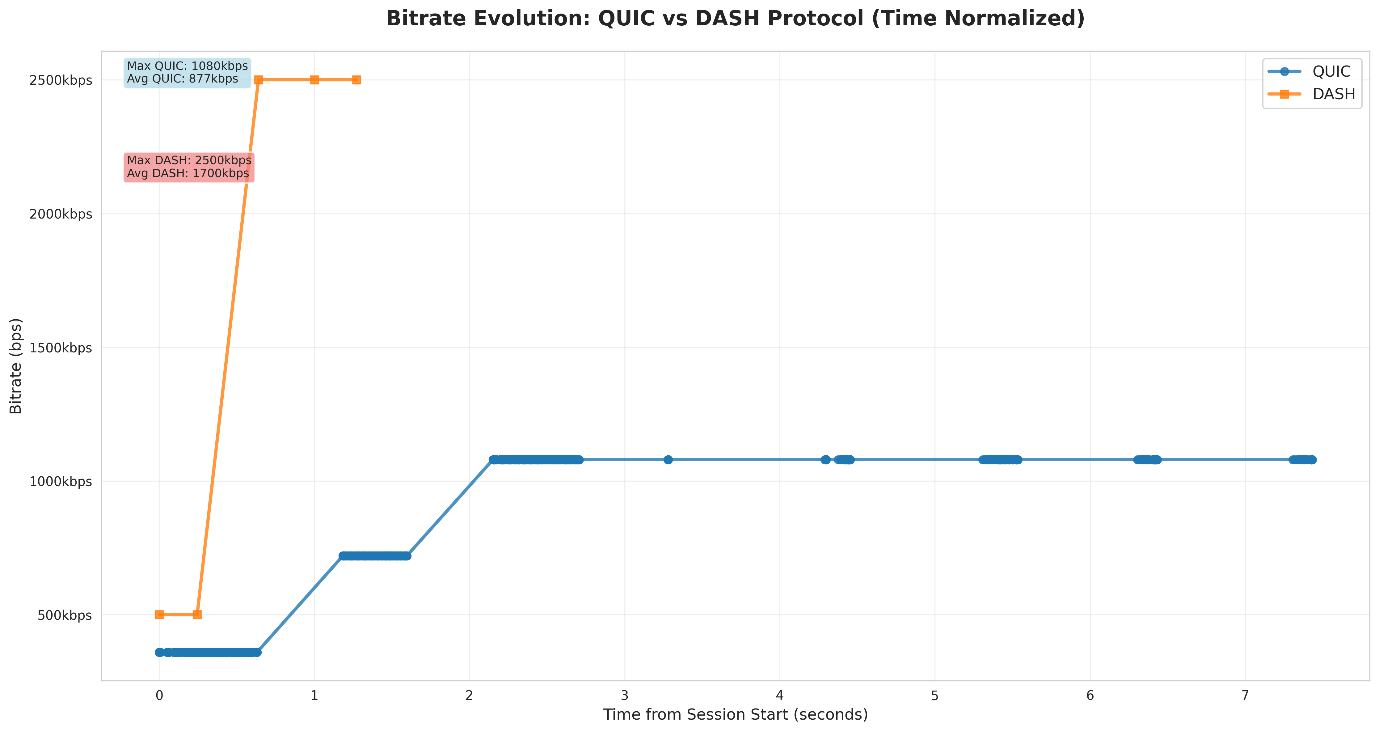
DASH با یک وقفه کوتاه مواجه است که به طور سریع ادامه پیدا می‌کند، در حالی که QUIC با وقفه‌ای طولانی‌تر، مواجه است و به آرامی ادامه می‌یابد. 

نمودار میزان Throughput بر زمان برای سناریو هفتم

در (شکل 4-36) نمودار تکامل توان در طول زمان برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC توان از ۵.۶ Mbps شروع می‌شود و در نهایت تا ۱۰.۷ Mbps افزایش می‌یابد. در DASH توان از ۶.۷ Mbps شروع می‌شود و به ۸.۱ Mbps می‌رسد. میانگین توان در QUIC بسیار پایین‌تر از DASH است.  
DASH دارای توان ثابت و بالاتر از QUIC است، که نشان‌دهنده قابلیت بهینه‌تر DASH در حفظ توان بالا برای مدت زمان بیشتر می‌باشد.

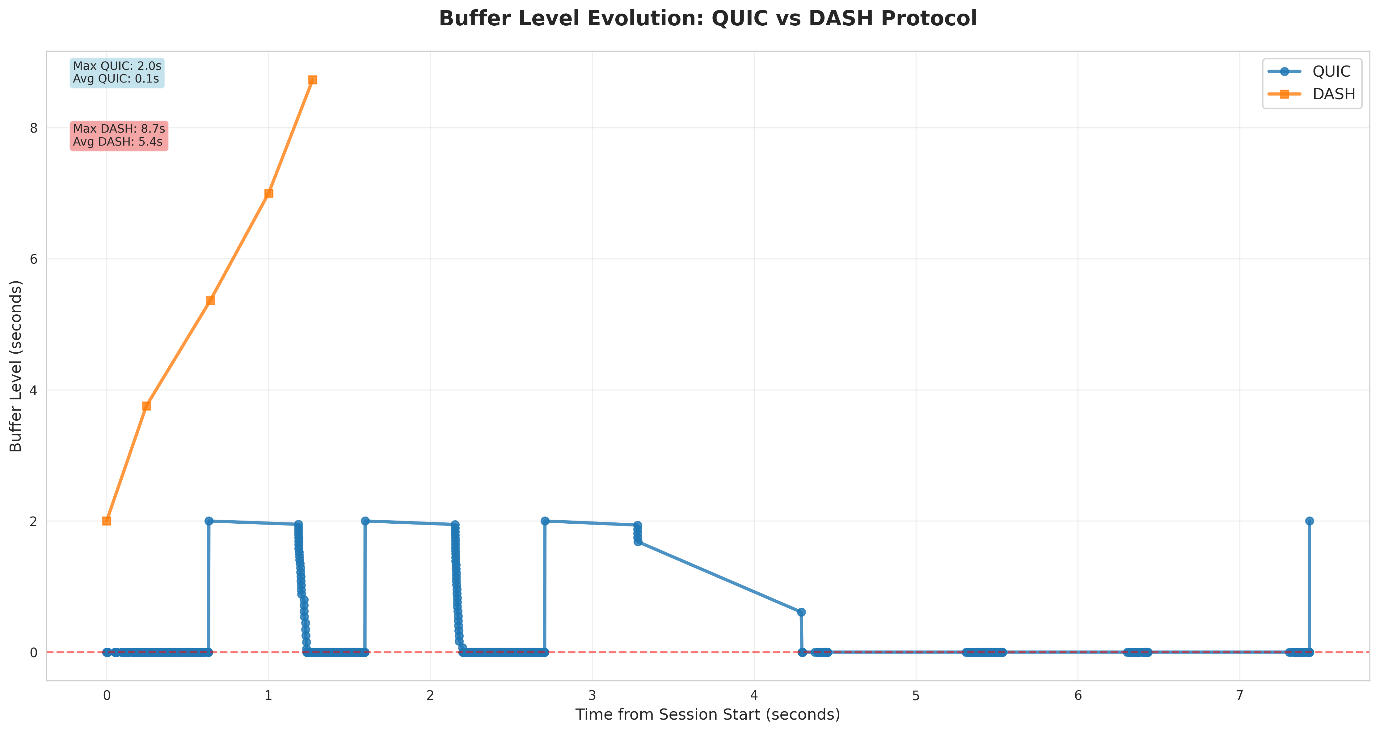
* + 1. سناریو هشتم(WiFi شلوغ)

در این سناریو مشابه سناریو سوم، تلاش شده است تا شبکه یک WiFi که کاربران زیادی نیز به آن متصل هستند اما در کلاینت در فاصله مناسبی از آن وجود دارد شبیه سازی شود، بدین منظور در لینک گلوگاه مقدار 10Mbps پهنای باند و مقدار 40ms تاخیر در نظر گرفته شده است و جیتر معادل 10ms و از دستن رفتن بسته ها 1% است. در چنین سناریویی ترافیک های رقابتی به صورت تصادفی و ترکیبی از 1 تا 5 درخواست TCP و UDP قرار داده شده است.



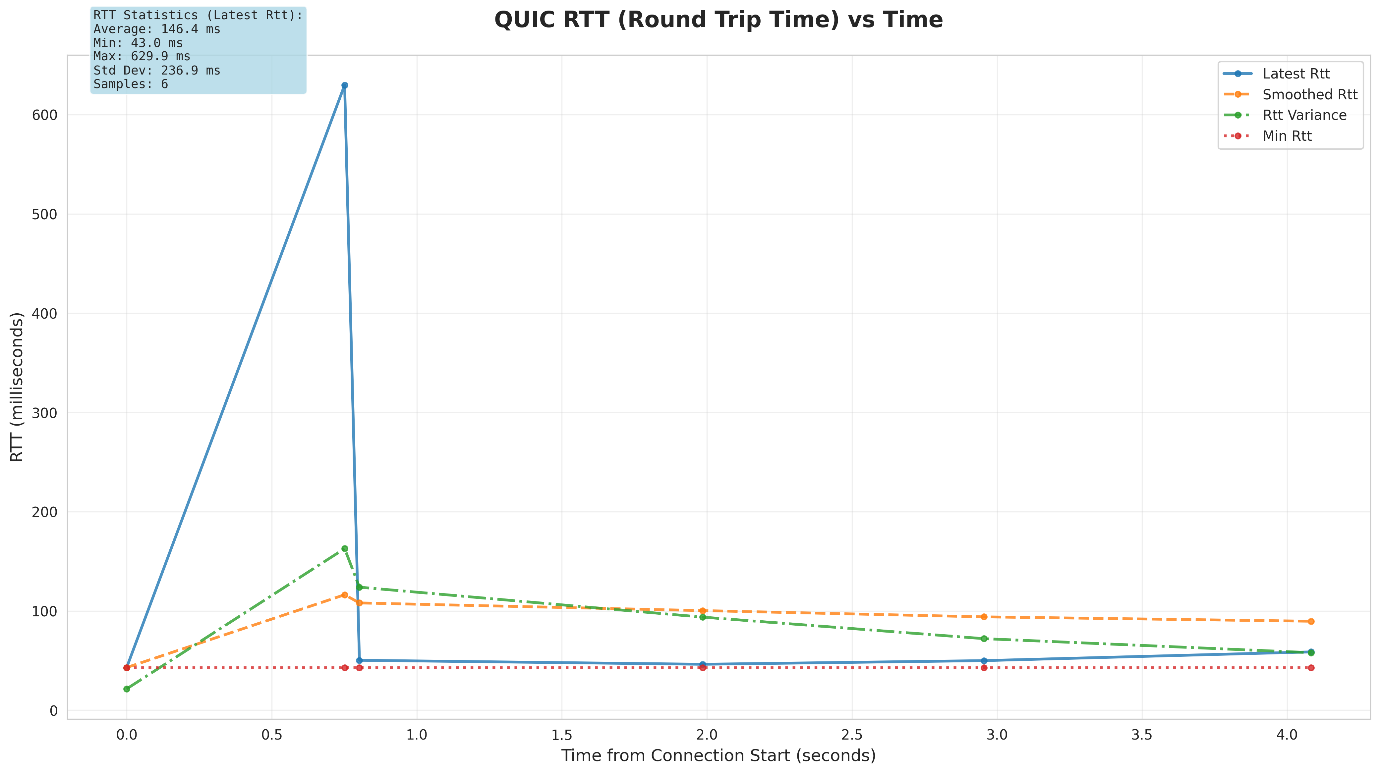
نمودار میزان Bitrate بر زمان برای سناریو هشتم

در (شکل 4-37) نمودار تکامل نرخ بیت‌ریت در طول زمان برای پروتکل‌های QUIC و DASH نشان داده شده است. در پروتکل QUIC، نرخ بیت‌ریت از ۵۰۰ kbps شروع می‌شود و پس از حدود ۱.۵ ثانیه به ۱.۰۸ Mbps می‌رسد. حداکثر نرخ بیت‌ریت در QUIC برابر با ۱۰۸۰ kbps و میانگین آن ۸۷۸ kbps است. در مقابل، در پروتکل DASH، نرخ بیت‌ریت به سرعت به ۲.۵ Mbps می‌رسد و در آن نقطه ثابت می‌ماند. حداکثر نرخ بیت‌ریت در DASH برابر با ۲۵۰۰ kbps و میانگین آن ۱۷۰۰ kbps است. DASH به سرعت به بالاترین نرخ‌های بیت‌ریت می‌رسد و به‌طور میانگین نرخ بیت‌ریت بالاتری نسبت به QUIC دارد. QUIC به‌طور تدریجی و با شیب ملایم‌تری به افزایش بیت‌ریت می‌پردازد که ممکن است به دلیل مدیریت محافظه‌کارانه ازدحام و بافر کوچک آن باشد.



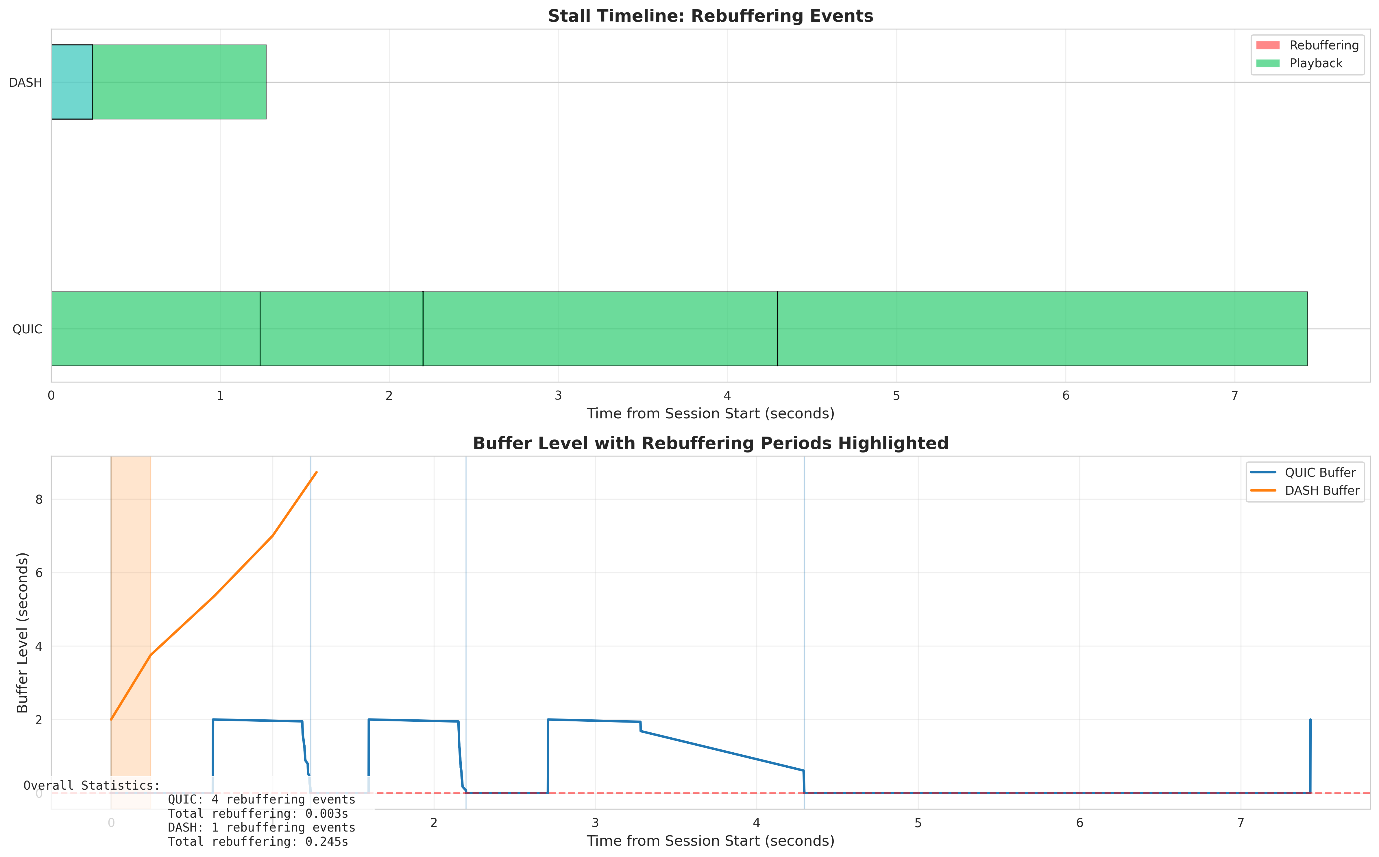
نمودار میزان Buffer-Level بر زمان برای سناریو هشتم

در (شکل 4-38) نمودار تکامل سطح بافر در طول زمان برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC، سطح بافر به‌طور ناگهانی از ۰ به ۲ ثانیه می‌رسد و سپس به مدت طولانی در این سطح باقی می‌ماند. حداکثر سطح بافر در QUIC 2.0 ثانیه و میانگین آن ۰.۱ ثانیه است. در DASH، سطح بافر به‌طور پیوسته از ۰ به ۸.۷ ثانیه افزایش می‌یابد و در این سطح باقی می‌ماند. حداکثر سطح بافر در DASH ۸.۷ ثانیه و میانگین آن ۵.۴ ثانیه است. DASH عملکرد بهتری در مدیریت سطح بافر دارد و سطح بافر بالاتری نسبت به QUIC دارد که این موضوع باعث تجربه پخش پایدارتر در DASH می‌شود.

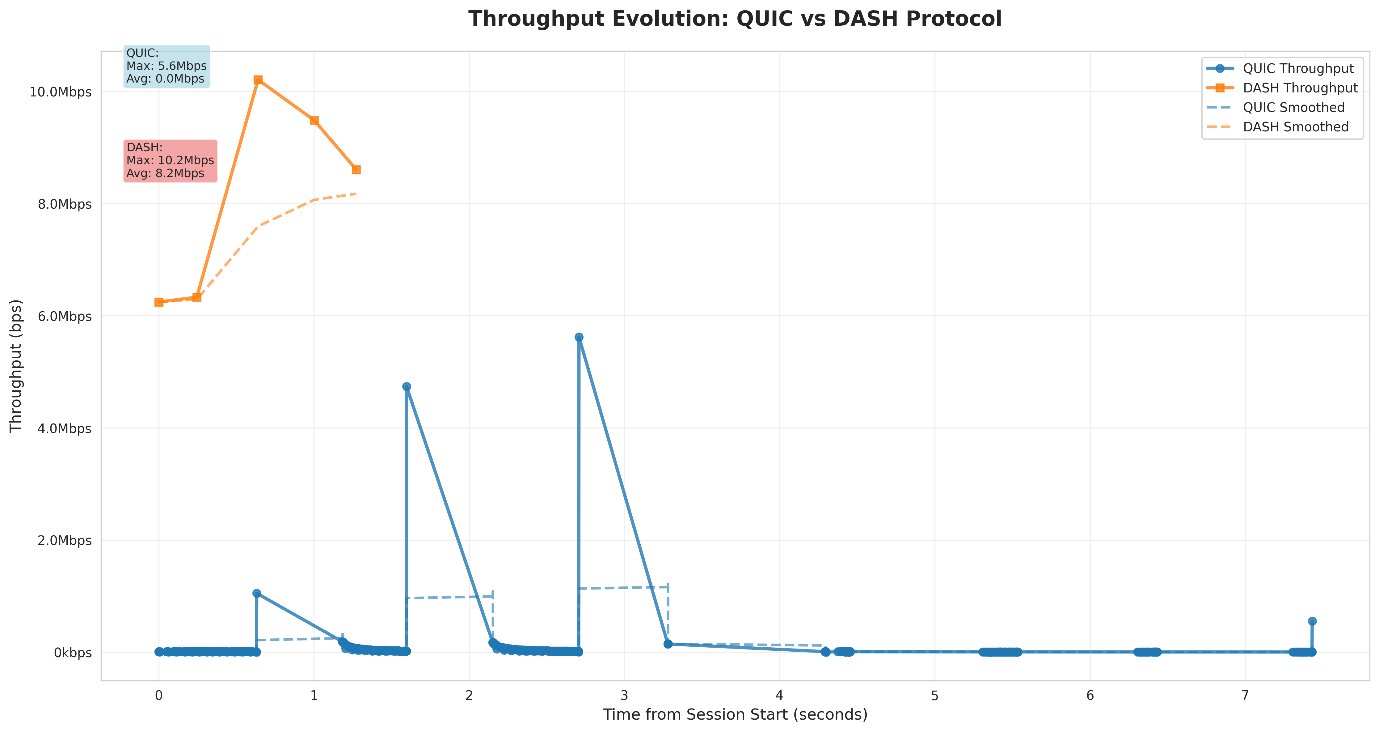


نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو هشتم

در (شکل4-39 ) نمودار زمان رفت‌وبرگشت (RTT) برای پروتکل QUIC در طول زمان نشان داده شده است. در این نمودار، تغییرات زیادی در RTT مشاهده می‌شود. در ابتدا، زمان RTT بسیار پایین است (حدود ۴۰ میلی‌ثانیه) و پس از ۱ ثانیه افزایش می‌یابد. حداکثر RTT در QUIC برابر با ۶۲۹.۹ میلی‌ثانیه است، که نشان‌دهنده نوسانات زیادی در کیفیت اتصال است. میانگین RTT در QUIC 146.4 میلی‌ثانیه است، که این میزان در مقایسه با DASH به مراتب بیشتر است. این نوسانات می‌تواند به دلیل تداخل در ارتباطات و مدیریت بار پروتکل باشد.



نمودار میزان Stall-Timeline بر زمان برای سناریو هشتم

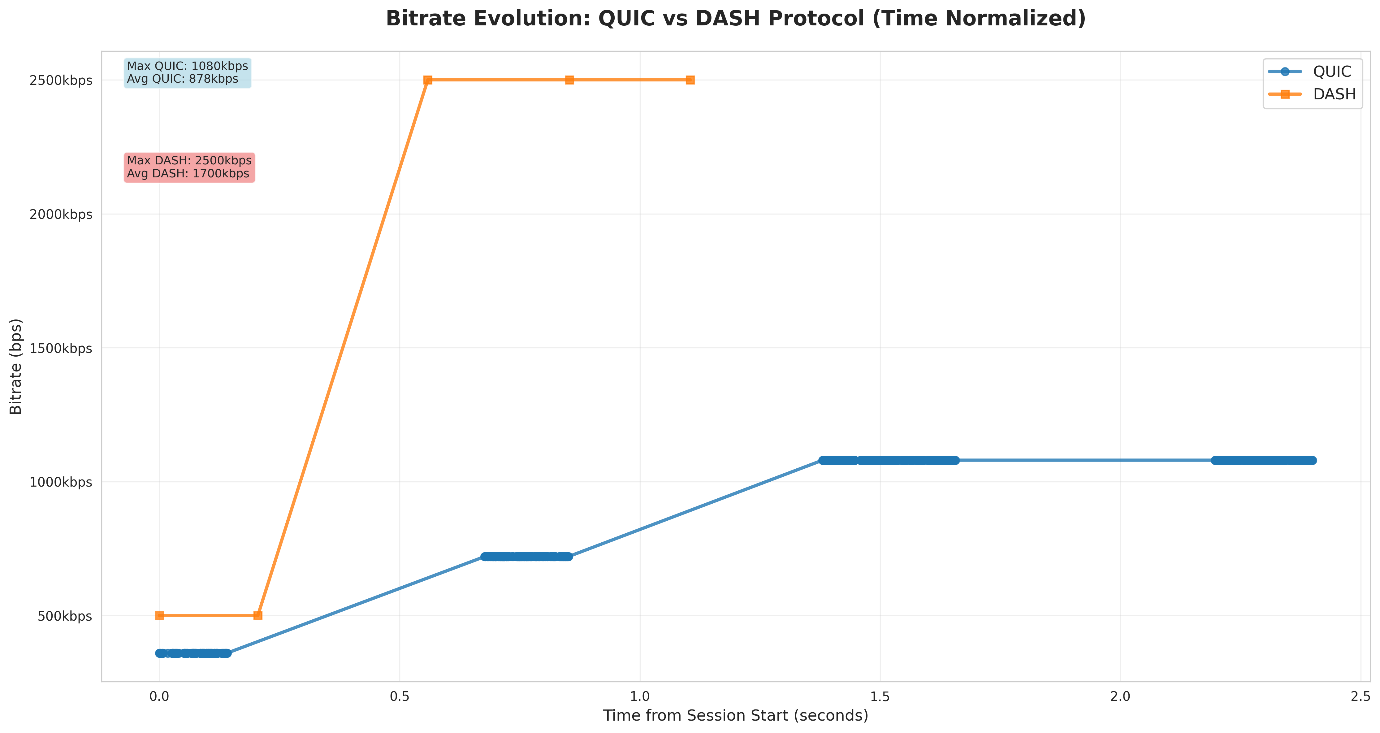
در (شکل 4-40) نمودار زمان وقفه‌ها و نمایش بافر با وقفه‌های رفرش برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC، تعداد وقفه‌های رفرش ۴ عدد است که مجموع زمان رفرش آن تنها ۰.۰۰۳ ثانیه است. این نشان‌دهنده عملکرد سریع و بدون وقفه در پخش است. در DASH، تعداد وقفه‌های رفرش تنها ۱ عدد است، اما مجموع زمان رفرش برابر با ۰.۲۴۵ ثانیه است که از QUIC بیشتر است. این امر نشان می‌دهد که DASH به‌طور کلی بیشتر دچار وقفه در پخش و نیاز به بافرسازی مجدد می‌شود.

نمودار میزان Throughput بر زمان برای سناریو هشتم

در (شکل 4-41) نمودار تکامل throughput برای پروتکل‌های QUIC و DASH در طول زمان نشان داده شده است. در QUIC، throughput از حدود ۰.۵ Mbps شروع می‌شود و با نوسانات زیادی همراه است. حداکثر throughput در QUIC برابر با ۵.۶ Mbps و میانگین آن ۰.۰ Mbps است. در مقابل، در DASH throughput به سرعت به ۱۰.۲ Mbps می‌رسد و تا پایان ثابت می‌ماند. حداکثر throughput در DASH ۱۰.۲ Mbps و میانگین آن ۸.۲ Mbps است. DASH به‌طور پیوسته و با ثبات throughput بالاتری نسبت به QUIC دارد.

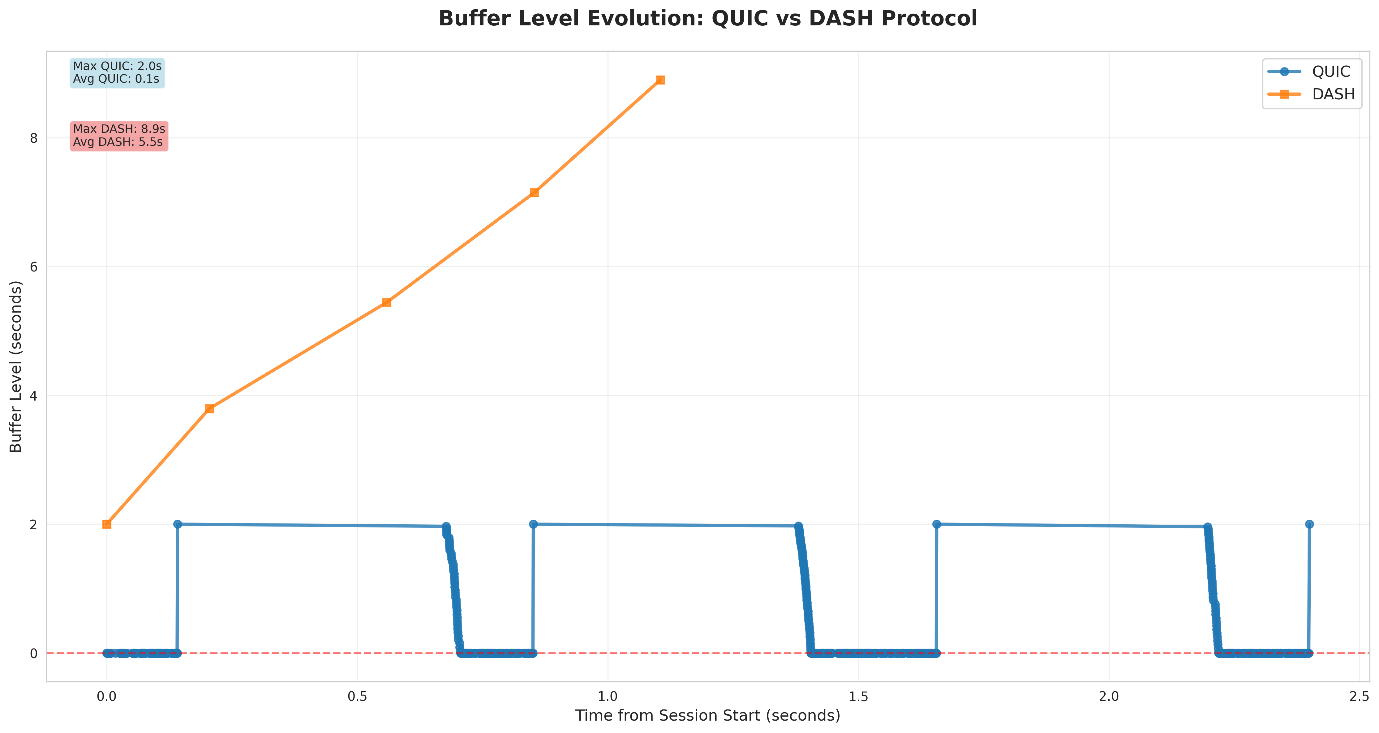
* + 1. سناریو نهم(WiFi شلوغ ناپایدار)

در این سناریو مشابه سناریو هشتم، تلاش شده است تا شبکه یک WiFi که کاربران زیادی نیز به آن متصل هستند اما به علت فرسودگی سخت افزار این WiFi دیگر به صورت پایدار کار نمیکند شبیه سازی شود، بدین منظور در لینک گلوگاه مقدار 10Mbps پهنای باند و مقدار 40ms تاخیر در نظر گرفته شده است و جیتر معادل 30ms و از دستن رفتن بسته ها 1% است. در چنین سناریویی ترافیک های رقابتی به صورت تصادفی و ترکیبی از 1 تا 5 درخواست TCP و UDP قرار داده شده است.



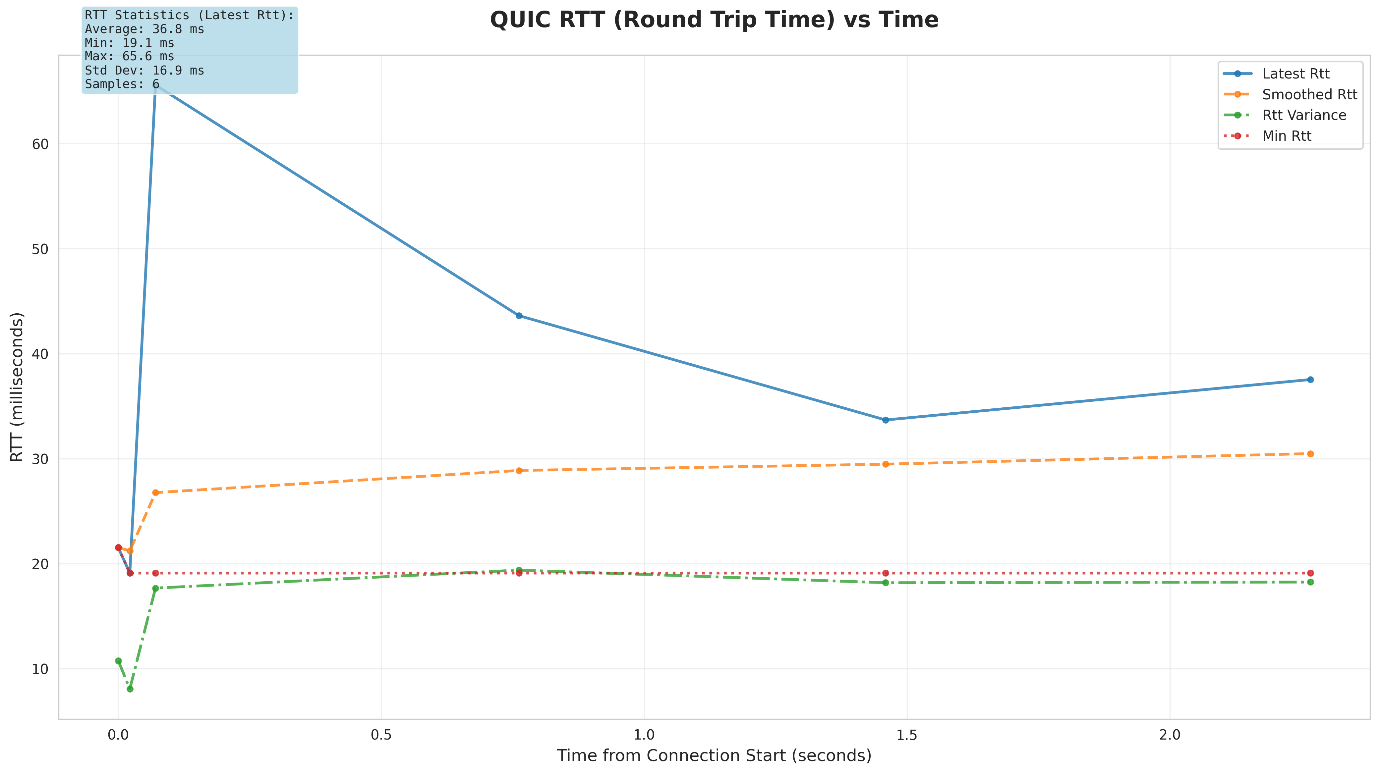
نمودار میزان Bitrate بر زمان برای سناریو نهم

در (شکل 4-42) نمودار تکامل نرخ بیت‌ریت در طول زمان برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC نرخ بیت‌ریت از ۵۰۰ kbps شروع می‌شود و به‌طور تدریجی به ۱.۰۸ Mbps می‌رسد. حداکثر نرخ بیت‌ریت برابر با ۱۰۸۰ kbps و میانگین آن ۸۷۸ kbps است. در DASH نرخ بیت‌ریت به سرعت به ۲.۵ Mbps می‌رسد و در آن نقطه ثابت می‌ماند. حداکثر نرخ بیت‌ریت برابر با ۲۵۰۰ kbps و میانگین آن ۱۷۰۰ kbps است. DASH به سرعت به بالاترین نرخ‌های بیت‌ریت می‌رسد و به‌طور میانگین نرخ بیت‌ریت بالاتری نسبت به QUIC دارد. QUIC به‌طور تدریجی و با شیب ملایم‌تری به افزایش بیت‌ریت می‌پردازد که ممکن است به دلیل مدیریت محافظه‌کارانه ازدحام و بافر کوچک آن باشد.



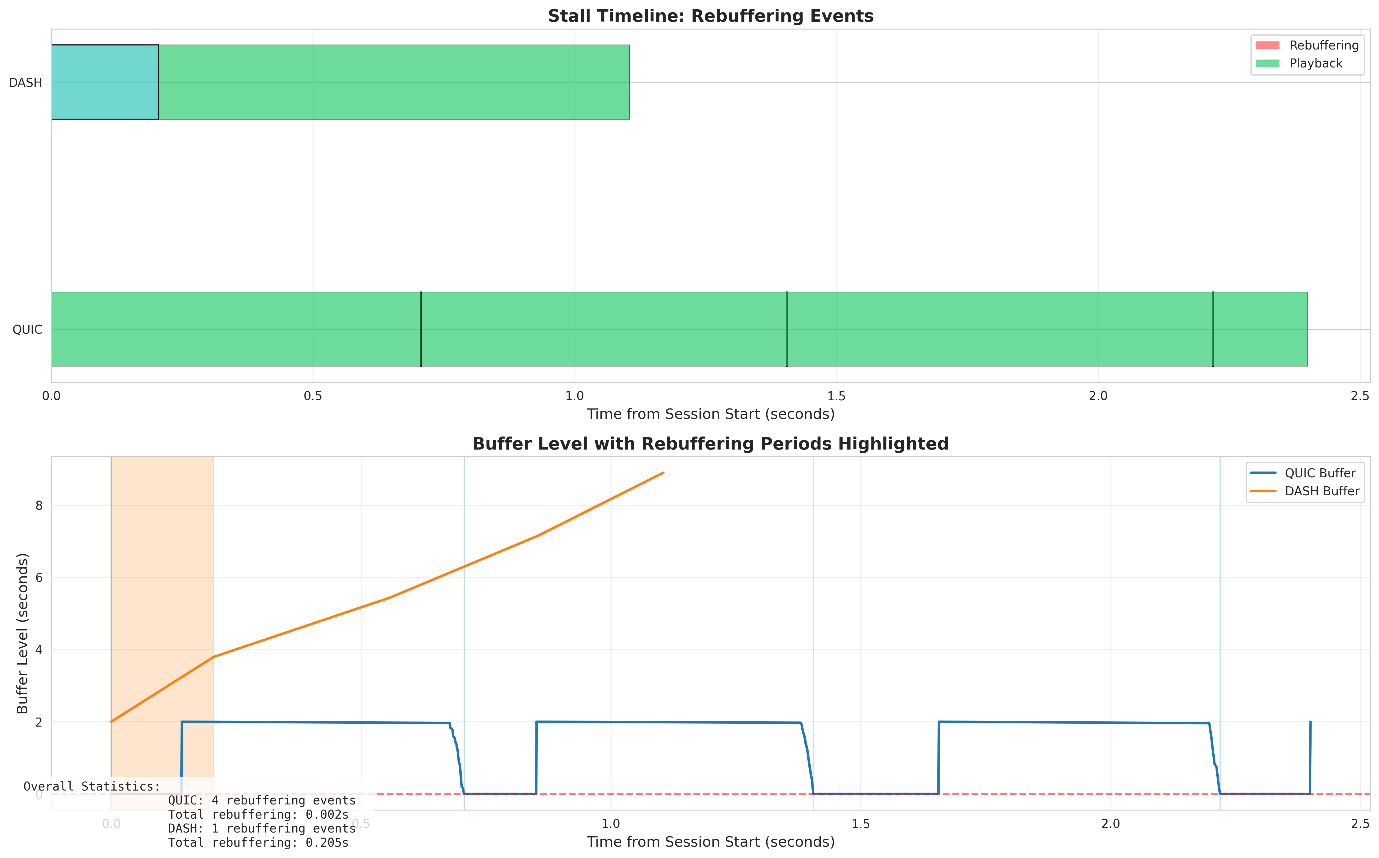
نمودار میزان Buffer-Level بر زمان برای سناریو نهم

در (شکل 4-43) تکامل سطح بافر در طول زمان برای پروتکل‌های QUIC و DASH نمایش داده شده است. در QUIC سطح بافر ابتدا در حدود ۰ ثانیه شروع می‌شود و پس از حدود ۱ ثانیه به ۲ ثانیه می‌رسد و سپس سطح آن در زمان‌های بعدی نوسانات زیادی دارد. حداکثر سطح بافر برای QUIC برابر با ۲.۰ ثانیه و میانگین آن ۰.۱ ثانیه است. در DASH سطح بافر به طور مداوم از ۰ ثانیه به بالاترین سطح بافر یعنی ۸.۷ ثانیه در حدود ۲ ثانیه می‌رسد. حداکثر سطح بافر برای DASH برابر با ۸.۷ ثانیه و میانگین آن ۵.۴ ثانیه است. DASH با افزایش سریع سطح بافر خود، نرخ بهبود مطلوب‌تری را به نمایش می‌گذارد، در حالی که QUIC نوسانات بیشتری در سطح بافر دارد.



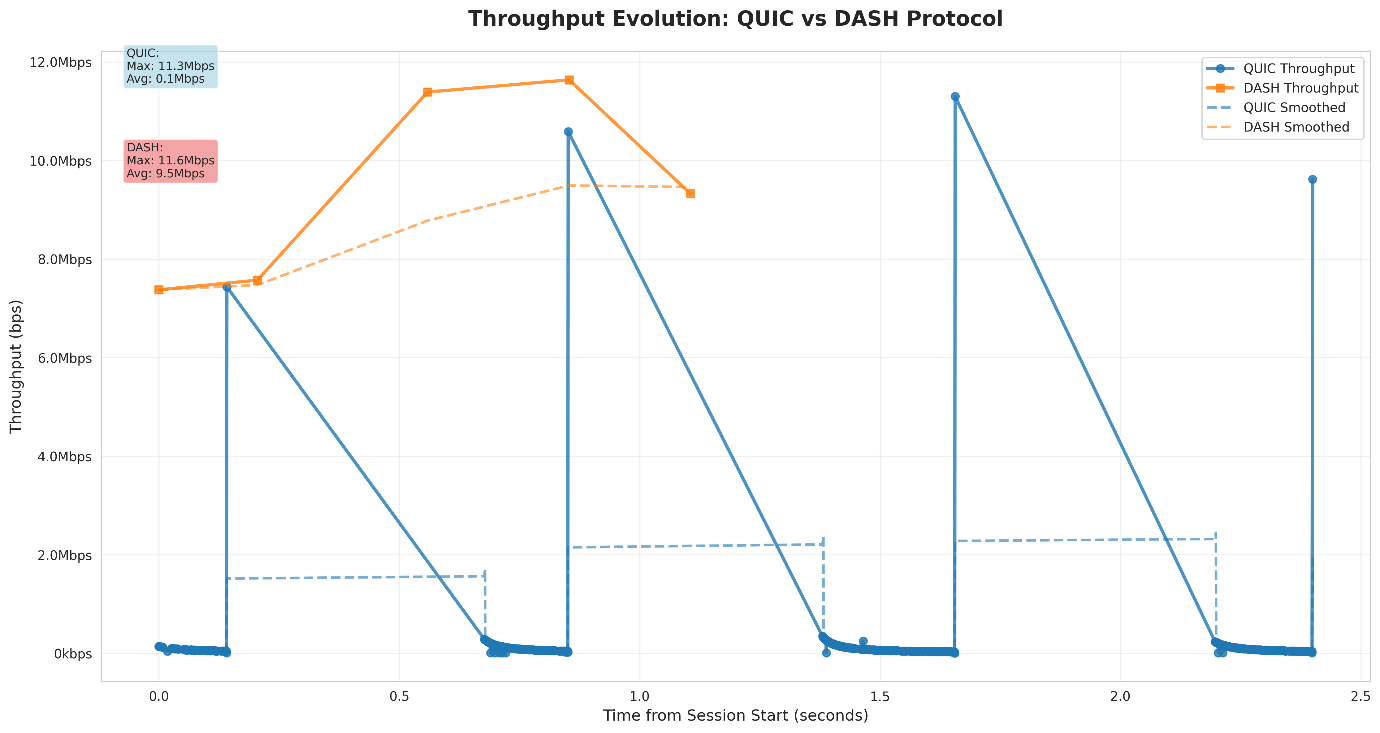
نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو نهم

در (شکل4-44 ) نمودار زمان‌سنجی تأخیر بازگشت (RTT) برای پروتکل QUIC نشان داده شده است. در این نمودار تأخیر RTT در شروع اتصال بسیار بالا است و پس از آن به سرعت کاهش می‌یابد و به حدود ۴۰ میلی‌ثانیه می‌رسد. حداکثر تأخیر RTT برابر با ۶۵ میلی‌ثانیه و میانگین آن ۳۶.۸ میلی‌ثانیه است. این تغییرات ممکن است نشان‌دهنده فرآیند بهبود شبکه و تعاملات پیچیده در پروتکل QUIC باشد.



نمودار میزان Stall-Timeline بر زمان برای سناریو نهم

در (شکل 4-45) نمودار زمان‌سنجی رویدادهای بازسازی (stall) برای پروتکل‌های QUIC و DASH ارائه شده است. رویدادهای بازسازی در پروتکل DASH بیشتر از QUIC است، به‌طوری‌که DASH تنها یک بازسازی دارد که به مدت ۰.۲۴۵ ثانیه ادامه می‌یابد، در حالی‌که QUIC چهار بازسازی دارد که در مجموع ۰.۰۰۳ ثانیه زمان می‌برد. این نشان‌دهنده بهبود عملکرد و نرخ پخش بدون وقفه در QUIC است.

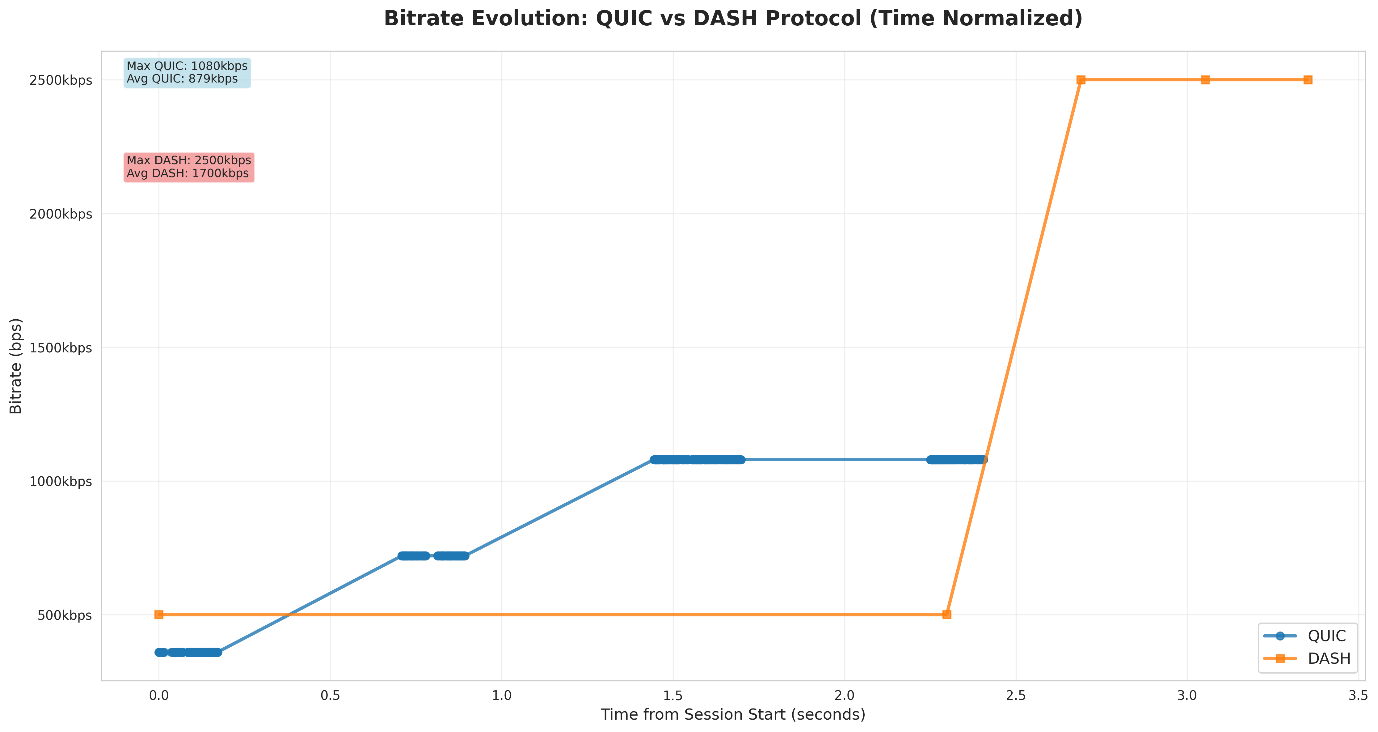


نمودار میزان Throughput بر زمان برای سناریو نهم

در (شکل 4-46) نمودار تکامل توان خروجی (Throughput) در طول زمان برای پروتکل‌های QUIC و DASH نمایش داده شده است. در QUIC، توان خروجی به‌طور ناگهانی از ۰ kbps به ۵.۶ Mbps افزایش می‌یابد و سپس در حدود ۱.۵ ثانیه ثابت می‌ماند. حداکثر توان خروجی برای QUIC برابر با ۵.۶ Mbps و میانگین آن ۰.۱ Mbps است. در مقابل، در DASH، توان خروجی به سرعت به ۱۰.۲ Mbps می‌رسد و در این نقطه ثابت می‌ماند. حداکثر توان خروجی برای DASH برابر با ۱۰.۲ Mbps و میانگین آن ۸.۲ Mbps است. DASH با رسیدن به توان خروجی ثابت و بالا در زمان کوتاهی، توان خروجی بالاتری را نسبت به QUIC نشان می‌دهد، در حالی‌که QUIC در ابتدا نوسانات زیادی دارد و پس از آن به‌طور تدریجی به ثبات می‌رسد.

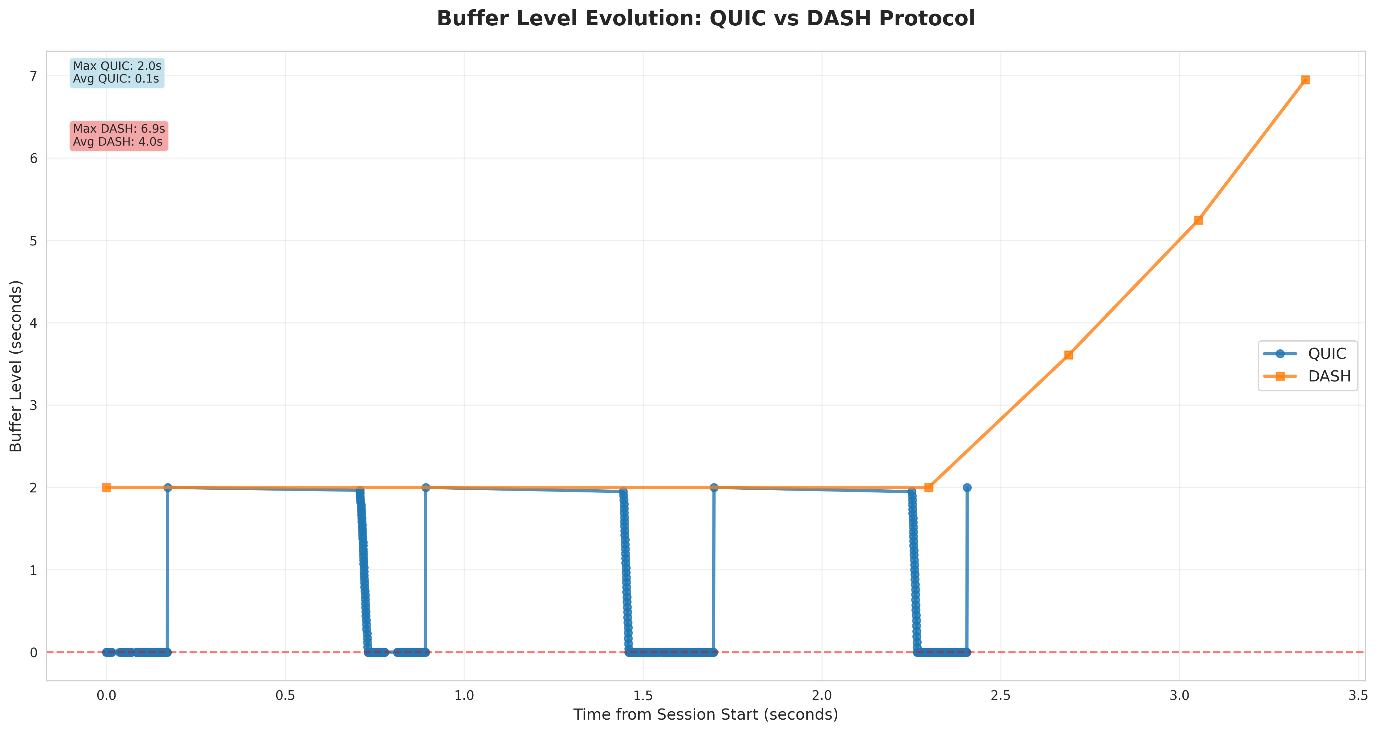
* + 1. سناریو دهم(شبکه معیوب)

این سناریو تلاش شده است تا یک شبکه معیوب شیبه سازی شود که کلاینت در آن درخواست پخش ویدیو را میدهد، بدین منظور در لینک گلوگاه مقدار 10Mbps پهنای باند و مقدار 40ms تاخیر در نظر گرفته شده است و جیتر معادل 10ms و از دستن رفتن بسته ها 3% است. در چنین سناریویی ترافیک رقابتی وجود ندارد.



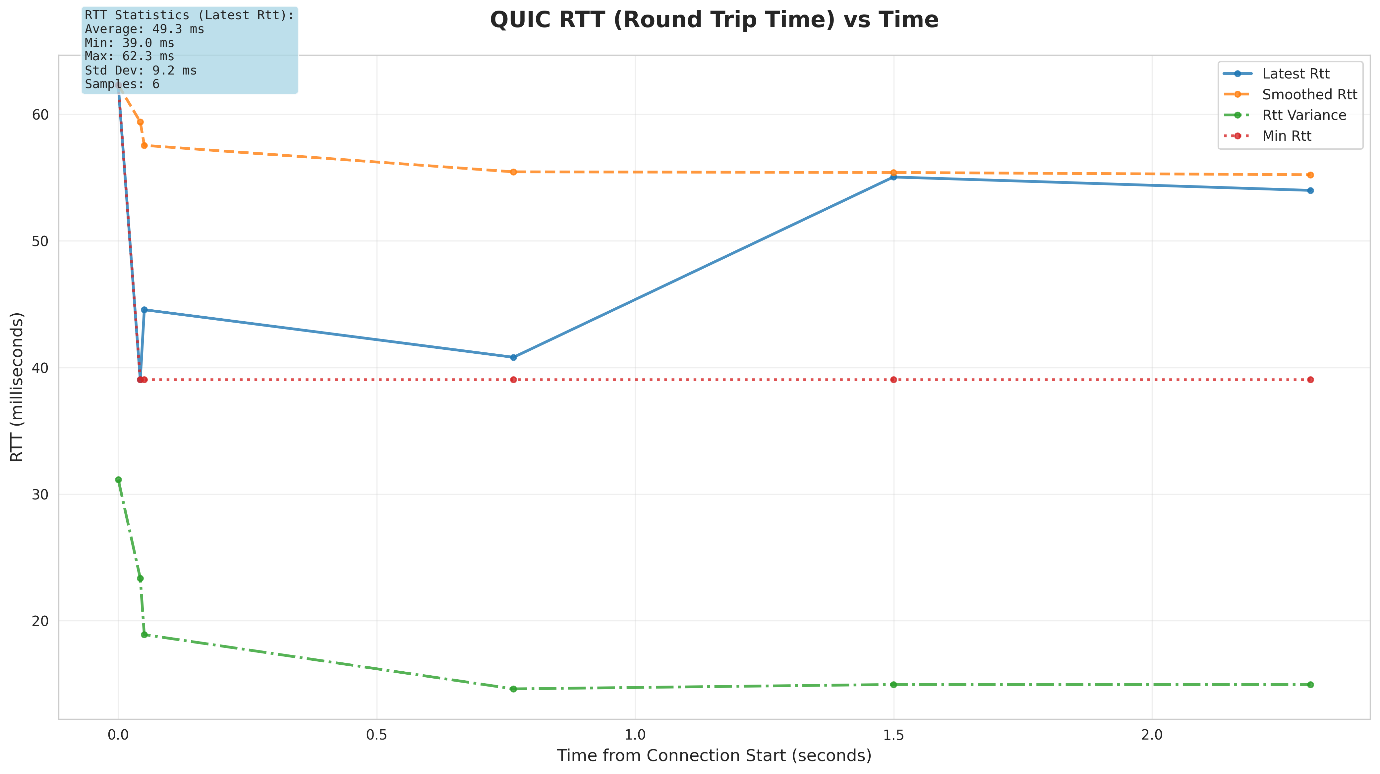
نمودار میزان Bitrate بر زمان برای سناریو دهم

در (شکل 4-47) نمودار تکامل نرخ بیت‌ریت در طول زمان برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC نرخ بیت‌ریت از ۳۵۰ kbps شروع می‌شود و پس از حدود ۱.۵ ثانیه به ۱.۰۸ Mbps می‌رسد. حداکثر نرخ بیت‌ریت برابر با ۱۰۸۰ kbps و میانگین آن ۸۷۸ kbps است. در DASH نرخ بیت‌ریت به سرعت به ۲.۵ Mbps می‌رسد و در آن نقطه ثابت می‌ماند. حداکثر نرخ بیت‌ریت برابر با ۲۵۰۰ kbps و میانگین آن ۱۷۰۰ kbps است. DASH به سرعت به بالاترین نرخ‌های بیت‌ریت می‌رسد و به‌طور میانگین نرخ بیت‌ریت بالاتری نسبت به QUIC دارد. QUIC به‌طور تدریجی و با شیب ملایم‌تری به افزایش بیت‌ریت می‌پردازد که ممکن است به دلیل مدیریت محافظه‌کارانه ازدحام و بافر کوچک آن باشد.



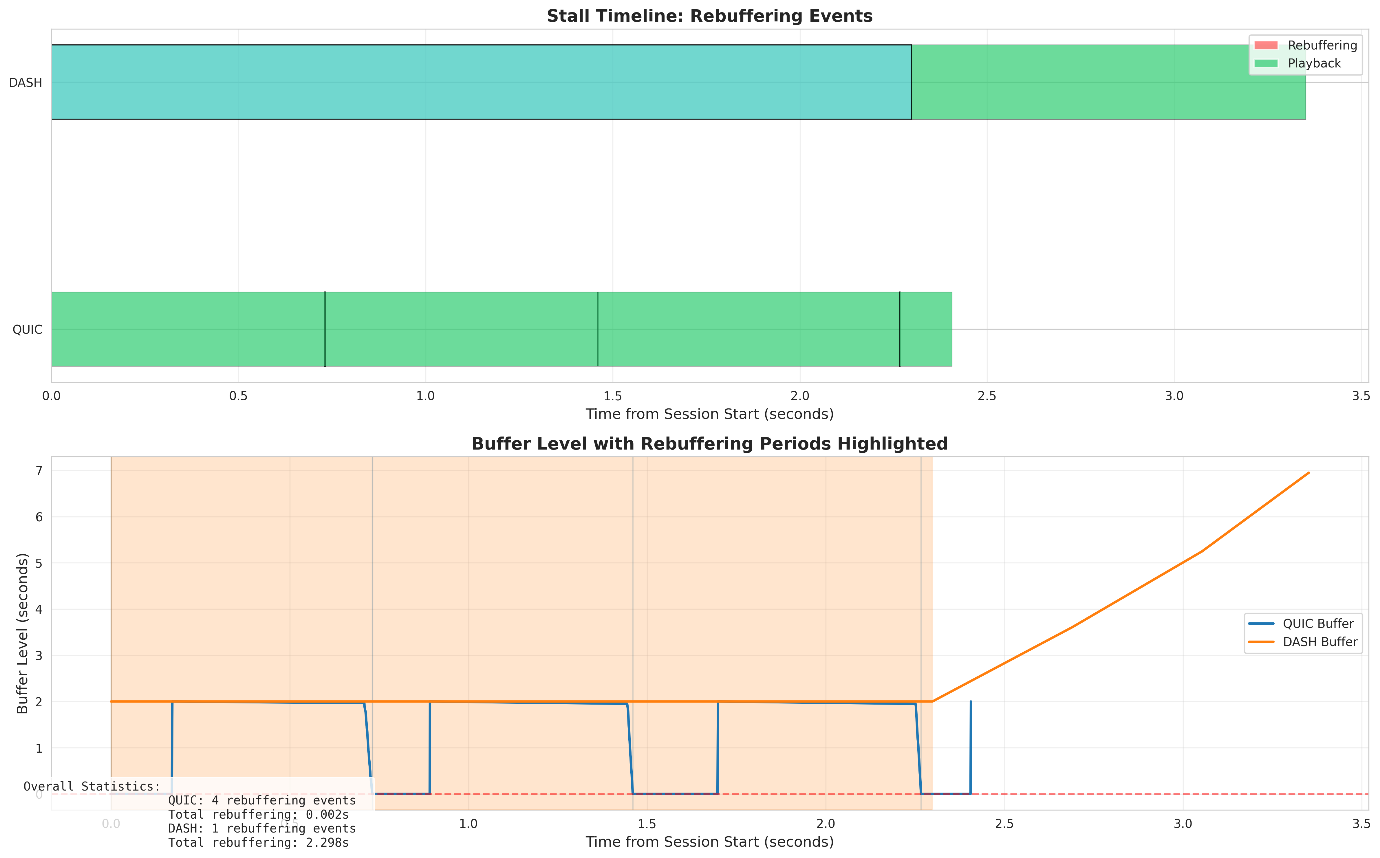
نمودار میزان Buffer-Level بر زمان برای سناریو دهم

در (شکل 4-48) نمودار تکامل سطح بافر در طول زمان برای پروتکل‌های QUIC و DASH نشان داده شده است. در QUIC سطح بافر به تدریج افزایش می‌یابد و پس از ۲ ثانیه به ۲ ثانیه می‌رسد. این پروتکل از نرخ‌های بیت‌ریت پایین‌تر بهره می‌برد، به همین دلیل سطح بافر کندتر افزایش می‌یابد. در DASH سطح بافر به سرعت افزایش می‌یابد و به حدود ۸.۷ ثانیه می‌رسد. DASH به دلیل استفاده از نرخ بیت‌ریت بالاتر قادر است که بافر بیشتری را ذخیره کند. این نمودار نشان‌دهنده تفاوت در مدیریت بافر و نحوه استفاده از پهنای باند در این دو پروتکل است..

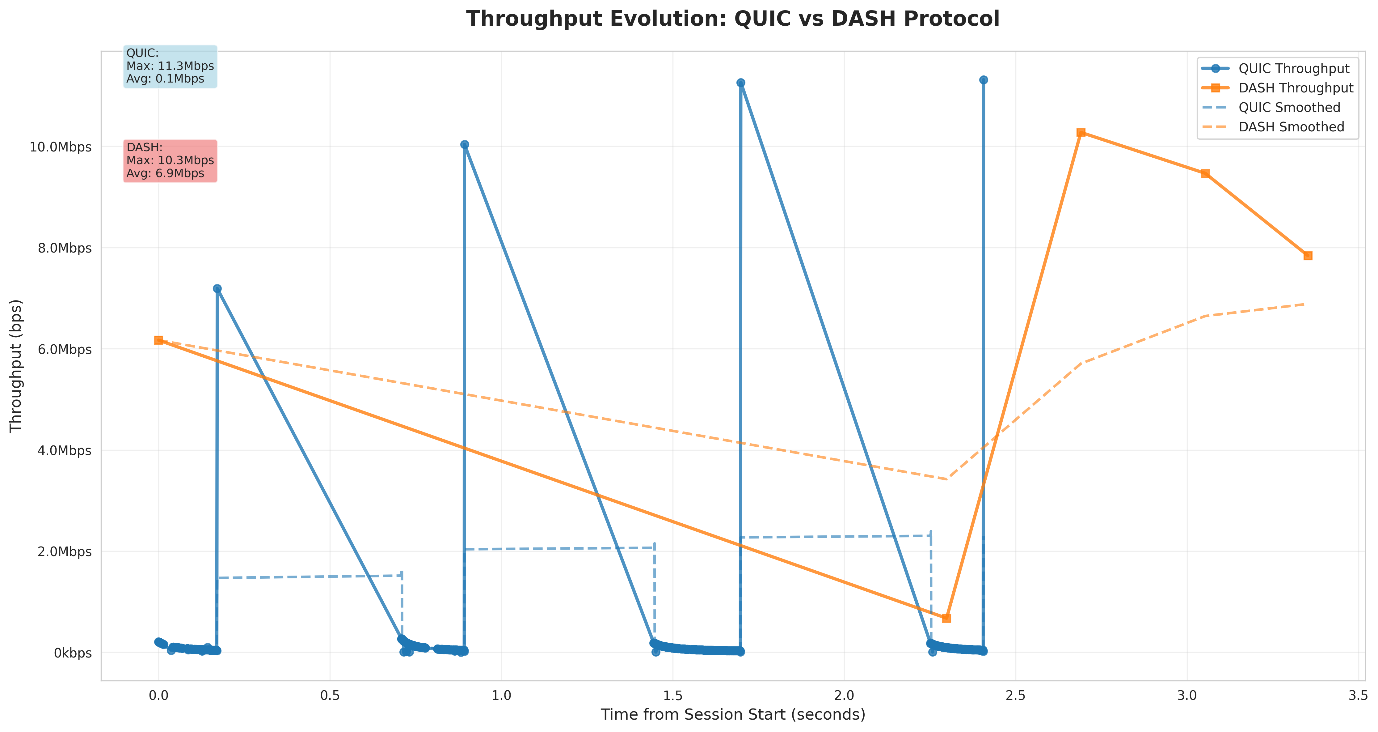


نمودار میزان RTTبر زمان برای پروتکل QUIC سناریو دهم

در (شکل4-49 ) نمودار زمان تأخیر دور (RTT) پروتکل QUIC در طول زمان نمایش داده شده است. در ابتدا، تأخیر دور برابر با ۶۳ میلی‌ثانیه است و پس از ۱ ثانیه به ۴۵ میلی‌ثانیه کاهش می‌یابد. پس از آن، تأخیر دور به‌طور تدریجی به مقدار ۳۶ میلی‌ثانیه کاهش می‌یابد و تا انتهای نمودار ثابت می‌ماند. این کاهش تأخیر دور نشان‌دهنده توانایی QUIC در بهینه‌سازی ارتباطات و کاهش زمان پاسخ‌دهی است. با وجود نوسانات اولیه، QUIC موفق به کاهش تأخیر به یک مقدار پایدار می‌شود.



نمودار میزان Stall-Timeline بر زمان برای سناریو دهم

در (شکل 4-50) دو نمودار نمایش داده شده است. نمودار اول نشان‌دهنده زمان‌های ریوفرینگ برای هر پروتکل است. در QUIC، چهار رویداد ریوفرینگ وجود دارد که مجموعاً حدود ۰.۰۰۲ ثانیه طول می‌کشد. در DASH، تنها یک رویداد ریوفرینگ ثبت شده که ۰.۲۴۵ ثانیه طول کشیده است. نمودار دوم سطح بافر را نشان می‌دهد که در آن بازه‌های ریوفرینگ با رنگ‌های مختلف برجسته شده‌اند. در DASH، بافر به‌طور مداوم در حال افزایش است و تنها در زمان ریوفرینگ افت می‌کند، در حالی که در QUIC، سطح بافر به صورت غیرخطی و با نوسانات مختلف تغییر می‌کند. 

نمودار میزان Throughput بر زمان برای سناریو دهم

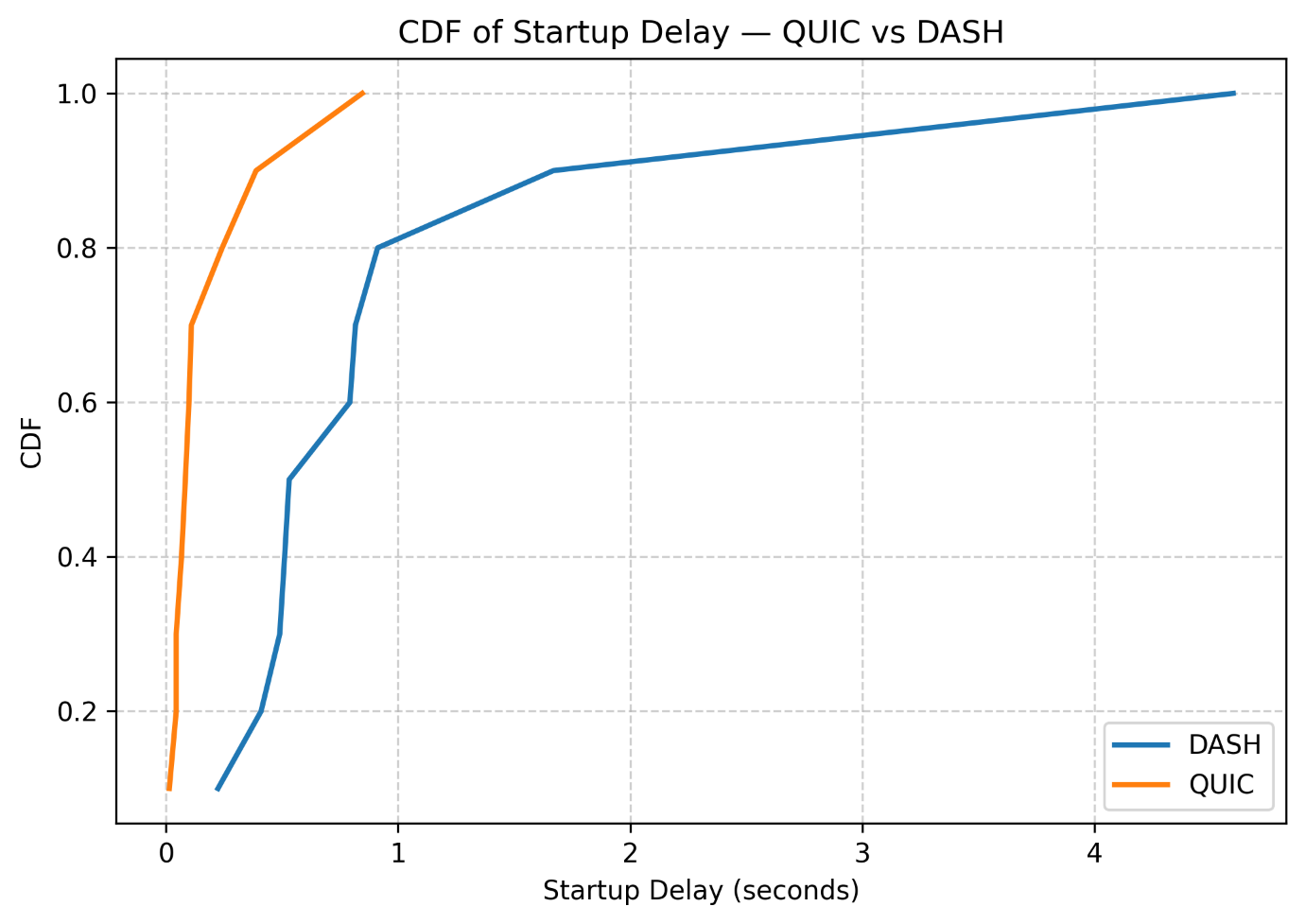
در (شکل 4-50) نمودار تکامل نرخ انتقال برای پروتکل‌های QUIC و DASH نمایش داده شده است. در QUIC، نرخ انتقال در ابتدا برابر با ۱.۰ Mbps است و پس از آن به طور ناگهانی به ۱۰.۷ Mbps می‌رسد. در DASH، نرخ انتقال ابتدا ۱.۰ Mbps است و پس از مدتی به ۱۰.۲ Mbps می‌رسد. پروتکل DASH نرخ انتقال بالاتری در طول زمان دارد و در مقایسه با QUIC، تغییرات کمتری در نرخ انتقال مشاهده می‌شود. این تفاوت به این معنی است که DASH قادر به حفظ یک نرخ انتقال ثابت‌تر است، در حالی که QUIC به سرعت به پیک‌های بالاتری از نرخ انتقال می‌رسد.

* 1. بررسی تجمیعی سناریوها برای معیار تاخیر راه‌اندازی(startup delay)
     1. تفسیر عمومی نمودار CDF برای Startup-Delay

در نمودار Cumulative Distribution Function (CDF) of Startup Delay (شکل 4-52) برای پروتکل‌های QUIC و DASH، توزیع تأخیر راه‌اندازی برای هر یک از این پروتکل‌ها در مدت زمان مشخصی از زمان شروع نشست نشان داده شده است. این نمودار با نشان دادن درصد از کل زمان‌ها که تأخیر راه‌اندازی کمتر از مقدار خاصی است، قادر به مقایسه عملکرد دو پروتکل از نظر سرعت راه‌اندازی است.

محور افقی نشان‌دهنده زمان تأخیر راه‌اندازی (Startup Delay) است که به ثانیه اندازه‌گیری شده است. محور عمودی نشان‌دهنده CDF (تابع توزیع تجمعی) است که درصد نمونه‌ها را که تأخیر راه‌اندازی آنها کمتر از مقدار خاصی است، به نمایش می‌گذارد.

دو خط در نمودار وجود دارد که خط آبی نشان‌دهنده تأخیر راه‌اندازی پروتکل DASH و خط نارنجی نشان‌دهنده تأخیر راه‌اندازی پروتکل QUIC است.



نمودار میزان Buffer-Level بر زمان برای سناریو دهم

* + 1. تفسیر نتایج داخل نمودار

پروتکل QUIC از نظر تأخیر راه‌اندازی، عملکرد بهتری نسبت به DASH نشان می‌دهد. در این زمینه، مراحل مختلف عملکرد QUIC به‌طور دقیق‌تر به شرح زیر است:

* QUIC:
  + **شروع سریعتر:** در نمودار، پروتکل QUIC به طور قابل توجهی سریع‌تر از DASH به ۱۰۰ درصد CDF می‌رسد. این بدان معناست که بیشتر نمونه‌ها در QUIC دارای تأخیر راه‌اندازی کمتر از ۱ ثانیه هستند.
  + **نرخ تثبیت:** QUIC به سرعت به درصدهای بالا در CDF می‌رسد و قبل از ۱ ثانیه به ۱۰۰ درصد می‌رسد. این نشان‌دهنده آن است که پروتکل QUIC به سرعت به تثبیت می‌رسد و بیشتر تأخیر‌ها در این پروتکل پایین هستند.
* DASH:
  + **شروع کندتر:** در ابتدای نمودار، DASH رشد CDF کندتری دارد و این نشان می‌دهد که بیشتر نمونه‌ها در DASH تأخیر راه‌اندازی بیشتری دارند. در واقع تأخیر راه‌اندازی در DASH معمولاً بالاتر از QUIC است.
  + **نسبت به QUIC، تأخیر زیادتر:** به‌طور کلی، DASH تأخیر راه‌اندازی بیشتری نسبت به QUIC دارد و به این علت که رشد نمودار DASH به مراتب کند تر از QUIC است و شیب نمودار کمتری را دارد که این موضوع اینگونه تفسیر میشود که تعداد بیشتری از سناریوها در حالت DASH مدت زمان تاخیر راه اندازی بیشتری را نسبت به QUIC تجربه کرده‌اند.

1. جمع‌بندي و پيشنهادها
   1. مقدمه

در سال‌های اخیر، پروتکل‌های مختلفی برای بهبود کیفیت پخش ویدیو و انتقال داده‌ها در شبکه‌های اینترنتی توسعه یافته‌اند. دو پروتکل مهم که در زمینه‌ی پخش ویدیو و انتقال داده‌ها استفاده می‌شوند، QUIC و DASH هستند. هر کدام از این پروتکل‌ها ویژگی‌ها و مزایای خاص خود را دارند که آنها را برای شرایط خاص شبکه بهینه می‌کند. پروتکل QUIC با هدف کاهش تأخیر و افزایش سرعت انتقال داده در شبکه‌های با شرایط متغیر طراحی شده است، در حالی که DASH به‌عنوان یک پروتکل برای پخش ویدیوهای تطبیقی، بهینه‌سازی پهنای باند و تضمین کیفیت ثابت را هدف قرار داده است.

در این تحقیق، به مقایسه عملکرد این دو پروتکل در شرایط مختلف شبکه پرداخته شد. معیارهای مختلفی همچون نرخ Bitrate، سطح بافر، تأخیر راه‌اندازی، زمان رفت‌وبرگشت (RTT)، نرخ انتقال داده (Throughput) و زمان توقف/ریبافر برای هر کدام از این پروتکل‌ها ارزیابی شد. هدف این تحقیق، بررسی نقاط قوت و ضعف هر پروتکل و تعیین مناسب‌ترین پروتکل برای شرایط مختلف شبکه بوده است.

در ادامه، نتایج حاصل از این مقایسه در زمینه‌های مختلف، شامل تأخیر راه‌اندازی، پایداری پخش ویدیو، و استفاده از پهنای باند، آورده شده است. این نتایج می‌تواند به تصمیم‌گیری در انتخاب بهترین پروتکل برای کاربردهای مختلف کمک کند و همچنین درک بهتری از رفتار این پروتکل‌ها در شرایط شبکه‌های مختلف ارائه دهد.

* 1. نتایج و پیشنهاد
     1. جمع‌بندي

پروتکل‌های QUIC و DASH دو پروتکل متفاوت در زمینه‌ی انتقال داده هستند که در شبکه‌های پخش ویدیو و سایر برنامه‌های تحت شبکه کاربرد دارند. در این تحقیق، عملکرد این دو پروتکل در چندین جنبه‌ی مهم از جمله نرخ Bitrate، سطح بافر، تأخیر راه‌اندازی، زمان رفت‌وبرگشت (RTT)، نرخ انتقال داده (Throughput) و زمان توقف مورد بررسی قرار گرفت. نتایج این بررسی به وضوح تفاوت‌های موجود در عملکرد این پروتکل‌ها را نشان می‌دهد.

* **نرخ Bitrate:** پروتکل DASH به سرعت به نرخ‌های بالاتر می‌رسد و از نظر میانگین نرخ Bitrate در مقایسه با QUIC عملکرد بهتری دارد. در حالی که QUIC به‌طور تدریجی نرخ بیت‌ریت را افزایش می‌دهد و در نهایت به حداکثر مقدار خود می‌رسد، DASH با سرعت بیشتری به نرخ‌های بالای بیت‌ریت دست می‌یابد و تا حد زیادی این نرخ‌ها را ثابت نگه می‌دارد. DASH عملکرد بهتری در دسترسی به نرخ‌های بالاتر دارد، که به‌ویژه در شرایطی که کیفیت ویدیو یا پهنای باند بالاتر اهمیت دارد، مفید است. QUIC در مواردی که نیاز به شروع سریع و رسیدن به نرخ بیت‌ریت کم‌تری وجود دارد، مناسب‌تر است.
* **سطح بافر (Buffer Level):** در مورد سطح بافر، DASH به‌طور کلی بافر بیشتری در مقایسه با QUIC ایجاد می‌کند و می‌تواند پخش ویدیو را بدون وقفه‌های طولانی‌مدت انجام دهد. این به آن معناست که در شرایطی که تأخیر و نوسانات شبکه زیاد است، DASH می‌تواند با استفاده از بافر بزرگ‌تر، پخش را بدون وقفه نگه دارد. QUIC در این زمینه دارای نوسانات بیشتر و سطح بافر کمتری است که می‌تواند منجر به تأخیر بیشتر در شرایط ناپایدار شبکه شود. DASH در شرایط شبکه‌ای که نیاز به پخش پایدارتر و بدون وقفه‌های زیاد دارند، عملکرد بهتری نشان می‌دهد.
* **تأخیر راه‌اندازی (Startup Delay):** تأخیر راه‌اندازی در QUIC به طور قابل توجهی کمتر از DASH است. QUIC با سرعت بیشتری شروع به کار می‌کند و به سرعت به نرخ‌های پایین‌تری دست می‌یابد. QUIC به‌طور کلی برای سناریوهایی که نیاز به راه‌اندازی سریع دارند مناسب‌تر است. DASH در شروع به‌طور کندتری عمل می‌کند و تأخیر بیشتری در زمان راه‌اندازی تجربه می‌کند، اما در پخش‌های طولانی‌تر عملکرد پایدارتری دارد.
* **زمان رفت‌وبرگشت (RTT):** در مورد RTT (زمان رفت‌وبرگشت)، QUIC نسبت به DASH عملکرد بهتری در کاهش تأخیر دارد. پس از گذشت مدت کوتاهی از شروع ارتباط، QUIC می‌تواند تأخیر را به طور قابل توجهی کاهش دهد و در این زمینه از DASH پیشی می‌گیرد. به عبارت دیگر، QUIC در نوسانات شبکه و پردازش سریع اطلاعات به‌طور مؤثری عمل می‌کند.
* **نرخ انتقال داده (Throughput):** در نرخ انتقال داده (Throughput)، DASH عملکرد پایدارتری دارد و قادر به حفظ نرخ‌های بالاتر برای مدت طولانی‌تر است. در مقایسه با QUIC، که در برخی لحظات نوسانات زیادی در throughput مشاهده می‌شود، DASH می‌تواند سرعت انتقال بالاتری را با ثبات بیشتری فراهم کند. DASH توانایی بیشتری در استفاده مؤثر از پهنای باند دارد و در نهایت نرخ انتقال داده بالاتری نسبت به QUIC نشان می‌دهد.
* زمان توقف/ریبافر (Stall Timeline): در بخش زمان توقف و بازپخش، QUIC عملکرد بهتری دارد و تعداد کمتری از وقفه‌های بازپخش را تجربه می‌کند. این به آن معناست که QUIC قادر است پخش ویدیو را در شرایط شبکه‌ای پرنوسان، سریعتر و بدون تأخیرهای زیاد انجام دهد. در حالی که DASH برخی اوقات دچار توقف‌های طولانی‌تری می‌شود که باعث کاهش کیفیت تجربه کاربری می‌شود.

نتیجه‌گیری نهایی که از این آزمایشات در سناریو های مختلف میتوان گرفت این است که، پروتکل‌های QUIC و DASH در سناریوهای مختلف عملکرد متفاوتی دارند. QUIC از نظر تأخیر راه‌اندازی، RTT و شروع سریع بهینه‌تر است و می‌تواند در شبکه‌هایی که نیاز به سرعت بالا در شروع پخش دارند، عملکرد بهتری نشان دهد. در مقابل، DASH از نظر پایداری پخش ویدیو، نرخ بیت‌ریت، سطح بافر و نرخ انتقال داده به‌طور کلی عملکرد بهتری دارد و برای شرایطی که نیاز به پخش ویدیو بدون وقفه و کیفیت پایدار وجود دارد، مناسب‌تر است.

در نهایت، انتخاب بین این دو پروتکل به نیاز خاص سیستم و شرایط شبکه بستگی دارد. برای کاربردهایی که تأخیر کم و شروع سریع اهمیت دارد، QUIC مناسب‌تر است، در حالی که DASH برای پخش پایدار و با کیفیت و مدیریت بهتر پهنای باند انتخاب بهتری خواهد بود.

* + 1. نوآوري

تحقیق حاضر در زمینه مقایسه پروتکل‌های QUIC و DASH برای پخش ویدیو در شبکه‌ها، چندین نوآوری کلیدی دارد که آن را از دیگر مطالعات مشابه متمایز می‌کند. این نوآوری‌ها در چندین جنبه از عملکرد پروتکل‌ها، تحلیل داده‌ها و ارائه پیشنهادات برای بهینه‌سازی قابل مشاهده هستند.

1. **مقایسه جامع عملکرد QUIC و DASH در شرایط مختلف شبکه**

یکی از نوآوری‌های اصلی این تحقیق، مقایسه جامع و سیستماتیک عملکرد پروتکل‌های QUIC و DASH در شرایط مختلف شبکه است. به طور خاص، این تحقیق به بررسی تأخیرهای راه‌اندازی، نرخ انتقال داده، سطح بافر، تأخیر رفت‌وبرگشت (RTT) و زمان توقف پرداخته و تفاوت‌های عملکردی بین این دو پروتکل را با استفاده از سیستم‌های واقعی و سناریوهای متفاوت شبکه بررسی کرده است.

1. **تأکید بر جنبه‌های مختلف عملکرد پروتکل‌ها**

در این تحقیق، برخلاف بسیاری از مطالعات که تنها به یک یا دو جنبه از عملکرد پروتکل‌ها می‌پردازند، ما به‌طور جامع به چندین جنبه کلیدی عملکرد پروتکل‌ها توجه کرده‌ایم. این جنبه‌ها شامل نرخ بیت‌ریت، تأخیر راه‌اندازی، پایداری در نرخ انتقال داده، توقف‌ها و بافرینگ، و زمان‌های رفت‌وبرگشت (RTT) می‌شود. این نگاه جامع و همه‌جانبه به تحلیل عملکرد پروتکل‌ها، می‌تواند به بهینه‌سازی هر دو پروتکل در شرایط شبکه‌های مختلف کمک کند.

1. **نوآوری در تحلیل و ارائه نتایج با استفاده از CDF**

این تحقیق از نمودارهای تابع توزیع تجمعی (CDF) برای تجزیه و تحلیل تأخیر راه‌اندازی و مقایسه آن‌ها استفاده کرده است. استفاده از این روش در تحلیل تأخیر راه‌اندازی، امکان مقایسه دقیق‌تری بین پروتکل‌ها و مشاهده رفتار آن‌ها در شرایط مختلف شبکه را فراهم کرده است. این نوع تحلیل به‌ویژه در مقایسه سریع پروتکل‌ها و بررسی زمان‌هایی که به ثبات می‌رسند، مفید بوده و نتایج جدیدی را به همراه داشته است.

1. **مقایسه بین پروتکل‌های QUIC و DASH در سناریوهای مختلف**

در این تحقیق، برای اولین بار مقایسه دقیقی بین QUIC و DASH در سناریوهای مختلف، از جمله شبکه‌های پر ترافیک، کم ترافیک، و پهنای باند متغیر انجام شده است. این مقایسه‌های جزئی و دقیق در شرایط واقعی، به درک بهتری از کاربرد هر پروتکل در شرایط مختلف شبکه کمک کرده است.

* + 1. پيشنهادها

با توجه به نتایج به‌دست آمده از این تحقیق، چندین پیشنهاد برای پژوهش‌های آینده و بهینه‌سازی‌های عملی پروتکل‌های QUIC و DASH در زمینه پخش ویدیو و انتقال داده‌ها در شبکه‌های مختلف ارائه می‌شود. این پیشنهادات می‌تواند به تحقیق‌های آینده و توسعه پروتکل‌ها کمک کند و در زمینه بهبود عملکرد در شرایط مختلف شبکه، کاربردهای عملی داشته باشد.

1. **بهینه‌سازی مدیریت بافر در پروتکل‌های QUIC و DASH**

یکی از نتایج اصلی تحقیق نشان‌دهنده تفاوت‌های چشمگیر در نحوه مدیریت بافر توسط پروتکل‌های QUIC و DASH بود. به‌ویژه DASH در شرایط شبکه با نوسانات بالا، عملکرد بهتری در مدیریت بافر و جلوگیری از توقف‌های طولانی‌مدت دارد. در این راستا، تحقیق بیشتر در زمینه بهینه‌سازی مدیریت بافر در QUIC به منظور بهبود عملکرد آن در شرایط مختلف شبکه و کاهش نوسانات، پیشنهاد می‌شود.

1. **تحقیق بر روی الگوریتم‌های بهینه‌سازی نرخ Bitrate**

در این تحقیق مشاهده شد که DASH عملکرد بهتری در نرخ انتقال داده (Throughput) در مقایسه با QUIC دارد. برای بهبود عملکرد پروتکل QUIC در زمینه نرخ انتقال داده، توسعه و بهینه‌سازی الگوریتم‌های تنظیم نرخ Bitrate در QUIC به منظور به حداقل رساندن نوسانات و افزایش پایداری در نرخ‌های بالاتر پیشنهاد می‌شود.

1. **گسترش پروتکل‌های ترکیبی QUIC و DASH**

یکی از پیشنهادات مهم در راستای بهبود عملکرد، ترکیب ویژگی‌های هر دو پروتکل QUIC و DASH برای استفاده در سناریوهای مختلف است. ترکیب ویژگی‌های QUIC در کاهش تأخیر و DASH در پخش پایدار و بافر بزرگ می‌تواند عملکرد بهتری در شرایط پیچیده شبکه ایجاد کند.

پیشنهاد می‌شود که پروتکل‌های ترکیبی برای استفاده در شبکه‌هایی که نیاز به تاخیر کم و پخش پایدار و بدون وقفه دارند، طراحی و پیاده‌سازی شود.

1. **ارزیابی عملکرد در شبکه‌های 5G و محیط‌های با سرعت بالا**

با توجه به پیشرفت‌های اخیر در شبکه‌های 5G و شبکه‌های با سرعت بالا، پیشنهاد می‌شود که عملکرد پروتکل‌های QUIC و DASH در این محیط‌ها ارزیابی شود. این شبکه‌ها دارای پهنای باند بسیار بالا و زمان تأخیر بسیار کم هستند که می‌تواند چالش‌ها و فرصت‌های جدیدی برای بهینه‌سازی این پروتکل‌ها فراهم کند. انجام ارزیابی‌های عملکرد پروتکل‌ها در شبکه‌های 5G به‌ویژه در زمینه تأخیر و نرخ انتقال داده می‌تواند زمینه‌ساز بهینه‌سازی‌های بیشتر در آینده باشد.

1. **پژوهش‌های آینده در زمینه امنیت و حریم خصوصی**

با توجه به اینکه امنیت و حفظ حریم خصوصی در شبکه‌های اینترنتی اهمیت زیادی دارد، پیشنهاد می‌شود که تحقیقات آتی بر روی امنیت و حریم خصوصی پروتکل‌های QUIC و DASH متمرکز شود. به‌ویژه در زمینه انتقال داده‌های حساس مانند ویدیو و صدا، بررسی چالش‌ها و راه‌حل‌های امنیتی این پروتکل‌ها ضروری است. به‌ویژه بررسی حملات سایبری و تهدیدات امنیتی احتمالی در این پروتکل‌ها و روش‌های مقابله با آنها می‌تواند به بهبود استفاده از این پروتکل‌ها کمک کند.

مراجع

مراجع

1. A. Bhat, A. Pathania, and V. Sahula, “Performance evaluation of DASH over QUIC protocol,” *International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1817–1823, 2018.
2. A. Bhat and V. Sahula, “Enhancing video streaming QoE using QUIC transport protocol under lossy networks,” *IEEE Access*, vol. 8, pp. 141230–141240, 2020.
3. S. Mondal and S. Chakraborty, “Adaptive bitrate streaming over QUIC: Challenges and performance evaluation,” *Computer Communications*, vol. 187, pp. 22–34, 2022.
4. X. Zhang, L. Li, and K. Chen, “An experimental study on HTTP/3 and QUIC for video streaming,” *IEEE Transactions on Multimedia*, vol. 25, no. 3, pp. 812–824, 2023.
5. H. Hosseini, “Simulation-based evaluation of QUIC protocol performance in comparison with DASH over TCP,” *M.Sc. Thesis*, University of Tehran, 2024.

پيوست‌ها

پيوست الف

پیوستی در این پژوهش وجود ندارد.

**Abstract:**

With the rapid growth of online video streaming services, there is an increasing demand for network protocols that offer high performance, low latency, and resilience to network fluctuations. In this project, the QUIC protocol, a modern transport protocol developed on top of UDP, was implemented and its performance evaluated.

The main objective of this study is to analyze and assess the performance of the QUIC protocol in video streaming under various network conditions, including delay, jitter, and packet loss. To provide a better understanding of QUIC’s behavior, the DASH protocol was used as a reference for comparison.

The implementation and experiments were conducted using the Mininet network emulator on a Linux environment with Python. The results show that QUIC achieves lower startup delay and higher data throughput, demonstrating satisfactory performance in real-time video streaming. These findings indicate that QUIC can be an effective solution for enhancing user experience in future video streaming systems.

**Keywords: Video Streaming, QUIC, UDP, Dash, Mininet**



Iran University of Science and Technology

Computer Engineering Department

Video Streaming Over QUIC Protocol

A Thesis Submitted in Partial Fulfillment of the Requirement for the Degree of Bachelor in Computer Engineering

By:

Hessam Hosseini

Supervisor:

Dr. Vesal Hakami

October 2025