

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](#_Toc216492441)

[1-1- مقدمه 2](#_Toc216492442)

[فصل 2: مروري بر منابع 3](#_Toc216492443)

[2-1- مقدمه 4](#_Toc216492444)

[2-2- تعاريف، اصول و مباني نظري 4](#_Toc216492445)

[2-2-1- تيتر 4](#_Toc216492446)

[2-3- مروري بر ادبيات موضوع 5](#_Toc216492447)

[2-4- نتيجه‌گيري 5](#_Toc216492448)

[فصل 3: روش تحقيق 6](#_Toc216492449)

[3-1- مقدمه 7](#_Toc216492450)

[3-2- محتوا 7](#_Toc216492451)

[3-2-1- علت انتخاب روش 7](#_Toc216492452)

[3-2-2- تشريح كامل روش تحقيق 7](#_Toc216492453)

[فصل 4: نتايج و تفسير آنها 9](#_Toc216492454)

[4-1- مقدمه 10](#_Toc216492455)

[4-2- محتوا 10](#_Toc216492456)

[4-2-1- توليد داده‌ها 10](#_Toc216492457)

[فصل 5: جمع‌بندي و پيشنهادها 12](#_Toc216492458)

[5-1- مقدمه 13](#_Toc216492459)

[5-2- محتوا 13](#_Toc216492460)

[5-2-1- جمع‌بندي 13](#_Toc216492461)

[5-2-2- نوآوري 13](#_Toc216492462)

[5-2-3- پيشنهادها 13](#_Toc216492463)

[مراجع 15](#_Toc216492464)

[پيوست‌ها 17](#_Toc216492465)

فهرست اشکال

[شکل (2-1) نمونه شکل 5](#_Toc215371265)

[شکل (3-1) زيرنويس شکل 8](#_Toc215371266)

[شکل (4-1) زيرنويس شکل 11](#_Toc215371267)

[شکل (5-1) زيرنويس شکل 14](#_Toc215371268)

فهرست جداول

[جدول (2-1) نتيجه بررسي پرسش نامه ها در ارتباط با عوامل موثر 4](#_Toc215371272)

[جدول (3-1) بالانويس جدول 8](#_Toc215371273)

[جدول (4-1) بالانويس جدول 10](#_Toc215371274)

[جدول (5-1) بالانويس جدول 14](#_Toc215371275)

فهرست علائم اختصاري

شتاب گرانش  *a* (*m/s2*)

نيرو *F* (*N*)

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: Traffic-Cross

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.

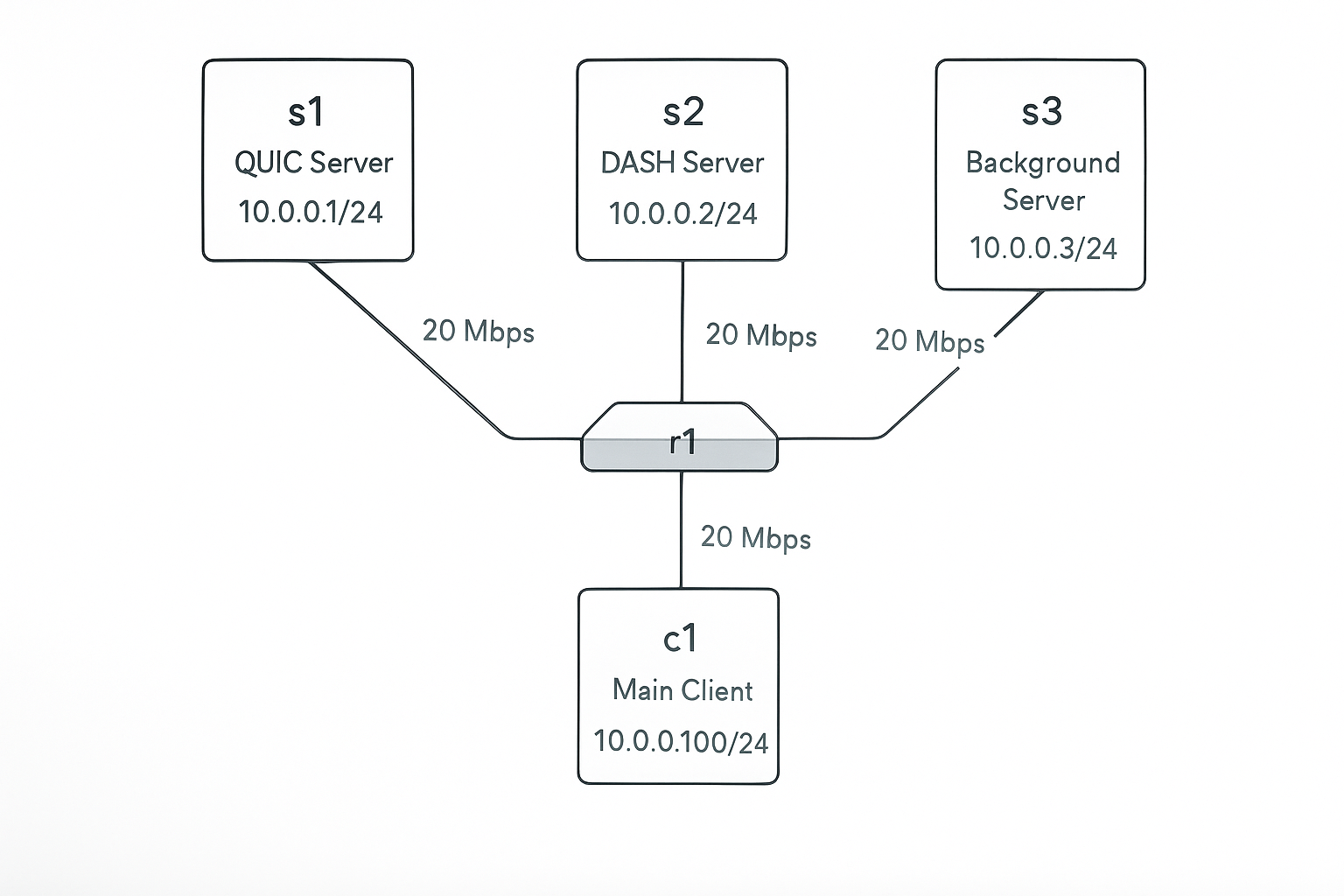
در آخر هم به ترتیب ابتدا اسکریپت quic\_client.py و سپس اسکریپت dash\_clinet.py را اجرا میکند.

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%, Traffic-Cross: random
* S4: Bandwidth: 5 Mbps, Delay: 80ms, Jitter: 30ms, Packet-Loss: 1%
* S5: Bandwidth: 2 Mbps, Delay: 80ms, Jitter: 30ms, Packet-Loss: 3%, Traffic-Cross: 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%, Traffic-Cross: random
* S9: Bandwidth: 10Mbps, Delay: 40ms, Jitter: 30ms, Packet-Loss: 1%, Traffic-Cross: random
* S10: Bandwidth: 10Mbps, Delay: 40ms, Jitter: 10ms, Packet-Loss: 3%

1. نتايج و تفسير آنها
   1. مقدمه

ارائه‌ي داده‌ها، نتايج و تحليل و تفسير آنها در فصل چهارم ارائه مي‌شود. تفاوت، تضاد يا تطابق بين نتايج تحقيق با نتايج ديگر محققان بايد ذكر شود. تفسير و تحليل نتايج نبايد بر اساس حدس و گمان باشد، بلكه بايد برمبناي نتايج عملي استخراج‌شده از تحقيق و يا استناد به تحقيقات ديگران باشد. در ارائه‌ي نتايج با توجه به راهنماي كلي نگارش فصل ها،، تا حد امكان تركيبي از نمودار و جدول استفاده شود. با توجه به حجم و ماهيت تحقيق و با صلاحديد استاد راهنما، اين فصل مي‌تواند تحت عنواني ديگر بيايد يا به دو فصل جداگانه با عناوين مناسب، تفكيك شود. در صورتي که حجم داده‌ها زياد باشد، بهتر است به صورت نمودار يا در قالب ضميمه ارائه نشده و فقط نمونه‌ها در متن آورده شود. اين فصل فقط بايد به جمع‌بندي دست‌آوردهاي فصل‌هاي چهارم و پنجم محدود و از ذكر موارد جديد در آن خودداري شود. در عنوان اين فصل، به جاي کلمه‌ي «تفسير» مي‌توان از واژگان «بحث» و «تحليل» هم استفاده کرد. اين فصل حدود 40 صفحه است.

* 1. محتوا
     1. توليد داده‌ها

‌ارائه‌ي داده‌ها و نتايج به‌ صورت تابعي از پارامترهاي مستقل است.

بالانويس جدول

جدول

متن

شکل

زيرنويس شکل

متن

فرمول

متن

1. جمع‌بندي و پيشنهادها
   1. مقدمه

ارائه‌ي خلاصه‌اي از يافته‌هاي تحقيق جاري است. اين فصل مي‌تواند حاوي يک مقدمه شامل مروري اجمالي بر مراحل انجام تحقيق باشد (حدود يک صفحه). مطالب پاراگراف‌بندي شود و هر پاراگراف به يك موضوع مستقل اختصاص يابد. فقط به ارائه‌ي يافته‌ها و دست‌آوردها بسنده شود و از تعميم بي‌مورد نتايج خودداري شود. از ارائه‌ي جداول و نمودارها اجتناب شود. از ارائه‌ي عناوين كلي در حوزه‌ي تحقيق و پيشنهاد تحقيقات آتي خودداري شود و كاملاً در چارچوب و زمينه‌ي مربوط به تحقيق جاري باشد. اين فصل حدود 5 صفحه است.

* 1. محتوا

به ترتيب شامل موارد زير است:

* + 1. جمع‌بندي

خلاصه‌اي از تمام يافته‌ها و دست‌آوردهاي تحقيق جاري است.

* + 1. نوآوري

نوآوري تحقيق را بر اساس يافته‌هاي آن تشريح مي‌کند.

* + 1. پيشنهادها

عناوين و موضوعات پيشنهادي را براي تحقيقات آتي بيشتر در زمينه‌ي مورد بحث در آينده ارائه مي‌کند.

بالانويس جدول

جدول

متن

شکل

زيرنويس شکل

متن

فرمول

متن

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

**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