

به نام خدا

پروژه درس شبکه‌های کامپیوتری

(نیم سال دوم ۱۴۰۱)

## پیاده‌سازی یک سرویس انتقال داده بر پایه رویه نظیر به نظیر<sup>۱</sup>

دانشکده مهندسی کامپیوتر دانشگاه صنعتی امیرکبیر

بهار ۱۴۰۲

- 
- ۱- مقدمه ..... ۲
  - ۲- پروژه ..... ۴
    - ۲-۱ Peer ..... ۵
      - ۲-۲-۱ سرور مدیریت آدرس peer ها (STUN Server) ..... ۶
      - ۲-۲-۲ ارسال مشخصات peer ..... ۶
      - ۲-۲-۳ دریافت لیستی از تمامی peer ها ..... ۶
      - ۲-۲-۴ دریافت اطلاعات یک peer ..... ۷
    - ۲-۳ داده‌های سرویس ..... ۸
      - ۲-۳-۱ داده متنی ..... ۸
      - ۲-۳-۲ داده تصویری ..... ۸
  - ۳- معماری سیستم ..... ۱۰
  - ۴- امتیازی ..... ۱۱

---

<sup>۱</sup> Peer to Peer

## نکات انجام و ارزیابی پروژه

- انجام پروژه باید به صورت فردی باشد، در صورت مشاهده کدهای مشابه به هم یا هرنوع تقلب، نمره پروژه برای هر دو طرف صفر در نظر گرفته خواهد شد.
- جهت ارزیابی تسلط شما بر پروژه، پروژه تحویل مجازی خواهد داشت. بعد از ددلاین پروژه، زمان‌های تحویل پروژه مشخص خواهند شد.
- ددلاین انجام پروژه درس جمعه ۵ خرداد ساعت ۲۳:۵۹ می‌باشد. امکان تمدید پروژه به دلیل تداخل با امتحان‌های پایان ترم وجود **ندارد**.
- زبان برنامه‌نویسی پیشنهادی برای انجام پروژه پایتون می‌باشد، اما شما مجاز به استفاده از دیگر زبان‌های برنامه‌نویسی نیز می‌باشید.
- نوشتن گزارش کار برای پروژه الزامی می‌باشد. گزارش کار شما باید شامل توضیح فایل‌ها، توضیح معماری، توضیح پروتکل‌های پیاده‌سازی شده، نحوه اجرای برنامه‌ها، اسکرین‌شات از نتایج برنامه و توضیح کتابخانه‌های مورد استفاده می‌باشد.
- ارزیابی پروژه شما بر اساس ارائه و گزارش کار پروژه می‌باشد. از آنجایی که ارائه در جهت ارزیابی تسلط شما می‌باشد، و بخش دیگر نمره از روی گزارش شما داده خواهد شد؛ گزارش کار خود را کامل و مرتب بنویسید.
- هر سوال یا ابهام در رابطه با پروژه را از طریق راه‌های ارتباطی با تدریس‌یار، که در سامانه کورسز قرار داده شده‌است، بیان کنید.
- پس از اتمام پروژه، گزارش کار خود را به همراه فایل‌های پروژه در یک پوشه زیپ شده به فرمت student\_number:zip در سامانه کورسز بارگذاری کنید.

## مقدمه

امروزه اکثر سیستم‌هایی که برای انتقال مدیا<sup>۲</sup> (داده‌هایی نظیر تصویر یا ویدئو) پیاده‌سازی شده‌اند، از رویه نظیر به نظیر برای انتقال داده‌ها استفاده می‌کنند. زمانی که از رویه نظیر به نظیر استفاده می‌کنیم، داده‌ها مستقیم از فرستنده به گیرنده ارسال می‌شوند و دیگر نیازی به سرور مرکزی<sup>۳</sup> یا نقطه واسطه<sup>۴</sup> برای تبادل اطلاعات نخواهیم داشت. این رویه از نظر سرعت تبادل پیام بسیار سریعتر از دیگر رویه‌ها می‌باشد و معمولاً برای پیاده‌سازی سیستم‌های بیدرنگ<sup>۵</sup> مورد استفاده قرار می‌گیرد. در این رویه، هر کاربر که در لبه شبکه<sup>۶</sup> وجود دارد را یک peer می‌نامیم.

چالش‌های اصلی این رویه ارتباط میان دو همتا و نحوه فرستادن اطلاعات می‌باشند. از آنجایی که هر همتا، آدرس مخصوص به خود در شبکه را دارد؛ یافتن اطلاعات هر همتا و برقراری ارتباط میان دو جفت، ممکن است مشکل‌ساز باشد. این مشکلات ناشی از ثابت نبودن آدرس هر همتا در شبکه می‌باشد. برای اینکه دو همتا بتوانند آدرس یکدیگر را بیابند و با یکدیگر ارتباط بگیرند؛ نیازمند یک عامل با آدرس ثابت می‌باشند که همواره در دسترس همه همتا می‌باشد. همچنین باید اطلاعات شبکه‌ای آن‌ها را ذخیره کند و این اطلاعات را در اختیار مابقی همتاها قرار دهد؛ همچنین در صورت نیاز آن‌ها را به‌روزرسانی کند.

توجه کنید که این سرور هیچ داده‌ای از انتقال مدیا را از خود عبور نمی‌دهد و تنها اطلاعات شبکه‌ای همتاها را مدیریت می‌کند.

برای کسب اطلاعات بیشتر در رابطه با ارتباط نظیر به نظیر به این [پیوند](#) مراجعه فرمایید.

<sup>۲</sup> media

<sup>۳</sup> central server

<sup>۴</sup> middle node

<sup>۵</sup> real-time

<sup>۶</sup> Network Edge

## پروژه

در این پروژه شما باید یک سرویس برای انتقال متن و تصویر را پیاده‌سازی کنید. معماری مورد استفاده در این پروژه، نظیر به نظیر می‌باشد. این سرویس از دو بخش تشکیل شده است، همتاها<sup>۷</sup> و سرور مدیریت آدرس همتاها (STUN server). هر همتا یک کاربر از سرویس ما می‌باشد و سرور استان برای ذخیره‌سازی اطلاعات شبکه‌ای کاربران مورد استفاده قرار می‌گیرد. سرویس شما باید این امکان را بدهد تا کاربران به شبکه متصل بشوند و بتوانند به صورت بیدرنگ، داده‌های متنی و تصویری را به اشتراک بگذارند.

هدف از این پروژه آشنایی با رویه نظیر به نظیر، استفاده از سوکت‌های TCP و UDP، آشنایی با سرور HTTP و کار کردن با حافظه نهان Redis می‌باشد.

برای این پروژه شما نیازی به طراحی یک رابط کاربری ساده برای آزمایش سرویس به هنگام ارائه می‌باشید، اما نیازی نیست رابط کاربری شما گرافیکی باشد. لازم به ذکر است که طراحی رابط کاربری گرافیکی دارای **نمره امتیازی** می‌باشد.

در ادامه بخش‌های اصلی پروژه، معماری مورد استفاده و داده‌های پروژه را بررسی خواهیم کرد.

---

<sup>۷</sup> peers

## همتا<sup>۸</sup>

هر همتا (peer) یکی از کاربران سیستم می‌باشد. هر همتا در ابتدای ایجاد، با استفاده از درخواست‌های HTTP باید خود را به STUN server معرفی کند و آدرس خودش را برای سرور ارسال کند. سپس باید بتواند لیستی از دیگر کاربران موجود در شبکه را دریافت کند. در ادامه یک کاربر را انتخاب می‌کند و با استفاده از سوکت UDP، سعی می‌کند تا آن کاربر را صدا کند<sup>۹</sup> و درخواست شروع ارتباط را ارسال کند. در صورتی که طرف دیگر ارتباط برخط<sup>۱۰</sup> بود، این امکان را دارد تا درخواست را تایید یا رد کند. در صورت تایید درخواست، دو همتا به برقراری ارتباط و انتقال داده‌ها مشغول می‌شوند.

شما باید سیستم را به شکلی طراحی کنید تا در صورت بروز مشکلاتی از قبیل قطع شدن ارتباط، ارسال نشدن داده‌ها یا ...، به آن‌ها واکنش منطقی بدهد. به عنوان مثال در صورتی که طرف دیگر ارتباط، به هنگام ارسال داده، قطع شد، باید ارسال داده‌ها متوقف بشود و پیام مناسبی به کاربر نمایش داده بشود.

---

<sup>۸</sup> peer

<sup>۹</sup> call

<sup>۱۰</sup> online

## سرور مدیریت آدرس همتاها (STUN server)

این سیستم یک سرور HTTP می‌باشد. در این سرور اطلاعات کاربران سیستم در یک حافظه نهان ذخیره می‌شود. این حافظه نهان یک کلاستر Redis می‌باشد که داده‌های درون آن به صورت جفت‌های کلید-ارزش<sup>۱۱</sup> ذخیره شده‌اند. برای اطلاعات بیشتر از این حافظه و نحوه نصب و کار با آن به این [پیوند](#) مراجعه کنید.

این سرور HTTP دارای سه نقطه پایانی<sup>۱۲</sup> می‌باشد که به صورت زیر تعریف شده‌اند. نقاط پایانی توابعی هستند که درخواست‌های HTTP کاربران را مدیریت کرده و پاسخ مناسبی را در نتیجه درخواست ارسال می‌کنند.

### ارسال مشخصات همتا

در این نقطه پایانی، کاربر باید یک درخواست HTTP، از نوع POST، با داده‌های زیر در بدنه درخواست ارسال کند. سپس این اطلاعات باید در حافظه نهان ردیس نوشته بشوند.

- نام یکتای کاربری
- آدرس اینترنتی

### دریافت لیستی از تمامی همتاها<sup>۱۳</sup>

در این نقطه پایانی، کاربر با ارسال یک درخواست GET، لیستی از نام‌های کاربری افراد موجود در شبکه را دریافت می‌کند.

---

<sup>۱۱</sup> key-value

<sup>۱۲</sup> endpoint

<sup>۱۳</sup> peers

## دریافت اطلاعات یک همتا

در این نقطه پایانی، کاربر با استفاده از یک درخواست گت، یک نام کاربری را ارسال می‌کند و در پاسخ اطلاعات موردنیاز برای ارتباط با همتای موردنظر خود را دریافت می‌کند.

توجه کنید که پروتکل ارسال و دریافت داده‌ها در این سرور باید JSON باشد. برای یافتن اطلاعات بیشتر در رابطه با پروتکل، به این [پیوند](#) مراجعه نمایید.

همچنین توجه داشته باشید که کدهای پاسخ به درخواست‌ها باید منطقی و طبق مطالبی که در درس آموختید باشند. به عنوان مثال برای درخواست موفقیت آمیز در ثبت اطلاعات باید کدهای ۲۰۰ یا ۲۰۱ یا ۲۰۲ بازگردانده بشوند.

برای ایجاد یک سرویس HTTP در زبان پایتون به این [پیوند](#) مراجعه نمایید. برای ایجاد سرویس HTTP استفاده از فریم‌ورک‌های<sup>۱۴</sup> زبان‌های برنامه‌نویسی مجاز نمی‌باشد.

---

<sup>۱۴</sup> framework

## داده‌های سرویس

در این سرویس ما دو نوع داریم؛ داده متنی و داده تصویری. شیوه ارسال این داده‌ها به هنگام ارتباط با یکدیگر متفاوت است.

### داده‌های متنی

برای ارسال داده متنی، از سوکت TCP استفاده می‌کنیم و تمام متن را در یک انتقال ارسال می‌کنیم. در صورتی که متن در طرف گیرنده دریافت نشده باشد، باید مجدداً پیام را ارسال کنیم. شما نیازمند طراحی یک پروتکل برای ارسال این داده می‌باشید.

### داده‌های تصویری

از آنجایی که داده‌های تصویری جزو داده‌های مدیا<sup>۱۵</sup> به حساب می‌آیند، امکان ارسال مستقیم آن‌ها در شبکه وجود ندارد. شما باید در ابتدا، یک عکس را به ماتریسی از اعداد تبدیل کنید و سپس ماتریس را به صورت جزیه‌جز با استفاده از یک سوکت UDP برای گیرنده ارسال کنید. از آنجایی که این ارتباط از نوع UDP می‌باشد، ممکن است داده‌ها به ترتیب دریافت نشوند یا حتی از بین بروند؛ در نتیجه شما باید پروتکلی طراحی کنید تا بتوانید در گیرنده، تمام بخش‌های ماتریس را به ترتیب دریافت کنید و سپس ماتریس اصلی را تشکیل بدهید و تصویر اصلی را بازسازی کنید.

برای تبدیل تصاویر به ماتریس اعداد، از کتابخانه [پیلو](#)<sup>۱۶</sup> در زبان برنامه‌نویسی پایتون استفاده کنید.

---

<sup>۱۵</sup> media

<sup>۱۶</sup> Pillow



راهنمایی: می‌توانید از [روش‌هایی](#) که برای انتقال پکت‌ها در درس شبکه‌های کامپیوتری یاد گرفتید استفاده کنید.

در تصویر زیر می‌توانید مشاهده کنید که ساختار عکس پس از تبدیل شدن به ماتریس باید چگونه باشد و در ادامه سطر به سطر آن باید به صورت جدا برای گیرنده فرستاده بشوند.

$$\begin{bmatrix} (r_{1,1}, g_{1,1}, b_{1,1}) & \cdots & (r_{1,m}, g_{1,m}, b_{1,m}) \\ \vdots & \ddots & \vdots \\ (r_{n,1}, g_{n,1}, b_{n,1}) & \cdots & (r_{n,m}, g_{n,m}, b_{n,m}) \end{bmatrix}$$

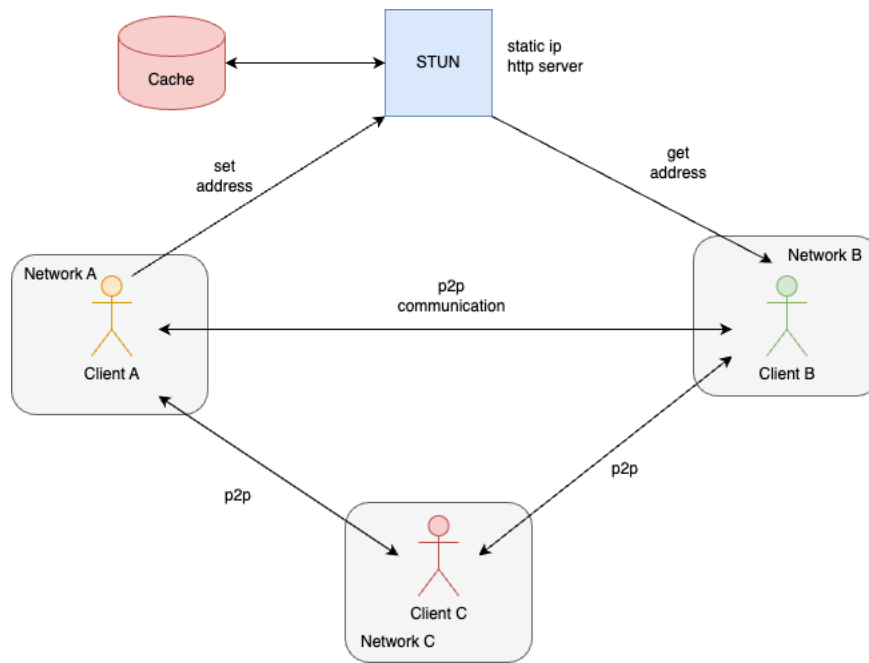
→

$$[(r_{1,1}, g_{1,1}, b_{1,1}) \quad \cdots \quad (r_{1,m}, g_{1,m}, b_{1,m})], \dots, [(r_{n,1}, g_{n,1}, b_{n,1}) \quad \cdots \quad (r_{n,m}, g_{n,m}, b_{n,m})]$$

## معماری سیستم

معماری سیستم موردنظر به صورت زیر می‌باشد. شما می‌توانید در صورت نیاز معماری سیستم را تغییر دهید اما باید رویه نظیر به نظیر را در معماری خودتان داشته باشید.

همانطور که مشاهده می‌کنید، آدرس سرور STUN ثابت است؛ اما همتاها آدرس‌های متفاوتی دارند، در نتیجه نیاز است تا همواره اطلاعات آن‌ها در حافظه‌نهمان به‌روزرسانی شده باشد.



- نمودار بلوکی پروژه

## بخش امتیازی

در بخش امتیازی شما باید از ابزار مجازی‌سازی داکر<sup>۱۷</sup> استفاده کنید. داکر یک ابزار برای ایجاد کانتینر<sup>۱۸</sup> است که هر کانتینر در اصل شامل یک پردازنده ایزوله شده در سیستم‌عامل ما می‌باشد. با استفاده از داکر شما می‌توانید شبکه و کانتینرهای مجازی ایجاد کنید و یک شبیه‌سازی از دنیای واقعی را انجام بدهید. برای اطلاعات بیشتر در رابطه با داکر به [سایت اصلی](#) این ابزار مراجعه کنید.

در این بخش با استفاده از ابزار داکر، یک شبکه مجازی ایجاد کنید و تمام سرویس‌ها را بر روی داکر اجرا کنید. دقت کنید که استفاده از فایل داکر-کامپوز<sup>۱۹</sup> مجاز نمی‌باشد و شما باید کانتینرهای خودتان را به صورت دستی به شبکه اضافه کنید. دقت کنید که سرور STUN و peer ها و حافظه نهان باید به صورت کانتینر در یک شبکه قرار بگیرند و هیچ بخش از سیستم نباید خارج از این شبکه باشد.

چالش اصلی این بخش، گرفتن IP هر همتا و اتصال کلاینت‌های مجازی به یکدیگر می‌باشد. این بخش شبیه‌ساز دنیای واقعی از پروژه ما می‌باشد.

---

<sup>۱۷</sup> Docker

<sup>۱۸</sup> Container

<sup>۱۹</sup> docker-compose

راهنمایی: برای انجام این بخش ابتدا به سراغ ایجاد کانتینرها بروید. برای دسترسی به نمونه‌ها، به پیوندهای زیر مراجعه کنید. سپس در ادامه یک شبکه ایجاد کنید و با مراجعه به پیوندهای زیر، سعی کنید تا تمام کانتینرهای خود را به داخل یک شبکه انتقال دهید.

- [ایجاد شبکه](#)
- [اتصال کانتینرها به شبکه](#)
- [دریافت آی-پی کانتینر](#)

موفق باشید.