به نام خدا

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

شبکه‌های کامپیوتری (بهار ۱۴۰۳)

پروژه پایانی

سیستم چت Peer-to-Peer با استفاده از HTTP و TCP

امروزه اکثر سیستم‌هایی که برای انتقال مدیا (داده‌هایی نظیر تصویر یا ویدئو) پیاده‌سازی شده‌اند، از رویه نظیر به نظیر (P2P) برای انتقال داده‌ها استفاده می‌کنند. در این رویه، داده‌ها مستقیم از فرستنده به گیرنده ارسال می‌شوند و دیگر نیازی به سرور مرکزی یا نقطه واسط برای تبادل اطلاعات نخواهیم داشت. این روش از نظر سرعت تبادل پیام بسیار سریع‌تر از دیگر روش‌ها می‌باشد و معمولاً برای پیاده‌سازی سیستم‌های بیدرنگ (real-time) مورد استفاده قرار می‌گیرد. در این رویه، هر کاربر که در لبه شبکه وجود دارد را یک همتا (peer) می‌نامیم.

استفاده از سیستم‌های نظیر به نظیر مزایای بسیاری دارد، اما چالش‌های خاص خود را نیز به همراه دارد. یکی از چالش‌های اصلی این رویه، ارتباط میان دو همتا و نحوه فرستادن اطلاعات می‌باشد. 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 برقرار کنند، نیازمند یک عامل با آدرس ثابت هستند که همواره در دسترس همه همتاها باشد. این عامل وظیفه دارد اطلاعات شبکه‌ای همتاها را ذخیره کرده و این اطلاعات را در اختیار مابقی همتاها قرار دهد و در صورت نیاز آنها را به‌روزرسانی کند. این سرور هیچ داده‌ای از انتقال مدیا را از خود عبور نمی‌دهد و تنها اطلاعات شبکه‌ای همتاها را مدیریت می‌کند.

هدف از این پروژه، پیاده‌سازی یک سیستم چت نظیر به نظیر است که در آن کاربران ابتدا با استفاده از پروتکل HTTP به یک سرور مرکزی متصل می‌شوند و پس از ثبت نام و دریافت اطلاعات همتا، به صورت مستقیم و از طریق پروتکل TCP به هم متصل شده و پیام‌های خود را به صورت real-time ارسال و دریافت می‌کنند.

در این پروژه، شما باید یک سرویس برای انتقال پیام را پیاده‌سازی کنید. معماری مورد استفاده در این پروژه، نظیر به نظیر (P2P) می‌باشد. این سرویس از دو بخش تشکیل شده است: همتاها و سرور مدیریت آدرس همتاها (STUN server). هر همتا یک کاربر از سرویس ما می‌باشد و سرور STUN برای ذخیره‌سازی اطلاعات شبکه‌ای کاربران مورد استفاده قرار می‌گیرد. سرویس شما باید این امکان را بدهد تا کاربران به شبکه متصل شوند و بتوانند به صورت بیدرنگ، پیام‌های خود را به اشتراک بگذارند.

برای این پروژه شما نیاز 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.

همتا

هر همتا (peer) یکی از کاربران سیستم می‌باشد. هر همتا در ابتدای ایجاد، با استفاده از درخواست‌های HTTP باید خود را به سرور مدیریت آدرس همتاها (STUN server) معرفی کرده و آدرس خود 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 TCP، سعی می‌کند تا آن کاربر را صدا کند و درخواست شروع ارتباط را ارسال کند. در صورتی که طرف دیگر ارتباط آنلاین باشد، این امکان را دارد تا درخواست را تایید یا رد کند. در صورت تایید درخواست، دو همتا به برقراری ارتباط و انتقال پیام مشغول می‌شوند.

سیستم باید به گونه‌ای طراحی شود که در صورت بروز مشکلاتی از قبیل قطع شدن ارتباط یا ارسال نشدن داده‌ها، واکنش منطقی نشان دهد. به عنوان مثال، در صورتی که طرف دیگر ارتباط به هنگام ارسال داده قطع شد، باید ارسال داده‌ها متوقف شود و پیام مناسبی به کاربر نمایش داده شود.

فاز اول: پیاده‌سازی سرور مدیریت آدرس همتاها (STUN Server) و اتصال همتاها

در این فاز شما باید علاوه بر پیاده‌سازی سرور مدیریت آدرس همتاها، پیاده‌سازی بخش‌های مربوط به اتصال همتاها به سرور را هم انجام دهید.

سرور مدیریت آدرس همتاها (STUN Server)

این سیستم یک سرور HTTP می‌باشد که اطلاعات کاربران سیستم را مدیریت می‌کند. در این سرور، اطلاعات کاربران در یک حافظه نهان ذخیره می‌شود. این حافظه نهان می‌تواند Redis باشد که داده‌های درون آن به صورت جفت‌های کلید-ارزش ذخیره شده‌اند. استفاده از Redis به عنوان حافظه نهان در این پروژه امتیازی است، اما می‌تواند به بهبود عملکرد سیستم کمک کند. برای اطلاعات بیشتر در مورد Redis و نحوه نصب و کار با آن به این پیوند مراجعه کنید. در صورتیکه نمی‌خواهید از Redis استفاده کنید می‌توانید اطلاعات را در حافظه اصلی ذخیره کنید.

این سرور HTTP دارای سه نقطه پایانی (Endpoints) 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:

1. ثبت نام مشخصات همتا

- **مسیر:** `register`
- **روش:** `POST`
- **توضیح:** 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 Redis نوشته شوند.
  - نام یکتای کاربری
  - آدرس اینترنتی (IP) و پورت

2. دریافت لیستی از تمامی همتاها

- **مسیر:** `peers`
- **روش:** `GET`
- **توضیح:** در این نقطه پایانی، کاربر با ارسال یک درخواست GET، لیستی از همتاهای موجود در شبکه را دریافت می‌کند.

3. دریافت اطلاعات یک همتا

- **مسیر:** `peerinfo`
- **روش:** `GET`
- **توضیح:** در این نقطه پایانی، کاربر با استفاده از یک درخواست GET، یک نام کاربری را ارسال می‌کند و در پاسخ اطلاعات موردنیاز برای ارتباط با همتای موردنظر خود را دریافت می‌کند.
- **پارامترهای ورودی:**
  - نام کاربری: `username`

توجه کنید که پروتکل ارسال و دریافت داده‌ها در این سرور باید JSON باشد. همچنین، کدهای پاسخ به درخواست‌ها باید منطقی و طبق مطالبی که در درس آموختید باشند. به عنوان مثال، برای درخواست موفقیت‌آمیز در ثبت اطلاعات باید کدهای 200، 201 یا 202 بازگردانده شوند.

فاز دوم: پیاده‌سازی ارتباط مستقیم بین همتاها با استفاده از TCP

در فاز دوم این پروژه، پس از پیاده‌سازی سرور مدیریت آدرس همتاها (STUN Server) و اتصال همتاها به سرور، تمرکز بر روی برقراری ارتباط مستقیم بین همتاها با استفاده از پروتکل TCP خواهد بود. در این فاز، هر همتا (کاربر) می‌تواند با همتای دیگر به صورت مستقیم ارتباط برقرار کرده و شروع به ارسال پیام کند.

انتخاب همتا و شروع ارتباط

هر همتا که قبلاً با استفاده از HTTP به سرور متصل شده و اطلاعات شبکه‌ای خود را ثبت کرده است، می‌تواند لیستی از همتاهای دیگر را از سرور دریافت کند. با دریافت این لیست، همتا می‌تواند یک همتای دیگر را انتخاب کرده و با استفاده از سوکت TCP، درخواست شروع ارتباط را به آن ارسال کند. این درخواست شامل آدرس IP و پورت همتا است که از طریق سرور دریافت شده است.

برای برقراری ارتباط مستقیم، هر همتا باید بتواند به عنوان یک کلاینت عمل کرده و به همتای دیگر متصل شود، و همزمان به عنوان یک سرور عمل کرده و منتظر دریافت درخواست‌های اتصال از همتاهای دیگر باشد. برای انجام این کار، هر همتا یک سرور TCP را اجرا می‌کند که به طور مداوم منتظر دریافت درخواست‌های اتصال از دیگر همتاها است. به محض دریافت یک درخواست اتصال، همتا می‌تواند این درخواست را تایید یا رد کند. در صورت تایید درخواست، ارتباط TCP بین دو همتا برقرار شده و آن‌ها می‌توانند پیام‌های خود را به صورت real-time به اشتراک بگذارند.

سیستم باید به گونه‌ای طراحی شود که در صورت بروز مشکلاتی از قبیل قطع شدن ارتباط یا ارسال نشدن داده‌ها، واکنش منطقی نشان دهد. به عنوان مثال، در صورتی که طرف دیگر ارتباط به هنگام ارسال داده قطع شود، باید ارسال داده‌ها متوقف شود و پیام مناسبی به کاربر نمایش

داده شود. همچنین، در صورت بروز هرگونه خطا در ارتباط، سیستم باید بتواند کاربر را از وضعیت خطا مطلع سازد.

در صورت پیاده‌سازی امکان ارسال فایل بین همتاها، این ویژگی به عنوان یک قابلیت اختیاری نمره امتیازی خواهد داشت. این قابلیت به کاربران اجازه می‌دهد که علاوه بر پیام‌های متنی، فایل‌های مختلف را نیز به یکدیگر ارسال کنند. پیاده‌سازی این ویژگی نیازمند مدیریت فایل‌ها و انتقال آن‌ها از طریق پروتکل TCP می‌باشد.

بخش امتیازی: استفاده از ابزار مجازی‌سازی Docker

در این پروژه، استفاده از Docker برای مجازی‌سازی و اجرای کانتینرها به عنوان یک ویژگی اختیاری نمره امتیازی خواهد داشت. Docker یک ابزار قدرتمند برای ایجاد و مدیریت کانتینرها است که هر کانتینر در اصل شامل یک پردازه ایزوله شده در سیستم‌عامل می‌باشد. با استفاده از Docker، شما می‌توانید شبکه و کانتینرهای مجازی ایجاد کرده و یک شبیه‌سازی از دنیای واقعی را انجام دهید.

مزایای استفاده از Docker

- **ایزولاسیون:** هر کانتینر به طور کامل از دیگر کانتینرها و سیستم‌عامل میزبان ایزوله شده است، که این امر به افزایش امنیت و پایداری سیستم کمک می‌کند.
- **پرتابل بودن:** Docker به شما امکان می‌دهد تا برنامه‌ها و وابستگی‌های آن‌ها را در قالب یک کانتینر بسته‌بندی کنید و به راحتی آن‌ها را در محیط‌های مختلف اجرا کنید.
- **مدیریت آسان:** با استفاده از Docker، مدیریت و استقرار برنامه‌ها بسیار ساده‌تر و سریع‌تر می‌شود، زیرا نیاز به تنظیمات پیچیده کاهش می‌یابد.

کاربرد Docker در پروژه

در این پروژه، می‌توانید از Docker برای مجازی‌سازی و اجرای سرور مدیریت آدرس همتاها (STUN Server) و همتاها (Peers) استفاده کنید. این کار به شما امکان می‌دهد تا شبکه‌ای از همتاها را در محیطی مجازی و کنترل شده ایجاد کرده و ارتباطات بین آن‌ها را شبیه‌سازی کنید.

نکات انجام و ارزیابی پروژه

- پروژه باید به صورت فردی انجام شود. در صورت مشاهده کدهای مشابه یا هر نوع تقلب، نمره پروژه برای هر دو طرف صفر در نظر گرفته خواهد شد.
- پروژه تحویل مجازی خواهد داشت. بعد از ددلاین پروژه، زمان‌های تحویل پروژه مشخص خواهند شد.

- **زبان برنامه‌نویسی:**
  - زبان برنامه‌نویسی پیشنهادی برای انجام پروژه، پایتون می‌باشد. با این حال، شما مجاز به استفاده از دیگر زبان‌های برنامه‌نویسی نیز می‌باشید.