



دانشگاه صنعتی شریف
دانشکده مهندسی کامپیوتر

عنوان:

پروژه شبکه‌های کامپیوتری
Computer Network Project

شماره درس
۴۰۴۴۳

تاریخ تحویل
۱۴۰۳/۱۱/۱۸

موضوع پروژه
شبکه تورنت (Torrent)

استاد درس
دکتر سید امیر مهدی صادق زاده مسگر

نیم سال اول سال تحصیلی ۱۴۰۳-۱۴۰۴

۱ موضوع پروژه: شبکه تورنت (Torrent)

در این پروژه، شما به طراحی و پیاده‌سازی یک شبکه مشابه با تورنت خواهید پرداخت که از پروتکل‌های TCP و UDP استفاده می‌کند. این شبکه به طور خاص برای اشتراک‌گذاری فایل‌ها به صورت peer-to-peer و غیرمتمرکز طراحی خواهد شد. در این پروژه شما با نحوه تعامل و ارتباط بین کاربران شبکه آشنا می‌شوید و فرآیندهایی همچون دانلود، آپلود و به‌روزرسانی اطلاعات در شبکه را پیاده‌سازی خواهید کرد.

۱.۱ Peer چیست؟

در شبکه‌های تورنت، به هر دستگاهی که به شبکه متصل می‌شود و به اشتراک‌گذاری فایل‌ها می‌پردازد، peer گفته می‌شود. هر peer می‌تواند بخشی از یک فایل را دانلود کرده و آن را برای کاربران دیگر به اشتراک بگذارد. در واقع، هر peer همزمان نقش دانلودکننده و آپلودکننده را ایفا می‌کند.

۲.۱ Tracker چیست؟

trackerها سرورهای متمرکزی هستند که نقش هماهنگ‌کننده و مدیریت اطلاعات در شبکه تورنت را دارند. این سرورها اطلاعاتی مانند این که هر peer چه بخش‌هایی از فایل‌ها را دارد و چه کسانی در شبکه حضور دارند را ذخیره می‌کنند. به کمک peer، tracker می‌تواند به یکدیگر متصل شوند و فرآیند دانلود و آپلود به طور کارآمد انجام شود. هدف اصلی این پروژه، فهم عمیق‌تر نحوه عملکرد شبکه‌های غیرمتمرکز و پروتکل‌های ارتباطی است که برای اشتراک‌گذاری فایل‌ها در چنین شبکه‌هایی استفاده می‌شود.

۲ نحوه عملکرد شبکه تورنت:

فهمیدیم که در شبکه تورنت، هر کاربر یا دستگاهی که به شبکه متصل است و اقدام به دانلود یا آپلود فایل‌ها می‌کند، peer نامیده می‌شود. زمانی که یک peer تمامی بخش‌های یک فایل را دانلود کرده و به کاربران دیگر آن را ارسال می‌کند، تبدیل به یک seeder می‌شود. این فرآیند باعث توزیع غیرمتمرکز فایل‌ها بین کاربران و تسریع در اشتراک‌گذاری می‌شود. بیایید با یک مثال بهتر، عملکرد مرحله‌ای در این شبکه را بررسی کنیم:

۱.۲ اشتراک‌گذاری فایل توسط اولین کاربر (Seeder)

فرض کنید کامپیوتر X می‌خواهد یک فایل ویدیویی به نام ویدیو A را در شبکه تورنت به اشتراک بگذارد. این کاربر ابتدا فایل ویدیو A را به بخش‌های کوچک تقسیم می‌کند، به طوری که هر بخش اندازه‌ای مانند ۲ مگابایت دارد. این بخش‌ها با نام chunk شناخته می‌شوند. بعد از تقسیم فایل، کامپیوتر X به tracker اعلام می‌کند که تمام chunkهای فایل ویدیو A را دارد و آماده است که آنها را با دیگر کاربران به اشتراک بگذارد. tracker به عنوان یک واسطه عمل می‌کند و اطلاعات مربوط به موجودیت chunkها و peerها در شبکه را در اختیار دیگر کاربران قرار می‌دهد.

۲.۲ دریافت فایل توسط کاربر جدید (Peer)

حالا فرض کنید که کامپیوتر Y به شبکه متصل می‌شود و می‌خواهد فایل ویدیو A را دانلود کند. برای شروع، کامپیوتر Y از tracker درخواست می‌کند که فهرستی از peerها و اطلاعاتی درباره اینکه هر کدام از آنها کدام بخش از فایل ویدیو A را دارند، ارسال کند.

tracker پس از دریافت درخواست کامپیوتر Y، اطلاعات مربوط به کامپیوتر X را به او ارسال می‌کند و به او می‌گوید که کامپیوتر X تمامی بخش‌های ویدیو A را دارد. با این اطلاعات، کامپیوتر Y می‌تواند درخواست خود را به کامپیوتر X ارسال کند.

۳.۲ درخواست و دریافت بخش‌های فایل از Seeder

در مرحله بعد، کامپیوتر Y به کامپیوتر X متصل می‌شود و از او می‌خواهد که یکی از chunk‌های فایل ویدیو A را برایش ارسال کند. کامپیوتر X یکی از بخش‌های فایل را به کامپیوتر Y ارسال می‌کند.

۴.۲ اطلاع‌رسانی به Tracker پس از دریافت هر Chunk

پس از دریافت یکی از chunk‌ها، کامپیوتر Y به tracker اعلام می‌کند که اکنون این بخش از فایل را در اختیار دارد. این اطلاعات به tracker کمک می‌کند تا اطلاعات دقیقی از وضعیت فایل‌ها و peerها موجود در شبکه داشته باشد. حالا اگر کامپیوتر Z بخواهد همان بخش از فایل را دانلود کند، tracker به او می‌گوید که هم کامپیوتر X و هم کامپیوتر Y آن chunk را دارند. این اطلاعات باعث می‌شود که tracker بتواند به peerهای جدید کمک کند تا به راحتی بخش‌های مختلف فایل را دریافت کنند.

۵.۲ تبدیل Peer به Seeder

زمانی که کامپیوتر Y دانلود تمام chunk‌های فایل را تکمیل کرد، او نیز به یک seeder تبدیل می‌شود. به این معنی که کامپیوتر Y اکنون تمام بخش‌های فایل ویدیو A را در اختیار دارد و می‌تواند آن را با دیگران به اشتراک بگذارد. به همین ترتیب، هر peer جدید که به شبکه متصل می‌شود و فایل را دانلود می‌کند، به یک seeder تبدیل می‌شود و فرآیند اشتراک‌گذاری فایل به صورت غیرمتمرکز ادامه پیدا می‌کند.

۳ صورت پروژه

هدف از این پروژه پیاده‌سازی یک شبکه تورنت ساده است که در آن حداقل سه peer و یک tracker وجود دارد که به شرح زیر است:

۱.۳ تعداد مورد نیاز

حداقل سه peer (برای دانلود و آپلود فایل‌ها) حداقل یک tracker (برای مدیریت و اطلاع‌رسانی وضعیت فایل‌ها در شبکه) تعداد peerها و trackerها قابل افزایش است، اما رعایت حداقل تعداد ذکر شده برای انجام پروژه الزامی است.

۲.۳ عملکرد Tracker

Tracker در این شبکه مسئول نگهداری اطلاعات مربوط به فایل‌های موجود در هر peer است. وظایف tracker عبارتند از: نگهداری اطلاعات وضعیت فایل‌ها: tracker باید بداند که هر فایل را کدام peerها به طور کامل دارند. اطلاع‌رسانی به درخواست‌کننده: وقتی یک کاربر (یا peer) بخواهد فایلی را دانلود کند، tracker اطلاعاتی شامل ساینز فایل و لیست peerهایی که فایل مورد نظر را به طور کامل دارند، به درخواست‌کننده ارسال می‌کند.

۳.۳ فرآیند دانلود فایل

درخواست دانلود: درخواست‌کننده پس از دریافت اطلاعات از tracker، یکی از peerهایی که فایل را به طور کامل دارد، به صورت تصادفی انتخاب می‌کند. ارسال درخواست به peer منتخب: درخواست‌کننده فایل را از peer انتخابی دانلود می‌کند. تبدیل به seeder: پس از دانلود فایل، peer درخواست‌دهنده به seeder تبدیل می‌شود. در این حالت، این peer فایل را به دیگر کاربران در شبکه ارائه می‌دهد.

۴.۳ مدیریت لاگ‌های فعالیت‌ها

Tracker باید فعالیت‌های آپلود و دانلود فایل‌ها را ثبت کرده و اطلاعات مربوط به اینکه چه کسی چه فایلی را دانلود یا آپلود کرده، ذخیره کند. این اطلاعات به صورت لاگ باید نگهداری شود.

۵.۳ تعامل بین اجزای شبکه

tracker: مسئول هماهنگی و ارسال اطلاعات وضعیت فایل‌ها و peerها.
peerها: کاربران شبکه که فایل‌ها را به اشتراک می‌گذارند و دانلود می‌کنند. هر peer پس از دانلود کامل فایل، به seeder تبدیل می‌شود و آماده است تا فایل را به دیگران ارائه دهد.

۶.۳ الزامات پیاده‌سازی

پیاده‌سازی tracker که قادر به ذخیره‌سازی و ارسال اطلاعات مربوط به peerها و فایل‌هایشان باشد.
پیاده‌سازی peerها که قادر به ارسال و دریافت درخواست‌های دانلود و آپلود فایل‌ها باشند.
پیاده‌سازی لاگ‌ها برای ثبت تمام فعالیت‌های آپلود و دانلود همراه با جزئیات.

۴ برنامه‌هایی که باید پیاده شوند:

۱.۴ برنامه Tracker

این برنامه نقش مدیریت و ذخیره‌سازی اطلاعات فایل‌ها و نظارت بر فعالیت‌های کاربران (peerها) را بر عهده دارد. پیش از اجرا، نیاز است که یک ورودی به فرمت IP:PORT ارائه شود؛ برای مثال: 127.0.0.1:6771. برنامه پس از راه‌اندازی روی پورت ۶۷۷۱ UDP آماده دریافت و پاسخ به درخواست‌ها خواهد بود. تمامی ارتباطات به صورت محلی برقرار می‌شود.

۲.۴ برنامه Peer

هر peer باید برای شناسایی منحصر به فرد یک نام یا شناسه (id) داشته باشد. در کاربردهای واقعی BitTorrent، کلاینت‌ها توانایی دانلود و به اشتراک‌گذاری چندین فایل به طور همزمان را دارند. اما برای ساده‌سازی این پروژه، فرض می‌کنیم هر peer فقط در یکی از حالت‌های share یا get فعال است.

۱.۲.۴ حالت Share

در این حالت، کاربران می‌توانند فایل‌های موجود روی کامپیوتر خود را در شبکه‌ی تورنت به اشتراک بگذارند. فایل‌ها بر اساس نامشان در شبکه شناسایی می‌شوند و برای جلوگیری از تکرار، نام‌ها باید یکتا باشند. ساختار دستوری پیشنهادی برای اشتراک‌گذاری فایل به شکل زیر است:
(نام فایل) (آدرس ترک) (آدرس گوش دادن) share

۲.۲.۴ حالت Get

در این حالت، هدف ما دریافت یک فایل از شبکه‌ی تورنت است. ابتدا، peer باید به tracker متصل شود تا اطلاعات مربوط به فایل مورد نظر را دریافت کند. tracker لیستی از peerهایی که فایل را دارند، دارد. سپس، peer یکی از این منابع را انتخاب کرده و درخواست دریافت فایل را به آن ارسال می‌کند.
ساختار آرگومان‌ها در این حالت مشابه حالت share است، با این تفاوت که به جای share از get استفاده می‌شود.
برخلاف دانلودهای عادی، پس از پایان دانلود، برنامه به صورت خودکار به یک seeder تبدیل می‌شود و تمام نمی‌شود.

۳.۴ لاگ سیستم

با هر درخواستی که از یک peer برای tracker ارسال می‌شود، نام آن peer، درخواست آن، peerهایی که آن فایل را در اختیار دارند و در نهایت موفق بودن یا نبودن در گرفتن فایل در tracker ثبت می‌شود که با زدن دستور `logs request` در خط فرمان tracker نشان داده می‌شوند. همچنین هر فایلی که در شبکه منتشر می‌شود یک لاگ برای tracker ثبت می‌کند که در صورت زدن دستور `all-logs` در خط فرمان tracker، کلیه این لاگ‌ها نمایش داده می‌شود (که هر قسمت از کدام فایل در دست کدام peerها است) و در صورت زدن دستور `file_logs<file_name>` در خط فرمان tracker، لاگ‌های مربوط به یک فایل نمایش داده می‌شوند که در صورت عدم وجود فایل، باید یک پیام خطای مناسب نمایش داده شود. در برنامه peer نیز باید لاگ تمامی پاسخ‌های آمده از طرف سرور جهت گرفتن یک قسمت از فایل ثبت شود که با دستور `request logs` در خط فرمان peer نمایش داده می‌شود. هنگام وصل شدن هر peer به tracker در برنامه peer، یک لاگ اتصال حاوی نام یا آیدی peer نمایش داده می‌شود و نیز یک لاگ قطع شدن peer از tracker باید لیست فایل‌های مربوط به آن peer در برنامه tracker آپدیت شود و نیز یک لاگ قطع شدن حاوی نام یا آیدی peer در برنامه tracker نمایش داده می‌شود.

۵ نکات پیاده سازی

۱. بین tracker و هر peer از یک سازوکار استفاده کنید تا به tracker کمک کند که متوجه شود کدام peerها هنوز در شبکه هستند و کدامشان دیگر در شبکه نیستند. فرض کنید یکی از peerها که شامل فایلی هست، از شبکه خارج شود، در این صورت اگر peer دیگری همان فایل را از tracker درخواست کند، باتوجه به اینکه peer مربوطه از شبکه خارج شده است، نمی‌تواند به این درخواست پاسخ دهد و peer درخواست دهنده معطل می‌ماند! نحوه پیاده سازی این سازوکار بر عهده خودتان می‌باشد، فقط دقت کنید که بعد از قطع شدن ارتباط یک peer با شبکه، باید اسم آن را از لیست دارندگان فایل (در صورت داشتن فایل مورد درخواست) از شبکه حذف کنید تا دیگر کسی به آن درخواست ندهد.
۲. ارتباط بین tracker و peer باید به صورت UDP و ارتباط بین peerها باید به صورت TCP باشد.
۳. پروتکل ارتباطی بین تمام برنامه‌ها بر عهده خودتان است.
۴. برخلاف شبکه‌ی واقعی تورنت، لازم نیست فایل‌ها را `chunk chunk` کنید.
۵. دقت کنید فایل‌های درخواستی نباید توسط tracker دانلود یا آپلود شود، بلکه وظیفه tracker صرفاً فرستادن `ip` و `port` دارندگان فایل خواسته شده می‌باشد.
۶. تمام برنامه‌ها باید به صورت `multithreaded` پیاده سازی شوند. برای مثال tracker می‌تواند همزمان به چندین درخواست پاسخ دهد و همچنین هر peer می‌تواند چندین فایل را آپلود یا دانلود کند. (برای سادگی فرض کنید که آپلود و دانلود در کنارهم به طور همزمان صورت نمی‌گیرد).
۷. دقت کنید که برنامه tracker و برنامه peer باید مستقل از هم پیاده سازی شده باشند و هرکدام را جداگانه اجرا کنیم.
۸. شما فقط مجاز به استفاده از زبان‌های برنامه نویسی جاوا یا پایتون هستید.