به نام خدا

درس برنامه‌سازی پیشرفته

معماری P2P

دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

نیم سال دوم ۹۹ـ۹۸

اساتید:

مهدی مصطفی‌زاده، ایمان عیسی‌زاده، امیر ملک‌زاده، علی چکاه

نگارش و تهیه محتوا:

صابر ظفرپور

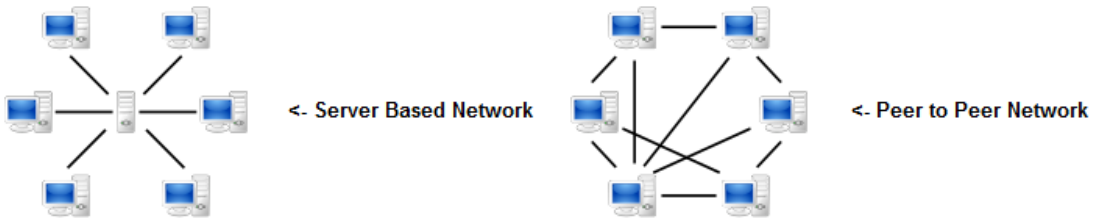
تنظیم داک:

امیرمهدی نامجو و صابر ظفرپور

فهرست

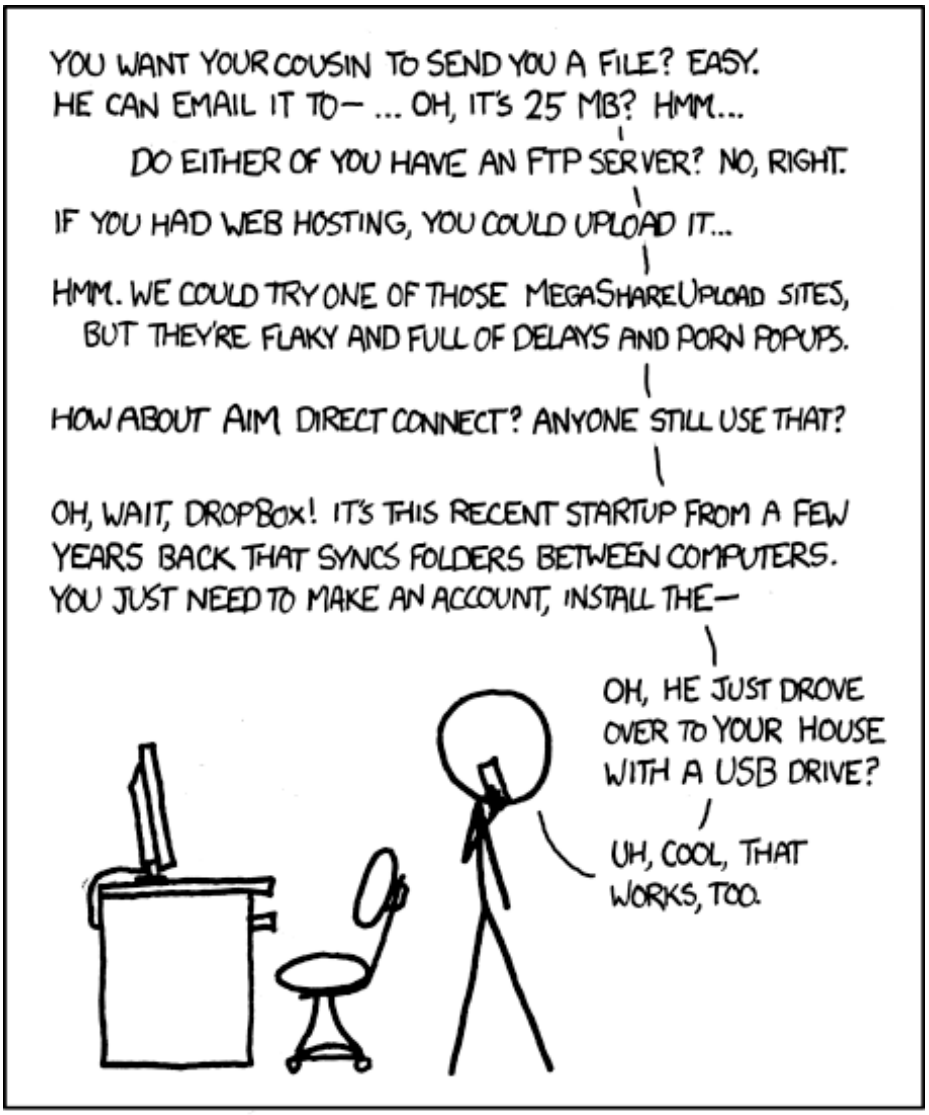
اینو دیگه کجای دلمون باس جا بدیم؟

یکی بود ولی هنوز اینترنت نبود، اواخر دهه هفتم قرن ۲۰ام میلادی پروژه‌ای مشترک بین تعدادی از دانشگاه‌های آمریکا و به واسطه پشتیبانی DoD (وزارت دفاع آمریکا) آغاز به کار کرد که ۴ راس در نقاط مختلف آمریکا را به یکدیگر وصل کرد. بعدها این شبکه گسترش یافت و با نام اینترنت به صورت تجاری و برای عموم مردم قابل استفاده شد. در اوایل این دوران معماری که مورد استفاده قرار می‌گرفت چیزی نبود جز یک معماری P2P ، اما با گذر زمان به علل مختلف (امنیتی ، تجاری ، فنی و ...) معماری مورد استفاده توسط اینترنت طی یک دگرگونی تبدیل به چیزی شد که ما الان با آن آشنایی داریم .



معماری بر مبنای مدل کلاینت سرور مدلی از معماری شبکه است که در حال حاضر به صورت کاملا وسیعی مورد استفاده قرار گرفته است. در این مدل تعدادی سرور وجود دارد که بقیه کاربران فقط حق ارتباط مستقیم با همین سرورها را دارند ، به طور مثال فرض کنید شما (Alice) در این مدل قصد دارید برای دوست خود(Bob) یک فایل موزیک را ارسال کنید، احتمالا تمام راه‌های انتقال فایلی که در ذهن دارید از طریق همین نوع معماری است (تقریبا تمامی نرم افزارهای فضای ابری و یا پیام‌رسان‌ها) ، اما سوالی که در ذهن ایجاد می‌شود اینست که آیا واقعا این مدل معماری شبکه در تمامی حالات منطقی است؟؟؟ چرا در چنین شبکه‌ای (اینترنت) که تمامی راس‌های عضو آن در یک گراف همبند قرار دارند، این راس‌ها نباید قادر به ارتباط مستقیم و بدون حضور شخص سوم با یکدیگر باشند؟(اگر این

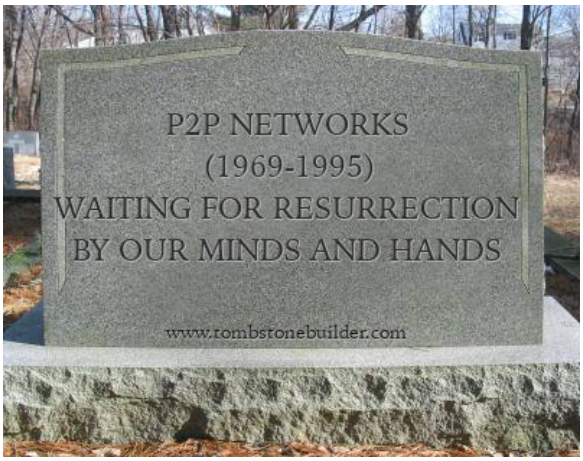
مسئله باعث دغدغه ذهنی شما نشده، خواندن ادامه داک احتمالا برای شما بی ثمر خواهد بود)



I LIKE HOW WE'VE HAD THE INTERNET FOR DECADES, YET "SENDING FILES" IS SOMETHING EARLY ADOPTERS ARE STILL FIGURING OUT HOW TO DO.

بازگشت به ریشه‌ها

در طی تمامی این سال‌ها ، توسعه دهندگان متن باز (open-source developers) بر روی ایده‌ها و پیاده‌سازی‌های خلاقانه p۲p کار کرده‌اند. احتمالا نام بعضی از این ایده ها و ابزارها برایتان آشنا باشد (بیت کوین (واحد پول بدون بانک مرکزی)، بیت تورنت (اشتراک گذاری فایل در اینترنت)، Usenet ، Darknet Napster، Skype، Gnutella).



اگر هم این سنگ قبر برایتان سوالی ایجاد کرده و به دنبال قاتل این موجود شریف و اصیل هستید، شما را به این موجود (NAT) ارجاع میدهم که باعث بسیار سخت شدن پیاده سازی این معماری در سطح اینترنت شده است. البته تا سال‌های ۲۰۰۵ نیز همچنان بیشتر ترافیک اینترنت توسط اپلیکیشن های با معماری P2P مصرف میشد.

یکمی هم حرف فنی بزن دیگه!!

در این معماری هیچ گونه تفاوتی بین راس‌ها وجود ندارد، یعنی هر راس هم می‌تواند سرور باشد و هم کلاینت ، یعنی می‌تواند همزمان که یک فایل را برای فرد دیگر میفرستد ، فایل مورد نیاز خود را از فرد دیگری در شبکه دریافت کند.

همانطور که احتمالا می‌دانید به هر موجود در شبکه یک شماره منحصر به فرد به نام IP داده می‌شود، اما این IP کافی نیست ، یک عدد دیگر نیز تعیین کننده است، این عدد port است،که از ۰ تا ۶۵۵۳۵ بازه آن است، این عدد تعیین کننده این است که اگر بسته‌ای به IP شما ارسال شد باید برای کدام برنامه سیستم عامل ارسال شود (به طور مثال اگر شما همزمان در حال استفاده از دو مرورگر باشید، باید سیستم عامل بداند که هر بسته مربوط به کدام مرورگر است)، پس اگر شما بر روی یک پورت رندوم در حال شنیدن باشید و IP و آن پورت را به فرد دیگری بدهید، شما و آن فرد میتوانید به صورت مستقیم به هم وصل شوید و در صورت استفاده از پروتکل (مانند زبان مورد استفاده انسان‌ها) یکسان می‌توانید با هم

ارتباط برقرار کنید. البته ذکر این نکته ضروری است که این مسئله به علت وجود NAT نزدیک به ۲۵ سال است که تبدیل به یک چالش شده است، البته تعدادی روش برای دور زدن این موجود (NAT Traversal) به وجود آمده است که به وسیله آن هنوز می‌توانیم به ادامه این معماری امید داشته باشیم. تقسیم می‌شود.

خب این Peer-to-Peer اصلا چه مزایایی داره؟؟

هزینه راه اندازی سرورها برای اکثریت سرویس‌ها یک بار اضافه و از نظر مالی بسیار سنگین است ، به طور مثال فرض کنید شما یک پیام‌رسان بومی تولید کرده‌اید و این پیام‌رسان قابلیت تماس تصویری دارد و همچنین فرض کنید در حال حاضر که همه در خانه‌های خود هستند (در زمان نگارش این داک کرونا شکست نخورده بود:) بیش از ۵۰۰۰۰ کاربر از قابلیت تماس تصویری شما استفاده کنند و با یکدیگر تماس برقرار کنند، اگر فرض کنیم ثانیه‌ای ۳۰۰kb توسط هر راس اطلاعات به سرور برای فرستادن به راس دیگر فرستاده شود، آنگاه شما باید قادر به تامین گذردهی حداقل ۱۵Tb در ثانیه برای سرور خود باشید که حتی بدون حساب کردن قدرت پردازشی تلف شده یک هزینه کاملا بی دلیل است، در صورت استفاده از یک معماری P2P آنگاه دیگر نیاز به چنین سرور با چنین منابع بالایی نبود. نکته دیگر موضوع امنیت است که در این نوع معماری با توجه اینکه می‌توان رمز گذاری‌های متنوعی استفاده کرد، دغدغه امنیت ارتباط کاربران برطرف می‌شود. هرگز مزیت privacy موجود برای هر راس در این نوع شبکه را از یاد نبرید، این مزیت در کنار غیر قابل سانسور بودن، دو مزیت بسیار بسیار مهم این نوع معماری هستند.

چطور یک Peer-to-Peer دولوپر شم؟

کاملا واضح است که با توجه به شرایط فعلی (وجود NAT در اکثر شبکه‌ها)، شما نمی‌توانید همانند معماری سرور و کلاینت با چند خط کد ساده یک ساختار شبکه P2P پیاده کنید. اما با مطالعه کدهای موجود در مخازن گیت هاب به ایده‌های بسیار جذابی بر می‌خورید، پس یکی از منابع اصلی یادگیری P2P کدهای موجود در گیت هاب است (میتوانید با جستجوی tag هایی نظیر P2P به این مخازن دسترسی داشته باشید)

گوگل چندین سال است که در حال توسعه تکنولوژی به نام WebRTC است که در حال حاضر توسط اکثر دولوپرهای این حوزه درحال استفاده است. WebTorrent نیز بر روی ترکیب این تکنولوژی و تورنت ساخته شده است. در حال حاضر بهترین زبان برای پیاده سازی و استفاده از معماری P2P زبان javascript (که هیچ ربطی به جاوا نداره) است. البته توصیه می‌شود قبل از تلاش برای شروع کد زدن و پیاده سازی این معماری در ابتدا با مطالعه مفاهیم NAT Traversal و پروتکل‌های رایج مورد استفاده نظیر STUN، TURN،

ICE آشنایی حداقلی را با این موضوع به دست بیاورید.

پس پروژه چی شد این وسط؟؟

در فاز سه، قسمتی امتیازی وجود دارد که شما باید برای قسمت فروش فایل، از این معماری به صورت محدود (فقط برای انتقال فایل) استفاده کنید ، به این صورت که خریدار فایل خود را از سرور دریافت نکند بلکه آن را به صورت مستقیم از فروشنده دریافت کند.

۱. برای بررسی کردن و نمره دادن به این بخش، به پیاده سازی پروتکل و برقراری ارتباط نمراتی تعلق می گیرد (پس با کمی وقت گذاشتن می‌تونید از این قسمت امتیازی نمره مناسبی دریافت کنید).

۲. همچنین ما فقط ساده‌ترین حالت معماری P2P، یعنی حالتی که هر دو راس در یک شبکه محلی قرار دارند (مثلا به یک مودم وصل‌اند و به اصطلاح پشت یک NAT هستند)را بررسی می‌کنیم ، پس عملا چالش‌های شما بسیار محدود و کم است.

نمونه های واقعی و منابع

۱. P2P vs client-server

۲. P2P history

۳. WebRTC samples

۴. My P2P network for Sharif(SHoReNT) which is under development