

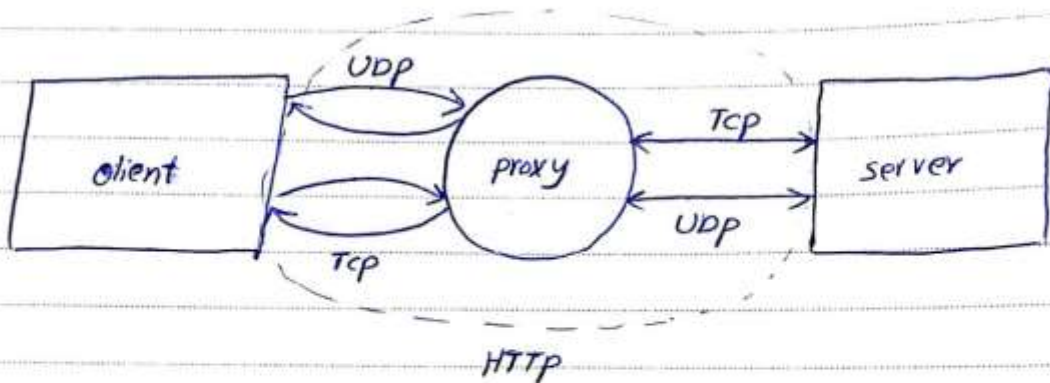
شرح پروژه پایانی شبکه

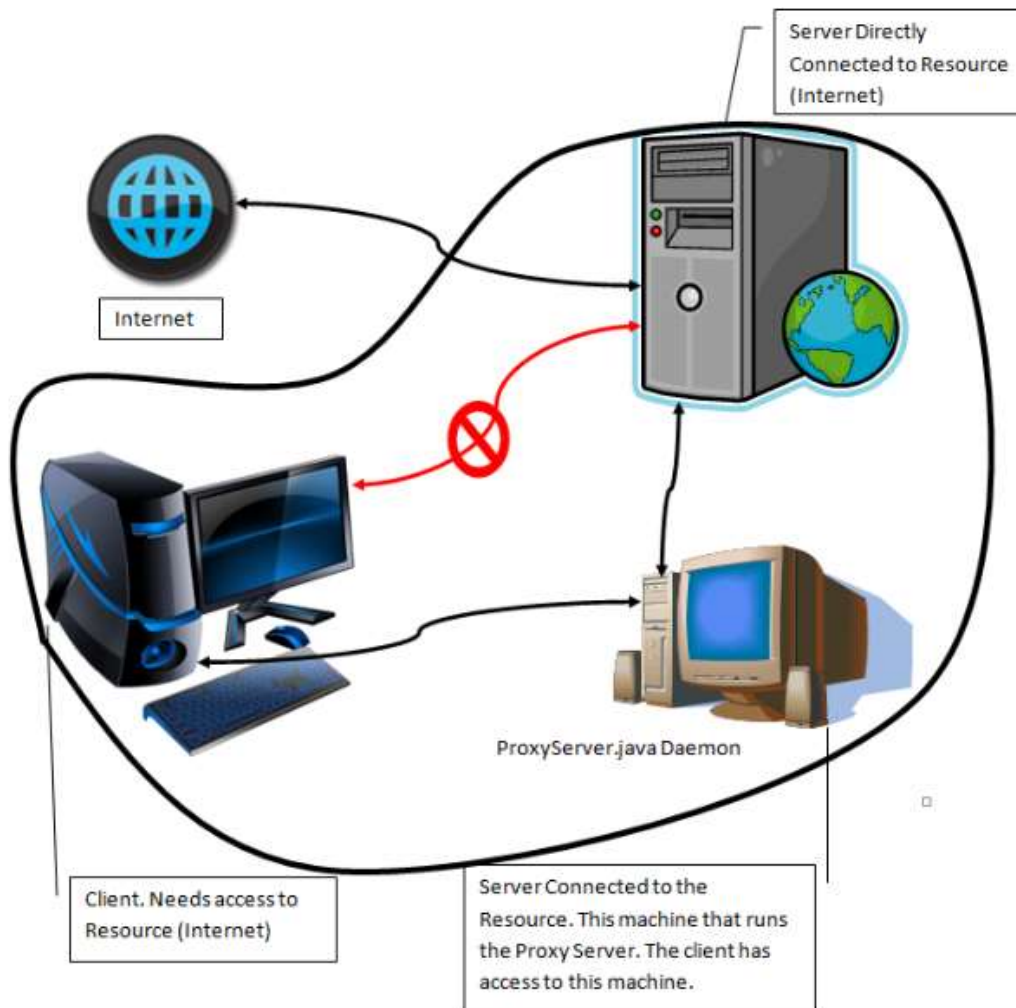
مقدمه

پروتکل UDP فاقد مکانیسمهای کنترل ازدحام است و میتواند باعث شود تا ارسال کنندگان داده با ارسال زیاد بسته های UDP باعث کاهش کارایی سیستمهایی شوند که از پروتکل TCP به عنوان پروتکل لایه انتقال خود استفاده میکنند. به همین علت ممکن است در برخی از شبکه ها استفاده از پروتکل UDP محدود شود. به این منظور کاربران این شبکه ها از پروتکلهای لایه حمل استفاده میکنند که بستههای UDP را به بسته های TCP تبدیل کرده و پس از دریافت جواب این تبدیل معکوس را مجددا انجام میدهد.

شکل دیاگرام:

Subject Network project  
Date \_\_\_\_\_





## پروکسی

برنامه پروکسی باید قادر باشد، پیامهای UDP ارسال شده از طرف سرویس گیرنده را بر روی اتصال TCP به سمت سرویس دهنده که بر روی پورت مشخصی در حال گوش کردن است ارسال کند. جواب درخواستها بر روی اتصال UDP به سرویس گیرنده ارسال میشود. همچنین پروکسی پیامهای TCP ارسال شده از طرف مشتری را باید بتواند به پیامهای UDP تبدیل کند و به سرویس دهنده ارسال کند. جواب پیامها بر روی اتصال TCP به سرویس گیرنده ارسال میشود.

پروژه از سه بخش اصلی تشکیل میشود:

۱. سرویس دهنده (server())

۲. پروکسی

۳. سرویس گیرنده (client())

برای بخش سرور ها از سرورهای خود اینترنت استفاده میشود؛ فقط لازم است ما در سمت کلاینت و پروکسی، پیاده سازی را انجام دهیم.

برنامه ی پروکسی درخواست های کلاینت را به فرمت مناسب تبدیل کرده و برای سرور ارسال میکند.

### سرویس گیرنده (client)

برای اینکه برنامه ی نوشته شده به راحتی بتواند با این پروکسی ها کار کند، پیاده سازی ابزار مخصوصی در سمت کلاینت نیاز است.

این ابزار از دو کاربرد اینترنت پشتیبانی میکند:

### ۲ کاربرد HTTP بر روی پروتکل UDP

مشابه یک مرورگر اینترنت، ولی با قابلیت های بسیار کمتر عمل میکند. برنامه باید به عنوان ورودی یک درخواست HTTP استاندارد را از کاربر دریافت کند و آن را در قالب بسته UDP به پروکسی ارسال کند.

۱. برنامه شما باید قابلیت پشتیبانی از کدهای 301 ، 302 و 404 را داشته باشد و با نمایش دادن یک پیغام کاربر را از نتیجه درخواست آگاه کند. (در صورتی که جواب کد 301 یا 302 را داشت، درخواست برای Location جدید باید به صورت خودکار در برنامه شما آنقدر تکرار شود تا به کد 200 برسد.)

### ۲ کاربرد DNS بر روی پروتکل TCP

برنامه کلاینت باید مشابه یک سرویس گیرنده DNS عمل کند و بتواند درخواستهای DNS را ارسال و جواب آنها را پردازش کند.

پیغام ایجاد شده سپس از طریق پروتکل TCP به پروکسی ارسال میشود. پروکسی، محتوای پیغام را بازیابی کرده و در قالب یک پیغام DNS به آدرسی که در پیغام مشخص شده است ارسال میکند.

۱. برنامه شما باید قادر باشد بسته دریافت شده را تحلیل کند و برای نوع A، آدرس همه IP های دریافتی را نمایش دهد:

۲. برنامه شما باید قادر باشد مشخص کند که آیا جواب Authoritative است یا خیر.

۳. برنامه باید بتواند در صورت عدم دریافت جواب پرسوجوی کاربر در زمان مشخص، آن را مجددا ارسال کند و تا دریافت کردن جواب مناسب این کار را ادامه دهد.

شرح ماژول ها و منطق پروژه :

پروژه شامل سه ماژول پراکسی، سرویس گیرنده ی HTTP و سرویس گیرنده ی DNS است.

برنامه سپس به پراکسی متصل میشود و اطلاعات را دریافت میکند و در خروجی نمایش می دهد.  
پراکسی در جواب، اگر جواب HTTP، مقادیر ۳۰۱ و ۳۰۲ باشد، انقدر این روند تکرار میشود که به ۲۰۰ و ۴۰۴ میرسد.

۴۰۴ → خطایی بافت نشده است.

۲۰۰ → خروجی به برنامه بازگشت داده میشود.

برنامه ی Client tool DNS، از طریق پروتکل UDP به پروکسی پیغام ارسال میکند.  
پروکسی این خط را بررسی میکند و یک DNS Query و نتایج آن را در قالب یک فایل UDP به برنامه بر میگردداند که در نهایت به عنوان خروجی در نظر گرفته میشود.

در ابتدا یک سوکت UDP ایجاد میشود. بافر ۵۱۲ بایت است.  
در آخر یک شی برای دریافت بسته ها ایجاد می شود.

اگر درخواست از سمت کلاینت HTTP باشد، آدرس از سمت آن دریافت می شود.  
بعد از دریافت آدرس، پراکسی پیام می دهد که باید منتظر ارسال html: باشد. اگر کد ۴۰۴ دریافت شد، باید یعنی جوابی داده نشده است.

در غیر این صورت پیغام html ارسال میشود.

اگر درخواست از سمت dns باشد، ابتدا مشخصات آن دریافت میشود.  
ابتدا بررسی میکنیم که جواب ۳۰۰ یا ۳۰۱ است یا نه (که باید آنرا REDIRECT کنیم)  
و دوباره روی مقدار جدید ارسال میکنیم تا به مقدار ۲۰۰ یا ۴۰۴ برسیم.  
سپس html آنرا بر میگردداند.

Yasaman Mirmohammad 9431022  
Melika Abdollahi 9431804