پروژه پایانی

شبکه‌های کامپیوتری - بهار ۱۳۹۹

NetWolf

دانشکده مهندسی کامپیوتر

فهرست مطالب

مقدمه

در این تمرین قصد داریم شما با پیاده‌سازی یک نرم افزار توزیع فایل به صورت P2P آشنا کنیم. در این نرم‌افزار کاربران یک خوشه تشکیل می‌دهند که در این خوشه می‌توانند برای دریافت فایل تقاضا ارسال کنند. این نرم‌افزار

مدیریت خوشه را به صورت خودکار و به وسیله‌ی ارسال دوره‌ای پیام Discovery صورت می‌دهد. در ادامه به توضیح این نرم‌افزار می‌پردازیم.

قوانین

۱. تمرین به صورت **انفرادی** انجام می‌شود.

۲. در این پروژه فقط مجاز هستید از کتابخانه‌های socket برای ارتباط شبکه استفاده کنید. به این معنا که استفاده از هر گونه ارتباط سطح بالاتر و پروتکلی به غیر پروتکل‌های لایه‌ی Transport فاقد ارزش می‌باشد.

۳. استفاده از هرگونه کتابخانه‌ی آماده برای Serialization و Deserialization در هیچ یک از گام‌ها مجاز نمی‌باشد. تمامی اطلاعات می‌بایست به صورت مشخص توسط شما رمز و ارسال شوند و در سمت دیگر نیز توسط شما رمزگشایی و دریافت شوند.

پیاده‌سازی

هر کاربر برای اجرای برنامه نیاز دارد یک لیست را به عنوان اعضا خوشه در اختیار برنامه قرار دهد. این لیست می‌تواند خالی باشد.

```
N1 192.168.73.1
N2 192.168.73.2
```

بعد از شروع، برنامه در بازه‌های مشخص پیام Discovery که شامل لیست اعضا خوشه‌اش می‌باشد را برای تمام اعضا خوشه‌اش ارسال می‌کند. در نظر داشته باشید که هر نود به صورت مجزا لیستی را به عنوان اعضا خوشه‌اش مدیریت می‌کند که این لیست در طی پیام‌های Discovery در بازه‌های مشخص به روزرسانی می‌شود.

کاربر در هر لحظه می‌تواند برای یک فایل تقاضا دهد. این تقاضا در قالب یک پیام Get برای تمامی نودهای خوشه‌ی نودی که کاربر تقاضا را داده است (این نود را مبدا می‌نامیم)، ارسال می‌گردد. نودها پیام درخواست را پردازش می‌کنند و صورتی که فایل موردنظر را در اختیار داشته باشند پاسخ می‌دهند. نود مبدا این پاسخ‌ها را دریافت کرده و بهترین پاسخ از نظر زمان دریافت را انتخاب می‌کند. در صورتی که نود مبدا در یک بازه‌ی زمانی مشخص پاسخی دریافت نکند فرض می‌کند فایل مورد تقاضا در خوشه وجود ندارد. نود مبدا با استفاده از یک ارتباط TCP فایل را از نودی که انتخاب کرده است، دریافت می‌کند.

در ادامه جزئیات این روند را به صورت مرحله به مرحله شرح می‌دهیم.

اکتشاف

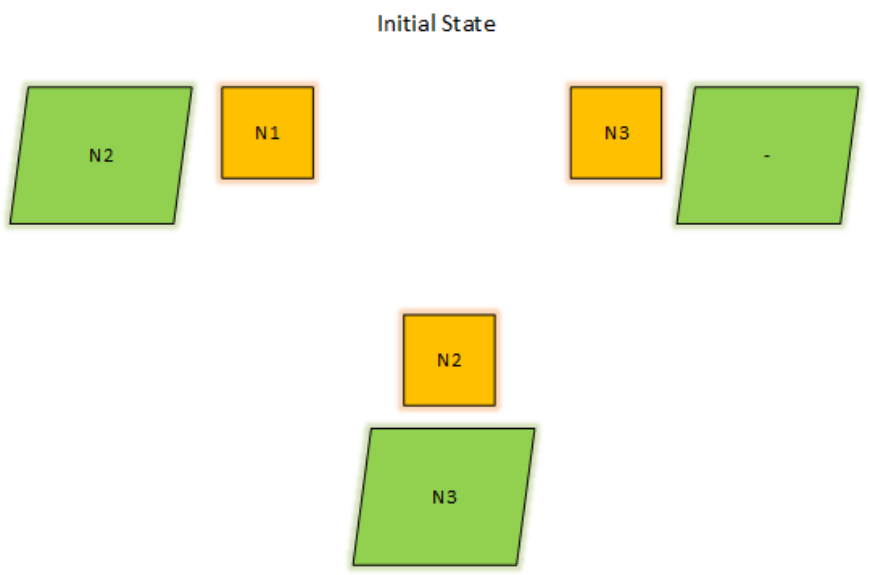
برنامه در ابتدا یک لیست در قالب فایل به عنوان خوشه‌ی آغازین دارد. این لیست می‌تواند خالی باشد. این فایل شامل دو ستون می‌باشد که با فاصله از یکدیگر جدا شده‌اند. هر سطر از این فایل نمایده یک نود در خوشه می‌باشد که شامل اسم و آدرس آی‌پی این نود می‌باشد. در ادامه نمونه‌ای از این فایل آورده شده است:

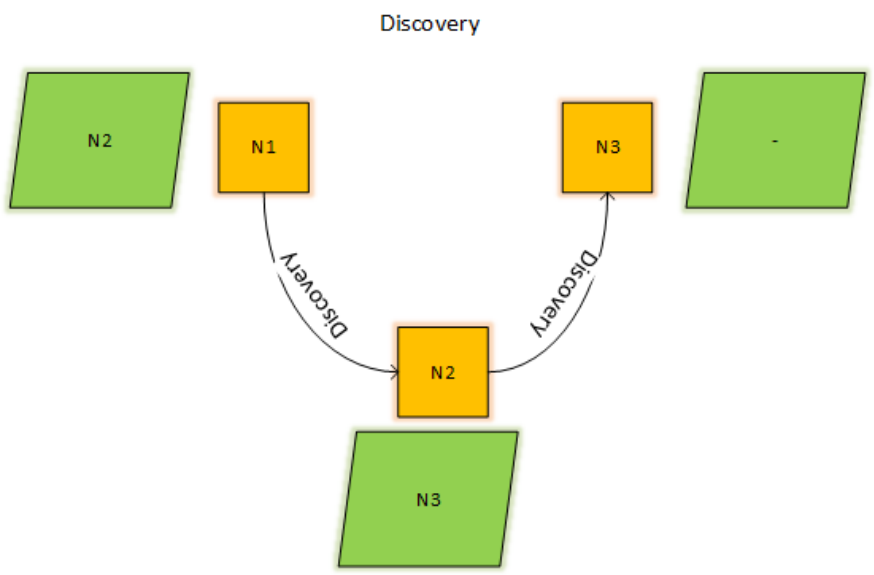
```
N1 192.168.73.1

N2 192.168.73.2
```

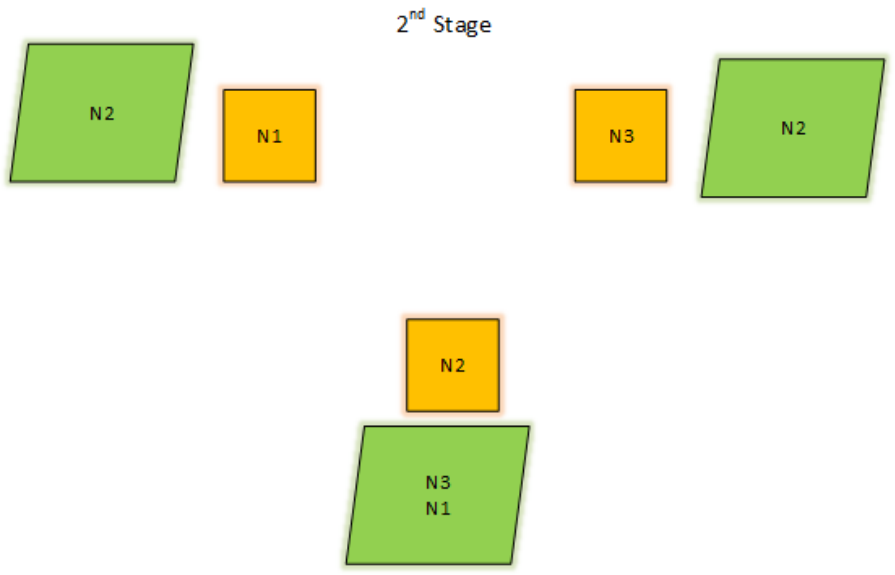
در هر بازه‌ی مشخص هر نود لیست فعلی خوشه‌ی خود را به تمام اعضای خوشه‌ی خودش به اشتراک می‌گذارد. ساختار پیام Discovery بر عهده‌ی خودتان می‌باشد ولی می‌بایست شامل آدرس آی‌پی‌ها و شناسه‌ی نودها باشد. هر نود با دریافت پیام Discovery لیستی که دریافت کرده است را با لیست فعلی خود Merge می‌کند.

در نظر داشته باشید اکتشاف مستقل از سایر قسمت‌های برنامه در حال اجرا بوده و لیست خوشه‌ی نود را به روزرسانی می‌کند. به این ترتیب هر نود به صورت مستقل برداشتی از خوشه داشته و آن را به صورت همزمان به روزرسانی می‌کند.

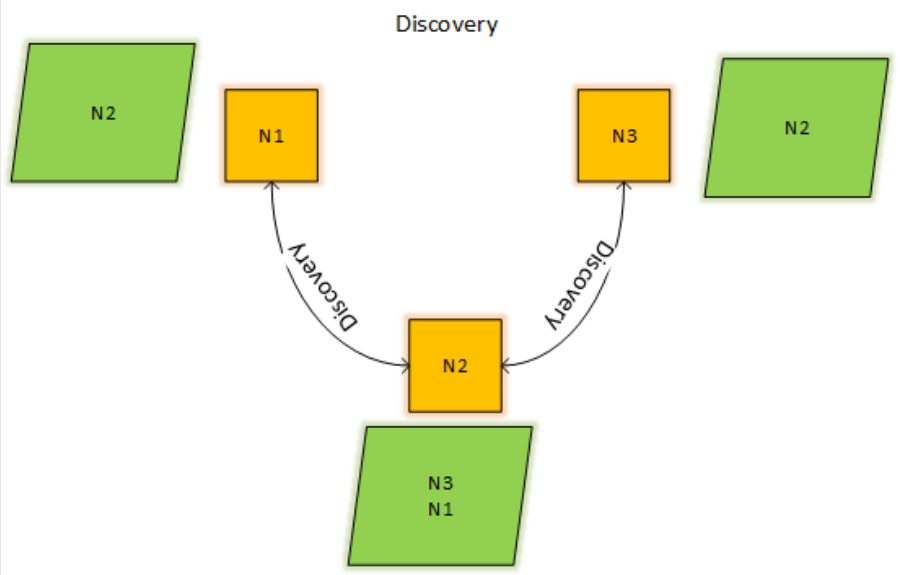
از آنجایی که تعداد این پیام‌ها زیاد می‌باشد برای ارسال آن‌ها از پروتکل UDP استفاده می‌شود. در این پیام‌ها نیازی به قابلیت اطمینان نمی‌باشد چرا که این پیام‌ها به صورت دوره‌ای تکرار می‌شوند و اگر در یک دوره هم از دست بروند مشکلی به وجود نیامده و سیستم می‌تواند این پیام را در دوره‌ی بعدی دریافت کند.

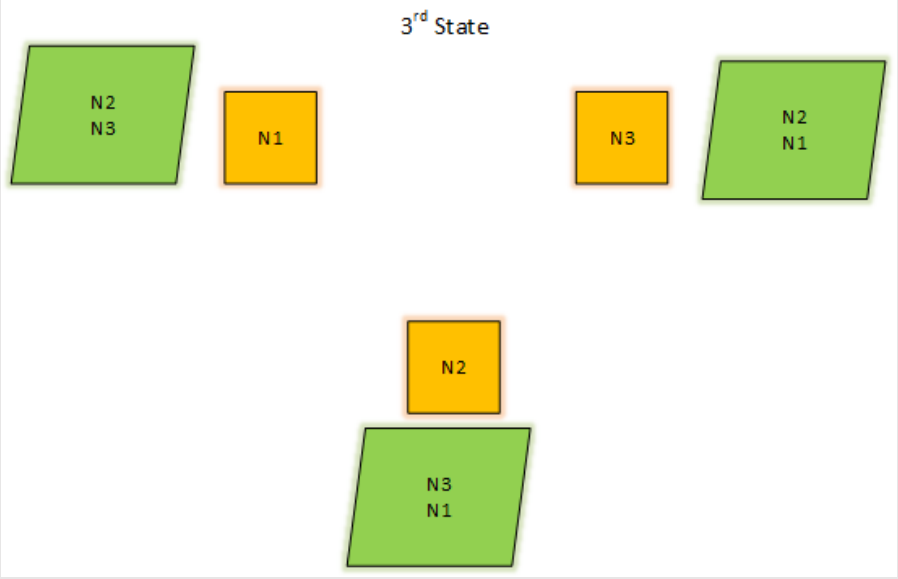



Discovery
N2
N1
N3
-
Discovery
Discovery
N2
N3

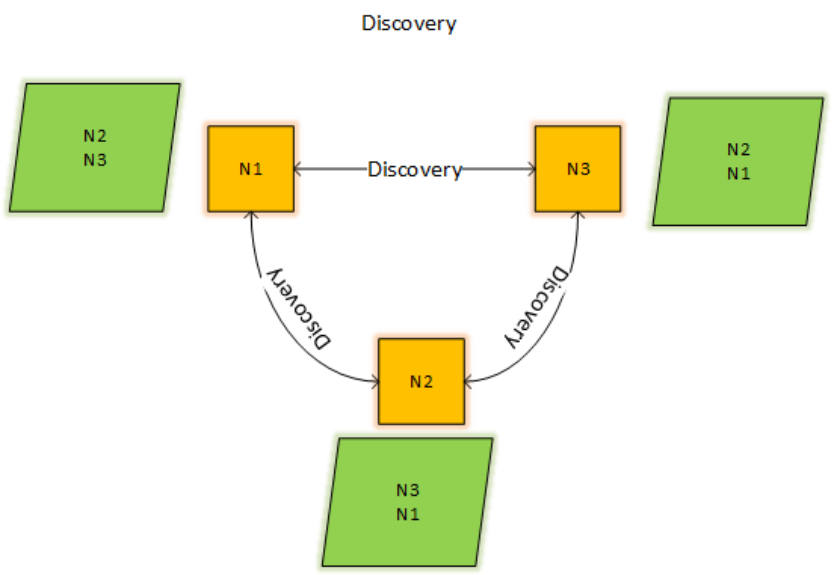

2nd Stage
N2
N1
N3
N2
N2
N3
N1

[4]


Discovery
N2
N1
N3
N2
Discovery
Discovery
N2
N3
N1


3rd State
N2
N3
N1
N3
N2
N1
N2
N3
N1

[5]



تقاضا

کاربر می‌تواند از طریق هر نود تقاضای فایل را در خوشه ارسال کند. این تقاضا را Get می‌نامیم. در نظر داشته باشید که تقاضای Get ارتباطی با تقاضای Get در پروتکل HTTP ندارد. این تقاضای Get بر روی پروتکل UDP برای تمامی اعضای خوشه‌ی نود ارسال می‌گردد. این تقاضا شامل نام فایل می‌باشد و هر نودی که این فایل را داشته باشد بر روی پروتکل UDP به این تقاضا پاسخ می‌دهد. در نظر داشته باشید طراحی تقاضای Get و پاسخ آن برعهده‌ی خودتان می‌باشد.

در پاسخ نودی که فایل را دارد می‌بایست آدرس پورتی که بر روی آن گوش می‌دهد را نیز ارسال کند. در قسمت بعد در رابطه با این پورت و کاربر آن بیشتر صحبت می‌شود.

نود مبدا یک بازه‌ی زمانی مشخص را برای دریافت پاسخ‌ها منتظر می‌ماند. از بین پاسخ‌های دریافتی پاسخی که تاخیر کمتری دارد به عنوان پاسخ اصلی انتخاب می‌شود. در صورتی که نود مبدا در این بازه پاسخی دریافت نکند فرض می‌کند فایل در خوشه وجود ندارد.

در نظر داشته باشید که پروتکلی که در این قسمت طراحی می‌کنید می‌بایست روش برای تشخیص پاسخ با کمترین تاخیر را داشته باشد. در اینجا تاخیر از زمان ارسال پاسخ تا رسیدن آن نود مبدا می‌باشد.

فایل‌ها

هر نود در ابتدا یک فولدر را از کاربر گرفته و فرض می‌شود که می‌تواند فایل‌های داخل این فولدر را سرویس دهد. به این معنی که تقاضای Get برای هر فایل داخل این فولدر می‌تواند توسط این نود پاسخ داده شود. برای جستجو فایل تنها نام فایل کفایت می‌کند.

در قسمت‌های قبل همانطور که بیان کردیم از پروتکل UDP استفاده کردیم اما در این قسمت برای انتقال فایل نیاز است که از پروتکل TCP استفاده کنیم. هر نود در زمان اجرا به صورت تصادفی روی یک پورت TCP آزاد در سیستم گوش فرا می‌دهد. بنابراین نودها از قبل پورت‌های TCP یکدیگر را نمی‌دانند. بنابراین همانطور که بیان شد سرور می‌بایست در زمان پاسخ به تقاضای Get پورت TCP خود را نیز ارسال کند.

برای دریافت فایل نیاز به پروتکل خاصی نیست کافی است نام فایل را برای نود پاسخ‌دهنده ارسال کرده و فایل را دریافت کنید. این فایل در همان فولدری که برای نود مشخص شده است ذخیره می‌شود.

روند اجرا برنامه

```
netwolf -l cluster-list.txt -d directory/
listening on 0.0.0.0:1378 TCP
listening on 0.0.0.0:1373 UDP
> get hello.txt
Getting hello.txt from node N1
……
> list
N1 192.168.73.1
> list
N1 192.168.73.1
N2 192.168.73.2
```

سواری مجانی!

در این نرم‌افزار نودهای یک خوشه می‌توانند به صورت نابرابر ایفای نقش کنند مثلا یک نود ممکن است بدون پاسخ دادن به هیچ تقاضایی فایل‌های بسیاری را دریافت کند. روش‌های زیادی برای جلوگیری از این نابرابری وجود دارد.

در اینجا از یک روش ساده استفاده می‌کنیم. می‌دانیم که نودها برای تقاضا بر اساس تاخیر رفتار می‌کنند پس هر نود در هنگام پاسخ در صورتی که قبلتر از نود متقاضی داده‌ای دریافت نکرده باشد پاسخ را با یک تاخیر دست‌ساز ارسال می‌کند. به طور مثال:

```
if prior_communications[source] is None:
  sleep(10)
send_response()
```

توزیع بار

نودها از پهنای باند مشخصی استفاده می‌کنند، بنابراین در صورتی که تقاضاهای زیادی را سرویس بدهند پهنای‌باند کمی به نودهای متقاضی می‌رسد. برای کنترل این موضوع از روند زیر استفاده می‌کنیم:

در صورتی که نود در حال سرویس دادن به یک تعداد مشخصی از تقاضاها باشد تقاضای جدیدی را در این بازه پاسخ نخواهد داد.

در نظر داشته باشید که روند پیشنهادی یک روند ساده است که لزوما کارآیی بالایی ندارد. در قسمت امتیازی از شما خواسته شده است در صورت امکان این روند را بهبود بخشید.

نکات پیاده‌سازی

۱. در نظر داشته باشید که هر نود به صورت همزمان می‌بایست کلاینت TCP، سرور TCP، سرور UDP و کلاینت UDP باشد. تمامی ارتباطات UDP از یک پورت مشخص استفاده می‌کنند و پورت TCP همانطور که بیان شد می‌بایست به صورت تصادفی انتخاب شود.

۲. تمامی قسمت‌هایی که با کلمه مشخص آورده شده‌اند می‌بایست در برنامه شما قابل تنظیم باشند. این تنظیمات می‌تواند قابل پرچم‌هایی در هنگام اجرا برنامه باشد.

۳. در نظر داشته باشید که به روزرسانی خوشه‌ی نود، سرویس دادن فایل‌ها و سرویس دادن به کاربر سه سرویس متفاوت است که هر نود به تنهایی هر سه‌ی آن‌ها را اجرا می‌کند. این سرویس‌ها می‌توانند در قالب Threadهای مختلف پیاده‌سازی شوند.

۴. طراحی‌هایی که در قالب پروتکل خودتان ارائه می‌دهید می‌بایست تمامی ویژگی‌های گفته شده را داشته باشد. بنابراین برای طراحی آن‌ها وقت گذاشته و صورت پروژه را با دقت مطالعه کنید.

۵. پروژه به قسمت‌های مشخصی شکسته و آن را به صورت در مراحل مختلف پیاده‌سازی کنید.

امتیازی

- پیاده‌سازی یک پروتکل قابل اطمینان بر اساس UDP برای قسمت انتقال فایل و استفاده از آن به جای پروتکل TCP. در نظر داشته باشید که برای فراهم آوردن قابلیت اطمینان می‌بایست از یکی از سه روش S&W، GoBackN یا SR استفاده کنید. بدیهی است که نمره امتیازی هر از این پیاده‌سازی‌ها متفاوت است.
- در این پروژه فرض می‌شود که تمامی اعضا خوشه توسط نود مبدا دیده می‌شوند و تقاضای Get برای آن‌ها به صورت مستقیم ارسال می‌گردد. این فرض برای تقاضای Discovery نیز وجود دارد. پیاده‌سازی را به گونه‌ای تغییر دهید که اگر نود N1 در خوشه‌ی نود N2 قرار دارد تقاضای Discovery و Get به آن نود از طریق نود N1 ارسال شود.
- برای جلوگیری از سواری مجانی راه‌حل‌های بهتری ارائه دهید. (ارائه راه‌حل بدون پیاده‌سازی نمره‌ای ندارد.)
- برای توزیع بار راه‌حل‌های بهتری ارائه دهید. (ارائه راه‌حل بدون پیاده‌سازی نمره‌ای ندارد.)

ابهام‌ها

در صورت هرگونه ابهام در پروژه از طریق ایمیل زیر مورد را پیگیری نمایید.

parham.alvani@gmail.com

قدردانی

ایده‌ی اصلی این پروژه توسط پارسا اسکندرنژاد تهیه شده است که جا دارد اینجا از ایشان قدردانی شود.

آرزوی موفیت روزافزون در تمامی لحظات پیشرو

تیم تدریسیاری

بهار ۱۳۹۹