



بهار ۱۴۰۰

CE-40443

شبکه‌های کامپیوتری: پروژه

مدرس: مهدی جعفری

شما پیش از این با معماری لایه لایه شبکه‌های کامپیوتری و نقش و عملکرد لایه‌های مختلف آشنا شده‌اید. در این پروژه تلاش می‌کنیم مروری بر رویکردهای موجود در شبکه‌های کامپیوتری به صورت گذرا و در یک سیستم فرضی داشته باشیم.

۱ هدف

در این پروژه قصد داریم به پیاده‌سازی یک شبکه Peer to Peer بپردازیم. شبکه مورد نظر یک شبکه‌ی درختی دودویی از گره‌ها^۱ است. مدیر این شبکه وظیفه‌ی مدیریت شبکه را دارد و درختی از گره‌های کاربران^۲ شبکه وجود دارد. گره‌ها به ترتیب اضافه شدن به درخت در سطوح مختلف قرار می‌گیرند. هر گره اختیاراتی برای کنترل ورود و خروج داده به زیردرختش دارد که از طریق دیوار آتش به کار می‌بندد. کاربران طی پروتکلی با درخواست به مدیر، به شبکه متصل می‌شوند و می‌توانند پیام‌هایی را تبادل کنند. در این پروژه شما بایستی پروتکل‌های توصیف شده در لایه‌های مختلف شبکه را پیاده‌سازی کنید.

۲ مقدمه

شبکه P2P یا نظیر به نظیر توزیعی از شبکه‌های کامپیوتری است که در آن هر Peer هم نقش Client و هم نقش Server را بازی می‌کند و از این طریق می‌تواند به تبادل پیام در شبکه‌ای از Peer ها بپردازد. در این نوع شبکه‌ها برخلاف شبکه‌های Client-Server، سرور مرکزی وجود ندارد. Tor، Bitcoin و مثال‌های معروفی از شبکه‌های P2P هستند. نکته‌ای که حائز اهمیت است، این است که در عمل شبکه‌ای که به طور کامل P2P باشد وجود ندارد و حتی در شبکه‌های P2P هم گره‌هایی وجود دارد که سبب پابرجایی شبکه هستند و حذف آن‌ها از شبکه سبب اختلال در شبکه می‌شود.

۳ ساختار

همانطور که گفته شد باید به پیاده‌سازی یک شبکه دودویی درختی بپردازید که توسط اتصال تعدادی Peer ساخته شده است. در ادامه، ابتدا نحوه اتصال یک کاربر جدید به شبکه را بررسی کرده سپس ساختار و ویژگی‌های لایه‌های شبکه که باید توسط شما پیاده‌سازی شوند را توصیف می‌کنیم. در شبکه یک مدیر وجود دارد که صرفاً اضافه شدن کاربران به شبکه را مدیریت می‌کند.

Nodes^۱
Clients^۲

۴ پیاده‌سازی

همه‌ی کاربران بر روی میزبان محلی^۳ اجرا می‌شوند. هر کاربر شبکه پورتی دارد که بسته‌های ارسالی به آن‌ها به آن پورت داده می‌شوند. برای ارسال هر بسته از A به B، A بایستی به پورتی که B روی آن گوش می‌دهد متصل شود (با استفاده از یک پورت دلخواه) و پس از ارسال بسته ارتباط را ببندد.

بنابراین به صورت کلی دو ریشه^۴ برای یک کاربر در حال اجرا شدن است:

- ریشه‌ای که بر روی پورت مخصوص کاربر در حال گوش دادن برای دریافت پیام می‌باشد.
- ریشه‌ای که اجرای معمول کاربر از طریق آن است. خواندن دستورات ورودی ترمینال از طریق این ریشه است.

برای پورت دلخواهی که فرستنده از آن استفاده می‌کند می‌توانید به دلخواه خود استراتژی‌ای را برگزینید. مثلاً اگر کاربر از پورت شماره‌ی i برای دریافت بسته‌های جدید استفاده می‌کند، از پورت $i + 1$ برای ارسال استفاده می‌کند. می‌توانید فرض کنید که پورت‌های نسبت داده شده به گره‌ها رزرو شده نیستند و مقادیر پراکنده‌ای نیز دارند.

۵ اتصال به شبکه

در شبکه، هر گره یک شناسه^۵ مشخص دارد. این شناسه توسط خود کاربر مشخص می‌شود. پورت مدیر یک مقدار عمومی است که همه از آن مطلع هستند (به دلخواه خود آن را انتخاب کنید). مدیر روی پورت مذکور در حال گوش دادن است تا ارتباطی با کاربر جدیدی که می‌خواهد به شبکه اضافه شود برقرار شود. فرض کنید یک کاربر بخواهد به شبکه متصل گردد. در این صورت پس از وارد کردن پیغام

```
CONNECT AS  $ID_{new}$  ON PORT  $Port$ 
```

در ترمینال کاربر مذکور، یک اتصال TCP با مدیر برقرار می‌شود. سپس درخواست زیر را برای مدیر ارسال می‌شود:

```
 $ID_{new}$  REQUESTS FOR CONNECTING TO NETWORK ON PORT  $Port$ 
```

با این پیام کاربر با شناسه‌ی ID_{new} به مدیر می‌گوید که می‌خواهد به شبکه متصل شود و روی پورت $Port$ در حال گوش دادن برای پیام‌های دریافتی خواهد بود.

سپس در جواب، مدیر پورت و شناسه‌ی گرهی را که کاربر جدید باید به آن متصل شود را به وی ارسال می‌کند (مطابق پیام زیر).

```
CONNECT TO  $ID_{parent}$  WITH PORT  $Port_{parent}$ 
```

اگر هیچ گرهی در شبکه وجود نداشت، مدیر مقدار 1- را برای هر دوی شناسه و پورت به کاربر برمی‌گرداند و کاربر متوجه می‌شود که ریشه‌ی درخت کاربران خواهد بود. توجه کنید که مدیر باید به گونه‌ای اضافه شدن گره‌های جدید به شبکه را مدیریت کند که درخت دودویی بماند و کمترین عمق را نیز داشته باشد. پورتی که مدیر برای کاربر ارسال می‌کند، در واقع پورت اتصال پدر وی در شبکه است.

پس از آنکه کاربر پورت گرهی که باید به آن متصل شود را دریافت کرد، ارتباط قبلی خود با مدیر را خاتمه داده و اگر خود ریشه نبود، یک ارتباط جدید با پدر خود در شبکه ایجاد می‌کند. سپس یک بسته از نوع ۴۱ (مطابق بخش بعد) می‌سازد، آدرس مبدأ آن را شناسه‌ی خود و آدرس مقصد را شناسه‌ی پدرش قرار می‌دهد و در قسمت داده شماره‌ی پورتی که روی آن به بسته‌های دریافتی گوش می‌دهد را می‌گذارد و پس از ارسال بسته، ارتباط را می‌بندد.

Host Local^۳
Thread^۴
ID^۵

۶ لایه‌ی شبه انتقال!

شما باید زیرساختی برای ارتباط‌های کاربردی در این شبکه را پیاده‌سازی کنید. ما اسم این زیرساخت را لایه‌ی شبه انتقال گذاشته‌ایم. شما بایستی کلاس Packet را پیاده‌سازی کنید. این کلاس ۴ صفت ۶ دارد:

- int :Type
- int :Source ID
- int :Destination ID
- string :Data

تمام پیام‌های منتقل شده در شبکه (جز پیام‌های شروع ارتباط) باید نمونه‌ای از این کلاس باشند. پس از ساخت یک نمونه و تکمیل اجزای آن، شیء مذکور از طریق ارتباط ارسال می‌گردد. نوع بسته‌های مختلف در جدول زیر آمده است.

بسته	نوع
۰	Message
۱۰	RoutingRequest
۱۱	RoutingResponse
۲۰	ParentAdvertise
۲۱	Advertise
۳۱	DestinationNotFoundMessage
۴۱	ConnectionRequest

یک ویژگی بسیار مهم در بسته‌ها آن است که اگر آدرس مقصد در بسته‌ای برابر 1- تنظیم شود، بسته بایستی به تمامی گره‌ها ارسال شود و هر گره نیز بایستی آن را به تمامی گره‌هایی که با آن‌ها در ارتباط است (جز گرهی که بسته را از آن دریافت کرده) ارسال نماید. همچنین تمام گره‌ها با توجه به محتوای بسته بایستی پاسخ لازم را به ارسال کننده باز گردانند.

۷ شناخت هر گره از شبکه

هر کاربر باید لیستی از گره‌هایی که از وجودشان در شبکه مطلع شده داشته باشد. کاربر تنها به گره‌هایی می‌تواند پیام دهد که از وجودشان مطلع شده است. مواردی که باعث شناخته شدن گره A توسط گره B می‌شود موارد زیر هستند.

- همه‌ی گره‌ها از گره پدرشان مطلع هستند.
- اگر A فرزند B باشد، در هنگام اضافه شدن A گره B از وجودش مطلع می‌شود.
- اگر یک بسته‌ی پیام از A به مقصد B فرستاده شود، B از حضور A مطلع می‌گردد. در صورتی که آدرس مقصد 1- قرار گرفته باشد، همه‌ی گره‌ها (به جز گره مبدأ) مقصد پیام محسوب می‌شوند.
- با ارسال پیام شروع چت، تمام گره‌های حاضر در چت به لیست گره‌های شناخته شده توسط گره دریافت کننده‌ی پیام اضافه می‌شوند.
- اگر در ترمینال گره A عبارت زیر نوشته شود، بایستی لیستی از شناسه‌های گره‌های دیگری که توسط A شناخته شده‌اند در ترمینال آن چاپ گردد.

SHOW KNOWN CLIENTS

همچنین در صورتی که پیغامی بخواهد از گره A ارسال شود ولی مقصد آن ناشناخته باشد، بایستی ارسال اتفاق بیفتد. در این حالت پیام زیر در ترمینال A چاپ می‌گردد. ($ID_{destination}$ شناسه‌ی مقصدی است که برای A ناشناخته است).

Unknown destination $ID_{destination}$

Attribute^۶

۸ مسیریابی و انتقال بسته‌ها

مسیریابی در این شبکه به صورت خاصی انجام می‌شود. هر گره بایستی شناسه‌ی گره‌هایی که در زیردرختش قرار دارند را داشته باشد. زمانی که گرهی با شناسه‌ی A بخواهد پیامی به گرهی با شناسه‌ی B بفرستد، ابتدا بررسی می‌کند که آیا B در زیردرختش قرار دارد یا خیر. اگر قرار داشت، بسته را به فرزندی می‌فرستد که B زیردرخت آن است و اگر قرار نداشت، بسته را به پدرش می‌فرستد. گرهی که یک پیام از فرزندان یا پدرش دریافت می‌کند که مقصدش خود وی نیست، همین فرآیند را تکرار می‌کند و بسته را به گره بعدی انتقال می‌دهد. اگر بسته‌ای به ریشه رسید و ریشه شناسه‌ی مقصد را در لیست فرزندانیش نیافت، یک بسته‌ی نوع ۳۱ می‌سازد و عبارت زیر را نیز در قسمت داده‌ی آن قرار می‌دهد و برای آدرس مبدا بازگشت می‌دهد. ارسال‌کننده‌ی پیام بایستی این پیغام را در ترمینال خود چاپ کند (در صورتی که حالت گفت‌وگو نباشد. این حالت در بخش لایه‌ی کاربرد توضیح داده شده است) که در آن $ID_{Destination}$ شناسه‌ی مقصد است.

DESTINATION $ID_{Destination}$ NOT FOUND

۹ نوع بسته‌ها

بسته‌ها انواع مختلفی دارند که در صفت نوع در کلاس بسته تعیین می‌شود.

۱.۹ مسیریابی

فرض کنید کاربر A بخواهد مسیری که از وی تا کاربر B برای ارسال پیام طی می‌شود را بیابد. به این منظور ابتدا پیام زیر در ورودی کاربر A وارد می‌شود:

ROUTE ID_B

سپس A یک بسته از نوع ۱۰ می‌سازد و مبدا و مقصد را به تناسب قرار می‌دهد و پیام را مطابق بخش ۸ ارسال می‌کند. بسته مسیر را تا مقصد ادامه می‌دهد. پس از آنکه B پیام را دریافت کرد، یک بسته از نوع ۱۱ می‌سازد و مبدا آن را شناسه‌ی خود و مقصد را شناسه‌ی A قرار می‌دهد. همچنین شناسه‌ی خود را نیز در قسمت داده‌ی بسته قرار می‌دهد. سپس بسته را برای A ارسال می‌کند. هر گره در مسیر که بسته‌ی نوع ۱۱ را دریافت کرد، شناسه‌ی خود و اینکه بسته را از فرزند یا پدرش دریافت کرده است را در قسمت داده اضافه می‌کند. در انتها و پس از رسیدن پیام به A، مسیر بین A و B به دست A می‌رسد و A بایستی آن را در ترمینال چاپ کند.

۱.۱.۹ مثال

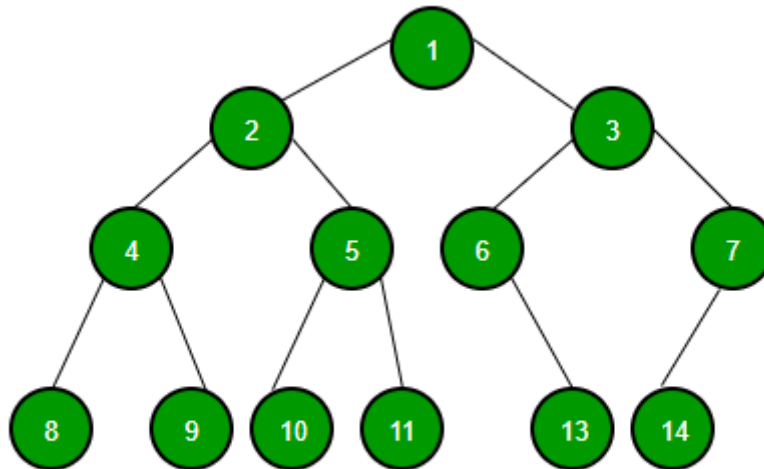
فرض کنید در تصویر ۱ گره ۷ بخواهد مسیر خود تا ۹ را بیابد. ابتدا یک بسته از نوع ۱۰ می‌سازد. چون ۹ در زیردرخت ۷ نیست، آن را برای پدرش ۳ ارسال می‌کند. ۳ بسته را دریافت می‌کند. چون ۹ در زیر درختش نیست آن را برای ۱ ارسال می‌کند. ۱ می‌داند که ۹ در زیردرخت فرزندش با شناسه‌ی ۲ است، بسته را برای وی ارسال می‌کند. ۲ نیز مانند ۱ عمل می‌کند و بسته را برای ۴ ارسال و ۴ نیز برای ۹ ارسال می‌کند. ۹ بسته را دریافت می‌کند و می‌بیند که یک بسته‌ی مسیریابی با مقصد وی است. یک بسته از نوع ۱۱ می‌سازد، آدرس مبدا آن را ۹ و آدرس مقصد را مبدا بسته‌ی دریافتی (۷) قرار می‌دهد. سپس در قسمت داده، صرفاً شناسه‌ی خود را قرار می‌دهد:

9

بسته را برای ۴ ارسال می‌کند. ۴ بسته را دریافت می‌کند و می‌بیند که از نوع ۱۱ است. شناسه‌ی خود و رشته‌ی -> را به ابتدای قسمت داده می‌افزاید (چون بسته را از فرزندش دریافت کرده است) و مطابق مسیریابی گفته شده بسته را برای ۲ می‌فرستد:

4->9

به همین ترتیب ۲ و ۱ نیز بسته‌ای از نوع ۱۱ را به سمت ۳ عبور می‌دهد و بسته با پیام زیر به ۳ می‌رسد.



تصویر ۱: یک درخت دودویی.

1 -> 2 -> 4 -> 9

۳ بسته را دریافت می‌کند و مطابق مسیریابی برای ۷ ارسال می‌کند. تنها تفاوت اینجاست که چون ۳ بسته‌ی از نوع ۱۱ را از پدرش دریافت کرده، به جای -> استفاده می‌کند و پیام زیر به دست ۷ می‌رسد.

3 <- 1 -> 2 -> 4 -> 9

۷ نیز چون بسته را از پدرش دریافت کرده، عبارت مناسب را به قسمت داده اضافه کرده و آن را در ترمینالش چاپ می‌کند:

7 <- 3 <- 1 -> 2 -> 4 -> 9

۲.۹ تبلیغات

دو نوع تبلیغات در شبکه وجود دارد.

۱.۲.۹ تبلیغ شناسه به گره‌های پدر

هنگامی که یک گره به گره پدرش در شبکه اتصال می‌یابد، تمام گره‌هایی که گره جدید در زیردرخت آن‌هاست باید از اضافه شدن وی مطلع گردند. این اقدام بدین صورت اتفاق می‌افتد که هر گره‌ی که از اضافه شدن گره جدید مطلع می‌گردد، آن را در یک بسته از نوع ۲۰ به پدرش اطلاع می‌دهد. مثلاً در تصویر ۱

فرض کنید گره ۱۵ به عنوان فرزند گره ۶ به درخت اضافه شود. گره ۶ بایستی یک بسته از نوع ۲۰ بسازد و آدرس مبدا آن را شناسه‌ی خود و آدرس مقصد را شناسه‌ی گره ۳ قرار دهد. سپس در قسمت داده، شناسه‌ی گره ۱۵ را قرار داده و بسته را برای ۳ ارسال نماید. ۳ نیز پس از دریافت بسته و انجام اقدامات لازم، بسته‌ی مشابهی را از طرف خودش به ۱ می‌فرستد و اضافه شدن ۱۵ به زیردرخت وی را اطلاع می‌دهد.

۲.۲.۹ تبلیغ عمومی

با وارد کردن دستور زیر در ترمینال یک گره: گره پیامی از نوع ۲۱ می‌سازد و برای مقصد می‌فرستد. به این ترتیب، با ارسال پیام به شناسه‌ی مقصد، مقصد از حضور گره مطلع می‌گردد.

`Advertise IDDestination`

اگر شناسه‌ی مقصد 1- تنظیم شود، پیام حضور در شبکه به همه‌ی گره‌ها ارسال می‌گردد.

۳.۹ پیام

این بسته‌ها از نوع ۰ هستند و به عنوان بسته‌ی پوشش برای ارتباطات لایه‌ی کاربرد به کار می‌روند. قسمت داده در این پیام‌ها بسته به کاربرد تعیین می‌گردد.

۱۰ لایه‌ی کاربرد

بر روی زیرساخت گفته شده، می‌خواهیم دو قابلیت را ایجاد کنیم.

۱.۱۰ سلام علیک

این قابلیت برای به جا آوردن ادب در شبکه قرار گرفته است. به این صورت که اگر گره A یک بسته‌ی پیام برای B ارسال کند که محتوای آن

`Salam Salam Sad Ta Salam`

باشد، مقصد بایستی پیام زیر را در جواب برگرداند.

`Hezaro Sisad Ta Salam`

۱.۱.۱۰ مثال

اگر گره‌ی یک پیام سلام را با آدرس مقصد 1- ارسال کند، سلی از پیام‌های محبت‌آمیز جواب سلام از همه‌ی گره‌ها به سمت وی ارسال می‌شود. (بالاخره جواب سلام واجب است دیگر!)

۲.۱۰ دورهمی

شما بایستی یک محیط چت چند نفره را در با پروتوکل توضیف شده در ادامه پیاده‌سازی کنید. در خط اول بخش داده‌ی تمام بسته‌های پیام مرتبط با چت که در شبکه منتقل می‌شوند (در تمام مراحل چت، یعنی آغاز چت، پیام‌های مربوط به چت و پایان یافتن آن) عبارت زیر نوشته می‌شود:

`CHAT:`

و در خطوط بعدی مواردی که در ادامه گفته خواهد شد می‌آید.
کارکرد این چت به صورت زیر است:
ابتدا دستور زیر در ترمینال یک گره مانند A وارد می‌شود:

`START CHAT CHAT_NAMEA: ID1, ID2, ID3`

کاربران در هر چت اسم مستعاری دارند. زمانی که A پیام شروع ارتباط را می‌فرستد بایستی نام خود را در $CHAT_NAME_A$ مشخص نماید. هر تعداد شناسه‌های یکتا به غیر از شناسه‌ی خود گره شروع کننده‌ی ارتباط بعد از START CHAT می‌توانند بیایند. اگر شناسه‌ای در لیست شناسه‌های شناخته شده توسط A نبود، از لیست کنار گذاشته می‌شود. گره شروع کننده‌ی ارتباط پیام زیر را برای تمام شناسه‌ها ارسال می‌کند.

REQUESTS FOR STARTING CHAT WITH $CHAT_NAME_A$: ID_A, ID_1, ID_2, ID_3

که ID_A شناسه‌ی شروع کننده‌ی ارتباط است. در واقع پس از نام مستعار A شناسه‌ی تمام گره‌های شرکت کننده در ارتباط را برای همه‌ی گره‌ها ارسال می‌کند. ID_A بایستی اولین شناسه در شناسه‌های ارسالی برای همه باشد. در ترمینال هر یک از گره‌هایی که پیام را دریافت می‌کنند پیام زیر نوشته می‌شود:

$CHAT_NAME_A$ with id ID_A has asked you to join a chat. Would you like to join?[Y/N]

در صورتی که کاربر N (یا هر عبارت دیگری غیر از Y) را وارد کرد درخواست رد می‌شود و به دورهمی اضافه نمی‌شود. اگر Y را وارد کرد، پیام زیر در ترمینال وی چاپ می‌شود:

Choose a name for yourself

سپس کاربر یک نام برای خود انتخاب می‌کند. پس از وارد شدن این نام از سوی i، پیام زیر برای تمام شناسه‌هایی که در لیست ارسالی از سوی A حضور داشتند فرستاده می‌شود:

$ID_i : CHAT_NAME_i$

اگر چنین پیامی به فردی که حضور در دورهمی را پذیرفته است برسد، باید پیام زیر در ترمینال وی چاپ گردد:

$CHAT_NAME_i(ID_i)$ was joind to the chat.

در طی گفت و گو، هر پیامی که توسط هر یک از افراد حاضر در چت در ترمینال نوشته شود، برای باقی ارسال می‌گردد. پیام‌ها در ترمینال کاربرانی که در چت حضور دارند پس از نوشتن نام فرد ارسال کننده چاپ می‌شود. برای مثال اگر فرد A با اسم Taghi پیام Salam be Hamegi را در چت بنویسد، در ترمینال افراد حاضر در گفت‌وگو Taghi: Salam be Hamegi چاپ می‌گردد. هنگامی که یک کاربر چت را پذیرفته باشد، تمامی پیام‌های وی که در ترمینال نوشته می‌شود به عنوان یک پیام در گفت‌وگو در نظر گرفته شده و تاثیری به عنوان دستور ندارد. مثلاً اگر فرد در زمانی که در چت می‌باشد، عبارت

Advertise -1

را بنویسد، این پیام به معنی تبلیغ شناسه‌ی وی به تمام گره‌ها نخواهد بود؛ بلکه تنها متن این پیام به تمام کسانی که در گفت و گو هستند ارسال می‌شود. برای خروج از چت، در صورتی که فرد با شناسه‌ی i که در چت حاضر است، پیام زیر را در ترمینال خود بنویسد، از گفت‌وگو خارج می‌شوند.

EXIT CHAT

در این شرایط، به تمام افرادی که در لیست اولیه‌ی چت که از طرف شروع کننده رسیده بود پیام زیر ارسال می‌گردد:

EXIT CHAT ID_i

سپس پیام زیر باید در ترمینال کاربرانی که حضور در چت را پذیرفته بودند و همچنان در گفت‌وگو حضور دارند نوشته شود:

CHAT_NAME_i(ID_i) left the chat.

نکته‌ی مهم: فرد شروع‌کننده‌ی گفت‌وگو از زمان نوشته شدن دستور شروع گفت‌وگو در ترمینالش در چت حضور دارد و برای خروج از آن باید مطابق آنچه گفته شد عمل کند. زمانی که یک فرد در یک دورهمی حضور دارد، درخواست‌های دیگر دورهمی که به وی ارسال می‌گردد نادیده گرفته می‌شوند و تاثیری نخواهند داشت.

۱۱ دیوار آتش

بر روی هر گره بایستی یک دیوار آتش قرار داشته باشد که با دستورات زیر عمل کند. برای اجرای این دستورات گره اجراکننده نباید در حالت چت باشد.

۱.۱۱ دیوار آتش پالایش‌گر بسته

این دیوار، با توجه به سرآیند بسته‌های عبوری عمل می‌کند و دستورات آن به صورت زیر هستند:

FILTER Direction ID_{Source} ID_{Destination} Type Action

- **Direction** جهت بسته را مشخص می‌کند. این بخش می‌تواند یکی از حالت‌های INPUT (بسته‌های که مقصدشان گره اجراکننده است)، OUTPUT (بسته‌هایی که مبدأشان گره اجراکننده است) یا FORWARD (بسته‌ای که مبدأ آن گره اجراکننده نیست و مقصد دیگری نیز دارد) باشد. دیوار آتش بر روی قسمتی که در این بخش مشخص می‌شود عمل می‌کند.
- **ID_{Source}** شناسه‌ی مبدأ بسته‌هایی نشان می‌دهد. در صورتی که با * مشخص گردد به معنای همه‌ی شناسه‌هاست. اگر **Direction** برابر OUTPUT باشد، این مقدار نادیده گرفته می‌شود.
- **ID_{Destination}** شناسه‌ی مقصد بسته‌ها را نشان می‌دهد. در صورتی که با * مشخص گردد به معنای همه‌ی شناسه‌هاست. اگر **Direction** برابر INPUT باشد این مقدار نادیده گرفته می‌شود.
- **Type** نوع بسته‌هایی را مشخص می‌کند که دستور بر روی آن‌ها اعمال می‌گردد. این مقدار می‌تواند مقادیر نوع بسته‌ها باشد که در جدول بخش ۶ معرفی گردید.
- **Action** عمل مربوط بر روی بسته‌هایی که در قسمت‌های قبل تعیین شده‌اند را مشخص می‌کند. می‌تواند برابر AC-CEPT (قبول بسته) یا DROP (حذف بسته) باشد.

۲.۱۱ دیوار آتش سطح کاربرد

در سطح کاربرد نیز می‌توان از دستور زیر برای فعال و غیرفعال کردن قابلیت چت استفاده کرد.

FW CHAT Action

تعریف **Action** مشابه دیوار آتش پالایش‌گر بسته است. در صورتی که دستور زیر وارد شود، درخواست‌های چت دیگر برای کاربر نمایش داده نمی‌شوند و خود کاربر نیز نمی‌تواند چتی را شروع کند.

FW CHAT DROP

اگر در این حالت کاربر بخواهد دورهمی‌ای را آغاز کند، پیام زیر در ترمینال وی چاپ می‌گردد.

Chat is disabled. Make sure the firewall allows you to chat.

این دیوار آتش صرفاً بر روی بسته‌هایی که مبدأ یا مقصدشان گره اجراکننده‌ی دستور است اجرا می‌شود.

۱۲ لاگ‌ها

در طول اجرا، اگر گرهی در چت حضور داشت تنها پیام‌های رد و بدل شده طی چت به وی می‌رسند. اگر گره در حالت چت نباشد، برای بسته‌هایی که به مقصد وی هستند مطابق موارد گفته شده در قسمت‌های قبل خروجی‌های مربوطه بایستی چاپ گردد. اگر گره مقصد یک پیام نباشد ولی بسته‌ای به وی برسد، بایستی پیام زیر در ترمینال گره چاپ شود:

Type Packet from ID_{Source} to $ID_{Destination}$.

در این پیام ID_{Source} ، $ID_{Destination}$ و $Type$ به ترتیب شناسه‌ی مبدا، شناسه‌ی مقصد و نوع پیام هستند. نوع پیام براساس جدولی که در بخش ۶ آمده است مشخص می‌گردد.

۱۳ نکات دیگر

- برای پیاده‌سازی پروژه از زبان پایتون یا جاوا استفاده نمایید.
- پروژه را می‌توانید به صورت انفرادی یا در گروه‌های دو یا سه نفره انجام دهید.
- استفاده از کد نوشته شده توسط دیگر افراد یا گروه‌ها ممنوع است.
- مهلت ارسال پروژه ساعت ۲۳:۵۵ روز ۲۶ تیر می‌باشد. زمان تحویل آنلاین متعاقبا اعلام می‌گردد.
- در صورت داشتن هرگونه سوال از طریق Quera اقدام کنید.

موفق باشید