



## اهداف تمرین

- آشنایی با پروتکل DHCP
- آشنایی با سامانه‌ی پرتو
- آشنایی با ارتباطات کارگزار-کارخواه
- آشنایی با لایه‌ی دو و Ethernet
- آشنایی با Socket

## ۱. مقدمه

این تمرین شامل دو بخش است. بخش اول مربوط به برنامه نویسی سوکت و آشنایی با آن است و در سطح خیلی ساده طراحی شده است. بخش دوم و اصلی تمرین مربوط به DHCP و یادگیری کار با سامانه‌ی پرتو است که در تمرین‌های آینده به آن نیاز دارید.

از این رو توزیع نمره به طور کلی حدود ۲۰ درصد مربوط به بخش اول و ۸۰ درصد مربوط به بخش دوم است. جزئیات نمره دهی به همراه سامانه داوری مطابق زمان‌بندی ارائه می‌شود.

\* با سپاس از مهدی بهروزی‌خواه، امیرپاشا قابوسی، آرش ملاجعفری و سولماز سلیمی

## ۲. بخش اول: برنامه نویسی سوکت

### ۱.۲. مقدمه

همانطور که در درس خواندید، برای انتقال داده‌ها می‌توانیم از واسط برنامه‌نویسی سوکت استفاده کنیم. در این تمرین، هدف این است که شما یک بار با برنامه‌نویسی سوکت آشنا شوید و یک ارتباط ساده برقرار کنید.

### ۲.۲. توضیح تمرین

به صورت ساده، شما باید کد همتهایی<sup>۱</sup> را بزنید که هر کدام قابلیت این را داشته باشند که به دیگری وصل شوند و به یکدیگر پیام ارسال نمایند. هر همته، در واقع یک پردازنده در سامانه است که هنگام اجرا، درگاهی<sup>۲</sup> که قرار است روی آن گوش کند را دریافت می‌کند. در صورتی که همتهایی به شما متصل شد، دیگر نیازی نیست روی این درگاه گوش کنید و منتظر ارتباط‌های بعدی باشید و تنها کافیست با این همته ارتباط برقرار کنید. پس به طور کلی رفتار برنامه‌ی شما اینگونه است:

ابتدا با دستور

```
./start PORT
```

برنامه شما شروع به اجرا می‌کند و روی درگاه وارد شده گوش می‌کند.

نکته: عدد وارد شده بزرگتر از ۱۰۰۰۰ است، پس نیازی نیست نگران اشغال درگاه‌های رزرو شده باشید.

هنگامی که پردازنده دیگری به شما متصل شد، باید عبارت

```
connected to PORT
```

را در هر دو طرف اتصال چاپ کنید که هر کدام، شماره‌ی درگاهی که سر دیگر سوکت آن‌ها به آن متصل شده را چاپ می‌کنند.

همچنین در صورتی که در خط فرمان دستور

```
connect PORT
```

وارد شد، باید از طریق ارتباط سوکت به این پردازنده با پورت مبدأ خودتان به‌علاوه‌ی یک، وصل شوید. بنابراین سر دیگر ارتباط شما را با یک عدد بیشتر می‌بیند.

در صورتی که به همته‌ای دیگر متصل شدید، با دستور

```
exit
```

<sup>۱</sup>peer  
<sup>۲</sup>port

ارتباط خود را با سر دیگر قطع کنید. و دوباره طرفین روی درگاهی که گوش می‌کردند، منتظر می‌مانند.  
هنگامی که دستور

`send Message`

را دیدید، برای همتایی که به آن متصل هستید، پیامی که به جای قسمت Message آمده را به ترتیب کاراکتر از چپ به راست ارسال کنید و چیز بیشتر یا کمتری ارسال نکنید. سپس در طرف دیگر این پیام را اینگونه چاپ کنید:

`recv Message`

نکته: اگر در ارتباطی قرار داشتید، با دیدن دستور connect یا اگر متصل نبودید با دیدن دستور exit و send پیام

`invalid command`

را چاپ کنید.

## ۳.۲. نکات پیاده سازی

برای پیاده سازی می‌توانید از دو زبان ++c و جاوا استفاده کنید. استفاده از ابزار telnet یا nc می‌تواند برای تست برنامه به شما کمک کند.

آموزش بسیار خوبی برای زبان ++c در مورد socket وجود دارد با نام beej's که در وب‌گاه درس موجود است. خواندن بخشی از آن شما را برای تمام قسمت‌های این تمرین آماده می‌کند.

کار با زبان جاوا راحت‌تر است و پیچیدگی‌های کمتری دارد، ولی فراموش نکنید که واسط برنامه‌نویسی سوکت برای ارتباط با سامانه‌عامل طراحی شده و سامانه‌های لینوکسی با زبان سی توسعه پیدا کرده و به همین دلیل برنامه‌های سی ارتباط بهتری با سامانه دارند، اگرچه پیچیدگی‌های بیشتری هم دارند.

در هر صورت، چه با زبان جاوا و چه با زبان ++c باید فایل شما قابلیت این را داشته باشد که با دستور make کامپایل شده و سپس با دستوراتی که پیشتر گفته شد، اجرا شود در غیر اینصورت شما هیچ نمره‌ای از این تمرین نخواهید گرفت!

پس از پایان کار پروژه‌ی خود را آرشیو (با قالب زیپ) نمایید و در وب‌گاه quera بارگذاری کنید. (فایل زیپ شما باید به گونه‌ای باشد که Makefile و فایل start در اولین پوشه باشند.) نام این پوشه را PA0\_1 قرار دهید.

## ۳. بخش دوم: DHCP

### ۱.۳. مقدمه

همان‌طور که در درس خواندید، در یک شبکه‌ی لایه سوم، نیاز داریم تا اجزاء شبکه با ساختارهایی مانند IP با هم ارتباط برقرار کنند<sup>۳</sup>. از این رو باید به هر گره در این شبکه یک IP نسبت دهیم. انتساب IP روش‌های مختلفی دارد که سه روش کلی آن به صورت زیر است:

- دستی: در این روش، مدیر شبکه به صورت دستی به هر عضو شبکه یک IP نسبت می‌دهد. این روش کمی سخت است و زمان بیشتری نسبت به سایر روش‌های دیگر نیاز دارد، اما در عوض امن‌ترین روش برای اختصاص IP است.

- خودکار: در این روش به صورت خودکار به هر گره شبکه یک IP جدید اختصاص داده می‌شود.

- پویا: در این روش در واقع برای هر درخواستی از جانب اعضای شبکه، یک قرارداد وضع می‌شود و IP جدیدی براساس این قرارداد به آن گره اختصاص پیدا می‌کند. در نتیجه اعضا می‌توانند به شبکه وارد یا از آن خارج شوند.

روش پویا در این میان، پرکاربردترین روش برای اختصاص IP است و تمرکز تمرین نیز روی این بخش است.

### ۲.۳. مقدمه‌ای بر DHCP

پروتکل Dynamic Host Configuration Protocol (DHCP) در واقع به نوعی نسخه‌ی به‌روز شده‌ی پروتکل Bootstrap Protocol (BOOTP) است و به صورت عقبگرد از این پروتکل پشتیبانی می‌کند.

همان‌طور که گفته شد، این پروتکل مواقعی کاربرد دارد که اعضای شبکه به صورت موقتی به شبکه وارد و از آن خارج می‌شوند. در این مواقع، ما نیاز به دست‌کم یک کارگزار DHCP داریم. وظیفه کارگزار DHCP این است که به عنوان مدیر شبکه عمل کند و آی‌پی‌ها را مدیریت کند.

هر عضو شبکه که درخواستی دارد، به این کارگزار درخواست ارسال می‌کند و او به ازای مک آدرسی که درخواست را ارسال کرده، یک آی‌پی پیشنهاد می‌دهد. ممکن است در شبکه چند کارگزار DHCP وجود داشته باشند در نتیجه چند پیشنهاد برای این کارخواه می‌آید. او یکی از آن‌ها را انتخاب می‌کند و پیام را به کارگزار مربوطه می‌فرستد. کارگزار در صورت موافقت، پیامی را به همگان broadcast می‌کند تا سایر پیشنهادات را لغو کند و کارخواه<sup>۴</sup> این IP را بگیرد.

---

<sup>۳</sup>وبگاه [https://en.wikibooks.org/wiki/Communication\\_Networks/DHCP\\_Protocol](https://en.wikibooks.org/wiki/Communication_Networks/DHCP_Protocol) مرجع توضیحات مقدمه و تصاویر مربوط به پروتکل DHCP است.

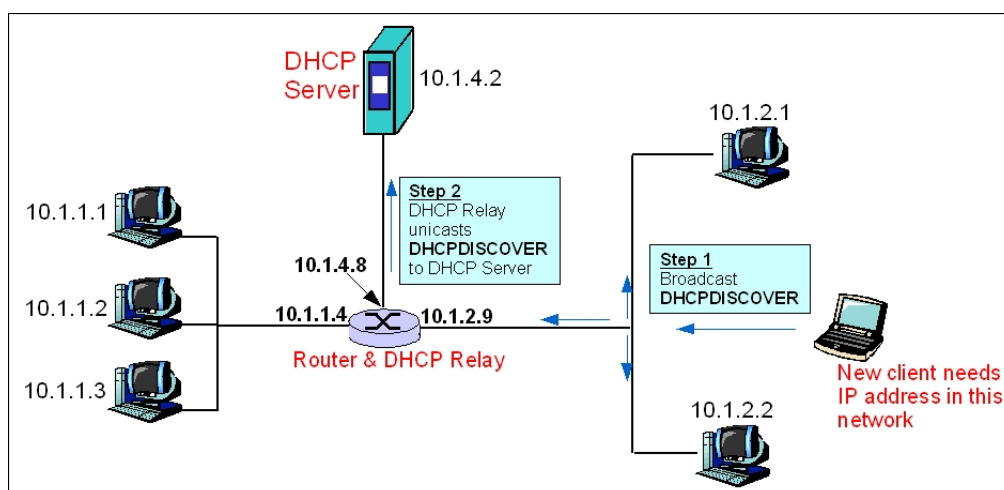
<sup>۴</sup>client

بسته به نوع پیکربندی شبکه، ممکن است در شبکه لایه دو، کارگزار موجود باشد یا کارگزار در یک شبکه دیگر باشد. در ادامه یک مثلاً از فرآیندهای DHCP برای شما آمده است.

۱. فرض کنید یک عضو جدید به شبکه اضافه شده است و می‌خواهد آی‌پی دریافت کند. پس ابتدا پیامی مبنی بر DHCPDISCOVER در کل شبکه broadcast می‌کند. هدف از این بسته پیدا کردن کارگزاران DHCP است. محتوای این بسته به این صورت است که مک آدرس مبدأ را شامل می‌شود و از درگاه ۶۸ به آی‌پی 255.255.255.255 و درگاه ۶۷ ارسال می‌شود. همچنین درخواست می‌تواند شامل آی‌پی درخواستی و مدت زمان اعتبار این آی‌پی باشد.

۲. حال این بسته به دست DHCP relay می‌رسد و او چون آدرس DHCP Server را می‌داند بسته را برای DHCP Server به صورت unicast ارسال می‌کند. همچنین فیلد giaddr را با آدرس gateway 10.1.2.9 پر می‌کند تا کارگزار DHCP بداند که برای کدام زیرشبکه<sup>۵</sup> باید آی‌پی اختصاص دهد.

شکل ۱: ارسال DHCPDISCOVER از طرف گرهی جدید



۳. حال که بسته دست کارگزار رسید، یک آی‌پی جدید پیشنهاد می‌دهد و بسته حاوی آدرس جدید یعنی بسته‌ی DHCPOFFER را broadcast می‌کند.

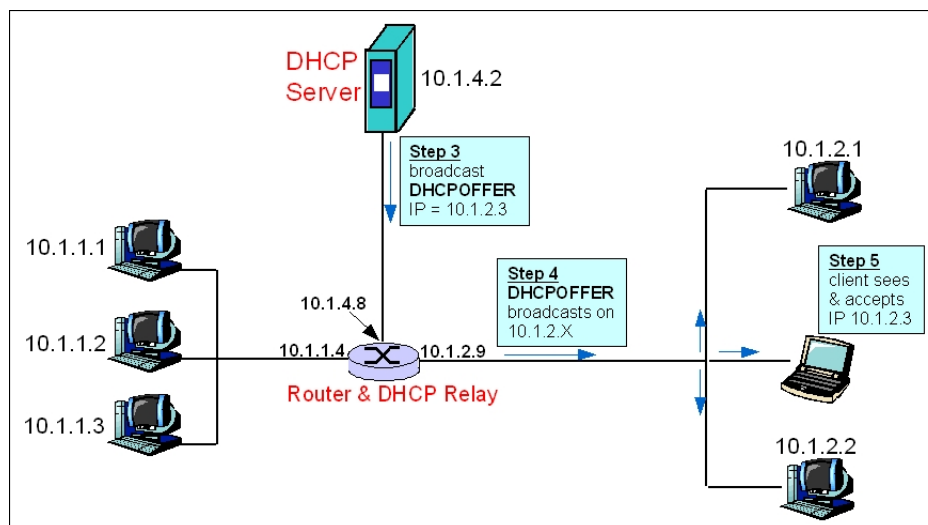
۴. سپس DHCP Relay بسته‌ی DHCPOFFER را تنها روی واسط مورد نظر broadcast می‌کند.

۵. حال کارخواه و سایر اعضای شبکه بسته پیشنهادی را می‌بینند و کارخواه ما در صورت تمایل، آی‌پی پیشنهادی را قبول می‌کند.

۶. کارخواه، در صورت قبول پیشنهاد، بسته DHCPREQUEST را برای کارگزار می‌فرستد.

<sup>۵</sup>sub-network

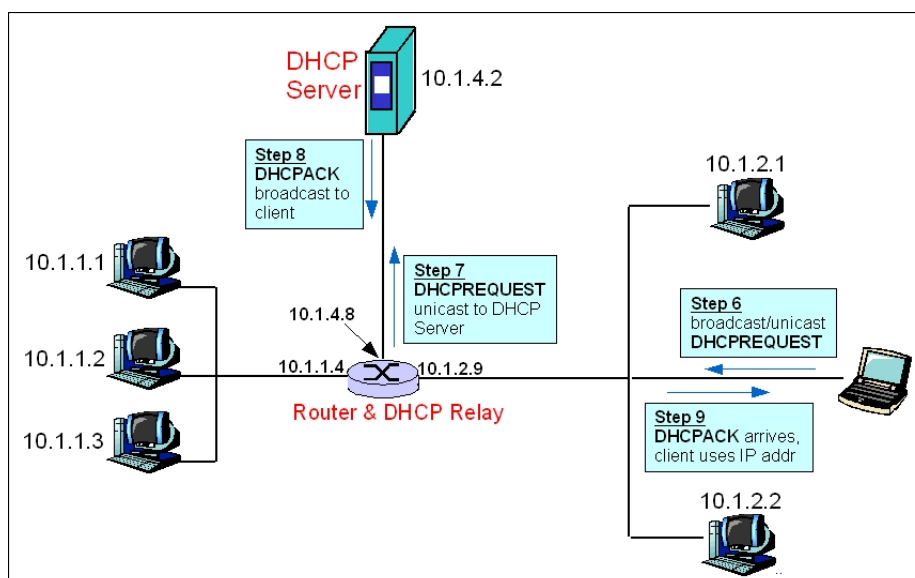
شکل ۲: ارسال بسته‌ی DHCPPOFFER



۷. DHCP relay بسته را به سرور می‌فرستد.

۸. سرور بسته DHCPACK را در صورت تایید برای تمام اعضای شبکه broadcast می‌کند تا سایر DHCPPOFFER ها از بین بروند و این کارخواه تنها همین آی‌پی را بگیرد. در صورتی که موافق نباشد می‌تواند بسته DHCPNACK را بفرستد و پس از این کارخواه چاره‌ای ندارد جز اینکه همه مراحل را از اول شروع کند.

شکل ۳: ارسال بسته‌ی DHCPREQUEST و دریافت پاسخ



به‌جز بسته‌های مطرح‌شده در سناریو بالا، بسته‌های دیگری نیز می‌توانند در این پروتکل وجود داشته باشند مانند

درخواست تجدید زمان، درخواست رهاسازی آی پی گرفته شده، درخواست دریافت اطلاعات بیشتر. از نظر امنیتی، این پروتکل ناامن است، چرا که روش درستی برای احراز هویت در آن وجود ندارد. برای مثال کارگزار نمی‌داند آیا مک آدرسی که درخواست آی پی جدید دارد واقعاً در شبکه موجود است یا یکی از گره‌ها این درخواست را داده است. یا مثلاً کارخواه نمی‌داند پیشنهادات از سوی یک کارگزار واقعی است یا خیر.

### ۳.۳. توپولوژی شبکه

توپولوژی این سوال به صورت درختی است. هر درخت می‌تواند چندین کارگزار و یا کارخواه داشته باشد. تضمین می‌شود در تمام تست‌ها حداکثر ۱۵ عضو در شبکه حاضر باشند. همچنین تضمینی برای وجود کارگزار وجود ندارد. ممکن است دو کارگزار مجاور هم باشند. تضمین می‌شود که در تمام تست‌ها، شبکه همبند است و تمام اجزا شبکه به نوعی بهم متصل‌اند.

### ۴. توضیح تمرین

هدف شما در این تمرین پیاده سازی پروتکل DHCP در یک شبکه‌ی درختی است. تمرکز اصلی تمرین در لایه‌ی دو است و شما باید سعی کنید بسته‌ها را به درستی مسیریابی کنید و به مقصد برسانید.

دو نوع گره در شبکه وجود دارند که هر کدام کارهای مخصوص به خود را انجام می‌دهند. شما باید دستورات گفته شده برای هر کدام از این انواع را پیاده سازی کنید. برای داوری و اجرای برنامه‌ها شما نیاز به اتصال به شبکه پرتو دارید.

ساختار تمرین به این گونه است که شما باید در نقش هر کدام از کارگزارها و کارخواه‌ها بسته‌ها را دریافت کنید، در صورت نیاز ارسال کنید و دستوراتی از صفحه کلید بگیرید و با توجه به دستورات کارهایی انجام دهید. نقش کارخواه این‌گونه است که این گره‌ها در شبکه حضور دارند و نیازمند دریافت آی پی هستند و کارگزاران باید برای هر کارخواهی که درخواست آی پی ارسال کرده است، آی پی جدیدی در نظر بگیرند. در این میان شما باید بسته‌ها را مسیریابی کنید و به مقصد برسانید. همچنین ممکن است آی پی‌ها منقضی شوند و یا کارخواهی درخواست آی پی جدیدی داشته باشد. پس به طور کلی شما باید همه این فرایندها را پیاده سازی کنید.

نکته: در طول تمرین می‌توانید فرض کنید در هر مرحله حداکثر یکی از فرایندها در حال اجراست و تا تمام نشود، دستور بعدی وارد نمی‌شود. یعنی به طور مثال هنگامی که کارخواهی درخواست آدرس جدید می‌دهد، تا این بسته در کل شبکه نچرخیده باشد و تمام پیشنهادهای کارگزاران را ندیده باشد، در هیچ کارخواه دیگری دستوری وارد نمی‌شود. اما پس از آن ممکن است قبل از اینکه پیشنهادی را قبول کند، در کارخواه دیگری دستور درخواست آدرس جدید وارد شود.

## ۱.۴. انواع بسته‌ها

تمامی بسته‌هایی که در این تمرین تولید و بین گره‌ها جابه‌جا می‌شوند، ساختار زیر را دارند و شما موظفید تمام این قسمت‌ها را پر کنید، سپس بسته خود را ارسال کنید:

Ethernet	Data
14 Byte	11 Byte

جدول ۱: ساختار بسته‌ها

### ۱.۱.۴. Ethernet:

آدرس مبدأ را آدرسی که در Interface ارسالی شما نوشته شده است بگذارید، آدرس مقصد را Broadcast و Type را برابر ۰ (۰x۰۰۰۰) قرار دهید.<sup>۶</sup>

### ۲.۱.۴. Data:

با توجه به نوع بسته، محتویات بسته متفاوت خواهد بود. در ادامه جزییات این قسمت به صورت جدول آمده است. دقت کنید که همیشه قسمت MAC را برابر با مک آدرس واسط شماره ۰ کارخواه بگذارید و هیچ‌گاه در قسمت IP و یا MAC اطلاعاتی از کارگزار قرار ندهید.

Name	Sender	Data Type(1)	Mac(6)	IP(4)
DHCPDISCOVER	Client	0	Mac	0
DHCPOFFER	Server	1	Mac	Offer IP
DHCPREQUEST	Client	2	Mac	Offer IP
DHCPACK	Server	3	Mac	Offer IP
DHCPRELEASE	Client	4	Mac	Release IP
DHCPTIMEOUT	Server	5	Mac	Release IP
Request Extend	Client	6	Mac	Extend IP
Response Extend	Server	7	Mac	New IP

جدول ۲: انواع بسته‌ها

<sup>۶</sup>[https://en.wikipedia.org/wiki/Ethernet\\_frame#Ethernet\\_II](https://en.wikipedia.org/wiki/Ethernet_frame#Ethernet_II)



## ۲.۴. کارخواه

برنامه‌ی شما در نقش کارخواه وظایفی دارید که به شرح زیر است:

- مسیریابی بسته‌ها

به طور کلی شما وظیفه دارید تمام بسته‌هایی را که دریافت می‌کنید، در صورتی که این بسته متعلق به شما نیست، روی تمام واسط‌هایتان (به جز واسطی که بسته از آن آمده) بسته را broadcast کنید.

- دریافت دستور

```
get ip
```

از کاربر.

هنگامی که کاربر این دستور را وارد کرد، شما وظیفه دارید تا یک IP جدید برای کاربر درخواست کنید (ارسال بسته‌ی DHCPDISCOVER) فرآیند درخواست به این صورت است که شما بسته درخواست را روی تمام واسط‌های خود می‌فرستید. حال باید منتظر بمانید تا کارگزاری به درخواست شما با بسته DHCPOFFER پاسخ دهد.

- دریافت بسته DHCPOFFER

هنگامی که این بسته را دریافت کردید، در صورتی که این بسته متعلق به شما بود آن را ذخیره کنید. سپس در خروجی پیام

```
new offer: X
```

که در اینجا  $X$  در واقع همان آدرس پیشنهاد شده به شماست.

- دریافت دستور

```
accept offer: X
```

از ورودی که  $X$  یکی از آدرس‌های پیشنهادی است.

در صورتی که پیشنهادی با آدرس درخواستی برای شما نیامده بود پیام

```
invalid offer
```

را چاپ کنید. در غیر این صورت، بر روی تمام واسط‌های خود بسته DHCPREQUEST را ارسال کنید.

- دریافت بسته DHCPACK

در این زمان درخواست شما تایید شده و در نتیجه شما IP درخواستی خود را دریافت کرده‌اید. پس پیام

```
now my ip is: X
```

که  $X$  آدرس دریافتی است، را چاپ کنید.

- دریافت دستور

```
release X
```

اگر از قبل آی‌پی  $X$  را گرفته بودید، پس از دریافت این دستور، بسته‌ای از جنس DHCPRELEASE تولید کنید و آن را روی تمام واسطه‌های خود بفرستید. همچنین عبارت

```
ip released
```

را در خروجی چاپ کنید و فرض کنید دیگر IP شما وجود ندارد.

نکته: اگر از قبل آی‌پی  $X$  را نگرفته بودید، نیازی نیست کاری انجام دهید یا چیزی چاپ کنید.

- دریافت بسته DHCPTIMEOUT

این بسته نشان‌دهنده این است که آی‌پی شما دیگر اعتباری ندارد، در نتیجه عبارت

```
ip released
```

را چاپ کنید.

- دریافت دستور

```
extend lease
```

به این معنی که زمان منقضی شدن را ۱۰ واحد افزایش دهید.

پس از دریافت این دستور، باید بسته‌ای از نوع Request Extend تولید کنید و روی تمام واسطه‌های خود بفرستید.

- دریافت بسته Response Extend

این پیام به این معنی است که با درخواست تمدید شما موافقت شده است و می‌توانید از آی‌پی جدیدی که در این بسته برای تمدید به شما داده شده استفاده کنید. دقت کنید که آی‌پی قبلی شما نامعتبر شده است. پس در خروجی عبارت

```
now my ip is: X
```

که  $X$  آدرس جدید است را چاپ کنید.

- دریافت دستور

```
print ip
```

در این صورت باید تمام IP هایی که در حال حاضر متعلق به این گره هستند را به ترتیب از کوچک به بزرگ، هر کدام را در یک خط جدا چاپ کنید.

## ۳.۴. کارگزار DHCP

شما در نقش کارگزار باید به درخواست‌های کارخواه‌ها پاسخ دهید و آدرس‌هایی که در اختیار دارید را به درستی مدیریت کنید. هر کارگزار یک IP Pool دارد که مجموعه آدرس‌هایی است که در اختیار دارد. هر بار که یک IP را اختصاص می‌دهید باید از Pool خود خارج کنید تا به گره دیگری نسبت ندهید. شرح وظایف این نوع گره‌ها در ادامه آمده است.

- مسیریابی بسته‌ها

در این نوع از گره‌ها در صورتی که بسته‌ای به دست شما رسید، نیازی نیست آن را به اطرافیان خود انتقال دهید. از این جهت این گره‌ها در درخت به نوعی مانند برگ در هر طرف یال خود هستند. همچنین بسته‌ای که به عنوان پاسخ بسته ورودی قرار است ارسال شود، تنها روی همان واسطی که بسته ورودی آمده، ارسال می‌شود.

- دریافت دستور

```
add pool IP/M
```

این دستور به معنای افزودن محدوده جدیدی به مجموعه آی‌پی‌های این کارگزار است. این محدوده به این صورت است که  $IP$  مقدار IP است و  $M$  مقدار mask آن، و شما باید کل آدرس‌های این بازه را (شامل آدرس اول و آخر) به مجموعه خود اضافه کنید.

برای مثال، در صورتی که دستور `add pool 192.168.1.10/30` وارد شد، شما باید آدرس‌های 192.168.1.8، 192.168.1.9، 192.168.1.10، 192.168.1.11 را به مجموعه اضافه کنید.

نکته: تضمین می‌شود که  $M$  در تمام تمرین تست‌ها بین ۲۵ تا ۳۲ باشد.

نکته: تضمین می‌شود هیچ‌گاه دو کارگزار، محدوده مشترکی نداشته باشند. اما برای یک کارگزار ممکن است دو بازه مشترک داده شوند، شما باید اجتماع این بازه‌ها را بگیرید.

- دریافت بسته DHCPDISCOVER

پس از دریافت این بسته، شما باید اولین آی پی موجود خود را (کوچکترین) برای پیشنهاد ارسال کنید. (در قالب یک بستهی DHCP OFFER) و آن را از pool خود خارج کنید. دقت کنید که بسته را تنها روی واسطی می فرستید که بسته را از آن دریافت کرده اید. همچنین عبارت

`offer IP to MAC`

را چاپ کنید، که IP مقدار IP پیشنهادی و MAC آدرس MAC درخواست کننده است.

- دریافت بسته DHCPREQUEST

در صورت دریافت این بسته، اگر این IP متعلق به شما بود، باید بسته ای از نوع DHCPACK در پاسخ ارسال کنید و فرض کنید از این به بعد این IP اختصاص به این MAC آدرس دارد و عبارت

`assign IP to MAC for 10`

را چاپ کنید. که IP آدرس اختصاص داده شده و MAC آدرس مک درخواست کننده است. نکته: از الان تا قبل از ۱۰ واحد زمانی آینده، این قرارداد اعتبار دارد و اگر زمان را ۱۰ واحد به جلو ببریم، دیگر این قرارداد اعتباری ندارد.

- دریافت بسته DHCPACK

در صورتی که این بسته را دریافت کردید، به این معناست که گره مورد نظر، درخواست شما را رد کرده است. در نتیجه باید IP که به او پیشنهاد داده بودید را دوباره به Pool خود بازگردانید و عبارت

`IP back to pool`

را چاپ کنید. که IP مقدار آی پی بازگردانده شده است.

- دریافت بسته DHCPRELEASE

اگر این IP را شما اختصاص داده بودید، حال باید آن را به Pool بازگردانید و فرض کنید به گره ای اختصاص ندارد.

- دریافت دستور

`add time t`

باید زمان کارگزار را  $t$  واحد جلو ببرید. در صورتی که در این زمان، IP ای منقضی شده بود، آن را به Pool بازگردانید و سپس بسته DHCP Timeout را برای اطلاع کارخواه بفرستید. ترتیب ارسال بسته‌ها باید به ترتیب زمان انقضا باشد. اگر دو زمان انقضا باهم برابر بودند، به هر ترتیب دلخواهی می‌توانید آن دو را بفرستید.

#### ● دریافت بسته Request Extend

در پاسخ، همیشه افزایش زمان را تایید می‌کنید؛ اما ابتدا آدرس قبلی را با اضافه کردن به Pool بی اعتبار می‌کنید و پس از آن، کوچکترین آدرسی که در Pool موجود است را (ممکن است دوباره همین آدرس باشد) با زمانی معادل با زمان باقیمانده از آدرس قبلی، به علاوه ۱۰ به کارخواه اختصاص می‌دهید. بسته را ارسال می‌کنید و در خروجی عبارت:

```
assign IP to MAC for t
```

را چاپ می‌کنید. در این عبارت آدرس جدید، مک کارخواه و زمان جمع زده شده را چاپ می‌کنید.

#### ● دریافت دستور

```
print pool
```

باید کل Pool را به ترتیب از کوچک به بزرگ چاپ کنید. دقت کنید که آی‌پی‌هایی که در حال حاضر اختصاص یا OFFER داده‌اید را نباید چاپ کنید.

### ۴.۴. موارد خاص

در این قسمت حالت‌های خاصی از مسئله که ممکن است پیش بیاید و در حالت‌های مسئله مطرح نشده است بررسی می‌شود. نکته بسیار مهم این است که کد شما به هیچ وجه نباید در زمان تست از کار بیافتد، زیرا ممکن است نمره برخی قسمت‌ها را به صورت کامل از دست بدهید. در زیر حالت‌های مهم که در تست‌ها باید رعایت شوند آمده است و کافی است همین حالت‌ها را بررسی کنید:

#### ● دور ریختن و چاپ عبارت

```
invalid packet, dropped
```

برای بسته‌هایی که Data Type آنها جزء موارد گفته شده نیست.

#### ● برای گره‌های کارخواه و کارگزار در صورتی که دستور وارد شده غلط باشد، باید عبارت:

```
invalid command
```

را چاپ کنید. و منتظر دستورات بعدی باشید.

## ۵. پیاده سازی

برای پیاده سازی این تمرین، شما امکان استفاده از دو زبان ++c و java را دارید. پیشنهاد ما استفاده از زبان java است، چرا که مشکلات کار با اشاره گرها را نخواهید داشت. همچنین کمتر درگیر Endianess خواهید شد و تجربه ترم های پیش نشان داده کار با جاوا به مراتب راحت تر است. اما از طرفی برنامه نویسی ++c بسیار جزئی تر است و شما کار با کتابخانه های اصلی و رایج را یاد می گیرید که به مراتب جذاب تر از جاوا است.

نکته: همه ی برنامه های شما در سامانه عامل لینوکس با هسته ی ۳/۱۹ به بالا کامپایل می شوند و شما هم باید کد خود را در سامانه عامل لینوکسی کامپایل نمایید.

نکته: در تمام این تمرین، برای شبیه سازی شبکه و ارسال پیام بین گره ها، شما نیاز به استفاده از سامانه ی پرتو دارید.

### ۱.۵. مشترک

- برای کار با سیستم پرتو، نام کاربری و رمز خود را در پرونده `info.sh` قرار دهید.
- برای کامپایل شدن کد خود، از دستور `make` استفاده کنید. دقت کنید که کد ارسالی شما باید از این طریق کامپایل شود وگرنه شما نمره ای نخواهید گرفت.
- پس از کامپایل، ابتدا به اینترنت متصل شوید. سپس جهت اجرا شدن کد، باید فایل `free.sh` را اجرا کنید تا اطلاعات نقشه قبلی از پرتو شما حذف شود. سپس، با اجرای `new.sh` یک نقشه جدید ایجاد کنید. پس از این می توانید کد کامپایل شده خود را با اجرای `run.sh X` اجرا کنید. که  $X$  شماره گره ای از شبکه است که کد شما قرار است جای آن بنشیند.

### ۲.۵. برنامه نویسی java

- در صورتی که زبان java را برای پیاده سازی انتخاب کردید، پیشنهاد ما استفاده از Eclipse IDE یا IntelliJ است، تا کارتان راحت تر شود. شما تنها حق تغییر فایل های پکیج `ir.sharif.ce.partov.machine` را دارید و فایل های دیگر خود را نیز تنها در این بخش قرار دهید.
- دو فایل `ServerMachine.java` و `ClientMachine.java` به صورت پیش فرض پر شده اند. شما باید این دو فایل را برای هر یک از حالت هایی که گره شما در نقش کارگزار DHCP و کارخواه باشد، پر کنید و منطق خود را پیاده سازی کنید.

### ۳.۵. برنامه نویسی C++

در صورتی که زبان C++ را انتخاب کردید، پیشنهاد ما استفاده از یکی از IDE های رایج مانند (inteliij, eclipse, codeblocks و غیره) است، چرا که ممکن است نیاز به استفاده از کتابخانه‌هایی داشته باشید که تا به حال به آن‌ها برنخورده‌اید. با امکانات این نرم‌افزارها می‌توانید کار خود را راحت‌تر انجام دهید و کتابخانه‌های جدید را راحت مطالعه کنید.

شما باید کد اصلی خود را در دو فایل `server_machine.cpp` و `client_machine.cpp` قرار دهید تا هرگاه کد شما به عنوان یکی از این اعضا اجرا شد، منطق گفته شده به درستی کار کند.

در صورتی که می‌خواهید چند فایل دیگر نیز اضافه کنید، آن‌ها را در پوشه user قرار دهید و مطمئن شوید که کد شما با روش گفته شده کامپایل می‌شود.

### ۴.۵. نقشه نمونه

برای راحتی کار شما، نقشه ساده‌ای جهت تست برنامه‌یتان وجود دارد با نام `DHCP_Simple` که به شکل زیر است. دقت کنید که نقشه مورد آزمون در داوری نمرات با این نقشه متفاوت خواهد بود.

شکل ۴: نقشه‌ی نمونه



## نکات ضروری

- به علت اینکه نمره‌ی تمرین به صورت خودکار داده می‌شود، ساختار پیام‌های مطرح شده باید دقیقاً به صورتی باشد که در مستند توضیح داده شده است.
- نقشه‌ای که برای ارزیابی استفاده می‌شود با نقشه تست که در اختیار شما قرار گرفته متفاوت خواهد بود.
- داوری خودکار به صورت کامل در اختیار شما قرار داده می‌شود و می‌توانید نمره خود را ببینید. اما ملاک ارزیابی نمره‌ای است که کد ارسالی شما روی کارگزار خواهد گرفت. اگر موارد گفته شده را رعایت کرده باشید، نمره شما نباید تغییری داشته باشد.
- به دلیل مشکلات اینترنتی بهتر است داوری را هنگامی که به شبکه‌ی دانشگاه متصل هستید انجام دهید.
- در صورتی که هر مشکل یا پرسشی داشتید که فکر می‌کنید پاسخ آن برای همه مفید خواهد بود، آن را به گروه اینترنتی درس ارسال کنید.
- از فرستادن جواب تمرین به گروه اینترنتی درس خودداری کنید.
- تمام برنامه‌ی شما باید توسط خود شما نوشته شده باشد. فرستادن کل یا قسمتی از برنامه‌تان برای افراد دیگر، یا استفاده از کل یا قسمتی از برنامه‌ی فرد دیگری، حتی با ذکر منبع، تقلب محسوب می‌شود.
- پس از اتمام کارتان لازم است با اجرای دستور `make archive` فایل زیبایی شامل تمام فایل‌هایی که برای اجرا شدن کد شما نیاز است بسازید. (این دستور فایل `info.sh` شما را درون زیپ قرار نمی‌دهد زیرا نیازی به این فایل نیست!) در صورتی که از کلاس‌ها و فایل‌های اضافه شده خودتان استفاده می‌کنید، سعی کنید در پوشه گفته شده باشد. در هر صورت فایل آرشیو شما باید قابلیت کامپایل/اجرا شدن را به روش سیستمی داشته باشد، در غیر اینصورت نمره شما صفر خواهد شد.
- نسخه نهایی تمرین خود را به **وبسایت کوئرا** ارسال نمایید.