

به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

شبکه‌های کامپیوتری

پروژه سوم - DHCP

خرداد ۱۴۰۰

در این پروژه قصد داریم با پروتکل DHCP آشنا شویم و برخی کارکردهای آن را پیاده‌سازی کنیم.

نکات مهم قبل از شروع:

- می‌توانید از هر زبانی برای پیاده‌سازی پروژه استفاده کنید.
- پروژه به صورت تک نفره انجام می‌شود. مشورت با دوستان مشکلی ندارد و حتی توصیه می‌شود اما پروژه باید توسط خودتان پیاده‌سازی شود و تقلب یا کپی کردن از یکدیگر و یا کپی از اینترنت موجب از بین رفتن نمره شما خواهد شد.
- پروژه تحویل حضوری دارد و تسلط کافی به سورس کد برنامه ضروری است و بخشی از نمره به صورت ضریب به تسلط شما وابسته می‌باشد.
- پاسخ خود را در فرمت **CN_Proj3_9731000** آپلود نمایید.

بخش اول) پرسش‌های تئوری:

- ۱) کاربردها، مزایا و معایب پروتکل **DHCP** را شرح دهید.
- ۲) قالب بسته‌های **DHCP** را رسم نموده، کاربرد و وظیفه‌ی هر **Field** را بیان نمایید. (می‌تواند از **RFC** [۲۱۳۱](#) کمک بگیرید)
- ۳) نحوه تبادل پیغام‌ها در پروتکل **DHCP** را با رسم شکل تشریح کنید و هر مرحله را کامل توضیح دهید.
- ۴) **DHCP Client** , **DHCP Server** از چه پورت‌هایی استفاده می‌کنند؟
 - ۴-۱) چرا کلاینت از یک پورت خاص استفاده می‌کند؟
 - ۴-۲) چرا در مرحله دوم، تخصیص آدرس به کلاینت خاتمه نمی‌یابد؟
 - ۴-۳) دریافت تاییدیه از سوی سرور در مرحله آخر چه مفهومی دارد؟
- ۵) به طور مختصر توضیح دهید که **MAC Address** چیست؟

بخش دوم) برنامه‌نویسی و پیاده‌سازی

در این بخش هدف ما شبیه‌سازی کلاینت و سرور در پروتکل DHCP و پیاده‌سازی سرویس تخصیص IP به Host های شبکه است.

DHCP SERVER SETTINGS (OPTIONAL)

Use this section to configure the built-in DHCP Server to assign IP addresses to the computers on your network.

☒ **Enable DHCP Server**

DHCP IP Address Range : to

DHCP Lease Time : (hours)

DHCP RESERVATIONS LIST

Status	Computer Name	MAC Address	IP Address
<div><input type="button" value="Add"/> <input type="button" value="Edit"/> <input type="button" value="Delete"/></div>			

NUMBER OF DYNAMIC DHCP CLIENTS : 3

Computer Name	MAC Address	IP Address	Expire Time	
Galaxy-A5-2017		192.168.1.2	23 hours, 12 minutes, 19 seconds	Reserve
DESKTOP-CRRQQM8		192.168.1.3	23 hours, 59 minutes, 1 seconds	Reserve
DESKTOP-CRRQQM8		192.168.1.4	23 hours, 59 minutes, 22 seconds	Reserve

شکل ۱- تصویری از تنظیمات DHCP Server ای که به صورت built-in در مودم D-Link موجود است.

نکات کلی برای کلاینت و سرور:

- بسته‌های DHCP را در هر ۴ مرحله (Discover, Offer, Request, Ack) باید خودتان بسازید و فیلدهای لازم را مقداردهی کنید. (باتوجه به ساختار بسته‌های DHCP و کاربرد هر فیلد که در بخش تئوری مطالعه کرده‌اید). استفاده از کتابخانه‌های آماده برای اینکار مجاز نمی‌باشد.
- تمامی ارتباط‌های بین کلاینت و سرور باید به صورت UDP باشد.
- اگر پورت‌های استاندارد DHCP Client , Server روی دستگاه شما بسته بود می‌توانید از سایر پورت‌ها استفاده نمایید.

کلاینت:

روند کلی اجرای کلاینت:

کلاینت اجرا می‌شود و Discovery را ارسال می‌کند و منتظر می‌ماند تا Offer از سرور دریافت شود. بعد از دریافت Offer پیام Request را ارسال می‌کند و منتظر می‌ماند تا Ack بیاید. بعد از دریافت Ack باید IP دریافت شده را در ترمینال چاپ نماید.

جزئیات پیاده‌سازی که باید در نظر گرفته شوند:

- در کلاینت باید درخواست‌های DHCPDiscover و DHCPRequest پیاده‌سازی شوند.
- اگر Ack بعد از بازه‌ی زمانی مشخصی (timeout) نیامد، کلاینت باید دوباره با ارسال پیام DISCOVER، فرایند را از سر بگیرد. Timeout را در کلاینت مقداری دلخواه در کد در نظر بگیرید.
- پیام DISCOVER خود یک زمان‌بند دارد و به این صورت عمل می‌کند که اگر از زمان ارسال پیام DISCOVER قبلی مدت زمان مشخصی گذشته باشد، بررسی می‌کند که آیا IP گرفته‌ایم یا نه، اگر نگرفته بودیم و یا گرفته بودیم ولی expire شده بود، دوباره پیام DISCOVER ارسال می‌شود. نحوه‌ی محاسبه زمان انتظار برای این زمان‌بند به شرح زیر می‌باشد: (نکته: این زمان‌بند همیشه در حال کار است).
- ◀ در سمت کلاینت دو ثابت با نام‌های backoff-cutoff و initial-interval با مقدار پیش‌فرض به ترتیب 10 و 120 نگهداری می‌شود که هر دو به ثانیه می‌باشند.
- ◀ پس از ارسال اولین پیام DISCOVER این زمان‌بند شروع به کار می‌کند و به اندازه initial-interval صبر می‌کند و طبق گفته قبلی بعد از این مدت اگر نیاز بود دوباره DISCOVER ارسال می‌کند.
- ◀ بعد از ارسال دومین DISCOVER حال به اندازه initial-interval صبر نمی‌کند بلکه بعد از باز ارسال DISCOVER این بازه با این فرمول محاسبه می‌شود: $P*2 \leq R$ که در آن R یک عدد تصادفی بین 0 تا 1 است و P مخفف Previous interval است یعنی بازه‌ی زمانی که قبلاً صبر کرده است.
- ◀ این افزایش بازه برای مدت انتظار تا حد backoff-cutoff بالا می‌رود و اگر از این بیشتر شد روی همین مقدار تنظیم می‌شود. مثال زیر یک سناریو احتمالی برای درک بهتر است:

- ارسال پیام DISCOVER

- انتظار به اندازه ۱۰ ثانیه

- IP گرفته نشده است - ارسال پیام DISCOVER

- انتظار به اندازه $10 * 2 = 20$ ثانیه (۰.۶ تصادفی تولید شده است)

- IP گرفته نشده است - ارسال پیام DISCOVER

- انتظار به اندازه $20 * 2 = 40$ ثانیه (۰.۸۶ تصادفی تولید شده است)

- و ...

این بازه حداکثر تا ۱۲۰ ثانیه می‌تواند افزایش یابد که مقدار پیش‌فرض برای backoff-cutoff است.

نکته مهم: لطفاً به تمیزی کد در کل پروژه و بخصوص این بخش بیشتر اهمیت بدهید.

سرور:

روند کلی اجرای سرور:

سرور ران می شود و همیشه آماده ی دریافت Discovery است. وقتی Discovery در سرور دریافت می شود ، Offer را ارسال می کند و سپس منتظر می ماند تا Request را دریافت کند. پس از دریافت Request، پیام Ack را ارسال می کند.

جزئیات پیاده سازی که باید در نظر گرفته شوند:

- در سرور باید DHCPOffer و DHCPAck پیاده سازی شوند.
- سرور باید به صورت مولتی ترد باشد یعنی پس از دریافت Discovery برای هر کلاینت جداگانه سایر مراحل را ادامه دهد و پیوسته آماده ی دریافت پیغام های Discovery باشد.
- سرور دارای یک IP Pool است که اختصاص IP از IP های آزاد موجود در این pool انجام می پذیرد. اگر IP ای مورد استفاده ی یک کلاینت است و قبلتر تخصیص داده شده دیگر نباید به کلاینت دیگری تخصیص داده شود. (IP تکراری نباید تخصیص داده شود.)
- سرور باید Lease time را برای هر کلاینت در نظر بگیرد و بعد از اتمام این زمان IP را به Pool بازگرداند و بعد آن IP جزو IP های آزاد باشد.
- همچنین اگر قبلاً یک IP به یک MAC Address تخصیص داده شده بود، در صورتی که از همان MAC Address دوباره درخواست IP آمد IP نباید جدید به آن تعلق گیرد و صرفاً همان IP قدیمی که از lease time آن نگذشته است دوباره ارسال می گردد و lease time آن renew می شود. (البته اگر Lease time آن گذشته بود مشکلی ندارد و می توان IP جدید اختصاص داد).
- سرور باید قابلیت رزرو کردن IP برای MAC Address های خاص را داشته باشد. (یک IP برای هر MAC Address). یعنی به نوعی برای دستگاه های خاص IP استاتیک در نظر می گیریم. وقتی یک IP رزرو می شود دیگر به هیچ کلاینت دیگری داده نمی شود حتی اگر کلاینتی که این IP برایش رزرو شده در آن لحظه آفلاین باشد.
- همچون رزرو کردن، قابلیت بلاک کردن MAC Address های خاص را نیز داریم. بدین صورت که اگر از MAC های خاص درخواست برای IP داشتیم، به درخواست پاسخی نمی دهیم.
- سرور باید برای هر MAC Address ای که به آن IP اختصاص داده شده، اسم دستگاه، زمان باقی مانده تا expire شدن آن IP و خود IP اختصاص داده شده را نگهداری کند. (همچون شکل ۱) و با نوشتن دستور `show_clients` در کنسول، در هر سطر Computer name, Mac Address, IP Address, Expire Time به ترتیب نمایش داده شوند.

وقتی سرور می‌خواهد اجرا شود **باید** مقداری همچون بازه‌ی IP Pool و Lease time و ... از یک فایل configs.json خوانده شوند.

فرمت json درون این فایل بدین شکل می‌باشد:

```
{
  "pool_mode": "range",
  "range": {
    "from": "192.168.1.2",
    "to": "192.168.1.10"
  },
  "subnet": {
    "ip_block": "192.168.1.0",
    "subnet_mask": "255.255.255.224"
  },
  "lease_time": 10,
  "reservation_list": {
    "mac1": "ip1",
    "mac2": "ip2",
    ...
  },
  "black_list": [
    "mac1", "mac2", ...
  ]
}
```

نکات مربوط به فایل json :

- pool_mode می‌تواند یکی از ۲ مقدار range و subnet را داشته باشد.
- وقتی pool_mode روی حالت range است باید محدوده‌ی IP Pool از فیلد range خوانده شود و از خود from تا خود to باشد.
- وقتی pool_mode روی حالت subnet است باید محدوده‌ی IP Pool از فیلد subnet خوانده شده و از ip_block شروع شود و تا جایی که subnet_mask تعیین می‌کند باشد (آدرس شبکه و آدرس برادکست را نباید جزو IP Pool در نظر بگیرید یعنی فرضاً 192.168.1.0 جزو IP Pool نخواهد بود چون آدرس شبکه است و از 192.168.1.1 را در نظر می‌گیریم).
- lease_time را به ثانیه در نظر بگیرید.
- در reservation_list زوج‌های MAC-IP را قرار می‌دهیم.
- در black_list، لیست MAC‌های بلاک شده را قرار می‌دهیم.
- جای mac، باید مقدار واقعی قرار گیرد. مثلاً ff:c1:9a:d6:4d:00

فایل config.json قبل از اینکه برنامه اجرا شود مقداردهی می‌شود و نیازی نیست برنامه‌ی شما در هنگام اجرا قابلیت عوض کردن range یا lease_time یا reservation_list و ... را داشته باشد.

امتیازی:

۱. VirtualBox را روی سیستم میزبان خود نصب کنید.
۲. یک ماشین مجازی Ubuntu Server روی سیستم میزبان خود ایجاد کنید.
۳. پیکربندی این VM را طوری تنظیم کنید که از سیستم میزبان قابل دسترس باشد.
۴. DHCP Server خود را روی VM اجرا کنید و از صحت عملکرد آن با اجرا کردن کلاینت مطمئن شوید.
۵. یک DHCP Server روی سیستم میزبان خود و یکی دیگر روی VM اجرا نمایید و با یک کلاینت درخواست Discovery ارسال نمایید. تنها یکی از دو سرور باید Ack نهایی را بفرستد. تمامی مراحل ارتباط با کلاینت را در سرور خود چاپ کنید و از صحت عملکرد سرورهایتان مطمئن شوید.

شفاف سازی:

- بعد از اینکه IP به کلاینت اختصاص داده شد قرار نیست کلاینت واقعا با استفاده از آن IP به اینترنت وصل شود و این پروژه صرفا یک شییه سازی از عملکرد پروتکل DHCP است.
- وقتی کلاینت قطع می شود در سرور IP همچنان برای همان کلاینت است مگر اینکه lease time آن گذشته باشد. اگر lease time نگذشته باشد، وقتی کلاینت دوباره اجرا شود و درخواست Discovery و ... را ارسال کند، سرور باید همان IP قدیمی را برای کلاینت ارسال کند.