



## بخش اول، تمرین پینگر ICMP

در این تمرین، درک بهتری از پروتکل پیام کنترل اینترنت (ICMP) به دست خواهید آورد. یاد خواهید گرفت که یک برنامه کاربردی Ping را با استفاده از پیام‌های درخواست و پاسخ ICMP پیاده‌سازی کنید.

- Ping یک برنامه کاربردی شبکه کامپیوتری است که برای آزمایش این که آیا یک میزبان خاص در یک شبکه IP قابل دسترسی است یا خیر استفاده می شود.
  - همچنین برای خودآزمایی کارت رابط شبکه کامپیوتر یا به عنوان یک آزمایش تأخیر (Latency) استفاده می شود.
  - این برنامه با ارسال بسته های "echo request" ICMP به میزبان هدف و گوش دادن برای پاسخ های "echo" ICMP reply کار می کند.
  - "echo reply" گاهی اوقات "pong" نامیده می شود.
  - Ping زمان رفت و برگشت (round-trip time) را اندازه گیری می کند، از دست دادن بسته را ثبت می کند، و یک خلاصه آماری از بسته های echo reply دریافت شده (حداقل، حداکثر، و میانگین زمان های رفت و برگشت و در برخی نسخه ها انحراف معیار میانگین) را چاپ می کند.
- وظیفه شما این است که برنامه Ping خود را در پایتون توسعه دهید. برنامه شما از ICMP استفاده خواهد کرد اما، برای ساده نگه داشتن آن، دقیقاً از مشخصات رسمی در RFC 1739 پیروی نخواهد کرد. توجه داشته باشید که شما فقط نیاز به نوشتن سمت کلاینت برنامه دارید، زیرا عملکرد مورد نیاز در سمت سرور در تقریباً تمام سیستم عامل ها تعبیه شده است.
- شما باید برنامه Ping را طوری تکمیل کنید که درخواست های پینگ را به یک میزبان مشخص شده با فاصله تقریباً یک ثانیه ارسال کند. هر پیام شامل یک محموله داده است که شامل یک مهر زمانی (timestamp) است. پس از ارسال هر بسته، برنامه تا یک ثانیه منتظر می ماند تا پاسخی دریافت کند. اگر یک ثانیه بدون پاسخ از سرور بگذرد، کلاینت فرض می کند که یا بسته پینگ یا بسته پونگ در شبکه گم شده است (یا این که سرور از کار افتاده است).

## نکات اضافی

۱. در متد "receiveOnePing"، باید ساختار ICMP\_ECHO\_REPLY را دریافت کنید و اطلاعات مورد نیاز خود مانند checksum، sequence number، time to live (TTL) و غیره را واکشی کنید. قبل از تلاش برای تکمیل متد "receiveOnePing"، متد "sendOnePing" را مطالعه کنید.
۲. لازم نیست نگران checksum باشید، زیرا در کد داده شده است.
۳. این تمرین نیاز به استفاده از سوکت های خام (raw sockets) دارد. در برخی از سیستم عامل ها، ممکن است برای اجرای برنامه Pinger خود نیاز به امتیازات مدیر/روت (administrator/root privileges) داشته باشید.

۴. برای اطلاعات بیشتر در مورد ICMP به انتهای این بخش برنامه نویسی مراجعه کنید.

## تست پینگر

- ابتدا، کلاینت خود را با ارسال بسته ها به localhost، یعنی 127.0.0.1 آزمایش کنید.
- سپس، باید ببینید که برنامه Pinger شما چگونه در سراسر شبکه با پینگ کردن سرورها در قاره های مختلف ارتباط برقرار می کند.

## چه چیزی را باید تحویل دهید

شما باید **starter-code-CA5-P1.py** کد کامل کلاینت و اسکرین شات هایی از خروجی Pinger خود برای چهار میزبان هدف، که هر کدام در یک قاره متفاوت هستند، تحویل دهید. پس هم کد و هم گزارش لازمه نمره کامل هستند.

## تمرینات اختیاری

۱. در حال حاضر، برنامه زمان رفت و برگشت (RTT) را برای هر بسته محاسبه کرده و آن را به صورت جداگانه چاپ می کند. این را طوری اصلاح کنید که با نحوه کار برنامه استاندارد ping مطابقت داشته باشد. شما باید حداقل، حداکثر، و میانگین RTT ها را در پایان تمام پینگ های کلاینت گزارش دهید. علاوه بر این، نرخ از دست دادن بسته (بر حسب درصد) را محاسبه کنید.
۲. برنامه شما فقط می تواند تایم اوت ها (timeouts) را در دریافت پاسخ های ICMP echo تشخیص دهد. برنامه Pinger را طوری اصلاح کنید که کدهای خطای پاسخ ICMP را تجزیه و تحلیل کرده و نتایج خطای مربوطه را به کاربر نمایش دهد. مثال هایی از کدهای خطای پاسخ ICMP عبارتند از: 0: Destination Network Unreachable (شبکه مقصد غیر قابل دسترسی)، 1: Destination Host Unreachable (میزبان مقصد غیر قابل دسترسی).

## پروتکل پیام کنترل اینترنت (ICMP)

### هدر ICMP

هدر ICMP پس از بیت ۱۶۰ هدر IP شروع می شود (مگر این که گزینه های IP استفاده شده باشند).

176-191	168-175	160-167	بیت ها
Checksum (چک سام)	Code (کد)	Type (نوع)	160
Sequence (توالی)	ID (شناسه)		192

• Type (نوع): نوع ICMP.

• Code (کد): زیرنوعی برای نوع ICMP داده شده.

• Checksum (چک سام): داده های بررسی خطا که از هدر ICMP + داده ها محاسبه می شوند، با مقدار 0 برای این فیلد.

• ID (شناسه): یک مقدار ID، که باید در صورت echo reply (پاسخ اکو) بازگردانده شود.

• Sequence (توالی): یک مقدار توالی، که باید در صورت echo reply بازگردانده شود.

### درخواست اکو (Echo Request)

درخواست اکو یک پیام ICMP است که انتظار می رود داده های آن در یک echo reply ("pong") بازگردانده شود. میزبان باید به تمام درخواست های اکو با یک پاسخ اکو حاوی داده های دقیقی که در پیام درخواست دریافت شده است، پاسخ دهد.

• Type (نوع) باید روی 8 تنظیم شود.

• Code (کد) باید روی 0 تنظیم شود.

• Identifier (شناسه) و Sequence Number (شماره توالی) می توانند توسط کلاینت برای تطبیق پاسخ با درخواستی که باعث پاسخ شده است، استفاده شوند.

○ در عمل، اکثر سیستم های لینوکس از یک شناسه منحصر به فرد برای هر فرآیند پینگ استفاده می کنند، و شماره توالی یک عدد افزایشی در آن فرآیند است.

○ ویندوز از یک شناسه ثابت استفاده می کند که بین نسخه های ویندوز متفاوت است و یک شماره توالی که فقط در زمان راه اندازی (boot time) بازنشانی می شود.

- داده های دریافت شده توسط درخواست اکو باید به طور کامل در پاسخ اکو گنجانده شوند.

## پاسخ اکو (Echo Reply)

پاسخ اکو یک پیام ICMP است که در پاسخ به یک درخواست اکو تولید می شود و برای همه میزبان ها و روتر ها اجباری است.

- Type (نوع) و Code (کد) باید روی 0 تنظیم شوند.

- شناسه و شماره توالی می توانند توسط کلاینت برای تعیین این که کدام درخواست های اکو با پاسخ های اکو مرتبط هستند، استفاده شوند.

- داده های دریافت شده در درخواست اکو باید به طور کامل در پاسخ اکو گنجانده شوند.

## بخش دوم، تمرین اختیاری با نمره اضافی

در این تمرین، شما یاد خواهید گرفت که چگونه یک برنامه Traceroute (مسیریابی) را با استفاده از پیام های درخواست و پاسخ ICMP پیاده سازی کنید.

پیش نیاز:

دانشجویان قویاً تشویق می شوند که قبل از انجام تمرین ICMP Traceroute، ابتدا تمرین ICMP Ping را انجام دهند، زیرا این آزمایش با همان رویکرد انجام می شود. محاسبه چک سام (Checksum) و ساخت هدر (Header) در این تمرین پوشش داده نمی شود؛ برای این منظور به تمرین ICMP Ping مراجعه کنید. نام گذاری اکثر متغیرها و سوکت ها نیز مشابه است.

تئوری و نحوه عملکرد:

تریس روت (Traceroute) یک ابزار تشخیص شبکه کامپیوتری است که به کاربر اجازه می دهد مسیر را از میزبانی که برنامه Traceroute را اجرا می کند تا هر میزبان دیگری در جهان ردیابی کند. Traceroute با پیام های ICMP پیاده سازی می شود. نحوه کار آن به این صورت است که پیام های "ICMP echo" (نوع ۸) را با مقدار افزایشی در فیلد

“زمان بقا” (TTL) به همان مقصد ارسال می‌کند. روترهای موجود در مسیر Traceroute، هنگامی که مقدار فیلد TTL به صفر می‌رسد، پیام “ICMP Time Exceeded” (نوع ۱۱) را باز می‌گردانند. زمانی که مقصد نهایی پیام ICMP echo request را دریافت کند، یک پیام “ICMP reply” (نوع ۰) ارسال می‌کند. آدرس‌های IP روترهایی که پاسخ ارسال می‌کنند، از بسته‌های دریافتی قابل استخراج است. زمان رفت و برگشت (RTT) بین میزبان فرستنده و یک روتر، با تنظیم یک تایمر در میزبان فرستنده تعیین می‌شود.

### وظیفه شما:

وظیفه شما توسعه برنامه Traceroute خودتان در پایتون (Python) با استفاده از ICMP است. برنامه شما از ICMP استفاده خواهد کرد، اما برای ساده‌سازی، دقیقاً از مشخصات رسمی RFC 1739 پیروی نخواهد کرد. در نتیجه کامل کردن کد starter-code-CA5-P2.py و گزارش آن از شما خواسته شده است.

کد:

در پوشه (Starter Code) برای کلاینت آورده شده است. شما باید این کد ناقص را تکمیل کنید.

□ مکان‌هایی که نیاز به نوشتن کد دارند با #Fill in start و #Fill in end مشخص شده‌اند.

□ هر مکان ممکن است به یک یا چند خط کد نیاز داشته باشد.

### نکات اضافی:

۱. نیازی نیست نگران محاسبه Checksum باشید، زیرا کد آن قبلاً داده شده است.
۲. این تمرین نیاز به استفاده از سوکت‌های خام (Raw Sockets) دارد. در برخی سیستم‌عامل‌ها، ممکن است برای اجرای برنامه Traceroute نیاز به دسترسی ادمین/روت (Administrator/Root) داشته باشید.
۳. برای اطلاعات بیشتر در مورد ICMP، به [پایان بخش برنامه‌نویسی ICMP Ping](#) مراجعه کنید.
۴. این برنامه برای وبسایت‌هایی که ترافیک ICMP را مسدود می‌کنند کار نخواهد کرد.
۵. ممکن است لازم باشد فایروال یا آنتی‌ویروس خود را خاموش کنید تا اجازه ارسال و دریافت پیام‌ها داده شود.