تمرین دوم درس شبکه‌های کامپیوتری

پیاده‌سازی ارتباط TCP کلاینت-سرور با تحلیل ترافیک درWireshark

هدف:

- آشنایی با مفاهیم عملی ارتباطاتTCP
- کار با کتابخانه‌های سوکت در Python یا زبان دلخواه
- تحلیل بسته‌های TCP در) Wireshark مانند سه‌مرحله‌ای اتصالTCP ، ارسال داده و خاتمه ارتباط(

شرح تمرین:

۱. یک برنامه ساده TCP به صورت **Client-Server**بنویسید:
   - **سرور** روی یک پورت مشخص منتظر اتصال کلاینت باشد.
   - **کلاینت** پس از اتصال، یک پیام ساده (مانند یک خط متن یا عدد) برای سرور ارسال کند.
   - سرور پیام دریافتی را روی صفحه چاپ کرده و یک پاسخ برای کلاینت ارسال کند.
   - کلاینت پاسخ را چاپ کرده و اتصال خاتمه یابد.
۲. برنامه باید قابلیت اجرای روی ویندوز یا لینوکس را داشته باشد.
۳. از ابزار **Wireshark**برای ضبط ترافیک TCP در حین اجرای برنامه استفاده کنید.

تحلیل در:Wireshark

درWireshark ، بسته‌های TCP مربوط به اجرای برنامه خود را پیدا کرده و موارد زیر را بررسی و در گزارش بیاورید:

- بسته‌های مربوط به **سه‌مرحله‌ای اتصال**(Three-way Handshake: SYN, SYN-ACK, ACK) **TCP**
- بسته‌های مربوط به ارسال پیام از کلاینت به سرور و پاسخ از سرور
- بسته‌های مربوط به خاتمه ارتباط(FIN, ACK)
- شماره پورت‌های مبدا و مقصد
- شماره ترتیب (Sequence Number) و شماره تأیید(Acknowledgment Number)
- RTT (Round Trip Time)بین ارسال و دریافت داده

موارد قابل تحویل:

۱. **کد برنامه** Client)و (Server با توضیح خط به خط
۲. **ویدئو ضبط‌شده** شامل موارد زیر:
   - اجرای برنامه و نمایش ارتباط موفق
   - استفاده از Wireshark برای مشاهده بسته‌ها
   - توضیح خط به خط کد
   - تحلیل بسته‌های TCP در) Wireshark با اشاره به پورت‌ها، شماره ترتیب، و(RTT

نکات تکمیلی:

- برای اجرای راحت‌تر، می‌توانید از (localhost) `127.0.0.1`استفاده کنید یا ارتباط دو سیستم مجازی ایجاد کنید. (مجازی سازی نمره اضافه دارد)
- از شماره پورت‌های بالای ۱۰۲۴ برای سرور استفاده کنید (مثلاً ۵۰۰۰ یا ۸۸۸۸).