

Zadanie 3: Tunel UDP przez TCP SKJ (2018)

Wstęp

Zadanie polega na napisaniu pary aplikacji: agenta i przekaźnika, implementujących funkcjonalność tunelu przesyłającego dane z protokołu UDP z wykorzystaniem protokołu TCP. Aplikacja przekaźnika jest uruchamiana na komputerze zdalnym gdzieś w sieci. Aplikacja agenta jest uruchamiana na komputerze użytkownika. Po uruchomieniu agenta nawiązuje on połączenie TCP z wyspecyfikowanym przekaźnikiem, którego następnie konfiguruje, po czym otwiera w swoim systemie zadane porty UDP i nasłuchuje na nich. Jeśli odczyta na nich dowolny pakiet, zapamiętuje parametry klienta, który go przysłał, a następnie dane z niego pochodzące przekazuje do przekaźnika, który ma za zadanie przesłać te dane za pomocą protokołu UDP do odbiorcy (zgodnie z początkową konfiguracją) na określony port. Przekaźnik, jeśli otrzyma dane zwrotne od komputera, do którego przekazuje dane, jest zobowiązany przekazać te dane do agenta, który następnie odsyła te dane do klienta, który zainicjował połączenie z tym konkretnym portem. W ten sposób uzyskujemy możliwość połączenia się za pomocą protokołu UDP w przypadku, gdy (z jakichś przyczyn) połączenie bezpośrednie po UDP jest niemożliwe, ale możliwa jest komunikacja po TCP z komputerem zdalnym, na którym pracuje proces przekaźnika.

Sposób działania aplikacji

Aplikacja składa się z dwóch procesów: agenta i przekaźnika.

Przekaźnik

Proces ten jest uruchamiany w sieci na komputerze, do którego można się dostać z komputera użytkownika za pomocą protokołu TCP. Podczas uruchomienia wymaga podania jednego parametru, którym jest numer portu TCP do komunikacji z agentem (numer ten może być na sztywno zaszyty w jego kodzie, w takim przypadku nie ma konieczności podawania go). Dalsza jego konfiguracja jest zdalna. Proces ten jest “delegatem” komputera użytkownika służącym do efektywnego wysyłania pakietów UDP z miejsca, z którego mają one szansę na dotarcie do celu (np. w przypadku, gdy lokalna sieć nie pozwala na wysłanie danych UDP na określone porty).

- Po uruchomieniu, proces przekaźnika otwiera **konkretny** port TCP do nasłuchu. Numer tego portu musi być znany agentom (jeśli autor chce, może przekazywać ten numer jako parametr, ale ten sam parametr musi być później przekazywany agentowi). Następnie przekaźnik oczekuje na połączenie agenta na tym porcie TCP.
- Po nawiązaniu połączenia, przekaźnik jest konfigurowany przez agenta. Na konfigurację składa się adres komputera, **do którego** należy przekazywać dalej dane i **od którego** należy odbierać odpowiedzi. Jeśli jest to wymagane przez protokół, konfiguracja może obejmować inne elementy. Dodatkowo przekaźnik otwiera port UDP (o numerze losowym), który będzie służył do przekazywania dalej danych lub odbierania odpowiedzi.
- Po konfiguracji, przekaźnik oczekuje na komunikaty od agenta. Taki komunikat może zawierać:
 1. Komendę rozłączenia i zakończenia pracy. Po jej otrzymaniu, przekaźnik kończy swoją pracę na rzecz danego agenta, rozłącza się z nim oraz przechodzi ponownie w tryb oczekiwania na połączenie z kolejnym agentem.
 2. Dane wraz z numerem portu UDP, na który należy przekazać te dane. Po ich otrzymaniu, przekaźnik generuje komunikat UDP zawierający te dane oraz wysyła go do komputera zdalnego zgodnie ze swoją konfiguracją (patrz wyżej).
- Dodatkowo przekaźnik oczekuje na wszelkie dane na otworzonym przez siebie porcie UDP. Należy oczekiwać wyłącznie na komunikaty od komputera zgodnego z konfiguracją przekaźnika. Po otrzymaniu ich przekazuje je wraz z numerem portu UDP z którego te dane przyszły, do agenta (przez kanał TCP).

Agent

Proces agenta jest uruchamiany na komputerze użytkownika. Stanowi on pośrednik, który będzie odbierał dane z procesu użytkownika, następnie będzie je przysyłał do zdalnego procesu przekaźnika, który przekaże te dane dalej.

- Podczas uruchomienia, proces wymaga podania szeregu parametrów:
 1. Adres IP lub nazwa komputera, na którym pracuje proces przekaźnika.
 2. Numer portu TCP, na którym pracuje przekaźnik (jeśli ta wartość jest na stałe ustalona i zaszyta w kodzie, parametr ten nie jest wymagany).
 3. Adres IP lub nazwa komputera, do którego mają być przekazywane przez przekaźnik dane UDP.
 4. Lista co najmniej jednego portu UDP, na którym agent nasłuchuje na dane od procesu na komputerze.

- Po uruchomieniu, agent łączy się za pomocą protokołu TCP z przełącznikiem pracującym na zadanym adresie i porcie. Po nawiązaniu połączenia przekazuje mu konfigurację określającą adres IP lub nazwę komputera zdalnego, do którego należy przekazywać dane UDP (plus ewentualne inne dane wymagane przez protokół).
- Po zakończeniu konfiguracji przełącznika, agent otwiera porty UDP o numerach zadanych parametrami i zaczyna na nich oczekiwać na dane. Zakładamy, że w z każdego portu UDP danego agenta będzie korzystać tylko jeden proces (nazwany dalej procesem klienta). Ustalenie procesu klienta dla danego portu UDP następuje w momencie odebrania pierwszego komunikatu przez ten port.
- Każdy odebrany od procesu klienta pakiet UDP należy przekazać dalej do przełącznika przez nawiązane wcześniej połączenie TCP. Taka przesyłka musi zawierać numer portu, na którym te dane zostały przechwycone (numer ten jest tożsamy z numerem portu, na który dalej przełącznik musi te dane przekazać). Odbieranie pakietów UDP i przekazywanie ich dalej jest czynnością cykliczną, którą należy powtarzać w kółko.
- Jednocześnie agent nasłuchuje na ustanowionym kanale TCP na dane od przełącznika. Każdy odebrany komunikat musi zawierać numer portu, z którego został wysłany przez zdalny komputer do procesu przełącznika (numer portu nadawcy). Komunikat ten należy wysłać przez port UDP o takim samym numerze jak przekazany numer portu nadawcy do procesu klienta, który się przez ten port komunikuje.
- Użytkownik może w dowolnym momencie wpisać z klawiatury komunikat zakończenia pracy agenta. W takim wypadku najpierw wysyła on do przełącznika komunikat zakończenia połączenia, a następnie zamyka wszystkie porty i sam kończy pracę.

Wymagania i sposób oceny

W celu realizacji zadania należy zaprojektować i zaimplementować własny protokół komunikacyjny, umożliwiający realizację opisaną powyżej funkcjonalności (komunikację między agentem a przełącznikiem).

1. Poprawny i pełny projekt wart jest **8 punktów**. Można również uzyskać **dodatkowy 1 punkt** za funkcjonalność wykraczającą poza podstawowy zakres zadania, wyspecyfikowaną poniżej. Za zrealizowanie każdej z poniższych funkcjonalności można otrzymać punkty do podanej wartości:
 - **2 punkty** za szczegółowy opis zaprojektowanego i zaimplementowanego protokołu transmisji między agentem a przełącznikiem.
 - **2 punkty** za uzyskanie funkcjonalności tunelu dla jednego portu UDP działającego na zasadzie pytanie↔odpowiedź (czyli po odebraniu pakietu UDP od procesu klienckiego i przekazaniu go do komputera zdalnego oczekujemy na odpowiedź, którą przekazujemy do klienta, po czym całość powtarzamy).

- **2 punkty** za dodanie funkcjonalności asynchronicznej pracy tunelu z jednym przekazywanym portem, co oznacza możliwość jednoczesnej transmisji dowolnej liczby pakietów w jedną i drugą stronę.
- **2 punkty** za dodanie funkcjonalności jednoczesnego asynchronicznego przekazywania pakietów z wielu portów przez jeden tunel (pełna funkcjonalność).
- Za rozszerzenie możliwości pracy przekaźnika w taki sposób, aby mógł on obsługiwać jednocześnie wielu agentów można uzyskać **dodatkowy 1 punkt**.

Należy założyć, że w danej chwili tylko jeden agent może wykonywać komunikację z konkretnym hostem docelowym i konkretnym jego portem (pozwala to uniknąć konfliktów z odpowiedziami w przypadku, gdyby dwóch agentów chciało wysyłać dane do tego samego adresata). Założenie to musi być weryfikowane podczas konfiguracji przez agenta po podłączeniu (np. przez przesłanie przez agenta listy numerów obsługiwanych portów i porównanie ich z listą portów już zarezerwowanych na przekaźniku).

2. Aplikację piszemy w języku Java zgodnie ze standardem Java 8 (JDK 1.8). Do komunikacji przez sieć można wykorzystać jedynie podstawowe klasy do komunikacji z wykorzystaniem protokołu UDP.
3. Projekty powinny zostać zapisane do odpowiednich katalogów w systemie EDUX w nieprzekraczalnym terminie 27.01.2019 (termin może zostać zmieniony przez prowadzącego grupę).
4. Spakowany plik projektu powinien obejmować:
 - Plik Dokumentacja_(nr.indeksu)_Zad3.pdf, opisujący, co zostało zrealizowane, co się nie udało, jak zainstalować, gdzie ewentualnie są błędy, których nie udało się poprawić. W szczególności, plik musi zawierać szczegółowy opis zaprojektowanego i zaimplementowanego protokołu (brak opisu protokołu lub jego fragmentaryczność może spowodować znaczące obniżenie oceny rozwiązania zadania).
 - Pliki źródłowe (dla JDK 1.8) (włącznie z wszelkimi bibliotekami nie należącymi do standardowej instalacji Javy, których autor użył) – aplikacja musi dać się bez problemu skompilować na komputerach w laboratorium w PJA.

UWAGA: PLIK Z DOKUMENTACJĄ JEST WARUNKIEM KONIECZNYM PRZYJĘCIA PROJEKTU DO OCENY.

5. Prowadzący oceniać będą w pierwszym rzędzie poprawność działania programu i zgodność ze specyfikacją, ale na ocenę wpływać będzie także zgodność wytworzonego oprogramowania z zasadami inżynierii oprogramowania i jakością implementacji.

6. JEŚLI NIE WYSZCZEGÓLNIŁO INACZEJ, WSZYSTKIE NIEJASNOŚCI NALEŻY PRZEDYSKUTOWAĆ Z PROWADZĄCYM ZAJĘCIA POD GROŻBĄ NIEZALICZENIA PROGRAMU W PRZYPADKU ICH NIEWŁAŚCIWEJ INTERPRETACJI.