TIN - Z6

Projekt wstępny

Autorzy:

Gralak Filip

Kubiszewski Szymon

Libera Piotr

Satała Piotr

Treść zadania:

Napisać program obsługujący prosty kanał P2P w oparciu o protokół BitTorrent:

- Zasób to plik identyfikowany pewną nazwą, za takie same zasoby uważa się zasoby o takich samych nazwach.
- Początkowo dany zasób znajduje się w jednym węźle sieci, następnie może być propagowany do innych węzłów w ramach zainicjowanego przez użytkownika ręcznie transferu – raz pobrany zasób zostaje zachowany jako kopia
- Tak więc, po pewnym czasie działania systemu ten sam zasób może znajdować się w kilku węzłach sieci (na kilku maszynach).
- System ma informować o posiadanych lokalnie (tj. w danym węźle) zasobach i umożliwiać ich pobranie.
- Program powinien umożliwiać współbieżne:
    - wprowadzanie przez użytkownika (poprzez interfejs tekstowy) nowych zasobów oraz poleceń pobrania nazwanego zasobu ze zdalnego węzła
    - pobieranie zasobów (także kilku jednocześnie)
    - rozgłaszanie informacji o posiadanych lokalnie zasobach
- W przypadku pobierania zdalnego zasobu system sam (nie użytkownik) decyduje skąd zostanie on pobrany.
- Powinno być możliwe pobranie zasobu z kilku węzłów na raz (tj. “w kawałkach”).
- Zasób pobrany do lokalnego węzła jest kopią oryginału, zawsze występuje jeden oryginał i potencjalnie wiele kopii. Można założyć, że oryginał nie zmienia się nigdy. Istnienie kopii jest rozgłaszane tak samo jak oryginału. Oryginał nigdy nie zmienia swojej lokalizacji.
- Właściciel oryginału może go unieważnić wysyłając odpowiedni komunikat rozgłaszany.
- Szczegółowe zachowanie się posiadaczy lokalnych kopii w przypadku unieważnienia oryginału pozostaje do decyzji implementacyjnej i powinno być szczegółowo przeanalizowane i opisane w projekcie wstępnym.
- Nie jest wymagane implementowanie szyfrowania.
- Interfejs użytkownika nie musi być szczegółowy i mocno rozbudowany, powinien być prosty i przejrzysty, najważniejszy w implementacji jest protokół.

Założenia:

- Węzeł przy rozpoczynaniu pobierania decyduje po jakiej części i od kogo pobiera dany zasób
- Jeśli właściciel oryginału go unieważni to blokuje on możliwość dalszej propagacji zasobu do innych węzłów, czyli serwer usuwa informacje o posiadanych kopiach oryginału w sieci. Węzły, które posiadają kopie lokalnie nie muszą ich usuwać oraz mogą bez przeszkód dalej z nich korzystać
- W przypadku unieważnienia zasobu podczas jego pobierania, zezwala się na dokończenie procesu i pobranie go w całości
- Po unieważnieniu pliku, dowolny węzeł może dodać plik do sieci o takiej samej nazwie
- Jeśli po próbie pobrania okaże się że seeder nie posiada danego zasobu, część która miała być od niego pobrana zostaje losowo rozdzielona pozostałym seedom, o ile to możliwe
- Udostępnianie zasobu będzie możliwe tylko dla plików znajdujących się w ustalonym katalogu roboczym
- Pobierane zasoby będą zapisywane w ustalonym katalogu roboczym
- Serwer posiada informacje o wszystkich zasobach znajdujących się w sieci oraz węzłach, w których poszczególne zasoby się znajdują
- Wielkość bloku (chunku): 32 kB
- Niemożliwe jest dodanie dwóch plików o tej samej nazwie
- Węzeł pobiera losowe części pliku od węzłów go posiadających
- Serwer co pewien czas aktualizuje informacje o podłączonych węzłach i plikach w sieci

Opis funkcjonalny:

1. Udostępnianie zasobu
    - Użytkownik chce udostępnić nowy zasób do sieci
    - Jeżeli zasób znajduje się w ustalonym katalogu roboczym to zostaje wysłane polecenie do serwera
    - Serwer sprawdza czy zasób o podanej nazwie nie jest już dostępny w sieci
    - Serwer rejestruje nowy zasób oraz jego posiadacza
2. Pobieranie zasobu
    - Użytkownik wybiera zasób, który chciałby pobrać
    - Klient wysyła do serwera żądanie wysłania listy węzłów posiadających dany zasób
    - Węzeł decyduje od kogo pobierze poszczególne fragmenty (chunki) zasobu i nawiązuje wymagane połączenia
    - Po ukończeniu pobierania aktualizowana zostaje lista węzłów posiadających pobrany zasób
3. Unieważnianie
    - Użytkownik chce unieważnić oryginał pliku
    - Węzeł wysyła do serwera żądanie unieważnienia pliku
    - Serwer sprawdza czy plik istnieje, oraz czy dany węzeł jest jego właścicielem
    - Serwer usuwa informacje o danym pliku, oraz o pobranych kopiach
4. Udostępnianie listy zasobów
    - Użytkownik wysyła do serwera żądanie
    - Serwer wysyła do użytkownika listę plików znajdujących się w sieci

Opis i analiza protokołów komunikacyjnych:

Cała komunikacja odbywać się będzie wyłącznie za pomocą protokołu TCP w celu zachowania prostoty. Wstępna konwencja kodowania komunikatów:

- 1a0 - żądania i komunikaty między serwerem (trackerem) a węzłem
- 1a1 - odpowiedzi drugiej strony na wiadomości 1a0
- 2a0 - żądania i komunikaty między węzłami
- 2a1 - odpowiedzi na wiadomości 2a0
- 2a2 - komunikaty o błędach

Komunikaty serwer-węzeł

kod	kiedy wysyłany	zawartość wiadomości	zachowanie po otrzymaniu
100	węzeł dołącza do sieci	kod, identyfikator węzła	tracker aktualizuje informacje o stanie sieci
110	węzeł odłącza się od sieci	kod, identyfikator węzła	tracker aktualizuje informacje o stanie sieci
120	użytkownik (węzeł) prosi o przesłanie listy plików w sieci	kod	wysłanie wiadomości zawierającej listę plików
121	odpowiedź trackera na otrzymanie wiadomości o kodzie 120	kod, lista plików	wyświetlenie zawartości użytkownikowi węzła
130	użytkownik chce pobrać dany plik, dlatego węzeł prosi o listę węzłów go posiadających	kod	wysłanie wiadomości zawierającej listę węzłów posiadających plik
131	odpowiedź trackera na otrzymanie wiadomości o kodzie 130	kod, lista węzłów	węzeł rozpoczyna próbę pobrania pliku od węzłów w liście

140	serwer chce zaktualizować stan plików w sieci	kod	wysłanie wiadomości zawierającej listę plików
141	odpowiedź węzła na otrzymanie wiadomości o kodzie 140, węzeł informuje tracker o zmianie stanu posiadanych plików	kod, lista plików	tracker aktualizuje informacje o stanie sieci
150	węzeł informuje tracker o pojawieniu się nowego pliku możliwego do pobrania w sieci	kod, nazwa pliku	tracker sprawdza, czy możliwe jest wprowadzenie nowego pliku o takiej nazwie (np. niedopuszczalne takie same nazwy) i odpowiada węzłowi
151	odpowiedź trackera na wiadomość o kodzie 150	kod, wartość zwrócona przy próbie dodania pliku	wyświetlenie użytkownikowi efektu akcji
160	węzeł informuje tracker że od teraz posiada kopię pliku	kod, nazwa pliku	tracker aktualizuje informacje o stanie sieci
170	użytkownik decyduje o wycofaniu swojego pliku z sieci	kod, nazwa pliku	tracker usuwa wszystkie informacje o danym pliku, tj. listę węzłów go posiadających (o ile węzeł żądający był właścicielem)

Komunikaty węzeł-węzeł

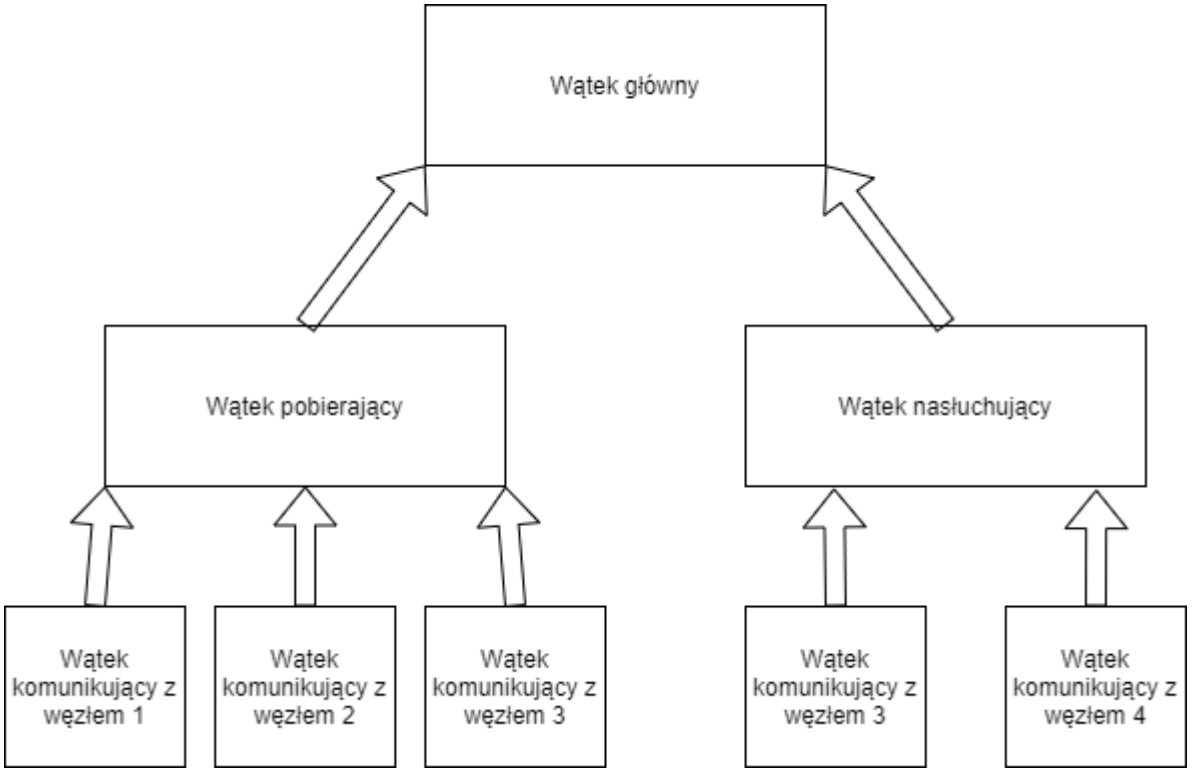
kod	kiedy wysyłany	zawartość wiadomości	zachowanie po otrzymaniu
200	próba nawiązania komunikacji między węzłami	kod	próba nawiązania komunikacji
210	węzeł żąda od drugiego chunk pliku	kod, nazwa pliku, numer bloku	wysłanie żądanego bloku
211	odpowiedź na żądanie chunku pliku	kod, chunk	zapisanie części pliku
212	błąd - węzeł nie posiada danego pliku	kod, kod błędu	zerwanie połączenia przez pobierającego, próba rozdzielenia chunków do pobrania między innymi hostami
220	zakończenie komunikacji	kod	zakończenie komunikacji

Podział na moduły oraz ich komunikacja:

Wątki które muszą się ze sobą komunikować będą robić to za pomocą współdzielonych struktur danych, do których dostęp będzie chroniony za pomocą mutexów.

Klient:

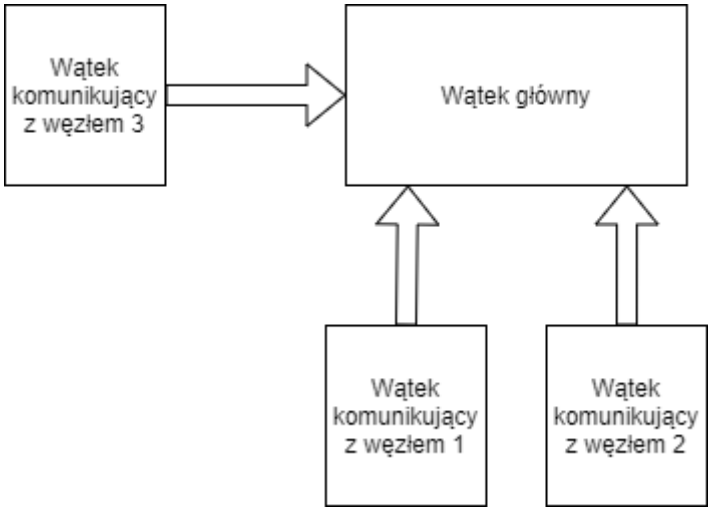
- wątek główny - wątek komunikujący się z użytkownikiem i tworzący wątki nasłuchujący i pobierający
- wątek pobierający - wątek odpowiedzialny za zarządzanie pobieraniem. Decyduje które części plików pobrać od którego węzła, tworzy wątki odpowiedzialne za pobranie wyznaczonych chunków od danego węzła (maksymalnie 1 wątek na węzeł).
- wątek nasłuchujący - nasłuchuje próśb o wysłanie danych plików, tworzy wątek wysyłający dla każdego pobierającego.
- wątek komunikujący - odpowiada za komunikację z danym węzłem i w zależności od tworzącego go rodzica wysyła lub pobiera pliki od wątku pod odpowiadającym węzłem.



Proponowana hierarchia wątków węzła. Na powyższym rysunku węzeł jednocześnie wysyła i pobiera pliki od węzła 3.

Serwer (tracker):

- wątek główny - wątek nasłuchujący komunikaty od innych węzłów i oddelegowujący zadania do wątków pomocniczych (komunikujących)
- wątek komunikujący - odpowiada za komunikację z węzłem



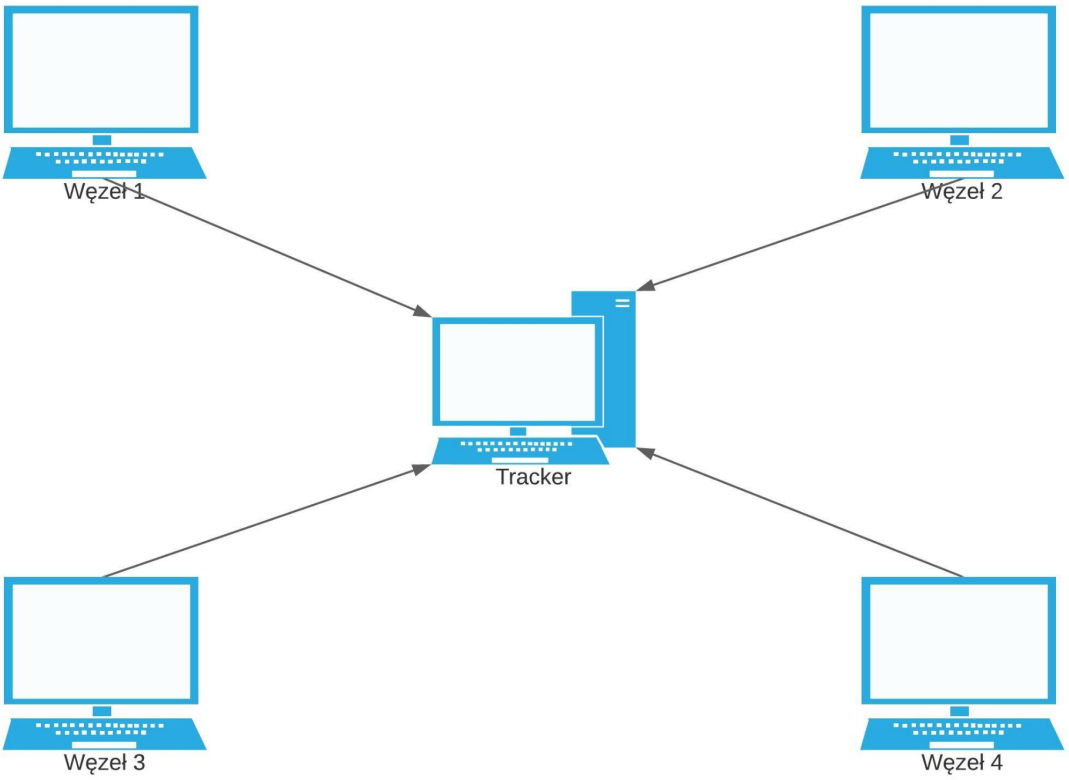
Proponowana hierarchia wątków serwera

Zarys implementacji:

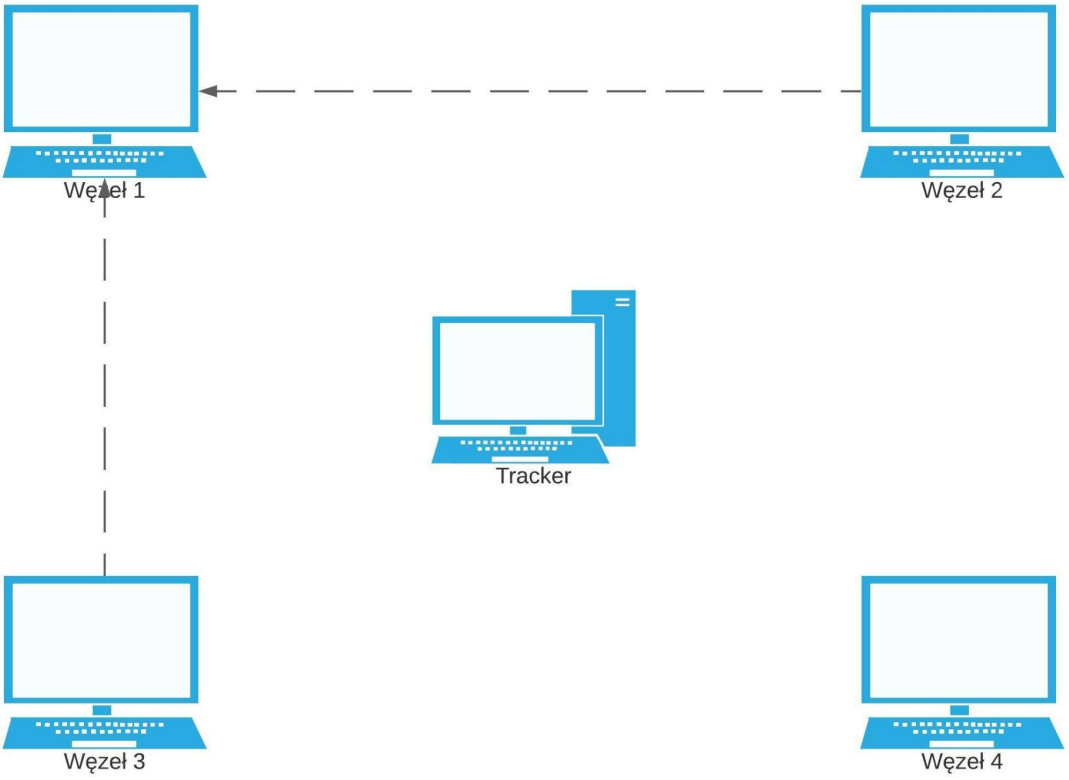
Dla projektu zostaną przeprowadzone testy jednostkowe sprawdzające czy jego poszczególne fragmenty działają prawidłowo. Bardziej ogólne testy zostaną przeprowadzone za pomocą programu GNS3, w którym zostanie zbudowana prosta sieć lokalna. Następnie na jednym z węzłów tej sieci zostanie uruchomiony serwer, a na pozostałych klienci. Wtedy za pomocą manualnych testów będzie można sprawdzić poprawność działania protokołu.

Projekt realizowany będzie w języku C/C++. Do testów jednostkowych wykorzystana zostanie biblioteka Gtest.

Projekt będzie przechowywany w repozytorium github.



Schemat połączeń węzłów z trackerem



Schemat połączeń podczas pobierania pliku przez węzeł pierwszy