

SIEĆ KOMPUTEROWA - DIY
CZYLI BĄDŹ ADMINEM WE WŁASNEJ
FIRMIE

KASIA JANOWA, MICHAŁ PIEKARZ

Spis treści

1	Wstęp	2
1.1	Po co taki projekt?	2
1.2	Naukowe określenie problemu	2
2	Dokumentacja użytkowa	3
2.1	Czego użytkownik powinien oczekiwać?	3
2.2	Tworzenie wykresu	3
2.3	Jak uruchomić program?	3
2.4	Podsumowanie	3
3	Dokumentacja implementacyjna	4
3.1	Algorymiczna idea rozwiązania	4
3.2	Utrudnienie i rozwiązanie tego problemu	4
3.3	Podstawowe dane na temat specyfikacji projektu	4
3.4	Dlaczego golang?	5
3.5	Dlaczego Google charts?	5
3.6	Szczegóły rozwiązania: dodawanie nowego komputera	5
3.7	Szczegóły rozwiązania: pierwotna idea na tworzenie wykresu	5
3.8	Szczegóły rozwiązania: problem z tworzeniem wykresu i jego rozwiązanie	5
3.9	Wybrane protokoły	6
3.10	Protokół SIP	6
3.11	Protokół STP	6
3.12	Krótkie uzasadnienie	7
4	Dokumentacja techniczna	8
4.1	Teoria	8
4.2	Zadania	8
4.3	Rozwiązania	8
5	Podział pracy	9
5.1	Teoria	9
5.2	Zadania	9
5.3	Rozwiązania	9
6	Podsumowanie	10
6.1	Dowcip	10

1 Wstęp

Czyli o co chodzi w naszym projekcie?

1.1 Po co taki projekt?

W dzisiejszych czasach młoda firma, która chce prężnie się rozwijać, często potrzebuje sieć komputerową na własne, wewnętrzne potrzeby. Postanowiliśmy stworzyć aplikację, która pomoże rozwiązać ten problem. Dzięki naszej aplikacji Użytkownik, który dokładnie i ze zrozumieniem przeczyta instrukcję, może samodzielnie zbudować prostą, ale niezwykle wydajną sieć komputerową na potrzeby swojej firmy.

Ma to dla Użytkownika duże znaczenie - początkujące, młode firmy niekoniecznie mogą pozwolić sobie na zatrudnienie specjalisty.

Dla ułatwienia Użytkownikowi pracy i zrozumienia działania aplikacji, wyposażyliśmy ją w dodatkową funkcjonalność - tworzenie wykresu sieci komputerowej. Dzięki niej Użytkownik pozna strukturę sieci, stanie się ona również niezwykle pomocą w razie awarii sieci.

1.2 Naukowe określenie problemu

Tworzymy sieć komputerową o strukturze drzewowej, dla każdej maszyny istnieje z góry określona maksymalna pojemność - ilość maszyn, z którymi może być połączona (zakładamy, że jest to liczba większa od 2). Chcemy zachować najmniejszą możliwą średnicę sieci, tj. chcemy, aby odległość między najbardziej oddalonymi wierzchołkami była jak najmniejsza. Dochodzi nowy komputer, który musimy dodać on-line. Co robimy?

2 Dokumentacja użytkowa

Czyli co użytkownik powinien wiedzieć

2.1 Czego użytkownik powinien oczekiwać?

Prostej aplikacji, która pozwoli mu utworzyć własną sieć komputerową. Użytkownik powinien przygotować się na używanie linii komend. Aplikacja, zainstalowana na danym komputerze, przyłączy komputer do sieci.

2.2 Tworzenie wykresu

2.3 Jak uruchomić program?

2.4 Podsumowanie

Jesteśmy pewni, iż dzięki powyższej instrukcji nawet Użytkownicy niezwiązani z branżą komputerową poradzą sobie z utworzeniem sieci komputerowej w swojej firmie!

Tych bardziej zorientowanych w temacie oraz ciekawych szczegółów technicznych zapraszamy do zapoznania się z dokumentacją implementacyjną oraz techniczną.

3 Dokumentacja implementacyjna

Czyli szczegóły implementacyjne

3.1 Algorymiczna idea rozwiązania

Tworzymy sieć w formie ukorzonego drzewa (korzeń stanowi pierwsza maszyna). Nowy komputer dołączamy do takiego wierzchołka, który nie osiągnął maksymalnej pojemności i ma najniższą głębokość. Jeśli takich wierzchołków jest więcej, wybieramy ten najbardziej na lewo.

1. Nowo dodany komputer połączony będzie z co najwyżej jednym komputerem dodanym wcześniej. Dlatego nigdy nie powstanie nam cykl i sieć zachowa strukturę drzewa.
2. Każde drzewo ma liście. Pojemność wierzchołków jest większa od 2, zatem zawsze istnieje wierzchołek, do którego możemy coś dodać.
3. Zauważmy, że średnica powiększa się jedynie wtedy, gdy tworzymy nowy poziom w drzewie. Nasze postępowanie prowadzi jednak do tego, by nowy poziom był tworzony jak najrzadziej.

Oczywiście, globalnie rozwiązanie to nie jest optymalne, ale my musimy radzić sobie z problemem on-line!

3.2 Utrudnienie i rozwiązanie tego problemu

JEDNAKŻE, POJAWIA SIĘ PEWIEN PROBLEM: Co będzie, jeśli dostaniemy dwa requesty o dołączenie do sieci na raz? Możemy rozwiązać go w następujący sposób: root będzie znał strukturę drzewa. To on będzie znajdował maszynę, do której podpięty zostanie nowo dodany komputer.

3.3 Podstawowe dane na temat specyfikacji projektu

Zatem... jak się do tego zabrać technicznie? Na początku dokonaliśmy następujących wyborów:

WYBRANY JĘZYK: go

POŁĄCZENIE MIĘDZY KOMPUTERAMI: socket

POZOSTAŁE TECHNOLOGIE: Google Charts (<https://developers.google.com/chart/>)

3.4 Dlaczego golang?

Go oferuje łatwość tworzenia aplikacji sieciowych oraz wydajność. Aplikacja napisana w pythonie byłaby podobnie skomplikowana, natomiast mało wydajna. Tworzenie aplikacji sieciowych w C++ jest na tyle utrudnione, że nie chcieliśmy się w nie zagłębiać.

3.5 Dlaczego Google charts?

Jest to prosta, przydatna technologia, którą warto znać. Pozwala na tworzenie nawet bardzo skomplikowanych wykresów stosunkowo niewielkim kosztem - należy znać jedynie podstawy JavaScript (a właściwie - schemat HTML/CSS/JAVASCRIPT). Duża ilość dostępnych na stronie przykładów pozwala każdemu zagłębić się w tę technologię. Dlatego uznaliśmy, że jest ona idealna dla tworzenia wykresu naszej sieci.

3.6 Szczegóły rozwiązania: dodawanie nowego komputera

Informację o tym, że przychodzi nowy komputer, otrzymuje dowolna maszyna w naszej sieci. Następnie każdy wysyła zapytanie o wynik do swojego rodzica, w końcu dojdzie ono do korzenia.

Korzeń znajduje wierzchołek, do którego należy podpiąć nową maszynę na wspomniany wcześniej sposób. Można do tego użyć zwyczajnego algorytmu DFS.

Rozsyła później informację do wszystkich swoich dzieci, które przekazują ją dalej. Wiadomość rozprzestrzenia się w całej sieci.

3.7 Szczegóły rozwiązania: pierwotna idea na tworzenie wykresu

wierzchołek otrzymuje polecenie "zbuduj wykres". Wysyła wówczas do swojego rodzica tę wiadomość. Gdy wiadomość dochodzi do korzenia, tworzy on skrypt budujący wykres oraz rozsyła go do swoich dzieci z komunikatem mówiącym ZBUDOWAŁEM SKRYPT. Zostaje ona przekazana dalej.

3.8 Szczegóły rozwiązania: problem z tworzeniem wykresu i jego rozwiązanie

Jednakże, dziecko nie wie, czy zmiany w drzewie zaszły, zatem przy częstych prośbach o wykres nasza sieć zostanie zasypana niepotrzebnymi danymi.

Zatem nowy wykres będzie tworzony razem z dodaniem nowego wierzchołka i wówczas rozsyłany, co oznacza, że każda maszyna trzyma sobie update'owany wykres.

3.9 Wybrane protokoły

PROTOKÓŁ WARSTWY TRANSPORTOWEJ: tcp. Nie zależy nam na szybkości, ponieważ mamy do czynienia z małą, wewnętrzną siecią. Natomiast przy tworzeniu sieci bardzo ważne są dla nas bezpieczeństwo danych oraz dokładność.

Poniżej opisane są protokoły warstwy aplikacji.

3.10 Protokół SIP

PROTOKÓŁ 1 - PRZESYŁANIE INFORMACJI MIĘDZY KOMPUTERAMI (podczas tworzenia sieci, przekazywania informacji o klientach, "dogadywania się" w sprawie przesłania skryptu wykresu): wystarczy bardzo prosty protokół tekstowy. Jedyne informacje, jakie potrzebujemy, to: typ wiadomości, jej treść oraz ewentualna informacja o błędzie. Zauważmy, że jest to wiadomość, która jest bardzo niewielka, nie potrzebujemy zatem w żaden sposób "porcjować" danych. Będzie ona przekazywana w taki sposób, że wysyłamy tę samą wiadomość póki nie otrzymamy informacji zwrotnej - potwierdzenia jej otrzymania.

Opis użytych przez nas wiadomości oraz format danych znajduje się w dokumentacji.

Protokołowi temu zdecydowaliśmy nadać nazwę Simple Information Protocol - w skrócie SIP.

3.11 Protokół STP

PROTOKÓŁ 2 - PRZESYŁANIE WYKRESU: plik z wykresem może być duży, musimy zatem przysyłać go w mniejszych pakietach. Możemy użyć protokołu tekstowego i przekazać po prostu kod wykresu w JavaScriptcie. Zauważmy, że komputery mogą ustalić między sobą fakt, że zaraz nastąpi tranfser danych za pomocą naszego protokołu. Następnie możemy podzielić przesyłany skrypt na odpowiednio małe fragmenty i przysyłać je do momentu otrzymania informacji zwrotnej na ich temat.

Protokół ten nazwaliśmy Script Transfet Protocol - STP.

3.12 Krótkie uzasadnienie

Dlaczego zamiast używania SIP oraz STP nie użyjemy jednego protokołu, o formacie jak SIP, skoro równie dobrze można wysyłać podzielony plik za pomocą SIP? Zwróćmy uwagę na to, ile niepotrzebnych informacji zostałyby wówczas przekazane i jak bardzo spowolniłoby to transfer danych!

4 Dokumentacja techniczna

Czyli szczegóły techniczne

4.1 Teoria

4.2 Zadania

4.3 Rozwiązania

5 Podział pracy

Co kto robił?

5.1 Teoria

5.2 Zadania

5.3 Rozwiązania

6 Podsumowanie

Wnioski?

6.1 Dowcip

Czego potrzeba do ukończenia projektu z sieci?

Szczyпки kreatywności.