Санкт-Петербургский политехнический университет Петра Великого

Физико-механический институт

Высшая школа прикладной математики и вычислительной физики

Компьютерные сети
Отчёт по лабораторной работе №3
“Ситуативные сети”

Выполнил:

Студент: Парусов Владимир
Группа: 5040102/30201

Принял:

к. ф.-м. н., доцент
Баженов Александр Николаевич

2024 г.

Contents

1. Постановка задачи

Проводится турнир CTF. Имеется 4 противоборствующих команды, где задачей каждой команды является выгрузка большого файла с некоторого сервера. Прямого доступа к серверу нет, поэтому командам необходимо подключаться через промежуточные комньютеры. У каждого промежуточного компьютера есть своя область видимости, что позволяет представить данную ситуацию как ориентированный граф: где вершины - промежуточные компьютеры сервер и команды, а связи - наличие доступа из одной вершины к другой. Каждая команда использует вершины данного графа чтобы построить кратчайший путь до сервера и начинает закачку файла. Для того чтобы воспользоваться промежуточным комьютером как узлом, каждой команде необходимо его захватить при этом один промежуточный компьютер может быть захвачен только одной командой.

Необходимо реализовать симуляцию вышеописанного турнира, и рассмотреть различные модели поведения команд.

2. Теория

Для симуляции различной области видимости у каждого из промежуточных компьютеров, было решено отобразить каждый компьютер на плоскости и для каждого ввести "радиус". Тогда мы имеем 3 возможных ситуации

1. Два компьютера не пересекаются радиусами (Рис 1) - что соответствует отсутствию связи
2. Один компьютер попадает в окружность другого, но не находится захватывает его своей окружностью (Рис 2) - что соответствует тому, что один из компьютеров находится в дочерней локальной сети, относительно второго
3. Оба компьютера попадают в окружности друг друга (Рис 3) - что соответствует тому, что они находятся в одной локальной сети

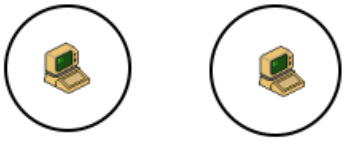

Figure 1: Пример отсутсвия связи

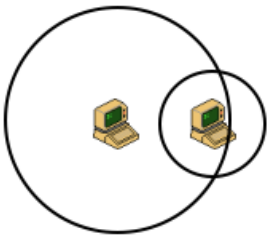

Figure 2: Пример односторонней связи

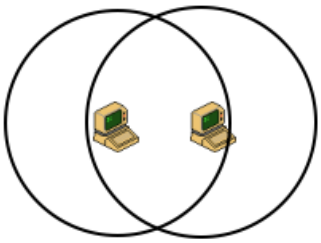

Figure 3: Пример двусторонней связи

Для симуляции захвата вводится величина "*мощности*" узла, выдаваемся следующим образом. Если промежуточный компьютер не принадлежит никакой команде - то он получает 100 еденицы "*мощности*". Если промежуточный компьютер принадлежит какой-то команде, то каждые $10ms$ он получает 1 еденицу мощности. Каждая команда может произвести "*атаку*" - перенаправить с любого своего промежуточного компьютера $power - 1$ еденицы "*мощности*" (где *power* - текущее значение "*мощности*" на данном устройстве) на любой другой промежуточный компьютер в области видимости. Далее:

- Если целевой промежуточный компьютер является компьютером этой команды то он увеличивет свой показатель "*мощности*" на переданную величину
- Если целевой промежуточный компьютер является компьютером другой команды (или нейтральным), то он уменьшает свой показатель "*мощности*" на переданную величину. Если показатель стал не положительным, тогда целевой компьютер захватывается, а его показатель мощности домножается на -1.

Для симуляции процесса закачки с возможными обрывами связи используется алгоритм Selective Repeat, рассмотренный в лабораторной №1, а для симуляции поиска соединения между командой и сервером используется алгоритм Open Shortest Path First, рассмотренный в лабораторной №2.

3. Реализация

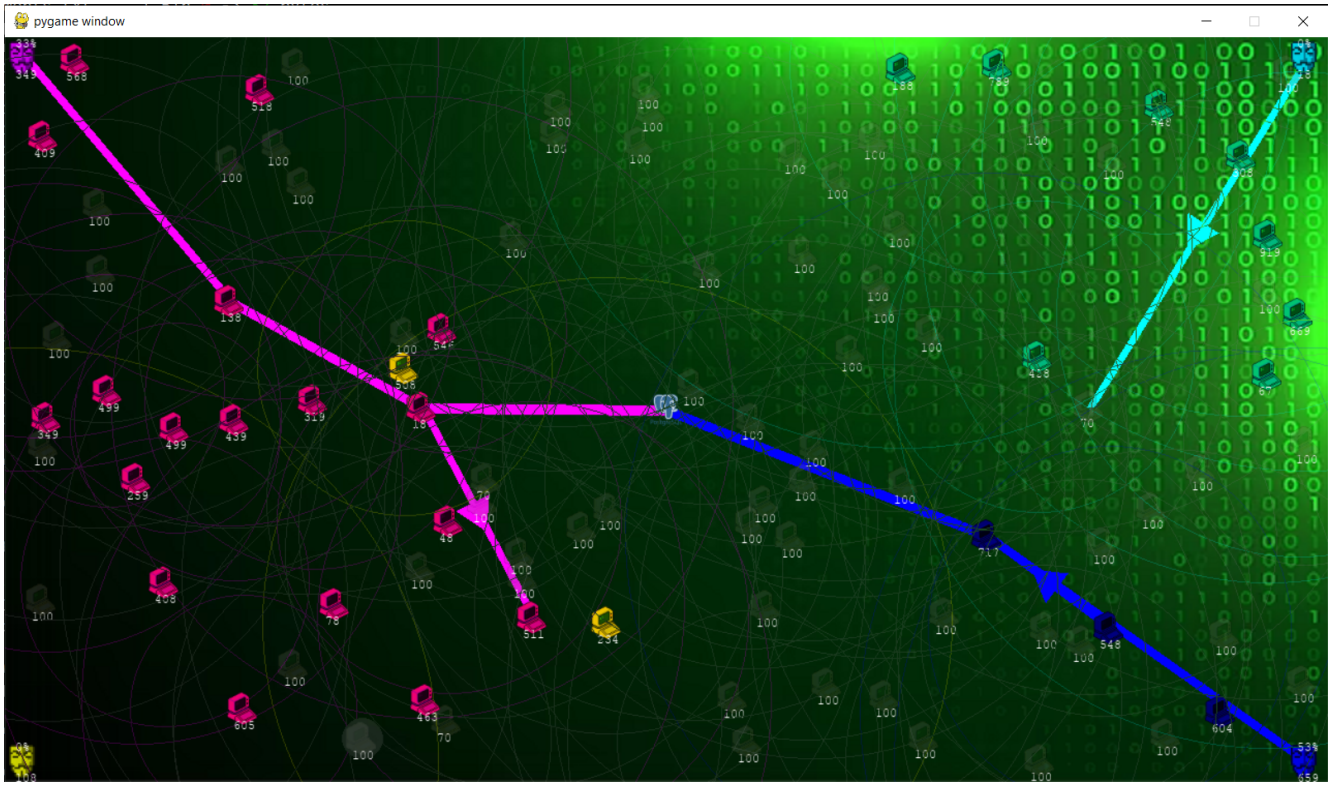

Figure 4: Скриншот работы приложения

Приложение было реализовано на языке программирования Python с использованием библиотеки pygame. Каждая команда представляется своим цветом и иконкой в одном из 4 углов. Сервер с данными представлен в виде иконки по центру экрана. Каждый из промежуточных компьютеров также представляется в виде иконки с цветовой индентификацией команды, а также числом - показателем "*мощности*". При наличии связи между какой-либо командой и сервером с данными, кратчайший путь рисуется линией цвета команды. При совершении *атаки*, на экране рисуется линия цвета команды со стрелкой по центру.

В качестве расмотриваемых моделей поведения команд, были выбраны следующие:

- Случайная - каждый ход выполняется случайное из возможных действий
- Жадная - каждый ход производится атака на наименее "*мощный*" промежуточный узел не принадлежащий данной команде, который возможно захватить. Если таких нет - ход пропускается.
- Агрессивная - аналогично жадной, но при этом отдается предпочтение захвату узлов другой команды
- Пользовательская - каждый ход контроллируется пользователем

4. Результаты

По результатам проведенных симуляций было замечено, что при разных конфигурациях промежуточных узлов разные модели являются выигрышными.

Если узлы расположены случайно, то Жадная модель побежает чаще, чем Агрессивная и Случайная.

Если узлы расположены равномерно, то Агрессивная модель побежает чаще, чем Жадная и Случайная.

5. Обсуждение

В результате работы реализовано приложение, использующее алгоритмы Selective Repeat и Open Shortest Path First для симуляции CTF турнира описанного в постановке задачи. Данная симуляция является симуляцией ситуативных сетей, так как любой имеющийся у команды узел, может быть захвачен другой командой. Благодаря описанным выше алгоритмам в такой ситуации есть возможность перестроить свой путь связи и всё равно получитьтребуемый файл.

6. Приложения

1. Репозиторий с кодом программы и GIF демонстрацией работы проекта https://github.com/sairsey/compnet