Санкт-Петербургский Политехнический университет Петра Великого
Институт прикладной математики и механики
Высшая школа прикладной математики и вычислительной физики

КУРСОВАЯ РАБОТА

по дисциплине «Компьютерные сети»

Выполнил:
Чернова В.С.
Группа:
3640102/90201

Проверил:
к.ф.-м.н., доцент
Баженов А.Н.

Санкт-Петербург
2020 г

ОГЛАВЛЕНИЕ

ПОСТАНОВКА ЗАДАЧИ

Курсовая работа включает в себя следующие задачи:

- Моделирование системы слежения за солнечным зайчиком, состоящей из четырех камер (отдельных узлов сети) и связанного с ними роутера;

- Реализация алгоритма, решающего задачу византийских генералов, для устранения угрозы подмены данных с камеры злоумышленником;

- Включение разработанной системы в уже существующую адаптивную систему фокусировки зеркал.

ХОД РАБОТЫ

1. Описание программы

Программа была выполнена на языке программирования Python 3.7 в среде разработки JetBrains PyCharm.

Программа содержит следующие основные классы:

Sender – класс, отвечающий за отправку пакетов. Может работать как по протоколу Go back N, так и по протоколу Selective repeat. Выбор протокола происходит при инициализации класса. Также передаваемыми параметрами являются: размер окна, количество данных для передачи, вероятность потери передаваемого пакета.

Receiver – класс, отвечающий за прием пакетов. Отправляет сообщение ACK, если пакет был успешно получен, и сообщение NAK в противном случае.

Screen – класс, представляющий собой экран по которому движется солнечный зайчик. Размеры и координаты экрана, а также координаты и диаметр солнечного зайчика могут быть заданы произвольно. В данном классе реализован расчет точек пересечения с матрицами камер.

Sensor – класс, представляющий собой узел сети с камерой (сенсором). Имеет возможность проведения массовой рассылки по протоколу Go back N либо Selective repeat всем остальным узлам с камерами. Данная рассылка используется для реализации алгоритма византийских генералов.

DesignatedRouter – класс, представляющий собой специально выделенный роутер. Он является смежным со всеми остальными узлами (с камерами) в сети и хранит её топологию. После получения от камер координат их пересечения с солнечным зайчиком рассчитывает его центр.

2. Демонстрация работы программы

На рис. 1 представлен экран с солнечным зайчиком, реализуемый классом Screen. Солнечный зайчик движется случайно в пределах экрана с некоторым заданным шагом. При каждом новом движении рассчитываются точки пересечения окружности с матрицами камер. На рисунке границы видимости камер выделены черными линиями. Каждая камера видит ровно 1/4 экрана.

Для размера солнечного зайчика пришлось ввести ограничение. Необходимо, чтобы его диаметр был не меньше ширины видимости камеры. В противном случае зайчик может оказаться в поле зрения только одной камеры, что исключает его пересечения с матрицами камер.

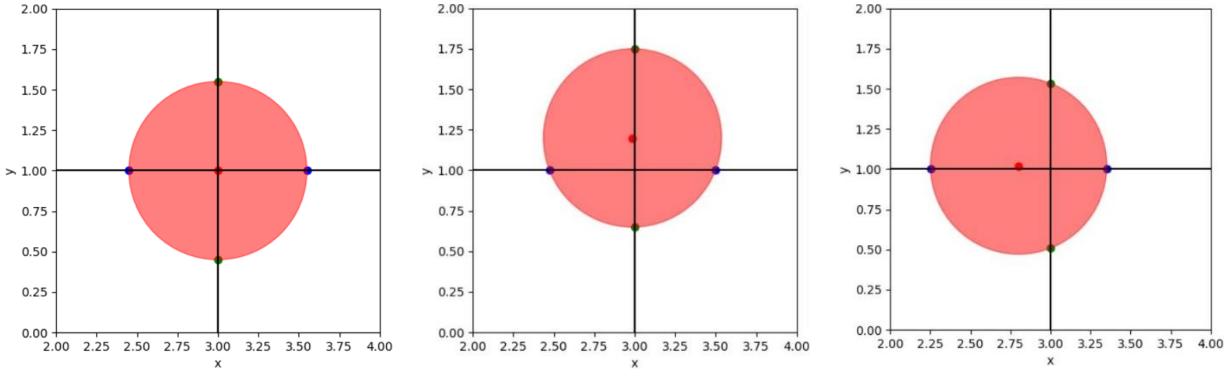

Рис. 1. Движение солнечного зайчика по экрану

Узел с камерой (класс Sensor), получив видимые ей координаты пересечения с солнечным зайчиком, начинает обмениваться координатами с другими узлами сети, реализуя алгоритм византийских генералов. Передача данных осуществляется по протоколам Go back N либо Selective repeat (можно выбрать один из них). Далее подробно рассмотрим реализацию алгоритма византийских генералов в данной программе.

В нашем случае имеется четыре камеры, одна из которых взломана нарушителем и передает неверные данные. Пусть взломанной камерой будет камера под номером 3. В подобном случае алгоритм осуществляется в четыре шага.

1-й шаг. Каждый узел с камерой посылает всем остальным сообщение со своими координатами пересечения. Нетронутые злоумышленником камеры указывают действительные координаты, в то время как взломанная камера посылает случайные значения.

2-й шаг. Каждый узел формирует свой вектор из имеющейся информации:

- Вектор камеры №1: (1,2,x,4);
- Вектор камеры №2: (1,2,y,4);
- Вектор камеры №3: (1,2,3,4);
- Вектор камеры №4: (1,2,z,4).

Здесь: 1 – координаты, полученные от первой камеры, 2 – от второй, 4 – от четвертой. А x, y, z – случайные координаты, отправленные злоумышленником.

3-й шаг. Каждый посылает свой вектор всем остальным (узел 3 посылает опять произвольные значения). После этого у каждого узла с камерой есть по четыре вектора:

cam1	cam2	cam3	cam4
(1,2,x,4)	(1,2,x,4)	(1,2,x,4)	(1,2,x,4)
(1,2,y,4)	(1,2,y,4)	(1,2,y,4)	(1,2,y,4)
(a,b,c,d)	(e,f,g,h)	(1,2,3,4)	(i,j,k,l)
(1,2,z,4)	(1,2,z,4)	(1,2,z,4)	(1,2,z,4)

4-й шаг. Каждый узел с камерой определяет для себя координаты пересечений всех остальных камер. Чтобы определить координаты i-й камеры, каждый узел сети берёт по три числа — координаты этой камеры, пришедшие от всех узлов, кроме узла под номером i. Если какое-то значение повторяется среди этих трех чисел как минимум два раза, то оно помещается в результирующий вектор, иначе соответствующий элемент результирующего вектора остается пустым. Таким образом все исправные камеры получают один и тот же вектор, согласие достигнуто.

Реализованная программа выводит на экран информацию о совершаемых действиях. Пример вывода программы представлен на рис. 2.

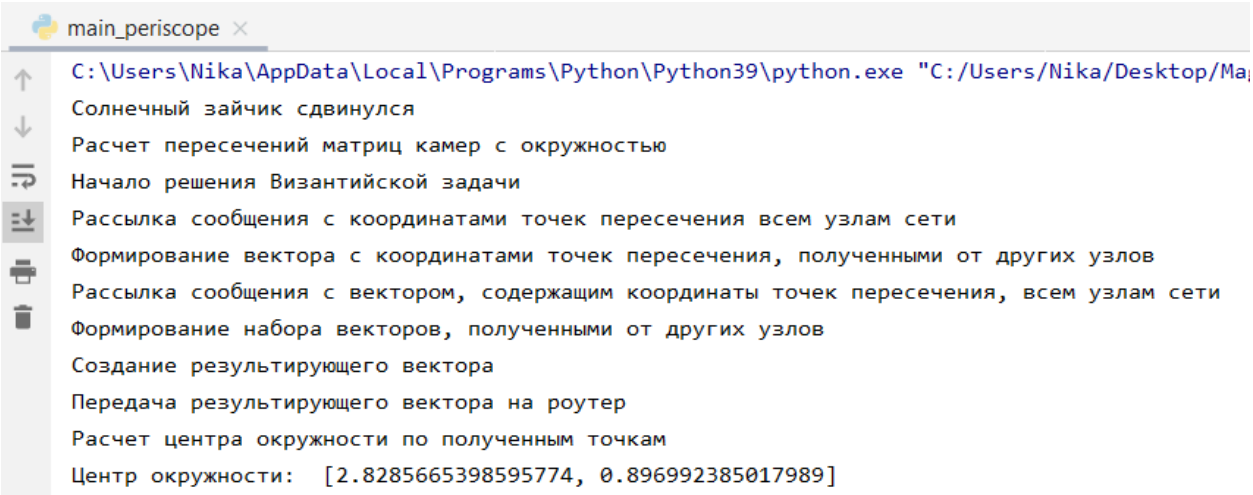

Рис. 2. Пример вывода программы

После совершения всех описанных выше действий программа передает координаты центра окружности в адаптивную систему фокусировки зеркал. На рис. 3 можно увидеть, что точка фокусировки луча E соответствует посчитанным выше координатам, а именно координатам {2.83; 0.89}.

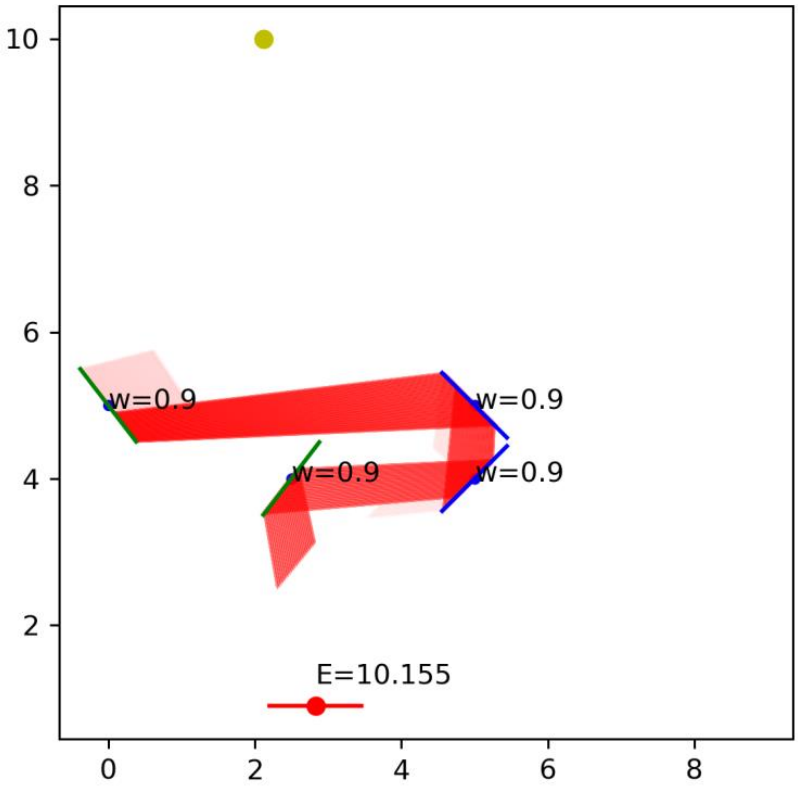

Рис. 3. Поведение луча в системе фокусировки зеркал

ЗАКЛЮЧЕНИЕ

На языке программирования Python 3.7 была реализована программа, моделирующая систему слежения за солнечным зайчиком, которая состоит из четырех камер (отдельных узлов сети) и связанного с ними роутера.

Был реализован алгоритм, решающий задачу византийских генералов, для устранения угрозы подмены данных с камеры злоумышленником.

Разработанная система была успешно включена в уже существующую адаптивную систему фокусировки зеркал.

Код программы Андрея Константинова, реализующей систему фокусировки зеркал, находится по следующей ссылке: https://github.com/andruekonst/computer-networks/tree/master/lab_3.

Код написанной в данной работе программы, вместе с включенной в нее программой фокусировки зеркал, представлен по следующей ссылке: https://github.com/nika2506/comp_networks_labs/tree/main/Coursework.

ЛИТЕРАТУРА

1. Задача византийских генералов [Электронный ресурс], URL: https://ru.wikipedia.org/wiki/Задача_византийских_генералов. Дата обращения: 27.12.2020

2. The Python Tutorial [Электронный ресурс], URL: https://docs.python.org/3/tutorial/index.html. Дата обращения: 27.12.2020