

**ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ**  
Федеральное государственное образовательное бюджетное учреждение высшего профессионального  
образования  
**«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕЛЕКОММУНИКАЦИЙ И  
ИНФОРМАТИКИ»**

Кафедра Мультисервисных сетей и информационной безопасности

Н.В. Киреева, М.А. Буранова

## **Моделирование сети Ethernet**

**Методические указания к выполнению лабораторной работы**

Самара, 2015

УДК

ББК

Рекомендовано к изданию методическим советом ПГУТИ,

Протокол № 18 ,от 02.04.2015 г.

**Рецензент:**

Доцент кафедры МСИБ, к.т.н. В.В. Пугин

Киреева, Н.В. Буранова М.А.

Моделирование сети Ethernet: методические указания к выполнению лабораторной работы / Н.В. Киреева, М.А Буранова. - Самара: ПГУТИ, 2015. – 28с.

Методические указания «Моделирование сети Ethernet» содержат необходимую информацию для написания лабораторных работ, разработано в соответствии с ФГОС ВПО по направлению подготовки специальностей 11.03.02, 10.05.02 и предназначено для выполнения лабораторных работ студентами.

©, Киреева Н.В., Буранова М.А., 2015

## 1. Цель работы

Исследовать методы маршрутизации в компьютерных сетях, провести моделирование сети Ethernet и рассчитать характеристики сети.

## 2. Общие понятия

Алгоритм Дейкстры (англ. Dijkstra's algorithm) — алгоритм на графах, изобретённый нидерландским ученым Э. Дейкстрой в 1959 году. Находит кратчайшее расстояние от одной из вершин графа до всех остальных. Алгоритм работает только для графов без рёбер отрицательного веса. Алгоритм широко применяется в программировании и технологиях, например, его использует протокол OSPF для устранения кольцевых маршрутов.

Каждой вершине из  $V$  сопоставим метку — минимальное известное расстояние от этой вершины до  $a$ . Алгоритм работает пошагово — на каждом шаге он «посещает» одну вершину и пытается уменьшать метки. Работа алгоритма завершается, когда все вершины посещены.

### 1) Инициализация

Метка самой вершины  $a$  полагается равной 0, метки остальных вершин — бесконечности. Это отражает то, что расстояния от  $a$  до других вершин пока неизвестны. Все вершины графа помечаются как непосещённые.

### 2) Шаг алгоритма

Если все вершины посещены, алгоритм завершается. В противном случае, из ещё не посещённых вершин выбирается вершина  $u$ , имеющая минимальную метку. Мы рассматриваем всевозможные маршруты, в которых  $u$  является предпоследним пунктом. Вершины, в которые ведут рёбра из  $u$ , назовем соседями этой вершины. Для каждого соседа вершины  $u$ , кроме отмеченных как посещённые, рассмотрим новую длину пути, равную сумме значений текущей метки  $u$  и длины ребра, соединяющего  $u$  с этим соседом. Если полученное значение длины меньше значения метки соседа, заменим значение метки полученным значением длины. Рассмотрев всех соседей, пометим вершину  $u$  как посещённую и повторим шаг алгоритма.

## Пример

Рассмотрим выполнение алгоритма на примере графа, показанного на рисунке. Пусть требуется найти кратчайшие расстояния от 1-й вершины до всех остальных.

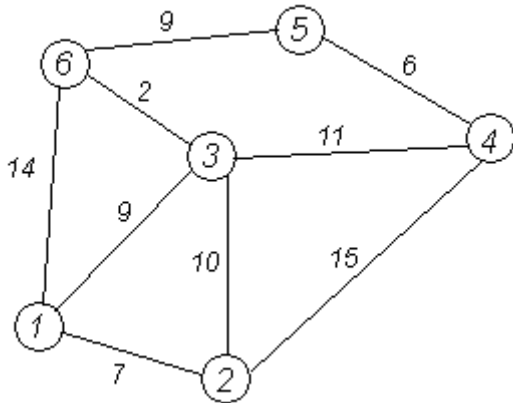


Рис.1

Кружками обозначены вершины, линиями — пути между ними (ребра графа). В кружках обозначены номера вершин, над ребрами обозначена их «цена» — длина пути. Рядом с каждой вершиной красным обозначена метка — длина кратчайшего пути в эту вершину из вершины 1.

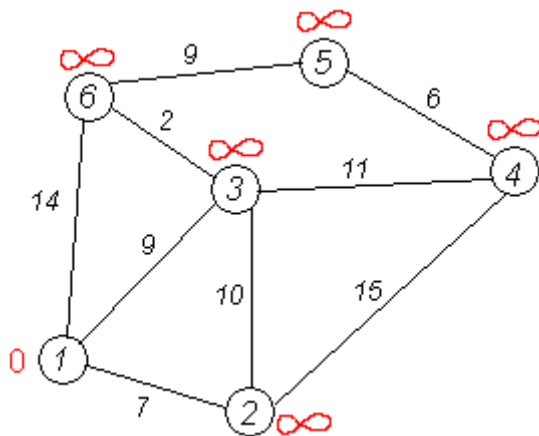


Рис.2

Первый шаг. Рассмотрим шаг алгоритма Дейкстры для нашего примера. Минимальную метку имеет вершина 1. Её соседями являются вершины 2, 3 и 6.

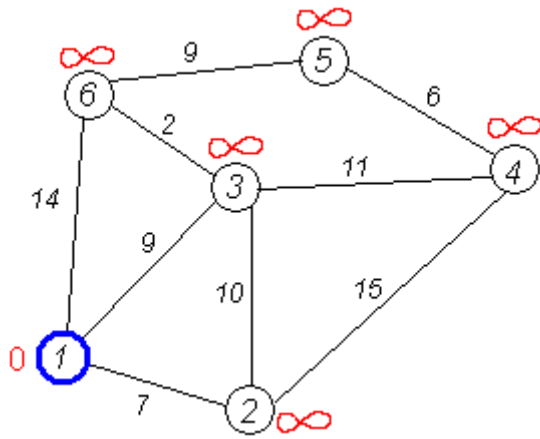


Рис.3

Первый по очереди сосед вершины 1 — вершина 2, потому что длина пути до неё минимальна. Длина пути в неё через вершину 1 равна сумме кратчайшего расстояния до вершины 1, значению её метки, и длины ребра, идущего из 1-й в 2-ю, то есть  $0 + 7 = 7$ . Это меньше текущей метки вершины 2, бесконечности, поэтому новая метка 2-й вершины равна 7.

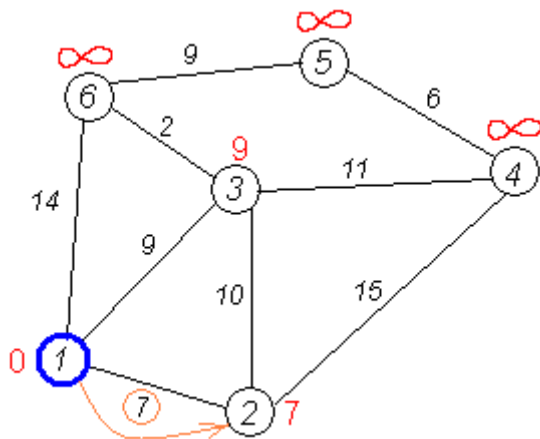


Рис.4

Аналогичную операцию проделываем с двумя другими соседями 1-й вершины — 3-й и 6-й.

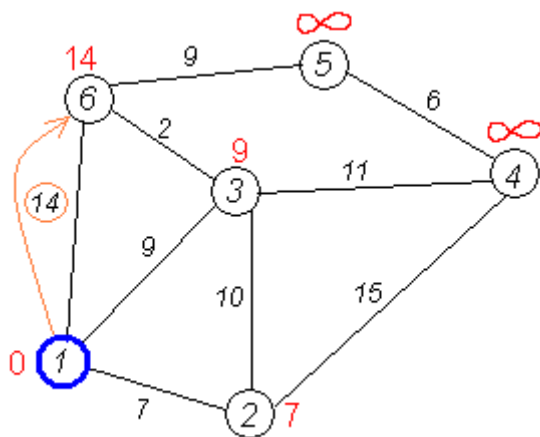


Рис.5

Все соседи вершины 1 проверены. Текущее минимальное расстояние до вершины 1 считается окончательным и пересмотру не подлежит (то, что это действительно так, впервые доказал Э. Дейкстра). Вычеркнем её из графа, чтобы отметить, что эта вершина посещена.

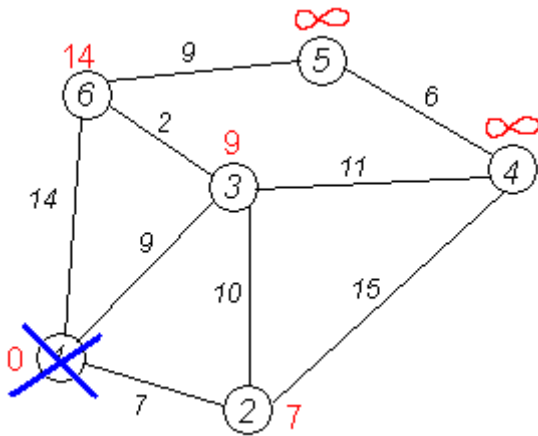


Рис.6

Второй шаг. Шаг алгоритма повторяется. Снова находим «ближайшую» из непосещенных вершин. Это вершина 2 с меткой 7.

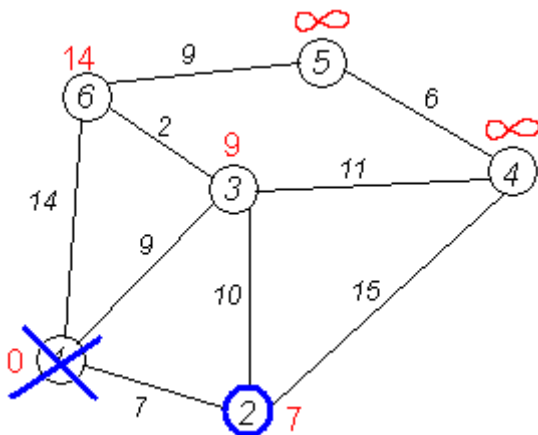


Рис.7

Снова пытаемся уменьшить метки соседей выбранной вершины, пытаемся пройти в них через 2-ю вершину. Соседями вершины 2 являются вершины 1, 3 и 4.

Первый (по порядку) сосед вершины 2 — вершина 1. Но она уже посещена, поэтому с 1-й вершиной ничего не делаем.

Следующий сосед вершины 2 — вершина 3, так как имеет минимальную метку из вершин, отмеченных как не посещённые. Если идти в неё через 2, то длина такого пути будет равна 17 ( $7 + 10 = 17$ ). Но текущая метка третьей вершины равна  $9 < 17$ , поэтому метка не меняется.

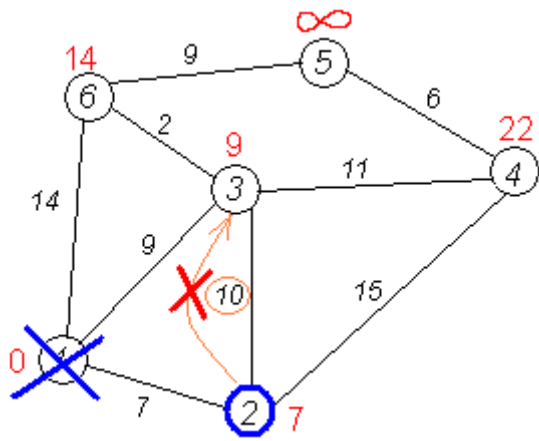


Рис.8

Ещё один сосед вершины 2 — вершина 4. Если идти в неё через 2-ю, то длина такого пути будет равна сумме кратчайшего расстояния до 2-й вершины и расстояния между вершинами 2 и 4, то есть 22 ( $7 + 15 = 22$ ). Поскольку  $22 < \infty$ , устанавливаем метку вершины 4 равной 22.

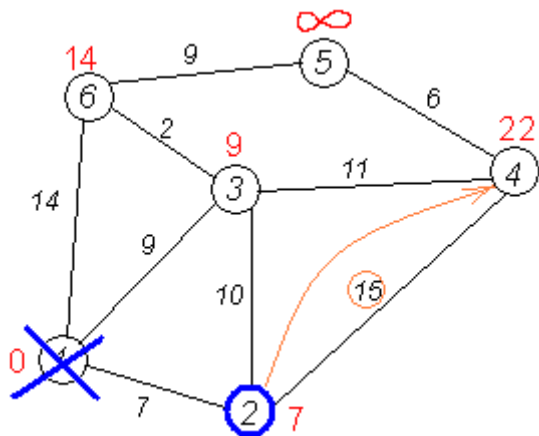


Рис.9

Все соседи вершины 2 просмотрены, замораживаем расстояние до неё и помечаем её как посещённую.

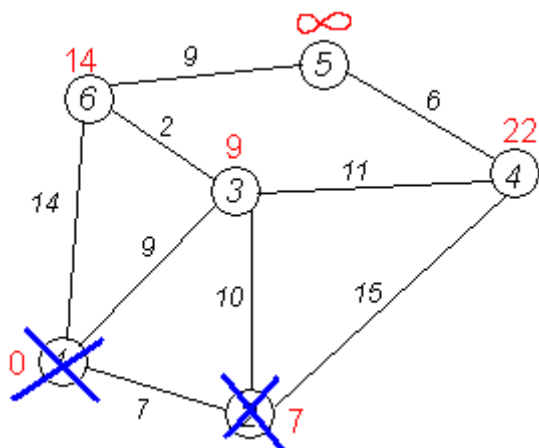


Рис.10

Третий шаг. Повторяем шаг алгоритма, выбрав вершину 3. После её «обработки» получим такие результаты:

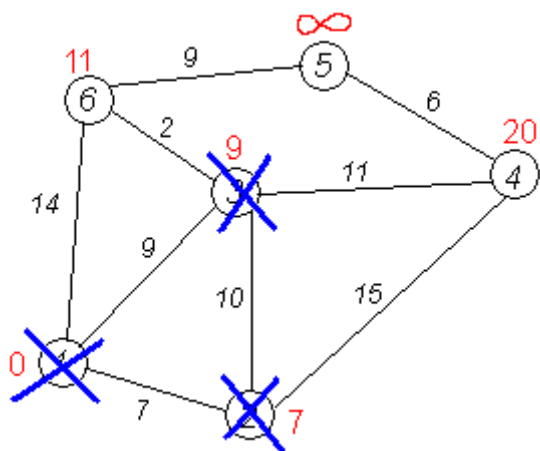


Рис.11

Дальнейшие шаги. Повторяем шаг алгоритма для оставшихся вершин. Это будут вершины 6, 4 и 5, соответственно порядку.

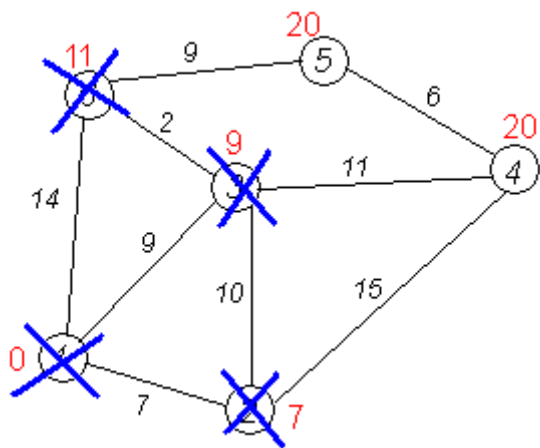


Рис.12



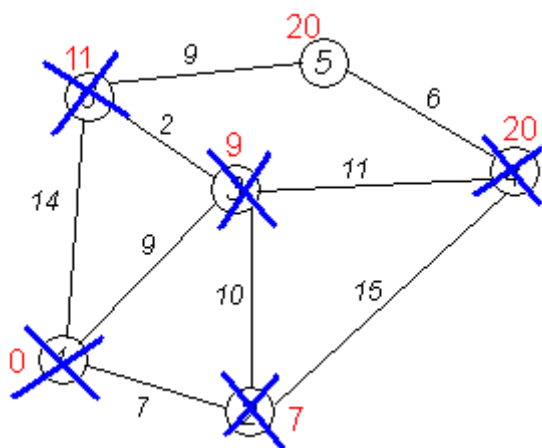


Рис.13

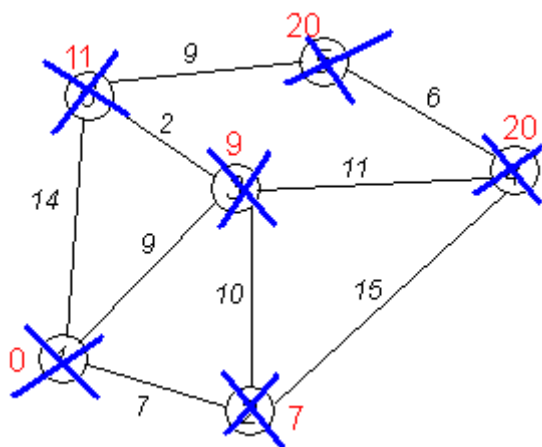


Рис.14

Завершение выполнения алгоритма. Алгоритм заканчивает работу, когда нельзя больше обработать ни одной вершины. В данном примере все вершины зачеркнуты, однако ошибочно полагать, что так будет в любом примере - некоторые вершины могут остаться незачеркнутыми, если до них нельзя добраться. Результат работы алгоритма виден на последнем рисунке: кратчайший путь от вершины 1 до 2-й составляет 7, до 3-й — 9, до 4-й — 20, до 5-й — 20, до 6-й — 11.

### 3. Расчет конфигурации сетей Fast Ethernet

Как и все некоаксиальные варианты Ethernet технология Fast Ethernet рассчитана на использование повторителей (концентраторов) для образования древовидной иерархической топологии. Повторители Fast Ethernet делятся на два класса:

- повторители 1-го класса (поддерживают все типы логического кодирования - 4В/5В и 8В/6Т);

- повторители 2-го класса (поддерживают только один тип логического кодирования).

Повторители 1-го класса могут транслировать протоколы (например из 100Base-TX в 100Base-FX и наоборот) из-за чего вносят при передаче сигнала большую задержку - 70 битовых интервалов (bt). Повторители второго класса не транслируют протоколы и поэтому вносят гораздо меньшую задержку - 46bt для TX/FX и 33,5bt для 100Base-T4.

В одном домене коллизий возможно наличие только одного повторителя 1-го класса или 2-х повторителей 2-го класса (при этом они должны быть соединены кабелем не длиннее 5 метров). Небольшое количество повторителей в домене коллизий не препятствует построению больших сетей Fast Ethernet, т.к. современные сети строятся на коммутаторах, которые делят сеть на несколько доменов коллизий.

При определении корректности конфигурации сети можно не руководствоваться вышеизложенными общими правилами, а рассчитывать время двойного оборота (PDV), как это делалось в Ethernet-сетях на 10Мбит/с.

Для этого комитет IEEE 802.3 приводит данные об удвоенных задержках, вносимых кабельными сегментами, сетевыми адаптерами и повторителями Fast Ethernet. По сравнению с аналогичными данными для Ethernet-сетей методика расчета несколько изменилась - сегменты теперь не делятся на левый, правый и промежуточные; кроме того, вносимые сетевыми адаптерами задержки учитывают теперь преамбулы кадров, поэтому рассчитанное значение PDV нужно сравнивать не с 575bt, а с 512bt, т.е. временем передачи кадра минимальной длины без преамбулы. В соответствии с рекомендациями IEEE достаточным является запас в 4-6 битовых интервалов.

Задержки, вносимые кабелем

Тип кабеля	Задержка, bt на 1м
UTP cat.3	1,14
UTP cat.4	1,14
UTP cat.5	1,112
STP	1,112
Оптическое волокно	1,0

## Удвоенные задержки Router'ов

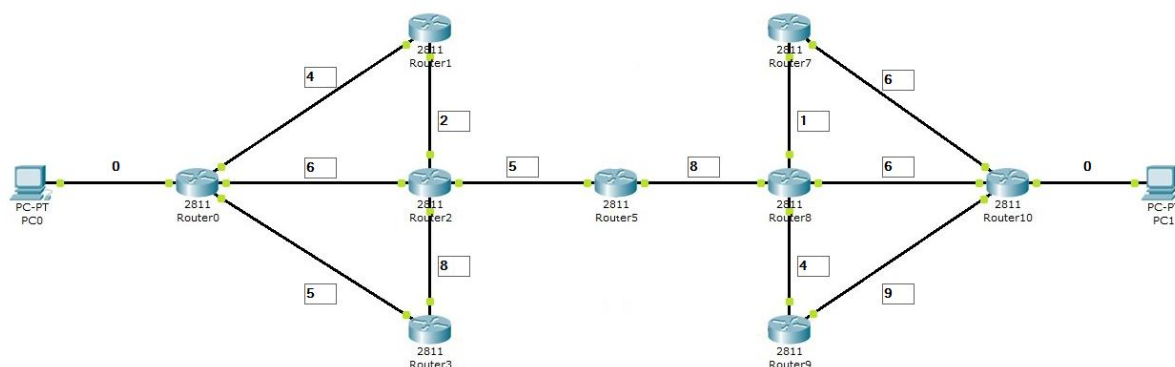
Тип повторителей	Задержка, bt
класса 1	140
класса 2(RJ-45)	67
класса 2(TX/FX)	92

Для упрощения расчетов будем считать что сеть построена с помощью кабеля STP и используем роутеры 2 класса. Вес графа примем условно за 1 метр, Тип повторителей класс 2(RJ-45).

$T=l*1.112+tr*67$ , где T суммарная задержка, l длина до роутера, tr задержка в роутере.

### Задание:

Для приведенной ниже схемы графов рассчитать минимальное расстояние до роутера, согласно варианту:



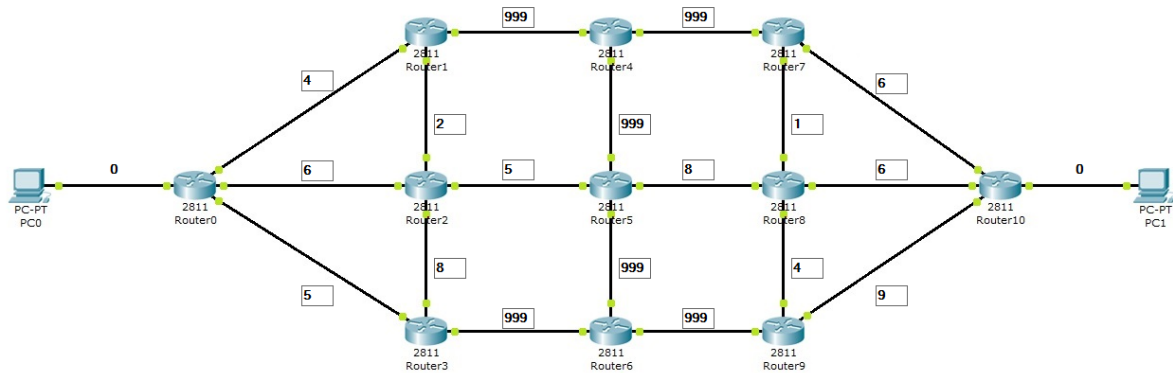
№ Варианта	Расчет до Router
1	7
2	8
3	9
4	10

1. Рассчитать задержку до роутера.
2. Результат показать преподавателю.
3. Проверить расчет с помощью программы.

Ход работы:

Необходимо расставить веса графов согласно приведенной ниже таблице.

Router0-Router1	4
Router0-Router2	6
Router0-Router3	5
Router1-Router4	2
Router1-Router2	8
Router2-Router5	999
Router2-Router3	5
Router3-Router6	999
Router4-Router7	999
Router4-Router5	999
Router5-Router8	999
Router5-Router6	8
Router6-Router9	999
Router7-Router10	1
Router7-Router8	4
Router8-Router10	6
Router8-Router9	6
Router9-Router10	9



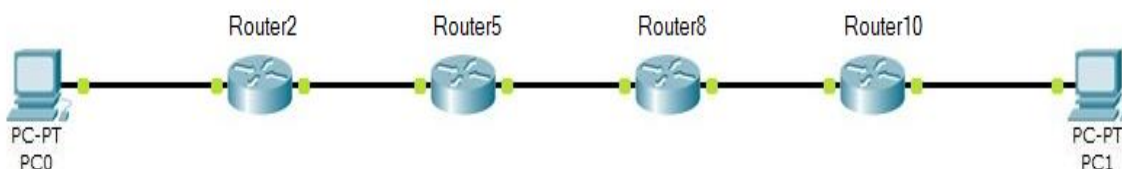
Произвести моделирование сети Ethenet.

Сравнить полученный результат с результатом полученным при расчете.

№ Варианта	Расчет до Router
1	7
2	8
3	9
4	10

Минимальный путь рассчитанный в программе показывается в обратном порядке.

Граф нарисованный программой показывает кратчайшее расстояние от Роутера0 до Роутера10



Результат проверки расчета показать преподавателю.

Сделать выводы.

### **Контрольные вопросы:**

1. В каком году был изобретен алгоритм Дейкстры?
2. Основой какого протокола маршрутизации стал алгоритм Дейкстры?
3. В чем отличие концентраторов 1 и 2 класса?
4. Каков достаточный запас битовых интервалов на сети Ethernet?
5. На каких скоростях может работать технология Ethernet?

## Глоссарий

ELAN (Emulated Local Area Network) – эмулируемая локальная сеть

IP (Internet Protocol) – протокол Internet

ISU (Information Symbol Unit) – единица передачи информации

LAN (Local Area Network) – локальная вычислительная сеть

LLC (Logical Link Control) – подуровень управления логической связью

MAC( Media Access Control) – подуровень управления доступом к устройствам

NIC (Network Interface Card) – сетевой адаптер

NMS (Network Management System) – система управления сетью

OSI (Open System Interconnection) – сетевая модель

SCS (Structured Cabling System) – структурированная кабельная система

SLIP (Serial Line IP) – IP для последовательных линий

SMTP (Simple Mail Transfer Protocol) – простой протокол почтового обмена

SNMP (Simple Network Management Protocol) – простой протокол сетевого управления

TCP (Transmission Control Protocol) – протокол управления передачей стека TCP/IP

UDP (User Datagram Protocol) – пользовательский протокол дейта-грамм стека TCP/IP

UTP (Unshielded Twisted Pair) – неэкранированная витая пара

VLAN (Virtual Local Areal Network) – виртуальная локальная сеть

WAN (Wide Area Network) – глобальная вычислительная сеть

ПК – персональный компьютер

## Литература

1. Олифер В.Г., Олифер Н.А. «Компьютерные сети». Изд-е 5-е. СПб: Питер, 2013.
2. «Microsoft Основы компьютерных сетей». Бинوم. Лаборатория знаний. 2006.
3. Филимонов А.Ю. «Построение мультисервисных сетей Ethernet». М.: ВНУ, 2007.
4. Чекмарев Ю.В. Локальные вычислительные сети. Учебное пособие. ДМК пресс, 2009.
5. Кузин А.К. Компьютерные сети. М., Умные книги, 2011.
6. Кузьменко Н.Г. Компьютерные сети и сетевые технологии. М., Наука и техника, 2013.

