ФЕДЕРАЛЬНОЕ АГЕНТСТВО СВЯЗИ
Северо-Кавказский филиал ордена Трудового Красного Знамени федерального государственного бюджетного образовательного учреждения высшего образования
«Московский технический университет связи и информатики»

Методические указания для проведения лабораторных работ
по дисциплине

# Системное администрирование инфокоммуникационных систем

(направление подготовки 11.03.02
Инфокоммуникационные технологии и системы связи)

Ростов-на-Дону
2019

Методические указания для проведения лабораторных работ
по дисциплине

**Системное администрирование инфокоммуникационных систем**

(направление подготовки 11.03.02
Инфокоммуникационные технологии и системы связи)

Составил: И.А. Сосновский, доцент кафедры ИТСС

Рассмотрено и одобрено
на заседании кафедры
Протокол от «26» августа 2019 г. № 1

***Лабораторная работа №1.*** Бесклассовая адресация CIDR и маски переменной длины VLSM.

***Цель лабораторной работы***: Исследовать возможности распределения диапазонов IP-адресов.

**Теоретическая часть.**

Масштабируемая сеть требует схемы адресации допускающей рост. Однако вследствие неконтролируемого роста сети могут возникнуть ряд непредвиденных последствий. По мере добавления узлов и подсетей в сеть предприятия может возникнуть нехватка свободных адресов и потребуется изменение схемы существующих адресов. Этого можно избежать путём тщательного планирования масштабируемой адресной системы сети предприятия.

К сожалению, архитекторы TCP/IP не могли предсказать экспоненциального роста Интернет, и в настоящее время остро стоит проблема распределения адресов.

Когда в 80-х годах внедрялся TCP/IP, он базировался на двухуровневой адресной схеме. Старшая часть 32-битового IP адреса определяла номер (адрес) сети, а младшая - номер хоста. Адрес сети необходим для взаимодействия сетей. Маршрутизаторы используют сетевую часть адреса для организации связи между хостами из различных сетей.

Для удобства человеческого восприятия IP адрес записывается в виде четырёх десятичных чисел, разделённых точками. 32-битовый адрес делится на четыре группы по восемь бит, называемых октетами. Каждый октет записывается в десятичном виде и разделяется точками. Например

10101100000111101000000000010001

<-> 10101100 00011110 10000000 00010001

<-> 172 30 128 17

<->.172.30.128.17

Возникает вопрос, как в любом IP адресе выделить адрес сети и адрес хоста? В начале использования TCP/IP для решения этого вопроса использовалась классовая система адресации. IP адреса были разбиты на пять непересекающихся классов. Разбивка осуществлена согласно значениям нескольких первых бит в первом октете.

Если первый бит в первом октете равен нулю, то это адрес класса A. Адреса класса B начинаются с бинарных 10. Адреса класса C начинаются с бинарных 110.

В адресах класса A адрес сети располагается в первом октете. В классе B для адресации сети используется первый и второй октеты. В классе C для адресации сети используется первый, второй и третий октеты. Использование классов D и E специфично и здесь не рассматривается.

В современных сетях классы часто игнорируются, а используется бесклассовая IP схема, основанная на масках подсетей.

Здесь и далее мы будем использовать маски в виде последовательности бинарных единиц, переходящей в последовательность бинарных нулей общей длинной в 32 бита. Маски принято записывать в десятичной форме подобно IP адресам

11111111111111110000000000000000 <->

11111111 1111111 0000000 0000000 <->

255 255 0 0 <->

255.255.0.0

Маска подсети является необходимым дополнением к IP адресу. Если бит в IP адресе соответствует единичному биту в маске, то этот бит в IP адресе представляет номер сети, а если бит в IP адресе соответствует нулевому биту в маске, то этот бит в IP адресе представляет номер хоста. Так для маски 255.255.0.0 и адреса 172.24.100.45 номер сети будет 172.24.0.0, а для маски 255.255.255.0 номер сети будет 172.24.100.0.

Другая форма записи маски - /N, где N – число единиц в маске. Эта форма используется только в сочетании с IP адресом. Например, для маски 255.255.0.0 и адреса 172.24.100.45 пишут 172.24.100.45/16.

Все адреса класса А имеют маску 255.0.0.0, адреса класса В имеют маску 255.255.0.0, а адреса класса С имеют маску 255. 255. 255.0. Обратное утверждение неправомерно, так как при определении класса используются первые биты в первом октете адреса.

Если организация располагает сетью класса В (маска 255.255.0.0), то она может разбить эту сеть на подсети, используя маску 255.255.255.0. Например, если адрес 172.24.100.45 принадлежит организации, то номером сети класса В будет 172.24.0.0, а номер внутрикорпоративной подсети будет равен 172.24.100.0. Заметим, что полученные подсети не будут являться сетями класса С.

Если число нулей в маске равно М, то число доступных адресов хостов в подсети равно 2М-2. То есть два адреса в подсети использовать не рекомендуется. Один из этих адресов, у которого последние М бит равны нулю, называется адресом подсети, а второй из этих адресов у которого последние М бит равны единице называется широковещательным адресом. Так для адреса 172.24.100.45/24 адрес подсети равен 172.24.100.0, а широковещательным адрес равен 172.24.100.255. Число адресов в подсети равно $2^8$-2 =254.

Адреса класса А и В составляют около 75 процентов адресного пространства. Количество сетей классов А и В приблизительно равно 17000. Приобретение сети класса В, а тем более класса А в настоящее время весьма проблематично. Адреса класса С составляют около 12.5 процентов адресного пространства. Количество сетей класса С приблизительно равно 2.1 миллиона. К сожалению сеть класса С ограничена 254 адресами, что не отвечает нуждам больших организаций, которые не могут приобрести адреса класса А или В.

Классовая IP адресация, даже с использованием подсетей, не может удовлетворить требование по масштабируемости для Интернет сообщества.

Уже в начале 90-х годов почти все сети класса B были распределены. Добавление в Интернет новых сетей класса C приводило к значительному росту таблиц маршрутов и перегрузке маршрутизаторов. Использование бесклассовой адресации позволило в значительной мере решить возникшие проблемы [1-3].

**CIDR**

Современные маршрутизаторы используют форму IP адресации называемую безклассовой междоменной маршрутизацией (Classless Interdomain Routing (CIDR)), которая игнорирует классы. В системах, использующих классы, маршрутизатор определяет класс адреса и затем разделяет адрес на октеты сети и октеты хоста, базируясь на этом классе. В CIDR маршрутизатор использует биты маски для определения в адресе сетевой части и номера хоста. Граница разделения адреса может проходить посреди октета.

CIDR значительно улучшает масштабируемость и эффективность IP по следующим пунктам:

- гибкость;
- экономичное использование адресов в выделенном диапазоне;
- улучшенная агрегация маршрутов;
- Supernetting - комбинация непрерывных сетевых адресов в новый адрес надсети, определяемый маской.

CIDR позволяет маршрутизаторам агрегировать или суммировать информацию о маршрутах. Они делают это путём использования маски вместо классов адресов для определения сетевой части IP адреса. Это сокращает размеры таблиц маршрутов, так как используется лишь один адрес и маска для представления маршрутов ко многим подсетям.

Без CIDR и агрегации маршрутов маршрутизатор должен содержать индивидуальную информацию для всех подсетей.

Рассмотрим сеть класса A 44.0.0.0/8, в которой рассматривается 8 подсетей

Таблица 1.

| Сетевой номер | Первый октет | Второй октет | Третий октет | Четвёртый октет |
|---|---|---|---|---|
| 44.24.0.0/16 | 00101100 | 00011000 | 00000000 | 00000000 |
| 44.25.0.0/16 | 00101100 | 00011001 | 00000000 | 00000000 |
| 44.26.0.0/16 | 00101100 | 00011010 | 00000000 | 00000000 |
| 44.27.0.0/16 | 00101100 | 00011011 | 00000000 | 00000000 |
| 44.28.0.0/16 | 00101100 | 00011100 | 00000000 | 00000000 |
| 44.29.0.0/16 | 00101100 | 00011101 | 00000000 | 00000000 |
| 44.30.0.0/16 | 00101100 | 00011110 | 00000000 | 00000000 |
| 44.31.0.0/16 | 00101100 | 00011111 | 00000000 | 00000000 |

Первые два октета (16 бит) представляют адрес подсети. Так как первые 16 бит адреса каждой из этих восьми подсетей уникальны, то классовый маршрутизатор видит восемь уникальных сетей и должен создать строку в таблице маршрутов для каждой из этих подсетей.

Однако эти восемь адресов подсетей имеют общую часть: первые 13 бит одинаковы. CIDR-совместимый маршрутизатор может суммировать маршруты к этим восьми подсетям, используя общий 13-битовый префикс в адресах: 00101100 00011. Для представления этого префикса в десятичной форме дополним его справа нулями

10101100 00011000 00000000 00000000 = 172.24.0.0.

13-битовая маска подсети имеет вид

11111111 11111000 00000000 00000000 = 255.248.0.0.

Следовательно один адрес и одна маска определяет бесклассовый префикс, который суммирует маршруты к восьми подсетям: 172.24.0.0/13.

**Supernetting**

Supernetting это практика использования битовой маски для группировки нескольких классовых сетей в виде одного сетевого адреса. Supernetting и агрегирование маршрутов есть разные имена одного процесса. Однако термин supernetting чаще применяется, когда агрегируемые сети находятся под общим административным управлением. Supernetting берёт биты из сетевой порции маски, а subnetting берёт биты из порции маски, относящейся к хосту. Supernetting и агрегирование маршрутов является инверсным понятием по отношению к subnetting [1-3].

Так как сети классов A и B практически исчерпаны, то организации вынуждены запрашивать у провайдеров несколько сетей класса C. Если компания получает блок непрерывных адресов в сетях класса C, то можно использовать supernetting и все адреса в компании будут лежать в одной большей сети или надсети.

Рассмотрим компанию АБВ, которой требуется адреса для 400 хостов. При классовой адресации компания должна запросить у центральной интернет службы InterNIC сеть класа B. Если компания получит такую сеть, то десятки тысяч адресов в ней не будут использоваться. Альтернативой является получение двух сетей класса C, что даёт 254*2= 504 адреса для хостов. Недостаток этого подхода состоит в необходимости поддержки маршрутизации для двух сетей.

При бесклассовой адресной системе supernetting позволяет компании АБВ получить необходимое адресное пространство с минимальным количеством неиспользуемых адресов и без увеличения размера таблиц маршрутизации. Используя CIDR, АБВ запрашивает блок адресов у своего Интернет провайдера, а не у центральной Интернет службы InterNIC. Провайдер определяет потребности АБВ и выделяет адресное пространство из своего адресного пространства. Провайдер берёт на себя управление адресным пространством в своей внутренней безклассовой системе. Все внешние Интернет маршрутизаторы содержат только суммирующие маршруты к сети провайдера. Провайдер сам поддерживает маршруты, более специфичные для

своих клиентов, включая АБВ. Этот подход существенно уменьшает размеры таблиц маршрутов для всех маршрутизаторов в Интернет.

Пусть АБВ получил у провайдера две сети класса С, адреса в которых непрерывны: 207.21.54.0 и 207.21.55.0.

Таблица 2.

| | | | | |
|---|---|---|---|---|
| 207.21.54.0 | 110001111 | 00010101 | 00110110 | 00000000 |
| 207.21.55.0 | 110001111 | 00010101 | 00110111 | 00000000 |

Из таблицы видно, что адреса имеют общий 23-битовый префикс 11001111 00010101 0011011. Дополняя префикс справа нулями 11001111 00010101 00110110 00000000, получим надсеть с 23- битовой маской , 207.21.54.0/23.

Провайдер предоставляет сеть компании АБВ внешнему миру как сеть 207.21.54.0/23.

CIDR позволяет провайдерам эффективно распределять и суммировать непрерывные пространства IP адресов.

**VLSM**

Маска переменной длины (Variable-Length Subnet Mask (VLSM)) позволяет организации использовать более одной маски подсети внутри одного и того же сетевого адресного пространства. Реализацию VLSM часто называют «подсети на подсети».

Рассмотрим подсети, созданные путём заимствования трёх первых бит в хостовой порции адреса класса С 207.21.24.0

Таблица 3.

| Подсеть | Адрес подсети |
|---|---|
| 0 | 207.21.24.0/27 |
| 1 | 207.21.24.32/27 |

| | |
|---|---|
| 2 | 207.21.24.64/27 |
| 3 | 207.21.24.96/27 |
| 4 | 207.21.24.128/27 |
| 5 | 207.21.24.160/27 |
| 6 | 207.21.24.192/27 |
| 7 | 207.21.24.224/27 |

Мы получили восемь подсетей, каждая из который может содержать не более 30 хостов.

Каждое соединение через последовательный интерфейс требует для себя два адреса и отдельной подсети. Использование для этого любой из подсетей /27 приведёт к потере адресов. Для создания подсети из двух адресов лучше всего подходит 30-ти битовая маска. Это как раз то, что надо для последовательного соединения. Разобьём одну из подсетей 207.21.24.192/27 на восемь подсетей, используя 30-ти битовую маску.

Таблица 4

| | |
|---|---|
| 0 | 207.21.24.192/30 |
| 1 | 207.21.24.196/30 |
| 2 | 207.21.24.200/30 |
| 3 | 207.21.24.204/30 |
| 4 | 207.21.24.208/30 |
| 5 | 207.21.24.212/30 |
| 6 | 207.21.24.220/30 |
| 7 | 207.21.24.224/30 |

То есть каждую из оставшихся семи подсетей /27 можно использовать для адресации хостов в семи локальных сетях. Эти локальные сети можно связать в

глобальную сеть с помощью не более чем восьми последовательных соединений из наших восьми сетей.

Чтобы в сетях с VLSM правильно осуществлялась маршрутизация маршрутизаторы должны обмениваться информацией о масках в подсетях.

Использование CIDR и VLSM не только предотвращает пустую трату адресов, но и способствует агрегации маршрутов или суммированию. Без суммирования маршрутов Интернет перестал бы развиваться уже в конце 90-х годов. Рисунок иллюстрирует как суммирование сокращает нагрузку на маршрутизаторы.

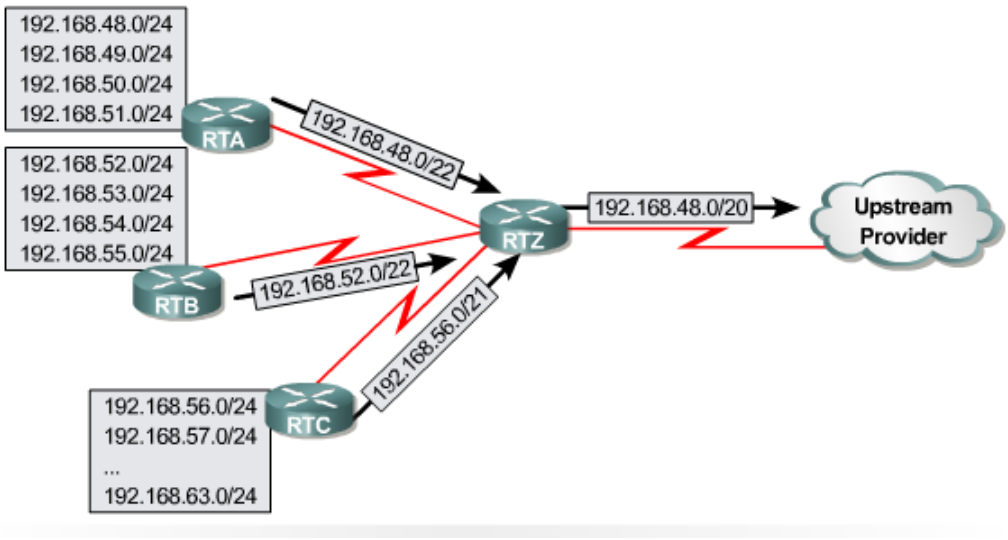



Рис. 1

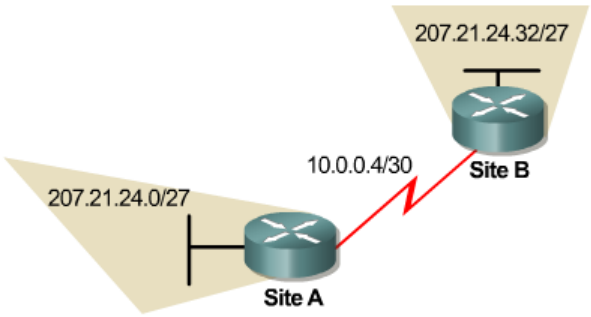



Рис. 2

Эта сложная иерархия сетей и подсетей суммируется в различных точках так, что вся сеть в целом выглядит извне как 192.168.48.0/20. Для правильной

работы суммирования маршрутов следует тщательно подходить к назначению адресов: суммируемые адреса должны иметь одинаковые префиксы.

**Разорванные подсети**

Разорванные подсети это сети из одной главной сети, разделённые сетью в совсем другом диапазоне адресов. Классовые протоколы маршрутизации RIP версии 1 и IGRP не поддерживают разрывные сети, так как маршрутизаторы не обмениваются масками подсетей. Если на рисунке 1 сайт A и сайт B работают на RIP версии 1, то сайт A будет получать от сайта B обновления маршрутной информации в сети 207.21.24.0/24, а не в сети 207.21.24.32/27.

Протоколы RIP v2 и EIGRP по умолчанию суммируют адреса на границах классов. Обычно такое суммирование желательно. Однако в случае разорванных подсетей не желательно. Отменить классовое автосуммирование можно командой no auto-summary.

**Практическая часть**

Конфигурируем VLSM и протестируем функциональность на двух протоколах маршрутизации RIP версии 1 и RIP версии 2.

Рассмотрим сеть класса C 192.168.1.0/24. Требуется выделить минимум по 25 адресов для двух локальных сетей и зарезервировать максимальное число адресов для дальнейшего развития.

Для поддержки 25 хостов в каждой подсети требуется минимум пять бит в восьмибитовой хостовой части адреса. Пять бит дадут максимум 30 возможных адресов хостов (25 = 32 - 2). Если пять бит должны быть использованы для хостов, то другие три бита в последнем октете адреса могут быть добавлены к 24-битовой маски нашей сети класса C. Следовательно, 27-битовая маска может быть использована для создания следующих 8 подсетей:

Таблица 5.

| | | | |
|---|---|---|---|
| 0 | 192.168.1.0/24 | 4 | 192.168.1.128/24 |
| 1 | 192.168.1.32/24 | 5 | 192.168.1.160/24 |
| 2 | 192.168.1.64/24 | 6 | 192.168.1.192/24 |
| 3 | 192.168.1.96/24 | 7 | 192.168.1.224/24 |

Для дальнейшей максимизации адресного пространства подсеть 192.168.1.0 /27 снова разделяется на 8 подсетей с использованием 30-битовой маски:

Таблица 6.

| | | | |
|---|---|---|---|
| 0 | 192.168.1.0/30 | 4 | 192.168.1.16/30 |
| 1 | 192.168.1.4/30 | 5 | 192.168.1.20/30 |
| 2 | 192.168.1.8/30 | 6 | 192.168.1.24/30 |
| 3 | 192.168.1.12/30 | 7 | 192.168.1.28/30 |

Эти подсети могут быть использованы для последовательных соединений точка-точка и минимизируют потери адресов, так как каждая подсеть содержит только два адреса.

Построим и настроим сеть, изображённую на рисунке 3. Для маршрутизатора Vista возьмём модель 2501, а для остальных двух модель 805. Используем коммутатор 2950.

1. Подымите интерфейсы и проверьте сеть командой **show cdp neighbors.**

2. Назначьте адреса согласно рисунку. Проверьте назначение командой **show ip interface brief.**

3. Для компьютера HostA имеем неоднозначность в назначении маршрута по умолчанию: либо на адрес 192.168.1.33 Ethernet интерфейса маршрутизатора SanJose1, либо на адрес 192.168.1.34 Ethernet интерфейса маршрутизатора SanJose2**.** Выберите произвольный, например

hostA#**ipconfig /dg 192.168.1.33**

4. На всех трёх маршрутизаторах настроим маршрутизацию по протоколу RIP с отключенным автосуммированием адресов. Это позволит прохождение информации о подсетях.

Router(config)# **router rip**

Router(config-router)#**version 2**

Router(config-router)# **no auto-summary**

Router(config-router)# **network 192.168.1.0**

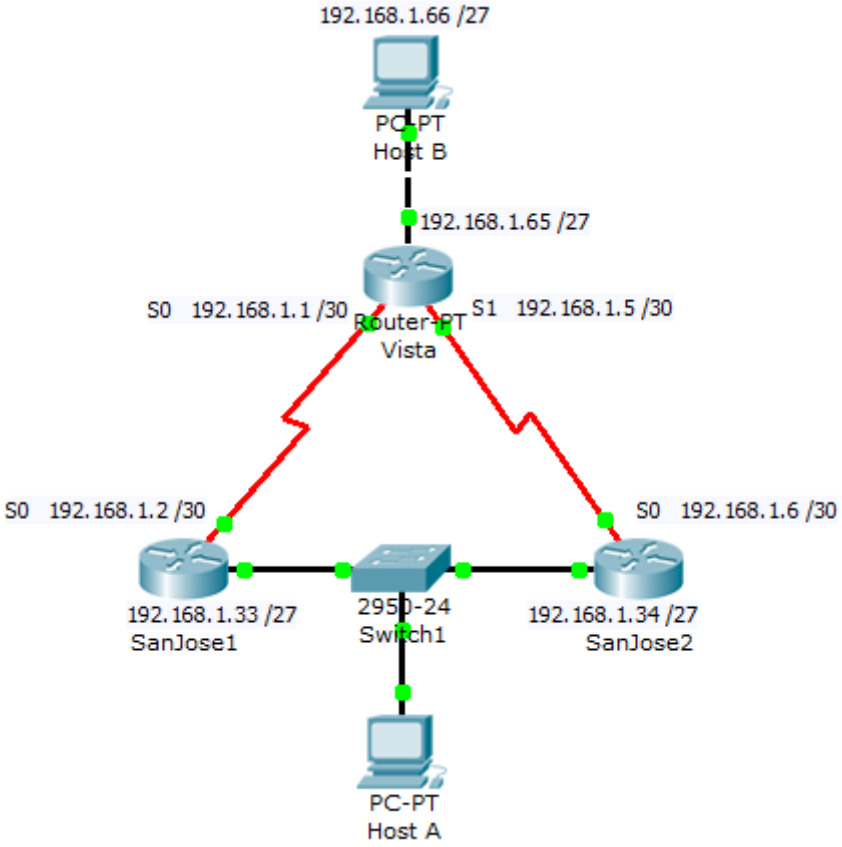



Рис.3

5. Должны увидеть, что на маршрутизаторе Vista есть маршрут на локальную сеть 192.168.1.32/27, где располагается компьютер hostA.

Vista#**show ip route**

```
C       192.168.1.0/30 is directly connected, Serial2/0
C       192.168.1.4/30 is directly connected, Serial3/0
R       192.168.1.32/27 [120/1] via 192.168.1.6, 00:00:15, Serial3/0
                        [120/1] via 192.168.1.2, 00:00:24, Serial2/0
C       192.168.1.64/27 is directly connected, FastEthernet0/0
```

Заметим, что в таблице выведен только один маршрут на сеть 192.168.1.32/27 – на адрес 192.168.1.2 через интерфейс Serial0, хотя есть ещё один маршрут на сеть 192.168.1.32/27 – через адрес 192.168.1.6 интерфейса Serial1. В реальном маршрутизаторе мы бы увидели оба эти маршрута. Симулятор выводит один

маршрут, но обменивается маршрутной информацией по обеим интерфейсам Serial0 и Serial1:

Vista#**debug ip rip**

```
RIP: sending update to 255.255.255.255 via Serial0 (192.168.1.1)
 subnet 192.168.1.64, metric 1
 subnet 192.168.1.4, metric 1
 subnet 192.168.1.32, metric 2

RIP: sending update to 255.255.255.255 via Serial1 (192.168.1.5)
 subnet 192.168.1.64, metric 1
 subnet 192.168.1.0, metric 1

RIP: sending update to 255.255.255.255 via Ethernet0 (192.168.1.65)
 subnet 192.168.1.0, metric 1
 subnet 192.168.1.4, metric 1
 subnet 192.168.1.32, metric 2

RIP: received update from 192.168.1.2 on Serial0
     192.168.1.32 in 1 hops
     192.168.1.4 in 2 hops

RIP: received update from 192.168.1.6 on Serial1
     192.168.1.32 in 1 hops
     192.168.1.0 in 2 hops
```

Vista#**no debug ip rip**

Сделайте скриншоты.

6. Должны увидеть, что на маршрутизаторе SanJose1 есть маршрут на локальную сеть 192.168.1.64/27, где располагается компьютер hostB.

```
C       192.168.1.0/30 is directly connected, Serial2/0
R       192.168.1.4/30 [120/1] via 192.168.1.34, 00:00:18, FastEthernet0/0
C       192.168.1.32/27 is directly connected, FastEthernet0/0
R       192.168.1.64/27 [120/1] via 192.168.1.1, 00:00:18, Serial2/0
```

Сделайте скриншот.

7. Должны увидеть, что на маршрутизаторе SanJose2 также есть маршрут на локальную сеть 192.168.1.64/27, где располагается компьютер hostB.

```
R       192.168.1.0/30 [120/1] via 192.168.1.33, 00:00:09, FastEthernet0/0
C       192.168.1.4/30 is directly connected, Serial3/0
C       192.168.1.32/27 is directly connected, FastEthernet0/0
R       192.168.1.64/27 [120/1] via 192.168.1.5, 00:00:26, Serial3/0
```

Сделайте скриншот.

Пропингуйте компьютер hostA из компьютера hostB и наоборот. Сделайте 2 скриншота.

Симулятор реализован так, что пример работает и без ввода команд version 2 и no auto-summary.

**Контрольные вопросы.**

1. Зачем нужна маска?
2. Что такое CIDR?
3. Что такое VLSM?
4. Как в IP адресе выделяют адрес хоста и адрес подсети?
5. Чему равно число доступных адресов в подсети?
6. По заданному числу хостов в подсети определите минимальную маску.
7. Какие формы записи маски вы знаете?
8. Почему последовательное соединение выделяют в отдельную подсеть?
9. Как CIDR и VLSM способствуют экономному использованию адресного пространства?
10. Что такое агрегация маршрутов и как она способствует уменьшению таблиц маршрутов на маршрутизаторах?
11. Что такое разорванные подсети, и какие протоколы маршрутизации их не поддерживают?

**Ход работы**

1. Изучить теоретическую и практическую часть.
2. Ответить на контрольные вопросы.
3. Выполнить в Packet Tracer практическую часть.

4. Выполните в Packet Tracer задание для получения повышенного бала.

5. Предъявите преподавателю результат выполнения пункта 11 задания для получения повышенного бала.

6. Оформите отчёт.

***Лабораторная работа №2.*** Списки управления доступом ACL.

Межсетевое экранирование с пакетной фильтрацией

Как известно [1-3], разделяют межсетевые экраны (FireWall) с пакетной фильтрацией и с сохранением состояний (экраны прикладного уровня). Рассмотрим сначала использование пакетной фильтрации.

Межсетевые экраны с фильтрацией пакетов представляют собой маршрутизаторы (например, Cisco) или работающие на сервере программы, сконфигурированные таким образом, чтобы фильтровать входящие и исходящие пакеты. Поэтому такие экраны называют иногда пакетными фильтрами. Фильтрация осуществляется путем анализа IP-адреса источника и получателя, а также портов входящих TCP- и UDP-сегментов и сравнением их со сконфигурированной таблицей правил. Эти межсетевые экраны просты в использовании, дешевы, оказывают минимальное влияние на производительность вычислительной системы. Основным недостатком является их уязвимость при подмене адресов IP (так называемый spoofing). Во всей линейке оборудования Cisco Systems пакетная фильтрация реализована с помощью так называемых списков контроля доступа (Access Control List). Однако следует отметить, что ACL могут использоваться не только для фильтрации трафика. Например, они используются для настройка NAT [1-3].

Списки доступа ACL могут быть созданы для всех сетевых протоколов, функционирующих на маршрутизаторе, например IP или IPX, и устанавливаются на интерфейсах маршрутизаторов. Запрет или разрешение сетевого трафика через интерфейс маршрутизатора реализуется на основании анализа совпадения определенных условий. Для этого списки доступа представляются в виде последовательных записей, в которых используют адреса и протоколы. Сетевые фильтры (списки доступа) создаются для входящих или исходящих пакетов на основании анализируемых параметров (адреса источника, адреса назначения, протокола и номера порта верхнего уровня), указанных в списке доступа ACL (рисунок 2.1).

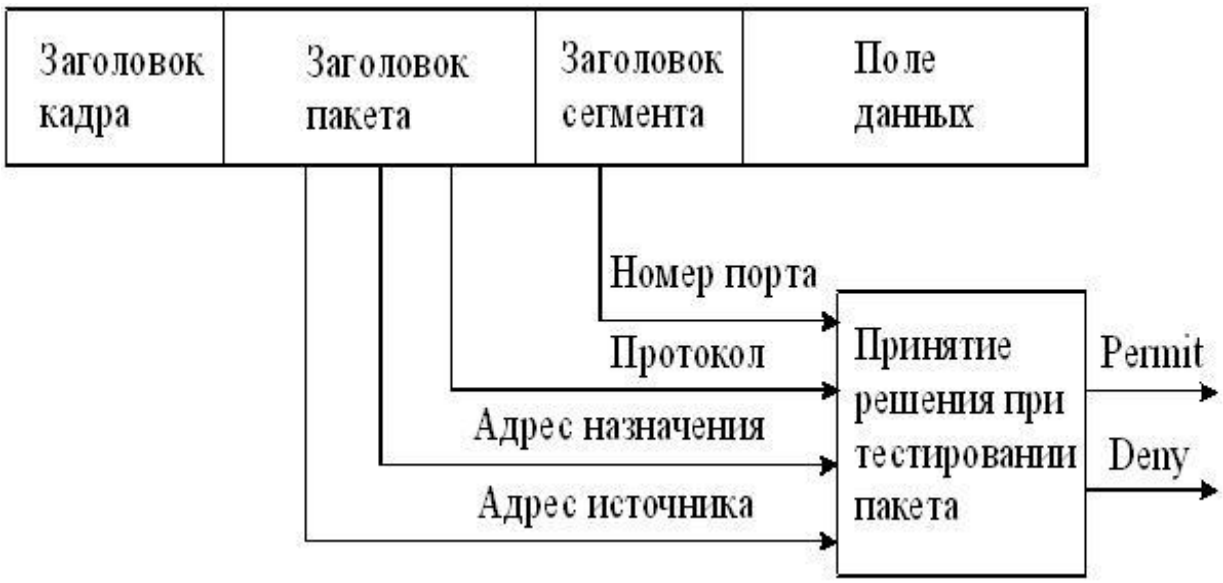



Рисунок 2.1 – Анализ заголовков пакета

Как видно из рисунка 2.1, на основе проведенного анализа служебной информации, устройство, реализующее межсетевое экранирование, принимает решение о дальнейшей передаче (permit) или о фильтрации (deny) [1-3].

Списки доступа могут быть определены для каждого установленного на интерфейсе протокола и для каждого направления сетевого трафика (исходящего и входящего). Поэтому для входящего и исходящего трафиков через интерфейс создаются отдельные списки.

Если списки доступа не формируются на маршрутизаторе, то все проходящие через маршрутизатор пакеты будут иметь доступ к сети.

Список доступа ACL составляется из утверждений (условий), которые определяют, следует ли пакеты принимать или отклонять во входных и выходных интерфейсах маршрутизатора. Программное обеспечение IOS Cisco проверяет пакет последовательно по каждому условию. Если условие, разрешающее продвижение пакета, расположено наверху списка, никакие условия, добавленные ниже его, не будут запрещать продвижение пакета. Если в списке доступа необходимы дополнительные условия, то список целиком должен быть удален и создан новый с новыми условиями.

Функционирование маршрутизатора по проверке соответствия принятого пакета требованиям списка доступа производится следующим образом. Когда кадр поступает на интерфейс, маршрутизатор проверяет IP-адрес. Если адрес назначения соответствует адресу интерфейса, то маршрутизатор извлекает (декапсулирует) из кадра пакет и проверяет его на соответствие условиям списка ACL входного интерфейса. При отсутствии запрета или отсутствии списка доступа пакет инкапсулируется в новый кадр второго уровня и отправляется интерфейсу следующего устройства.

Проверка условий (утверждений) списка доступа производится последовательно. Если текущее утверждение верно, пакет обрабатывается в соответствии с командами **permit** или **deny** списка доступа, остальная часть условий ACL не проверяется. Если все утверждения ACL неверны, то неявно заданная по умолчанию команда **deny any** (запретить все остальное) в конце списка не позволит передавать дальше по сети несоответствующие пакеты.

Существуют разные типы списков доступа: стандартные (standard ACLs), расширенные (extended ACLs) и именованные (named ACLs). Когда список доступа конфигурируется на маршрутизаторе, каждый список должен иметь уникальный идентификационный номер или уникальное имя. Номер идентифицирует тип созданного списка доступа и должен находиться в пределах определенного диапазона, заданного для этого типа списка (таблица 2.1).

Таблица 2.1 – Диапазоны идентификационных номеров ACL

| Диапазон номеров | Название списка доступа |
|---|---|
| 1-99 | IP standard access-list |
| 100-199 | IP extended access-list |
| 1300-1999 | IP standard access-list (extended range) |
| 2000-2699 | IP extended access-list (extended range) |
| 600-699 | Appletalk access-list |
| 800-899 | IPX standard access-list |
| 900-999 | IPX extended access-list |

В стандартном списке доступа для принятия решения в IP-пакете анализируется только адрес источника сообщения, чтобы фильтровать сеть (IPX-стандарт может фильтровать адреса как источника, так и назначения).

Расширенные списки доступа (Extended Access Lists) проверяют как IP-адрес источника, так и IP-адрес назначения, поле протокола в заголовке пакета сетевого уровня и номер порта в заголовке транспортного уровня.

Таким образом, для каждого протокола, для каждого направления трафика и для каждого интерфейса может быть создан свой список доступа. Исходящие фильтры не затрагивают трафик, который идет из местного маршрутизатора (например, сообщения протоколов динамической маршрутизации).

Из рекомендаций по установке списков доступа можно отметить следующее. Стандартные списки доступа рекомендуется устанавливать по возможности ближе к адресату назначения, а расширенные – ближе к источнику. Поэтому стандартные списки доступа должны блокировать устройство назначения и располагаться поближе к защищаемой сети, а расширенные списки доступа должны быть установлены близко к источнику сообщений [1-3].

Список доступа производит фильтрацию пакетов по порядку, поэтому в строках списков следует задавать условия фильтрации, начиная от специфических условий и заканчивая общими. Условия списка доступа обрабатываются последовательно от вершины списка к основанию, пока не будет найдено соответствующее условие. Если никакое условие не найдено, тогда пакет отклоняется и уничтожается, поскольку неявное условие **deny any** (запретить все остальное) присутствует неявно в конце любого списка доступа. Не удовлетворяющий списку доступа пакет протокола IP будет отклонен и уничтожен, при этом отправителю будет послано сообщение ICMP. Новые записи (линии) всегда добавляются в конце списка доступа.

Конфигурирование списков доступа производится в два этапа:

1. Создание списка доступа в режиме глобального конфигурирования.

2. Привязка списка доступа к интерфейсу в режиме детального конфигурирования интерфейса.

Формат команды создания стандартного списка доступа следующий:

**Router(config)#access-list {№} {permit / deny} {адрес источника}.**

Списки доступа могут фильтровать как трафик, входящий в маршрутизатор (in), так и трафик, исходящий из маршрутизатора (out). Направление трафика указывается при привязке списка доступа к интерфейсу. Формат команды привязки списка к интерфейсу следующий:

**Router(config-if)#{протокол} access-group {номер} {in или out}.**

После привязки списка доступа его содержимое не может быть изменено. Не удовлетворяющий администратора список доступа должен быть удален командой **no access-list** и затем создан заново.

Расширенный список доступа создается командой:

**Router(config)#access-list {№} {permit / deny} {трансп.протокол} {адр.ист} {адр.пол.} eq {№ порта или название прикладного протокола}**

Правила назначения в списке доступа номера порта (или, что тоже самое, прикладного протокола) представлены в таблице 2.2.

Таблица 2.2 – Правила назначения прикладных протоколов

| Обозначение | Действие |
|---|---|
| lt n | Все номера портов, меньшие n. |
| gt n | Все номера портов, большие n. |
| eq n | Порт n |
| neq n | Все порты, за исключением n. |
| range n m | Все порты от n до m включительно. |

Распространенные прикладные протоколы и соответствующие им стандартные номера портов приведены в таблице 2.3.

Таблица 2.3 – Номера портов некоторых прикладных протоколов

| Номер порта | Транспортный протокол | Прикладной протокол | Ключевое слово в команде access-list |
|---|---|---|---|
| 20 | TCP | FTP | data ftp_data |
| 21 | TCP | Управление сервером FTP | ftp |
| 22 | TCP | SSH | |
| 23 | TCP | Telnet | telnet |
| 25 | TCP | SMTP | Smtp |
| 53 | UDP, TCP | DNS | Domain |
| 67, 68 | UDP | DHCP | nameserver |
| 69 | UDP | TFTP | Tftp |
| 80, 8080 | TCP | HTTP (WWW) | www |
| 110 | TCP | POP3 | pop3 |
| 161 | UDP | SNMP | Snmp |

Рассмотрим пример создания стандартного списка доступа для сети, схема которой показана на рисунке 2.2. На рисунке укажем узлы, находящиеся в подсетях 192.168.0.0/24 и 192.168.1.0/24.

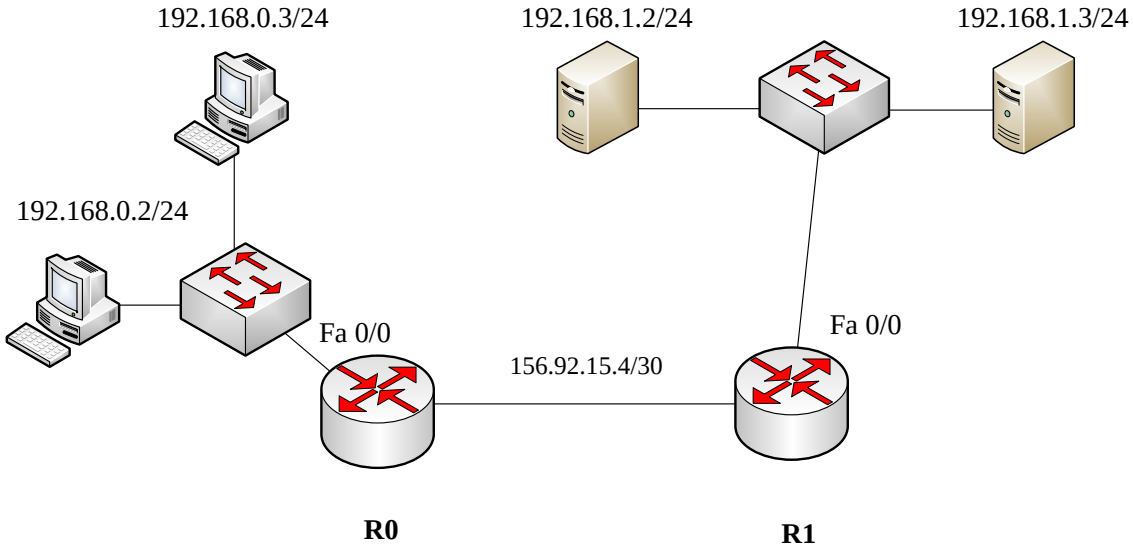



Рисунок 2.2 – Схема сети

Предположим, что к серверу, находящемуся в подсети 192.168.1.0/24 по адресу 192.168.1.2/24, доступ из подсети 192.168.0.0/24 разрешен только компьютеру 192.168.0.2/24. Это правило можно сконфигурировать с

использованием стандартного списка доступа на интерфейсе Fa 0/0 маршрутизатора R1.

Для этого в режиме глобального конфигурирования на маршрутизаторе R1 необходимо выполнить следующие команды:

**Router1(config)#access-list 10 permit 192.168.0.2**

**Router1(config)#interface fa 0/0**

**Router1(config-if)#ip access-group 10 out**

Первая команда создает на маршрутизаторе список доступа с номером 10, который разрешает (permit) передачу пакетов с адресом источника 192.168.0.2.

Вторая команда является командой перехода к конфигурированию интерфейса Fa 0/0.

Третья команда привязывает список доступа с номером 10 к интерфейсу Fa 0/0 и указывает на направление передачи – исходящее (out).

Созданный таким образом список доступа будет состоять из двух строк. Первая строка в явной форме разрешает передавать на интерфейс маршрутизатора Fa 0/0 пакеты с адресом источника 192.168.0.2. Вторая строка в неявном виде запрещает (deny any) передавать на этот интерфейс все остальные пакеты.

Проанализируем действия маршрутизатора R1 при поступлении на его внешний интерфейс пакета после создания списка доступа.

Если пакет поступил из подсети 156.92.15.4 и предназначен серверу 192.168.1.2, маршрутизатор, определив по таблице маршрутизации выходной интерфейс, передает этот пакет в буфер интерфейса Fa 0/0.

Затем анализируется список, начиная с первой строки. Если источник имеет адрес 192.168.0.2 (совпадение в первой строке списка произошло), пакет инкапсулируется в кадр Ethernet и передается серверу. Если источник имеет любой другой адрес (совпадения в первой строке списка не произошло), происходит обращение ко второй неявной строке списка (deny any), и пакет отбрасывается.

В случае, если необходимо обеспечить доступ к серверу и второго компьютера подсети 192.168.0.0/24 с адресом 192.168.0.3, команды конфигурирования будут выглядеть следующим образом:

**Router1(config)#access-list 10 permit 192.168.0.2**

**Router1(config)#access-list 10 permit 192.168.0.3**

**Router1(config)#interface fa 0/0**

**Router1(config-if)#ip access-group 10 out**

Очевидно, что список доступа теперь содержит три строки – две явные и одну неявную.

Очевидно, что рассмотренный способ конфигурирования списков доступа удобен в том случае, если доступ к какому-либо ресурсу (серверу) необходимо обеспечить небольшому количеству источников (компьютеров). Если же, например, в подсети 192.168.0.0/24 значительное количество компьютеров, такое конфигурирование становится неудобным и подверженным ошибкам, так как для каждого из них необходимо отдельно создавать строку списка.

Поэтому при создании списков доступа можно использовать wildcard маски. В этом случае в строке списка может содержаться указание на передачу или фильтрацию пакетов не с адресами конечных узлов, а с адресами сетей (подсетей), в которые они входят.

Правило использования масок в этом случае можно сформулировать следующим образом – нулевые значения разрядов маски означают требование обработки соответствующих разрядов адреса, а единичные значения разрядов маски означают игнорирование соответствующих разрядов адреса. Например, если wildcard маска имеет вид 0.0.0.0, то проверять условие необходимо для всех разрядов адреса источника прибывшего пакета. Если же маска имеет вид 0.0.0.255, то проверять условие необходимо только для первых трех байтов адреса источника.

Предположим, что доступ к тому же серверу (адрес 192.168.1.2) должны получить все компьютеры подсети 192.168.0.0/24. В этом случае на маршрутизаторе R1 необходимо выполнить команды:

**Router1(config)#access-list 10 permit 192.168.0.0 0.0.0.255**

**Router1(config)#interface fa 0/0**

**Router1(config-if)#ip access-group 10 out**

Необходимо отметить, что, если нужно разрешить какому-либо одному узлу из другой подсети (например, 192.168.2.2/24) доступ к этому же серверу, создаваемый список необходимо дополнить командой

**Router1(config)#access-list 10 permit 192.168.2.2 0.0.0.0**

или, что то же самое, командой

**Router1(config)#access-list 10 permit host 192.168.2.2**

Создание списков доступа очень похоже на создание «белых» и «черных» списков на телефоне. При создании «белого» списка принимать разрешено только вызовы от источников, номера которых внесены в «белый» список, остальные вызовы отбрасываются. При использовании «черного» списка отбрасываются только вызовы от источников, внесенных в список, остальные вызовы принимаются. Основное отличие от телефонных вызовов состоит в том, что в списки **permit** и **deny** вносятся не телефонные номера, а значения заголовков различных уровней.

Используя эту аналогию с «белыми» и «черными» списками телефона, можно отметить, что рассмотренные способы аналогичны созданию в телефоне «белых» списков – указанные в списке доступа адреса являются разрешенными, остальные – запрещенными.

В ряде случаев более удобным является использование аналогии «черного» списка – разрешено передавать данные от всех, кроме тех, кто указан в черном списке.

Предположим, что к тому же серверу необходимо обеспечить доступ всем компьютерам, кроме одного, имеющего адрес 192.168.0.15. Конфигурирование такого списка будет иметь вид:

**Router1(config)#access-list 11 deny host 192.168.0.15**

**Router1(config)#access-list 11 permit any**

**Router1(config)#interface fa 0/0**

**Router1(config-if)#ip access-group 11 out**

Напомним, что по умолчанию у создаваемых списков доступа неявно присутствует заключительная строка **deny any** – запретить все. В данном случае мы заменили эту строку на **permit any** – разрешить все. Соответственно, доступ к серверу будет разрешен всем, кроме компьютера с адресом 192.168.0.15.

Рассмотрим теперь применение расширенного списка для конфигурирования маршрутизатора R1 (рисунок 11.2), при этом должны быть выполнены следующие условия:

- компьютеру 192.168.0.2/24 необходимо предоставить доступ к web-серверу с адресом 192.168.1.2 по протоколу WWW;

- всем компьютерам подсети 192.168.0.0/24 необходимо предоставить доступ к FTP-серверу с адресом 192.168.1.3 по протоколу FTP.

Команды конфигурирования в этом случае будут выглядеть следующим образом:

**Router1(config)#access-list 110 permit tcp host 192.168.0.2 host 192.168.1.2 eq www**

**Router1(config)#access-list 110 permit tcp 192.168.0.0 0.0.0.255 host 192.168.1.3 eq ftp**

**Router1(config)#interface fa 0/0**

**Router1(config-if)#ip access-group 110 out**

Очевидно, что указанный способ аналогичен созданию «белого» списка в телефоне, так как третье неявное условие, находящееся в конце списка, блокирует все, что не разрешено.

Рассмотрим пример, когда удобнее использовать аналогию «черного» списка в телефоне.

На маршрутизаторе R1 должны быть выполнены следующие условия:

- компьютеру 192.168.0.2/24 необходимо запретить доступ к серверу с адресом 192.168.1.2 по протоколу WWW, но разрешить доступы к другим сервисам;

- всем компьютерам подсети 192.168.0.0/24 необходимо запретить доступ к серверу с адресом 192.168.1.3 по протоколу FTP, но разрешить доступ к другим сервисам.

Команды конфигурирования в этом случае будут иметь вид:

**Router1(config)#access-list 110 deny tcp host 192.168.0.2 host 192.168.1.2 eq www**

**Router1(config)#access-list 110 deny tcp 192.168.0.0 0.0.0.255 host 192.168.1.3 eq ftp**

**Router1(config)#access-list 110 permit ip any any**

**Router1(config)#interface fa 0/0**

**Router1(config-if)#ip access-group 110 out**

Запись **permit ip any any** означает, что весь остальной трафик от любого источника к любому получателю должен передаваться.

Просмотреть созданные на маршрутизаторе списки доступа можно по команде **show access-list**, а списки, настроенные на конкретных интерфейсах, командами **show ip interfaces** или **show running-config**. На рисунке 11.3 показаны настроенные списки доступа для рассмотренного здесь примера.

Списки доступа также желательно использовать и для конфигурирования удаленного доступа к устройствам (глава 9).

В настоящее время чаще используются не нумерованные, а именованные списки доступа. Удобство использования именованных списков доступа заключается прежде всего в том, что названию списка можно придать определенный смысл (INTERNET, ADMIN, FTP, и т.д.). Так как именованный список не имеет номера, который однозначно определяет его вид (таблица 11.1), при создании такого списка необходимо явно указать, какой именно список создается – стандартный или расширенный.

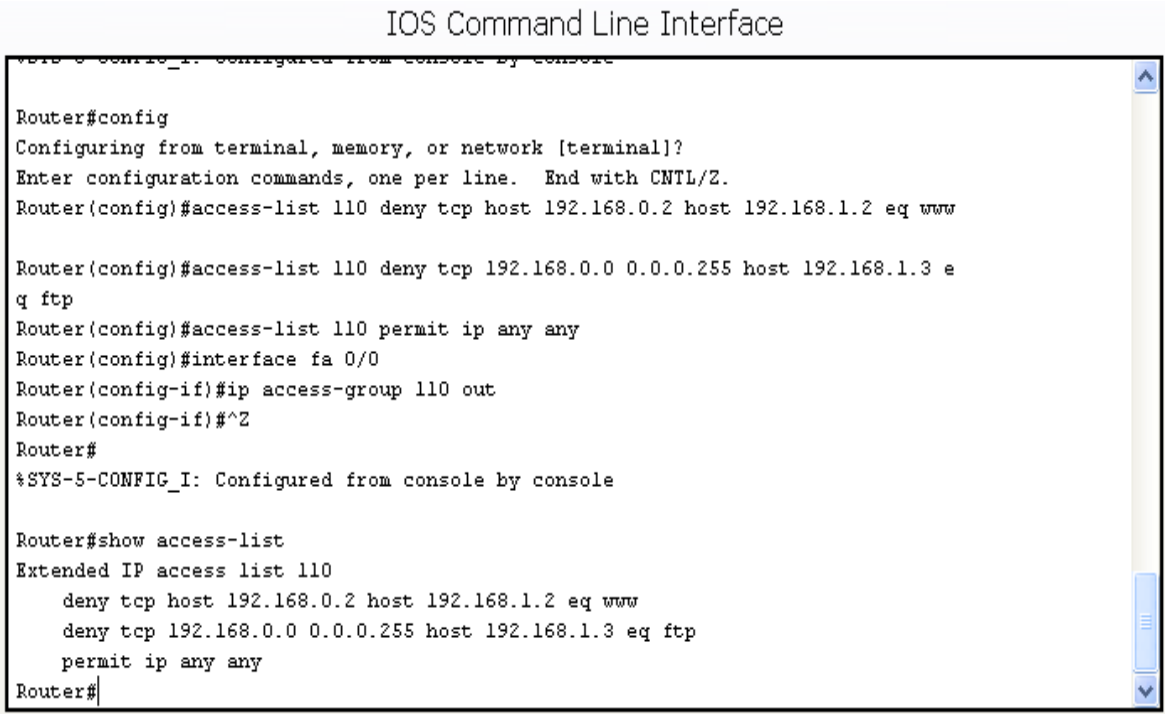



Рисунок 2.3 – Конфигурирование списка доступа и его просмотр

Команда создания именованного списка доступа имеет вид:

**ip access-list <standard/extended> <имя>**

**<правило 1>**

**<правило 2>**

**<правило n>**

Параметр **standard/extended** указывает на вид создаваемого списка, а правила прописываются аналогично нумерованным спискам.

**Межсетевое экранирование с сохранением состояний**

Такие межсетевые экраны еще называют экранами с сохранением сессий (statefull firewall). Суть заключается в том, что при запросе на установление соединения (например, TCP-сессии) маршрутизатор запоминает эту сессию и при поступлении извне пакета сверяет его со всеми текущими сессиями. Если принятый извне пакет относится к какой-либо текущей сессии, он продвигается во внутреннюю сеть, в противном случае – отбрасывается.

Для конфигурирования межсетевого экранирования на устройствах Cisco необходимо в явном виде указать, трафик каких протоколов должен отслеживаться (инспектироваться). Для этого используется команда

**ip inspect name <имя правила> <название протокола>**

Данная команда выполняется в режиме глобального конфигурирования.

Аналогично спискам доступа, созданное правило необходимо привязать к интерфейсу с указанием направления передачи:

**R1(config)#int fa0/0** – переход в режим конфигурирования интерфейса fa 0/0;

**R1(config-if)#ip inspect <имя правила> <in/out>** - привязка правила к интерфейсу с указанием направления передачи.

Необходимо отметить, что правило можно привязывать как к внутреннему, так и ко внешнему интерфейсу маршрутизатора, однако направление передачи должно соответствовать направлению запросов из внутренней сети ко внешней. Соответственно, если правило привязывается к внутреннему интерфейсу, направление передачи – входящее (in), если к внешнему – исходящее (out).

Приведем пример межсетевого экранирования для сети, показанной на рисунке 2.4. Для удобства разместим маршрутизатор R1 во внешней сети, в которой также располагаются два сервера – web-сервер и ftp-сервер. Произведем настройку всего оборудования таким образом, чтобы из внутренней сети были доступны оба сервера.

Создадим правило для инспектирования запросов к web-серверу (протокол HTTP) с именем HTTP и привяжем его к внутреннему интерфейсу fa0/0 с входящим направлением передачи (рисунок 2.5).

Следует отметить (и это очень важно!), что инспектирование трафика необходимо применять совместно со списками доступа. В нашем примере, когда списки доступа не были созданы, http-запросы, поступающие из внутренней сети, будут инспектироваться. Однако если из внешней сети также

поступит http-запрос, он пройдет во внутреннюю сеть, так как не существует списка доступа, обеспечивающего фильтрацию этого запроса.

Для проверки этого разместим во внешней сети ПК с адресом 213.80.65.4 и попробуем соединиться с сервером внутренней сети по протоколу HTTP (рисунок 2.6) [1-3].

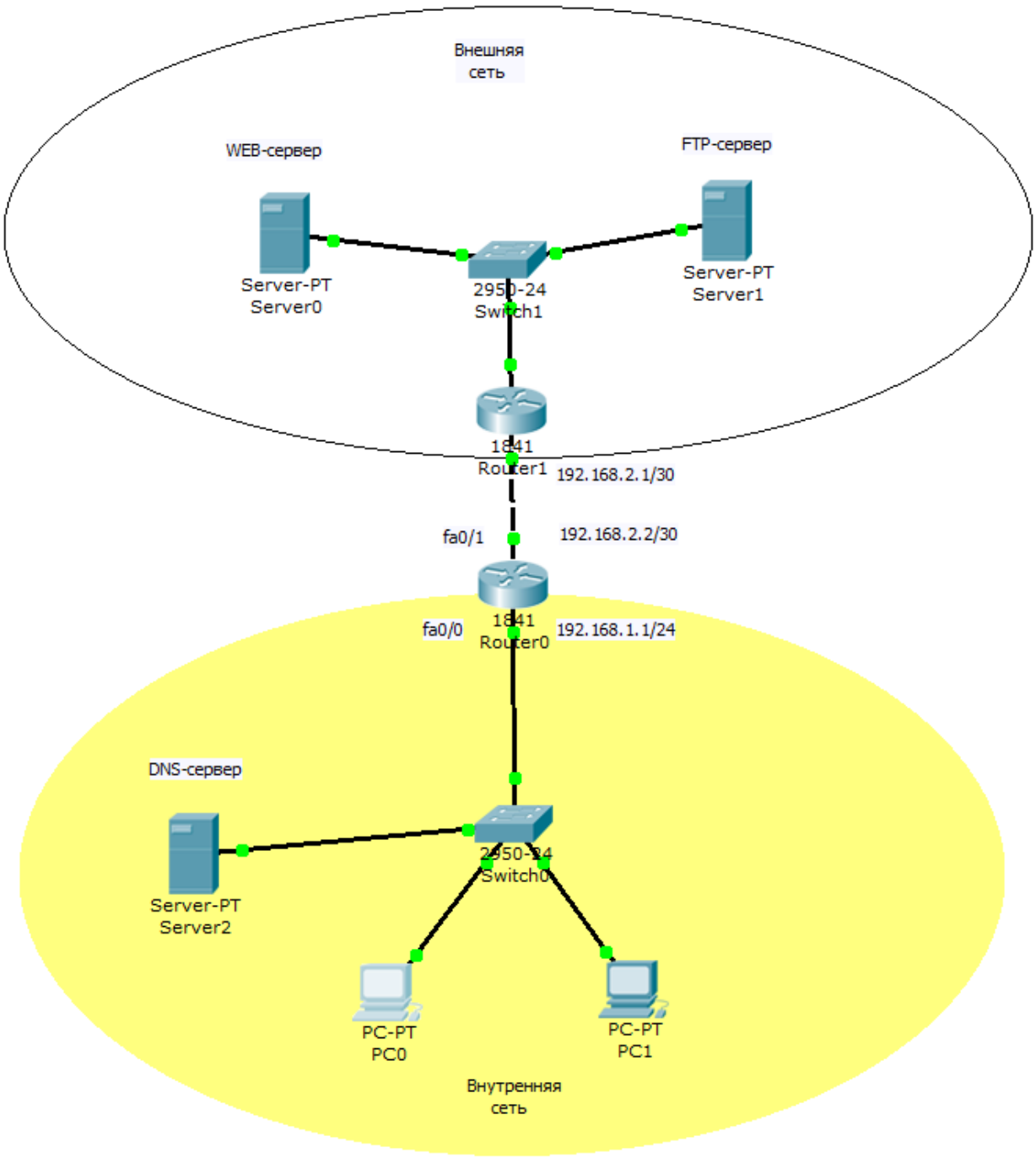



Рисунок 2.4 – Пример сети

```
Router>en
Router#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip inspect name HTTP http
Router(config)#int fa0/0
Router(config-if)#ip inspect HTTP in
Router(config-if)#
```

Рисунок 2.5 – Конфигурирование инспектирования протокола HTTP

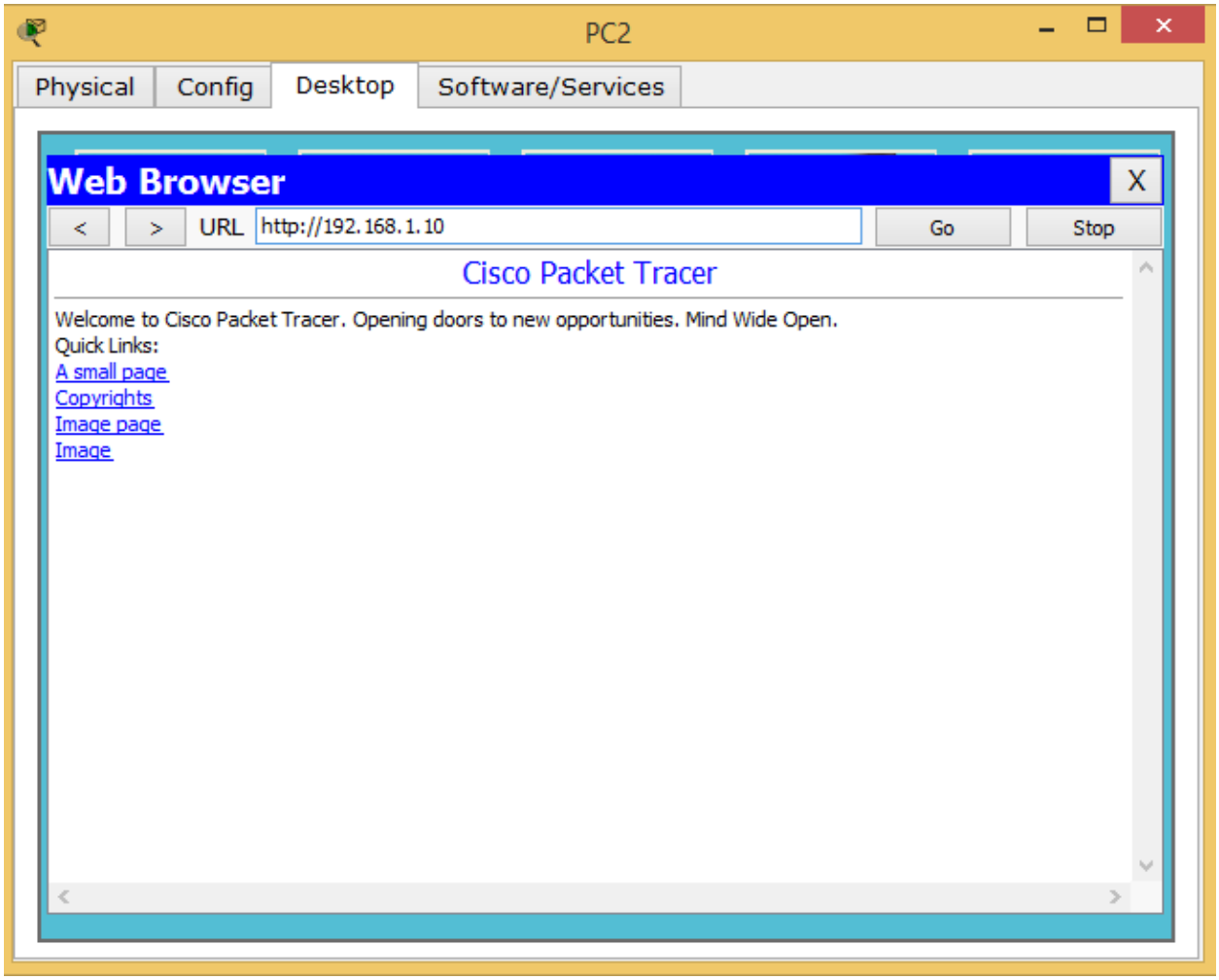



Рисунок 2.6 – Соединение с сервером внутренней сети по протоколу HTTP

В качестве внутреннего сервера мы использовали DNS-сервер с адресом 192.168.1.10/24. Как видно из рисунка 2.6, соединение прошло успешно.

Для защиты внутренней сети создадим на маршрутизаторе R0 список доступа, запрещающий передачу всех IP-пакетов, и привяжем его к внешнему интерфейсу fa0/1 с указанием входящего направления (рисунок 11.7).

```
Router>en
Router#conf
Configuring from terminal, memory, or network [terminal]?
Enter configuration commands, one per line.  End with CNTL/Z.
Router(config)#ip access-list ex FRW
Router(config-ext-nacl)#deny ip any any
Router(config-ext-nacl)#
```

Рисунок 2.7 – Создание списка доступа

Теперь снова попытаемся послать HTTP-запрос из внешней сети (рисунок 2.8).

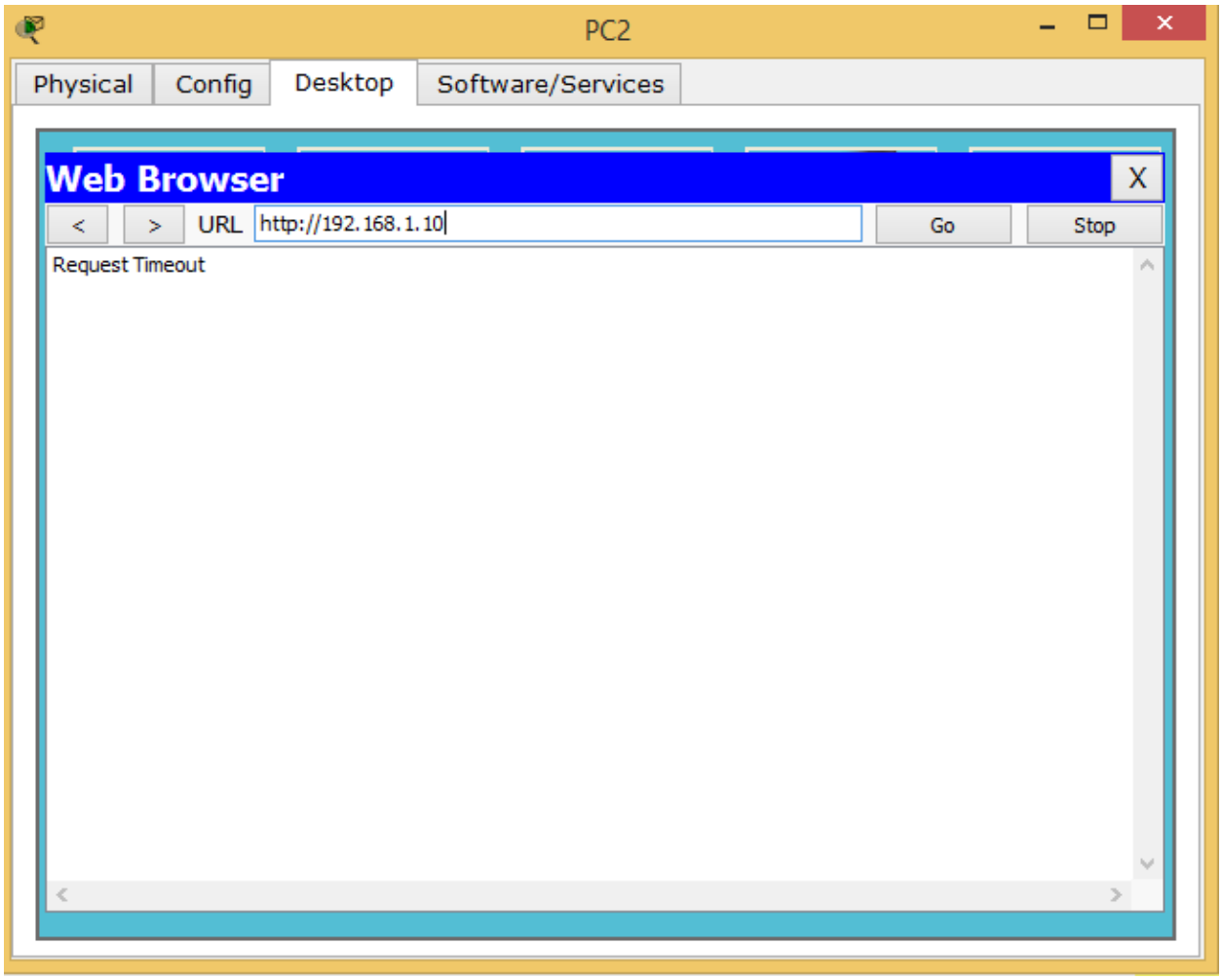



Рисунок 2.8 – Отсутствие соединения с сервером внутренней сети по протоколу HTTP

Как видно из рисунка 2.8, из внешней сети сервер не доступен. При этом из внутренней сети web-сервер остается доступным.

С учетом созданного списка доступа FRW остальные протоколы (кроме HTTP) не инспектируются. Следовательно, если послать из внутренней сети запрос по какому-либо другому протоколу, ответ получен не будет, так как его заблокирует список доступа на внешнем интерфейсе маршрутизатора. Попробуем получить из внутренней сети доступ к ftp-серверу (рисунок 2.9).

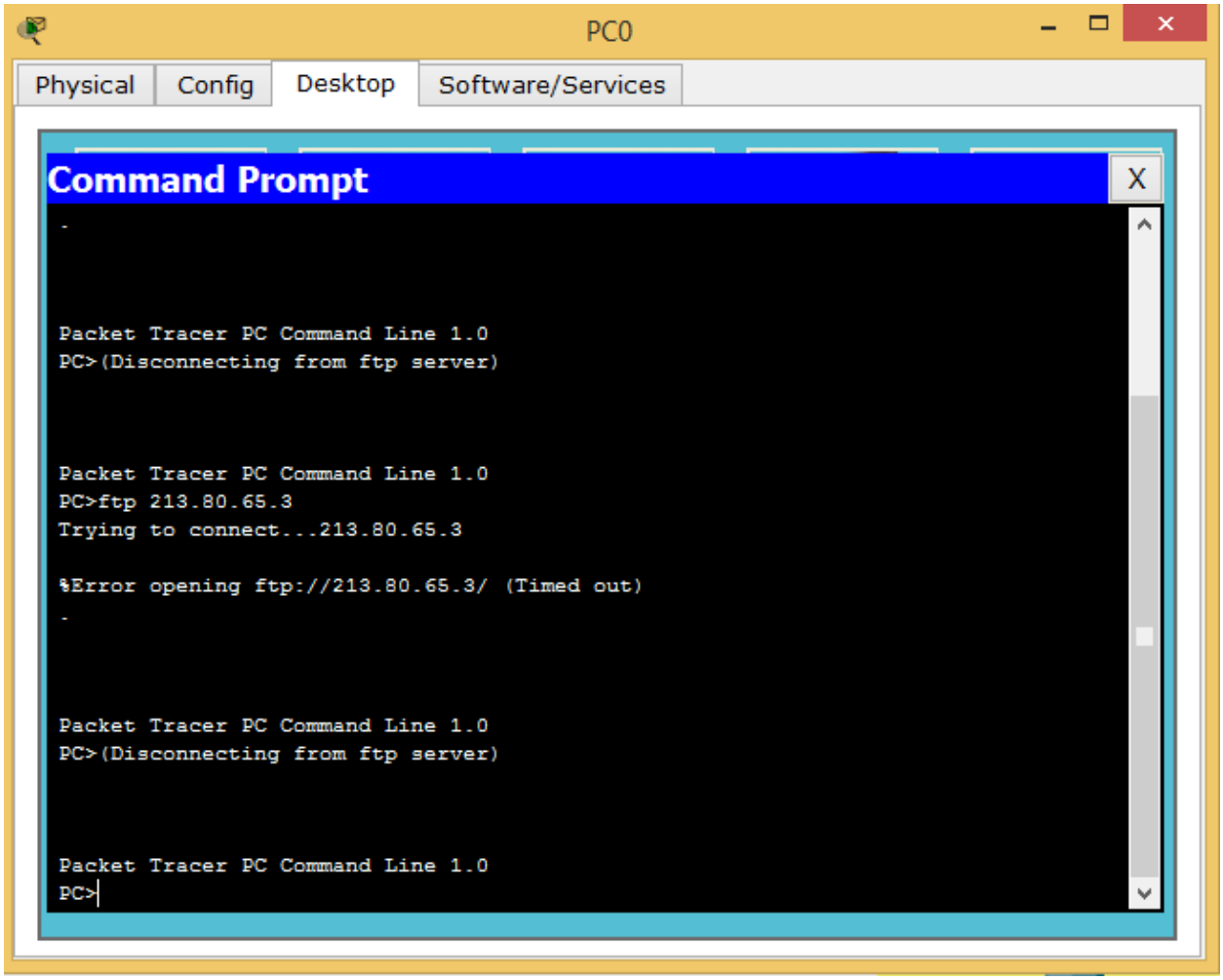



Рисунок 2.9 – Попытка соединения с ftp-сервером

Как видим, попытка не увенчалась успехом. Если же настроить инспектирование FTP-трафика, доступ к ftp-серверу будет возможен. Иллюстрировать здесь это не будем, так как Cisco Packet Tracer в силу своей ограниченной функциональности это не поддерживает. Поэтому

инспектирование разных видов трафика необходимо производить либо на реальном оборудовании, либо с использованием симулятора GNS3.

Однако Cisco Packet Tracer поддерживает инспектирование TCP-трафика. Так как и протокол HTTP, и протокол FTP использует для передачи TCP-сегменты, после настройки TCP-инспектирования доступны окажутся и http и ftp серверы (рисунки 2.10, 2.11).

```
Router(config)#ip inspect name HTTP tcp
Router(config)#
Router(config)#^Z
Router#
%SYS-5-CONFIG_I: Configured from console by console
```

Рисунок 2.10 – Настройка инспектирования TCP-трафика

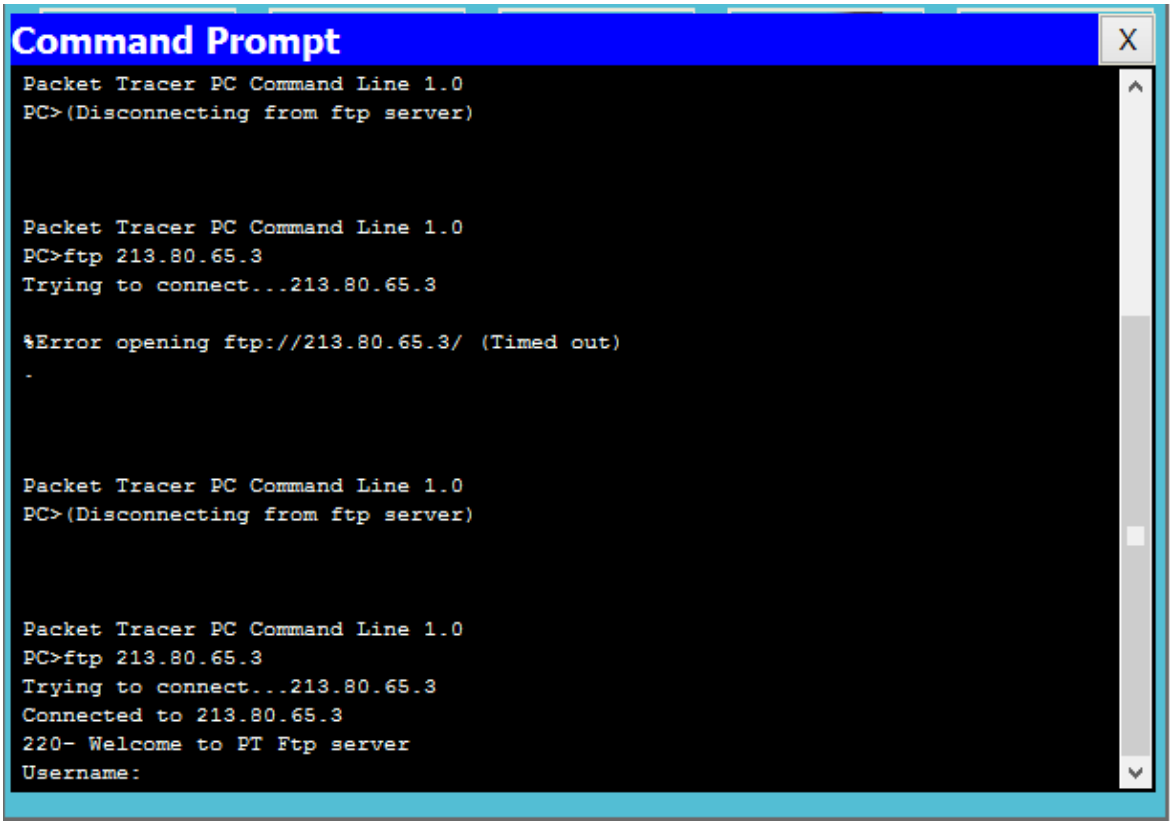



Рисунок 2.11 – Доступ к ftp-серверу после настройки инспектирования TCP-трафика

Рассмотренные настройка межсетевого экрана являются базовыми, а более тонкие настройки применяются в случае, если производится разделение

сети с различными зонами безопасности, которые рассмотрим в следующем параграфе.

**Zone-Based Policy Firewall (ZBFW)**

Zone-Based Policy Firewall (ZBFW) – относительно новое направление на маршрутизаторах под управлением операционной системы Cisco IOS для конфигурирования правил доступа между сетями. До появления этой технологии трафик фильтровался с помощью списков доступа ACL (рассмотренных в параграфе 2.1) и динамической инспекции трафика Context-Based Access Control (CBAC) (в настоящем пособии не рассматривается). И ACL и правила CBAC применяются непосредственно на физические интерфейсы, что во многих случаях не способствует масштабируемости и гибкости сетевых решений. Такая модель ограничивает степень детализации политик межсетевого экрана и вызывает путаницу правильного применения политики межсетевого экранирования, особенно в случаях, когда политика межсетевого экрана должна применяться между несколькими интерфейсами. Zone-Based Policy Firewall меняет конфигурацию межсетевого экрана от старой интерфейсной модели на более гибкую модель, основанную на зонах безопасности.

Зона безопасности состоит из набора различных интерфейсов, которые должны иметь одинаковую политику сетевой безопасности или, иначе говоря, одинаковый уровень доверия. В каждую из зон может входить один или несколько интерфейсов. После создания зон настраиваются правила для взаимодействий между зонами. Такой подход облегчает настройки правил межсетевого экрана, так как правила определяются не для отдельных интерфейсов, а для множества интерфейсов, входящих в одну зону. Кроме того в Zone-Based Policy Firewall используется язык Cisco Policy Language (CPL), который позволяет более гибко, чем в предыдущих версиях межсетевого экрана, настраивать правила фильтрации трафика.

В большинстве случаев сеть делится, как минимум, на три зоны:

- внутренняя зона, где расположены пользователи (inside);

- внешняя зона (Интернет – outside);

- демилитаризированная зона, где расположены серверы, к которым должен быть обеспечен доступ извне (dmz).

Важно, что по умолчанию весь трафик между различными зонами будет запрещен, весь трафик внутри зоны – разрешен.

На первом этапе настройки межсетевого экрана необходимо создать зоны. Зоны создаются командой, выполняемой в режиме глобального конфигурирования:

**zone security <имя зоны>**

После этого необходимо создать пары зон, между которыми будет передаваться трафик:

**zone-pair security <имя пары> source <имя зоны> destination <имя зоны>**

Необходимо отметить, что пары зон являются однонаправленными. То есть, если в нашей сети предполагается двунаправленная передача данных между внутренней и внешней зонами, необходимо создать две пары зон. Например (считаем, что внутренней зоне было присвоено имя IN, а внешней – OUT, имя пар IN_OUT и OUT_IN):

**zone-pair security IN_OUT source IN destination OUT**

**zone-pair security OUT_IN source OUT destination IN**

Как указывалось выше, по умолчанию передача трафика между созданными парами зон запрещена. Для формирования разрешенного для передачи трафика необходимо определить критерии, по которым отсортировывается нужный трафик. Для этого используется так называемый class-map (дословно – классовая карта). Class-map определяет, какой именно трафик будет инспектироваться (проходить между зонами, также проходить будут ответы на этот трафик). Фильтроваться трафик может по критериям с 3-го по 7-й уровней модели OSI (т.е. начиная от IP-адреса и заканчивая трафиком определенного приложения или сервиса прикладного уровня). Определяться трафик может списками доступа, значением CoS, типом протокола и еще рядом других параметров. Критериев может присутствовать одновременно несколько.

При этом можно указать, должен ли трафик попадать под все эти критерии (match-all) или под любой из них (match-any). Таким образом, основная задача class-map – отфильтровать необходимый тип трафика.

Для создания class-map используется команда:

**class-map type inspect match-all/match-any <имя class-map>**

После этого мы попадаем в конфигурирование созданного class-map, и далее необходимо использовать команду **match** с указанием критериев, по которым отсортировывается трафик:

- **access-group** - стандартный, расширенный или именованный список доступа, который может фильтровать трафик на основании IP адреса и порта источника и приемника. Это единственный способ выделить трафик от конкретного источника к конкретному получателю;

- **protocol** - это протоколы уровня 4 (TCP, UDP, ICMP), а также прикладные сервисы, такие как HTTP, SMTP, DNS, и т.д. Может быть указан любой известный или определяемый пользователем сервис;

- **class-map** - подчиненный класс, который предоставляет дополнительные критерии соответствия;

- **not** - определяет, что любой трафик, который не соответствует указанному сервису или протоколу, или листу доступа, будет выбран в данном class-map.

Важно, что критерии вводятся списком, и порядок обработки списка последовательный, как и у списков доступа. Например, если при конфигурировании class-map match-any мы использовали команды

**match protocol http**

**match protocol tcp**

то при обработке пакета сначала будет проверено его соответствие протоколу HTTP. Если найдено соответствие, то далее будет инспектироваться этот трафик, и следующее условие не будет проверяться. Если же команды поменять местами, то пакет сначала попадет под инспектирование трафика TCP.

Политики межсетевого экранирования определяются командой **policy-map**. Команда **policy-map** определяет действие, которое будет произведено с отфильтрованным с помощью команды **class-map** трафиком. Существует три основных действия, которые применимы к классифицированному трафику:

**Drop** – Трафик, обрабатываемый этим действием, отбрасывается и никакого уведомления на удаленный хост не высылается (в противоположность классическим листам доступа (ACL), когда высылается ICMP-сообщение Host Unreachable). Каждая карта политик имеет скрытый класс class-default, для которого сконфигурировано действие **Drop** (аналогично строке deny any any в любом списке доступа).

**Pass** – Пропускает трафик, не включая инспекцию протокола. Это действие позволяет маршрутизатору пересылать трафик из одной зоны в другую, при этом он не отслеживает состояние соединений или сессий. Это действие разрешает прохождение трафика только в одном направлении. Чтобы обратный трафик был передан, должна быть соответствующая политика и для него. Это действие полезно для таких протоколов, как IPSec ESP, IPSec AH, ISAKMP и других по своей сути безопасных протоколов с предсказуемым поведением.

**Inspect** - Включает динамическую инспекцию для трафика, который проходит от зоны источника к зоне приемника, и автоматически разрешает обратный трафик даже для сложных протоколов, таких как H.323. Например, если трафик передается из зоны IN в зону OUT, маршрутизатор поддерживает информацию о соединениях или сеансах для TCP и UDP трафика. Поэтому маршрутизатор разрешает обратный трафик из зоны OUT в зону IN в качестве ответов на запросы соединений из IN в OUT.

Формат команды:

**policy-map type inspect <имя policy-map >**

После этого мы попадаем в режим конфигурирования созданного policy-map, в котором указываем, какой именно class-map должен обрабатываться, и затем указываем необходимое действие:

**class type inspect <имя class-map>**

**inspect/pass/drop**

Теперь созданные политики необходимо применить к парам зон, которые уже были созданы ранее (пары зон можно создать и на этом шаге):

**zone-pair security <имя пары> source <имя зоны> destination <имя зоны>**

**service-policy type inspect <имя policy-map >**

Осталось в явном виде указать маршрутизатору, какие его интерфейсы относятся к какой зоне:

**interface <имя интерфейса>**

**zone-member security <имя зоны>**

Приведем пример конфигурирования межсетевого экранирования на примере сети, показанной на рисунке 11.12.

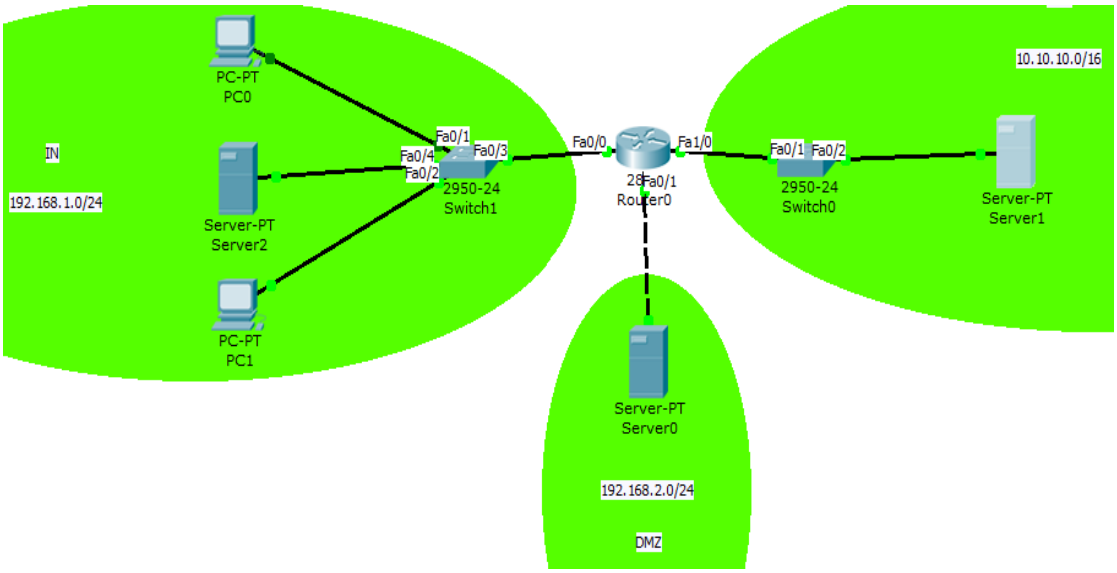



Рисунок 2.12 – Пример сети

Для простоты будем считать, что интерфейсам маршрутизатора присвоены первые адреса из адресного пространства подсетей (192.168.1.1/24 в подсети IN, 10.10.10.1/16 в подсети OUT, и т.д.)

Сначала необходимо определить зоны:

**R0(config)#zone security IN**

**R0(config)#zone security OUT**

**R0(config)#zone security DMZ**

Так как созданные зоны пока не привязаны ни к каким интерфейсам, никакого ограничения в передаче трафика после создания зон не произойдет.

Сначала сконфигурируем межсетевой экран для информационного обмена между зонами IN и OUT. Считаем, что из внутренней сети разрешены любые запросы к серверу, находящемуся во внешней сети. Таким образом, при создании class-map в этом направлении удобно использовать стандартный список доступа, разрешающий передавать данные от всех рабочих станций подсети 192.168.1.0/24:

**R0(config)#access-list permit 10 192.168.1.0 0.0.0.255**

Создаем class-map для трафика, удовлетворяющего условию списка доступа 10, присвоив самому class-map имя FROM-IN:

**R0(config)#class-map type inspect match-any FROM-IN**

и указываем, какой трафик входит в созданный class-map:

**R0(config-cmap)#match access-group 10**

Создаем policy-map с именем FROM-INSIDE (в принципе, имена class-map и policy-map могут совпадать, здесь специально выбраны разные имена):

**R0(config)#policy-map type inspect FROM-INSIDE**

указываем, какой class-map должна обрабатывать политика:

**R0(config-pmap)#class type inspect FROM-IN**

и указываем нужное действие – инспектировать:

**R0(config-pmap-c)#inspect**

Теперь обращаемся к интерфейсам для указания, к каким зонам они относятся. Одновременно можно назначить интерфейсам IP-адреса и включить их, если этого не было сделано раньше:

**R0(config)#int fa0/0**

**R0(config-if)#ip address 192.168.1.1 255.255.255.0**

**R0(config-if)#no shutdown**

**R0(config-if)#zone-member security IN**

**R0(config)#int fa1/0**

**R0(config-if)#ip address 10.10.10.1 255.255.0.0**

**R0(config-if)#no shutdown**

**R0(config-if)#zone-member security OUT**

Так как интерфейсы маршрутизатора теперь принадлежат к разным зонам, передача трафика между ними запрещена. Для разрешения передачи между ними трафика в соответствии с созданной политикой создадим зонную пару с именем IN-TO-OUT:

**R0(config)#zone-pair security IN-TO-OUT source IN destination OUT**

и применим к ней созданную политику:

**R0(config-sec-zone-pair)#service-policy type inspect FROM-INSIDE**

Заметим, что пару для обратного направления OUT-IN мы не создавали. Это связано с тем, что по условиям задачи любой трафик извне запрещен, за исключением ответов на запросы, которые поступили из внутренней сети (они инспектируются). Проверить работоспособность сконфигурированного межсетевого экрана достаточно просто – если из внутренней сети послать любой запрос (например, ping или http-запрос), то внутренний компьютер должен получить ответ (рисунок 2.13). Если же послать запрос с внешнего сервера, ответа не будет – запрос будет отброшен маршрутизатором (рисунок 2.14).

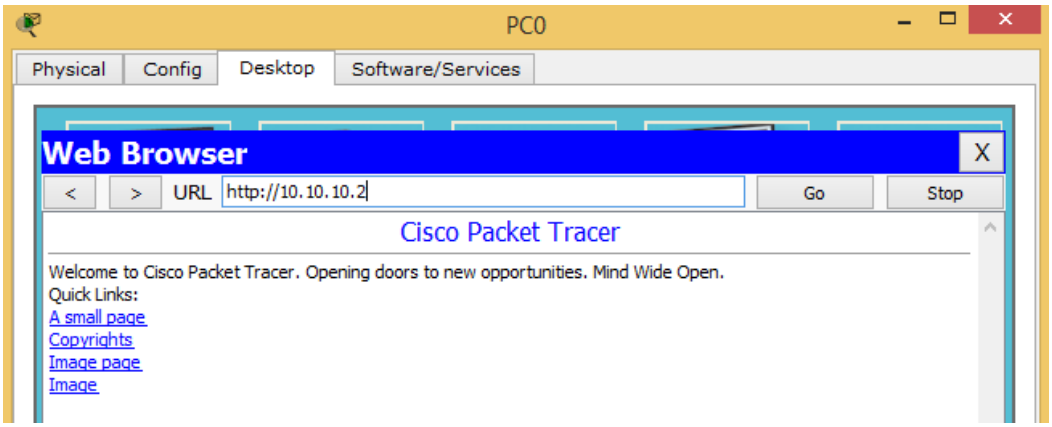



Рисунок 2.13 – Получение ответа при запросе из внутренней сети

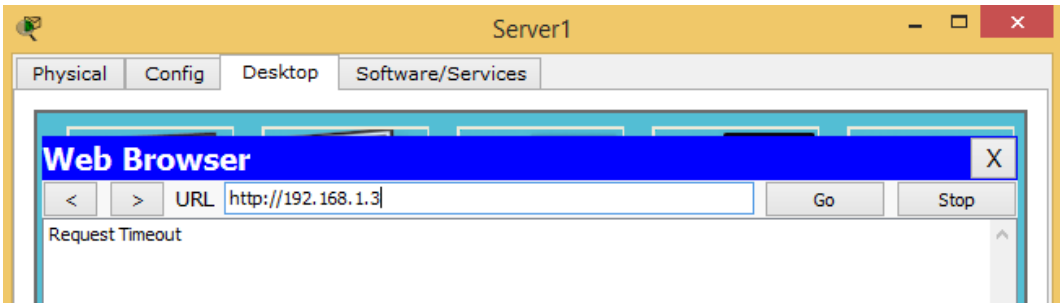



Рисунок 2.14 – Отсутствие ответа при запросе из внешней сети

Перейдем теперь к настройке демилитаризованной зоны. Здесь ситуация несколько иная – к серверам демилитаризированной должен быть обеспечен доступ как извне (например, по протоколу HTTP, если это web-сервер), так и изнутри (например, по протоколу SSH или Telnet для конфигурирования сервера). Соответственно, в этом случае необходимо создавать как минимум две пары зон – OUT-DMZ, IN-DMZ – с различными политиками. Если же предполагается наличие доступа из демилитаризованной зоны, необходимо создавать пары DMZ-OUT или DMZ-IN.

Предположим, что для нашей сети необходимо обеспечить следующие условия:

1. Доступ из внешней сети к серверу, находящемуся в DMZ, возможен только по протоколу HTTP;

2. Доступ из внутренней сети к серверу, находящемуся в DMZ, возможен по протоколам SSH, Telnet, HTTP, и только с тех IP-адресов, которые относятся к адресному пространству внутренней сети 192.168.1.0/24.

Привяжем интерфейс маршрутизатора fa0/1 к созданной ранее зоне DMZ, одновременно назначим ему IP-адрес и включим его:

**R0(config)#int fa0/1**

**R0(config-if)#ip address 192.168.2.1 255.255.255.0**

**R0(config-if)#no shutdown**

**R0(config-if)#zone-member security DMZ**

Очевидно, что для передачи трафика из внутренней сети к серверу DMZ необходимо создать несколько class-map, так как в каждом из них одновременно должно выполняться как минимум два условия:

- источником трафика является внутренняя сеть и используется протокол SSH;

- источником трафика является внутренняя сеть и используется протокол Telnet;

- источником трафика является внутренняя сеть и используется протокол HTTP.

Создадим class-map для передачи трафика из внутренней сети к серверу DMZ по протоколу SSH с именем IN-DMZ-SSH:

**R0(config)#class-map type inspect match-all IN-DMZ-SSH**

**R0(config-cmap)#match access-group 10**

**R0(config-cmap)#match protocol ssh**

Создадим class-map для передачи трафика из внутренней сети к серверу DMZ по протоколу Telnet с именем IN-DMZ-TLN:

**R0(config)#class-map type inspect match-all IN-DMZ-TLN**

**R0(config-cmap)#match access-group 10**

**R0(config-cmap)#match protocol telnet**

Создадим class-map для передачи трафика из внутренней сети к серверу DMZ по протоколу HTTP с именем IN-DMZ-HTTP:

**R0(config)#class-map type inspect match-all IN-DMZ-HTTP**

**R0(config-cmap)#match access-group 10**

**R0(config-cmap)#match protocol http**

Создадим policy-map с именем IN-DMZ, в которой укажем на необходимость инспектирования трафика, удовлетворяющего созданным class-map:

**R0(config)#policy-map type inspect IN-DMZ**

**R0(config-pmap)#class type inspect IN-DMZ-SSH**

**R0(config-pmap-c)#inspect**

**R0(config-pmap-c)#exit**

**R0(config-pmap)#class type inspect IN-DMZ-TLN**

**R0(config-pmap-c)#inspect**

**R0(config-pmap-c)#exit**

**R0(config-pmap)#class type inspect IN-DMZ-HTTP**

**R0(config-pmap-c)#inspect**

**R0(config-pmap-c)#exit**

**R0(config-pmap)#**

Для доступа к серверу DMZ из внешней сети должен использоваться только протокол HTTP, поэтому создадим один class-map с именем OUT-DMZ:

**R0(config)#class-map type inspect match-any OUT-DMZ**

**R0(config-cmap)#match protocol http**

Создадим policy-map с таким же названием и с указанием инспектировать трафик:

**R0(config)#policy-map type inspect OUT-DMZ**

**R0(config-pmap)#class type inspect OUT-DMZ**

**R0(config-pmap-c)#inspect**

Осталось создать пары зон и применить к ним созданные политики. Пара IN-TO-DMZ:

**R0(config)#zone-pair security IN-TO-DMZ source IN destination DMZ**

**R0(config-sec-zone-pair)#service-policy type inspect IN-DMZ**

Пара зон OUT-TO-DMZ:

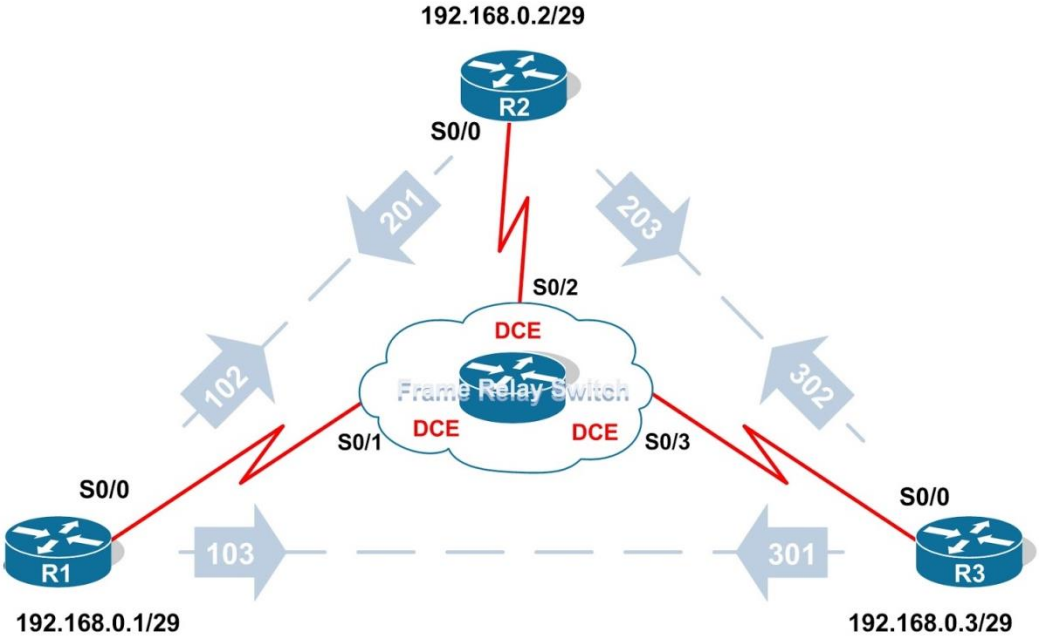
**R0(config)#zone-pair security OUT-TO-DMZ source OUT destination DMZ**

**R0(config-sec-zone-pair)#service-policy type inspect OUT-DMZ**

***Лабораторная работа №3.*** Удалённый доступ. Frame Relay.

Данная лабораторная работа может быть выполнена на реальном оборудовании или в **Cisco Packet Tracer**. Все необходимые действия указаны по порядку их выполнения. Для начала выполнения лабораторной работы необходимо соединить физическую сеть в соответствии со схемой сети или построить соответствующий проект в **Cisco Packet Tracer**. Сразу после схемы сети в таблице указана схема адресация, которую нужно применять только тогда, когда это будет явно указано в тексте лабораторной работы [1-3].

**Используемая топология**




План адресации.

| nput Interface | Input DLCI | Output Interface | Output DLCI |
|---|---|---|---|
| Serial0/1 | 102 | Serial0/2 | 201 |
| Serial0/1 | 103 | Serial0/3 | 301 |
| Serial0/2 | 201 | Serial0/1 | 102 |
| Serial0/2 | 203 | Serial0/3 | 302 |
| Serial0/3 | 301 | Serial0/1 | 103 |
| Serial0/3 | 302 | Serial0/2 | 203 |

| Device | Interface | IP Address | Subnet Mask | Default Gateway |
|---|---|---|---|---|
| R1 | S0/0 | 192.168.0.1 | 255.255.255.248 | N/A |
| R2 | S0/0 | 192.168.0.2 | 255.255.255.248 | N/A |
| R3 | S0/0 | 192.168.0.6 | 255.255.255.248 | N/A |

1. Базовая конфигурация оборудования

— Настроить hostname на маршрутизаторах.

— Отключить DNS lookup.

— Установить пароль для EXEC mode

— Настроить message-of-the-day banner.

— Установить пароль для console

**2. Конфигурация коммутатора Frame Relay и создание PVC**

· Активировать Frame Relay коммутацию на маршрутизаторе

FrSw(config)#**frame-relay switching**

· Изменить тактовую частоту интерфейса, а также тип используемой инкапсуляции

FrSw(config)#interface serial 0/0

FrSw(config-if)#clock rate 64000

FrSw(config-if)#encapsulation frame-relay

· Изменить тип Frame Relay интерфейса

FrSw(config-if)#frame-relay intf-type dce

Тип Frame Relay интерфейса и физического интерфейса не обязательно должны совпадать. Физический DTE– интерфейс может быть сконфигурирован как логический Frame Relay DCE-интерфейс.

· Отключение протокола Inverse ARP на данном интерфейсе

FrSw(config-if)#no frame-relay inverse-arp

Дальнейшие условия лабораторной работы требуют отключения данного протокола

· Настроить PVC коммутацию для данного интерфейса в соответствии с таблицей [1-3]

FrSw(config-if)#frame-relay route 102 interface Serial0/2 201

FrSw(config-if)#frame-relay route 103 interface Serial0/3 301

· Активировать интерфейс

FrSw(config-if)#no shutdown

· Повторить вышеперечисленные этапы конфигурирования для интерфейсов Serial 0/2 и Serial 0/3

FrSw(config)#interface Serial0/2

FrSw(config-if)#encapsulation frame-relay

FrSw(config-if)#clock rate 64000

FrSw(config-if)#frame-relay intf-type dce

FrSw(config-if)#no frame-relay inverse-arp

FrSw(config-if)#frame-relay route 201 interface Serial0/1 102

FrSw(config-if)#frame-relay route 203 interface Serial0/3 302

FrSw(config-if)#no shutdown

FrSw(config)#interface Serial0/3

FrSw(config-if)#encapsulation frame-relay

FrSw(config-if)#clock rate 64000

FrSw(config-if)#frame-relay intf-type dce

FrSw(config-if)#no frame-relay inverse-arp

FrSw(config-if)#frame-relay route 301 interface Serial0/1 103

FrSw(config-if)#frame-relay route 302 interface Serial0/2 203

FrSw(config-if)#no shutdown

· Проверить выполненную конфигурацию коммутатора Frame Relay командой **show frame-relay route**

FrSw#show frame-relay route

Input Intf Input Dlci Output Intf Output Dlci Status

| | | | | |
|---|---|---|---|---|
| Serial0/1 | 102 | Serial0/2 | 201 | **inactive** |
| Serial0/1 | 103 | Serial0/3 | 301 | **inactive** |
| Serial0/2 | 201 | Serial0/1 | 102 | **inactive** |
| Serial0/2 | 203 | Serial0/3 | 302 | **inactive** |
| Serial0/3 | 301 | Serial0/1 | 103 | **inactive** |
| Serial0/3 | 302 | Serial0/2 | 203 | **inactive** |

Созданные PVC на данном этапе не имеют сконфигурированных конечных узлов, поэтому в выводе команды **show frame-relay route** имеют статус **inactive.**

**3. Настроить маршрутизаторы для подключения по технологии Frame Relay**

· Установить для интерфейса **Serial 0/0** инкапсуляцию **Frame Relay** и IP-адрес

R1(config)#interface serial 0/0

R1(config-if)#encapsulation frame-relay

R1(config-if)#ip address 192.168.0.1 255.255.255.248

· Сконфигурировать статическую привязку IP-адресов к соответствующим DLCI

R1(config-if)#frame-relay map ip 192.168.0.2 102 **broadcast**

R1(config-if)#frame-relay map ip 192.168.0.3 103 **broadcast**

Использование параметра **broadcast** обеспечивает необходимую функциональность для протоколов динамической маршрутизации, использующих групповую или широковещательную отправку данных.

· Активировать интерфейс

R1(config-if)#no shutdown

· Повторить вышеперечисленные этапы конфигурирования для маршрутизаторов **R2** и **R3**

R2(config)#interface serial 0/0

R2(config-if)#encapsulation frame-relay

R2(config-if)#ip address 192.168.0.2 255.255.255.248

R2(config-if)#frame-relay map ip 192.168.0.1 201 broadcast

R2(config-if)#frame-relay map ip 192.168.0.3 203 broadcast

R2(config-if)#no shutdown

R3(config)#interface serial 0/0

R3(config-if)#encapsulation frame-relay

R3(config-if)#ip address 192.168.0.3 255.255.255.248

R3(config-if)#frame-relay map ip 192.168.0.1 301 broadcast

R3(config-if)#frame-relay map ip 192.168.0.2 302 broadcast

R3(config-if)#no shutdown

**4. Проверить правильность выполненных этапов конфигурирования**

· Выполнить команду ping на **R1** для **R2 и R3**

R1#ping 192.168.0.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.2, timeout is 2 seconds:

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/58/124 ms

R1#ping 192.168.0.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.3, timeout is 2 seconds:

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/38/48 ms

· Выполнить команду ping на **R2** для **R1 и R3**

R2#ping 192.168.0.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.1, timeout is 2 seconds:

Success rate is 100 percent (5/5), round-trip min/avg/max = 32/48/72 ms

R2#ping 192.168.0.3

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.3, timeout is 2 seconds:

Success rate is 100 percent (5/5), round-trip min/avg/max = 4/29/44 ms

· Выполнить команду ping на **R3** для **R1 и R2**

R3#ping 192.168.0.1

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.1, timeout is 2 seconds:

Success rate is 100 percent (5/5), round-trip min/avg/max = 28/36/52 ms

R3#ping 192.168.0.2

Type escape sequence to abort.

Sending 5, 100-byte ICMP Echos to 192.168.0.2, timeout is 2 seconds:

Success rate is 100 percent (5/5), round-trip min/avg/max = 8/28/40 ms

· Вывести информацию по конфигурированным PVC командой **show frame-relay pvc**

**R1#show frame-relay pvc**

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

| | Active | Inactive | Deleted | Static |
|---|---|---|---|---|
| Local | **2** | 0 | 0 | 0 |
| Switched | 0 | 0 | 0 | 0 |
| Unused | 0 | 0 | 0 | 0 |

DLCI = **102**, DLCI USAGE = LOCAL, **PVC STATUS = ACTIVE**, INTERFACE = Serial0/0

input pkts 10 output pkts 10 in bytes 1040

out bytes 1040 dropped pkts 0 in pkts dropped 0

out pkts dropped 0 out bytes dropped 0

in FECN pkts 0 in BECN pkts 0 out FECN pkts 0

out BECN pkts 0 in DE pkts 0 out DE pkts 0

out bcast pkts 0 out bcast bytes 0

5 minute input rate 0 bits/sec, 0 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec

pvc create time 01:43:05, last time pvc status changed 00:56:15

DLCI = **103**, DLCI USAGE = LOCAL, **PVC STATUS** = ACTIVE, INTERFACE = Serial0/0

input pkts 10 output pkts 10 in bytes 1040
out bytes 1040 dropped pkts 0 in pkts dropped 0
out pkts dropped 0 out bytes dropped 0
in FECN pkts 0 in BECN pkts 0 out FECN pkts 0
out BECN pkts 0 in DE pkts 0 out DE pkts 0
out bcast pkts 0 out bcast bytes 0
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec
pvc create time 01:43:10, last time pvc status changed 00:19:20

**R2#show frame-relay pvc**

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

| | Active | Inactive | Deleted | Static |
|---|---|---|---|---|
| Local | **2** | 0 | 0 | 0 |
| Switched | 0 | 0 | 0 | 0 |
| Unused | 0 | 0 | 0 | 0 |

DLCI = **201**, DLCI USAGE = LOCAL, **PVC STATUS = ACTIVE**, INTERFACE = Serial0/0

input pkts 10 output pkts 10 in bytes 1040
out bytes 1040 dropped pkts 0 in pkts dropped 0
out pkts dropped 0 out bytes dropped 0
in FECN pkts 0 in BECN pkts 0 out FECN pkts 0
out BECN pkts 0 in DE pkts 0 out DE pkts 0
out bcast pkts 0 out bcast bytes 0
5 minute input rate 0 bits/sec, 0 packets/sec
5 minute output rate 0 bits/sec, 0 packets/sec

pvc create time 01:01:13, last time pvc status changed 01:01:13

DLCI = **203**, DLCI USAGE = LOCAL, **PVC STATUS = ACTIVE**, INTERFACE = Serial0/0

input pkts 10 output pkts 10 in bytes 1040

out bytes 1040 dropped pkts 0 in pkts dropped 0

out pkts dropped 0 out bytes dropped 0

in FECN pkts 0 in BECN pkts 0 out FECN pkts 0

out BECN pkts 0 in DE pkts 0 out DE pkts 0

out bcast pkts 0 out bcast bytes 0

5 minute input rate 0 bits/sec, 0 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec

pvc create time 01:01:17, last time pvc status changed 00:23:37

**R3#show frame-relay pvc**

PVC Statistics for interface Serial0/0 (Frame Relay DTE)

| | Active | Inactive | Deleted | Static |
|---|---|---|---|---|
| Local | **2** | 0 | 0 | 0 |
| Switched | 0 | 0 | 0 | 0 |
| Unused | 0 | 0 | 0 | 0 |

DLCI = **301**, DLCI USAGE = LOCAL, **PVC STATUS = ACTIVE**, INTERFACE = Serial0/0

input pkts 10 output pkts 10 in bytes 1040

out bytes 1040 dropped pkts 0 in pkts dropped 0

out pkts dropped 0 out bytes dropped 0

in FECN pkts 0 in BECN pkts 0 out FECN pkts 0

out BECN pkts 0 in DE pkts 0 out DE pkts 0

out bcast pkts 0 out bcast bytes 0

5 minute input rate 0 bits/sec, 0 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec

pvc create time 00:25:49, last time pvc status changed 00:25:49

DLCI = **302**, DLCI USAGE = LOCAL, **PVC STATUS = ACTIVE**, INTERFACE = Serial0/0

input pkts 10 output pkts 10 in bytes 1040

out bytes 1040 dropped pkts 0 in pkts dropped 0

out pkts dropped 0 out bytes dropped 0

in FECN pkts 0 in BECN pkts 0 out FECN pkts 0

out BECN pkts 0 in DE pkts 0 out DE pkts 0

out bcast pkts 0 out bcast bytes 0

5 minute input rate 0 bits/sec, 0 packets/sec

5 minute output rate 0 bits/sec, 0 packets/sec

pvc create time 00:25:52, last time pvc status changed 00:25:52

· Вывести информацию по статическим привязкам IP- адресов к соответствующим DLCI командой **show frame-relay map**

**R1#show frame-relay map**

Serial0/0 (up): ip **192.168.0.2** dlci **102**(0x66,0x1860), static,

broadcast,

CISCO, status defined, **active**

Serial0/0 (up): ip **192.168.0.3** dlci **103**(0x67,0x1870), static,

broadcast,

CISCO, status defined, **active**

**R2#show frame-relay map**

Serial0/0 (up): ip **192.168.0.1** dlci **201**(0xC9,0x3090), static,

broadcast,

CISCO, status defined, **active**

Serial0/0 (up): ip **192.168.0.3** dlci **203**(0xCB,0x30B0), static,

broadcast,

CISCO, status defined, **active**

**R3#show frame-relay map**

Serial0/0 (up): ip **192.168.0.1** dlci **301**(0x12D,0x48D0), static,

broadcast,

CISCO, status defined, **active**

Serial0/0 (up): ip **192.168.0.2** dlci **302**(0x12E,0x48E0), static,

broadcast,

CISCO, status defined, **active**

***Лабораторная робота №4.*** Виртуальные локальные сети VLAN.

***Цель лабораторной работы***: Исследовать возможности распределения широковещательных доменов посредством создания VLAN.

**Теоритическая часть.**

Режимы конфигурирования коммутаторов.

Коммутаторы Cisco могут находиться в одном из режимов, представленных в таблице 4.1 (представлены оригинальные названия на английском языке) [1-3].

Таблица 4.1 – Режимы конфигурирования

| Название режима | Приглашение (Prompt) | Описание |
|---|---|---|
| User EXEC Mode | Switch> | Пользовательский режим |
| Privileged EXEC Mode | Switch # | Привилегированный режим |
| Global Configuration Mode | Switch (config)# | Режим глобального конфигурирования |

В таблице 4.1 в первом столбце представлены названия режимов, во втором – приглашение, отображаемое в командной строке, в третьем – описание.

Пользовательский режим (user mode) используется для просмотра состояния устройства, а также для перехода в привилегированный режим (privileged mode). Никакие изменения в конфигурационном файле, в том числе удаления и сохранения текущей конфигурации, в пользовательском режиме производиться не могут. В этом режиме доступны только некоторые команды **show**, т. е. команды просмотра состояния устройства [1-3].

Для перехода в привилегированный режим необходимо выполнить команду

**enable**

Следует отметить, что оборудование Cisco допускает сокращенный ввод команд, если невозможно ее двоякое толкование. Например, вместо enable можно набрать en, и эта команда будет понятна коммутатору, так как никакая другая команда не начинается с сочетания символов en.

Выполнение каждой команды начинается после нажатия клавиши Enter. Для повтора ранее введенных команд можно использовать клавишу ↑.

В привилегированном режиме доступны все команды **show**, возможно удаление конфигурации и сохранение конфигурационного файла в памяти NVRAM. Возврат в пользовательский режим производится командой **disable** или **exit**:

**Switch#exit**

Команда exit позволяет вернуться на один уровень вверх. Например, после выполнения команды в режиме глобального конфигурирования коммутатор переходит в привилегированный режим. Если необходимо из любого состояния устройства выйти сразу в пользовательский режим, используется комбинация CTRL-Z.

В глобальном режиме производятся изменения, которые затрагивают коммутатор в целом, поэтому этот режим и называется global configuration mode (режим глобального конфигурирования). Например, в нем можно устанавливать имя коммутатора командой hostname. Имя коммутатора не имеет значения в сети, оно удобно при конфигурировании. Пример:

**Switch(config)#hostname Subnet_A**

**Subnet_A(config)#**

В режиме глобального конфигурирования на коммутатор можно устанавливать пароли. Существует несколько видов паролей для обеспечения защиты устройств Cisco. Первые два пароля, enable secret и enable password, используются для обеспечения авторизованного входа в привилегированный режим. На коммутаторе устанавливается один (или оба) из этих паролей. После установки пароля система запрашивает его у пользователя, когда вводится

команда enable. Формат команд установки паролей на коммутатор и менем Cisco_A cisco и cisco1 для входа в привилегированный режим приведен ниже:

**Switch_A(config)#enable secret cisco**

**Switch_A(config)#enable password cisco1**

Пароль enable secret криптографируется по умолчанию, поэтому является более строгим. Если установлены оба пароля – enable secret и enable password, – то в приведенном примере система будет реагировать на пароль cisco. Пароль enable password по умолчанию не криптографируется, поэтому его можно посмотреть, например, по команде show running-configuration (сокращенно sh run), которая выполняется из привилегированного режима.

Изменение и создание конфигурации коммутатора Cisco возможно в режиме глобального конфигурирования, вход в который реализуется из привилегированного по команде configure terminal (сокращенно – conf), которая вводит устройство в глобальный режим и позволяет изменять текущую конфигурацию (running-config). При этом приглашение изменяет вид на Switch(config)#:

**Switch >ena**

**Switch #conf t**

**Enter configuration commands, one per line. End with CNTL/Z.**

**Switch(config)#**

**Для просмотра адресной таблицы используется команда**

**show mac address-table**

Утилиты проверки связности сети.

Задача конфигурирования сети является достаточно сложной и трудоемкой, и сложность эта возрастает при увеличении масштабов сети и количества используемых приложений. Поэтому необходимы средства мониторинга и диагностики сети. Эти же средства необходимы нам для моделирования построенной и сконфигурированной сети с использованием Cisco Packet Tracer.

Одним из вспомогательных протоколов – протоколом межсетевых управляющих сообщений (Internet Control Message Protocol – ICMP) – как раз и предоставляются такие средства. Основная идея функционирования протокола ICMP заключается в том, что при возникающих проблемах в продвижении пакетов в сети источнику отсылается сообщение, оповещающее его об этой проблеме. Например, если пакет был отброшен маршрутизатором с использованием сетевого фильтра (маршрутизация и сетевые фильтры будут рассмотрены ниже), сообщение протокола ICMP оповестит источник об этом событии. То же относится и к другим проблемам, возникающим при продвижении пакета – истечение пакетом времени жизни (TTL), отсутствием нужной маршрутной информации в таблице маршрутизации и т.д. Исключение составляют сообщения о проблемах передачи самих ICMP-сообщений, а также некоторых видов пакетов (широковещательных, например), чтобы не наводнять сеть избыточным трафиком.

Так как проблемы с продвижением пакетов могут иметь различную природу, число различных типов ICMP-сообщений также достаточно велико, но все они имеют одинаковую структуру. Несмотря на значительное число типов ICMP-сообщений, они все могут быть разделены на два основных вида [3]:

- сообщения об ошибках;
- сообщения вида «запрос-ответ».

Сообщения об ошибках, как следует из их названия, оповещают узел-источник об ошибках, произошедших в процессе передачи, и каждая такая ошибка характеризуется своим типом сообщения. Сами ошибки могут быть вызваны различными причинами, поэтому в теле такого сообщения содержится дополнительный код, характеризующий причину ошибки.

Сообщения «запрос-ответ» связаны в пары, то есть после передачи сообщения-запроса ожидается получение сообщения-ответа.

В рамках данного пособия нет смысла описывать все типы сообщений протокола ICMP, с ними можно ознакомиться в многочисленных справочниках

[1-3]. Остановимся здесь более подробно на утилитах, позволяющих с использованием ICMP-сообщений проверять связность сети.

Утилита **ping** широко используется для создания ICMP-сообщений вида «запрос-ответ», с помощью которых принудительно вызывается ответ конкретного узла в сети. Это низкоуровневая утилита, поэтому она не требует наличия каких-либо запущенных процессов на удаленных узлах. Соответственно, успешное получение ответа от удаленного узла далеко не всегда означает правильность выполнения на нем прикладного процесса, а говорит только о том, что маршрут к этому узлу находится в работоспособном состоянии, запрашиваемый узел находится в сети и включен.

Удаленный узел, получив ICMP-сообщение, сгенерированное утилитой ping (эхо-запрос), формирует и передает ответное сообщение (эхо-ответ). Запросы обычно посылаются несколько раз (количество запросов по умолчанию различно у разных ОС, кроме того, его можно задавать соответствующим ключом), после чего утилита ping выводит статистические данные.

Если не использовать дополнительные опции, утилита вызывается с использованием командной строки командой ping <доменное имя> или ping <IP-адрес>.

Пример использования утилиты ping в ОС Windows приведен на рисунке 4.1.

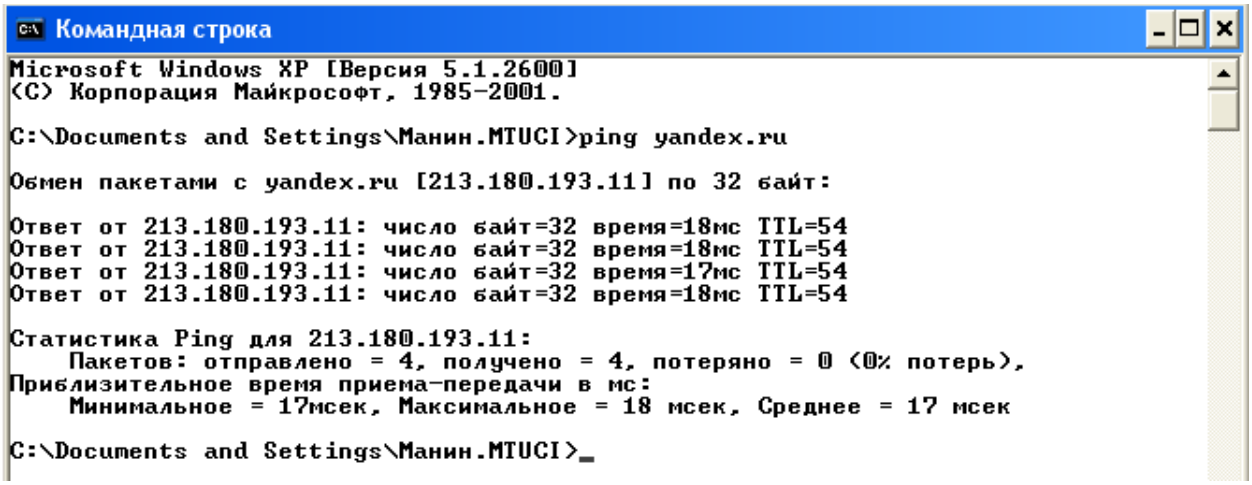



Рисунок 4.1 – Использование утилиты ping

Как следует из рисунка, было отправлено четыре запроса на сервер yandex.ru, получено четыре ответа объемом по 32 байта каждый. Параметр «время» указывает на величину интервала между отправкой эхо-запроса и получением на него эхо-ответа. Параметр TTL позволяет определить количество маршрутизаторов, через которые проходили пакеты, пока не добрались до узла назначения, так как каждый из них уменьшает TTL на единицу.

При конфигурировании сети утилита ping находит широкое применение для проверки связности сети, правил сетевых фильтров, протоколов маршрутизации и т.д., соответственно, и мы будем ее широко использовать в дальнейшем.

Утилита **traceroute** (в реализациях Windows используется написание tracert) позволяет выявлять последовательность маршрутизаторов, через которые проходит пакет на пути к пункту своего назначения. У этой команды есть много опций, большинство из которых применяются редко. Традиционно используется формат traceroute <доменное имя>, которое может быть задано в символической или числовой форме. Выходная информация представляет собой список узлов, начиная с первого маршрутизатора и кончая пунктом назначения.

Принцип работы traceroute основан на установке поля времени жизни (TTL) исходящего пакета таким образом, чтобы это время истекало до достижения пакетом пункта назначения. При получении пакета с обнуленным полем TTL текущий маршрутизатор отправит сообщение об ошибке на узел-источник. Каждое приращение поля времени жизни позволяет пакету пройти на один шаг дальше.

Утилита traceroute посылает для каждого значения поля TTL три пакета. Если промежуточный шлюз распределяет трафик по нескольким маршрутам, то эти пакеты могут возвращаться разными промежуточными узлами (маршрутизаторами). Некоторые маршрутизаторы не посылают уведомлений о пакетах, время жизни которых истекло, а некоторые посылают уведомления,

которые поступают обратно с задержкой, превышающей время ожидания на узле-источнике. Эти маршрутизаторы обозначаются рядом звездочек. Если конкретный маршрутизатор определить нельзя, все равно с помощью утилиты traceroute можно увидеть следующие за ним узлы маршрута. Заметим, что в связи с использованием на сетях динамической маршрутизации, в разные моменты времени можно получить различные маршруты прохождения пакетов.

Пример использования утилиты **tracert** в ОС Windows представлен на рисунке 4.2.

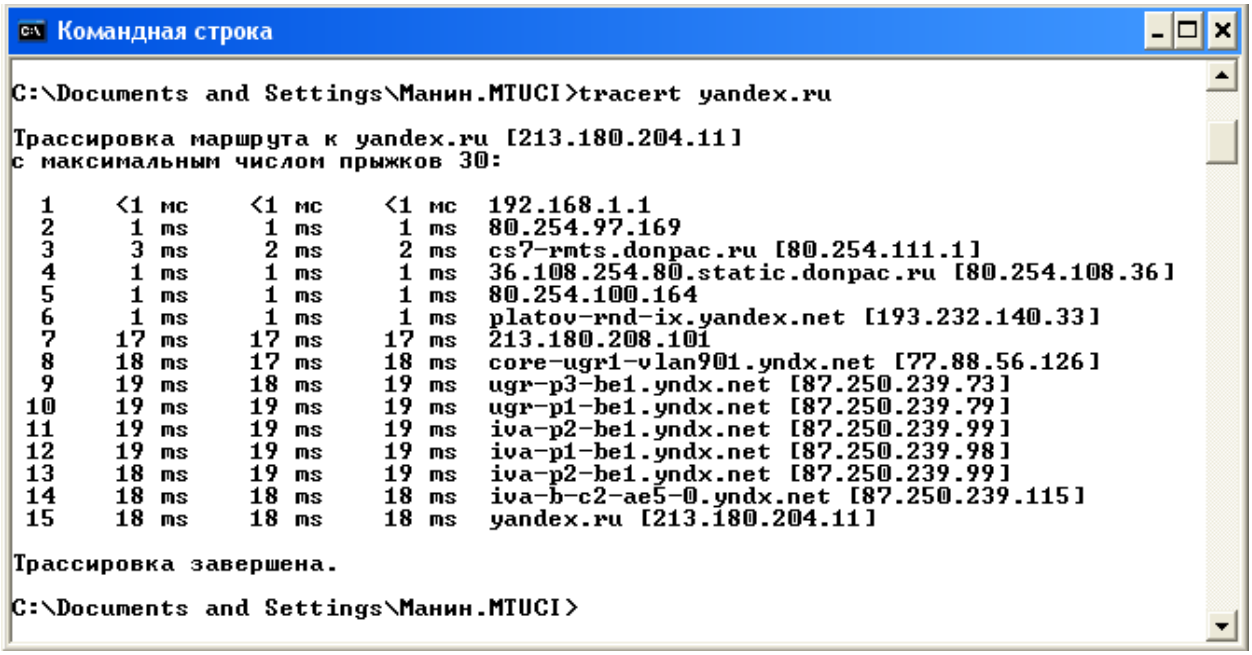



Рисунок 4.2 – Использование утилиты tracert

Как видно из рисунка, утилита позволяет отобразить не только IP-адреса промежуточных узлов, но и их доменные имена, если они существуют.

Утилиты **ping** и **traceroute** сочетают в себе хорошие возможности, предназначенные для тестирования сетевых подключений, но существует команда, которая совмещает в себе все эти возможности, а также содержит некоторые дополнительные особенности. Утилита **PathPing** отправляет многочисленные сообщения с эхо-запросами каждому маршрутизатору, который находится между исходным узлом и узлом назначения, после чего, на основании пакетов, полученных от каждого из них, вычисляет процентное

соотношение пакетов, возвращаемых в каждом шаге. Так как утилита **PathPing** показывает степень потери пакетов на каждом маршрутизаторе или узле, с ее помощью можно точно определить маршрутизаторы и узлы, на которых возникают сетевые проблемы. Эквивалентно утилите traceroute, утилита **PathPing** идентифицирует маршрутизаторы, которые расположены на пути к узлу назначения, после чего она периодически в течение заданного времени обменивается пакетами со всеми маршрутизаторами и на основании числа пакетов, полученных от каждого из них, обрабатывает статистику.

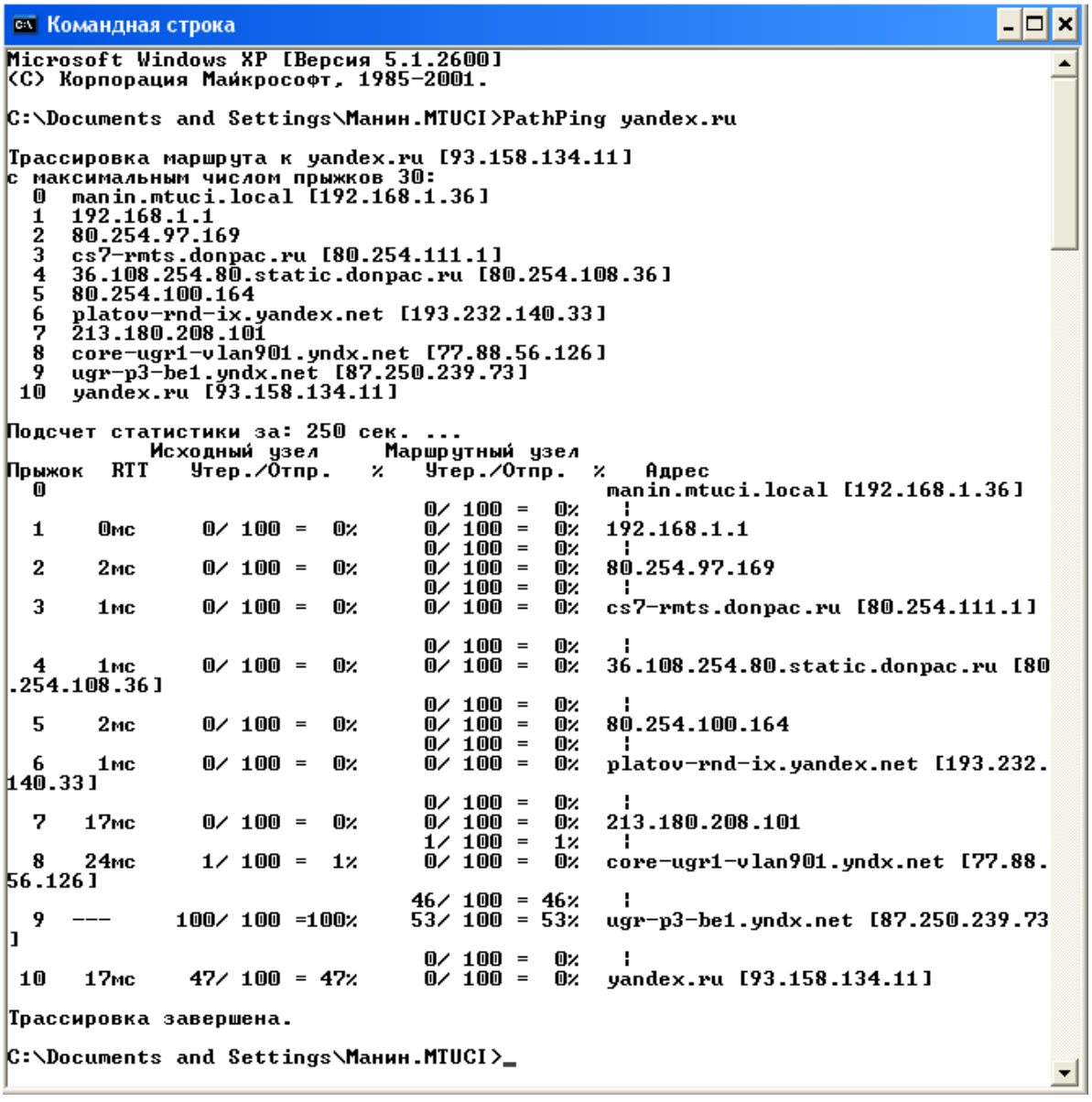



Рисунок 4.3 – Использование утилиты PathPing

В отличие от предыдущих утилит, во избежание перегрузки сети пакеты должны передаваться через довольно большие интервалы времени. Подобно утилите **traceroute** утилита **PathPing** сначала выводит путь, после чего в течение некоторого времени выдает сообщение о том, что она занята. Именно в этот период происходит сбор сведений со всех маршрутизаторов, перечисленных в статистике, и со всех соединений между ними.

В сети, в которой присутствуют только коммутаторы, имеет смысл использовать только утилиту **ping.**

Вернемся к сети, представленной на рисунке 3.10. Так как все рабочие станции соединены с коммутатором, они должны быть доступны друг другу. Для проверки этого необходимо ввести в рабочие станции минимальную сетевую конфигурацию, смысл которой будет рассмотрен ниже (в противном случае просто не будет возможности задать адрес узла в утилите **ping).**

В минимальную конфигурацию рабочей станции входят:

- IP-адрес;

- маска подсети;

Для ввода этой информации в Cisco Packet Tracer необходимо щелкнуть левой кнопкой мыши на рабочей станции, перейти на вкладку Desktop (рабочий стол), и нажать на ярлык IP Configuration (рисунок 4. 4).

Интерфейс рабочей станции позволяет сконфигурировать ее как в соответствии с протоколом IPv.4 (IP Configuration), так и в соответствии с протоколом IPv.6 (IP6 Configuration). В данном случае будем использовать поля IP Configuration.

Как указывалось выше, в нашем примере достаточно задать только два параметра – IP-адрес и маску подсети. Зададим станциям следующие адреса:

192.168.1.1;

192.168.1.2;

192.168.1.3;

192.168.1.4.

Маска подсети (Subnet Mask) для всех четырех станций принимает значение 255.255.255.0.

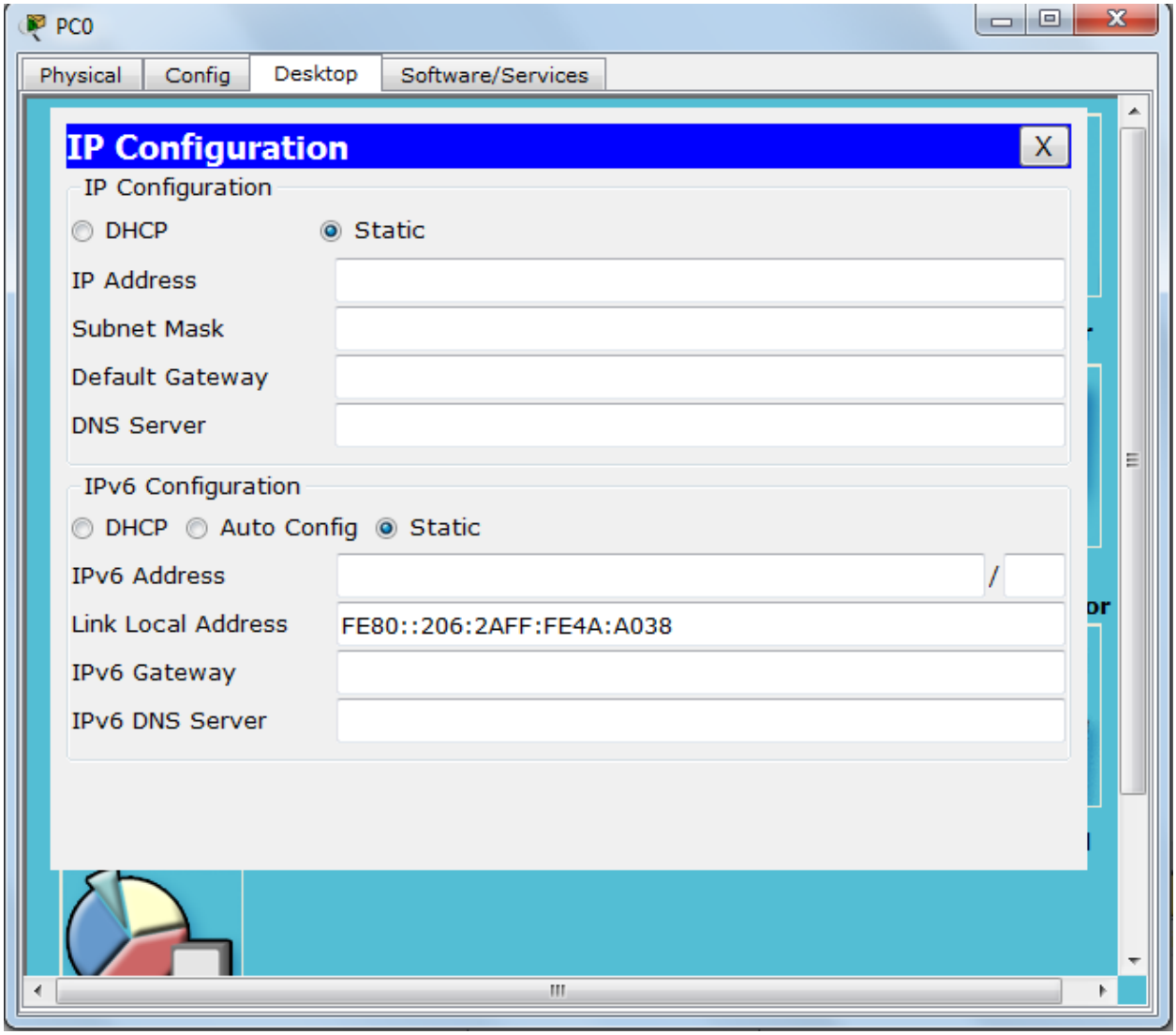



Рисунок 4. 4 – Конфигурирование рабочей станции

После этого можно использовать утилиту **ping** для проверки связности сети. Для этого на рабочем столе PC0 выберем значок Command Prompt (командная строка), и наберем команду

**ping 192.168.1.2**

Результат выполнения этой команды показан на рисунке 4.5.

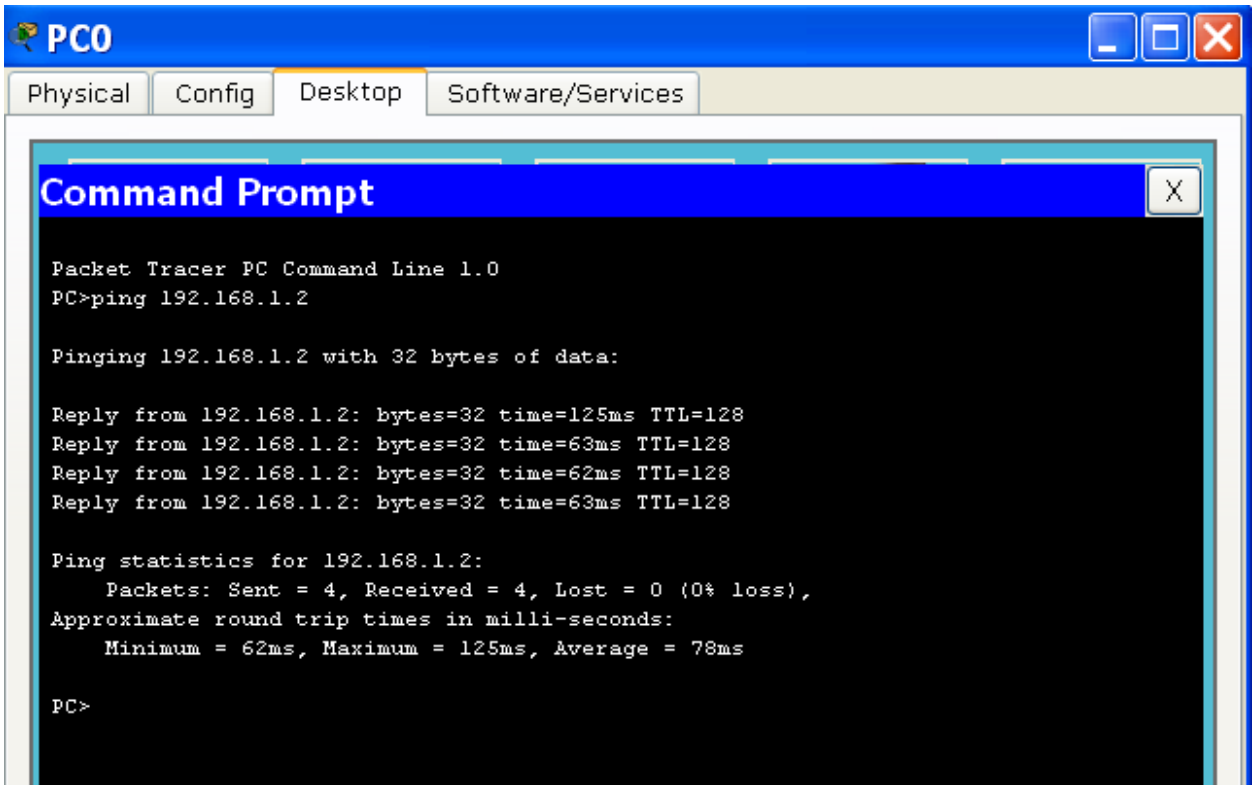



Рисунок 4.5 – Результат проверки связности PC0 и PC1

Как видно из рисунка, утилита **ping** четыре раза отсылала эхо-запросы по 32 байта каждый, параметр time (время) указывает на интервал между отправлением эхо-запроса и получением на него эхо-ответа. Параметр TTL (дословно расшифровывается как Time to live) помогает узнать через сколько маршрутизаторов прошли пакеты, пока добирались до пункта назначения. Операционная система устанавливает некоторое значение TTL по умолчанию (например, Windows устанавливает TTL равным 128), а при прохождении через каждый маршрутизатор данное значение уменьшается на 1. В нашем примере маршрутизаторов нет, поэтому TTL=128.

Внизу приведена статистика, показывающая, что отправлено было 4 пакета (Sent=4), принято 4 (Received=4), потеряно 0 (Lost=0), а также статистику интервала между передачей запроса и получением ответа – минимальное 62 мс (Minimum=62 ms), максимальное 125 мс (Maximum=125 ms), среднее 78 мс (Average=78 ms).

Таким образом, мы убедились, что рабочая станция PC0 и рабочая станция PC1 доступны друг другу. Чтобы увидеть результат выполнения команды в случае недоступности, отправим запрос по несуществующему в данной сети адресу, например, 192.168.1.5. Результат приведен на рисунке 4.6.

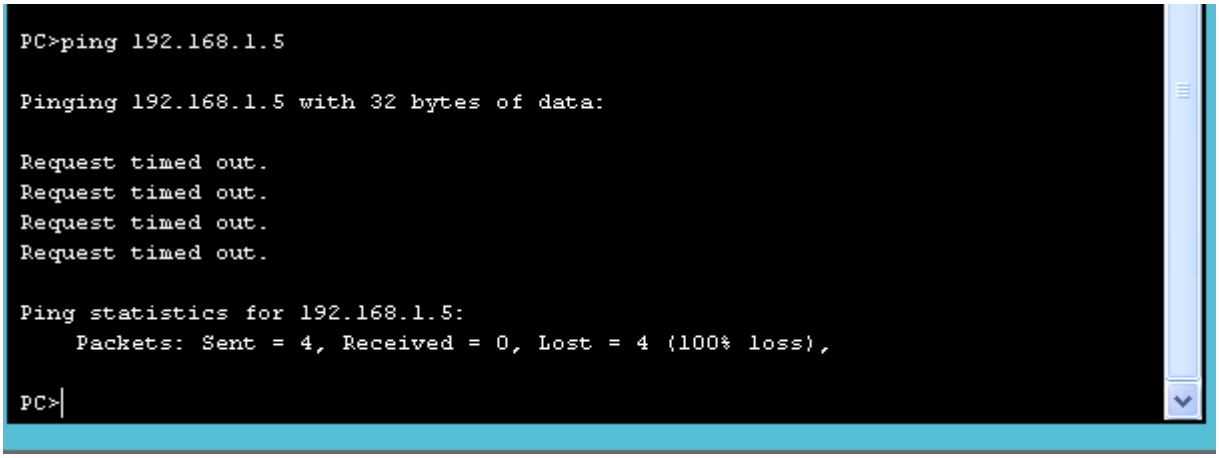



Рисунок 4.6 – Результат выполнения команды

Из рисунка 4.6 следует, что при ожидании ответа на каждый из четырех переданных пакетов был превышен интервал времени ожидания. Статистика сообщает, что было передано 4 запроса, получено 0 ответов, 4 пакета потеряно (100% потерь).

Конфигурирование коммутатора в этом примере не выполнялось, однако можно просмотреть его адресную таблицу. Для этого зайдем в привилегированный режим и выполним команду **show mac address-table** (рисунок 4.7).

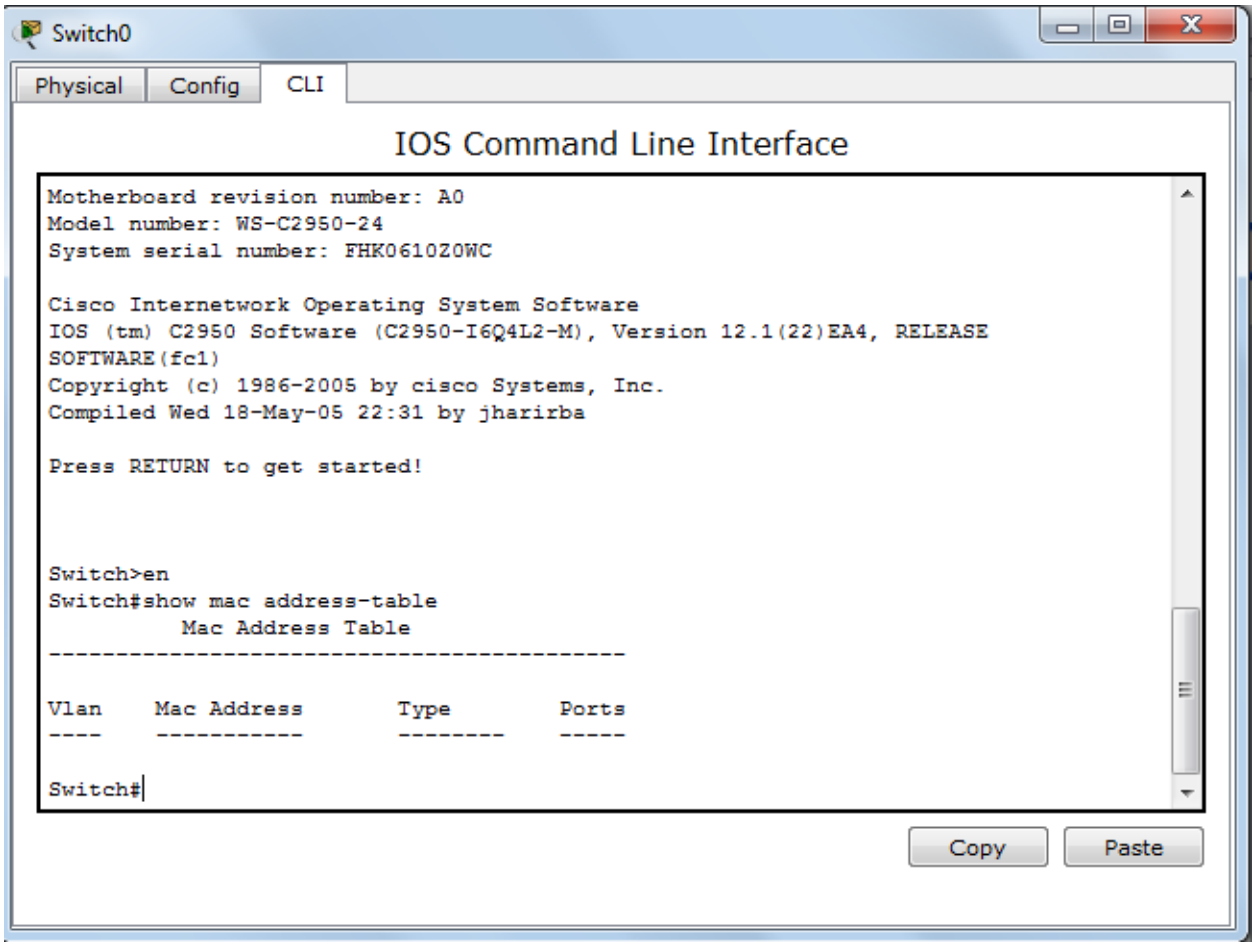



Рисунок 4.7 – Результат выполнения команды show mac address-table

Как видно из рисунка 4.7, адресная таблица пуста, в ней нет ни статических записей (мы их не вносили), ни динамических (коммутатор не прошел процедуру обучения). Для прохождения процедуры обучения достаточно передать от каждого из PC любые кадры, например, ICMP-запросы с использованием утилиты **ping**. Если после этого опять просмотреть адресную таблицу, то она примет вид, показанный на рисунке 4. 8.

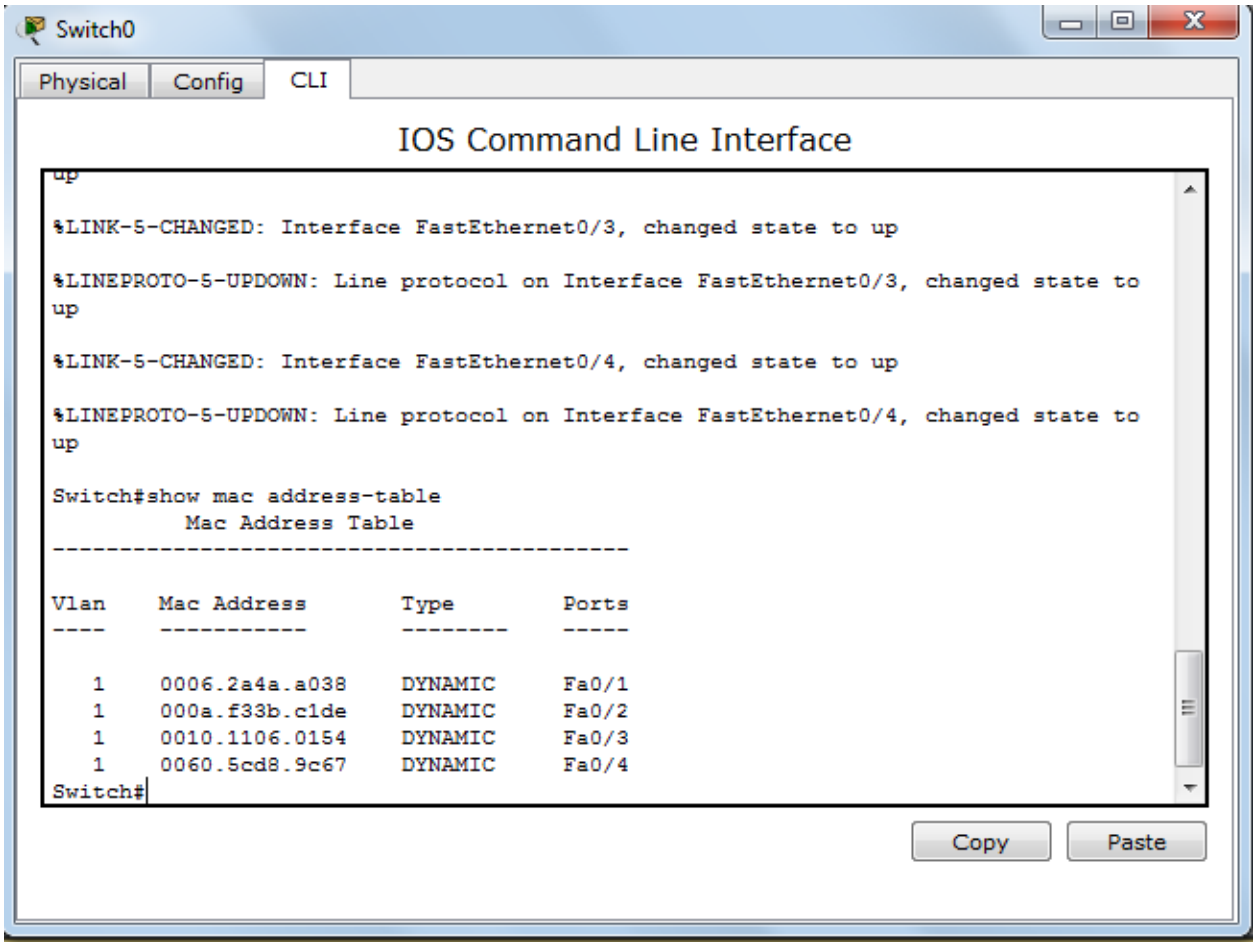



Рисунок 4.8 – Результат выполнения команды show mac address-table после обмена кадрами

Как видно, таблица изменилась. В первом ее столбце указан номер VLAN, MAC-адрес рабочей станции, тип записи (динамическая – Dynamic), и номер интерфейса коммутатора.

Конфигурирование виртуальных сетей.

Технология VLAN была рассмотрена в параграфе 2.5, поэтому в этом параграфе остановимся только на особенностях конфигурирования.

Рассмотрим сначала создание двух VLAN на одном коммутаторе. Для этого по-прежнему будем использовать сеть, представленную на рисунке 4.1. Перейдем в привилегированный режим, выполнив команду enable, и просмотрим информацию о существующих на коммутаторе VLAN (рисунок 4.9), используя для этого команду **show vlan brief**.

В результате выполнения команды на экране появятся: номера VLAN – первый столбец, название VLAN - второй столбец, состояние VLAN (активна она в данный момент или нет) – третий столбец, порты принадлежащие к данной VLAN – четвертый столбец. Как видно из таблицы, по умолчанию на коммутаторе существует пять VLAN . Все порты коммутатора по умолчанию принадлежат VLAN 1. Остальные четыре VLAN не относятся к Ethernet, поэтому рассматривать их здесь не будем.

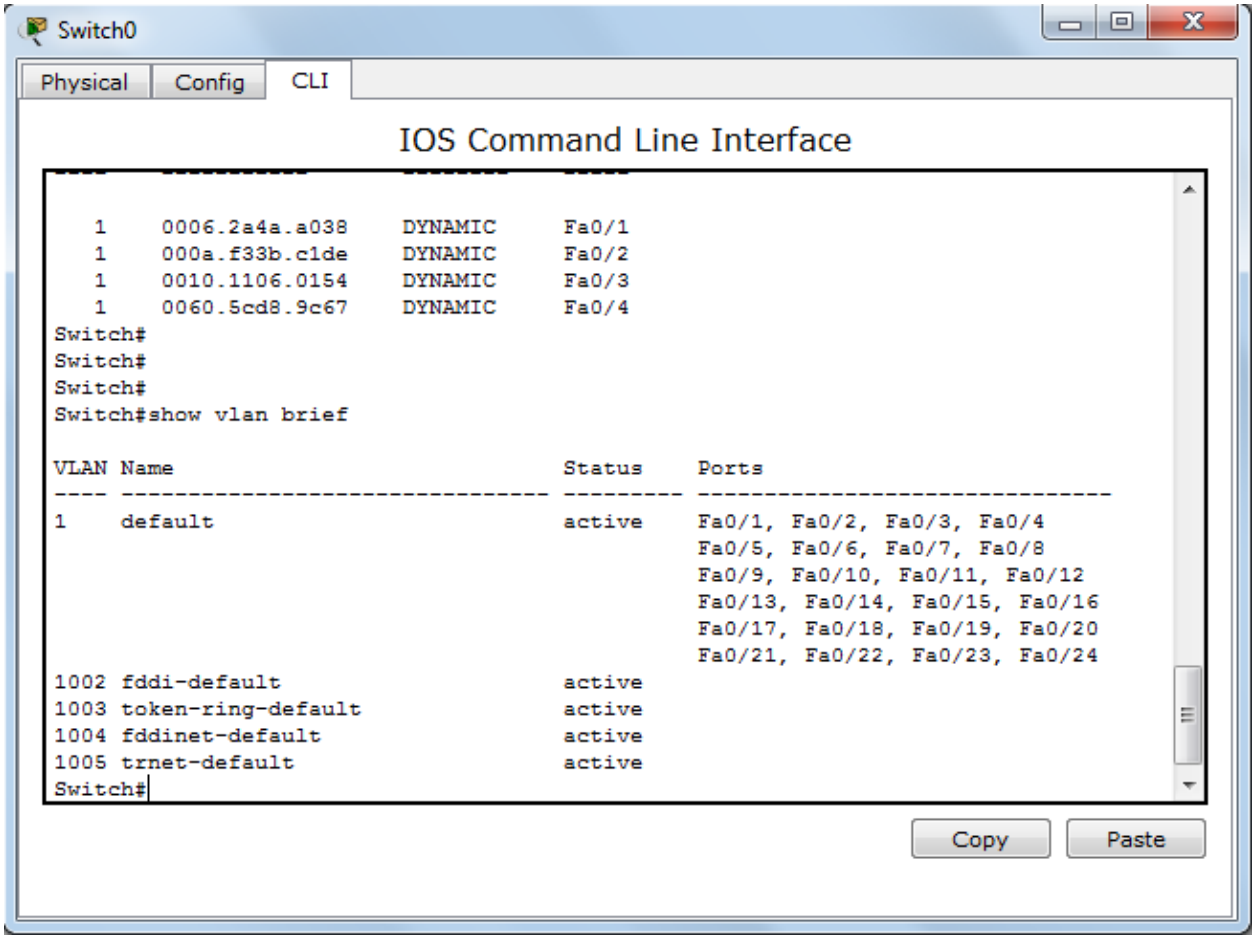



Рисунок 4. 9 – Просмотр конфигурации VLAN

Для реализации сети, которую мы запланировали создать, создадим на коммутаторе еще две VLAN. Для этого в привилегированном режиме необходимо выполнить команду **conf t** для перехода в режим глобального конфигурирования. Вводим команду **vlan 2**. Данной командой создается на коммутаторе VLAN с номером 2. Указатель ввода Switch(config)# изменится на

Switch(config-vlan)#, это свидетельствует о том, что конфигурируется уже не весь коммутатор в целом, а только отдельная VLAN, в данном случае номер 2. Если использовать команду **vlan x**, где x номер VLAN, когда VLAN x еще не создана на коммутаторе, то она будет автоматически создана и будет осуществлен переход к ее конфигурированию.

Для решения поставленной задачи коммутатор необходимо сконфигурировать следующим образом.

**Switch(config)#vlan 2**

**Switch(config-vlan)#name subnet_192**

**Switch(config)#interface range fastEthernet 0/1-2**

**Switch(config-if-range)#switchport mode access**

**Switch(config-if-range)#switchport access vlan 2**

Разберем приведенные команды. Как уже говорилось ранее, командой **vlan 2** мы создаем на коммутаторе новую VLAN с номером 2. **Команда name subnet_192** присваивает имя subnet_192 виртуальной сети номер 2. Выполняя команду **interface range fastEthernet 0/1-2**, мы переходим к конфигурированию интерфейсов fastEthernet 0/1 и fastEthernet 0/2 коммутатора. Ключевое слово **range** в данной команде указывает на то, что мы будем конфигурировать не один единственный порт, а целый диапазон портов, в принципе ее можно не использовать, но тогда последние три строки придется заменить на следующие:

**Switch(config)#interface fastEthernet 0/1**

**Switch(config-if)#switchport mode access**

**Switch(config-if)#switchport access vlan 2**

**Switch(config)#interface fastEthernet 0/2**

**Switch(config-if)#switchport mode access**

**Switch(config-if)#switchport access vlan 2**

Команда **switchport mode access** конфигурирует выбранный порт коммутатора как порт доступа (в принципе, эту команду можно не использовать, так как по умолчанию порты коммутатора находятся в режиме

доступа). Команда **switchport access vlan 2** привязывает данный порт к VLAN номер 2.

Как и в предыдущих примерах, команды можно набирать сокращенно, кроме того, вместо **fastEthernet** можно использовать обозначение **fa.**

Просмотрим результат конфигурирования, выполнив команду **show vlan br** после выполнения приведенных выше команд, рисунок 4.10.

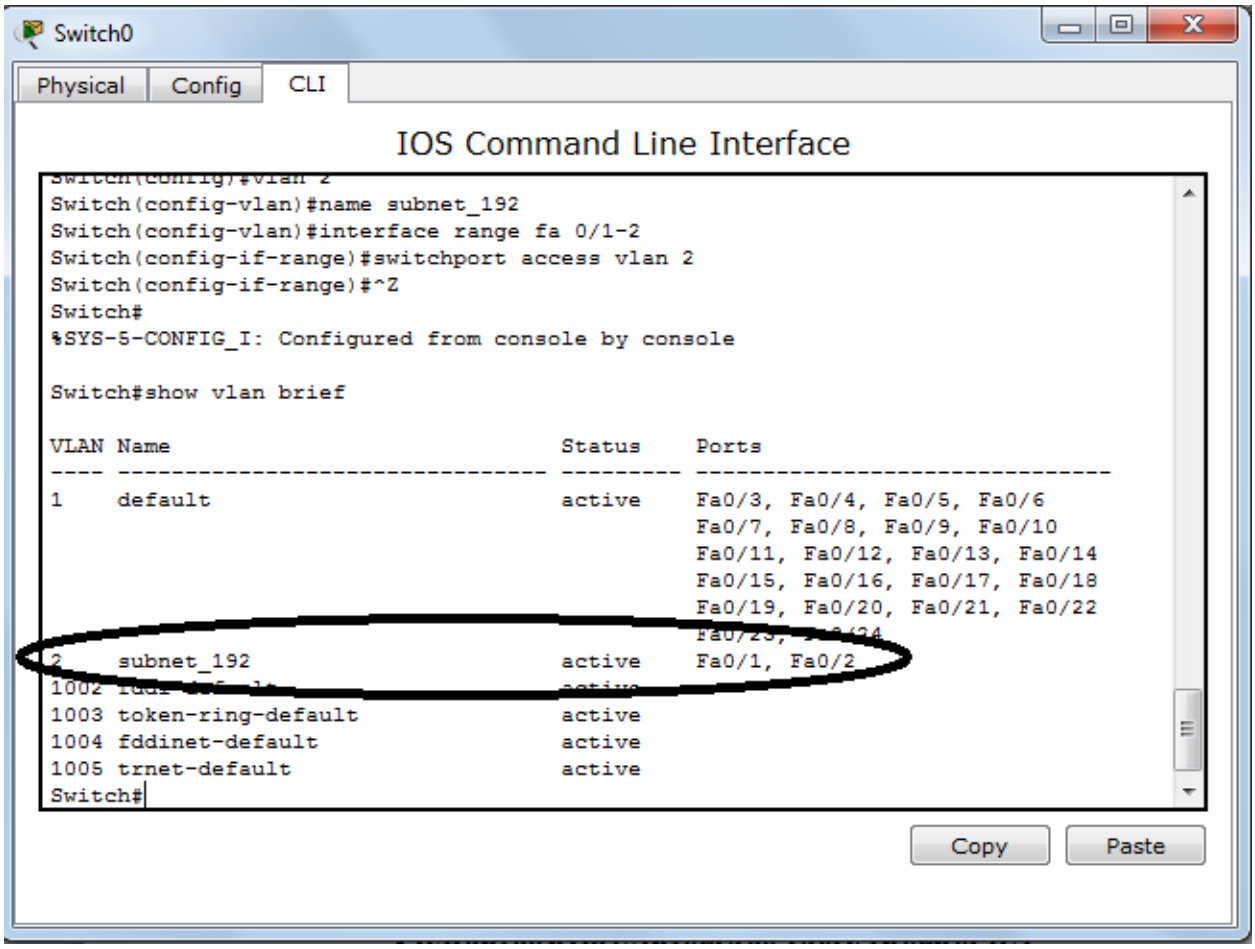



Рисунок 4.10 – Просмотр созданной VLAN 2 и привязанных к ней портов

Из рисунка видно, что в коммутаторе появилась вторая VLAN с именем subnet_192, к которой относятся порты fa 0/1 и 0/2 (выделены на рисунке).

Далее аналогичным образом создадим vlan 3 с именем subnet_172, и привяжем к ней интерфейсы fastEthernet 0/3 и fastEthernet 0/4.

Соответственно, компьютеры, находящиеся в разных виртуальных сетях, будут недоступны друг другу, что легко проверить с использованием утилиты **ping**.

Наибольшую практическую ценность, как было показано в параграфе 2.5, представляет конфигурирование VLAN на нескольких коммутаторах с использованием тэгированных портов. Рассмотрим конфигурирование коммутаторов в этом случае.

Рассмотрим сеть, показанную на рисунке 4.11.

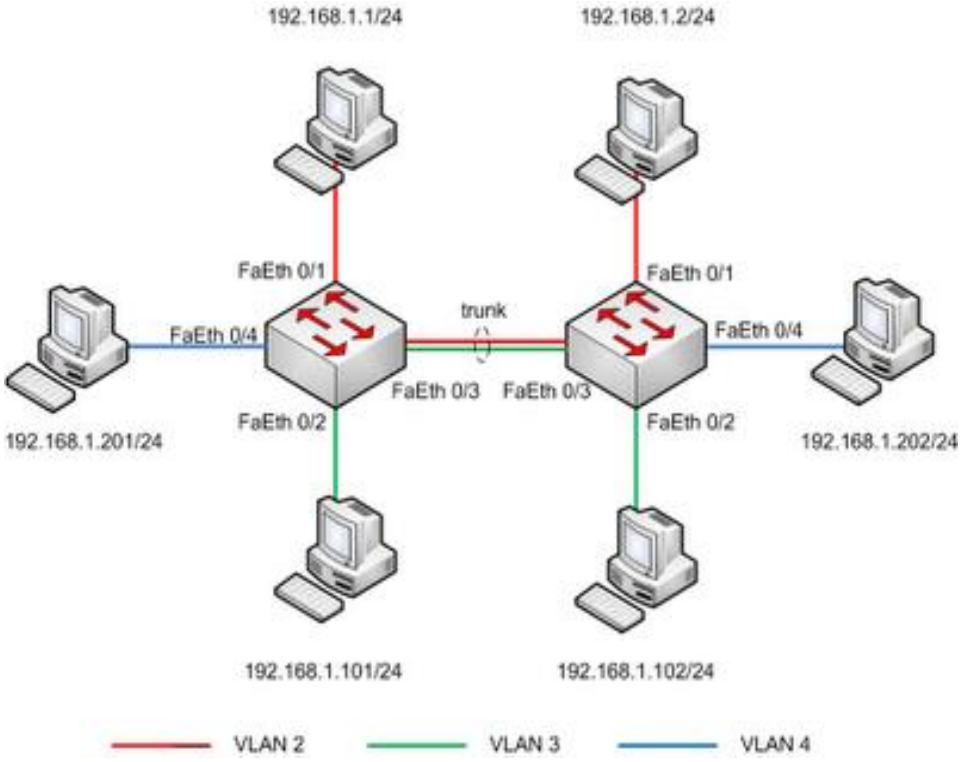



Рисунок 4.11 – Пример сети с двумя коммутаторами

По аналогии с предыдущим примером произведем конфигурирование обоих коммутаторов:

**Switch(config)#vlan 2**

**Switch(config-vlan)#name subnet_2**

**Switch(config-vlan)#int fa 0/1**

**Switch(config-if)#switchport mode access**

**Switch(config-if)#switchport access vlan 2**

**Switch(config-if)#exit**

**Switch(config)#vlan 3**

**Switch(config-vlan)#name subnet_3**

**Switch(config-vlan)#int fa 0/2**

**Switch(config-if)#switchport mode access**

**Switch(config-if)#switchport access vlan 3**

**Switch(config-if)#exit**

**Switch(config)#vlan 4**

**Switch(config-vlan)#name subnet_4**

**Switch(config-vlan)#int fa 0/4**

**Switch(config-if)#switchport mode access**

**Switch(config-if)#switchport access vlan 4**

**Switch(config-if)#exit**

Как следует из перечня команд, на обоих коммутаторах созданы три VLAN. Теперь необходимо перевести третьи порты каждого из коммутаторов в тэгированный режим:

**Switch(config)#int fa 0/3**

**Switch(config-if)# switchport mode trunk**

В результате трафик всех трех созданных ранее VLAN будет проходить через порт 3.

Используя на интерфейсе команду switchport mode trunk, мы перевели его в тэгированный режим, в котором интерфейс пропускает через себя трафик всех существующх на коммутаторе VLAN, но иногда необходимо передавать через данный интерфейс трафик не всех VLAN, а лишь некоторых. Для этого на обоих коммутаторов выполним команды:

**Switch(config)#interface fastEthernet 0/3**

**Switch(config-if)#switchport trunk allowed vlan 2-3**

Команда **switchport trunk allowed vlan 2-3** указывает тэгированному порту коммутатора, трафик каких VLAN ему пропускать через себя (в нашем случае второго и третьего). После того, как будет выполнена эта команда,

компьютер PC4 должен перестать видеть компьютер PC5. Команда **switchport trunk allowed vlan** при своем использовании каждый раз задает разрешенные порты заново, то есть если выполнить команду **switchport trunk allowed vlan 5**, а потом выполнить команду **switchport trunk allowed vlan 6**, то разрешенным окажется только трафик VLAN номер 6. Для удаления VLAN из списка разрешенных используется команда **switchport trunk allowed vlan remove x**, где x номер удаляемой VLAN. Для просмотра информации о настроенных на коммутаторе магистральных портах служит команда **show int trunk.**

Таким образом, с использованием коммутаторов второго уровня единую сеть можно разделить на виртуальные сети с изолированным друг от друга трафиком. Однако на практике часто возникают задачи гибкого объединения нескольких VLAN между собой. Эту задачу решают маршрутизаторы и коммутаторы третьего уровня, которые будут рассмотрены ниже.

***Лабораторная работа №5.*** Преобразование сетевых адресов NAT. Конфигурирование NAT для различных топологий сети.

**Цели работы**: Исследовать возможности применения NAT для расширения возможностей протокола IPV4.

**Теоретический материал.**

NAT расшифровывается как Network Address Translation – трансляция (преобразование) сетевых адресов. Рассмотрим сначала причины разработки этой технологии.

Во-первых, уже сейчас наблюдается дефицит IP-адресов четвертой версии. Кардинальным решением здесь может служить переход к шестой версии IP-протокола, но пока повсеместно используется IPv4. При использовании NAT в пределах внутренней сети могут использоваться частные адреса. К частным (немаршрутизируемым) адресам относятся адреса, входящие в специально выделенные для них диапазоны [3]:

10.0.0.1 – 10.255.255.254 для класса A (или с маской 255.0.0.0)

172.16.0.1 – 172.31.255.254 для класса B (или с маской 255.255.0.0)

192.168.0.1 – 192.168.255.254 для класса C (или с маской 255.255.255.0)

Частные адреса могут использоваться только в локальных сетях, магистральные маршрутизаторы такие адреса не обрабатывают, они работают только с общедоступными адресами.

Одни и те же частные адреса могут использоваться в различных локальных сетях, что и приводит к экономии адресного пространства. Преобразование частных адресов в общедоступные и обратно осуществляется с использованием NAT.

Во-вторых, NAT существенно повышает безопасность локальной сети, так как в этом случае извне сеть представляется единственным или несколькими общедоступными адресами. Поэтому определить структуру

локальной сети, проанализировать данные, циркулирующие в ней, становится проблематично.

Основная идея технологии NAT состоит в следующем. Локальная сеть использует адресное пространство частных адресов. В маршрутизаторе или другом устройстве, связывающем локальную (внутреннюю) сеть с внешней IP-сетью, настраивается протокол NAT, осуществляющий при передаче во внешнюю сеть преобразование частного адреса в общедоступный и обратное преобразование при приеме. Так как внутренняя сеть также может содержать маршрутизаторы для разделения ее на подсети, они должны получать объявления о маршрутной информации от маршрутизаторов внешней сети. В свою очередь, внешние маршрутизаторы не должны ничего знать о маршрутизаторах внутренней сети. Поэтому NAT-устройство должно пропускать из внешней сети во внутреннюю сообщения протоколов маршрутизации (RIP, OSPF и т.д.), но не пропускать эти сообщения в обратном направлении. Число общедоступных адресов чаще всего меньше числа частных адресов, за счет чего и достигается экономия адресного пространства. В частном, но далеко не самом редком случае, может использоваться всего один общедоступный адрес, настраиваемый на внешнем порту NAT-маршрутизатора.

Самой простой технологией является статический NAT. Суть его заключается в том, что в NAT-устройстве прописываются специальные таблицы трансляций (преобразований), жестко связывающие внутренние (частные) и внешние (общедоступные) адреса. Недостаток такой технологии очевиден – количество общедоступных адресов у NAT-устройства должно соответствовать количеству узлов внутренней сети, имеющих права доступа во внешнюю сеть. Как следствие, экономии адресного пространства не происходит.

Поэтому в настоящее время широко используется динамический NAT, суть которого рассмотрим на рисунке 5.1, заимствованным из [1-3].

На рисунке представлена внутренняя сеть, использующая частный адрес 192.168.1.0/24. Выход во внешнюю сеть организуется с использованием NAT-устройства, внешнему интерфейсу которого присвоен общедоступный адрес 80.254.97.169. Необходимо обеспечить всем четырем конечным узлам внутренней сети доступ к внешней сети, в частности, к web-серверу с адресом 213.180.193.11.

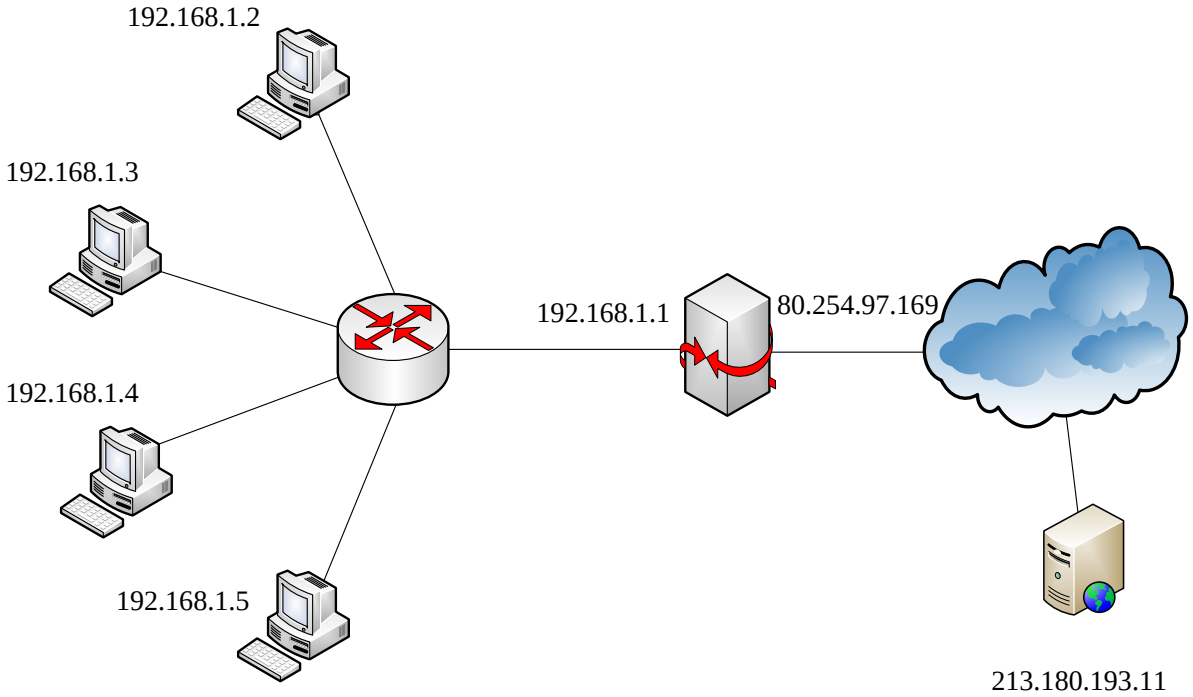



Рисунок 5.1 – Иллюстрация работы динамического NAT

Очевидно, что статический NAT для решения такой задачи непригоден, так как доступ узлов к внешней сети осуществляется с использованием единственного внешнего адреса (на практике внешних адресов также может быть несколько, но в любом случае количество внутренних узлов превышает количество внешних адресов).

При передаче пакета во внешнюю сеть NAT-устройство может подменить частный адрес отправителя на свой общедоступный адрес, как и в статическом NAT. Однако при приеме пакета-ответа из внешней сети от сервера необходимо определить, какому из внутренних конечных узлов этот пакет нужно передать. Или, другими словами, при приеме необходимо определить, на какой частный

адрес нужно изменить общедоступный адрес назначения, содержащийся в ответном IP-пакете.

Таким образом, для надежного различения принимаемых пакетов NAT-устройством необходима, помимо IP-адресов, дополнительная информация. В качестве такой информации можно использовать номера портов TCP- или UDP-сегментов, переносимых IP-пакетами. Однако в нашем примере все четыре узла могут обратиться с запросом к web-серверу с адресом 213.180.193.11, ответы которого будут иметь один и тот же номер порта 80 или 8080. Поэтому в данном случае используются так называемые назначенные номера портов. В качестве назначенных портов используются порты источника, которым в процессе передачи присваиваются значения, не стандартизированные в протоколах TCP и UDP. Назначенный порт может быть выбран произвольно, но с учетом того, что он должен быть уникален в пределах внутренней сети.

В соответствии с этим таблица NAT-устройства усложняется, в нее теперь должны входить не только IP-адреса, но и номера портов (таблица 5.1).

Поскольку описанная выше технология использует не только сетевые адреса, но и номера портов, она получила название NAPT (Network Address Port Translation) [1-3].

Таблица 5.1 – Примерный вид NAT-таблицы

| Частный адрес | Порт | Общедоступный адрес | Назначенный порт |
|---|---|---|---|
| 192.168.1.2 | 8080 | 80.254.97.169 | 61001 |
| 192.168.1.3 | 8080 | 80.254.97.169 | 61002 |
| 192.168.1.4 | 8080 | 80.254.97.169 | 61003 |
| 192.168.1.5 | 8080 | 80.254.97.169 | 61004 |

При передаче пакета, например, от узла 192.168.1.2 к серверу глобальной сети с адресом 213.180.193.11 в заголовок пакета в качестве адреса получателя будет указан 213.180.193.11, в качестве номера порта получателя – 8080. В качестве адреса отправителя будет указан 192.168.1.2, а в качестве номера порта отправителя – 8080. После приема этого пакета NAT-устройством будет

произведена подмена адреса отправителя на 80.254.97.169, а номера порта отправителя на 61001. Эта информация динамически заносится в таблицу 5.1.

При приеме ответа от сервера глобальной сети будет выполнено обратное преобразование – адрес получателя будет заменен на 192.168.1.2. При этом в качестве номера порта получателя будет указан назначенный порт, который сервер укажет исходя из номера порта источника принятого сегмента. При этом NAT-устройство «поймет», какому из внутренних узлов передать пакет, используя номер назначенного порта.

Если NAT-устройство имеет несколько общедоступных адресов (пул адресов), то таблица 5.1 ведется динамически, то есть при передаче пакета запоминается, на какой именно адрес из пула была осуществлена подмена, и данная информация заносится  таблицу. Эти действия, естественно, являются абсолютно прозрачными для конечных узлов.

Конфигурирование NAT на маршрутизаторе.

Рассмотрим настройку протокола NAT для примера, представленного на рисунке 5.1, полагая, что в качестве NAT-устройства используется маршрутизатор Cisco.

Предположим, что в маршрутизаторе, используемом в качестве NAT-устройства, порт с адресом 192.168.1.1 является портом fa 0/0, а порт с адресом 80.254.97.169 – портом fa 0/1. В терминологии NAT порт fa 0/0 является внутренним портом (inside), а порт fa 0/1 – внешним портом (outside).

Пакеты, прибывающие на внутренний порт и подлежащие передаче на внешний порт, подлежат трансляции в соответствии с Source NAT (SNAT), то есть подмене подлежит IP-адрес источника (Source IP). Пакеты, прибывающие на внешний порт, подлежат трансляции в соответствии с Destination NAT (DNAT), то есть подмене подлежит IP-адрес получателя (Destination IP).

Сначала необходимо создать список доступа (подробнее списки доступа будут рассмотрены в следующем разделе). Для этого в режиме глобального конфигурирования необходимо выполнить следующую команду:

**(config)# access-list 100 permit ip <адрес> <инвертированная маска> any**

Забегая вперед, отметим, что данной командой создается список доступа с номером 100, разрешающий передавать пакеты с адресом источника, указанного в команде, на любые адреса.

Пул адресов создается на маршрутизаторе в режиме глобального конфигурирования командой

**(config)# ip nat pool <имя> <начальный адрес> <конечный адрес> netmask <маска>.**

Если, как в нашем примере, используется единственный общедоступный адрес, начальный и конечный адреса в команде совпадают.

Затем назначаются внутренние и внешние интерфейсы командами:

**(config)# interface fa 0/0;**

**(config-if)# ip nat inside (outside)**

Включается NAT командой

**ip nat inside source list 100 pool <имя>**

Конфигурирование маршрутизатора Cisco с использованием указанных команд для нашего примера представлен на рисунке 5.2.

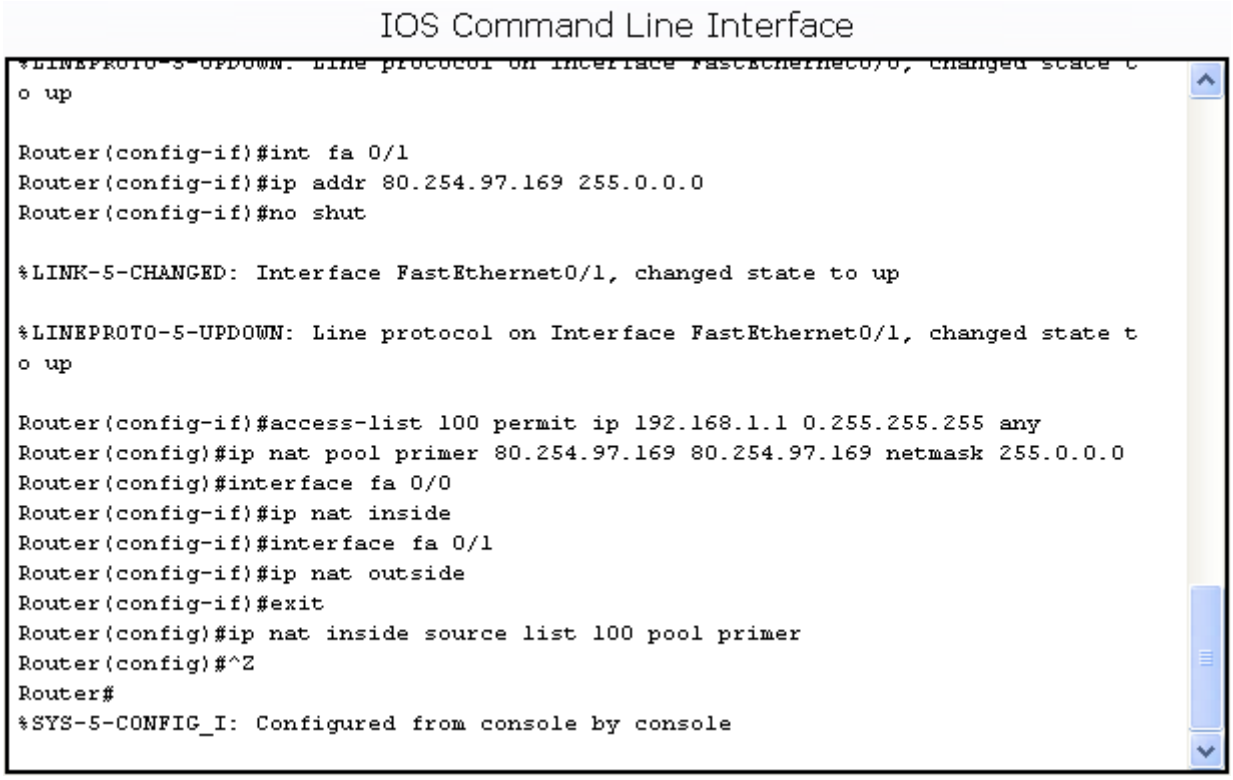



Рисунок 5.2 – Конфигурирование динамического NAT

После того, как какой-либо из внутренних узлов обменяется пакетами с внешней сетью, можно будет просмотреть трансляции адресов, произведенные NAT, с использованием команды **show ip nat translations** (рисунок 5.3).

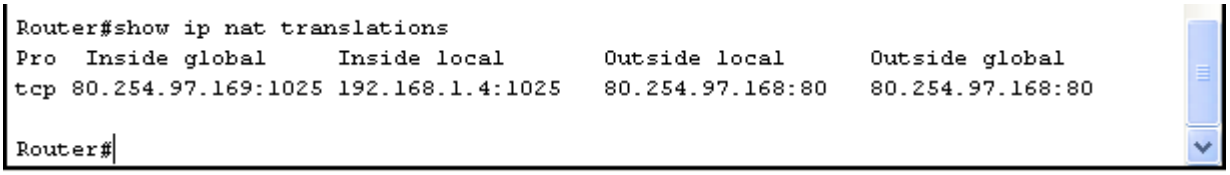



Рисунок 5.3 – Список трансляций после обращения к web-серверу

Из рисунка 5.3 следует, что была произведена одна трансляция, информация о которой представлена в четырех колонках.

Первая колонка указывает на транспортный протокол, в нашем случае это TCP.

Вторая колонка (Inside global) указывает на сокет (IP-адрес и номер порта), на который подменяется сокет отправителя.

Третья колонка (Inside local) указывает на внутренний IP-адрес отправителя с назначенным номером порта.

Четвертая колонка (Outside local) указывает на сокет узла назначения во внешней сети, который сформирован внутренним узлом-отправителем.

Пятая колонка (Outside global) указывает на IP-адрес и номер порта, используемые во внешней сети.

Таким образом, из рисунка 5.3 следует, что внутренний узел с адресом 192.168.1.4 направляет пакет web-серверу с адресом 80.254.97.168. Соответственно, IP-адрес и номер порта получателя, указанные в пакете:

80.254.97.168:80 (напомним, что для протокола HTTP используются порты 80 и 8080).

IP-адрес и порт источника в этом же пакете:

192.168.1.4:80 .

При передаче пакета во внешнюю сеть маршрутизатор подменяет IP-адрес и порт источника:

80.254.97.169:1025.

Соответственно, при приеме ответного пакета от сервера сокет 80.254.97.169:1025 будет изменен на 192.168.1.4:80, и пакет получит нужный узел внутренней сети.

**Задание для работы.**

Выполнить настройку NAT для сети, представленной на рисунке 5.1. Адресация для сети выбирается аналогичной той, которая представлена на рисунке, но с учётом номера студента по журналу **V**.

Для ПК 192.168.V.2 - ПК 192.168.V.5:

Для шлюза 192.168.V.1 и 80.254.V.169;

Для сервера 213.180.V.11.

Задание выполняется в среде Cisco Packet Tracer. Отчёт формируется в электронном виде. В обязательном порядке должны быть приведены скриншоты с производимой настройкой и результатами проверки проведённых настроек.

***Лабораторная робота №6.*** Конфигурирование IP-АТС Open Scape Office MX.

**Цель работы:**

- Изучение оборудования IP-АТС OSO MX и принципов начального конфигурирования оборудования.

- Получить навыки конфигурирования аппаратных локальных IP-АТС.

Состав системы.

Основным модулем системы является модуль материнской платы (рисунок 1), имеющий порты для подключения к IP-сети. Материнская плата содержит центральный процессор системы и является главной платой, управляющей работой всех остальных систем.

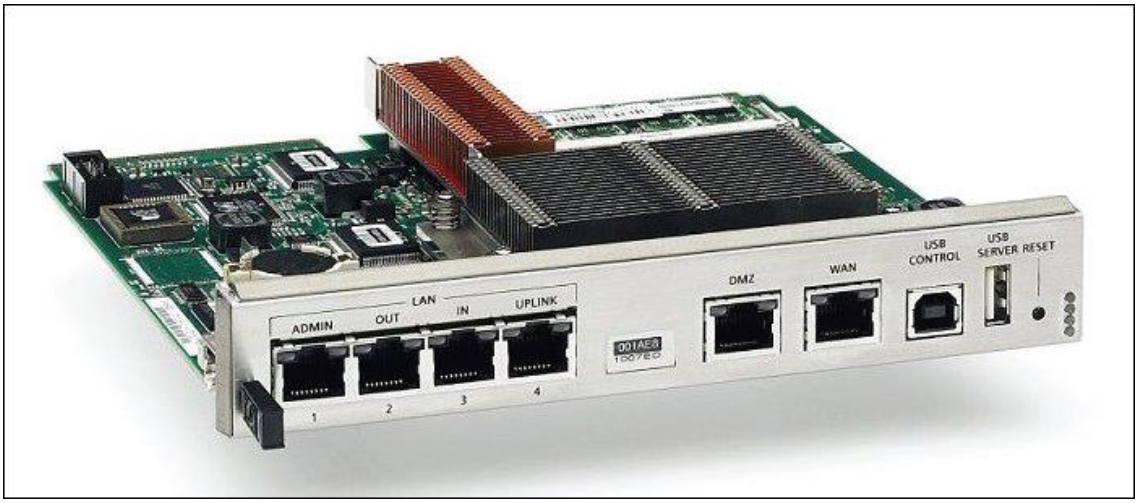



Рисунок 6.1 – Материнская плата

Фронтальная панель материнской платы представлена на рисунке 6.2.


Рисунок 6.2 – Фронтальная панель

Назначение интерфейсов фронтальной панели:

4 LAN-интерфейса:

1. ADMIN – для соединения с сервисным ПК и последующего администрирования системы;
2. OUT – в однобоксовой системе не используется. В многобоксовой системе используется для связи с другим боксом;
3. IN – в однобоксовой системе не используется. В многобоксовой системе используется для связи с другим боксом;
4. UPLINK – для связи с внешней сетью.

DMZ-интерфейс – для соединения с почтовым сервером или Web-сервером организации, в которой установлена система.

WAN-интерфейс – для соединения с оператором IP-телефонии (ITSP), например, по технологии xDSL.

USB-CONTROL – для соединения с сервисным ПК для диагностики.

USB-SERVER – для соединения с внешним диском (или другим накопителем с USB-интерфейсом) для сохранения параметров системы или обновления версии программного обеспечения.

Кроме того, на фронтальной панели находится кнопка RESET. При нажатии на кнопку в течение интервала менее 10 с происходит перезапуск системы без изменения каких-либо параметров конфигурации. После перезапуска система начинает функционировать в нормальном режиме.

При нажатии на кнопку в течение интервала более 10 с система возвращается в настройки по умолчанию (default).

Данную кнопку задействовать желательно только в чрезвычайной ситуации.

Кратко рассмотрим назначение остальных модулей системы.

Модуль GMS обеспечивает четыре интерфейса базового доступа ISDN BRI.

Модуль GME обеспечивает один интерфейс первичного доступа ISDN PRI.

Модуль GMAA предоставляет четыре интерфейса для подключения аналоговых соединительный линий и два интерфейса для подключения аналоговых абонентских линий.

Модуль GMAL обеспечивает подключение восьми аналоговых абонентских линий.

Инсталляция системы.

Система Open Scape Office предполагает использование различных технологий соединения с внешней коммутируемой сетью:

— ISDN-соединение точка-точка с использованием интерфейса $S_0$ (BRI-интерфейс без питания);

— ISDN-соединение точка-многоточка с использованием интерфейса $S_0$ (BRI-интерфейс без питания);

— ISDN-соединение с использованием интерфейса $S_{2M}$ (PRI-интерфейс);

— аналоговый транк.

ISDN-соединение точка-точка с использованием интерфейса $S_0$ использует плату GMS или GMSA. Схема такого соединения представлена на рисунке 6.3.

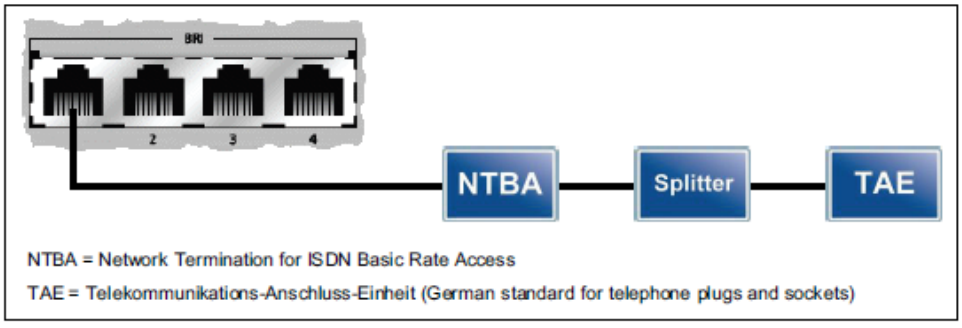



Рисунок 6.3 – Схема ISDN-соединения точка-точка с использованием интерфейса $S_0$

На рисунке 6.3 приняты следующие обозначения:

NTBA – сетевое окончание ISDN BRI;

Splitter – разделитель;

TAE – терминальное оборудование.

ISDN-соединение точка-многоточка с использованием интерфейса $S_0$ производится аналогично.

ISDN-соединение с использованием интерфейса $S_{2M}$ (PRI-интерфейс) производится с использованием платы GME. Схема соединения представлена на рисунке 6.4.

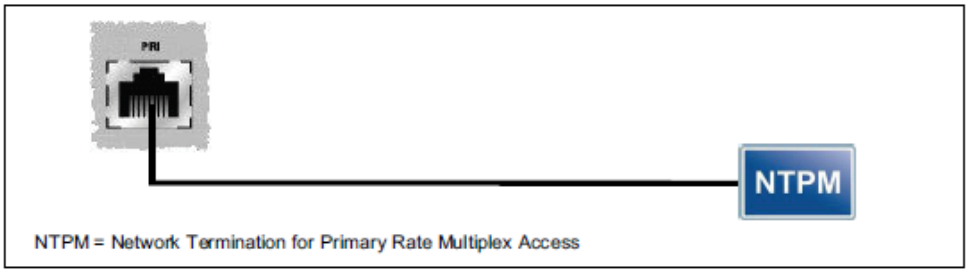



Рисунок 6.4 – Схема ISDN-соединения с использованием интерфейса $S_{2M}$ (PRI-интерфейс)

Аналоговое соединение производится с использованием платы GMAA. Схема соединения приведена на рисунке 6.5.

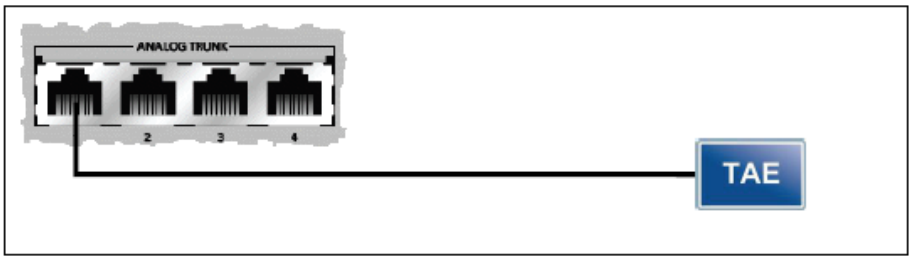



Рисунок 6.5 – Аналоговое соединение

Основные сведения об администрировании системы

Администрирование системы Open Scape Office производится с использованием web-интерфейса Open Scape Assistant. На управляющем ПК должно быть установлено соответствующее программное обеспечение:

— Microsoft Internet Explorer 7 (Windows XP, Windows 2003 and Windows Vista);

— Microsoft Internet Explorer Version 8 (Windows XP, Windows 2003, Windows Vista and Windows 7);

— Microsoft Internet Explorer Version 9 (Windows Vista and Windows 7);

— Mozilla Firefox 4 (Windows XP, Windows 2003, Windows Vista, Windows 7 and Linux).

Кроме того, должен быть установлен Java Script.

Элементы пользовательского интерфейса Open Scape Assistant представлены на рисунке 6.6.

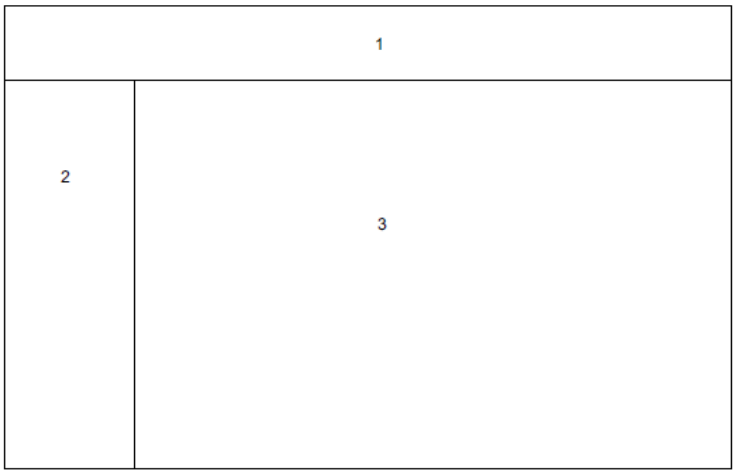



Рисунок 6.6 – Элементы пользовательского интерфейса Open Scape Assistant

На рисунке 6.6 приняты следующие обозначения.

1 – навигационная панель – открывает доступ к пунктам меню Home (Домой), Administrators (Администраторы), Setup (Установка), Expert Mode (Экспертный режим), Data Backup (Резервное копирование), License Management (Менеджер лицензий), Service-Center (Сервис-центр).

2 – область навигации – содержит пункты подменю, которые зависят от выбранного пункта меню на навигационной панели.

3 – рабочая область – область, в которой непосредственно выполняются задачи администрирования системы. В режиме Expert Mode в рабочей области отображаются пункты подменю нижнего уровня, запуск которых обеспечивается Java-приложениями. Навигация в дереве меню осуществляется путем двойного нажатия на левую кнопку мыши при выборе соответствующего пункта.

Вход в Open Scape Assistant и назначение Wizards.

При входе в систему осуществляется авторизация по логину и паролю. Для входа в систему необходимо:

1. Используя адресную строку браузера, перейти по ссылке **https://<IP-адрес OpenScape Office>**. По умолчанию OSO MX имеет IP-адрес **192.168.1.2**.

2. В случае, если интернет-браузер выдаст сообщение о недостоверном соединении, подтвердить добавление соответствующего исключения.

3. Ввести на отобразившейся странице логин и пароль. По умолчанию в системе прописан один пользователь с логином **administrator@service** и паролем **administrator.**

4. Нажать **Login.**

В случае, если система предложит сменить пароль, необходимо ввести новый пароль. После входа отобразится интерфейс Open Scape Assistant (рисунок 7).

В системе может быть прописано до 16 администраторов с несколько различающимися правами доступа. Рекомендуется при первом входе в систему добавить администратора с правами **Expert** (по умолчанию **administrator@service** имеет права **Advanced**). После этого необходимо выйти из системы (**Logoff**) и зайти под логином и паролем нового администратора.

Режим **Expert** отличается от режима **Advanced** возможностью доступа к режиму **Expert Mode.**

Большинство настроек системы производятся с использованием Мастеров – **Wizards**, которые доступны через пункт меню **Setup** на навигационной панели. Рассмотрим основные Мастера системы.

**Мастер базовой установки – Basic Installation** включает в себя следующие пункты подменю:

— Multibox System – настройка мультибоксовой системы;

— Initial Installation – начальные установки системы;

— Basic Installation – базовые установки системы;

— Licensing – активация лицензий;

— Networking Configurations – сетевые настройки.

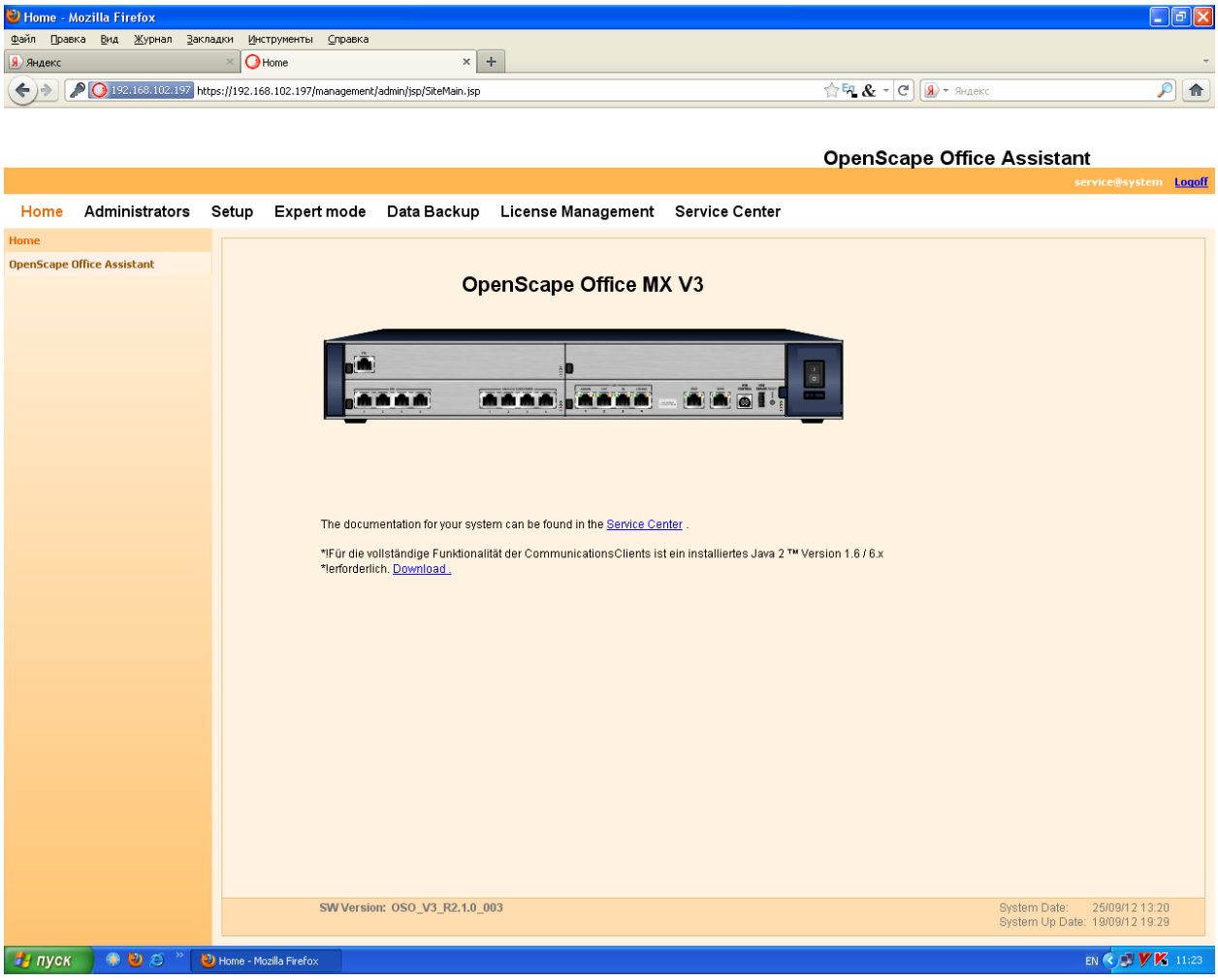



Рисунок 6.7 – Интерфейс Open Scape Assistant

**Мастер Сеть/Интернет – Network/Internet** включает в себя следующие пункты подменю:

— Network Configuration – настройки сети;

— Internet Configuration – настройки доступа в Интернет;

— VPN Configuration – настройки виртуальной частной сети VPN;

— WLAN Configuration – настройки беспроводной сети.

**Мастер Телефоны/Абоненты – Telephones/Subscribers** включает в себя:

— IP Telephones – прописывание в системе IP-телефонов;

— ISDN Devices – прописывание в системе ISDN-устройств;

— Analog Terminals – прописывание в системе аналоговых устройств;

— Key Programming – программирование функциональных клавиш терминалов Siemens.

**Мастер Центральная Телефония – Central Telephony** содержит следующие пункты:

— CO Trunk ISDN / Analog – настройка аналоговых транков и транков ISDN;

— Internet Telephony – настройка параметров соединения с провайдером Интернет-телефонии (ITSP);

— Voicemail – настройки голосовой почты;

— Directory / Speed Dialing – настройка параметров «быстрого» набора номера;

— Call Detail Recording

— Music on Hold / Announcements – настройка «музыки на удержании» и служебных сообщений, посылаемых вызывающему абоненту;

— Entrance Telephone (Door Opener) – используется для связи с домофоном.

**Мастер пользовательской телефонии – User Telephony** включает в себя:

— Class of Service – настройка классов обслуживания;

— Station Name and Release – настройка обозначения системы во внешнем окружении;

— Group call / Hunt group – настройка групп вызовов и групп поиска;

— Call Forwarding – настройка переадресации;

— Call Pickup – настройка групп «перехвата»;

— Team Configuration – настройка пользовательской группы («команды»);

— Mobile Phone Integration – настройка интеграции с мобильными устройствами;

— Executive / Secretary – настройка «Шеф-секретарской» группы;

— UCD

— Attendant Console – настройка «дежурных» номеров;

— Station Profiles

**Мастер UC Suite** рассмотрим ниже.

Базовая конфигурация системы.

При первом включении системы необходимо произвести начальные установки. Эти установки производятся с использованием пункта Initial Installation мастера **Basic Installation.**

**Определение имени системы и IP-адреса.**

При входе в пункт Initial Installation открывается окно **System Setting**, представленное на рисунке 6.8.

Необходимо указать адрес в рамках адресного пространства сети, в которой установлена система. Подтверждение новых параметров осуществляется нажатием клавиши **OK&Next.**

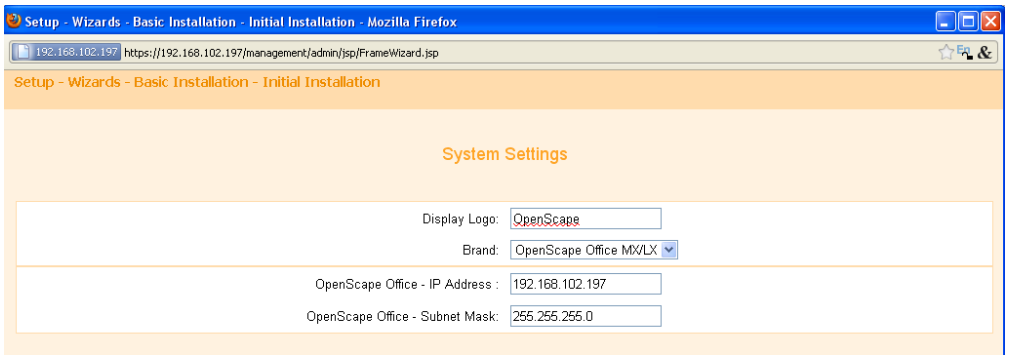



Рисунок 6.8 – Фрагмент окна окно **System Setting**

После нажатия данной клавиши происходит переход в следующее окно – **DHCP Global Setting** (рисунок 6.9).

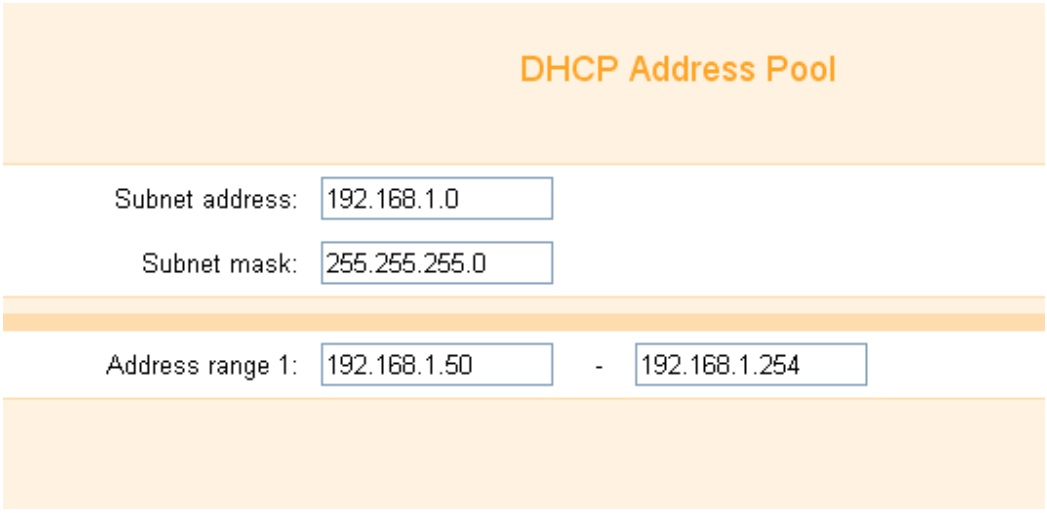



Рисунок 6.9 – Фрагмент окна **DHCP Global Setting**

В данном окне можно активировать внутренний DHCP-сервер системы и указать диапазон IP-адресов, которые будут впоследствии динамически раздаваться устройствам сети, например, IP-телефонам. В случае, если в сети, в которой установлена система, развернут собственный DHCP-сервер, данную функцию лучше не активировать во избежание конфликтов при адресации.

При нажатии клавиши **OK&Next** происходит переход в следующее окно – **Basic Configuration** (рисунок 6.10).

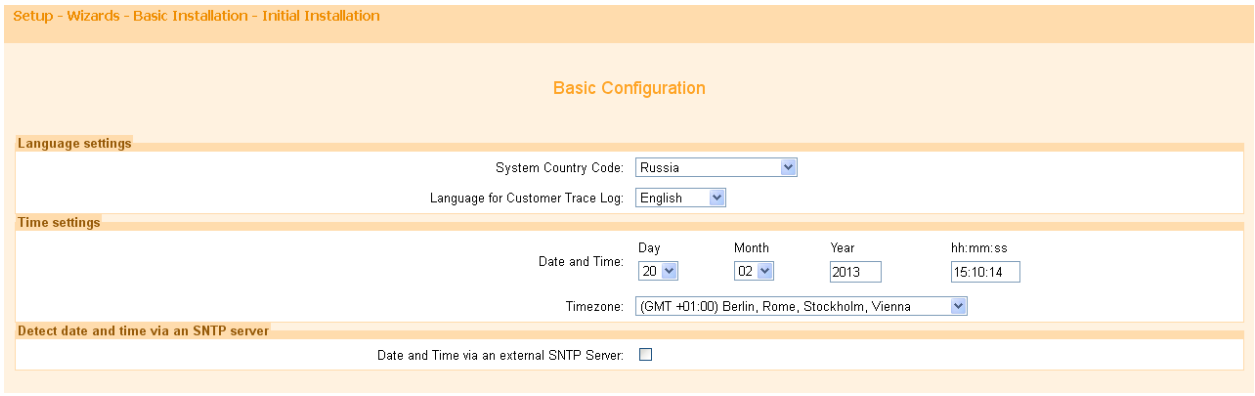



Рисунок 6.10 – Окно **Basic Configuration**

В данном окне устанавливаются параметры страны, дата и время. Дата и время, установленное в системе, будет затем отображаться на дисплее системных телефонов Siemens.

После внесения всех данных следует нажать клавишу **Finish.** Работа мастера Initial Installation завершится. При этом система перезагрузится, поэтому необходимо заново запустить Open Scape Assistant, задав в интернет-браузере новый IP-адрес.

Следующий подпункт данного мастера – Basic Installation (рисунок 6.11).

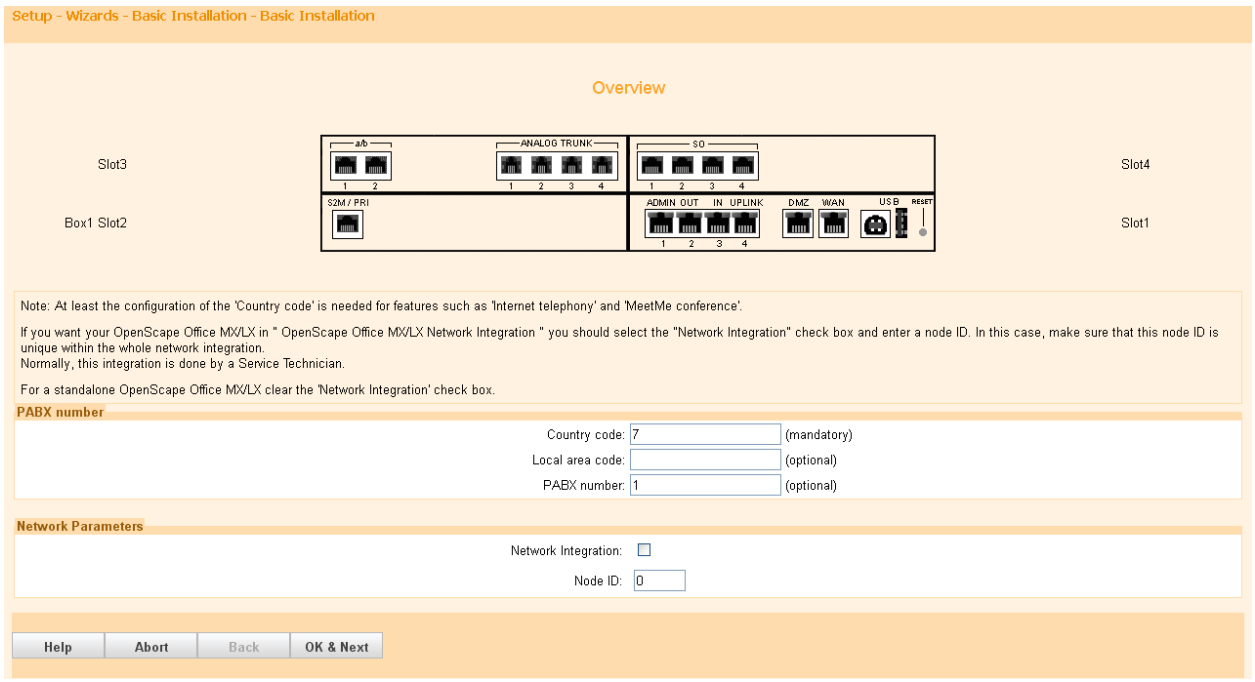



Рисунок 6.11 – Окно мастера Basic Installation

В данном методическом указании ограничимся пока только настройкой подключенных к системе аналоговых телефонов и IP-телефонов. Для настройки аналоговых телефонов используется окно **Select a station - A/B Phones**, рисунок 6.12.

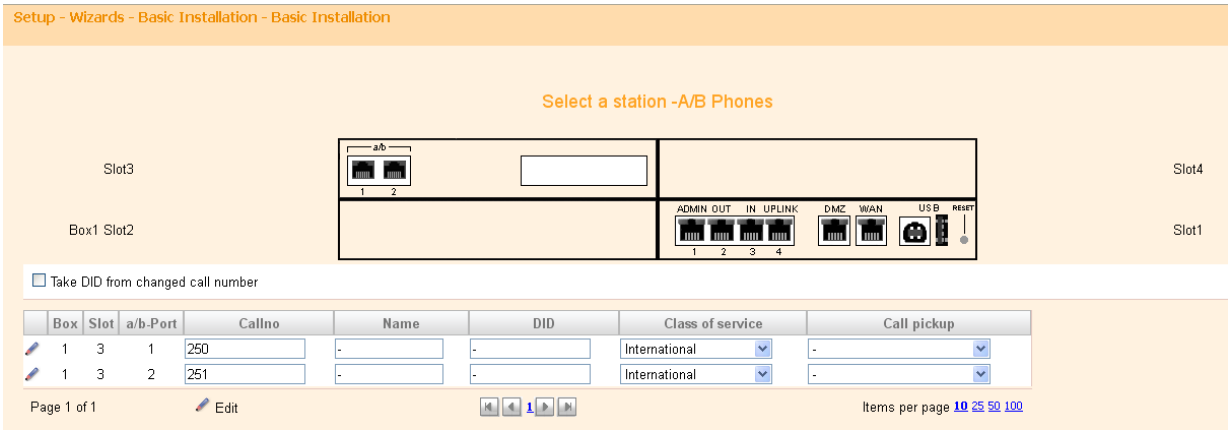



Рисунок 6.12 – Окно **Select a station - A/B Phones**

В данном окне необходимо назначить имена и номера аналоговым телефонам, при необходимости можно изменить некоторые другие параметры, например, установить различные виды звонков для внутренних и внешних вызовов. После нажатия кнопки **OK&Next** происходит переход в окно **Select a station - LAN Phones** (рисунок 6.13).

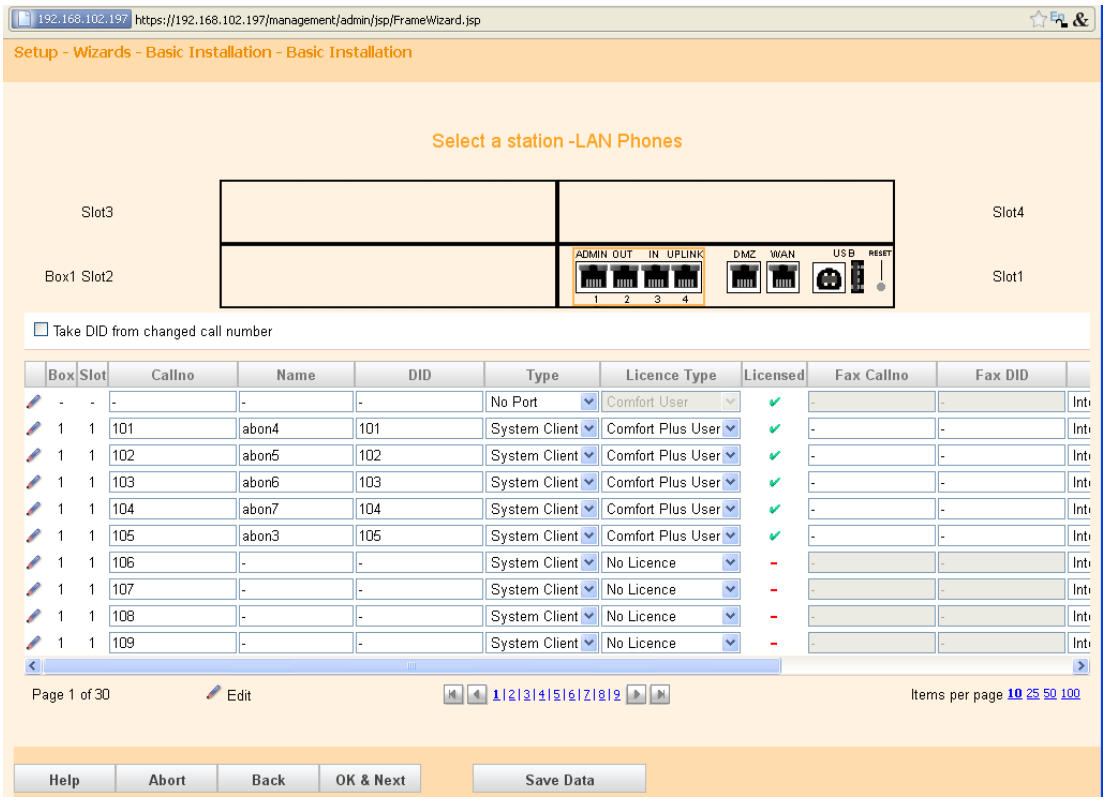



Рисунок 6.13 – Окно **Select a station - LAN Phones**

В данном окне настраиваются параметры IP-телефонов – имена абонентов (которые потом будут отображаться в телефонной книге, приложениях My Portal, My Attendant, на телефоне при входящем вызове и т.д.), тип телефона (например, System Client или SIP Client), вид лицензии и ряд других параметров.

После нажатия клавиши **OK&Next** происходит переход в окно **Edit Meet-Me Conference Settings**, рисунок 6.14.

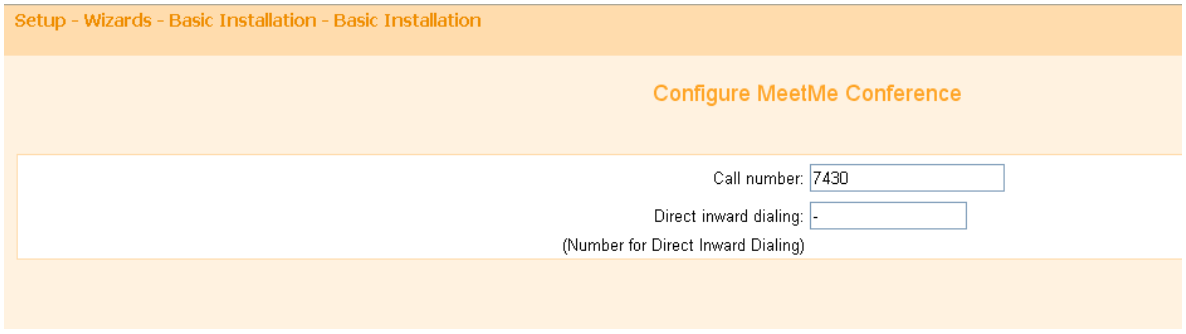



Рисунок 6.14 – Окно **Edit Meet-Me Conference Settings**

В данном окне настраиваются номера для конференц-связи. Подробности этой настройки рассмотрим позже.

Последнее окно данного мастера – **Edit E-Mail Forwarding** позволяет настроить параметры переадресации на электронную почту.

Затем необходимо подключить к системе все сконфигурированные устройства. Например, при использовании IP-телефона Siemens (System Client) необходимо указать IP-адрес системы OSO MX/LX, номер телефона, маску подсети и т.д. Настройку можно произвести как на самом телефоне через меню администратора (пароль по умолчанию – 123456), так и через Web-интерфейс, используя IP-адрес телефона.

**Задание:**

1. Произвести базовую конфигурацию системы – назначить IP-адрес, прописать IP и аналоговые телефоны, назначить абонентам имена в соответствии с указаниями преподавателя.
2. Подключить к системе аналоговые и IP-телефоны.
3. Произвести настройку и регистрацию в сети IP-телефонов.
4. Проверить работоспособность внутренней связи.

Настройка внешней связи.

Рассмотрим настройку внешней связи с сетями общего пользования (PSTN) с использованием ISDN и аналогового соединений.

Для настройки внешних линий (транков) используется Мастер **CO Trunk ISDN / Analog**, входящий в состав **Central Telephony**, рисунок 15.

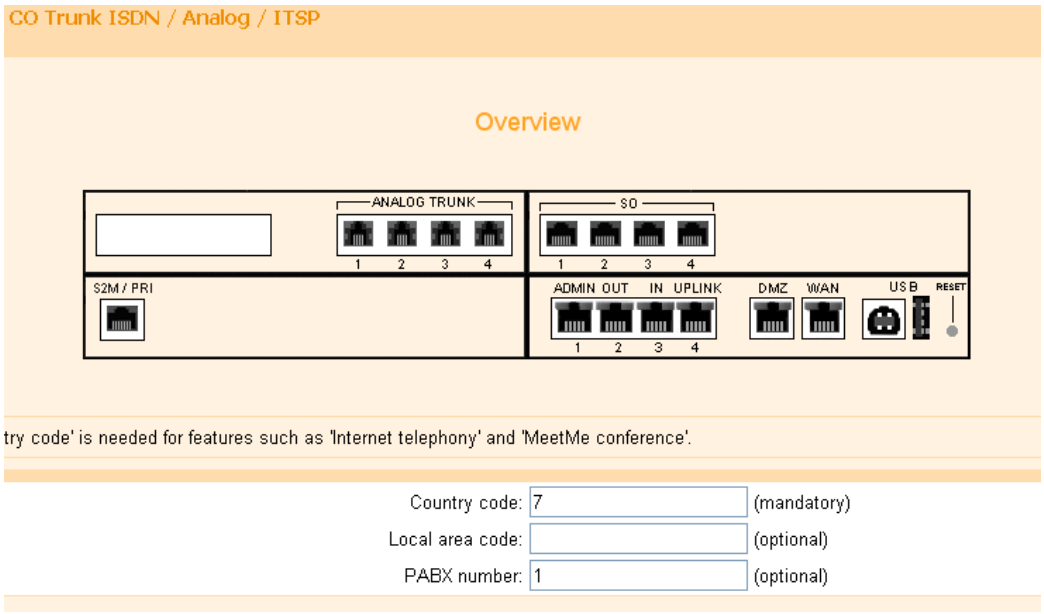



Рисунок 6.15 – Окно **CO Trunk ISDN / Analog**

Входящие вызовы можно перенаправлять на любые телефоны, прописанные в системе, а также на группы, конфигурацию которых рассмотрим ниже. Для назначения телефона, на который перенаправляется входящий вызов, используется **Expert Mode/Telephony Server/Trunks/Routing** (рисунок 6.16).

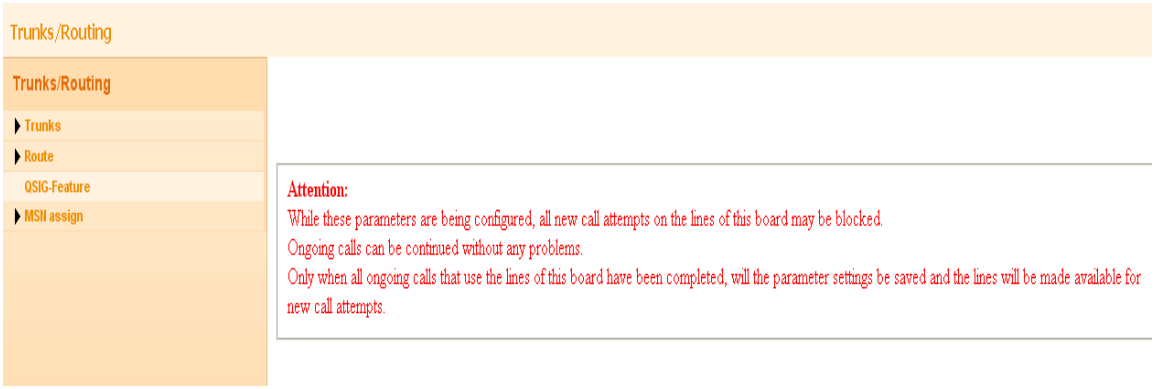



Рисунок 6.16 – Окно **Trunks/Routing**

При этом система выдает предупреждение о том, что во время конфигурирования параметров в данном окне вновь поступающие вызовы могут быть блокированы, имеющиеся вызовы продолжают обслуживаться.

Для примера рассмотрим случай, когда в системе используется внешний аналоговый транк с номером 1 (Box1, Slot1, port1). Выбрав соответствующий порт, увидим параметры выбранного транка (рисунок 6.17).


Рисунок 6.17 – Параметры аналогового транка

В нашем примере отображены параметры 39-й линии (ведется их сквозная нумерация в пределах данной системы). Из рисунка 17 видно, что данный транк относится к первой транковой группе, код выхода ##738, входящий вызов будет перенаправлен днем на номер 101, ночью – на номер 100.

Для выхода на внешнее направление с внутреннего телефона необходимо набрать код (в нашем случае ##738). Это не всегда удобно. Например, в нашей системе могут быть задействованы все 4 порта модуля TLANI, тогда каждая из четырех подключенных линий будет иметь свой код. Логичнее объединить все аналоговые линии в транковую группу (Trk Grp) и назначить один код для всей группы. Этот код называется кодом занятия линии (Seizure Code). При наборе этого кода с любого из внутренних телефонов занимается любая свободная линия в транковой группе.

Для этого необходимо включить линию в одну из транковых групп (на рисунке 17 – Trk Grp1), зайти в пункт **Route**, выбрать route 1 и назначить Seizure Code (например, 0), рисунок 6.18.

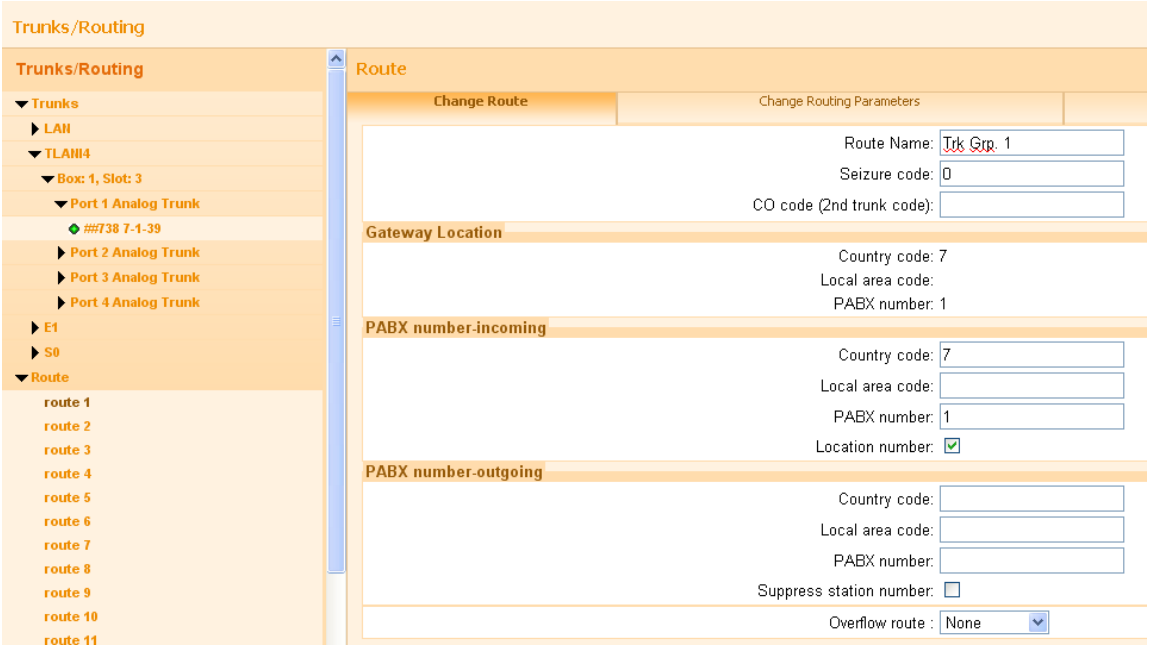



Рисунок 6.18 – Окно **Route**

Однако на практике, как правило, требуется перенаправить входящий вызов не на один из телефонов, а на несколько, установив правила распределения вызовов. Такая возможность предоставляется путем настройки групп.

**Практическое задание.**

1. Произвести подключение системы с использованием аналогового или ISDN-интерфейса (по указанию преподавателя).

2. Настроить исходящую связь с использованием префикса.

3. Настроить входящую связь, обеспечив направление внешнего вызова на телефоны, указанные преподавателем.

4. Проверить работоспособность системы при осуществлении входящих и исходящих вызовов.

Настройка пользовательских групп.

В системе Open Scape Office могут быть организованы, в зависимости от потребностей, несколько видов групп:

1. Группа перехвата – Pickup Group – поступивший вызов может быть «перехвачен» любым членом этой группы.

2. Группа для «общего» вызова – Group Call – вызов поступает на все телефоны, входящие в группу.

3. Группа поиска – Hunt Group – вызов последовательно либо циклически «обзванивает» телефоны, входящие в группу.

Группа перехвата настраивается в Мастере **Setup/User Telephony/Call Pickup**, рисунок 6.19.

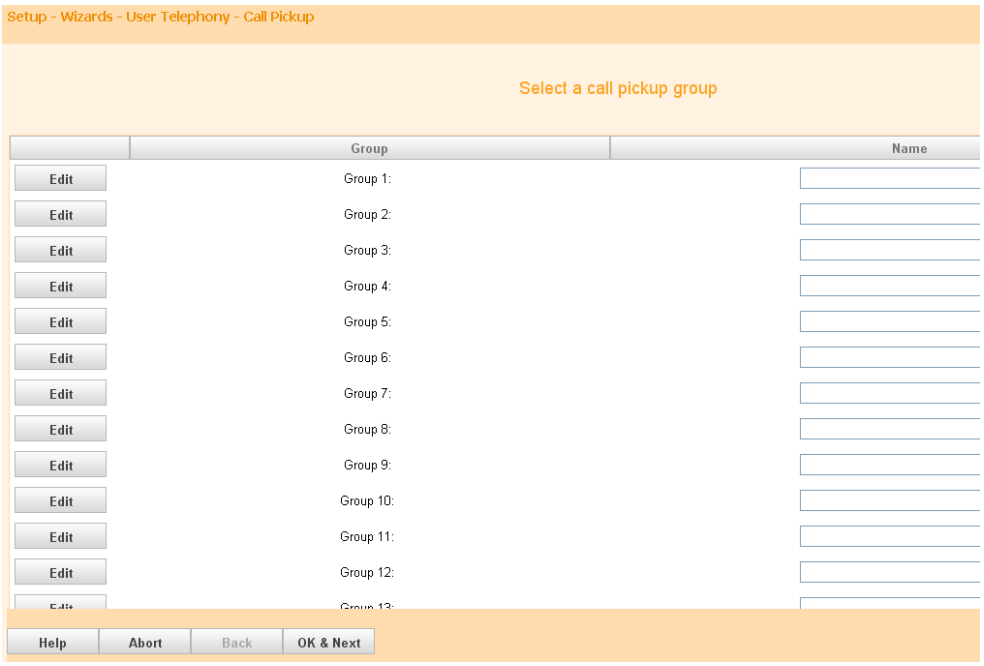



Рисунок 6.19 – Окно **User Telephony/Call Pickup**

В системе возможна настройка до 32 Pickup Group. Для добавления в группу участников в выбранной группе (например, Group 1) необходимо нажать Edit и в открывшемся списке поставить галочки в колонке Allocation Group 1 напротив телефонов, входящих в группу. При вызове любого телефона группы на остальных телефонах можно «перехватить» вызов. Приглашение для «перехвата» отображается на дисплее телефонных аппаратов Open Stage.

Группы для «общего» вызова и поиска настраиваются в Мастере **Setup/User Telephony/Group Call/Hunt Group**, рисунок 6.20.

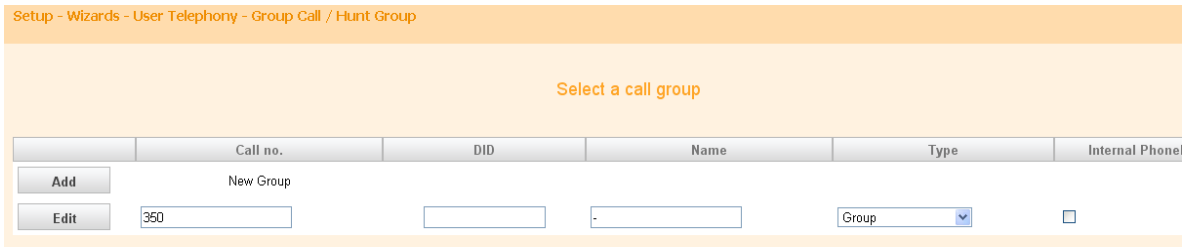



Рисунок 20 – Окно **User Telephony/Group Call/Hunt Group**

Для создания группы необходимо нажать Add (добавить), для изменения параметров – Edit (редактировать). Необходимо указать телефонный номер группы (например, 350), и добавить участников. В колонке Name можно указать имя группы (необязательный параметр), а в колонке Type – тип группы. При выборе типа Group создается группа «общего» вызова, при выборе типа Linear hunt group или Cyclical hunt group – группа поиска с линейным или циклическим обзвоном соответственно.

Необходимо отметить, что при настройке групп автоматически создается «ящик» голосовой почты, на который производится переадресация вызова в случае, если ни один из участников группы не ответил на вызов. При этом в дальнейшем необходимо настроить на него переадресацию «в случае неответа» (Мастер Call Forwarding будет рассмотрен ниже).

Кроме рассмотренных, в системе предусмотрены так называемые «командные» группы – Team Group – и шеф-секретарские группы Executive/Secretary or Top Group.

Группа универсального распределения вызовов – UCD group. Эта функция позволяет автоматически переключать входящие внутренние и внешние вызовы абоненту группы универсального распределения вызовов (агенту), который дольше всех не был занят.

Если все агенты UCD-группы заняты, входящий вызов устанавливается в очередь, при этом вызывающему абоненту проигрывается музыка на удержании (MOH) или воспроизводится сообщение (Announcements). Максимальное количество вызовов, установленных в очередь, может

устанавливаться индивидуально для каждой UCD-группы. Максимальное количество UCD-групп, которые можно организовать в одной системе, – 60.

Для конфигурирования UCD-групп используется Мастер **User Telephony/UCD**, рисунок 6.21.

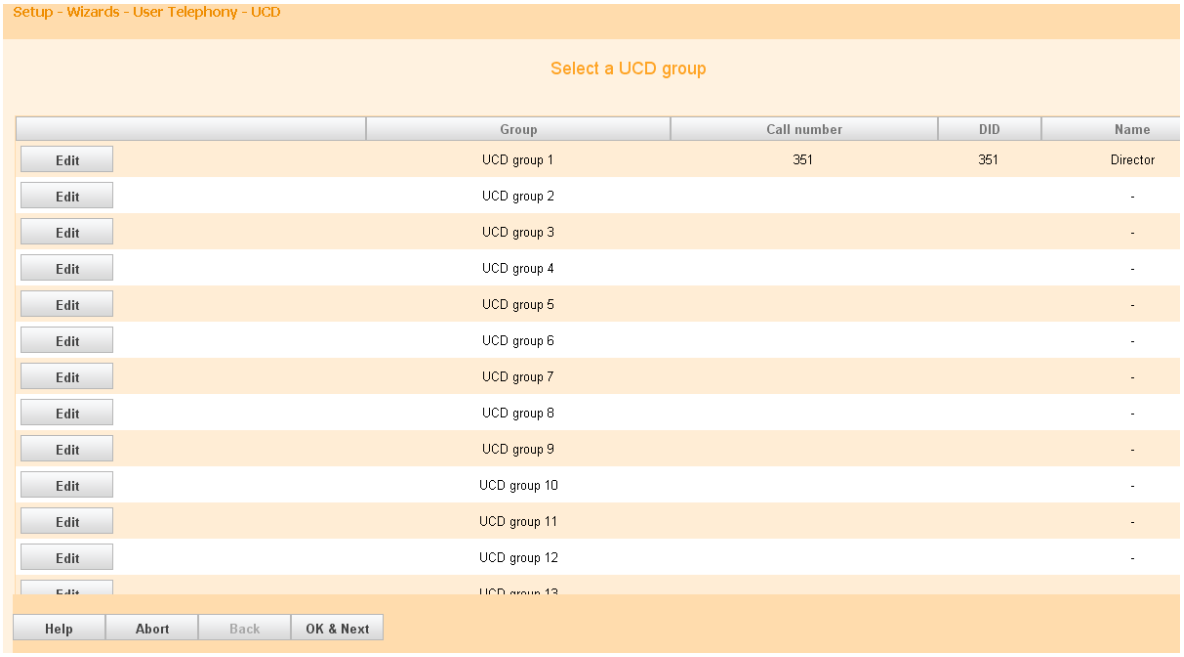



Рисунок 6.21 – Окно **User Telephony/UCD**

Напротив выбранной группы необходимо нажать клавишу Edit, после чего в появившемся окне необходимо ввести следующие параметры:

— номер группы;

— DID-номер группы;

— имя группы.

После нажатия клавиши **OK&Next** появляется окно добавления агентов в группу (Assign UCD agents), рисунок 6.22.

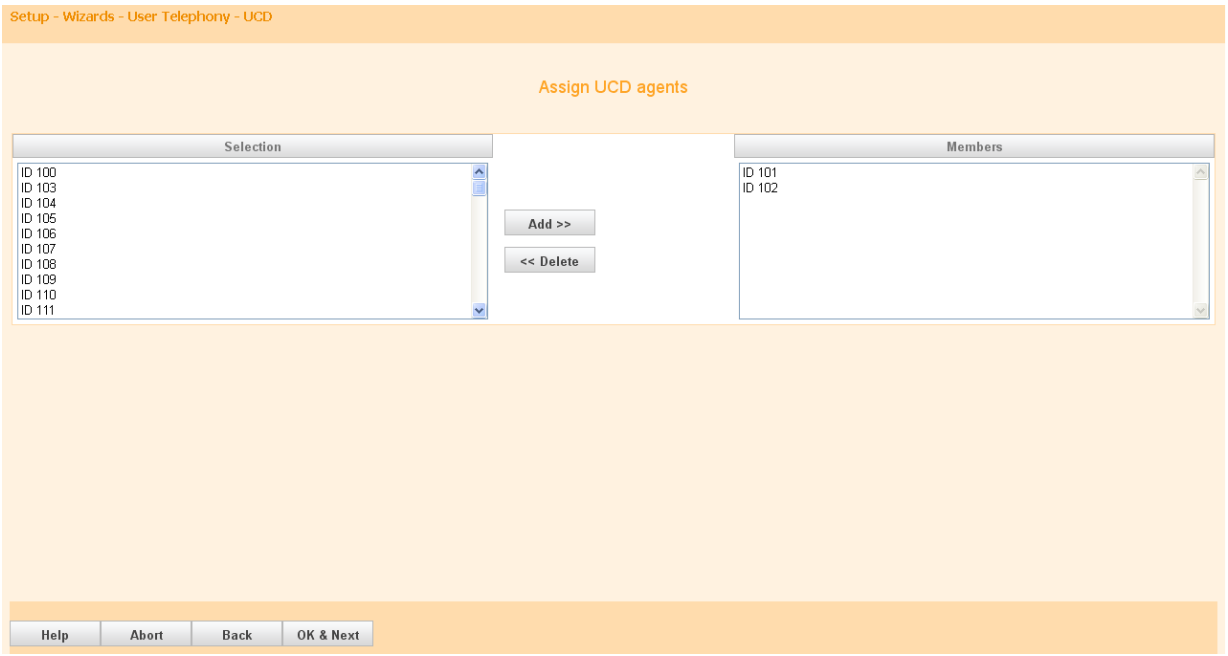



Рисунок 6.22 – Окно **Assign UCD agents**

Для добавления агента в группу необходимо выбрать его ID и нажать **Add**, для удаления агента из группы – **Delete.**

В примере, показанном на рисунке 22, в UCD Group 1 добавлены два агента – 101 и 102.

Агент UCD-группы может быть активен (Logon) или пассивен (Logoff). Для изменения статуса используется приложение My Agent, рассмотренное ниже.

Настройка UCD производится также в Мастере **Expert Mode/Incoming Calls/UCD**, рисунок 6.23.

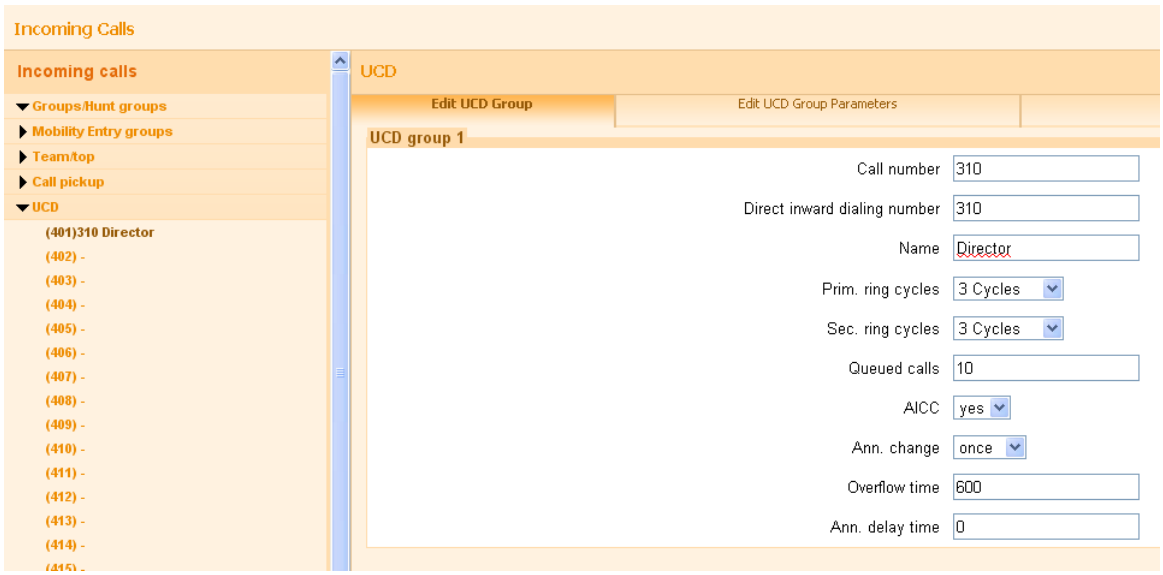



Рисунок 6.23 – Окно **Incoming Calls/UCD**

Для переадресации внешнего входящего вызова на UCD-группу необходимо установить переадресацию внешнего входящего вызова в окне **Call Forwarding**, рисунок 6.24.

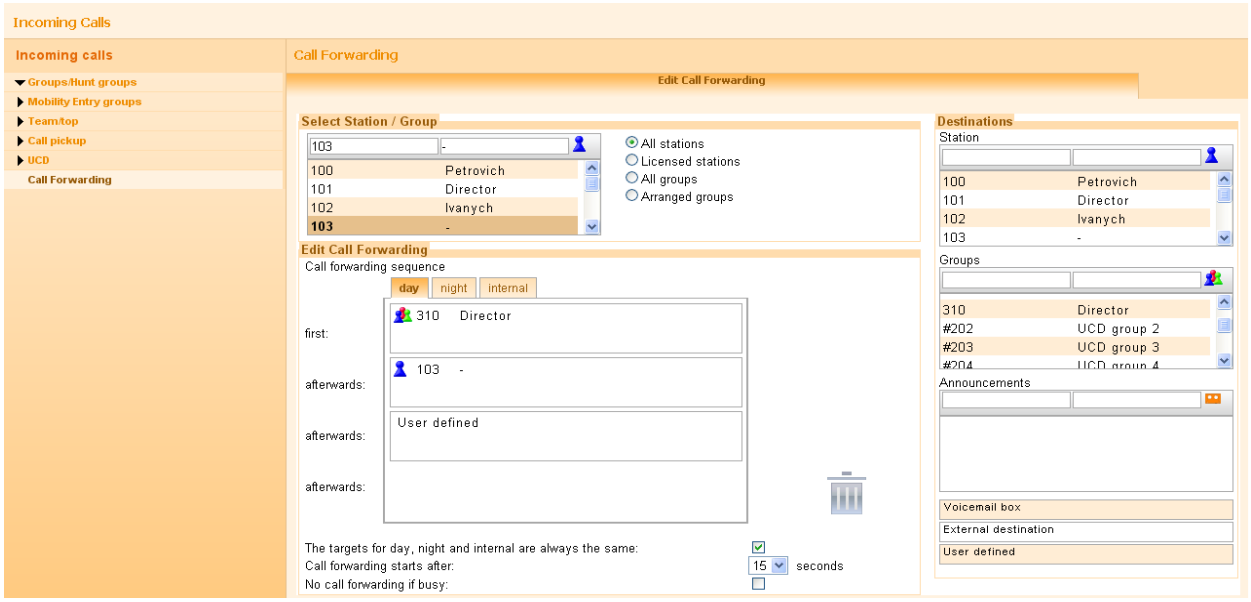



Рисунок 6.24 – Окно **Call Forwarding**

На примере, показанном на рисунке 6.24, вызов на номер 103 переадресовывается на UCD-группу с номером 310 и именем Director. Более

подробно UCD-группы рассмотрим ниже при изучении Контакт-центра системы.

**Задание:**

1. Настроить группы поиска и группу перехвата, добавив в них телефоны, указанные преподавателем.

2. Произвести настройку входящих вызовов на группу поиска и группу перехвата.

3. Произвести проверку корректности работы системы при внешних и внутренних вызовах к созданным группам.

4. Создать UCD-группы, добавив в них телефоны, указанные преподавателем.

5. Проверить корректность работы системы при входящем вызове, переадресованным на созданные UCD-группы.

**Список используемых источников:**

1. Манин А.А, Сосновский И.А. Системы коммутации. Принципы и технологии пакетной коммутации. Уч. пособие. 3е изд. Ростов-на-дону: СКФ МТУСИ, 2019. -245 с.

2. Тодд Лэммл, Шон Одом, Кевин Уоллес. CCNP. Маршрутизация. Учебное руководство. Издательство: Лори 2015 г .500 с.

2. http://linuxlearning.ooi.ru/mod/book/view.php?id=151&chapterid=192 – сайт системного администратора. Дата обращения 1.09.2019 г.