Министерство науки и высшего образования РФ
Севастопольский государственный университет

ИССЛЕДОВАНИЕ СПОСОБОВ НАЗНАЧЕНИЯ СПИСКОВ КОНТРОЛЯ ДОСТУПА В ЛОКАЛЬНЫХ КОМПЬЮТЕРНЫХ СЕТЯХ

Методические указания
к лабораторной работе №4
по дисциплине

«Архитектура (структуры и протоколы) инфокоммуникационных систем и сетей»

Для студентов, обучающихся по направлению 09.03.02
«Информационные системы и технологии»
по учебному плану подготовки бакалавров
дневной и заочной форм обучения

Севастополь
2019

УДК 004.732

Исследование способов назначения списков контроля доступа в локальных компьютерных сетях. Методические указания к лабораторным занятиям по дисциплине «Архитектура (структуры и протоколы) инфокоммуникационных систем и сетей» / Сост., В.С. Чернега, А.В. Волкова – Севастополь: Изд-во СевГУ, 2019 – 26 с.

Методические указания предназначены для проведения лабораторных работ по дисциплине «Архитектура (структуры и протоколы) инфокоммуникационных систем и сетей». Целью методических указаний является помощь студентам в исследовании способов назначения стандартных и расширенных списков контроля доступа (ACL). Излагаются теоретические и практические сведения необходимые для выполнения лабораторной работы, требования к содержанию отчета.

Методические указания рассмотрены и утверждены на методическом семинаре и заседании кафедры информационных систем

Рецензент: Моисеев Д.В., канд. техн. наук, доцент кафедры ИТиКС

1 ДИНАМИЧЕСКАЯ МАРШРУТИЗАЦИЯ

Статическая маршрутизация не подходит для больших, сложных сетей, поскольку обычно сети включают избыточные связи, многие протоколы и смешанные топологии. Маршрутизаторы в сложных сетях должны быстро адаптироваться к изменениям топологии и выбирать лучший маршрут из многих кандидатов.

IP сети имеют иерархическую структуру. С точки зрения маршрутизации сеть рассматривается как совокупность автономных систем.

Автономная система (AS – Autonomous System) – это система IP-сетей и маршрутизаторов, управляемых одним или несколькими операторами, имеющими единую политику маршрутизации с Интернетом. В автономных подсистемах больших сетей для маршрутизации на остальные автономные системы широко используются маршруты по умолчанию.

Динамическая маршрутизация может быть осуществлена с использованием одного и более протоколов. Эти протоколы часто группируются согласно того, где они используются. Протоколы для работы внутри автономных систем называют внутренними протоколами шлюзов (interior gateway protocols – IGP), а протоколы для работы между автономными системами называют внешними протоколами шлюзов (exterior gateway protocols – EGP). К протоколам IGP относятся RIP, RIP v2, IGRP, EIGRP, OSPF и IS-IS. Протоколы EGP3 и BGP4 относятся к EGP. Все эти протоколы могут быть разделены на два класса: дистанционно-векторные протоколы и протоколы состояния связи.

Маршрутизаторы используют метрики для оценки или измерения маршрутов. Когда от маршрутизатора к сети назначения существует много маршрутов, и все они используют один протокол маршрутизации, то маршрут с наименьшей метрикой рассматривается как лучший. Если используются разные протоколы маршрутизации, то для выбора маршрута используется административные расстояния, которые назначаются маршрутам операционной системой маршрутизатора.

Для того чтобы динамические протоколы маршрутизации обменивались информацией о статических маршрутах, следует осуществлять дополнительное конфигурирование.

1.1 Дистанционно-векторная маршрутизация

Дистанционно-векторная маршрутизация маршрутизация базируется на алгоритме Белмана-Форда. Через определенные моменты времени маршрутизатор передает соседним маршрутизаторам всю свою таблицу маршрутизации. Такие простые протоколы как RIP и IGRP просто распространяют информацию о таблицах маршрутов через все интерфейсы маршрутизатора в широковещательном режиме без уточнения точного адреса конкретного соседнего маршрутизатора.

Соседний маршрутизатор, получая широковещание, сравнивает информацию со своей текущей таблицей маршрутов. В нее добавляются маршруты к новым сетям или маршруты к известным сетям с лучшей метрикой. Происходит удаление несуществующих маршрутов. Маршрутизатор добавляет свои собственные значения к метрикам полученных маршрутов. Новая таблица маршрутизации снова распространяется по соседним маршрутизаторам (см. рисунок 1).

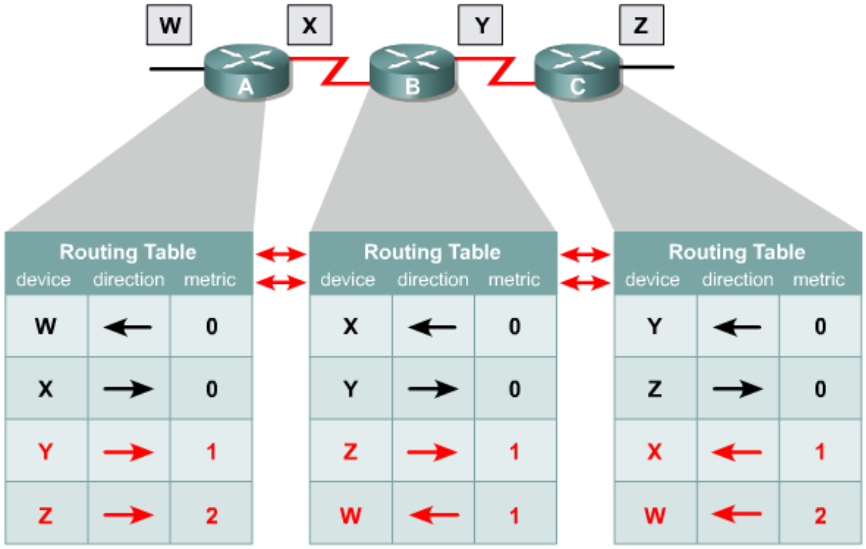


Рисунок 1 – Дистанционно-векторная маршрутизация

1.2 Протоколы состояния связи

Протоколы состояния связи предлагают лучшую масштабируемость и сходимость по сравнению с дистанционно-векторными протоколами. Протокол базируется на алгоритме Дейкстры, который часто называют алгоритмом «кратчайший путь – первым» (shortest path first – SPF). Наиболее типичным представителем является протокол OSPF (Open Shortest Path First).

Маршрутизатор берет в рассмотрение состояние связи интерфейсов других маршрутизаторов в сети. Маршрутизатор строит полную базу данных всех состояний связи в своей области, то есть имеет достаточно информации для создания своего отображения сети. Каждый маршрутизатор затем самостоятельно выполняет SPF-алгоритм на своем собственном отображении сети или базе данных состояний связи для определения лучшего пути, который заносится в таблицу маршрутов. Эти пути к другим сетям формируют дерево с вершиной в виде локального маршрутизатора.

Маршрутизаторы извещают о состоянии своих связей всем маршрутизаторам в области. Такое извещение называют LSA (link-state advertisements).

В отличие от дистанционно-векторных маршрутизаторов, маршрутизаторы состояния связи могут формировать специальные отношения со своими соседями.

Имеет место начальный наплыв LSA пакетов для построения базы данных состояний связи. Далее обновление маршрутов производится только при смене состояний связи или, если состояние не изменилось в течение определенного интервала времени. Если состояние связи изменилось, то частичное обновление пересылается немедленно. Оно содержит только состояния связей, которые изменились, а не всю таблицу маршрутов.

Администратор, заботящийся об использовании линий связи, находит эти частичные и редкие обновления эффективной альтернативой дистанционно-векторной маршрутизации, которая передает всю таблицу маршрутов через регулярные промежутки времени.

Протоколы состояния связи имеют более быструю сходимость и лучшее использование полосы пропускания по сравнению с дистанционно-векторными протоколами. Они превосходят дистанционно-векторные протоколы для сетей любых размеров, однако имеют два главных недостатка: повышенные требования к вычислительной мощности маршрутизаторов и сложное администрирование.

1.3 Сходимость

Этот процесс одновременно и совместный, и индивидуальный. Маршрутизаторы разделяют между собой информацию, но самостоятельно пересчитывают свои таблицы маршрутизации. Для того чтобы индивидуальные таблицы маршрутизации были точными, все маршрутизаторы должны иметь одинаковое представление о топологии сети. Если маршрутизаторы договорились о топологии сети, то имеет место их сходимость. Быстрая сходимость означает быстрое восстановление после обрыва связей и других изменений в сети. О протоколах маршрутизации и о качестве проектирования сети судят главным образом по сходимости.

Когда маршрутизаторы находятся в процессе сходимости, сеть восприимчива к проблемам маршрутизации. Если некоторые маршрутизаторы определили, что некоторая связь отсутствует, то другие ошибочно считают эту связь присутствующей. Если это случится, то отдельная таблица маршрутов будет противоречива, что может привести к отбрасыванию пакетов и петлям маршрутизации.

Невозможно, чтобы все маршрутизаторы в сети одновременно обнаружили изменения в топологии. В зависимости от использованного протокола, может пройти много времени пока все процессы маршрутизации в сети сойдутся. На это влияют следующие факторы:

– расстояние в промежуточных интерфейсах до точки изменения топологии;

– число маршрутизаторов, использующих динамические протоколы;

– полоса пропускания и загрузка каналов связи;

– загрузка маршрутизаторов.

Эффект некоторых факторов может быть уменьшен при тщательном проектировании сети.

1.4 Конфигурирование динамической маршрутизации

Для конфигурирования динамической маршрутизации используются две основные команды: `router` и `network`. Команда `router` запускает процесс маршрутизации и имеет форму:

```
Router(config)# router protocol [keyword]
```

где `protocol` – любой из протоколов маршрутизации: RIP, IGRP, OSPF и т.п., `keyword` – дополнительные параметры.

Затем необходимы команды `network`:

```
Router(config-router)#network network-number [keyword]
```

где `network-number` – идентифицирует непосредственно подключенную сеть, добавляемую в процесс маршрутизации;

`keyword` – дополнительные параметры.

`Network-number` позволяет процессу маршрутизации определить интерфейсы, которые будут брать участие в отсылке и приеме пакетов актуализации маршрутной информации.

Для просмотра информации о протоколах маршрутизации используется команда `show ip protocol`, которая выводит значения таймеров процессов маршрутизации и сетевую информацию, имеющую отношение к маршрутизации. Эта информация может использоваться для идентификации маршрутизатора, подозреваемого в поставке плохой маршрутной информации.

Содержимое таблицы IP-маршрутизации выводится командой `show ip route`. Она содержит записи про все известные маршрутизатору сети и подсети и указывает на способ получения этой информации.

1.4.1 Протокол OSPF

OSPF это динамический, иерархический протокол состояния связи, используемый для маршрутизации внутри автономных систем. Он базируется на открытых стандартах и был спроектирован как замена протоколу RIP. Он является развитием ранних версий протокола маршрутизации IS-IS. OSPF – устойчивый протокол, поддерживающий маршрутизацию с наименьшим весом и балансировку загрузки. Кратчайший путь в сети вычисляется по алгоритму Дейкстры. Cisco поддерживает свою версию стандарта OSPF.

Как только маршрутизатор настроен на работу с OSPF, он начинает процесс изучения окружения, проходя несколько фаз инициализации. В начале маршрутизатор использует Hello-сообщения для определения своих соседей и создания отношений для обмена обновлением маршрутной информацией с ними. Затем маршрутизатор начинает фазу ExStart начального обмена между базами маршрутов. Следующей является фаза обмена, в которой назначенный маршрутизатор отсылает маршрутную информацию и получает подтверждения от нашего нового маршрутизатора. В течение стадии загрузки, новый маршрутизатор компилирует таблицу маршрутов. По окончании вычислений маршрутизатор переходит в полное состояние, в котором он является активным членом сети.

Для запуска OSPF маршрутизации служит команда

```
Router(config)#router ospf N
```

где N – номер вычислительного процесса OSPF, он может быть различным для разных маршрутизаторов автономной системы. OSPF область Area организуется командой

```
Router(config-router)#network network-number area Area
```

и определяет автономную систему.

В OSPF `network-number` имеет особый формат. Для подключаемой в процесс маршрутизации сети используется инверсная маска. Так, чтобы сеть 212.34.0.0 255.255.0.0 поместить в область 7 OSPF маршрутизации следует дать команду

```
Router(config-router)#network 212.34.0.0 0.0.255.255 area 7
```

Команда `show ip ospf interface` для каждого интерфейса выводит всю OSPF информацию: IP адрес, область, номер процесса, идентификатор маршрутизатора, стоимость, приоритет, тип сети, интервалы таймера.

Команда `show ip ospf neighbor` показывает важную информацию, касающуюся состояния соседей.

Командой `show ip protocols` можно посмотреть с какими параметрами работает протокол OSPF.

2 СПИСКИ ДОСТУПА

Список управления доступом (access control list – ACL) – это последовательный список правил, которые используются для разрешения или запрета потока пакетов внутри сети на основании информации, приведенной внутри списка. Без списка доступа все пакеты внутри сети разрешаются без ограничений для всех частей сети. Список доступа может быть использован для контроля распространения и получения информации об изменении таблиц маршрутов и, главное, для обеспечения безопасности. По-

литика безопасности в частности включает защиту от внешних атак, ограничения доступа между отделами организации и распределение загрузки сети.

Список доступа позволяет использовать маршрутизатор как межсетевой экран, брандмауэр, для запрета или ограничения доступа к внутренней сети из внешней сети, например, Интернет. Брандмауэр, как правило, помещается в точках соединения между двумя сетями.

2.1 Стандартный ACL

При использовании стандартных ACL, единственным критерием для определения того, что пакет разрешен или запрещен, является IP адрес источника этого пакета. Формат элемента списка доступа следующий

```
Router(config)#access-list № permit | deny source-address source-mask
```

где № – целое число – номер списка доступа;
`source-address` – адрес источника пакета;
`source-mask` – маска в инверсной форме, накладываемая на адрес;
`permit` – разрешить прохождение пакета;
`deny` – запретить прохождение пакета.

Число № определяет принадлежность элемента списка доступа к определенному списку доступа с номером №. Первая команда `access-list` определяет первый элемент списка доступа, вторая команда определяет второй элемент списка доступа и т.д. Маршрутизатор обрабатывает каждый определенный в нем список доступа по элементам сверху вниз. То есть, если адрес `source-address` пакета с учетом маски удовлетворяет условию элемента списка, то дальнейшие элементы списка маршрутизатор не обрабатывает. Следовательно, для избегания лишней обработки, элементы, определяющие более общие условия, следует помещать в начале списка. Внутри маршрутизатора может быть определено несколько списков доступа. Номер стандартного списка должен лежать в диапазоне 1-99. Маска в списке доступа задается в инверсной форме, например маска 255.255.0.0 выглядит как 0.0.255.255.

Маршрутизаторы Cisco предполагают, что все адреса, не упомянутые в списке доступа в явном виде, запрещены. То есть в конце списка доступа присутствует невидимый элемент:

```
Router(config)#access-list № deny 0.0.0.0 255.255.255.255
```

Так, если необходимо разрешить только трафик от адреса 1.1.1.1 и запретить весь остальной трафик достаточно в список доступа поместить один элемент:

```
Router(config)#access-list 77 permit 1.1.1.1 0.0.0.0
```

Здесь предполагается, что был организован список доступа с номером 77.

Рассмотрим возможность применения стандартных списков доступа для диапазона адресов. Возьмем к примеру диапазон 10.3.16.0 – 10.3.31.255. Для получения инверсной маски необходимо вычесть из старшего адреса младший и получить 0.0.15.255. Тогда пример элемента списка можно задать командой

```
Router(config)#access-list 100 permit 10.3.16.0 0.0.15.255
```

Для того, чтобы список доступа начал выполнять свою работу, он должен быть применен к интерфейсу с помощью команды

```
Router(config-if)#ip access-group номер-списка-доступа in|out
```

Список доступа может быть применен либо как входной (in) либо как выходной (out). Когда список доступа применяется как входной, то маршрутизатор получает входной пакет и сверяет его входной адрес с элементами списка. Маршрутизатор разрешает пакету маршрутизироваться на интерфейс назначения, если пакет удовлетворяет разрешающим элементам списка либо отбрасывает пакет, если он соответствует условиям запрещающих элементов списка. Если список доступа применяется как выходной, то маршрутизатор получает входной пакет, маршрутизирует его на интерфейс назначения и только тогда обрабатывает входной адрес пакета согласно элементам списка доступа этого интерфейса. Далее маршрутизатор либо разрешает пакету покинуть интерфейс, либо отбрасывает его согласно разрешающим и запрещающим элементам списка соответственно. Так, созданный ранее список с номером 77 применяется к интерфейсу Ethernet 0 маршрутизатора как входной список командами

```
Router(config)#int Ethernet 0
Router(config-if)#ip access-group 77 in
```

Этот же список применяется к интерфейсу Ethernet 0 маршрутизатора как выходной список с помощью команд

```
Router(config-if)#ip access-group 77 out
```

Отменяется список на интерфейсе с помощью команды no

```
Router(config-if)#no ip access-group 77 out
```

Рассмотрим принцип создания более сложных списков доступа.
Пусть имеем сеть, представленную на рисунке 3.

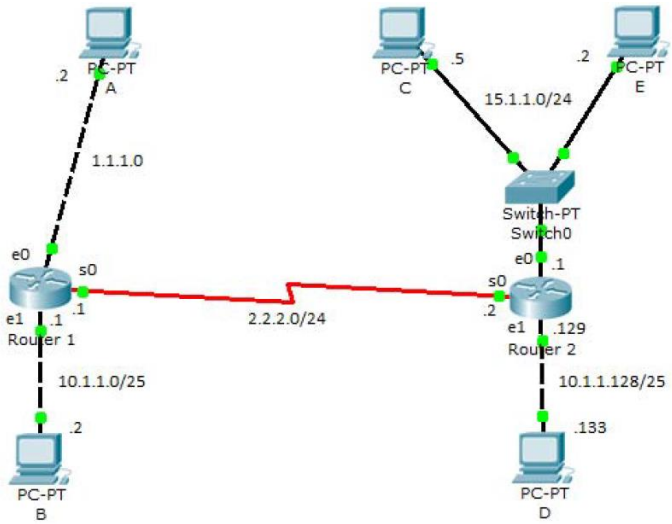


Рисунок 3 – Пример топологи сети для создания сложных списков доступа

Разрешим все пакеты, исходящие из сети 10.1.1.0 /25 (10.1.1.0 255.255.255.128), но запретим все пакеты, исходящие из сети 10.1.1.128 /25 (10.1.1.128 255.255.255.128). Также необходимо запретить все пакеты, исходящие из сети 15.1.1.0 /24 (15.1.1.0 255.255.255.0), за исключением пакетов от единственного хоста с адресом 15.1.1.5. Все остальные пакеты разрешаем. Списку дадим номер 2. Последовательность команд для выполнения поставленной задачи будет следующая

```
Router(config)#access-list 2 deny 10.1.1.128 0.0.0.127
Router(config)#access-list 2 permit 15.1.1.5 0.0.0.0
Router(config)#access-list 2 deny 15.1.1.0 0.0.0.255
Router(config)#access-list 2 permit 0.0.0.0 255.255.255.255
```

Отметим отсутствие разрешающего элемента для сети 10.1.1.0 255.255.255.128. Его роль выполняет последний элемент `access-list 2 permit 0.0.0.0 255.255.255.255.`

Удостоверимся, что поставленная задача выполнена.

1. Разрешить все пакеты, исходящие из сети 10.1.1.0 255.255.255.128. Последняя строка в списке доступа удовлетворяет этому критерию. Нет необходимости в явном виде разрешать эту сеть в нашем списке доступа так, как в списке нет строк, соответствующей этой сети за исключением последней разрешающей строки `permit 0.0.0.0 255.255.255.255.`

2. Запретить все пакеты, исходящие из сети 10.1.1.128 255.255.255.128. Первая строка в списке выполняет этот критерий. Важно отметить вид инверсной маски 0.0.0.127 для этой сети. Эта маска указывает, что мы не должны брать в рассмотрение последние семь бит четвертого октета адреса, которые назначены для адресации в данной подсети. Маска для этой сети 255.255.255.128, которая говорит, что последние семь бит четвертого октета определяют адресацию в данной сети.

3. Запретить все пакеты, исходящие из сети 15.1.1.0 255.255.255.0, за исключением пакетов от единственного хоста с адресом 15.1.1.5. Это требование удовлетворяется второй и третьей строкой нашего списка доступа. Важно отметить, что список доступа осуществляет это требование не в том порядке как оно определено. Обязательно следует помнить, что список доступа обрабатывается сверху вниз и при первом совпадении обработка пакетов прекращается. Мы вначале требуем запретить все пакеты, исходящие из сети 15.1.1.0 255.255.255.0 и лишь затем разрешить пакеты с адресом 15.1.1.5. Если в командах, определяющих список доступа мы, переставим вторую и третью команды, то вся сеть 15.1.1.0 будет запрещена до разрешения хоста 15.1.1.5. То есть, адрес 15.1.1.5 сразу же в начале будет запрещен более общим критерием `deny 15.1.1.0 0.0.0.255`.

4. Разрешить все остальные пакеты. Последняя команда разрешает все адреса, которые не соответствуют первым трем командам.

Таким образом, имеем следующую последовательность действий для воплощения списка доступа.

1. Определить критерии и ограничения для доступа.

2. Воплотить их с помощью команд `access-list`, создав список доступа с определенным номером.

3. Применить список к определенному интерфейсу либо как входящий, либо как исходящий.

Остановимся на последнем пункте. В общем случае стандартный список доступа следует помещать как можно ближе к точке назначения, а не к источнику пакетов. Хотя могут быть исключения. Так как стандартный список доступа работает только с исходными адресами, то не всегда возможна детальная конфигурация. Требуется приложить усилия, чтобы избежать возникновения не желаемых конфигураций доступа. Если список помещен вблизи источника пакетов, то очень вероятно, что доступ к устройствам, на которых не осуществляется никакая конфигурация доступа, будет затруднен.

Конкретизируем политику безопасности для сети на рисунке 5. Наша цель создать политику для компьютера A (адрес 1.1.1.2 сеть 1.1.1.0/24), которая изо всех устройств локальной сети 15.1.1.0/24 в которую входит компьютер C (15.1.1.5) разрешит доступ к компьютеру A лишь самого компьютера C. Мы также хотим создать политику, запрещающую удаленный доступ к компьютеру A из любого устройства локальной сети 10.1.1.128 / 25 компьютера D (10.1.1.133). Весь остальной трафик мы разрешаем. На рисунке 5 компьютер PC5 (15.1.1.5) играет роль произвольного отличного от компьютера C представителя локальной сети 15.1.1.0/24.

Размещение списка критично для воплощения такой политики. Возьмем созданный ранее список с номером 2. Если список сделать выходным на последовательном интерфейсе маршрутизатора 2, то задача для компьютера А будет выполнена, однако возникнут ограничения на трафик между другими локальными сетями. Аналогичную ситуацию получим, если сделаем этот список входным на последовательном интерфейсе маршрутизатора 1. Если мы поместим этот список как выходной на Ethernet A интерфейс маршрутизатора 1, то задача будет выполнена безо всяких побочных эффектов.

2.2 Расширенные ACL

Со стандартным ACL вы можете указывать только адрес источника, а маска не обязательна. В расширенных ACL вы должны указать и адрес приемника, и адрес источника с масками. Можете добавить дополнительную протокольную информацию для источника и назначения. Например, для TCP и UDP разрешено указывать номер порта, а для ICMP разрешено указывать тип сообщения. Как и для стандартных ACL, можно с помощью опции log осуществлять лог.

Общая форма команды для формирования строки списка расширенного доступа

access-list access-list-number {permit | deny} protocol source sourcewildcard [operator source-port] destination destination-wildcard [operator destination-port] [precedence precedence-number] [tos tos] [established] [log | log-input]

где `access-list-number` -100-199|2000-2699,

`protocol` – ip, icmp, tcp, gre, udp, igrp, eigrp, igmp, ipinip, nos и ospf.

Для порта source-port или destination-port можно использовать номер порта или его обозначение bgp, chargen, daytime, discard, domain, echo, finger, ftp, ftp-data, gopher, hostname, irc, klogin, kshell, lpd, nntp, pop2, pop3, smtp, sunrpc, syslog, tacacs-ds, talk, telnet, time, uucp, whois и www;

`operator` – это eq (равно), neq (не равно), gt (больше чем), lt (меньше чем), range – указывается два порта для определения диапазона (общепринятые номера портов представлена в таблице 1);

`precedence precedence - (0..7)` Первые 3-и бита поля TOS (тоже самое можно сделать через TOS);

`tos tos - (0..15)` Поле TOS IPv4 пакета (Type of service);

`log` – логирование на консоль (какой ACL, протокол, откуда и куда пакет пришел);

`log-input` – тоже что и log + интерфейс + MAC адрес отправителя.

Таблица 1 – Основные общепринятые номера портов

Порт / Протокол	Описание
20/TCP	протокол FTP – данные
21/TCP	протокол FTP – команды
22/TCP,UDP	протокол SSH
23/TCP,UDP	протокол Telnet
25/TCP,UDP	протокол SMTP
20/TCP	протокол FTP – данные
21/TCP	протокол FTP – команды
53/TCP,UDP	Domain Name System (DNS)
80/TCP	Hypertext Transfer Protocol (HTTP)
109/TCP	Post Office Protocol 2 (POP2)
110/TCP	Post Office Protocol 3 (POP3)
156/TCP,UDP	SQL Service
161/TCP,UDP	Simple Network Management Protocol (SNMP)
443/TCP	HTTP поверх TLS/SSL (HTTPS)

Порт / Протокол	Описание
445/TCP	Microsoft-DS Active Directory, Windows shares
520/UDP	Routing – RIP
1433/TCP,UDP	Microsoft SQL Server – Server
1434/TCP,UDP	Microsoft SQL Server – Monitor

Как и для стандартных ACL, расширенный ACL следует привязать к интерфейсу либо для входящего на интерфейс трафика

```
Router(config-if)# ip access-group №ACL in
```

либо для выходящего из интерфейса трафика

```
Router(config-if)# ip access-group №ACL out
```

здесь №ACL – номер списка.

2.2.1 Примеры элементов расширенного ACL

Разрешить SMTP отовсюду на хост

```
Router(config)#access-list 111 permit tcp any host 172.17.11.19 eq 25
```

Разрешить телнет отовсюду на хост

```
Router(config)#access-list 111 permit tcp any host 172.17.11.19 eq 23
```

`Any` – это специальное слово, которое означает адрес сети и обратную маску `0.0.0.0 0.0.0.0` и означает, что под правило подпадают абсолютно все узлы из любых сетей. Другое специальное слово – **`host`** – оно означает маску `255.255.255.255` – то есть именно один единственный указанный адрес.

Расширенный ACL позволяет очень тонко настроить права доступа.

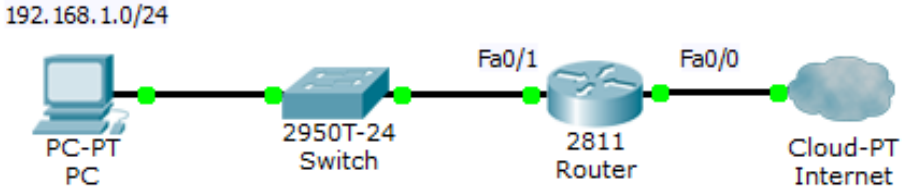
2.2.2 Применение established в ACL

Ключевое слово `established` используется в расширенных ACL для определения, принадлежит ли трафик к открытой TCP сессии. Маршрутизатор проверяет, соответствующий бит в заголовке TCP и принимает решение относительно того, относится ли трафик к уже установленному соединению.

Типичное использование established – организация доступа к интернету для сотрудников, чтобы извне нельзя было обращаться ко внутренней сети, но при этом ответы от веб серверов проходили вовнутрь нормально.

Established имеет ряд недостатков. Основной из них – работа только с протоколом TCP, так как используется его внутренний флаг. Если требуется работа с другими протоколами, следует использовать для этих же целей зеркальные (reflexive) ACL.

Допустим, есть топология:



Внутренняя сеть 192.168.1.0/24, надо обеспечить доступ из нее в интернет так чтобы ответы от серверов из интернета работали, но при этом обратиться вовнутрь извне было нельзя. Внутренняя сеть подключена к Fa0/1, внешняя – Fa0/0.

Настройка будет выглядеть следующим образом:

```
Router(config)#access-list 101 permit tcp 192.168.1.0 0.0.0.255 any eq 80
```

```
Router(config)#access-list 102 permit tcp any eq 80 192.168.1.0 0.0.0.255 established
Router(config)#interface fa0/1
Router(config-if)#ip access-group 101 in
Router(config-if)#interface fa0/0
Router(config-if)#ip access-group 102 in
```

Расширенный ACL 101 служит для выпуска трафика из пользовательской сети, он настроен на вход на интерфейса fa0/1. Он уничтожает весь трафик кроме того, что идет из сети на любой адрес на 80-ый порт.

ACL 102 используется на Fa0/0 – на вход. Когда из интернета приходит пакет, он проверяется сразу же этим ACL. Пропускается только трафик, идущий с 80-го порта на удаленном сервере (ответы от веб-серверов), только во внутреннюю сеть, и, самое главное только established трафик, то есть только трафик в рамках сессии которую установили мы изнутри.

В итоге, изнутри можно обращаться к сайтам и получать от них ответы, но если злоумышленник захочет подключиться к компьютеру внутри нашей сети (будучи сам снаружи), у него это не получится, даже если он будет пытаться подключиться с 80-го порта, так как при подключении он будет устанавливать новое соединение, и установить он его не сможет, так как в первых TCP сегментах не будет стоять необходимый флаг, соответственно, не произойдет TCP-рукопожатие.

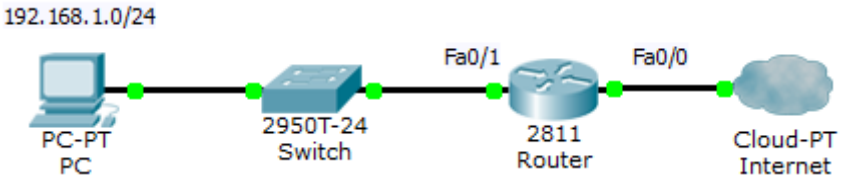
2.2.3 Reflexive ACL – зеркальные списки контроля доступа

Reflexive ACL – зеркальные списки контроля доступа, позволяют запоминать, кто обращался из нашей сети наружу (с каких адресов, с каких портов, на какие адреса, на какие порты) и автоматически формировать зеркальный ACL, который будет пропускать обратный трафик извне вовнутрь только в том случае, если изнутри было обращение к данному ресурсу.

Зеркальные (reflexive) ACL – это расширение технологии extended ACL, которое позволяет организовать пропуск трафика из интернета в локальную сеть только в ответ на предварительно сделанный запрос из локальной сети в интернет.

Технология эта напоминает внешне использование ключевого слова established, но имеется ряд отличий как в реализации, так и по функционалу. Суть технологии вот в следующем: на выход из сети ставится ACL, который выпускает трафик изнутри наружу. Одновременно с пропуском трафика, автоматически формируется встречный ACL, для пропуска трафика извне вовнутрь. Таким образом появляется возможность получать ответы на свои запросы из интернета.

Приведем пример: есть сеть 192.168.1.0/24 из нее надо организовать доступ в интернет по http, pop и smtp.



Пишется на выход следующие 2 ACL:

```
R1(config)#ip access-list extended IN-TO-OUT
R1(config-ext-nacl)#permit tcp 192.168.1.0 0.0.0.255 any eq www reflect BACK-WWW
R1(config-ext-nacl)#permit tcp 192.168.1.0 0.0.0.255 any eq pop3 reflect BACK-POP
R1(config-ext-nacl)#permit tcp 192.168.1.0 0.0.0.255 any eq smtp reflect BACK-SMTP
R1(config-ext-nacl)#exit
R1(config)#ip access-list extended OUT-TO-IN
R1(config-ext-nacl)#evaluate BACK-WWW
R1(config-ext-nacl)#evaluate BACK-POP
R1(config-ext-nacl)#evaluate BACK-SMTP
```

Применяем ACL

```
R1(config)#interface fa0/0
R1(config-if)#ip access-group OUT-TO-IN in
R1(config)#interface fa0/1
R1(config-if)#ip access-group IN-TO-OUT in
R1(config-if)#
```

IN-TO-OUT разрешает выход трафика изнутри наружу. Пропускается трафик на порты 25,80 и 110 параллельно формируются зеркальные ACL BACK-WWW, BACK-POP и BACK-SMTP, которые пропускают обратный трафик. Весь трафик извне фильтруется ACL OUT-TO-IN, который по умолчанию ничего не пропускает, но когда появляются зеркальные записи, то трафик начинает пропускаться.

Предположим, что человек обращается с адреса 192.168.1.100 к веб страничке на сервере 123.123.123.123 при обращении выбирается случайный порт отправителя (например, 1235), порт получателя используется стандартный – 80. Когда пакет проходит через маршрутизатор, он проверяется IN-TO-OUT. И по первой строчке проходит, одновременно в ACL BACK-WWW автоматически на время добавляется зеркальная запись:

```
permit tcp host 123.123.123.123 eq 80 host 192.168.1.100 eq 12345
```

То есть в настоящий момент весь трафик из интернета вовнутрь будет заблокирован, за исключением ответа от веб-сервера на наш запрос. Преимущество Reflexive ACL перед established заключается в том, что established пользуется только флагом в TCP сегменте, а Reflexive реально отслеживает соединения. Флаг можно подделать, в этом случае входящий трафик начнет пропускаться. Конечно, его вряд ли кто-то примет, но можно устроить, например, DOS атаку. Но самое важное преимущество, с помощью established в принципе нельзя организовать пропуск протоколов, отличных от TCP. Например, протоколов, базирующихся на UDP, или ICMP трафик. Зеркальные же ACL справляются с этими задачами отлично.

2.3 Именованные ACL

К именованным ACL обращаются по имени, а не по номеру, что дает наглядность и удобство для обращения. Для создания именованного ACL имеется команда

```
Router(config)#ip access-list standard|extended ACL_name
```

и далее команды для создания элементов списка

```
Router(config-ext-nacl)#permit|deny IP_protocol source_IP_address
wildcard_mask [protocol_information] destination_IP_address
wildcard_mask [protocol_information] [log]
```

Для завершения создания списка следует дать команду `exit`.

Имя именованного списка чувствительно к регистру. Команды для создания неименованного списка аналогичные командам для создания элементов нумерованного списка, но сам процесс создания отличен. Вы должны использовать ключевое слово `ip` перед главным ACL оператором и тем самым войти в режим конфигурации именно для этого именованного списка. В этом режиме вы начинаете с ключевых слов `permit` или `deny` и не должны вводить `access-list` в начале каждой строки.

Привязка именованных ACL к интерфейсу осуществляется командой

```
Router(config)#interface type [slot_№] port_№
Router(config-if)#ip access-group ACL_name in|out
```

ACL обрабатываются сверху вниз. Наиболее часто повторяющийся трафик должен быть обработан в начале списка. Как только обрабатываемый списком пакет удовлетворяет элементу списка, обработка этого пакета прекращается. Стандартные ACLs следует помещать ближе к точке назначения, где трафик должен фильтроваться. Выходные (`out`) расширенные ACLs следует помещать как можно ближе к источнику фильтруемых пакетов, а входные следует помещать ближе к точке назначения, где трафик должен фильтроваться.

Именованный ACLs разрешает вам себя редактировать. Для этого надо набрать команду, которая была использована для его создания

```
Router(config)#ip access-list standard|extended ACL_name
```

С помощью клавиш с вертикальными стрелками найти строку списка, которую вы хотите изменить. Изменить ее, используя горизонтальные стрелки. Нажать ввод. Новая строка добавится в конец списка. Старая не уничтожится. Для ее уничтожения следует ввести `no` в начале строки.

Для редактирования числовых ACLs следует его уничтожить и создать заново или изменить список офлайн и загрузить в устройство с помощью.

2.3.1 Пример именованного списка доступа

Создается стандартный списокк доступа с именем Internet_filter и расширенный список доступа с именем marketing_group:

```
Router(config)#interface Ethernet0/5
Router(config-if)#ip address 2.0.5.1 255.255.255.0
Router(config)#ip access-group Internet_filter out
Router(config-if)#ip access-group marketing_group in
Router(config)#ip access-list standard Internet_filter
Router(config-ext-nacl)#permit 1.2.3.4
Router(config-ext-nacl)#deny any
Router(config)#ip access-list extended marketing_group
Router(config-ext-nacl)#permit tcp any 171.69.0.0 0.0.255.255 eq telnet
Router(config-ext-nacl)#deny tcp any any
Router(config-ext-nacl)#permit icmp any any
Router(config-ext-nacl)#deny udp any 171.69.0.0 0.0.255.255 lt 1024
Router(config-ext-nacl)#deny ip any any log
```

2.3.2 Ограничение доступа к VTY при помощи ACL

ACL можно применять не только для фильтрации трафика, но и для ограничения адресов, с которых можно подключиться к маршрутизатору по telnet или ssh.

Сначала создается стандартный ACL, в котором перечисляем адреса и сети, из который доступ по telnet надо разрешить. Теперь его необходимо применить непосредственно на **`line vty 0 4`**, то есть, на линии виртульного терминала, к которым происходит подключение. Таким образом, не важно, через какой интерфейс маршрутизатора telnet-пакеты попадут на роутер, они будут отфильтрованы когда доберутся собственно до vty.

На маршрутизаторе создается стандартный список доступа **`VTY_ACCESS`**:

```
Router(config)#ip access-list standard VTY_ACCESS
Router(config-std-nacl)#permit 15.15.1.0 0.0.0.255
```

Устанавливается ограничение доступа к **VTY** на маршрутизаторе:

```
Router(config)#line vty 0 4
Router(config-line)#access-class VTY_ACCESS in
```

Теперь по telnet можно подключиться только из сети `15.15.1.0`.

Обратите внимание, что ACL применяется на интерфейсе командой `access-group`, а на vty – командой `access-class`.

3 ТРАНСЛЯЦИЯ СЕТЕВЫХ АДРЕСОВ

Network address translation (NAT – перенос сетевых адресов) создан для упрощения и сокрытия IP адресации. NAT позволяет представить внешнему миру внутреннюю структуру IP адресации предприятия иначе, чем она на самом деле выглядит. Это разрешает организации соединяться с Интернетом, не имея внутри себя глобальной уникальной IP адресации. Это даёт возможность выхода в Интернет для корпоративных внутренних IP сетей с внутренними IP адресами (intranet), которые глобально неуникальны и поэтому не могут маршрутизироваться в Интернете. NAT применяется также для связи территориально распределённых подразделений организации через Интернет.

Нужен NAT чаще всего для подключения вашей локальной сети к Интернету. Дело в том, что теоретически существует 255·255·255·255 = 4 228 250 625, т.е. 4 миллиарда адресов. Даже если бы у каждого жителя планеты был всего один компьютер, адресов бы уже не хватало. Так, ещё в начале 90-х было предложено разделить пространство адресов на публичные (белые) и приватные (частные, серые). Мировым сообществом для Интранет адресации были определены три диапазона серых адресов:

```
Class A (10.0.0.0/8): 10.0.0.0-10.255.255.255
Class B (172.16.0.0/12): 172.16.0.0-172.16.255.255
Class C (192.168.0.0/16): 192.168.0.0-192.168.255.255
```

NAT переводит внутренний IP адрес из внутреннего адресного пространства в IP адрес во внешнем адресном пространстве. Когда NAT получает пакет из intranet, он изменяет в нём адрес источника, пересчитывает контрольную сумму и отправляет его в Интернет.

NAT преобразует и отображает адреса из одной области в другую. Это обеспечивает прозрачную маршрутизацию от узла к узлу. В NAT существует несколько способов трансляции адресов, используемых в различных частных случаях.

Cisco использует для NAT специфическую терминологию для узлов в intranet и интернет как до, так и после преобразования адресов:

внутренний (inside) адрес – адрес, используемый в организации. Разные организации могут иметь одинаковые внутренние адреса;

внешний адрес (outside) – адрес, определённый где-либо вне данной организации. Внешний адрес иной организации может совпадать с внутренним адресом данной организации;

глобальный адрес – это зарегистрированный и законный адрес IP, который может проходить через Интернет;

локальный адрес – адрес IP, используемый внутренне в Intranet. Эти адреса не пересекают Интернет адреса и поэтому рассматриваются как локальные;

внутренний локальный адрес (inside local) – адрес, используемый в организации, не пересекающие Интернет адреса;

внутренний глобальный адрес (inside global) – адрес, используемый в организации, являющийся Интернет адресом;

внешний локальный адрес (outside local) – адрес, определённый где-либо вне данной организации, не являющийся Интернет адресом;

внешний глобальный адрес (outside global) – адрес, определённый где-либо вне данной организации, являющийся Интернет адресом.

Симулятор всегда показывает, что внешний локальный адрес (Outside Local) равен внешнему глобальному адресу (outside global).

При отправке пакетов от интерфейса внутреннего хоста NAT заменяет в нём адрес источника на некоторый глобальный адрес. При приёме ответного пакета NAT заменяет в нём глобальный адрес приёмника (адрес внешнего интерфейса локального маршрутизатора) на адрес интерфейса внутреннего хоста. Для такой замены маршрутизатор поддерживает специальные таблицы преобразований адресов, которые постоянно обновляются. Различают три способа преобразования адресов: *статический, динамический* и *перегрузка* (overload). При статическом NAT в явном виде с помощью команд IOS задаются пары «внутренний адрес – глобальный адрес». При динамическом преобразовании глобальные адреса берутся из определённого пула внешних адресов. При перегрузке все внутренние адреса, подлежащие преобразованию, заменяются на единственный глобальный адрес внешнего интерфейса маршрутизатора.

Для конфигурирования NAT следует определить на маршрутизаторе внутренние и внешние сети с помощью команд **`ip nat inside | outside`**. Эти команды определяются на уровне интерфейсов, то есть в контексте команды **`interface`**. Дополнительные команды зависят от используемого типа NAT. Это либо задание статического NAT, либо определение пула внешних адресов, либо задание команды для перегрузки. Как правило, следует также задать список управления доступом ACL для определения внутреннего трафика, который будет преобразовываться. Сам по себе ACL не осуществляет никакого NAT преобразования.

Процесс NAT прозрачен для внутренних адресов. Так хост с внутренним адресом, отправивший пакет во внешний мир и получивший ответ «не догадывается», что пакет прошел NAT преобразование на маршрутизаторе, как при отправке, так и при приёме. Внутреннему хосту представляется, что он имеет непосредственный выход во внешний мир.

Примечание: конфигурирование NAT осуществляется в режиме глобальной конфигурации маршрутизатора.

3.1 Конфигурация статической трансляции

В этом случае один внутренний адрес преобразуется в один внешний. И при этом все запросы, приходящие на внешний адрес, будут транслироваться на внутренний. Словно бы этот хост и является обладателем этого белого IP-адреса.

Для конфигурации статической трансляции необходимо выполнить следующие действия:

1. Установить режим статической трансляции между внутренним локальным (приватным) адресом и внутренним глобальным (публичным) адресом:

```
ip nat inside source static <локальный адрес> <глобальный адрес>
```

2. Указать внутренний интерфейс: **`interface`** `<тип> <номер>`
3. Пометить этот интерфейс, как принадлежащий внутренней сети: **`ip nat`** `inside`
4. Указать внешний интерфейс: **`interface`** `<тип> <номер>`
5. Пометить этот интерфейс, как принадлежащий внешней сети: **`ip nat`** `outside`

Такой подход бывает полезным, когда внутри сети есть сервер, к которому необходим полный доступ извне. Разумеется, этот вариант нельзя использовать, если в Ин-

тернет необходимо выпустить 300 хостов через один адрес. Такой вариант NAT'а никак не поможет сохранить белые IP-адреса, но тем не менее он бывает полезен.

3.2 Конфигурация динамической трансляции

У вас есть пул (диапазон) белых адресов, например, провайдер выделил вам сеть 198.51.100.0/28 с 16-ю адресами. Два из них (первый и последний) – адрес сети и широковещательный, ещё два адреса назначаются на оборудование для обеспечения маршрутизации. 12 оставшихся адресов вы можете использовать для NAT'а и выпускать через них своих пользователей. Ситуация похожа на статический NAT – один приватный адрес транслируется на один внешний, – но теперь внешний не чётко зафиксирован, а будет выбираться динамически из заданного диапазона.

Для конфигурации динамической трансляции необходимо выполнить следующие действия:

1. Определить пул (диапазон) глобальных адресов:

```
ip nat pool <имя> <первый адрес> <последний адрес>
[netmask <маска подсети> или prefix-length <длина префикса>]
```

2. Определить стандартный список доступа, регламентирующий адреса, подлежащие трансляции: **access-list** <номер> permit <адрес или блок адресов>

3. Установить динамическую трансляцию на основе списка доступа, определенного на предыдущем шаге:

```
ip nat inside source list <номер списка доступа> pool <имя>
```

4. Указать внутренний интерфейс: **interface** <тип> <номер>

5. Пометить этот интерфейс, как принадлежащий внутренней сети: **ip nat** inside

6. Указать внешний интерфейс: **interface** <тип> <номер>

7. Пометить этот интерфейс, как принадлежащий внешней сети: **ip nat** outside

Представленный ниже пример транслирует все адреса узлов-источников, определенных списком доступа 1 (разрешены адреса от 192.168.1.0/24), в пул адресов, названный net-208. Этот пул содержит адреса с 171.69.233.208 по 171.69.233.233.

```
ip nat pool net-208 171.69.233.208 171.69.233.233 netmask
255.255.255.240
ip nat inside source list 1 pool net-208
!
interface serial 0
ip address 171.69.232.182 255.255.255.240
ip nat outside
!
interface ethernet 0
ip address 192.168.1.94 255.255.255.0
ip nat inside
!
access-list 1 permit 192.168.1.0 0.0.0.255
```

Этот вариант тоже не универсальный, 300 пользователей так же не получится выпустить в Интернет, если конечно нет 300 внешних адресов. Как только белые адреса исчерпаются, никто новый уже не сможет получить доступ в Интернет. При этом те пользователи, что уже успели отхватить себе внешний адрес, будут работать.

Помимо динамического выделения внешних адресов, этот тип NAT отличается от статического тем, что без отдельной настройки проброса портов уже невозможно внешнее соединение на один из адресов пула.

По умолчанию таблица динамической трансляции адресов со временем очищается автоматически. Однако, имеется возможность проведения работ по мониторингу и сопровождению NAT с консоли управления маршрутизатором:

очистить все записи динамической трансляции адресов из таблицы NAT:

```
clear ip nat translation *
```

очистить простую запись динамической трансляции, содержащей информацию либо о внутренней трансляции, либо о внутренней и внешней трансляции:

```
clear ip nat translation inside <глобальный адрес> <локальный адрес>
        [outside <локальный адрес> <глобальный адрес>]
```

очистить простую запись динамической трансляции, содержащую информацию о внешней трансляции:

```
clear ip nat translation outside <локальный адрес> <глобальный адрес>
```

очистить расширенную запись динамической трансляции:

```
clear ip nat translation <протокол> inside <глобальный адрес> <глобальный порт> <локальный адрес> <локальный порт> [outside <локальный адрес> <локальный порт> <глобальный адрес> <глобальный порт>]
```

3.3 Использование одного внутреннего глобального адреса

Этот тип имеет несколько названий: NAT Overload, Port Address Translation (PAT), IP Masquerading, Many-to-One NAT. Последнее название говорит само за себя – через один внешний адрес выходит в мир много приватных. Это позволяет решить проблему с нехваткой внешних адресов и выпустить в мир всех желающих.

Существует возможность экономии пула внутренних глобальных адресов путем разрешения маршрутизатору использовать один глобальный адрес для трансляции нескольких локальных адресов. Если используется такой вариант конфигурации, то маршрутизатор использует информацию протоколов более высокого уровня (например, TCP и UDP) для обратной трансляции глобального адреса в корректные локальные адреса. При использовании соответствия нескольких локальных адресов одному глобальному адресу номера портов TCP или UDP каждого внутреннего узла указывают на локальные адреса этих узлов.

Для конфигурирования режима использования одного внутреннего глобального адреса необходимо выполнить следующие шаги:

1. Определить стандартный список доступа:

```
access-list <номер> permit <внутренний адрес или блок адресов>
```

2. Установить режим динамической трансляции адресов, разрешенных в списке доступа, определенном на предыдущем шаге:

```
ip nat inside source list <номер списка доступа>
        interface <тип> <номер> overload
```

3. Указать внутренний интерфейс: **interface** <тип> <номер>

4. Пометить этот интерфейс, как принадлежащий внутренней сети: **ip nat** inside

5. Указать внешний интерфейс: **interface** <тип> <номер>

6. Пометить этот интерфейс, как принадлежащий внешней сети: **ip nat** outside

4 ПЕРЕНАПРАВЛЕНИЕ ПОРТОВ

4.1 Понятие сетевых портов и сокетов

Основные прикладные сетевые сервисы используют средства транспортного уровня для взаимодействия. Любые 2 сетевых процесса могут идентифицировать друг друга при помощи 3-х компонент: IP-адрес, протокол(TCP/UDP), порт. Часто данные компоненты носят название *сокетов*. Сокеты – это название программного интерфейса для обеспечения информационного обмена между процессами. Т.е., для прикладных сетевых процессов взаимодействие осуществляется через сокеты.

Порт – параметр протоколов TCP и UDP, определяющий пункт назначения для данных, принимаемых по сети. Порту сопоставляется номер от 1 до 65535, позволяющие различным программам, выполняемым на одном хосте, получать данные независимо друг от друга. В этом случае каждая из них обрабатывает данные, поступающие на определённый порт (иногда говорят, что программа «слушает» на том или ином порту).

Согласно IP, в каждом пакете присутствуют IP-адрес узла-источника и IP-адрес узла-назначения. В TCP/UDP пакетах дополнительно указываются порт источника и порт назначения. Узел назначения, получив пакет, смотрит на порт назначения и передает пакет соответствующему у себя приложению. Использование портов позволяет независимо использовать TCP/UDP протокол сразу многим приложениям на одном и том же компьютере.

Для сетевых приложений нотация указания порта следующая: «ip:port». Например, http://web-service.org:8888.

Пояснение понятия портов представлено на рисунке 4. На самом деле сетевой порт – это всего лишь числовой параметр в сетевом пакете протоколов TCP и UDP. Такие понятия как «открыть порт» означают, что пакеты, адресованные на данный порт будут приниматься на обработку.

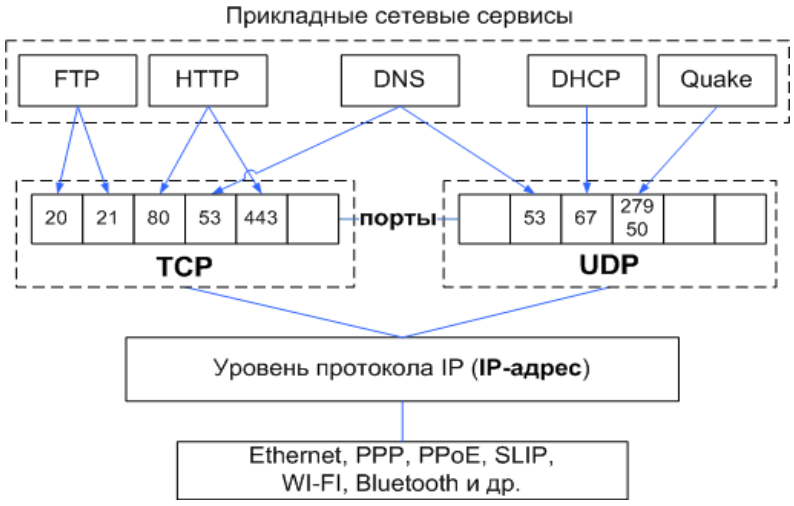


Рисунок 4 – Компоненты сокетов

Порты из диапазона 1-1024 являются привилегированными. Называются они так, потому что для их открытия (и, соответственно, запуска соответствующих сетевых сервисов) на большинстве ОС требуются права системного администратора. Большая часть привилегированных портов распределена для общеупотребительных сетевых протоколов. В таблице 2 перечислены некоторые протоколы и порты, за которыми они

закреплены. Данные порты являются портами по умолчанию для соответствующих служб и чаще всего не перенастраиваются.

Таблица 2 – Примеры некоторых стандартных сетевых портов

Порт	Протокол	Служба	Описание
20	TCP	ftp-data	Порт данных FTP
21	TCP	ftp	Порт протокола передачи файлов (File Transfer Protocol, FTP); иногда используется протоколом файловой службы (File Service Protocol, FSP)
22	TCP	ssh	Служба Безопасной ОБолочки (Secure SHell, SSH)
23	TCP	telnet	Служба Telnet
25	TCP	smtp	Протокол простой передачи почты (Simple Mail Transfer Protocol, SMTP)
53	UDP	DNS	Службы доменных имён (такие как BIND)
67	UDP	Bootps/DHCP	Порт сервера загрузки конфигурационной информации
68	UDP	Bootpc/DHCP	Порт клиента, получающего конфигурационную информацию
69	UDP	TFTP	Простейший протокол пересылки файлов
80	TCP	http	Протокол передачи гипертекста (HyperText Transfer Protocol, HTTP) для служб всемирной паутины (World Wide Web, WWW)
110	TCP	pop3	Протокол почтового отделения (Post Office Protocol) версии 3
111	UDP	SunRPC	Вызов удаленных процедур
119	TCP	NNTP	Протокол пересылки сетевых новостей
123	UDP	NTP	Служба Network Time Protocol
137	UDP	NetBIOS-NS	Служба имен NetBIOS
139	TCP	NETBIOSSSN	Служба сеанса NetBIOS
161	UDP	SNMP	Получение сетевых запросов обслуживания
162	UDP	SNMP-trap	Получение отчетов о проблемах в сети
443	TCP	https	Безопасный протокол пересылки гипертекстовых страниц
445	TCP	MDS/SMB	MDS (Microsoft Directory Services) SMB (Server Message Blocks over IP) - Служба поддержки TCP/IP NetBIOS
992	TCP	telnets	Telnet поверх SSL (TelnetS)
993	TCP	imaps	IMAP поверх SSL (IMAPS)
994	TCP	ircs	IRC поверх SSL (IRCS)
995	TCP	pop3s	POP 3 поверх SSL (POP3S)

4.2 Конфигурирование перенаправления портов

При использовании NAT трансляция один-в-один и все запросы, приходящие извне автоматически перенаправлялись на внутренний хост. Таким образом, можно было бы выставить сервер наружу в Интернет. Но если такой возможности нет? Можно указать, что все запросы, приходящие на конкретный белый адрес и конкретный порт маршрутизатора, должны быть перенаправлены на нужный порт нужного внутреннего адреса.

```
Router(config)#ip nat inside source static tcp|udp
<local address> <local port> <global address> <global port>
```

Применение данной команды означает, что TCP-запрос, пришедший из интернета на адрес `<global address>` по порту `<global port>`, будет перенаправлен на внутренний адрес `<local address>` на порт `<local port>`. Разумеется, можно пробрасывать и UDP и делать перенаправление с одного порта на другой. Это, например, может оказаться полезным, если у вас есть два компьютера, к которым нужен доступ по RDP извне. RDP использует порт 3389. Один и тот же порт вы не можете пробро-

сить на разные хосты (при использовании одного внешнего адреса). Поэтому вы можете сделать так:

```
Router(config)# ip nat inside source static tcp 172.16.6.61 3389 198.51.100.2 3389
Router(config)# ip nat inside source static tcp 172.16.6.66 3389 198.51.100.2 3398
```

Тогда, чтобы попасть на компьютер 172.16.6.61 вы запускаете RDP-сессию на порт 198.51.100.2:3389, а на 172.16.6.66 – 198.51.100.2:3398. Маршрутизатор сам раскидает всё, куда надо.

4.3 Примеры настройки

4.3.1 Доступ на WEB-сервер

Тут работает политика запрещено всё, что не разрешено. Поэтому сейчас надо что-то открыть, а всё остальное закрыть. Поскольку необходимо защитить сеть серверов, то и список досткпа необходимо вешать на интерфейс, идущий в их сторону. Если мы не хотим пускать пакеты в сторону серверов, которые уже оказались на маршрутизаторе, то это будет исходящий трафик. То есть, адреса назначения (destination) у нас будут в сети серверов (из них мы будем выбирать на какой именно сервер идёт трафик), а адреса источников (source) могут быть любыми – как из нашей корпоративной сети, так и из интернета. Ещё одно замечание: поскольку фильтровать мы будем в том числе по адресу назначения (на WEB-сервер одни правила, на почтовый – другие), то список контроля доступа понадобится расширенный (extended), только он позволяет делать это.

Итак, первое правило: разрешить доступ всем по порту 80

```
Router(config)#ip access-list extended Servers-out
Router(config-ext-nacl)#remark WEB
Router (config-ext-nacl)#permit tcp any host 172.16.0.2 eq 80
```

Разрешаем (**permit**) TCP-трафик от любого узла (**any**) на хост (**host** – именно один адрес) 172.16.0.2, адресованный на 80-й порт. Пробуем повесить этот список доступа на интерфейс FE0/0:

```
Router(config)#int fa0/0
Router(config-if)#ip access-group Servers-out out
```

Страничка открывается, но сервер не пингуется. Дело в том, что после всех правил в цисковских ACL в конце дописывается неявное *deny ip any any* (implicit deny). Любой пакет, выходящий с интерфейса и не отвечающий ни одному правилу из ACL, подпадает под implicit deny и отбрасывается. То есть хоть пинг, хоть фтп, хоть что угодно здесь уже не пройдёт.

4.3.2 Полный доступ администратору

Надо дать полный доступ компьютеру, с которого будет производиться управление. Это будет компьютер администратора с адресом 172.16.6.66 из внутренней сети. Каждое новое правило добавляется автоматически в конец списка, если он уже существует:

```
Router(config)#ip access-list extended Servers-out
Router(config-ext-nacl)#permit tcp host 172.16.6.66 host 172.16.0.2 range 20 ftp
Router(config-ext-nacl)#permit tcp host 172.16.6.66 host 172.16.0.2 eq telnet
```

Вот и всё. Проверяем с нужного узла (поскольку серверами в PT не поддерживается телнет, проверяем на FTP):

```
PC>ftp 172.16.0.2
```

4.3.3 Доступ на файловый сервер

Тут бы надо в первую очередь определиться с тем, кто будет «резидентом», кому нужно дать доступ. Конечно, это те, кто имеет адрес из сети 172.16.0.0/16 – только им и дадим доступ.

Теперь с общими папками. В большинстве современных систем уже используется для этого протокол SMB, которому нужен порт TCP 445. На более старых версиях использовался NetBios, который кормился аж через три порта: UDP 137 и 138 и TCP 139. Договорившись с нашим админом, настроим 445 порт (правда проверить в рамках PT, конечно, не получится). Но кроме этого, нам понадобятся порты для FTP – 20, 21, причём не только для внутренних хостов, но и для соединений из интернета:

```
Router(config)#ip access-list extended Servers-out
Router(config-ext-nacl)#permit tcp 172.16.0.0 0.0.255.255 host 172.16.0.3 eq 445
Router(config-ext-nacl)# permit tcp any host 172.16.0.3 range 20 21
```

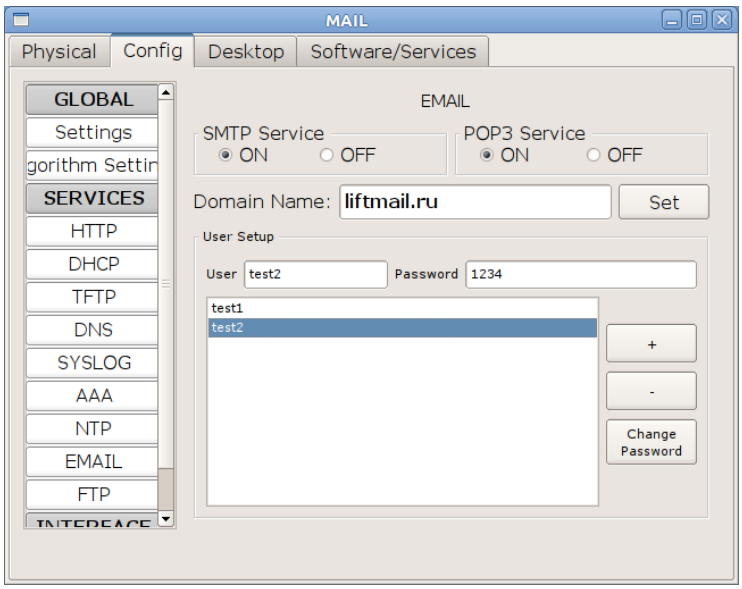
Тут применяется конструкцию **range 20 21** – для того, чтобы в одной строке задать несколько портов. Для FTP, вообще говоря, недостаточно только 21-го порта. Дело в том, что если открыть только его, то авторизация будет проходить, а передача файлов нет.

4.3.4 Доступ на почтовый сервер

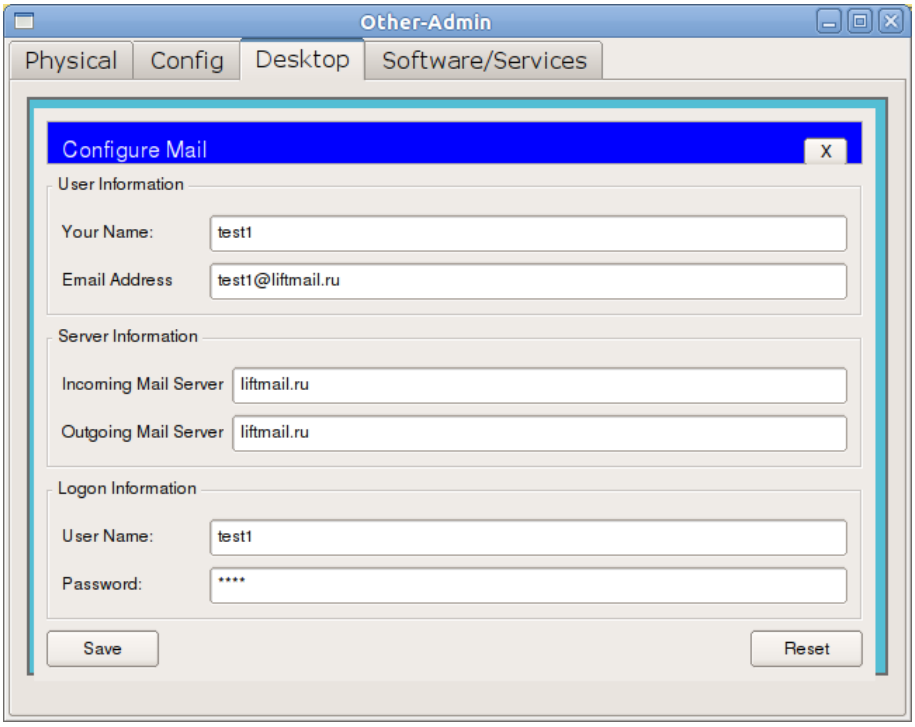
В рамках того же списка доступа добавляем новые нужные нам записи. Вместо номеров портов для широкораспространённых протоколов можно указывать их имена:

```
Router(config)#ip access-list extended Servers-out
Router(config-ext-nacl)#permit tcp any host 172.16.0.4 eq pop3
Router(config-ext-nacl)#permit tcp any host 172.16.0.4 eq smtp
```

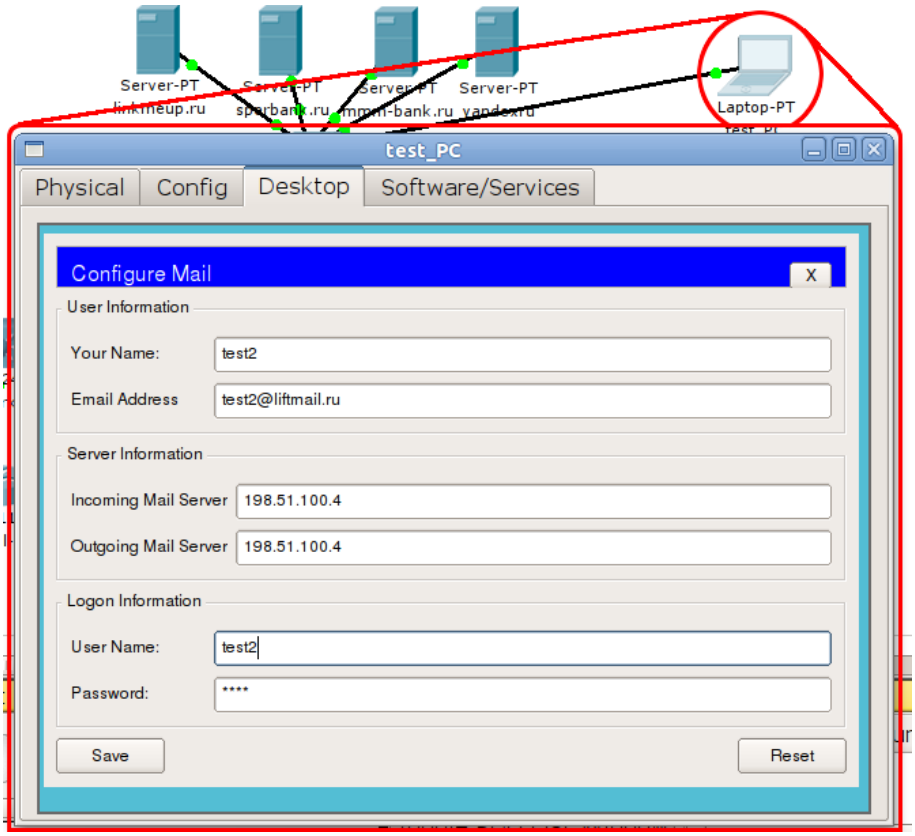
Проверим работу почтового сервера. Сначала настроим его. Указываем домен и создаём двух пользователей.



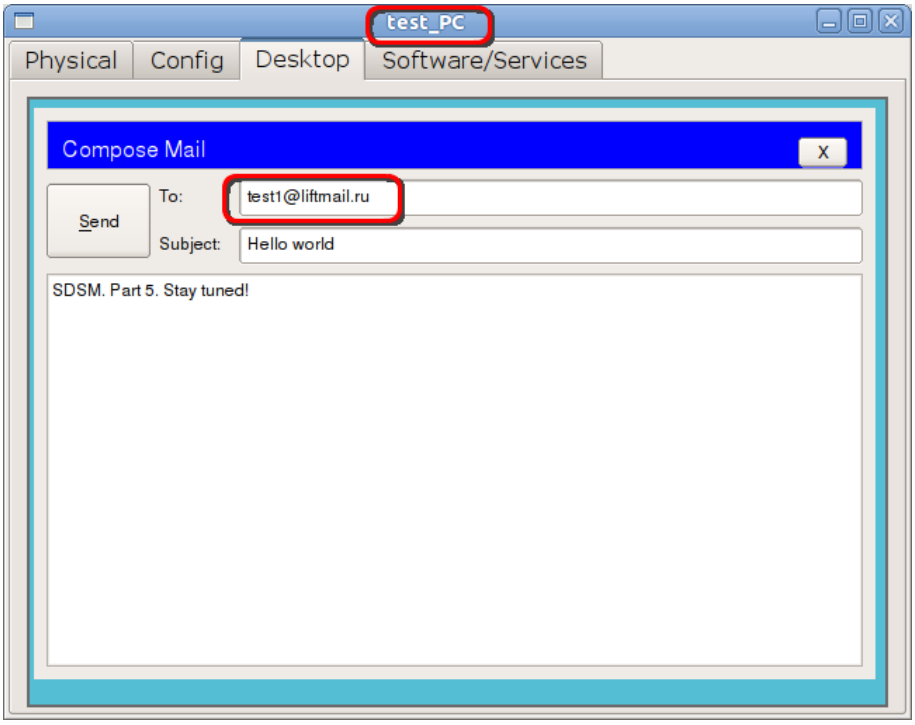
Настраиваем компьютер из внутренней сети:



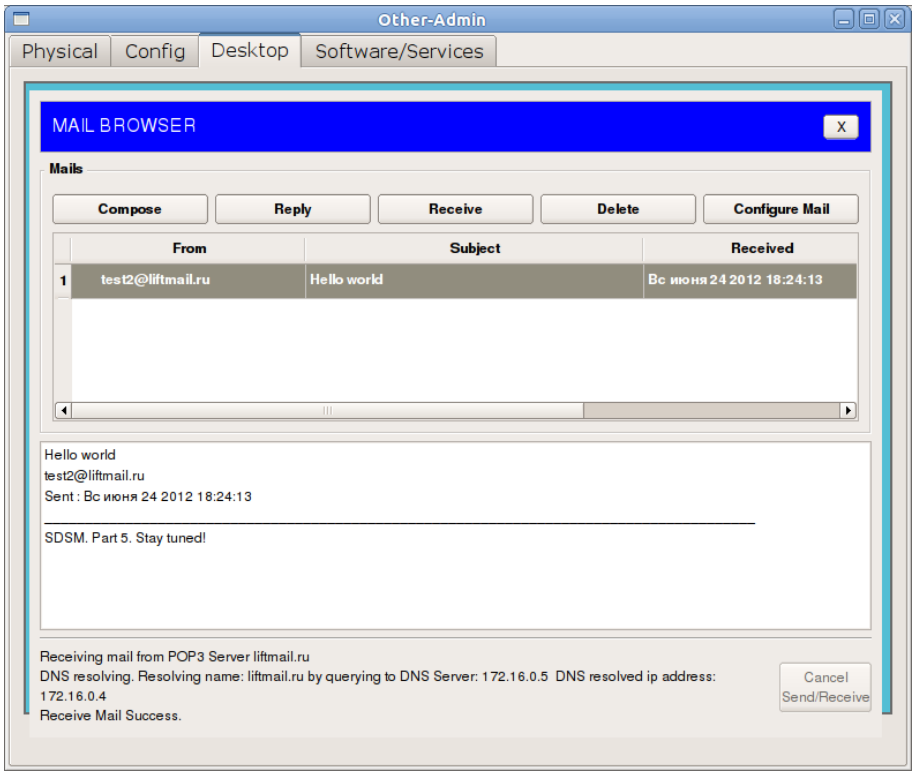
Из внешней:



Готовим письмо:

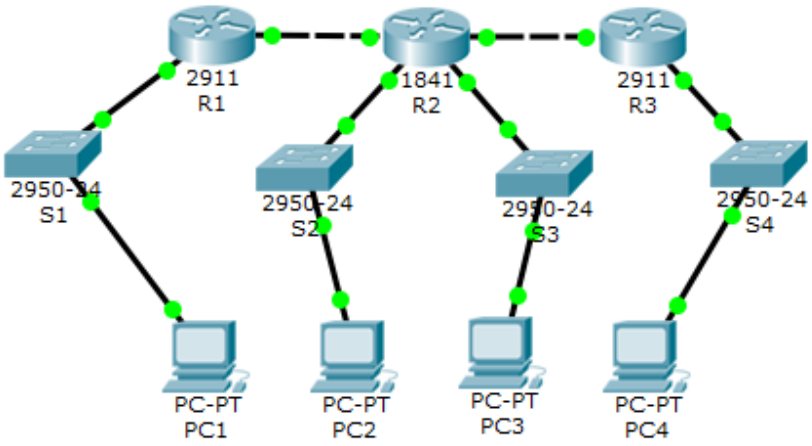


На локальном хосте нажимаем Receive:



5 ПРОГРАММА ВЫПОЛНЕНИЯ РАБОТЫ

1. В программе Cisco Packet Tracer необходимо построить сеть, представленную на рисунке ниже, настроить динамическую маршрутизацию с помощью протокола OSPF и обеспечить возможность взаимодействия конечных устройств PC1, PC2, PC3 и PC4 между собой.



Планирование адресного пространства необходимо выполнить самостоятельно. Результат необходимо занести в таблицу, представленную ниже.

Таблица сетевых адресов

Устройство	Интерфейс	IP-адрес	Маска	Шлюз
R1				
R2				
R3				
PC1				
PC2				
PC3				
PC4				

Проверьте настройки динамической маршрутизации:

просмотрите содержимое таблицы IP маршрутизации с помощью команды `show ip route`;

на каждом компьютере необходимо выполнить команду трассировки `tracert` других компьютеров;

просмотрите параметры протокола OSPF с помощью команд `show ip ospf interface`, `show ip ospf database` и `debug ip ospf events`.

2. В соответствии с Вашим вариантом курсового проекта сконфигурируйте политики безопасности на маршрутизаторе / L3-коммутаторе.

3. Настройте трансляцию сетевых адресов с перегрузкой для локальной сети в соответствии с Вашим вариантом курсового проекта. Объяснить какое устройство было выбрано для конфигурирования и почему.

4. Настройте перенаправление портов для всех внешних серверов в соответствии с Вашим вариантом курсового проекта.

6 СОДЕРЖАНИЕ ОТЧЕТА

Титульный лист.

Исходные данные в соответствии с индивидуальным вариантом.

Описание всех использованных команд.

Скриншоты топологии, реализованных настроек строек и результатов исследования функционирования сети.

Выводы.

7 КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Что такое автономная система?
2. Что такое метрика?
3. Какие существуют классы протоколов динамической маршрутизации?
4. Объясните работу дистанционно-векторных протоколов.
5. Объясните работу протоколов состояния связи.
6. В чём преимущества и недостатки дистанционно-векторных протоколов и протоколов состояния связи?
7. Как узнать, какие протоколы маршрутизации запущены на маршрутизаторе?
8. Перечислите основные этапы установки маршрутизатора.
9. Опишите схему работы протокола OSPF
10. Как на маршрутизаторе запустить и настроить протокол маршрутизации OSPF?
11. Как получить информацию об источнике маршрута удаленных сетей?
12. Какие возможны ошибки при настройке динамической маршрутизации?
13. Как выявлять ошибки настройке динамической маршрутизации?
14. Как просмотреть таблицу соседних устройств? Какую информацию о ней можно получить?
15. Опишите все возможные схемы работы службы NAT.
16. Какие частные IP адреса используются службой NAT в каждом классе адресов?
17. Перечислите преимущества и недостатки службы NAT.
18. Перечислите этапы настройки службы NAT.
19. Опишите схему проверки работы службы NAT.
20. Опишите основные проблемы в работе сервера NAT.