

## Лабораторная работа 7 Списки доступа ACL. Настройка статического и динамического NAT

Списки доступа (access-lists) используются в целом ряде случаев и являются механизмом задания условий, которые роутер проверяет перед выполнением каких-либо действий. *Маршрутизатор* проверяет каждый пакет и на основании вышеперечисленных критериев, указанных в *ACL* определяет, что нужно сделать с пакетом, пропустить или отбросить. Типичными критериями являются адреса отправителя и получателя пакета, тип протокола. Каждый критерий в списке доступа записывается отдельной строкой. *Список* доступа в целом представляет собой набор строк с критериями, имеющих один и тот же номер (или имя). Порядок задания критериев в списке существенен. Проверка пакета на соответствие списку производится последовательным применением критериев из данного списка (в том порядке, в котором они были введены). Пакет, который не соответствует ни одному из введенных критериев будет отвергнут. Для каждого протокола на *интерфейс* может быть назначен только один *список* доступа. Как пример ниже приведена *таблица* списка управления доступом по умолчанию:

№ правила	Подсеть	Конечная точка	Разрешить или запретить
100	0.0.0.0/0	3387	Разрешить

**Без ACL** - по умолчанию при создании конечной точки ей все разрешено.

**Разрешить** - при добавлении одного или нескольких диапазонов "разрешения" все остальные диапазоны по умолчанию запрещаются. Только пакеты из разрешенного диапазона *IP*-адресов смогут достичь конечной точки виртуальной машины.

**Запретить** - при добавлении одного или нескольких диапазонов "запретить" все другие диапазоны трафика по умолчанию разрешаются.

**Сочетание разрешения и запрета** - можно использовать сочетание правил "разрешить" и "запретить", чтобы указать вложенный разрешенный или запрещенный *диапазон IP*-адресов.

Рассмотрим два примера стандартных списков:

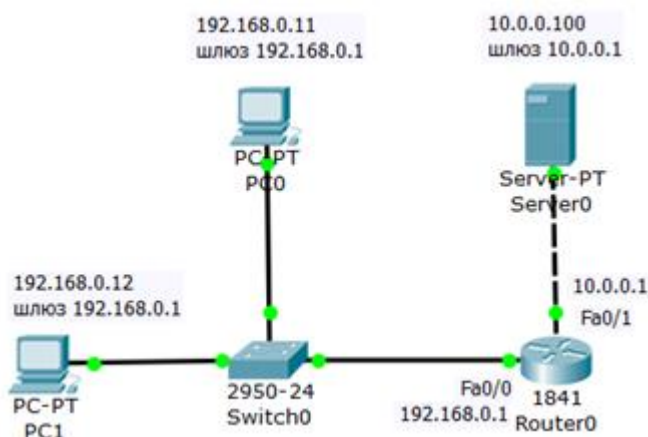
**# access-list 1 permit host 10.0.0.10** - разрешаем прохождение трафика от узла 10.0.0.10.

**# access-list 2 deny 10.0.1.0 0.0.0.255** - запрещаем прохождение пакетов из подсети 10.0.1.0/24.

### Практическая работа 7-1. Создание стандартного списка доступа

Списки доступа бывают нескольких видов: стандартные, расширенные, динамические и другие. В стандартных *ACL* есть возможность задать только *IP адрес* источника пакетов для их запретов или разрешений.

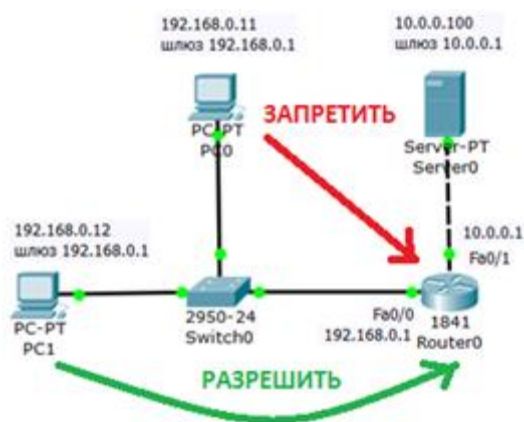
На показаны две подсети: 172.178.0.0 и 10.0.0.0.



**Рис. 7.1.** Схема сети

### Постановка задачи

Требуется разрешить доступ на сервер PC1 с адресом 172.178.0.12, а PC0 с адресом 172.178.0.11 – запретить (



**Рис. 7.2.** Постановка задачи

Соберем данную схему и настроим ее. Настройку PC0 и PC1 выполните самостоятельно.

### Настройка R0

Интерфейс 0/0 маршрутизатора 1841 настроим на адрес 172.178.0.1 и включим следующими командами:

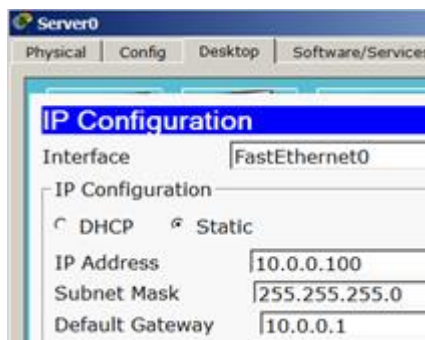
```
Router>en
Router#conf t
Router (config)#int fa0/0
Router (config-if)#ip addr 172.178.0.1 255.255.255.0
Router (config-if)#no shut
Router (config-if)#exit
```

Второй интерфейс маршрутизатора (порт 0/1) настроим на адресом 10.0.0.1 и так же включим:

```
Router (config)#int fa0/1
Router (config-if)#ip addr 10.0.0.1 255.255.255.0
Router (config-if)#no shut
```

### Настройка сервера

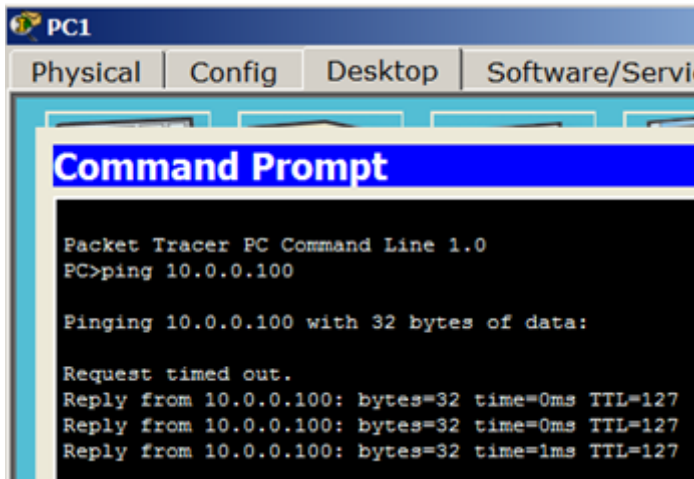
Настройки сервера приведены на .



**Рис. 7.3.** Конфигурирование S0

### Диагностика сети

Проверяем связь ПК из разных сетей ).



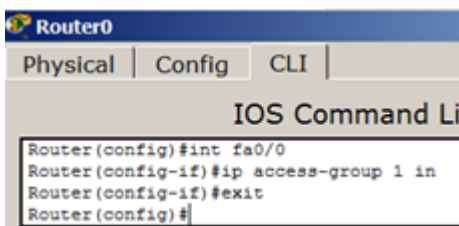
**Рис. 7.4.** ПК из разных сетей могут общаться  
**Приступаем к решению задачи**

Правило запрета и разрешения доступа будем составлять с использованием стандартных списков доступа (ACL). Пока не задан список доступа на интерфейсе всё разрешено (**permit**). Но, стоит создать список, сразу действует механизм "Всё, что не разрешено, то запрещено". Поэтому нет необходимости что-то запрещать (**deny**) – указываем что разрешено, а "остальным – запретить" подразумевается автоматически. По условиям задачи нам нужно на R0 пропустить пакеты с узла 172.168.0.12 на сервер



**Рис. 7.5.** Создаем на R0 разрешающий ACL

Применяется данное правило на интерфейс в зависимости от направления (PC1 расположен со стороны порта Fa0/0) – . Эта настройка означает, что список доступа (правило с номером 1) будет действовать на интерфейсе fa0/0 на входящем (in) от PC1 направлении.



**Рис. 7.7.** Применяем правило к порту Fa0/0  
**Примечание**

Входящий трафик (in) — этот тот, который приходит на интерфейс извне. Исходящий (out) — тот, который отправляется с интерфейса вовне. Список доступа вы можете применить либо на входящий трафик, тогда неудобные пакеты не будут даже попадать на маршрутизатор и соответственно, дальше в сеть, либо на исходящий, тогда пакеты приходят на маршрутизатор, обрабатываются им, доходят до целевого интерфейса и только на нём обрабатываются. Как правило, списки применяют на входящий трафик (in).

Проверяем связь ПК с сервером).

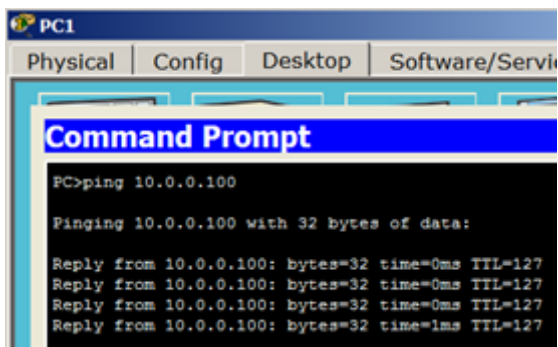


Рис. 7.7. Для PC1 сервер доступен

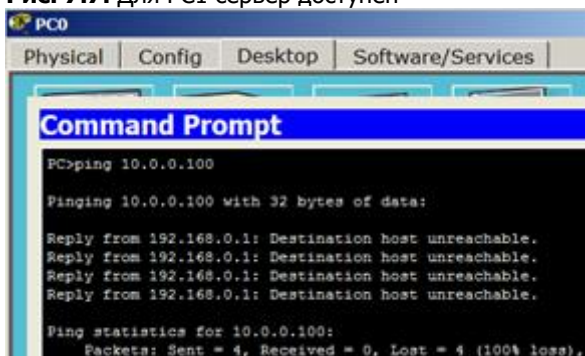


Рис. 7.8. Для PC0 сервер не доступен

Давайте посмотрим ACL

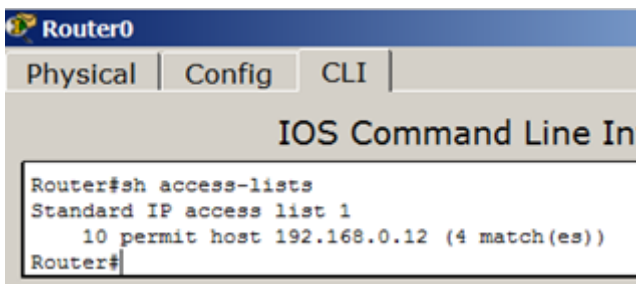


Рис. 7.7. Узел 172.178.0.12 разрешен

#### Примечание

Теперь, предположим, нужно добавить новый узел, например, PC2 с адресом 172.178.0.13 в раздел "разрешённых". Пишем команду **Router (config)#access-list 1 permit host 172.178.0.13**. Теперь адреса 172.178.0.12 и 172.178.0.13 могут общаться с сервером, в 172.178.0.11 – нет. А для отмены какого-либо правила – повторяем его с приставкой "no". Тогда это правило исключается из конфигурации. Например, если выполнить команду **Router (config-if)#no ip access-group 1 in**, то ACL будет отменен и снова все ПК могут пинговать сервер.

## Расширенные списки доступа ACL

Стандартные *права* не так гибки, как хотелось бы. В отличие от стандартных списков, расширенные списки фильтруют трафик более "тонко". При создании расширенных списков в правилах доступа можно включать фильтрацию трафика по протоколам и портам. Для указания портов в правиле доступа указываются следующие обозначения ( ):

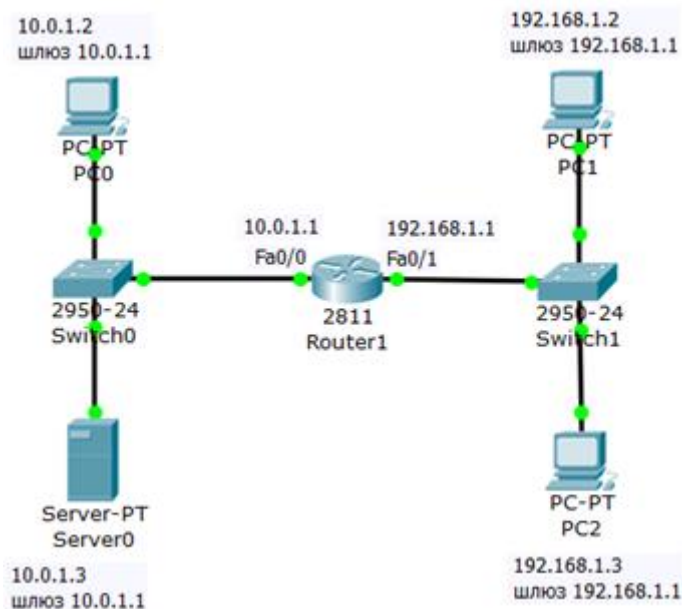
Таблица 7.1. Обозначение портов в ACL

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.

## Практическая работа 7-2-1. Расширенные списки доступа ACL

Соберите схему сети, показанную на

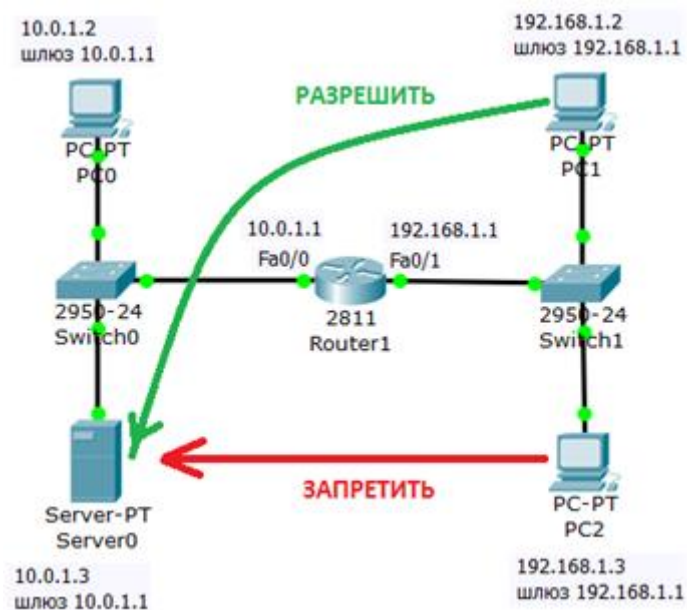


**Рис. 7.10.** Схема сети

Задача: разрешить *доступ* к *FTP* серверу 10.0.1.3 для узла 192.168.1.2 и запретить для узла 192.168.1.3.

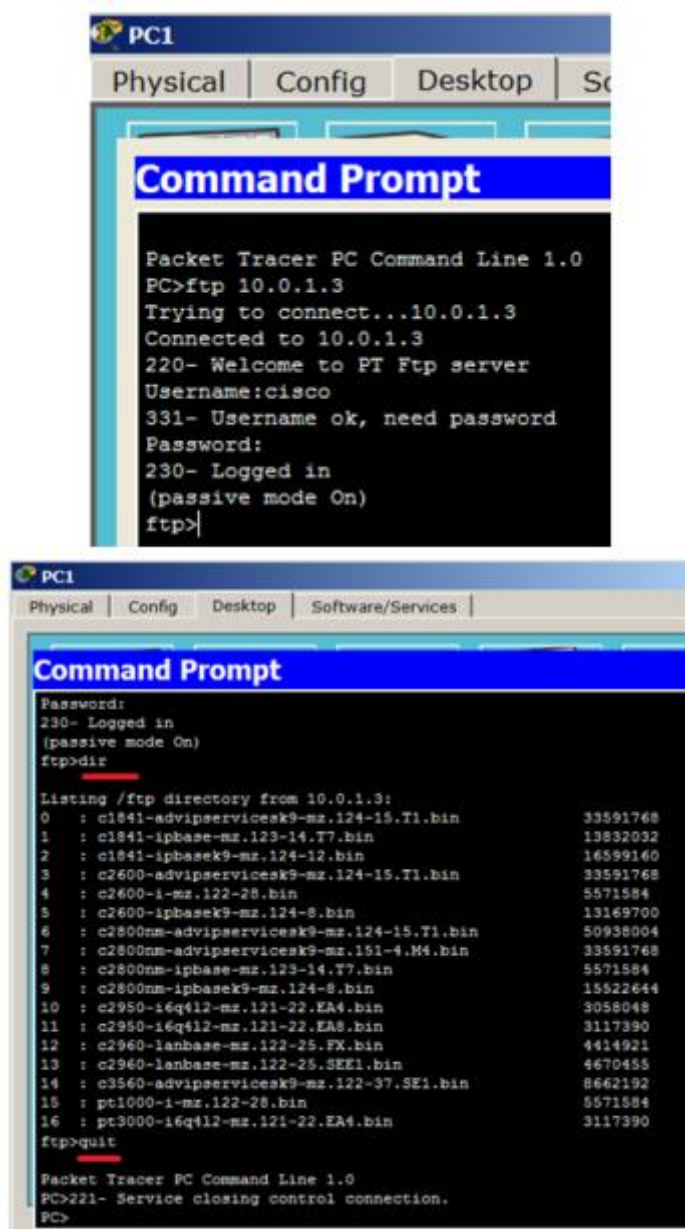
### Создаем расширенные списки доступа и запрещаем FTP трафик

Постановка задачи графически изображена на



**Рис. 7.11.** Стрелками показана цель нашей работы

Изначально на сервере 10.0.1.3 FTP сервис поднят по умолчанию со значениями имя пользователя Cisco, пароль Cisco. Убедимся, что узел S0 доступен и FTP работает, для этого заходим на PC1 и связываемся с сервером (рис. 7.12). Выполняем какие-либо команды, например, DIR – чтение директории.



**Рис. 7.12.** FTPсервер доступен

**Примечание**

При наборе пароля на экране ничего не отображается.

Теперь создадим список правил с номером 101 в котором укажем 2 разрешающих и по 2 запрещающих правила для портов сервера 21 и 20 (Эти порты служат для FTP - передачи команд и данных) –



```

Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#ip access-list extended 101
Router(config-ext-nacl)#permit tcp 192.168.1.2 0.0.0.0 10.0.1.3 0.0.0.0 eq 21
Router(config-ext-nacl)#permit tcp 192.168.1.2 0.0.0.0 10.0.1.3 0.0.0.0 eq 20
Router(config-ext-nacl)#deny tcp 192.168.1.3 0.0.0.0 10.0.1.3 0.0.0.0 eq 21
Router(config-ext-nacl)#deny tcp 192.168.1.3 0.0.0.0 10.0.1.3 0.0.0.0 eq 20
Router(config-ext-nacl)#deny tcp 192.168.1.3 0.0.0.0 10.0.1.3 0.0.0.0 eq 20
Router(config-ext-nacl)#

```

**Рис. 7.13.** Составляем расширенные списки доступа  
**Совет**

Набирайте команды аккуратно и внимательно: даже один лишний пробел может привести к ошибке при выполнении команды.

А теперь применяем наш список с номером 101 на вход (in) Fa0/1 потому, что трафик входит на этот порт роутера со стороны сети 172.178.1.0

```

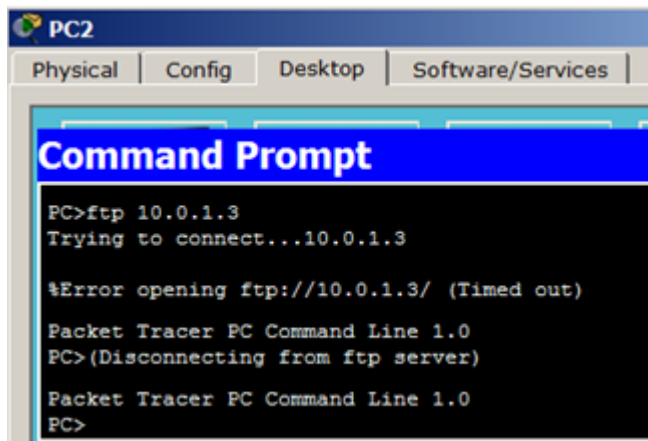
Router(config-ext-nacl)#int fa0/1
Router(config-if)#ip access-group 101 in
Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#wr mem
Building configuration...
[OK]
Router#

```

**Рис. 7.14.** Применяем правило с номером 101 к порту 0/1 роутера

Проверяем связь сервера с PC2



**Рис. 7.15.** Для PC2 FTP сервер не доступен

Проверяем связь сервера с PC1

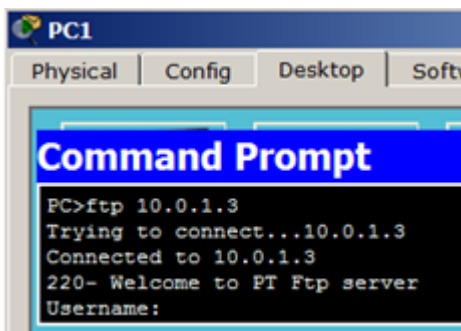


Рис. 7.17. Для PC1 FTP сервер доступен

## Настройка статического NAT

*NAT (Network Address Translation) — трансляция сетевых адресов, технология, которая позволяет преобразовывать (изменять) IP адреса и порты в сетевых пакетах. NAT используется чаще всего для осуществления доступа устройств из локальной сети предприятия в Интернет, либо наоборот для доступа из Интернет на какой-либо ресурс внутри сети. Локальная сеть предприятия строится на частных IP адресах:*

- 10.0.0.0 — 10.255.255.255 (10.0.0.0/255.0.0.0 (/8))
- 172.17.0.0 — 172.31.255.255 (172.17.0.0/255.240.0.0 (/12))
- 172.178.0.0 — 172.178.255.255 (172.178.0.0/255.255.0.0 (/17))

Эти адреса не маршрутизируются в Интернете, и провайдеры должны отбрасывать пакеты с такими IP адресами отправителей или получателей. Для преобразования частных адресов в Глобальные (маршрутизируемые в Интернете) применяют *NAT*.

### Новый термин

**NAT** — технология трансляции сетевых адресов, т.е. подмены адресов (или портов) в заголовке IP-пакета. Другими словами, пакет, проходя через маршрутизатор, может поменять свой адрес источника и/или назначения. Подобный механизм служит для обеспечения доступа из LAN, где используются частные IP-адреса, в Internet, где используются глобальные IP-адреса.

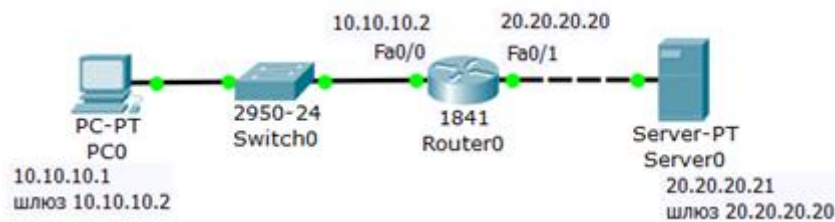
Существует три вида трансляции *Static NAT*, *Dynamic NAT*, *Overloading (PAT)*.

- **Static NAT (статический NAT)** осуществляет преобразование IP адреса один к одному, то есть сопоставляется один адрес из внутренней сети с одним адресом из внешней сети. Иными словами, при прохождении через маршрутизатор, адрес(а) меняются на строго заданный адрес, один-к-одному (Например, 10.1.1.5 всегда заменяется на 11.1.1.5 и обратно). Запись о такой трансляции хранится неограниченно долго, пока есть соответствующая строчка в конфигурации роутера.
- **Dynamic NAT (динамический NAT)** производит преобразование внутреннего адреса/ов в один из группы внешних адресов. То есть, перед использованием динамической трансляции, нужно задать nat-пул внешних адресов. В этом случае при прохождении через маршрутизатор, новый адрес выбирается динамически из некоторого диапазона адресов, называемого пулом (pool). Запись о трансляции хранится некоторое время, чтобы ответные пакеты могли быть доставлены адресату. Если в течение некоторого времени трафик по этой трансляции отсутствует, трансляция удаляется и адрес возвращается в пул. Если требуется создать трансляцию, а свободных адресов в пуле нет, то пакет отбрасывается. Иными словами, хорошо бы, чтобы число внутренних адресов было ненамного больше числа адресов в пуле, иначе высока вероятность проблем с выходом в WAN.
- **Overloading (или PAT)** позволяет преобразовывать несколько внутренних адресов в один внешний. Для осуществления такой трансляции используются порты, поэтому такой NAT называют PAT (Port Address Translation). С помощью PAT можно преобразовывать внутренние адреса во внешний адрес, заданный через пул или через адрес на внешнем интерфейсе.

## Практическая работа 7-3-1. Статическая трансляция адресов NAT



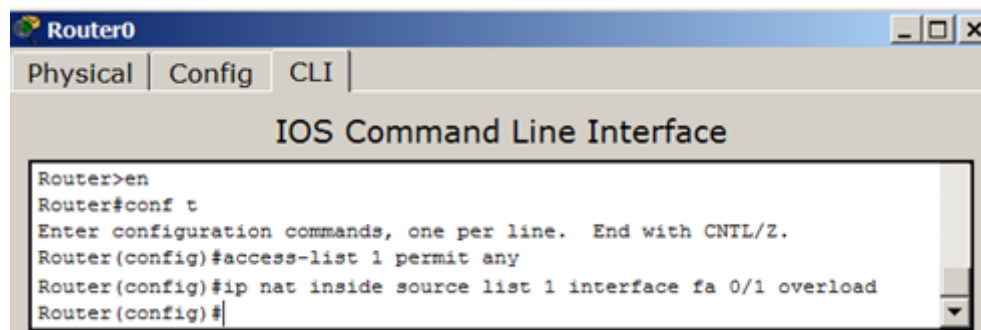
На имеется внешний *адрес* 20.20.20.20 (внешний *интерфейс* fa0/1) и внутренняя *сеть* 10.10.10.0 (внутренний *интерфейс* fa0/0). Нужно настроить *NAT*. Предполагается, что адреса уже прописаны, и *сеть* поднята (рабочая).



**Рис. 7.17.** Схема сети

## На R0 добавляем access-list, разрешаем всё (any)

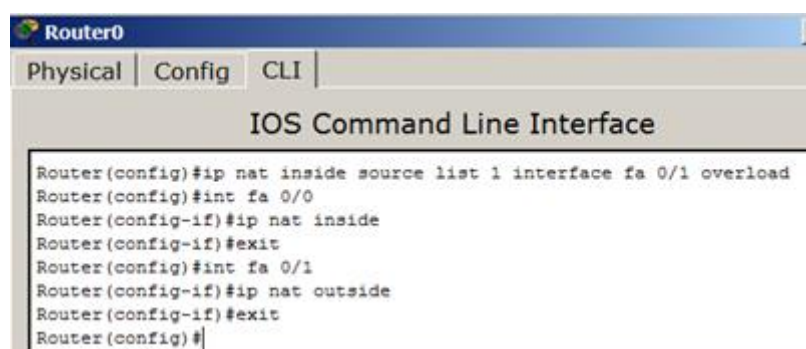
Разрешаем весь трафик, то есть, любой IP адрес



**Рис. 7.18.** Составляем лист допуска

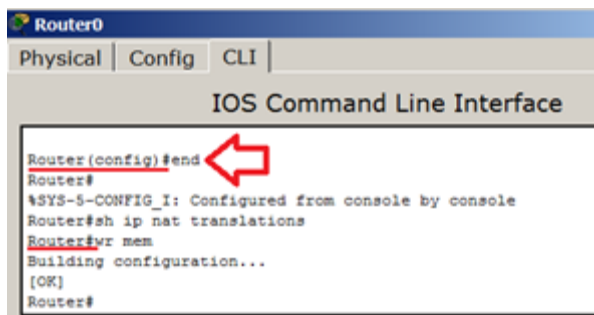
## Создаём правило трансляции

Теперь настроим трансляцию на интерфейсах (на внутреннем inside, на внешнем – outside), то есть, для R0 указываем внутренний и внешний порты



**Рис. 7.17.** Для R0 назначаем внутренний и внешний порты

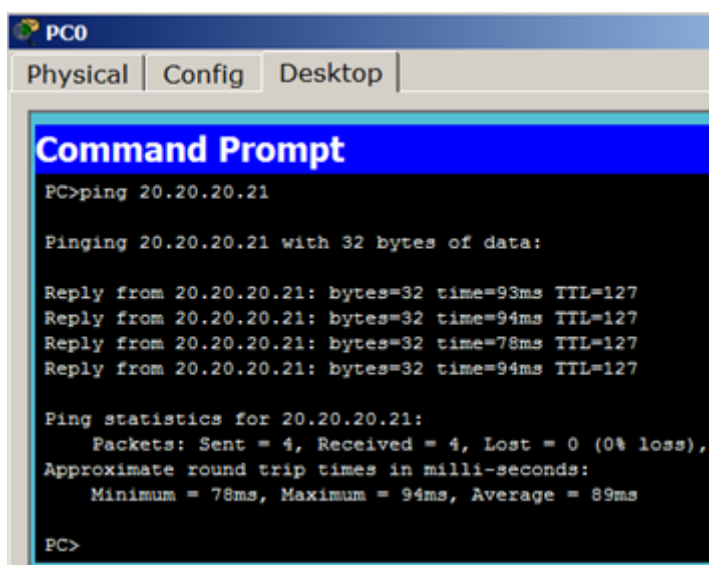
Выходим из режима глобального конфигурирования и записываем настройки роутера в микросхему памяти



**Рис. 7.20.** Сохраняем настройки в ОЗУ

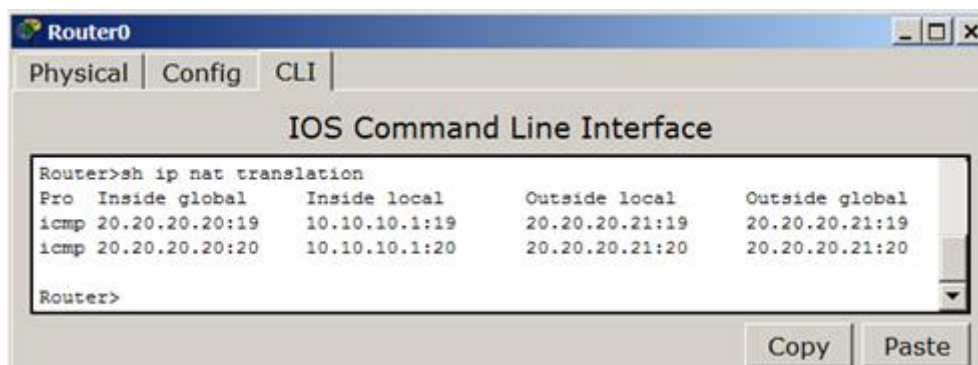
## Проверяем работу сети (просмотр состояния таблицы NAT)

С PC0 пингуем провайдера и убеждаемся, что PC1 и сервер могут общаться



**Рис. 7.21.** Из внутренней сети пингуем внешнюю сеть

Для просмотра состояния таблицы NAT, одновременно с пингом используйте команду **Router#sh ip nat translations** (я запустил пинг с машины 10.10.10.1, т.е., с PC1 на адрес 20.20.20.21, т.е., на S0) – .



**Рис. 7.22.** Вовремя пинга просматриваем состояние таблицы NAT

Убеждаемся в успешной маршрутизации в режиме симуляции

							Event List	Simulation		
Fire	Last Status	Source	Destination	Type	Color	Time (sec)	Periodic	Num	Edit	Delete
	Successful	PC0	Server0	ICMP		0.000	N	0	(edit)	(delete)

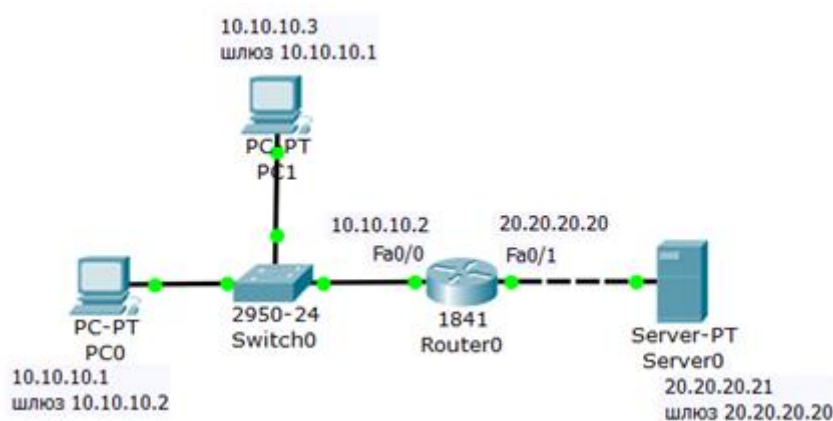
  

NAT Table for Router0						
Protocol	Inside Global	Inside Local	Outside Local	Outside Global		
icmp	20.20.20.20:10	10.10.10.1:10	20.20.20.21:10	20.20.20.21:10		

**Рис. 7.23.** Связь PC0 и S0 работает

### Задание 7.3

Если в схему добавить PC1 то будет ли работать статический NAT между ним и S0?

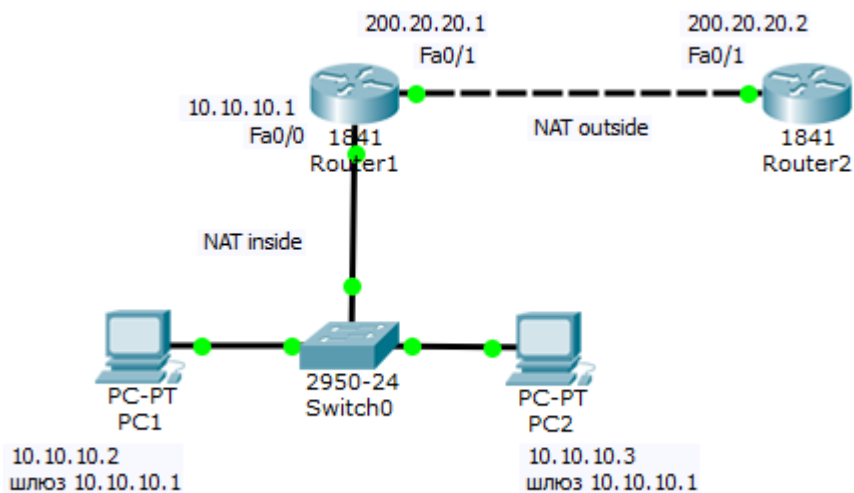


**Рис. 7.24.** Задание для самостоятельной работы

## Практическая работа 7-3-2. Настройка статического NAT

Статический NAT - сопоставляет один NAT inside (внутренний=частный локальный *ip-адрес*) с одним NAT outside (глобальным=публичным внешним *ip-адресом*) – [рис. 7.25](#).

Здесь ISP (*Internet Service Provider*) - поставщик *Интернет-услуг* (*Интернет-провайдер*).



**Рис. 7.25.** Схема сети

## Алгоритм настройки R1

Ниже приведена последовательность команд конфигурирования маршрутизатора R1 по шагам.

### Шаг 1. Настройка дефолта на R1

```
R1(config)# ip route 0.0.0.0 0.0.0.0 200.20.20.2
```

### Шаг 2. Настройка внутреннего интерфейса в отношении NAT

```
R1(config)# interface fastethernet 0/0  
R1(config-if)# ip nat inside
```

### Шаг 3. Настройка внешнего интерфейса в отношении NAT

```
R1(config)# interface fastethernet 0/1  
R1(config-if)# ip nat outside
```

### Шаг 4. Настройка сопоставления ip-адресов.

```
R1(config)# ip nat inside source static 10.10.10.2 200.10.21.5
```

В результате этой команды ip-адресу 200.10.21.5 всегда будет соответствовать внутренний ip-адрес 10.10.10.2, т.е. если мы будем обращаться к адресу 200.10.21.5 то отвечать будет PC1.

Полный листинг команд приведен на.



**Рис. 7.27.** Полный листинг команд по настройке R1

## Команды для проверки работы NAT

Проверим связь PC1 и R2

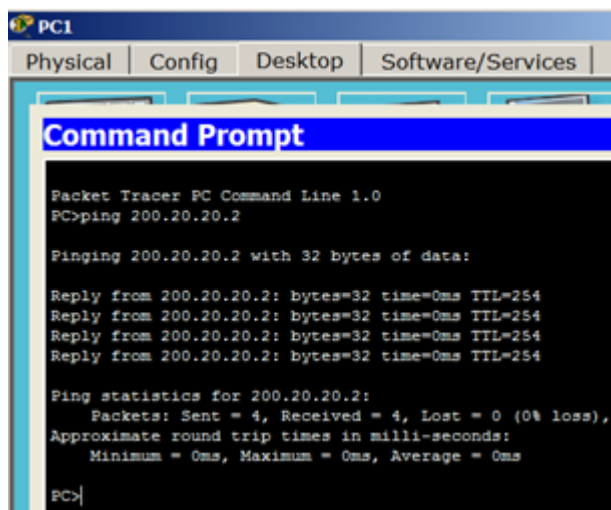


Рис. 7.27. PC1 видит R2

Проверим, что R1 видит соседние сети

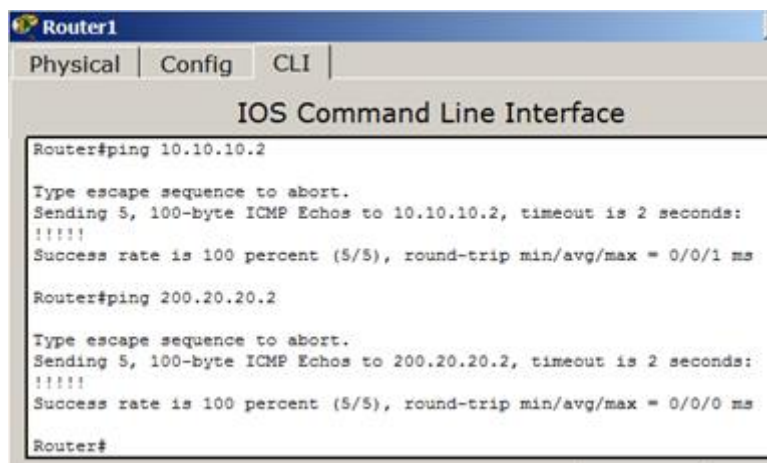
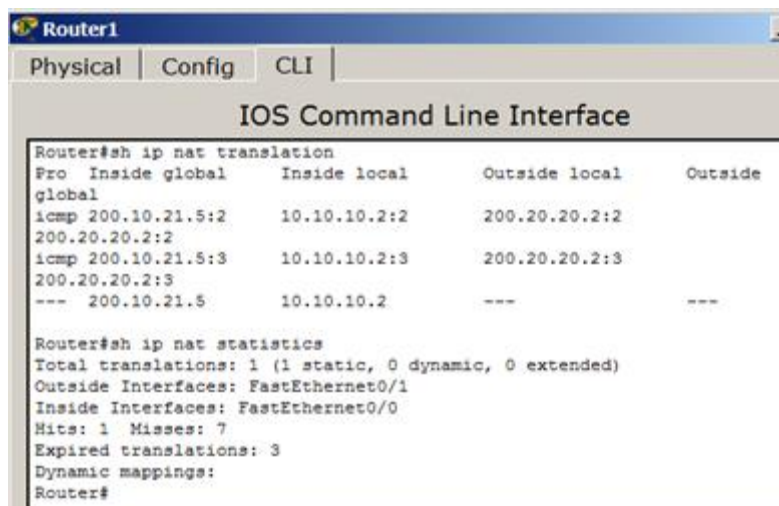


Рис. 7.28. R1 видит PC1 и R2

Проверим механизм работы статического NAT: команда **show ip nat translations** выводит активные преобразования, а команда **show ip nat statistics** выводит статистику по NAT преобразованиям

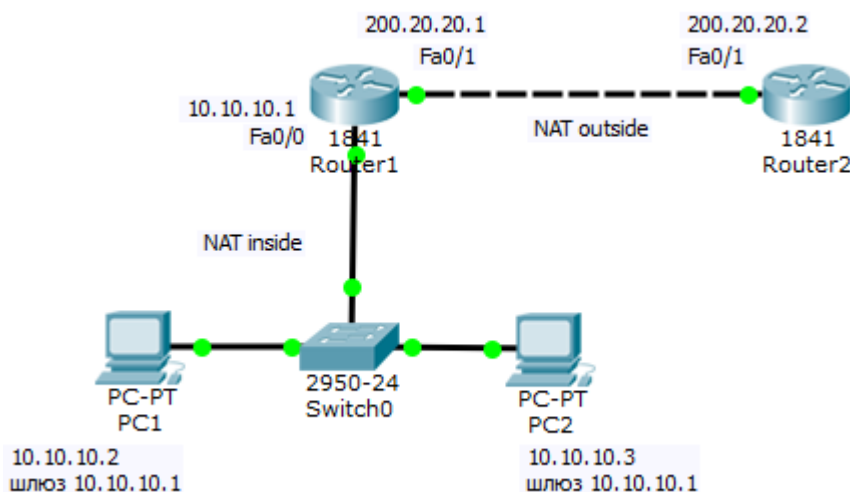


**Рис. 7.27.** Проверка механизма работы статического NAT

Из иллюстрации видим, что глобальному ip-адресу 200.10.21.5 соответствует локальный ip-адрес 10.10.10.2, а также, какой интерфейс является внешним, а какой - внутренним.

## Динамическая трансляция адресов. Настройка динамического NAT

Динамический NAT - использует пул доступных глобальных (публичных) ip-адресов и назначает их внутренним локальным (частным) адресам. Схема для нашей работы приведена на [рис. 7.30](#).



**Рис. 7.30.** Схема сети

## Практическая работа 7-4-1. Настройка динамического NAT на маршрутизаторе R1 по шагам.

### Шаг 1. Настройка на R1 списка доступа, соответствующего адресам LAN

```
R1 (config)# access-list 1 permit 10.10.10.0 0.0.0.255
```

Здесь 0.0.0.225 – обратная (инверсная) маска для адреса 10.10.10.0.

### Шаг 2. Настройка пула адресов

```
R1 (config)# ip nat pool white-address 200.20.20.1 200.20.20.30 netmask 255.255.255.0
```

### Шаг 3. Настройка трансляции

```
R1 (config)# ip nat inside source list 1 pool white-address
```

### Шаг 4. Настройка внутреннего интерфейса в отношении NAT

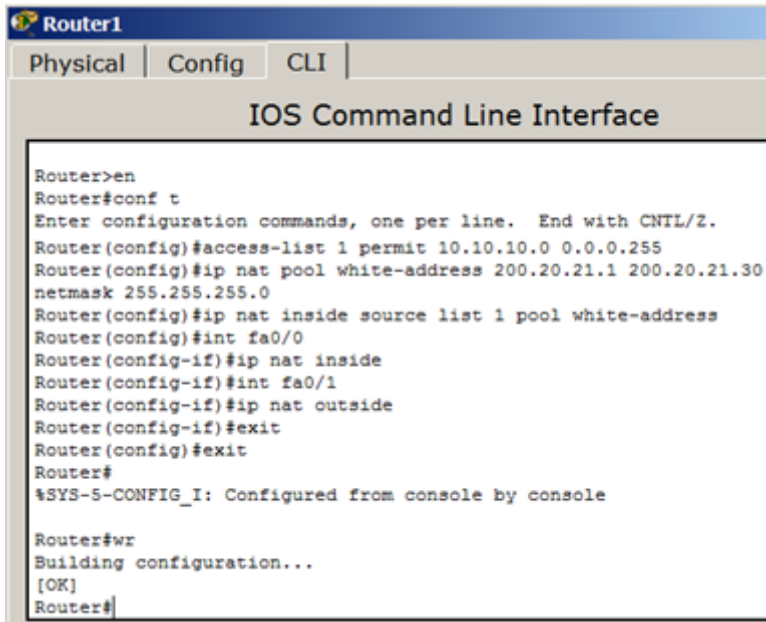
```
R1 (config)# interface fastethernet 0/0  
R1 (config-if)# ip nat inside
```

### Шаг 5. Настройка внешнего интерфейса в отношении NAT



```
R1 (config)# interface fastethernet 0/1
R1 (config-if)# ip nat outside
```

Ниже дан полный листинг команд по настройке R1.

The screenshot shows the 'Router1' window with tabs for 'Physical', 'Config', and 'CLI'. The 'CLI' tab is active, displaying the 'IOS Command Line Interface'. The command history shows the following sequence: 'Router>en', 'Router#conf t', 'Router(config)#access-list 1 permit 10.10.10.0 0.0.0.255', 'Router(config)#ip nat pool white-address 200.20.21.1 200.20.21.30 netmask 255.255.255.0', 'Router(config)#ip nat inside source list 1 pool white-address', 'Router(config)#int fa0/0', 'Router(config-if)#ip nat inside', 'Router(config-if)#int fa0/1', 'Router(config-if)#ip nat outside', 'Router(config-if)#exit', 'Router(config)#exit', 'Router#'. A system message '%SYS-S-CONFIG\_I: Configured from console by console' is displayed. The user then enters 'Router#wr', followed by 'Building configuration...' and '[OK]', and finally 'Router#'.

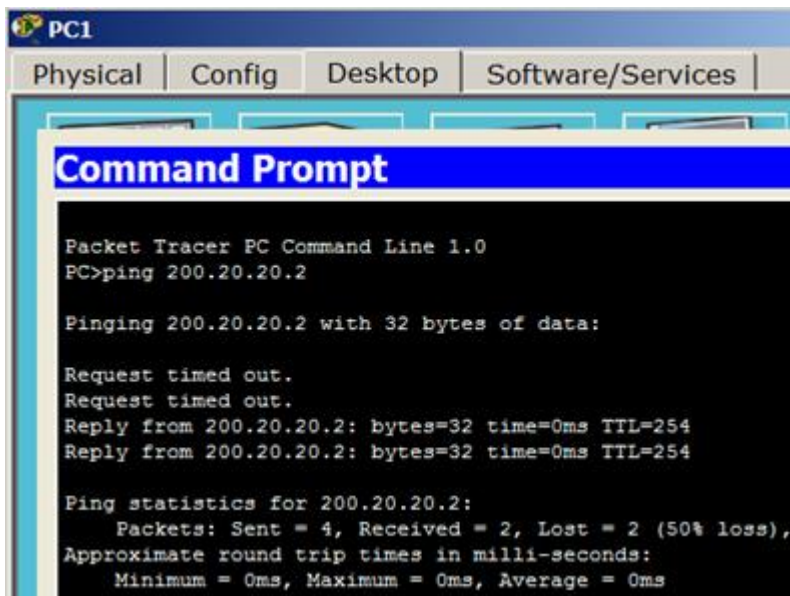
```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#access-list 1 permit 10.10.10.0 0.0.0.255
Router(config)#ip nat pool white-address 200.20.21.1 200.20.21.30
netmask 255.255.255.0
Router(config)#ip nat inside source list 1 pool white-address
Router(config)#int fa0/0
Router(config-if)#ip nat inside
Router(config-if)#int fa0/1
Router(config-if)#ip nat outside
Router(config-if)#exit
Router(config)#exit
Router#
%SYS-S-CONFIG_I: Configured from console by console

Router#wr
Building configuration...
[OK]
Router#
```

Рис. 7.31. Полный листинг команд по конфигурированию R1

## Команды для проверки работы динамического NAT

Проверим *связь* PC1 и R2

The screenshot shows the 'PC1' window with tabs for 'Physical', 'Config', 'Desktop', and 'Software/Services'. The 'Command Prompt' window is open, displaying the output of a ping command. The text shows 'Packet Tracer PC Command Line 1.0', 'PC>ping 200.20.20.2', 'Pinging 200.20.20.2 with 32 bytes of data:', 'Request timed out.', 'Request timed out.', 'Reply from 200.20.20.2: bytes=32 time=0ms TTL=254', 'Reply from 200.20.20.2: bytes=32 time=0ms TTL=254', and 'Ping statistics for 200.20.20.2: Packets: Sent = 4, Received = 2, Lost = 2 (50% loss), Approximate round trip times in milli-seconds: Minimum = 0ms, Maximum = 0ms, Average = 0ms'.

```
Packet Tracer PC Command Line 1.0
PC>ping 200.20.20.2

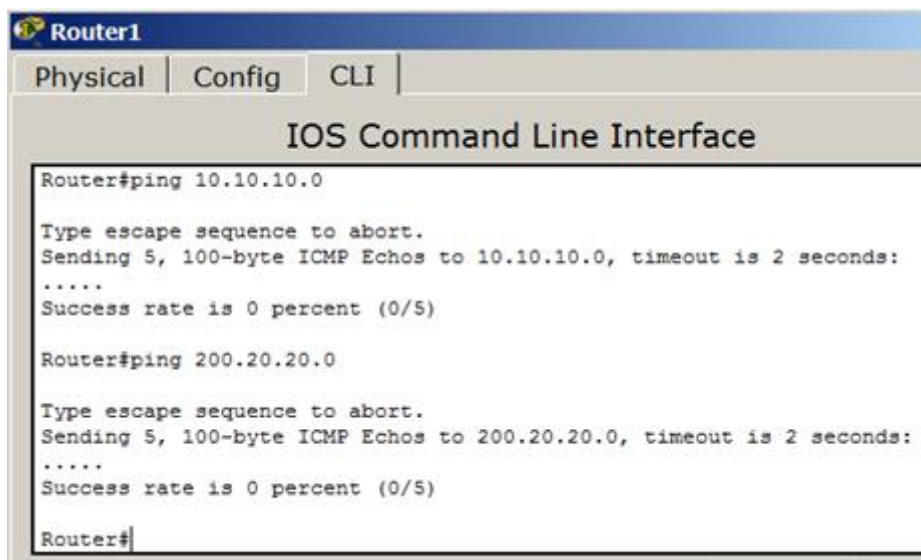
Pinging 200.20.20.2 with 32 bytes of data:

Request timed out.
Request timed out.
Reply from 200.20.20.2: bytes=32 time=0ms TTL=254
Reply from 200.20.20.2: bytes=32 time=0ms TTL=254

Ping statistics for 200.20.20.2:
    Packets: Sent = 4, Received = 2, Lost = 2 (50% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Рис. 7.32. PC1 видит R2

Проверим, что R1 видит соседние сети



```
Router1
Physical Config CLI
IOS Command Line Interface

Router#ping 10.10.10.0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.0, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

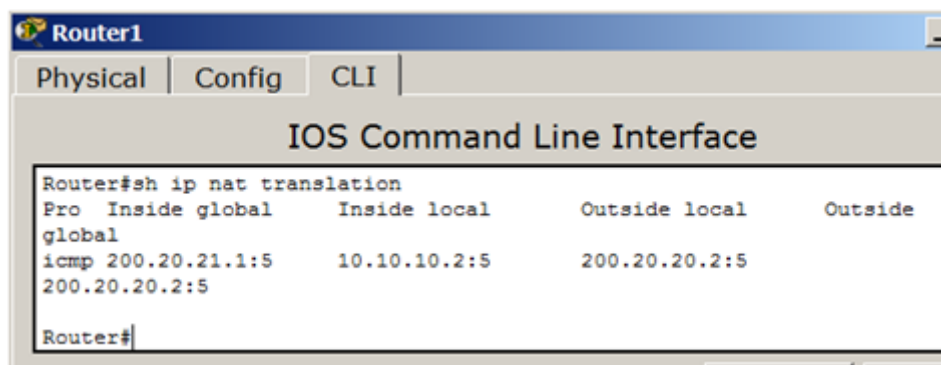
Router#ping 200.20.20.0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.20.20.0, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Router#
```

**Рис. 7.33.** R1 видит подсети 10.10.10.0 и 200.20.20.0

Проверим механизм работы динамического NAT: для этого выполним одновременно (параллельно) команды **ping** и **show ip nat translations**

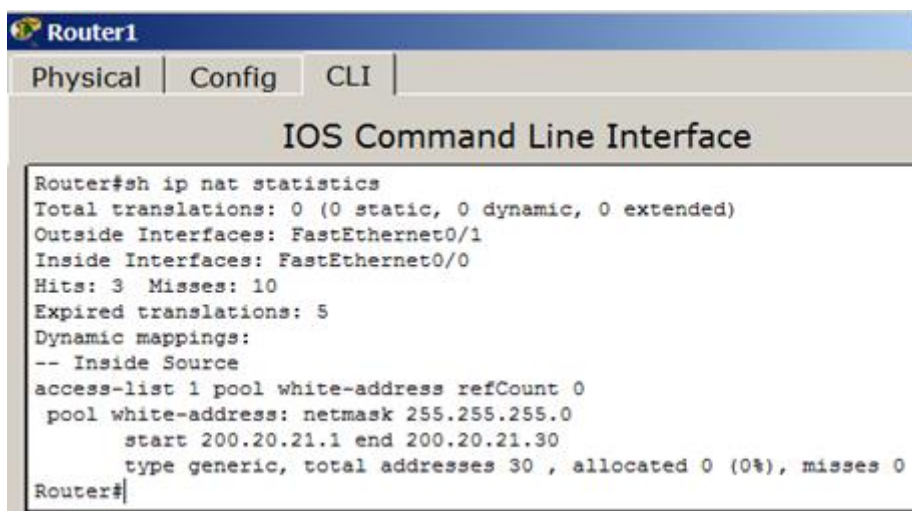


```
Router1
Physical Config CLI
IOS Command Line Interface

Router#sh ip nat translation
Pro Inside global      Inside local      Outside local      Outside
global
icmp 200.20.21.1:5      10.10.10.2:5      200.20.20.2:5
200.20.20.2:5
Router#
```

**Рис. 7.34.** Адреса: глобальный, внутренний, внешний

Командой **show ip nat statistics** выведем статистику по NAT преобразованиям



```
Router1
Physical Config CLI
IOS Command Line Interface

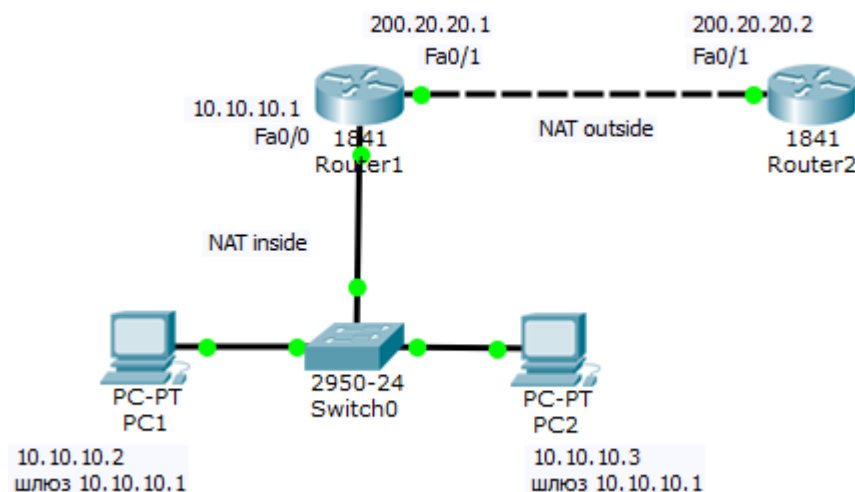
Router#sh ip nat statistics
Total translations: 0 (0 static, 0 dynamic, 0 extended)
Outside Interfaces: FastEthernet0/1
Inside Interfaces: FastEthernet0/0
Hits: 3 Misses: 10
Expired translations: 5
Dynamic mappings:
-- Inside Source
access-list 1 pool white-address refCount 0
pool white-address: netmask 255.255.255.0
start 200.20.21.1 end 200.20.21.30
type generic, total addresses 30 , allocated 0 (0%), misses 0
Router#
```

**Рис. 7.35.** Статистика работы динамического NAT

Из иллюстрации видим, что локальным адресам соответствует пул внешних адресов от 200.20.20.1 до 200.20.20.30.

## Практическая работа 7-4-2. Динамический NAT Overload: настройка PAT (маскарадинг)

PAT (*Port Address Translation*) - отображает несколько локальных (частных) *ip*-адресов в глобальный *ip*-адрес, используя различные порты (рис. 7.37).



**Рис. 7.37.** Схема сети на настройки трансляции адресов PAT

Рассмотрим *алгоритм* нашей работы по шагам.

### Шаг 1. Настройка списка доступа, соответствующего внутренним частным адресам

```
R1(config)# access-list 1 permit 10.10.10.0 0.0.0.255
```

### Шаг 2. Настройка трансляции

```
R1(config)# ip nat inside source list 1 interface fastethernet 0/1 overload
```

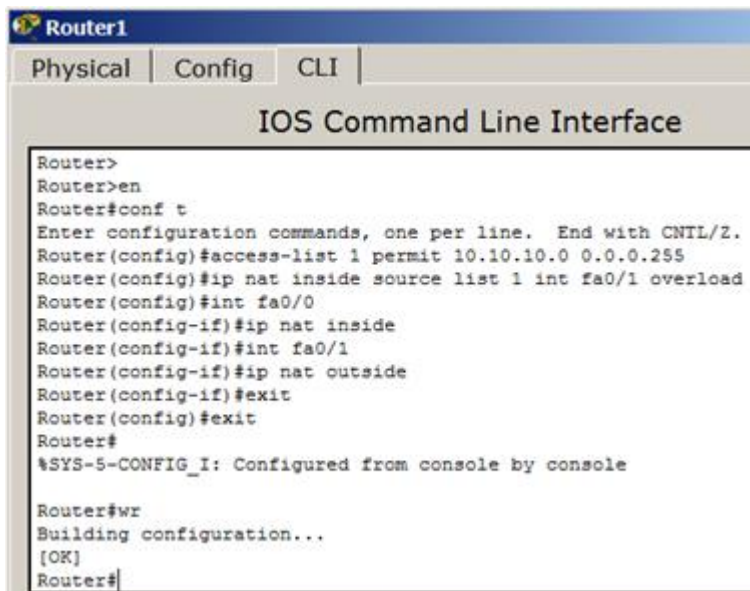
### Шаг 3. Настройка внутреннего интерфейса в отношении NAT

```
R1(config)# interface fastethernet 0/0  
R1(config-if)# ip nat inside
```

### Шаг 4. Настройка NAT на интерфейсе

```
R1(config)# interface fastethernet 0/1  
R1(config-if)# ip nat outside
```

Ниже дан полный листинг команд по конфигурированию R1 (рис. 7.37).



```
Router1
Physical Config CLI
IOS Command Line Interface

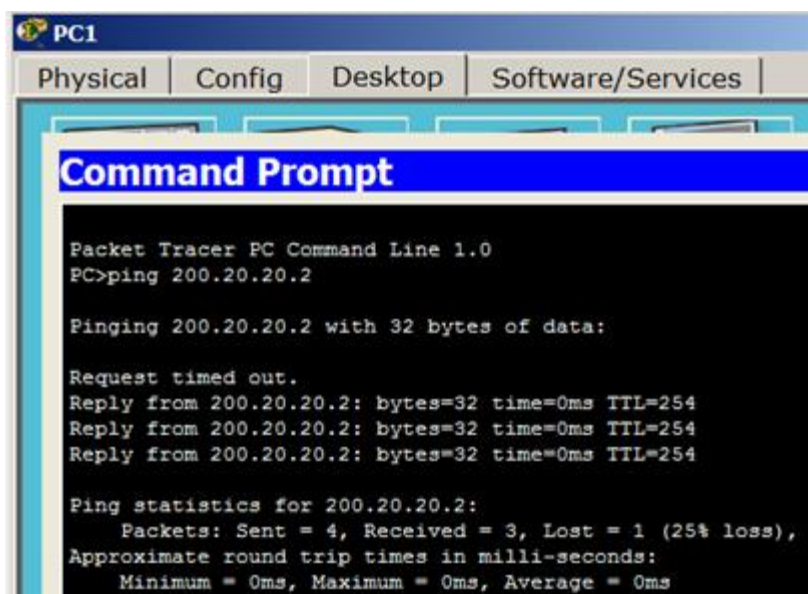
Router>
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#access-list 1 permit 10.10.10.0 0.0.0.255
Router(config)#ip nat inside source list 1 int fa0/1 overload
Router(config)#int fa0/0
Router(config-if)#ip nat inside
Router(config-if)#int fa0/1
Router(config-if)#ip nat outside
Router(config-if)#exit
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console

Router#wr
Building configuration...
[OK]
Router#
```

**Рис. 7.37.** Листинг команд по конфигурированию R1

## Команды для проверки работы маскардинга (PAT)

Проверим *связь* PC1 и R2



```
PC1
Physical Config Desktop Software/Services
Command Prompt

Packet Tracer PC Command Line 1.0
PC>ping 200.20.20.2

Pinging 200.20.20.2 with 32 bytes of data:

Request timed out.
Reply from 200.20.20.2: bytes=32 time=0ms TTL=254
Reply from 200.20.20.2: bytes=32 time=0ms TTL=254
Reply from 200.20.20.2: bytes=32 time=0ms TTL=254

Ping statistics for 200.20.20.2:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

**Рис. 7.38.** PC1 видит R2

Проверим, что R1 видит соседние сети

```

Router1
Physical Config CLI
IOS Command Line Interface

Router#ping 10.10.10.0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.10.10.0, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Router#ping 200.20.20.0

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 200.20.20.0, timeout is 2 seconds:
.....
Success rate is 0 percent (0/5)

Router#

```

**Рис. 7.37.** R1 видит подсети 10.10.10.0 и 200.20.20.0

Проверим механизм работы динамического *NAT*: для этого выполним одновременно (параллельно) команды **ping** и **show ip nat translations**

```

Router1
Physical Config CLI
IOS Command Line Interface

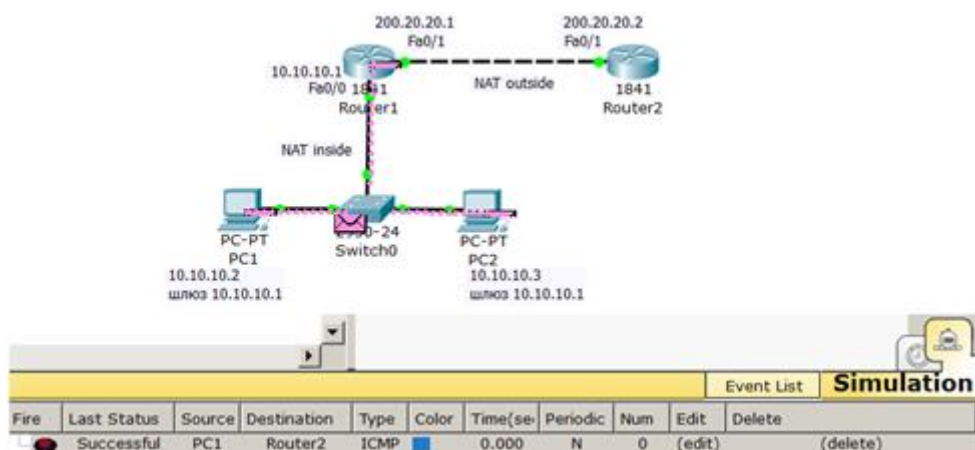
Router#sh ip nat translation
Router#sh ip nat translation
Pro Inside global      Inside local      Outside local      Outside
global
icmp 200.20.20.1:5     10.10.10.2:5     200.20.20.2:5
200.20.20.2:5

Router#

```

**Рис. 7.40.** Адреса: глобальный, внутренний, внешний

Проверим работу сети в режиме симуляции ( [рис. 7.41](#)).



**Рис. 7.41.** PAT работает, PC1 и R2 отправляют и получают пакеты Successful