Я **exit** и **no shutdown** не ставлю, но за этим нужно тщательно следить

Show vlan – покажет характеристики на switch

Всё нужно делать в режиме enable (а может и выше, т.е. если ты в роутере, то нужно делать ещё config term)

Show running-config – в router выдаёт много информации, но я в ней быстро нахожу то, что нужно

**Назначение ip-адреса:**

interface Gig0/0

ip addr 10.0.1.1 255.255.255.0

no shutdown

**если делаешь с виртуальными интерфейсами, то нужно сделать для физического (просто не забыть):**

interface gig0/0

no ip address

и можно мучать **виртуальные**, с добавлением (“.<цифра>” ) )

**Подключение vlan:**

vlan 3 – создали

**Ставим дырку для vlan**

interface Fa0/2

switchport access vlan 3

**Ставим trunk на некоторый интерфейс, по которому будут летать пакеты из vlan (т.е. пакеты с метками)**

interface Fa0/3

switchport mode trunk

**Принятие пакетов с метками от vlan на router**

interface Gig0/0.1

encapsulation dot1Q 2 – 2 – номер vlan

Этого достаточно для того, чтобы реализовать задание под номером 2, где нужно лишь расставить ip-адреса и расставить vlan (между которыми можно будет общаться, потому что их соединяет router, ибо он каждый пакет выймет из vlan, отмаршрутизирует, а потом вкинет в нужный vlan обратно)

**Суть OSPF:**

Тут везде будет указана одна и та же эрия с номером 1 и в пределах этой эрии построится единая таблица маршрутизации

Точнее мы указываем на какой ножке висит какая эрия, а дальше маршрутизатор заботится о том, чтобы если у него 2 ножки с одним и тем же номером эрии, то он их «синхронизирует» и обменивает таблицы маршрутизации

**Как назначить ospf**

router ospf 100 – 100 – это просто номер ospf на маршрутизаторе, не имеет никакого значения.

Несколько штук:

network 172.16.1.1 0.0.0.0 area 1

1 – это номер эрии, должен совпадать при конфигурировании нескольких разных маршрутизаторов

Маска! – она нулями указывает значащие биты (причём может быть например 0.225.0.255 – т.е. не обязательно сначала одни единицы, а потом одни нули)

фактически несколько таких правил соединяет лапки маршрутизатора в одну эрию.

Такое нужно провернуть на маршрутизаторах BorderGW и MainRouter

Причем, если я правильно понимаю задание (точнее его задумку, то интерфейс, на котором висит ISP не нужно включать в эрию, туда будет проложен маршрут дополнительно)

**Статический маршрут**

Это маршрут, который задаётся лапками на маршрутизаторе

ip route 192.168.1.0 255.255.255.0 10.2.2.3

все пакеты, которые направленны в подсеть 192.168.1.0 будут отправлены в gateway по адресу 10.2.2.3 (это адрес не выхода твоего маршрутизатора – а адрес на интерфейсе соседнего маршрутизатора, на который ссылаются)

Внимание – нужно понимать – если сделаешь такую маршрутизацию (кстати заметим – что нужно в обе стороны и то, что летит в подсеть DMZ и в обратную сторону, т.е. правила – 2 разных)

(кажется такие: ip route 158.250.17.0 255.255.255.0 158.250.17.254

Ip route 192.168.1.50 255.255.255.0 <а тут я не уверен, что должно стоять, но т.к. у нас там наведены порядки с 1-й эрией, то наверно можно указать либо ip-адрес маршрутизатора MainRouter (думаю, что это вероятнее сработает), либо на ip-адрес Fa0/1 BourderGW>)

Если так всё и оставить, то работать конечно будет, но нужно понимать, что это на самом деле везение, потому что если бы подсети в DMZ и локальной сети совпадали, то трафик из ISP, который шёл бы в локальную сеть, нашей статической маршрутизацией заворачивался бы в DMZ, что всё сломало бы, но опять же – это лишь, если бы были одинаковые подсети.

**Внедрённый статический маршрут:**

Всё что нужно это выполнить команду

redistribute static subnets

внутри настроек соответствующего ospf

это внедрит ВСЕ статические маршруты в ospf, если хотелось внедрить лишь некоторые, то это сделать лишь через route-map, но это думаю уже точно выходит за рамки необходимых знаний (но если спросит. Глянь - <http://subnets.ru/blog/?p=595> – тут вроде просто и с примерами (суть в том, что в route-map ляжет всё, что попадёт под acl, который тоже нужно создать, вот и всё, а в ospf можно указать нужный route-map))

**Махинации с доступом в бухгалтерию:**

Acl, которая пропускает всё, кроме бухгалтерии (из-за наличия первых 2-х «симметричных правил» с точки зрения получателя и отправителя – эту acl можно вешать как на вход, так и на выход (in и out)):

access-list 101 deny ip 192.168.1.2 0.0.0.7 any

access-list 101 deny ip any 192.168.1.2 0.0.0.7

access-list 101 permit ip any any – писать это – хороший тон

либо (это именованный, а выше был нумерованный)

(заметим инвертированные маски)

ip access-list extended trouble\_acl

deny ip 192.168.1.2 0.0.0.7 any

deny ip any 192.168.1.2 0.0.0.7

permit ip any any

как повесить acl на интерфейс (помним, что важно направление in или out)

interface Gig0/0

ip access-group 101 in

Предлагается создать соответствующие acl на каждом из 2-х маршрутизаторов

А теперь я предлагаю развесить следующим образом –

Запрещающую acl на оставшиеся 2 виртуальных интерфейса в MainRouter и на выход ISP у BorderGW

**Сделаем PAT:**

Создадим нужный acl, который просто перечислит и пропустит все подсети из внутренней области (тут хватит и простого acl, а не расширенного)

(сделаем именованным)

ip access-list standard nat\_acl

permit 192.168.1.50 0.0.0.7 – тут указывает адрес отправителя

…

deny any any

теперь на каждом интерфейсе нужно указать, он для ната выход или вход

interface Gig0/1

ip nat outside

interface Gig0/2

ip nat inside

ip nat inside source list nat\_acl interface Gig0/1 overload

тут сказали, что мол надо бы nat включить на таком то интерфейсе и с таким то acl (nj/ что не попадёт под acl – не будет обрабатываться nat-ом)

**NAT**

Думаю, что здесь подстава в следующем:

www – имеется ввиду tcp

DNS – имеется ввиду udp

Поэтому нужны следующие 2 правила:

ip nat inside source static tcp 192.168.1.50 80 158.250.17.103 80

ip nat inside source static udp 192.168.1.51 53 158.250.17.103 53

а в остальном этот пункт очень прост

не знаю только, совмещается ли этот nat с тем, который был PAT выше

но почти полностью уверен, что нет

и ещё тут наверно тоже нужно указать точки inside и outside (только не путайте, ведь трафик теперь идёт из ISP, как источника)

как я откопал эти правила ? – да просто использовал после каждого слова «?» и смотрел, их help – очень удобно