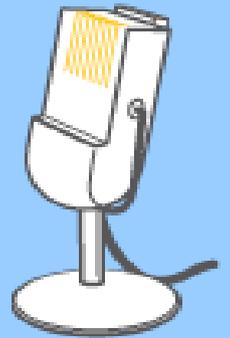


# Сети для самых маленьких

## Часть восьмая. BGP и IP SLA



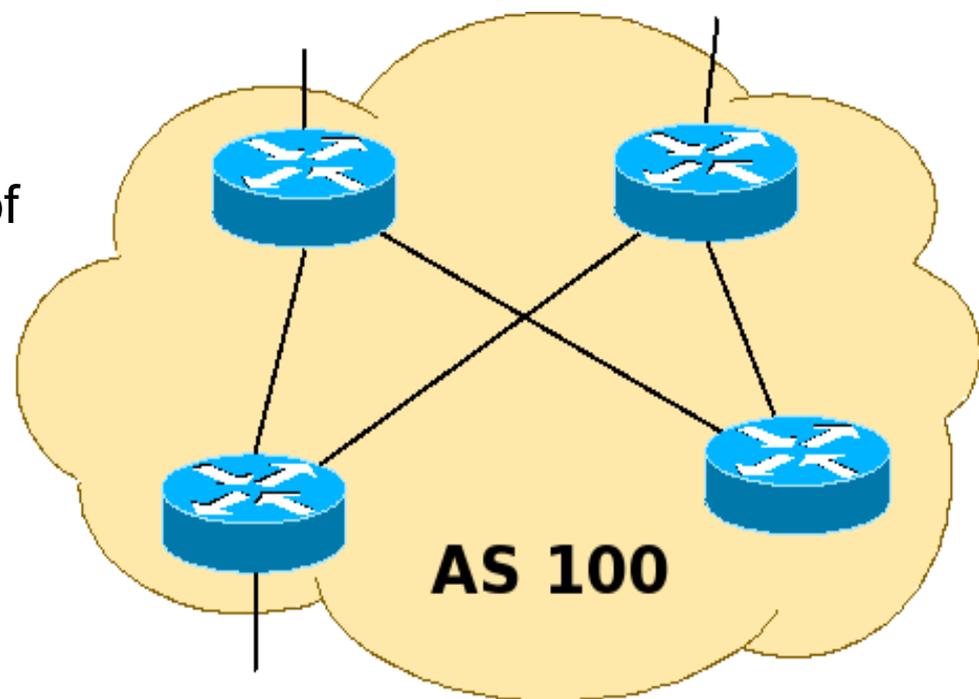
Наташа Самойленко

# Терминология

# Автономная система (AS)

- Автономная система (AS) — это система IP-сетей и маршрутизаторов, управляемых одним или несколькими операторами, имеющими единую политику маршрутизации с Интернетом.

- Autonomous System (AS) is a collection of connected Internet Protocol (IP) routing prefixes under the control of one or more network operators that presents a common, clearly defined routing policy to the Internet. (RFC 1930)



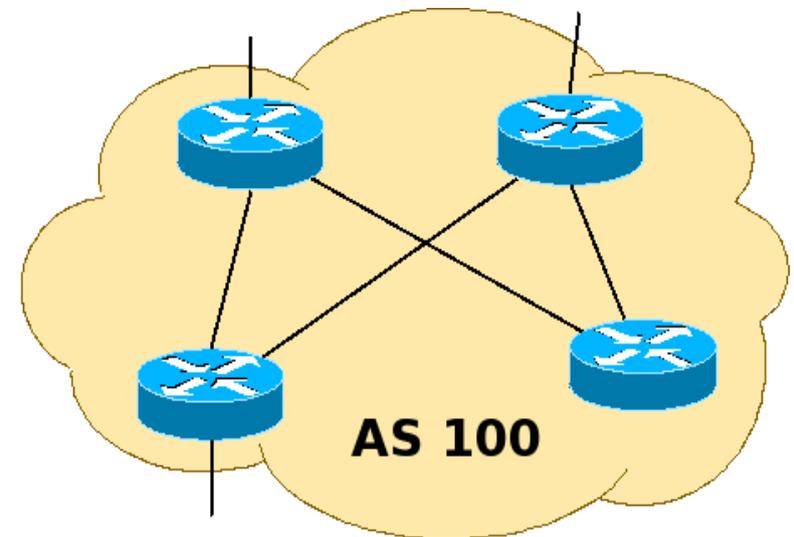
# Номер автономной системы (ASN)

## Два диапазона:

- 0-65535 (изначально определенный диапазон для ASN 16-bit)
- 65536-4294967295 (новый диапазон для ASN 32-bit (RFC4893))

## Использование:

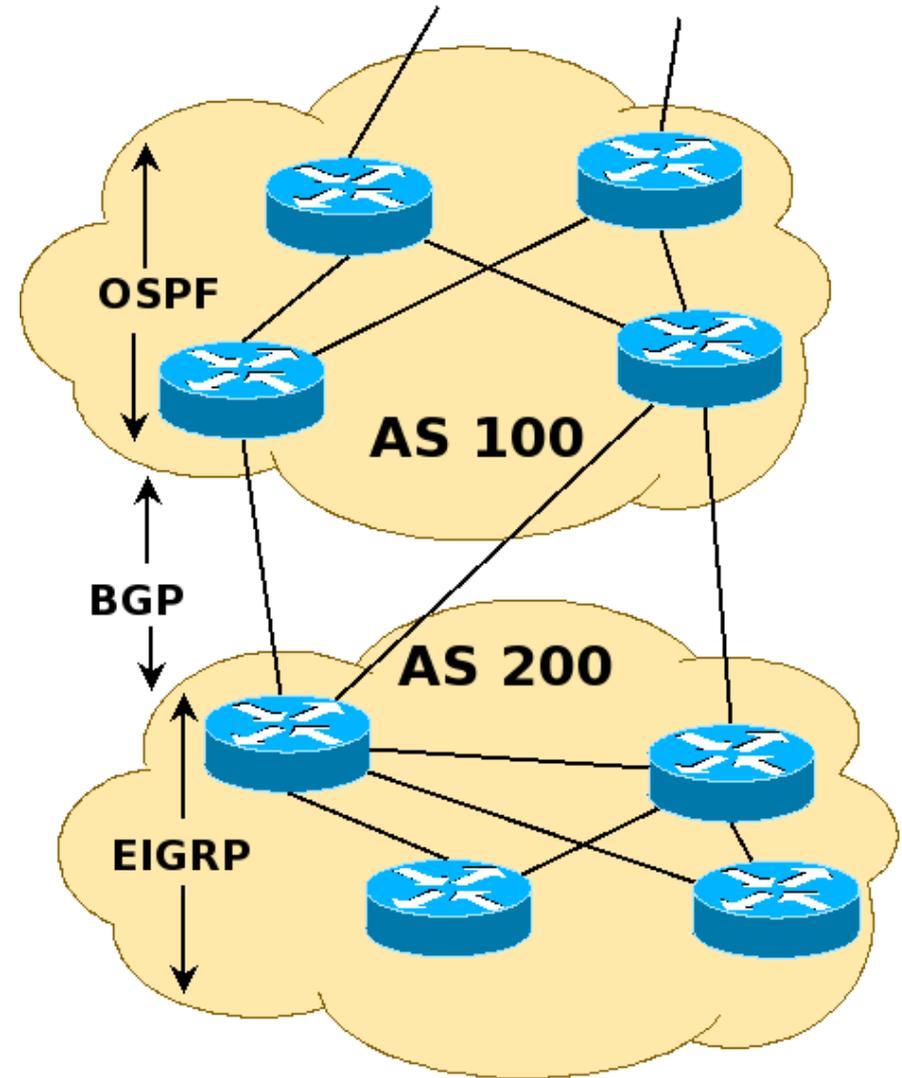
- 0 и 65535 (зарезервированы)
- 1-64495 (публичные номера)
- 65552-4294967295 (публичные номера)
- 64512-65534 (приватные номера)
- 23456 (представляет 32-bit диапазон на устройствах,
- которые работают с 16-bit диапазоном)



# Протоколы маршрутизации

Interior gateway protocol – протокол, который используется для передачи информации о маршрутах, внутри автономной системы.

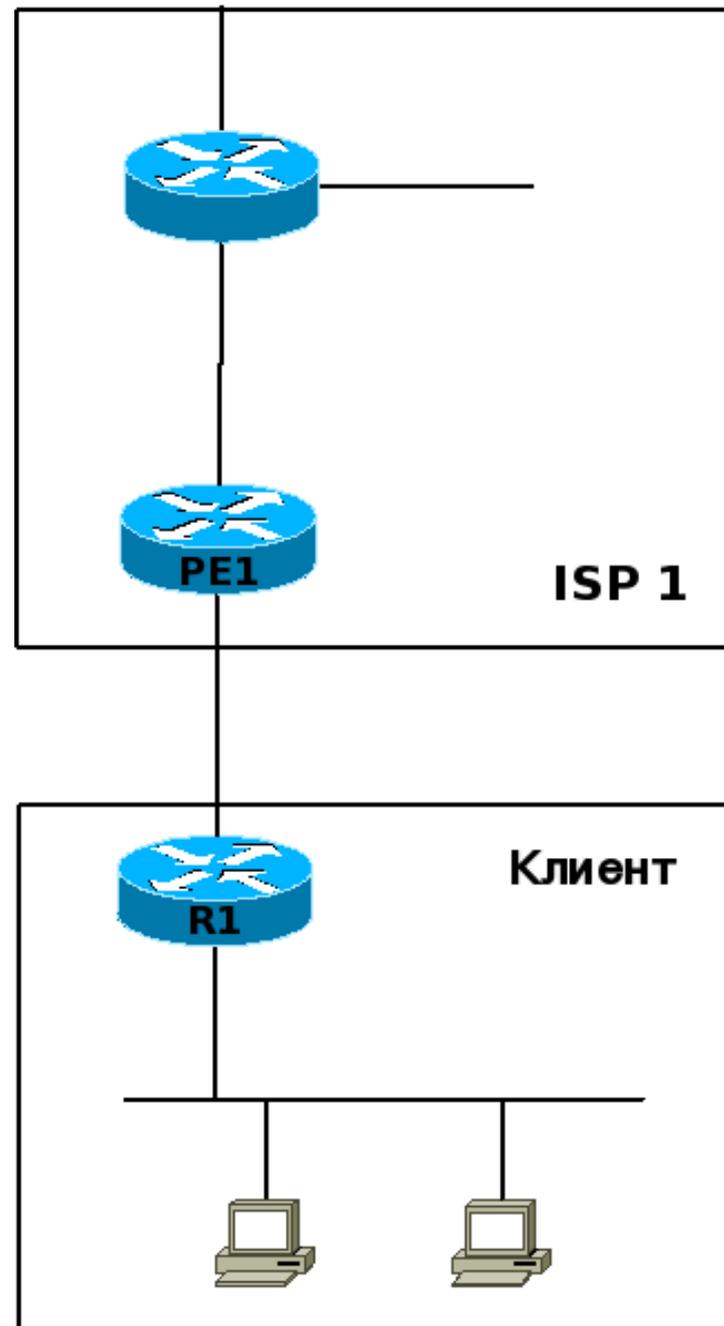
Exterior gateway protocol – протокол, который используется для передачи информации о маршрутах, между автономными системами.



# Варианты подключения к ISP

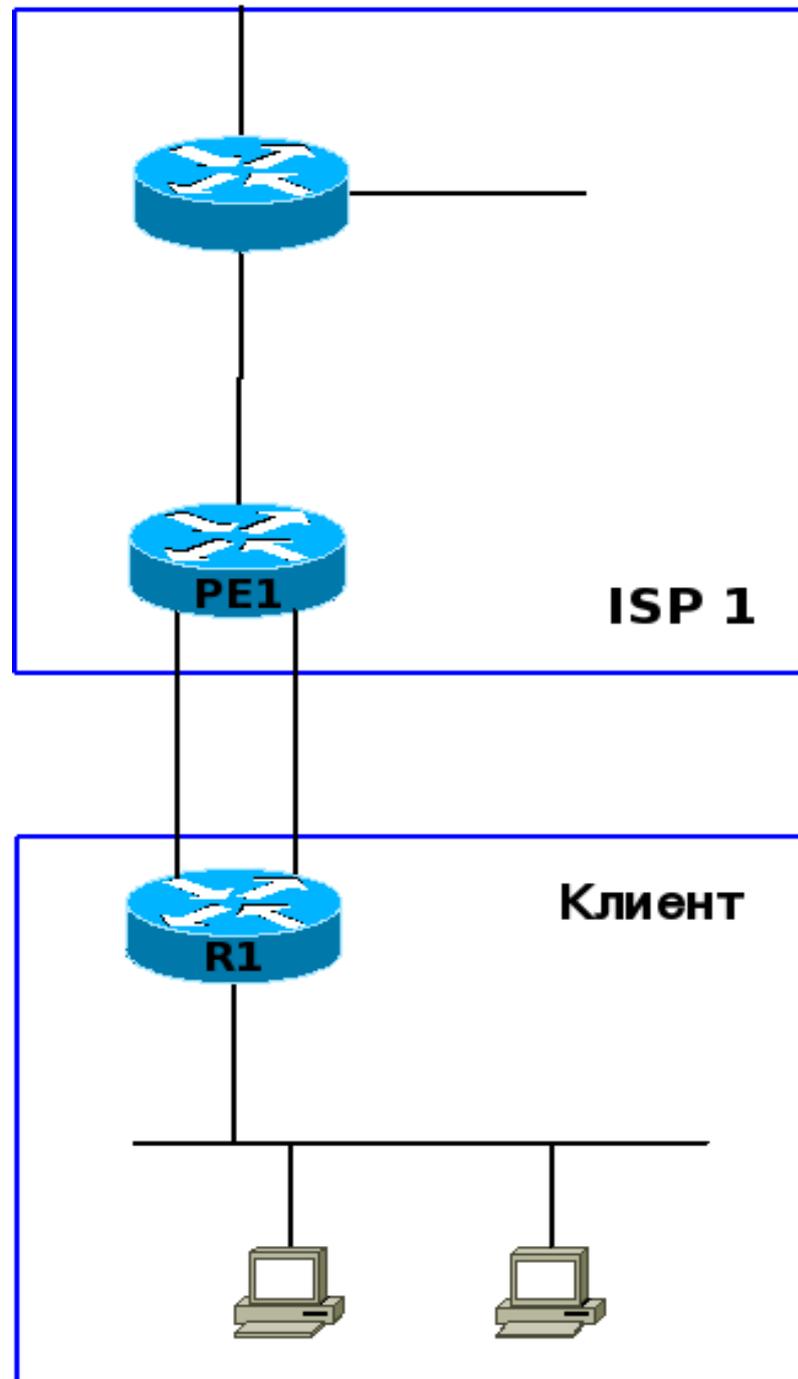
# Подключение к одному провайдеру (single-homed customer)

- Отсутствует избыточность:
  - по устройствам
  - по подключению к провайдеру
  - по провайдеру
- Маршрутизация:
  - статическая маршрутизация
- Адресация
  - RA блок адресов
- Автономная система
  - Не требуется



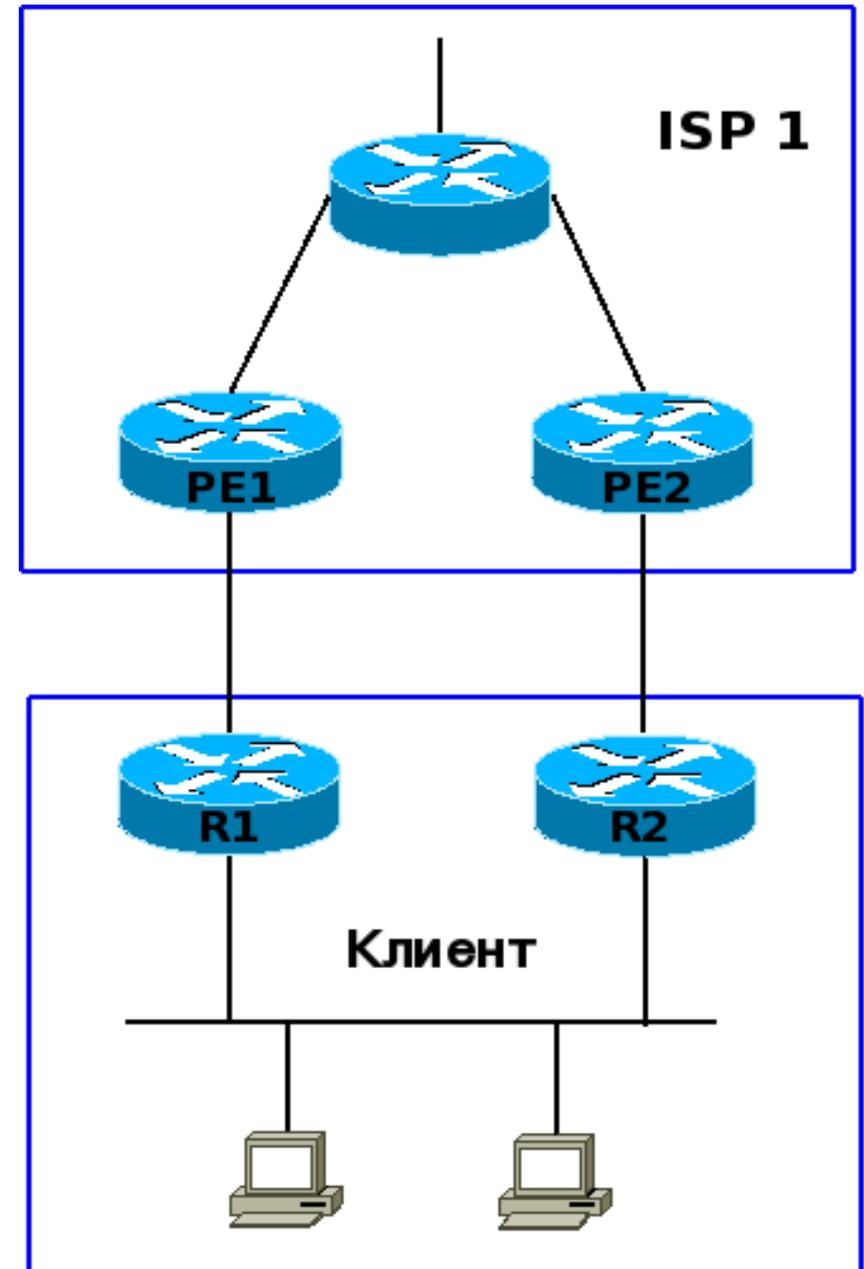
# Избыточное подключение к одному провайдеру

- Отсутствует избыточность:
  - по устройствам
  - по провайдеру
- Маршрутизация:
  - статическая маршрутизация
  - BGP
- Адресация
  - RA блок адресов
- Автономная система
  - Не требуется
  - Приватный номер (с BGP)



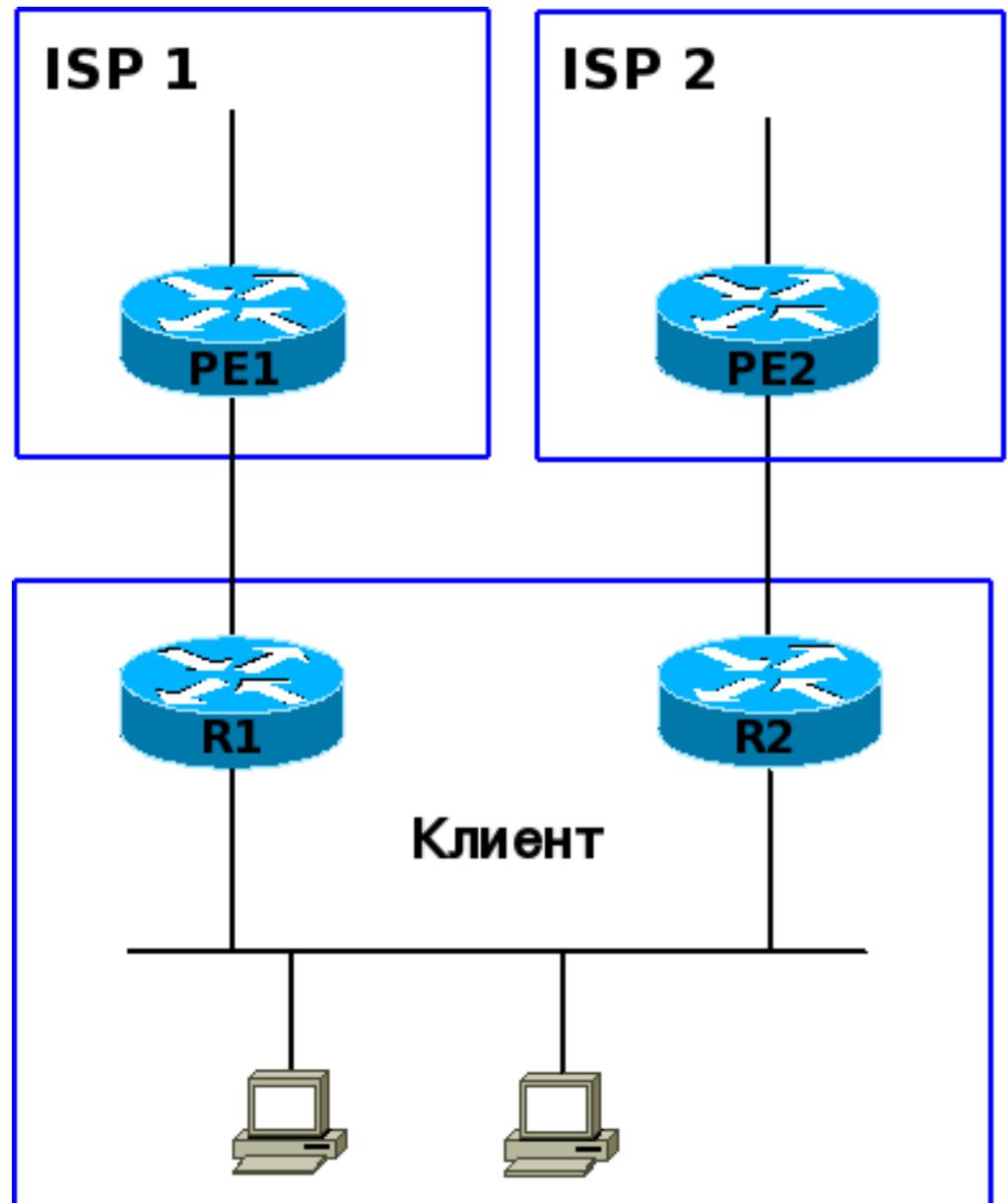
# Избыточное (по устройствам) подключение к одному провайдеру

- Отсутствует избыточность:
  - по провайдеру
- Маршрутизация:
  - BGP
  - статическая маршрутизация
- Адресация
  - RA блок адресов
- Автономная система
  - Приватный номер



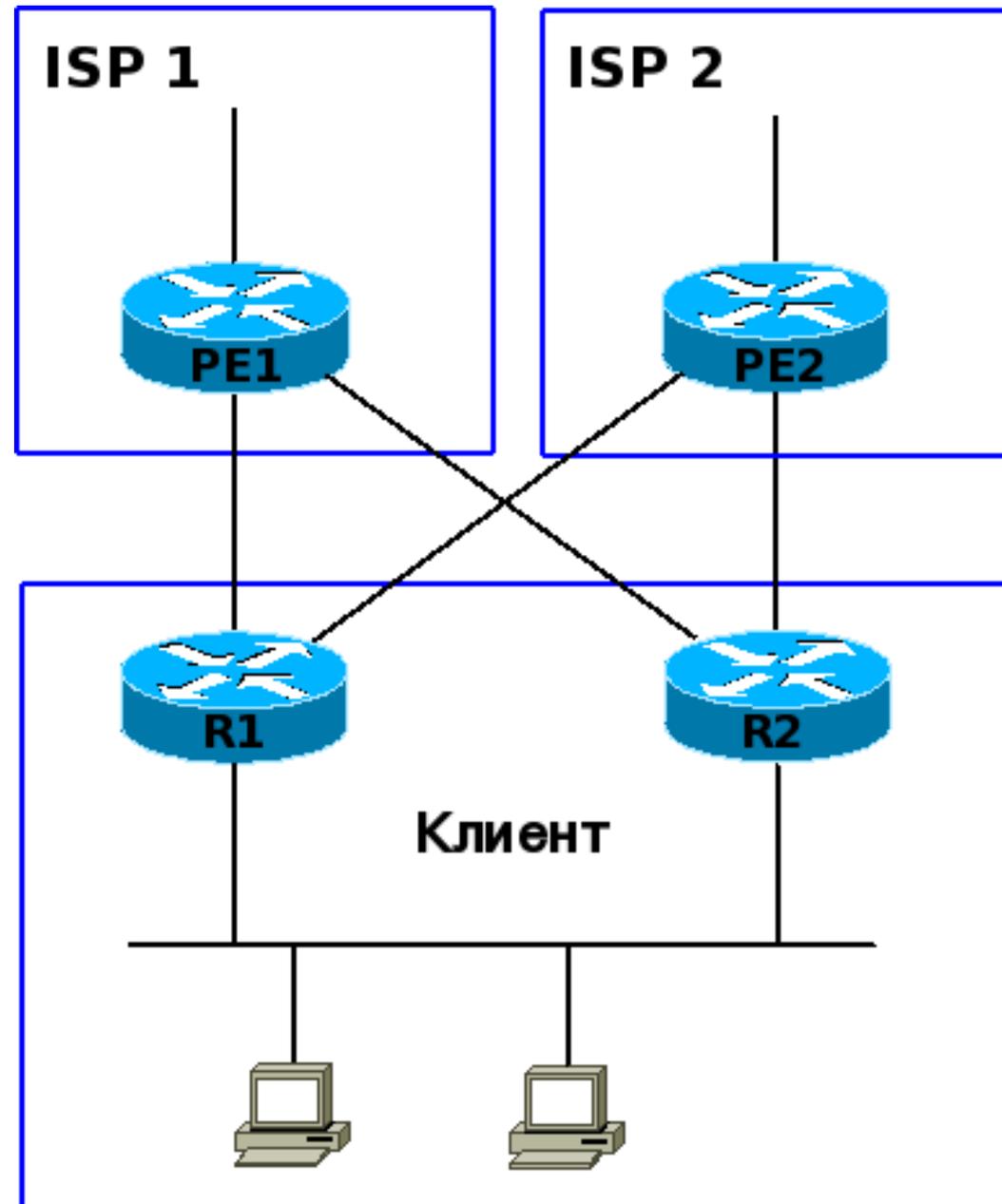
# Избыточное подключение к двум провайдерам (multi-homed customer)

- Избыточное подключение
- Маршрутизация:
  - BGP
- Адресация:
  - PI блок адресов
- Автономная система
  - Публичный номер



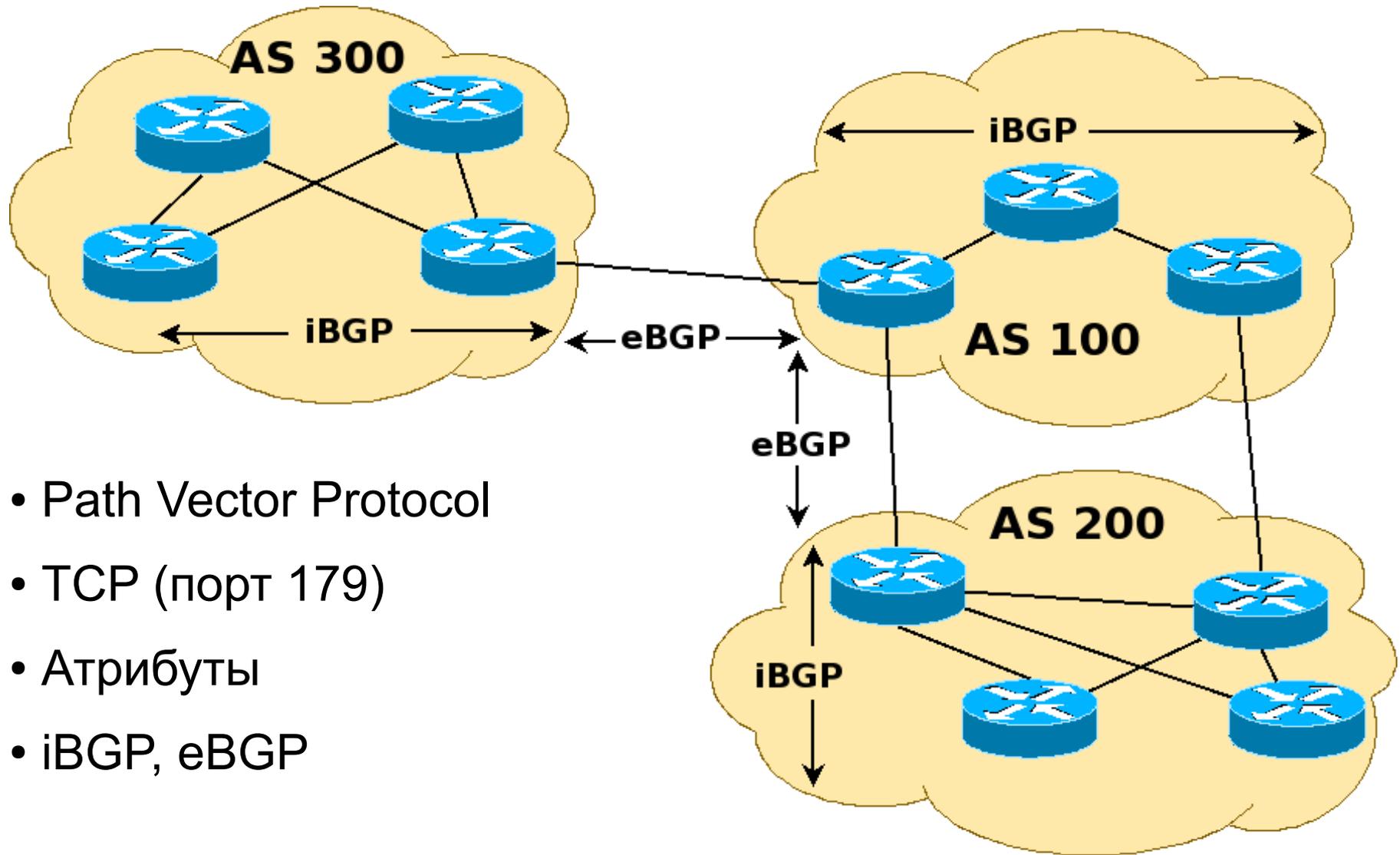
# Избыточное подключение к двум провайдерам (multi-homed customer)

- Полностью избыточное подключение
- Маршрутизация:
  - BGP
- Адресация:
  - PI блок адресов
- Автономная система
  - Публичный номер



# ОСНОВЫ BGP

# Border Gateway Protocol



- Path Vector Protocol
- TCP (порт 179)
- Атрибуты
- iBGP, eBGP

# Типы сообщений BGP

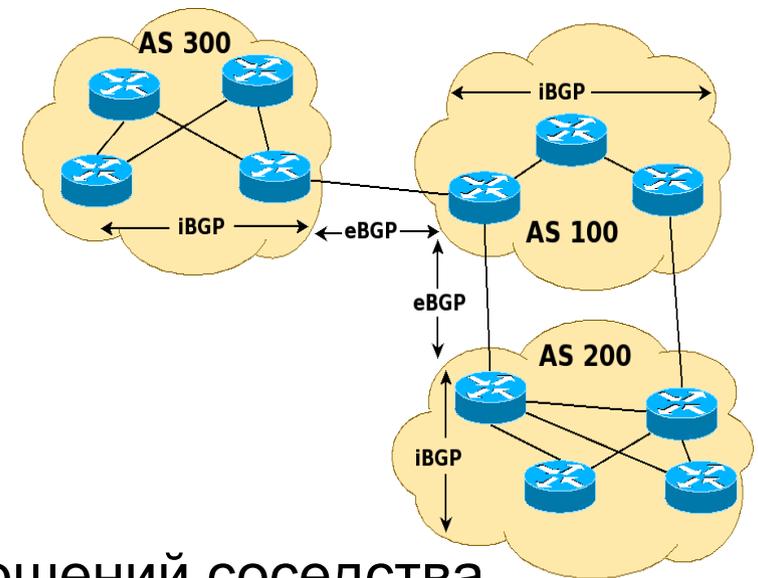
- Open** используется для установки отношений соседства и обмена базовыми параметрами. Отправляется сразу после установки TCP-соединения.
- Update** используется для обмена информацией о маршрутах.
- Notification** отправляется когда возникают ошибки BGP. После отправки сообщения сессия с соседом разрывается.
- Keepalive** используется для поддержания отношений соседства, для обнаружения неактивных соседей.

# BGP-соседи

BGP не обнаруживает соседей автоматически, как IGP протоколы. Каждый сосед должен быть настроен.

Соседи BGP разделяются на:

- Внутренних (iBGP)
- Внешних (eBGP)



Тип соседа мало влияет на установку отношений соседства.

Отличия между различными типами соседей проявляются в процессе отправки обновлений BGP и добавлении маршрутов в таблицу маршрутизации.

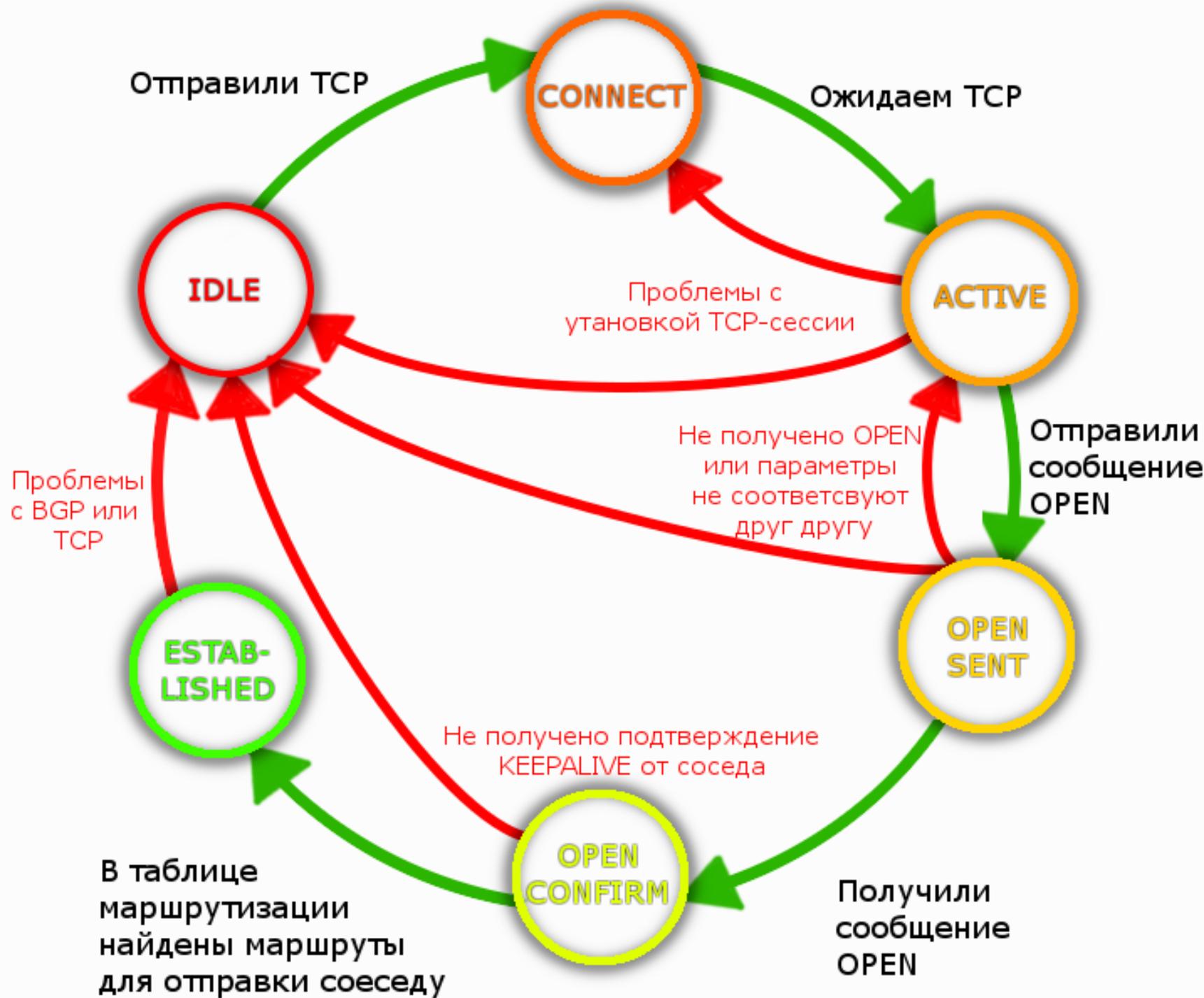
BGP выполняет такие проверки, когда формирует отношения соседства:

1. Маршрутизатор должен получить запрос на TCP-соединение с адресом отправителя, который указан в списке соседей.
2. Номер автономной системы локального маршрутизатора должен совпадать с номером автономной системы, который указан на соседнем маршрутизаторе командой `neighbor remote-as`
3. Идентификаторы маршрутизаторов (Router ID) не должны совпадать.
4. Если настроена аутентификация, то соседи должны пройти её.

# Состояния связи с соседями

- Idle
- Connect
- Open sent
- Open confirm
- Active
- Established

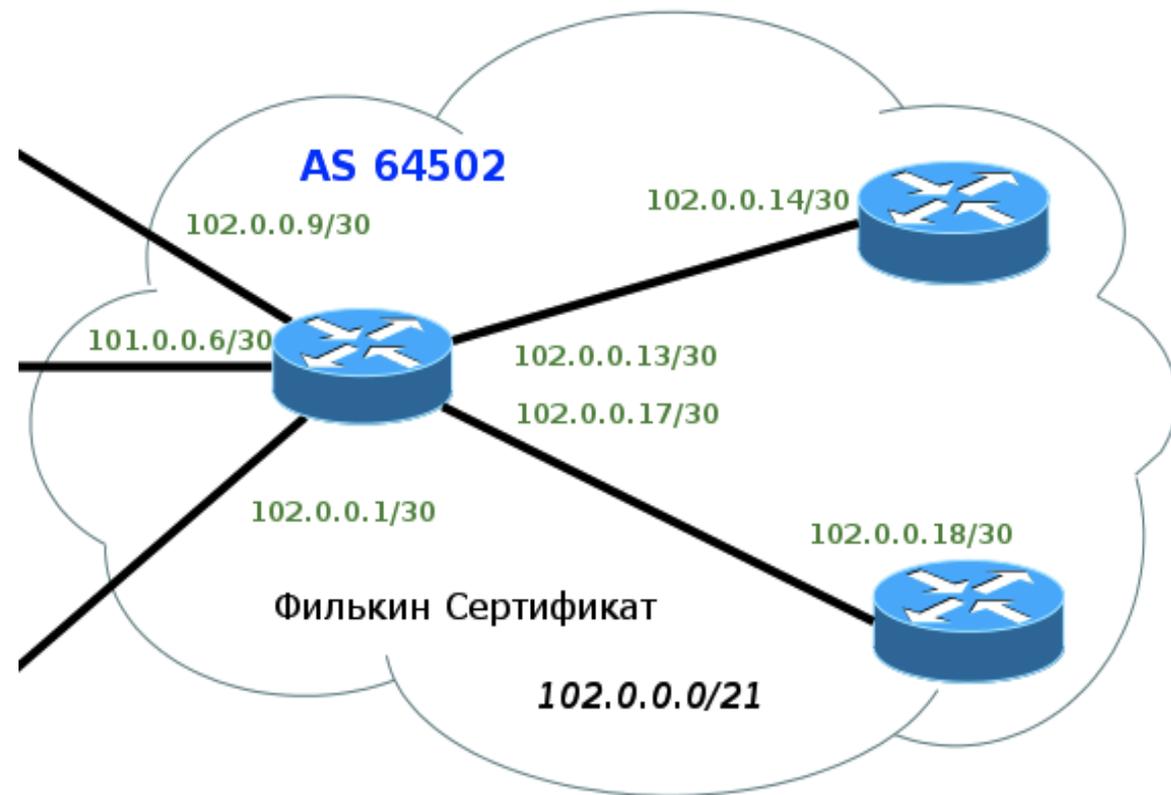
Состояние	Ожидание TSP?	Инициация TSP?	Установлено TSP?	Отправлено Open?	Получено Open?	Сосед Up?
Idle	Нет					
Connect	Да					
Active	Да	Да				
Open sent	Да	Да	Да	Да		
Open confirm	Да	Да	Да	Да	Да	
Established	Да	Да	Да	Да	Да	Да



# iBGP-соседи

Внутренний BGP-сосед (iBGP-сосед) — сосед, который находится в той же автономной системе, что и локальный маршрутизатор.

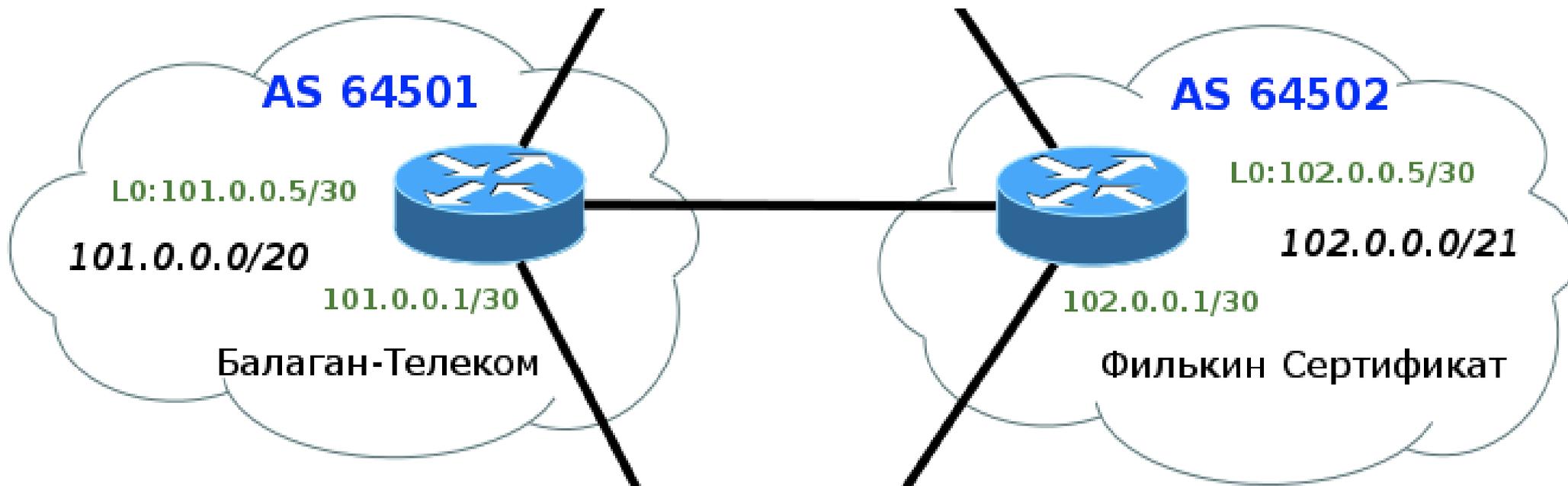
- Все iBGP-соседи внутри автономной системы должны быть соединены полносвязной топологией.
- iBGP-соседи не обязательно должны быть непосредственно соединены.



# eBGP-соседи

Внешний BGP-сосед (eBGP-сосед) — сосед, который находится в автономной системе отличной от локального маршрутизатора.

По умолчанию, eBGP-соседи должны быть непосредственно соединены.

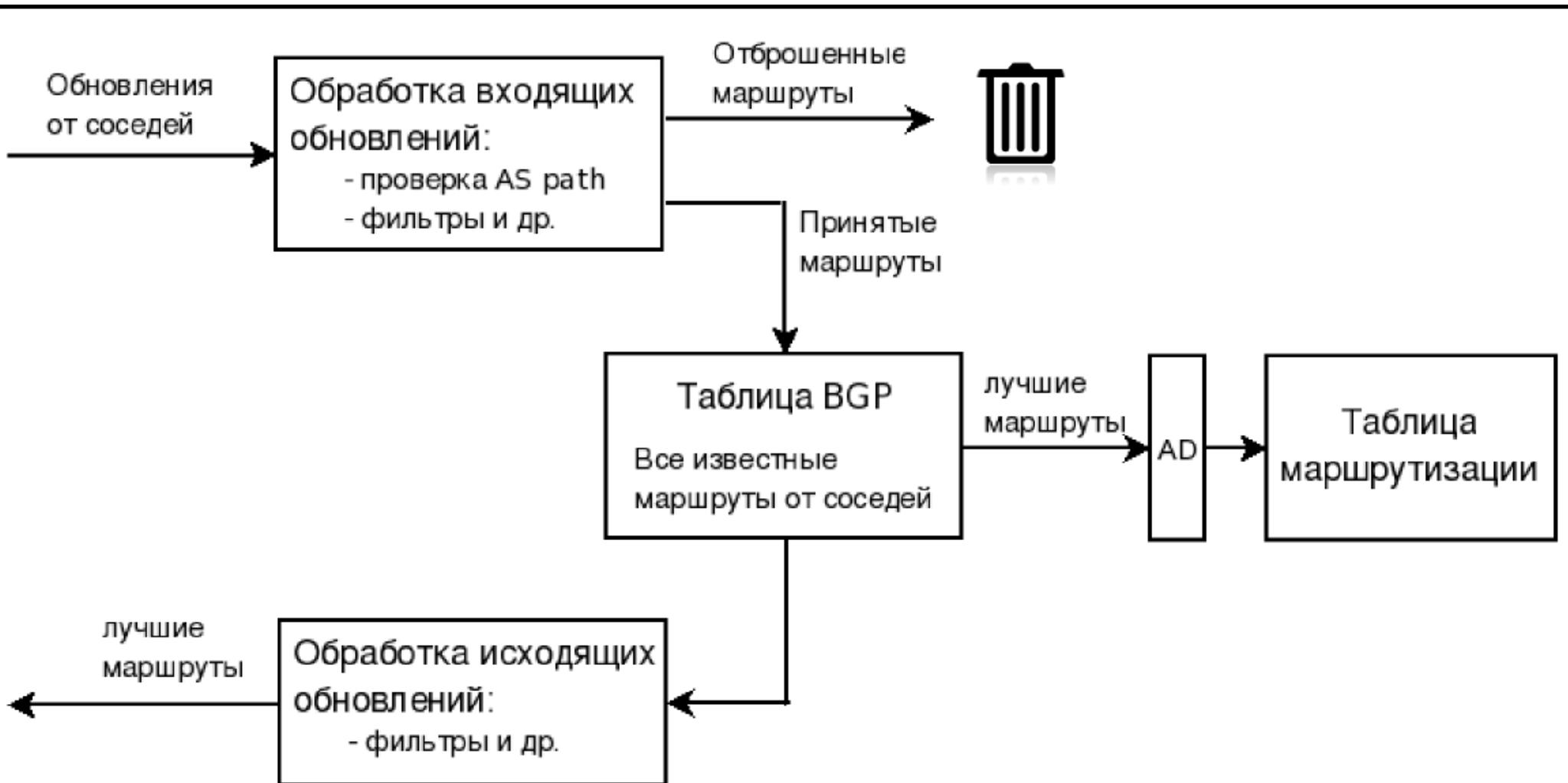


# Информация BGP

- Таблица соседей — соседи BGP
- Таблица BGP:
  - Список всех сетей выученных от каждого соседа
  - Может содержать несколько путей
  - Атрибуты BGP для каждого пути
- Таблица маршрутизации — лучшие маршруты к сетям

# Информация BGP

- Таблица соседей — соседи BGP
- Таблица BGP — все известные маршруты BGP
- Таблица маршрутизации — лучшие маршруты к сетям



# Атрибуты BGP

# Атрибуты BGP

Well-known mandatory	все маршрутизаторы должны распознавать эти атрибуты. Присутствуют во всех обновлениях
Well-known discretionary	все маршрутизаторы должны распознавать эти атрибуты. Могут присутствовать в обновлениях, но их присутствие не обязательно.
Optional transitive	могут не распознаваться всеми реализациями BGP. Если маршрутизатор не распознал атрибут, он отправляет его дальше соседям.
Optional non-transitive	могут не распознаваться всеми реализациями BGP. Если маршрутизатор не распознал атрибут, то атрибут при передаче соседям отбрасывается.

# Атрибуты BGP:

- \* Well-known mandatory:
  - o Autonomous system path
  - o Next-hop
  - o Origin
  
- \* Well-known discretionary:
  - o Local preference
  - o Atomic aggregate
  
- \* Optional transitive:
  - o Aggregator
  - o Communities
  
- \* Optional non-transitive:
  - o Multi-exit discriminator (MED)

# **Атрибуты BGP**

**Well-known mandatory**

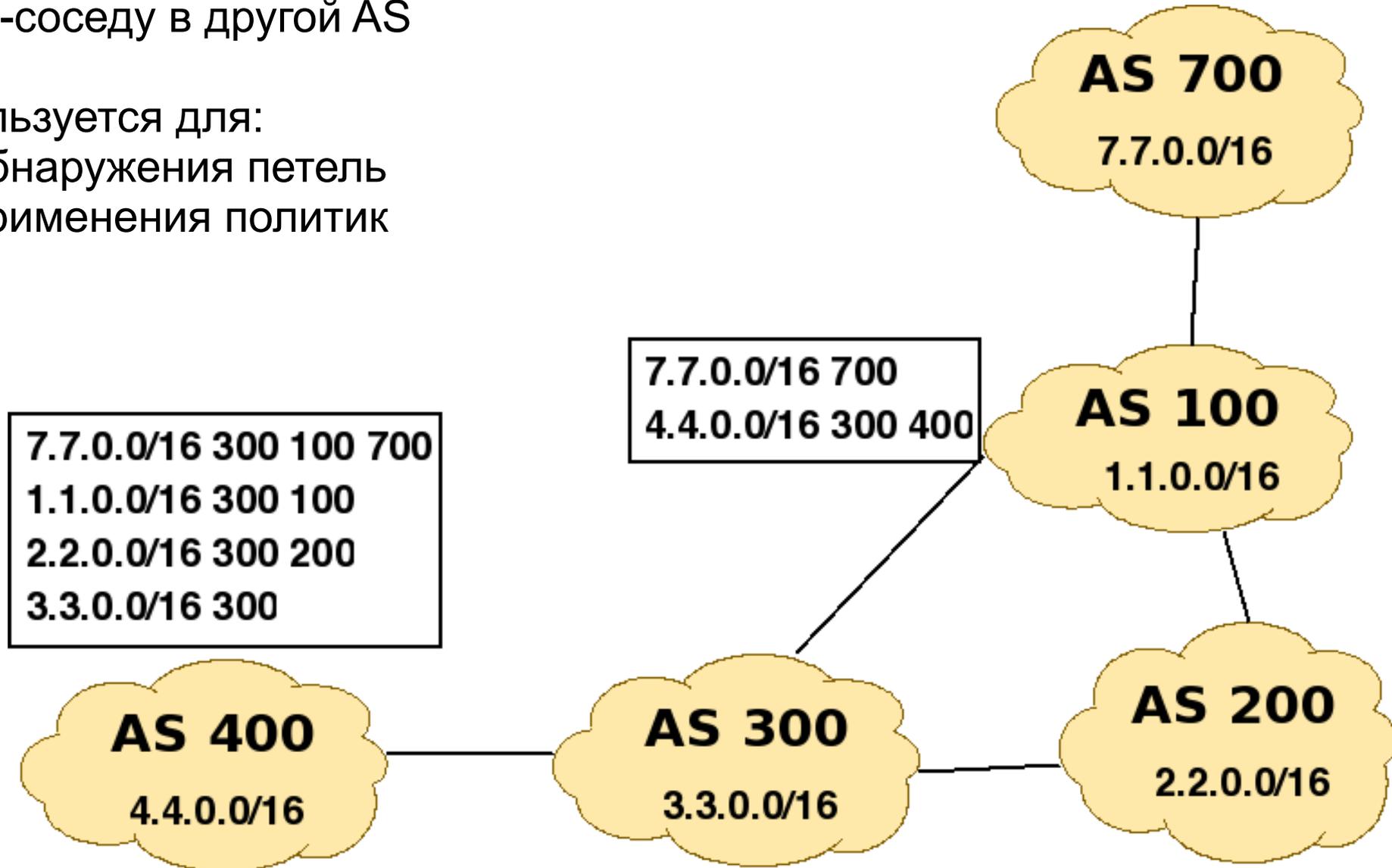
# Autonomous system path

Autonomous system path (AS Path):

- \* через какие автономные системы прошел маршрут
- \* номер AS добавляется при передаче обновления из одной AS eBGP-соседу в другой AS

Используется для:

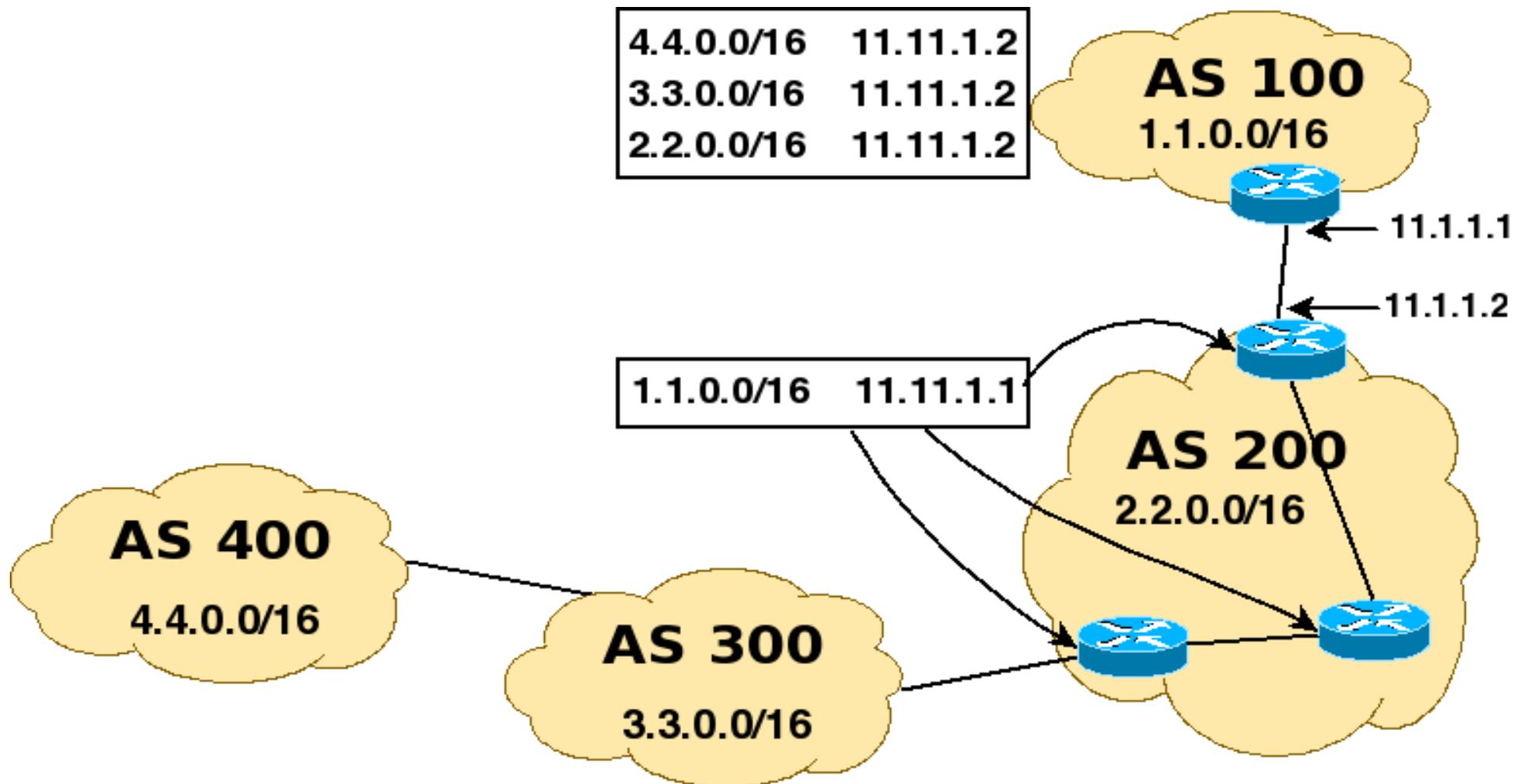
- \* обнаружения петель
- \* применения политик



# Next hop

Атрибут Next hop:

- \* IP-адрес следующей AS для достижения сети назначения.
- \* Это адрес eBGP-маршрутизатора, через который идет путь к сети назначения

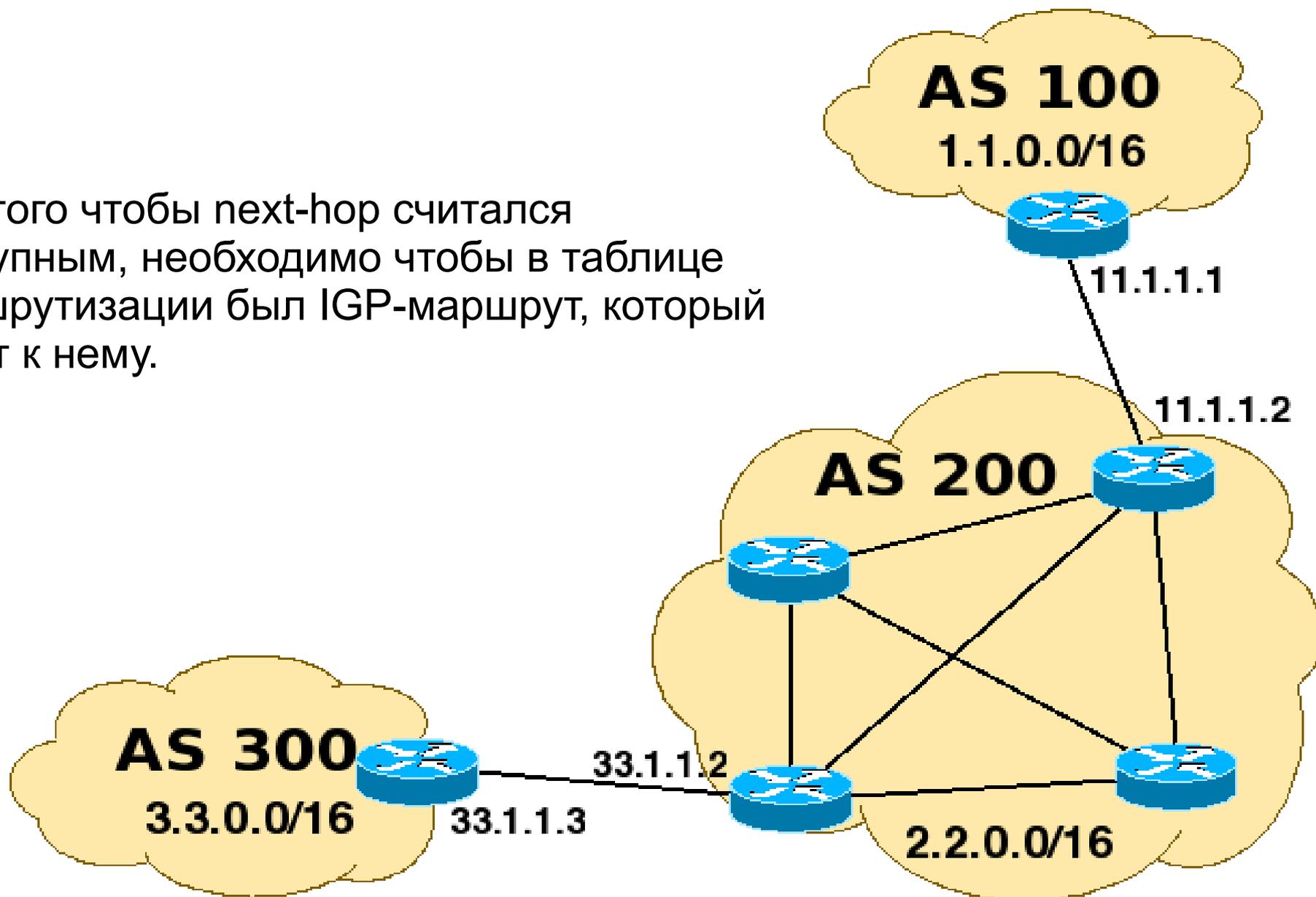


# iBGP и next hop

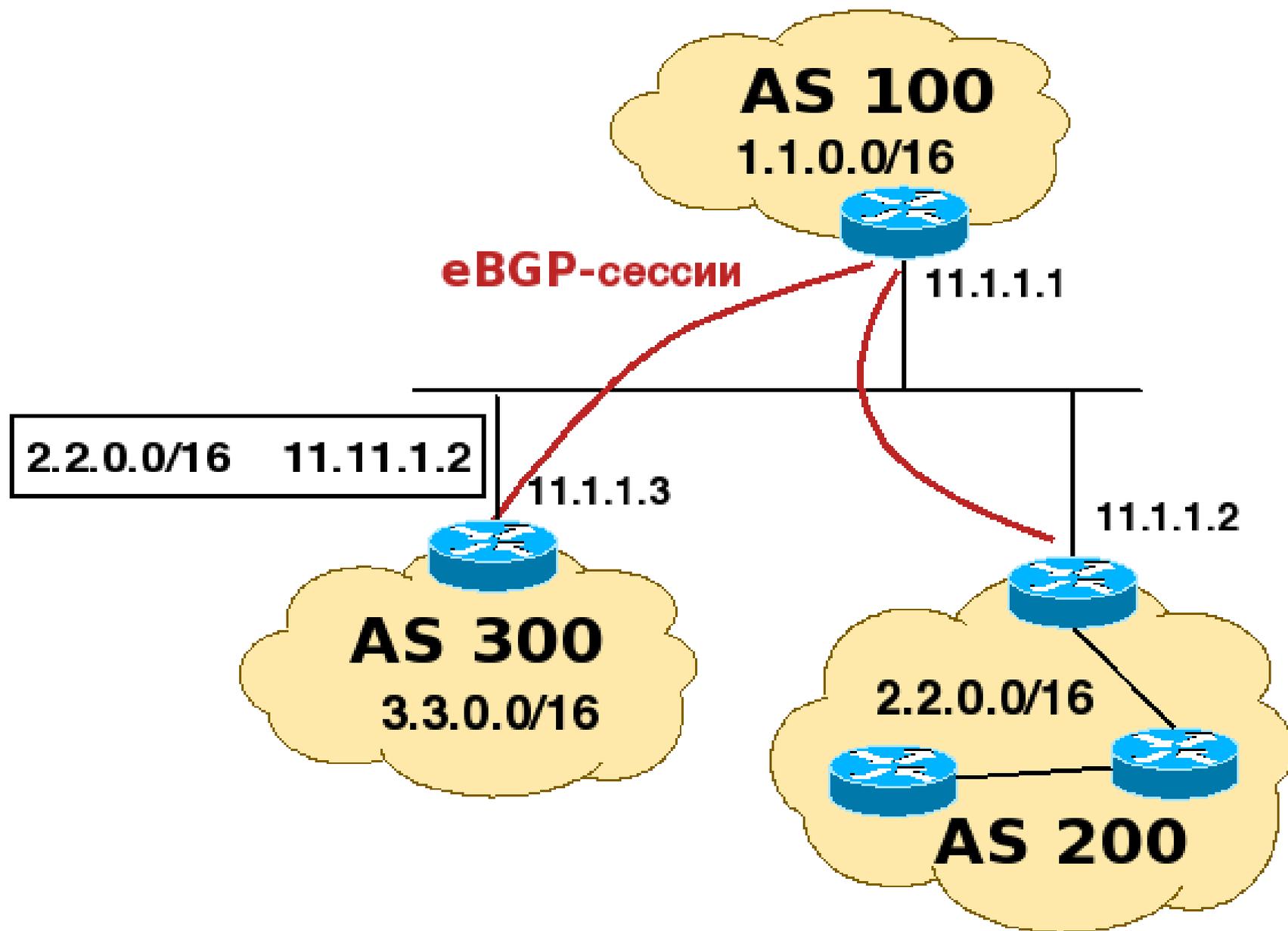
Первый шаг при выборе лучшего маршрута:

- доступность next-hop (Route Resolvability Condition)

Для того чтобы next-hop считался доступным, необходимо чтобы в таблице маршрутизации был IGP-маршрут, который ведет к нему.



# Third party next hop



# Origin

Атрибут Origin:

- \* указывает на “происхождение” маршрута в обновлении

Возможные значения:

- \* IGP — маршрут анонсирован в BGP (командой network)
- \* EGP — маршрут сгенерирован EGP
- \* Incomplete — перераспределенные маршруты

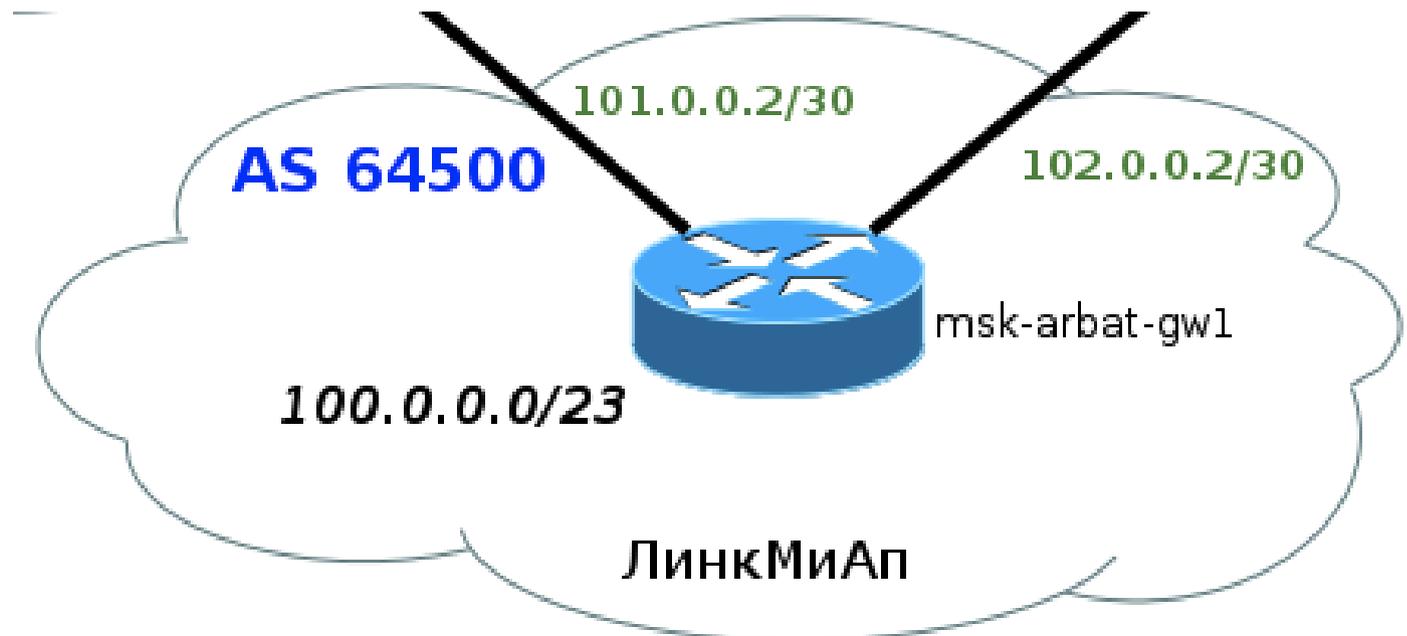
Учитывается при выборе маршрута.

# Базовые настройки BGP

# Создание процесса BGP

Создание процесса на msk-arbat-gw1:

```
msk-arbat-gw1(conf)# router bgp 64500  
msk-arbat-gw1(conf-router)#
```



На маршрутизаторе может существовать только один процесс BGP

# Настройка соседей BGP

Внутренний BGP-сосед (iBGP-сосед) — сосед, который находится в той же автономной системе, что и локальный маршрутизатор.

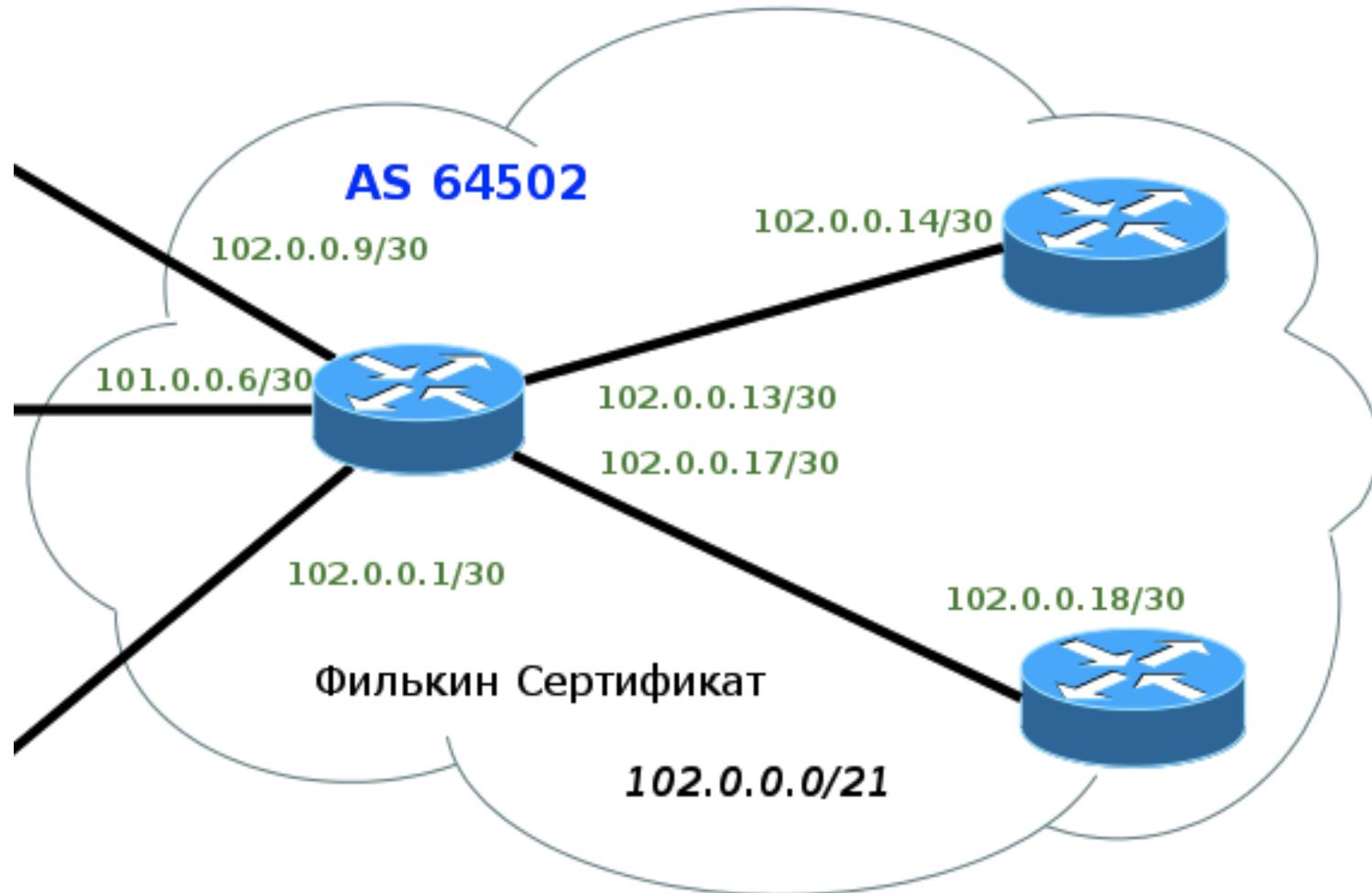
iBGP-соседи внутри автономной системы должны быть соединены полносвязной топологией.  
iBGP-соседи не обязательно должны быть непосредственно соединены.

Внешний BGP-сосед (eBGP-сосед) — сосед, который находится в автономной системе отличной от локального маршрутизатора.

По умолчанию, eBGP-соседи должны быть непосредственно соединены.

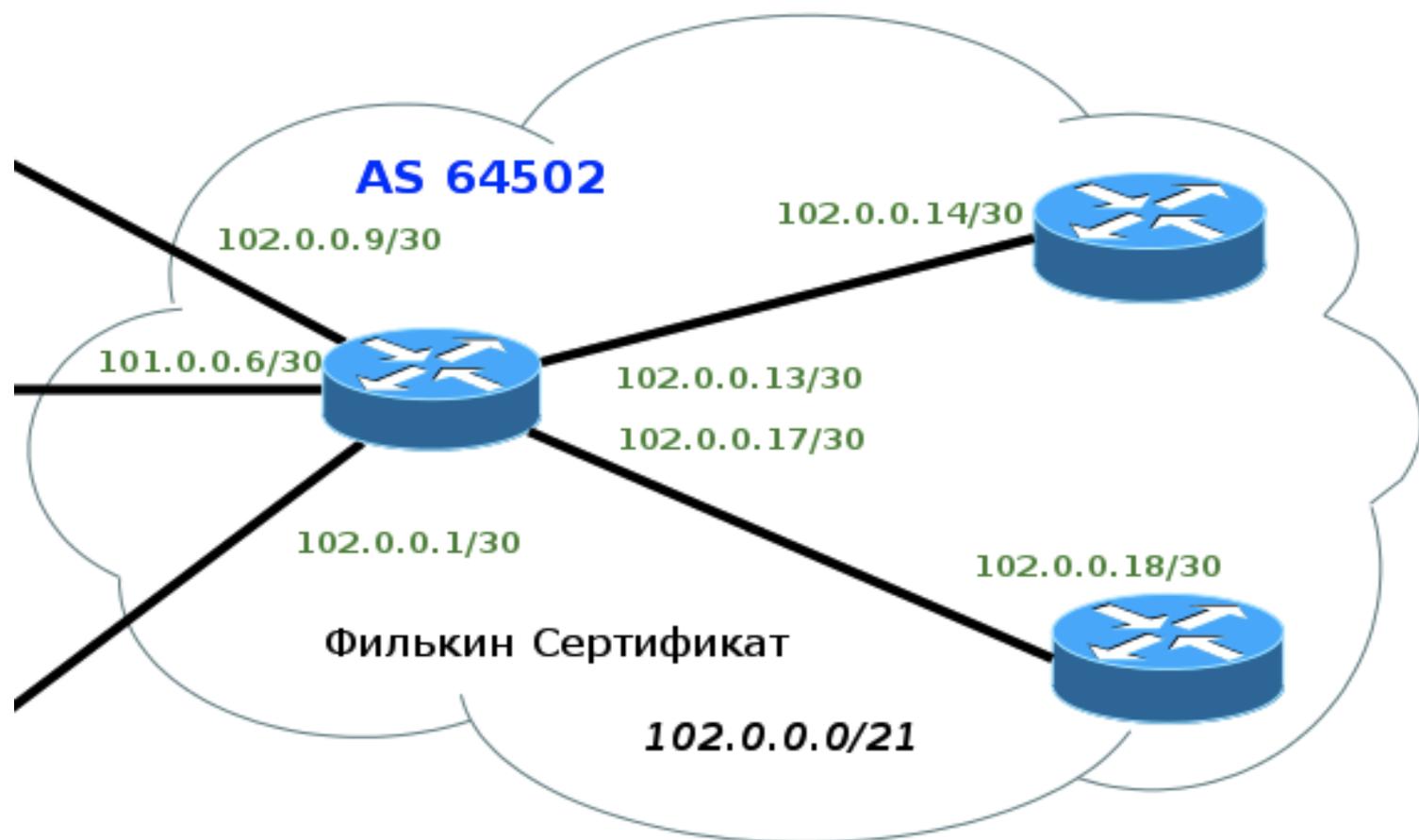
# Настройка внутренних соседей

```
router bgp 64502
  neighbor 102.0.0.14 remote-as 64502
  neighbor 102.0.0.18 remote-as 64502
```



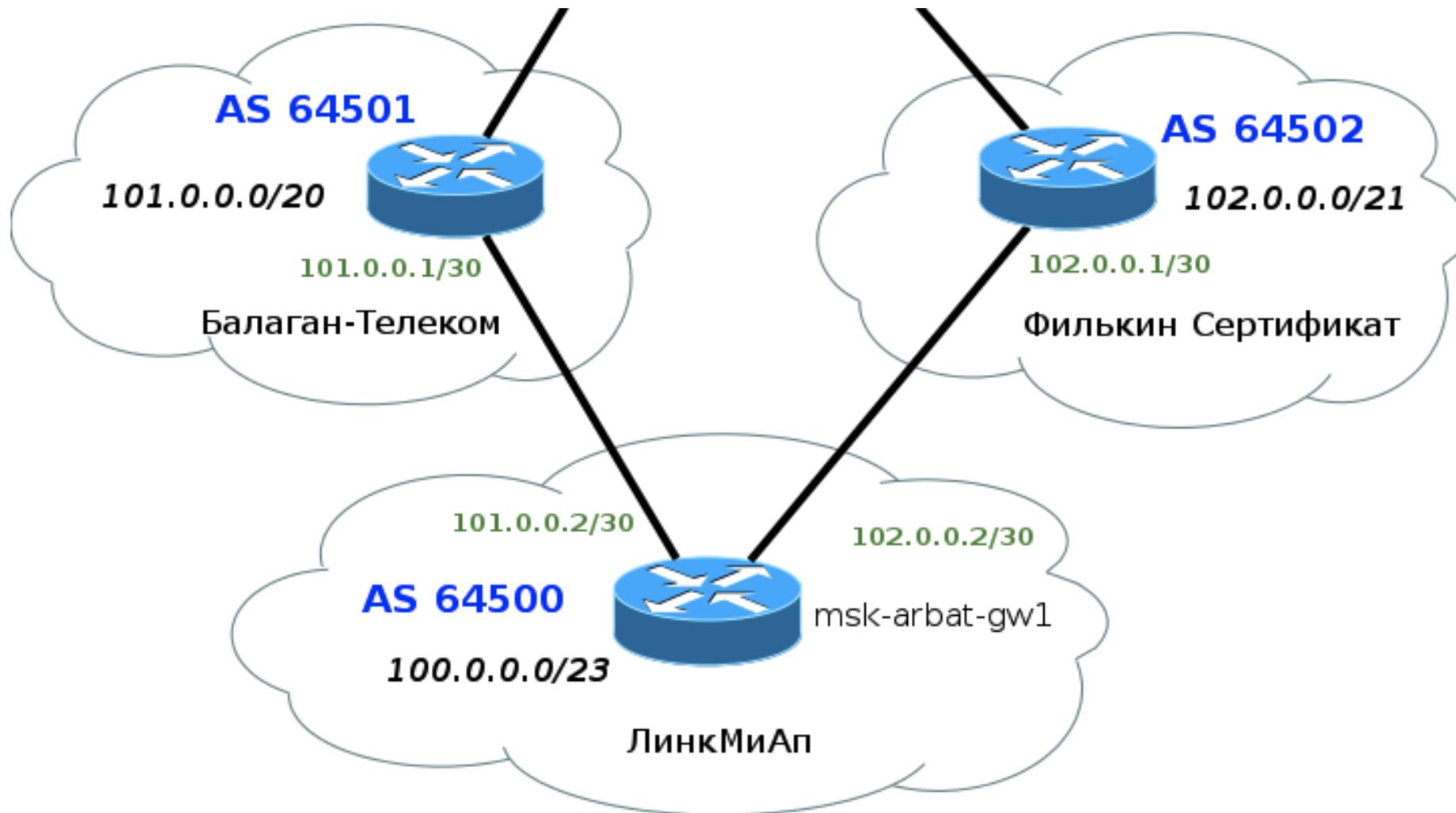
## Настройка внутренних соседей. Использование next-hop-self

```
router bgp 64502
 neighbor 102.0.0.14 remote-as 64502
 neighbor 102.0.0.14 next-hop-self
 neighbor 102.0.0.18 remote-as 64502
 neighbor 102.0.0.18 next-hop-self
```



# Настройка внешних соседей BGP

```
router bgp 64500  
neighbor 101.0.0.1 remote-as 64501  
neighbor 102.0.0.1 remote-as 64502
```



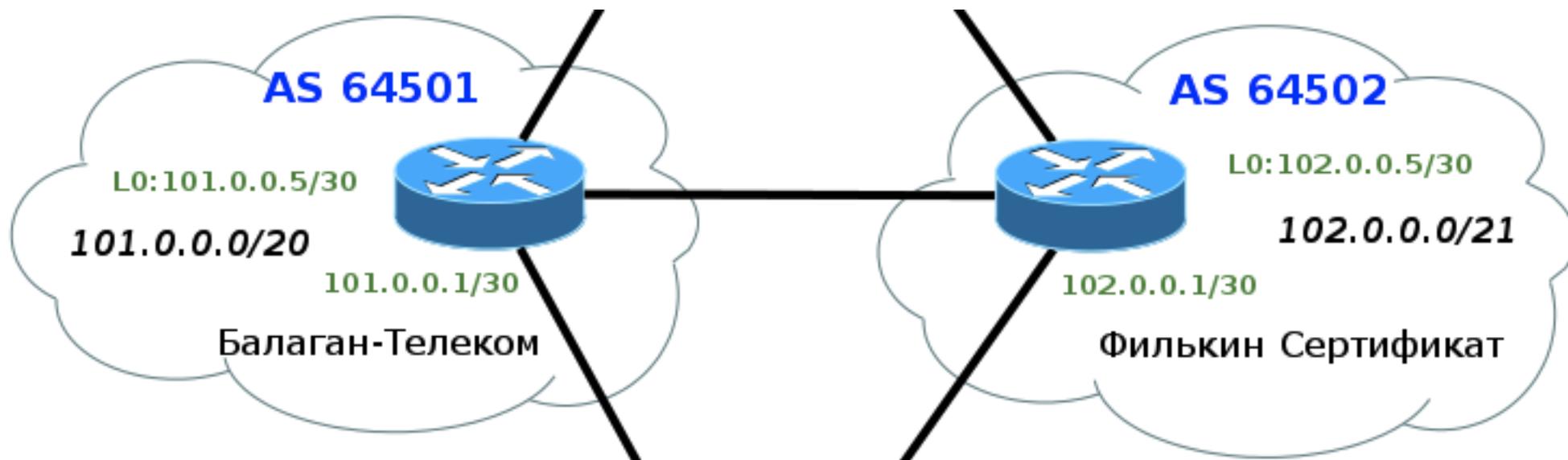
# Настройка внешних соседей BGP

```
router bgp 64501
 neighbor 102.0.0.5 remote-as 64502
 neighbor 102.0.0.5 ebgp-multihop 5
 neighbor 102.0.0.5 update-source Loopback0
```

```
ip route 192.168.100.1 255.255.255.255 192.168.1.2
```

```
router bgp 64502
 neighbor 101.0.0.5 remote-as 64501
 neighbor 101.0.0.5 ebgp-multihop 5
 neighbor 101.0.0.5 update-source Loopback0
```

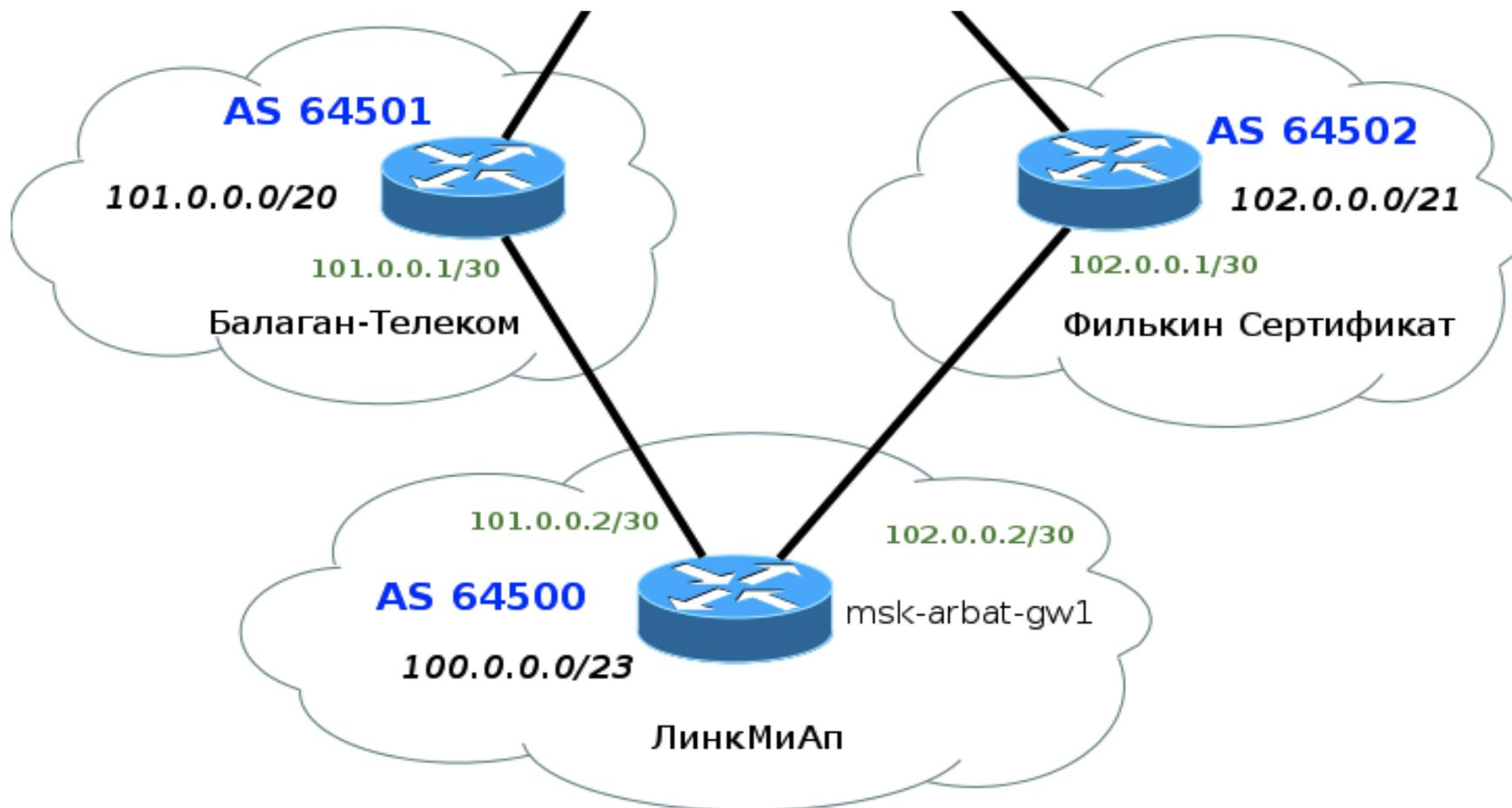
```
ip route 192.168.100.1 255.255.255.255 192.168.1.2
```



# Анонсирование сетей в BGP

Два механизма анонсирования сетей:

- команда `network`
- команда `redistribute`

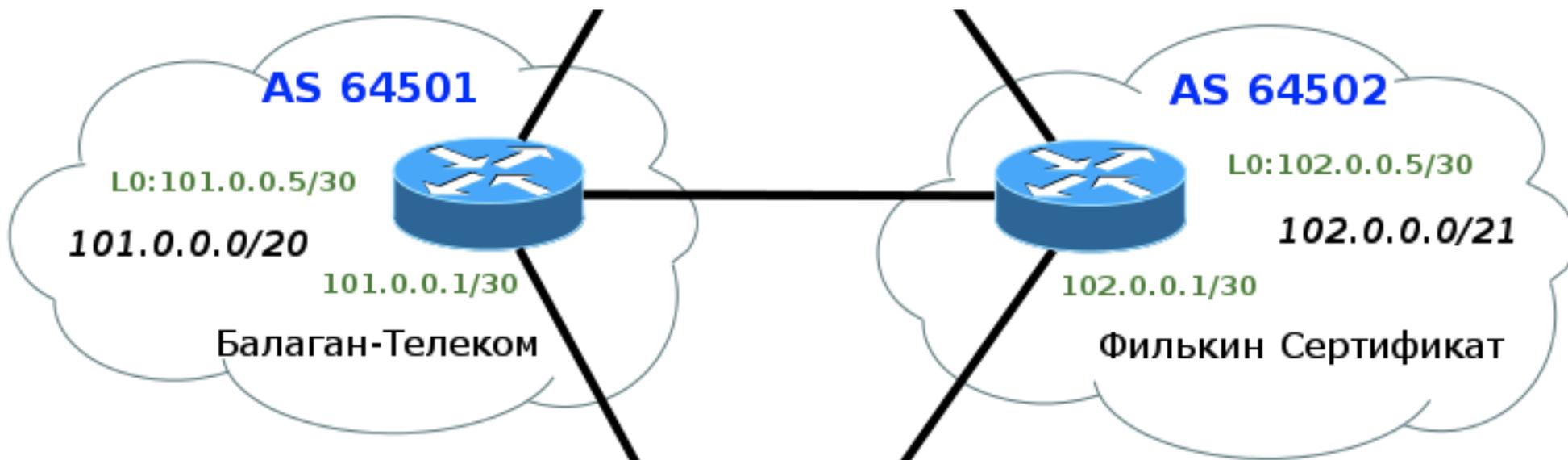


# Анонсирование сетей командой network

Соответствующий маршрут должен существовать в таблице маршрутизации, прежде чем сеть будет анонсирована

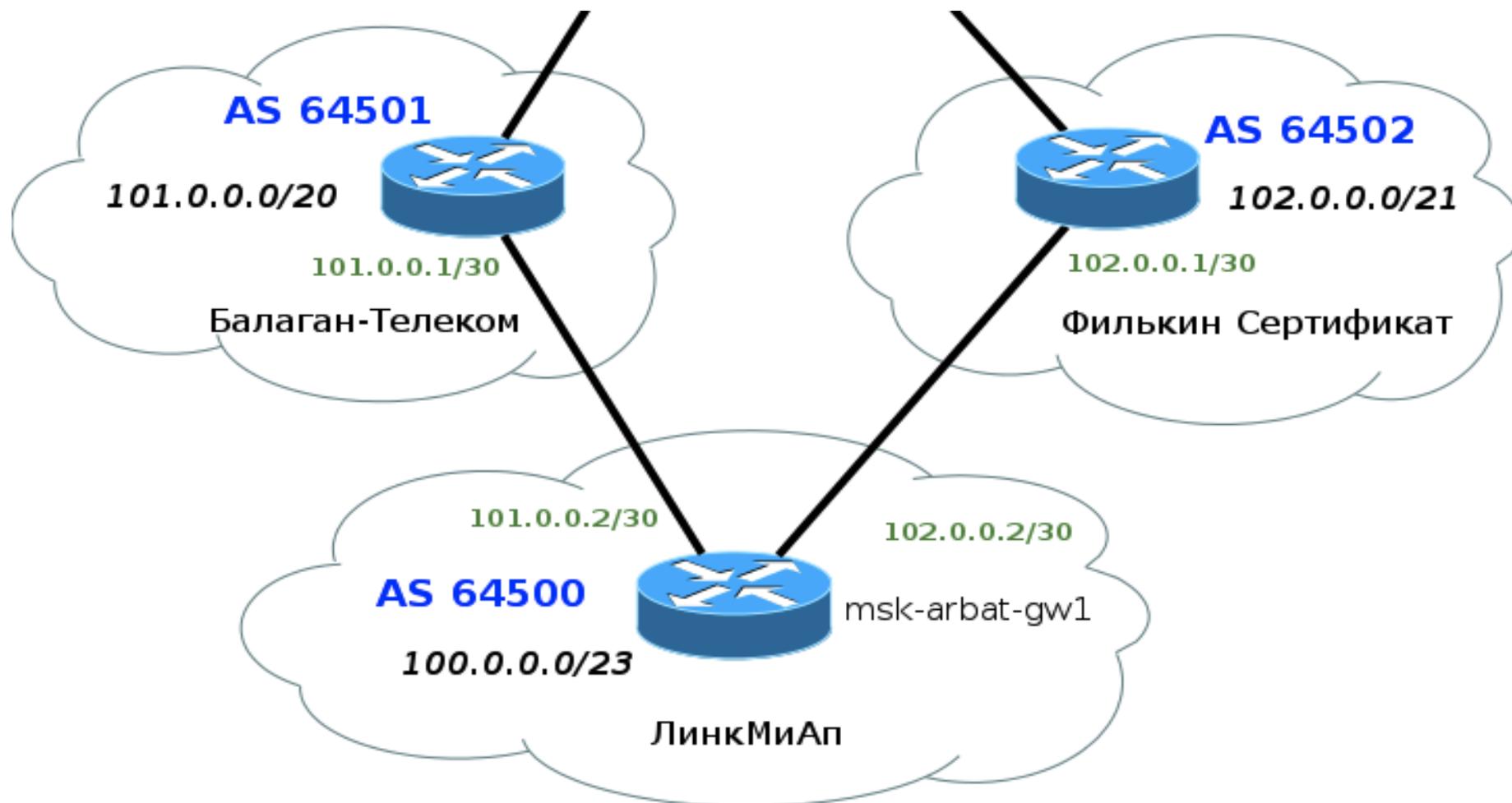
Значение атрибута Origin будет "IGP"

```
router bgp 64501
network 101.0.0.0 mask 255.255.240.0
```



```
router bgp 64502
network 102.0.0.0 mask 255.255.248.0
```

# Анонсирование сетей командой network



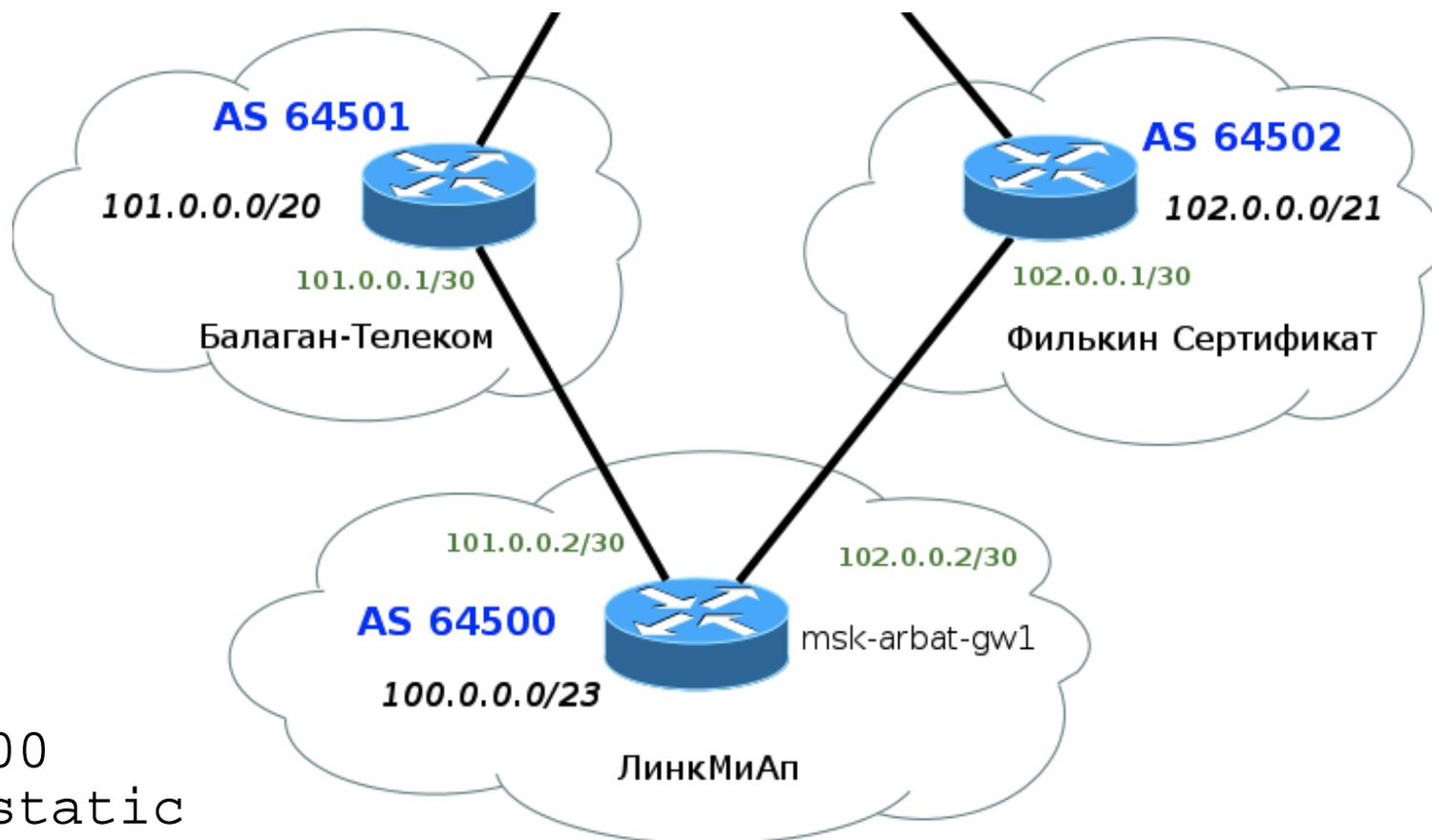
```
router bgp 64500
 network 100.0.0.0 mask 255.255.254.0

ip route 100.0.0.0 255.255.254.0 Null0
```

# Анонсирование сетей командой redistribute

Помещает маршруты протокола IGP или статические маршруты в процесс BGP.

Значение атрибута Origin будет "Incomplete"



```
router bgp 64500  
  redistribute static
```

```
ip route 100.0.0.0 255.255.254.0 192.168.101.3
```

# Просмотр информации BGP

# Настройки BGP

```
msk-arbat-gw1# sh run | s bgp
router bgp 64500
  no synchronization
  bgp log-neighbor-changes
  network 100.0.0.0 mask 255.255.254.0
  neighbor 101.0.0.1 remote-as 64501
  neighbor 101.0.0.1 prefix-list LAN out
  neighbor 101.0.0.1 weight 500
  neighbor 102.0.0.1 remote-as 64502
  neighbor 102.0.0.1 prefix-list LAN out
  neighbor 102.0.0.1 route-map INBOUND out
  no auto-summary
```

# Соседи BGP

## CORE# show ip bgp summary

BGP router identifier 192.168.7.6, local AS number 600  
BGP table version is 20, main routing table version 20  
13 network entries using 1716 bytes of memory  
17 path entries using 884 bytes of memory  
7/5 BGP path/bestpath attribute entries using 1036 bytes of memory  
4 BGP AS-PATH entries using 96 bytes of memory  
0 BGP route-map cache entries using 0 bytes of memory  
0 BGP filter-list cache entries using 0 bytes of memory  
Bitfield cache entries: current 3 (at peak 4) using 96 bytes of memory  
BGP using 3828 total bytes of memory  
BGP activity 13/0 prefixes, 19/2 paths, scan interval 60 secs

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	
192.168.3.2	4	100	3	8	0	0	0	04:35:36	Active
192.168.3.6	4	100	282	283	20	0	0	04:36:53	7
192.168.7.9	4	600	284	278	20	0	0	04:36:53	8
192.168.9.10	4	600	279	278	20	0	0	04:36:38	0

# Таблица BGP

```
dyn6# sh ip bgp
```

```
BGP table version is 20, local router ID is 192.168.7.6
```

```
Status codes: s suppressed, d damped, h history, * valid, > best,  
i - internal, r RIB-failure, S Stale
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 10.0.1.0/24	192.168.3.6			0 100 500	i
*> 192.168.1.0	192.168.3.6			0 100 500	i
*> 192.168.3.0	0.0.0.0	0		32768	i
*> 192.168.6.0	192.168.3.6			0 100 500	i
* i 192.168.7.0	192.168.7.9	0	100	0	i
*>	0.0.0.0	0		32768	i
*> 192.168.8.0	192.168.3.6			0 100 500	i
r>i 192.168.9.0	192.168.7.9	0	100	0	i
*> 192.168.10.0	192.168.3.6			0 100 500	i
* i	192.168.11.12		100	0 1200 1100	i
r>i 192.168.11.0	192.168.7.9	0	100	0	i
* 192.168.12.0	192.168.3.6			0 100 500 1100	i
*>i	192.168.11.12		100	0 1200 1100	i
*>i 192.168.13.0	192.168.11.12	0	100	0 1200	i
*>i	192.168.11.12	0	100	0 1200	i
r>i 192.168.16.0	192.168.7.9	0	100	0	i

## Подробная информация о маршруте в таблице BGP

```
dyn6# show ip bgp 192.168.10.0
```

```
BGP routing table entry for 192.168.10.0/24, version 19
```

```
Paths: (2 available, best #1, table Default-IP-Routing-Table)
```

```
  Advertised to update-groups:
```

```
    1
```

```
100 500
```

```
  192.168.3.6 from 192.168.3.6 (192.168.100.2)
```

```
    Origin IGP, localpref 100, valid, external, best
```

```
1200 1100
```

```
  192.168.11.12 (metric 11) from 192.168.7.9 (192.168.11.9)
```

```
    Origin IGP, localpref 100, valid, internal
```

Маршруты с кодом r>:

```
dyn6# show ip bgp rib-failure
```

Network	Next Hop	RIB-failure	RIB-NH	Matches
192.168.9.0	192.168.7.9	Higher	admin distance	n/a
192.168.11.0	192.168.7.9	Higher	admin distance	n/a
192.168.16.0	192.168.7.9	Higher	admin distance	n/a

# Таблица маршрутизации

```
dyn6#sh ip route
```

```
...
```

```
B 192.168.12.0/24 [200/0] via 192.168.11.12, 04:41:31
B 192.168.13.0/24 [200/0] via 192.168.11.12, 04:41:31
B 192.168.14.0/24 [200/0] via 192.168.11.12, 04:41:31
O 192.168.15.0/24 [110/21] via 192.168.7.9, 04:42:28, FastEthernet2/0
B 192.168.8.0/24 [20/0] via 192.168.3.6, 04:40:46
O 192.168.9.0/24 [110/11] via 192.168.7.9, 04:42:28, FastEthernet2/0
B 192.168.10.0/24 [20/0] via 192.168.3.6, 04:40:46
O 192.168.11.0/24 [110/11] via 192.168.7.9, 04:42:28, FastEthernet2/0
10.0.0.0/24 is subnetted, 1 subnets
B 10.0.1.0 [20/0] via 192.168.3.6, 04:40:46
B 192.168.6.0/24 [20/0] via 192.168.3.6, 04:40:46
C 192.168.7.0/24 is directly connected, FastEthernet2/0
O 192.168.16.0/24 [110/21] via 192.168.7.9, 04:42:28, FastEthernet2/0
B 192.168.1.0/24 [20/0] via 192.168.3.6, 04:40:47
192.168.100.0/32 is subnetted, 1 subnets
S 192.168.100.1 [1/0] via 192.168.3.2
192.168.3.0/24 is variably subnetted, 3 subnets, 2 masks
C 192.168.3.0/30 is directly connected, FastEthernet1/0
S 192.168.3.0/24 is directly connected, Null0
C 192.168.3.4/30 is directly connected, FastEthernet0/0
```

# Маршруты, которые анонсируются соседу BGP

```
dyn6#sh ip bgp neighbor 192.168.3.6 advertised-routes
```

```
BGP table version is 20, local router ID is 192.168.7.6
```

```
Status codes: s suppressed, d damped, h history, * valid, > best,  
i - internal, r RIB-failure, S Stale
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
Originating default network 0.0.0.0
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 192.168.3.0	0.0.0.0	0		32768	i
*> 192.168.7.0	0.0.0.0	0		32768	i
r>i192.168.9.0	192.168.7.9	0	100	0	i
r>i192.168.11.0	192.168.7.9	0	100	0	i
*>i192.168.12.0	192.168.11.12		100	0	1200 1100 i
*>i192.168.13.0	192.168.11.12	0	100	0	1200 i
*>i192.168.14.0	192.168.11.12	0	100	0	1200 i
r>i192.168.16.0	192.168.7.9	0	100	0	i

```
Total number of prefixes 8
```

# Маршруты, полученные от соседа BGP

```
dyn6#sh ip bgp neighbor 192.168.7.9 routes
```

```
BGP table version is 20, local router ID is 192.168.7.6
```

```
Status codes: s suppressed, d damped, h history, * valid, > best,  
i - internal, r RIB-failure, S Stale
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
* i192.168.7.0	192.168.7.9	0	100	0	i
r>i192.168.9.0	192.168.7.9	0	100	0	i
* i192.168.10.0	192.168.11.12		100	0	1200 1100 i
r>i192.168.11.0	192.168.7.9	0	100	0	i
*>i192.168.12.0	192.168.11.12		100	0	1200 1100 i
*>i192.168.13.0	192.168.11.12	0	100	0	1200 i
*>i192.168.14.0	192.168.11.12	0	100	0	1200 i
r>i192.168.16.0	192.168.7.9	0	100	0	i

```
Total number of prefixes 8
```

# Фильтрация таблицы BGP

```
dyn6#sh ip bgp 192.168.0.0/16
```

```
% Network not in table
```

```
dyn6# sh ip bgp 192.168.0.0/16 longer-prefixes
```

```
BGP table version is 20, local router ID is 192.168.7.6
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i -  
internal,
```

```
                  r RIB-failure, S Stale
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```

Network	Next Hop	Metric	LocPrf	Weight	Path
*> 192.168.1.0	192.168.3.6			0 100 500	i
*> 192.168.3.0	0.0.0.0	0		32768	i
*> 192.168.6.0	192.168.3.6			0 100 500	i
* i192.168.7.0	192.168.7.9	0	100	0	i
*>	0.0.0.0	0		32768	i
*> 192.168.8.0	192.168.3.6			0 100 500	i
r>i192.168.9.0	192.168.7.9	0	100	0	i
*> 192.168.10.0	192.168.3.6			0 100 500	i
* i	192.168.11.12		100	0 1200 1100	i
r>i192.168.11.0	192.168.7.9	0	100	0	i
* 192.168.12.0	192.168.3.6			0 100 500 1100	i
*>i	192.168.11.12		100	0 1200 1100	i
*>i192.168.13.0	192.168.11.12	0	100	0 1200	i
* 192.168.14.0	192.168.3.6			0 100 500 1100	i
*>i	192.168.11.12	0	100	0 1200	i
r>i192.168.16.0	192.168.7.9	0	100	0	i

# Фильтрация вывода по AS-path

Показать маршруты, AS path которых совпадают с регулярным выражением:

```
show ip bgp regexp <regexp>
```

Маршруты проходящие через автономную систему 67:

```
dyn1# sh ip bgp regexp _67_
```

```
BGP table version is 9, local router ID is 197.1.8.1
```

```
Status codes: s suppressed, d damped, h history, * valid, >
```

```
best, i - internal, r RIB-failure, S Stale
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	100.67.67.0/24	192.168.20.8		0	8	67 i
*>	100.78.78.0/24	192.168.20.8		0	8	67 i
*>	100.100.6.0/24	192.168.20.8		0	8	67 i

# Управление маршрутами BGP

# Управление маршрутами BGP

Управлять маршрутами можно на основании:

- Пути через автономные системы (AS path)
- Префикса (сети и маски)
- Значения community

Действия, которые можно выполнять:

- Передать маршрут
- Без изменений
- Поменять значения атрибутов
- Отбросить маршрут

Доступные механизмы:

- AS-path access-list
- Prefix-list
- Route-map

# Фильтрация маршрутов BGP

# Фильтрация маршрутов BGP

AS-path access-list

Фильтрация по пути через автономные системы

Prefix-list

Фильтрация по префиксам

Route-map

Возможность группировать as-path access-list и prefix-list

# **AS-path access-list**

**Фильтрация маршрутов BGP**

# AS-path access-list

## Синтаксис:

```
ip as-path access-list 1 <permit|deny> regexp
```

```
router bgp 100
  network 192.168.100.0 mask 255.255.252.0
  neighbor 192.168.1.1 filter-list 1 out
  neighbor 192.168.1.1 filter-list 10 in
```

```
ip as-path access-list 1 permit ^$
ip as-path access-list 10 permit ^500$
```

```
show ip as-path-access-list
AS path access list 1
  permit ^$
AS path access list 10
  permit ^500$
```

# Регулярные выражения

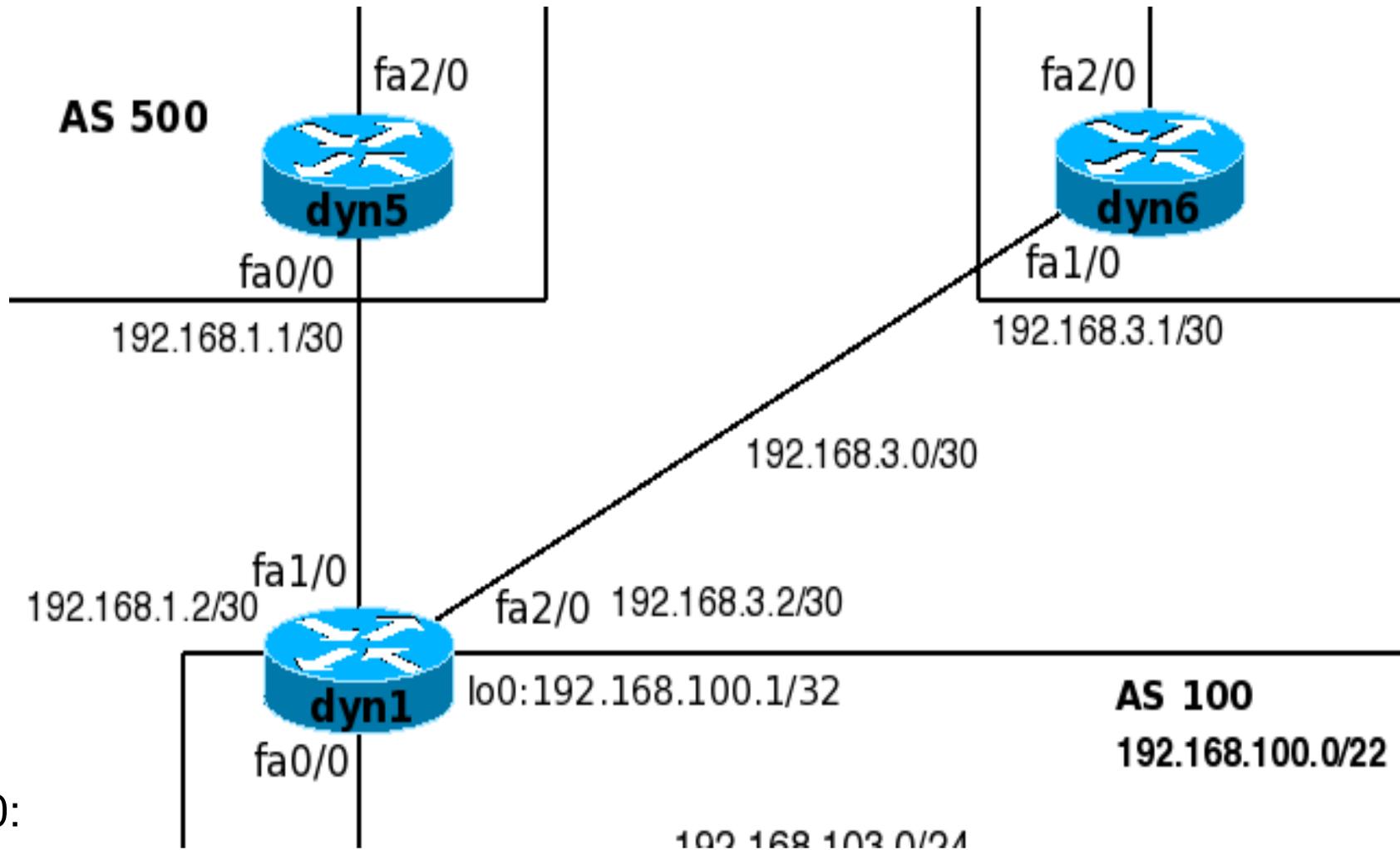
Символы, которые используются в регулярных выражениях:

- . любой символ, включая пробел
- \* ноль или больше совпадений с выражением
- + одно или больше совпадений с выражением
- ? ноль или одно совпадение с выражением
- ^ начало строки
- \$ конец строки
- \_ любой разделитель (включая, начало, конец, пробел)
- \ не воспринимать следующий символ как специальный
- [ ] совпадение с одним из символов в диапазоне
- | логическое или

# Примеры регулярных выражений

- \_67\_ маршруты проходящие через AS 67
- ^67\$ маршруты из соседней AS 67
- \_67\$ маршруты отправленные из AS 67
- ^67\_ сети находящиеся за AS 67
- ^\$ маршруты локальной AS
- .\* любая строка

# Фильтрация исходящих анонсов



dyn1 из AS 100:

```
ip as-path access-list 100 permit ^$
!
router bgp 100
 neighbor 192.168.1.5 filter-list 100 out
 neighbor 192.168.3.6 filter-list 100 out
```

**Prefix-list**

**Фильтрация маршрутов BGP**

# Проверка prefix-list

```
Dyn10# sh ip prefix-list
ip prefix-list LIST_IN: 2 entries
  seq 5 deny 192.168.10.0/24
  seq 10 permit 0.0.0.0/0 le 32
ip prefix-list LIST_OUT: 2 entries
  seq 5 permit 192.168.100.0/22
  seq 10 deny 0.0.0.0/0 le 32
```

```
dyn10#sh ip prefix-list detail
Prefix-list with the last deletion/insertion: dyn5_out
ip prefix-list LIST_IN:
  count: 2, range entries: 1, sequences: 5 - 10, refcount: 2
  seq 5 deny 192.168.10.0/24 (hit count: 2, refcount: 1)
  seq 10 permit 0.0.0.0/0 le 32 (hit count: 0, refcount: 1)
ip prefix-list LIST_OUT:
  count: 2, range entries: 1, sequences: 5 - 10, refcount: 4
  seq 5 permit 192.168.100.0/22 (hit count: 0, refcount: 1)
  seq 10 deny 0.0.0.0/0 le 32 (hit count: 0, refcount: 1)
```

Отобразить маршруты, которые совпадают с фильтром 1:

```
dyn1# show ip bgp prefix-list LIST_IN
```

# Prefix-list

Синтаксис:

```
ip prefix-list <list-name> [seq <value>] <deny|permit>  
<network/length> [ge <value>] [le <value>]
```

network/len:	сеть и маска
ge ge-value:	больше чем или равно
le le-value:	меньше чем или равно

10.0.0.0/8	только сеть 10.0.0.0/8,
10.0.0.0/8 le 11	маршруты у которых первый октет 10, и маска от 8 до 11,
10.0.0.0/8 ge 11	маршруты у которых первый октет 10, и маска от 11 до 32,
10.0.0.0/8 ge 11 le 13	маршруты у которых первый октет 10, и маска от 11 до 13.

# Prefix-list

ip prefix-list NetDay permit 10.0.0.0/8

10.0.0.0/8          10.0.0.0/10          10.0.0.0/16          10.0.0.0/24

ip prefix-list NetDay permit 10.0.0.0/8 le 10

10.0.0.0/8          10.0.0.0/10          10.0.0.0/16          10.0.0.0/24

ip prefix-list NetDay permit 10.0.0.0/8 ge 11

10.0.0.0/8          10.0.0.0/10          10.0.0.0/16          10.0.0.0/24

ip prefix-list NetDay permit 10.0.0.0/8 ge 10 le 20

10.0.0.0/8          10.0.0.0/10          10.0.0.0/16          10.0.0.0/24

# Prefix-list

## Синтаксис:

```
ip prefix-list <list-name> [seq <value>] <deny|permit>  
<network/length> [ge <value>] [le <value>]
```

```
router bgp 100  
  network 192.168.100.0 mask 255.255.252.0  
  neighbor 192.168.1.1 prefix-list DYN5_OUT out  
  neighbor 192.168.1.1 prefix-list DYN5_IN in
```

```
ip prefix-list DYN5_IN deny 192.168.14.0/24  
ip prefix-list DYN5_IN permit 0.0.0.0/0 le 32  
ip prefix-list DYN5_OUT permit 192.168.100.0/22  
ip prefix-list DYN5_OUT deny 0.0.0.0/0 le 32
```

**Route-map**

**Фильтрация маршрутов BGP**

# Route-map

```
route-map TEST permit 10
```

```
match ip address prefix-list dyn8 dyn6 dyn8 ИЛИ dyn6
```

```
set local-preference 120
```

```
!
```

```
route-map TEST permit 20
```

```
match ip address prefix-list dyn5
```

```
dyn5 И as-path 100
```

```
match as-path 100
```

```
set local-preference 80
```

```
!
```

```
route-map TEST permit 30
```

```
разрешить остальные
```

# Route-map

```
router bgp 100  
  neighbor 192.168.1.1 remote-as 200  
  neighbor 192.168.1.1 route-map AS_filter in
```

```
route-map AS_filter permit 10  
  match as-path 1
```

```
route-map AS_filter deny 20  
  match as-path 2
```

```
route-map AS_filter permit 50
```

```
ip as-path access-list 1 permit _1100$  
ip as-path access-list 2 permit _1200_
```

# **Управление входящим и исходящим трафиком**

# Атрибуты BGP:

- \* Well-known mandatory:
  - o Autonomous system path
  - o Next-hop
  - o Origin
  
- \* Well-known discretionary:
  - o Local preference
  - o Atomic aggregate
  
- \* Optional transitive:
  - o Aggregator
  - o Communities
  
- \* Optional non-transitive:
  - o Multi-exit discriminator (MED)

# Выбор лучшего пути BGP:

Сначала проверяется:

\* Доступен ли next-hop (Route Resolvability Condition)

1. Максимальное значение weight (локально для маршрутизатора)
2. Максимальное значение local preference (для всей AS).
3. Предпочесть локальный маршрут маршрутизатора (next hop = 0.0.0.0).
4. Кратчайший путь через AS (самый короткий AS\_PATH)
5. Минимальное значение origin code (IGP < EGP < incomplete).
6. Минимальное значение MED (распространяется между AS).
7. Путь eBGP лучше чем путь iBGP.
8. Выбрать путь через ближайшего IGP-соседа.
9. Выбрать самый старый маршрут для eBGP-пути.
10. Выбрать путь через соседа с наименьшим BGP router ID.
11. Выбрать путь через соседа с наименьшим IP-адресом.

# Управление входящим и исходящим трафиком

## Управление исходящим трафиком:

- Атрибут weight (локально на маршрутизаторе)
- Атрибут Local Preference (локально в AS)
- Балансировка трафика
- Распределение трафика

## Управление входящим трафиком:

- AS path prepend
- MED (подключение к одной и той же AS)
- Community (если поддерживает провайдер)
- Анонс разных префиксов через разных ISP

# Load sharing vs Load balancing

**Распределение нагрузки** – возможность распределять трафик (входящий или исходящий) по нескольким маршрутам.

**Балансировка нагрузки** – возможность распределять нагрузку между несколькими маршрутами для трафика передающегося в одну сеть

Load sharing is the ability to distribute outgoing traffic (or influence the flow of incoming traffic) over multiple paths.

Load balancing is the ability to split the load toward the same destination (host or IP prefix) over multiple paths.

# **Изменение атрибутов BGP с помощью route-map**

# Route-map

```
route-map TEST permit 10
```

```
match ip address prefix-list dyn8 dyn6 dyn8 ИЛИ dyn6
```

```
set local-preference 120
```

```
!
```

```
route-map TEST permit 20
```

```
match ip address prefix-list dyn5
```

```
dyn5 И as-path 100
```

```
match as-path 100
```

```
set local-preference 80
```

```
!
```

```
route-map TEST permit 30
```

```
разрешить остальные
```

## Логика работы route-map

```
router bgp 100
  neighbor 192.168.1.1 remote-as 500
  neighbor 192.168.1.1 route-map IN_FILTER in
```

```
route-map IN_FILTER permit 10
  match ip address prefix-list AS_1100
```

```
route-map IN_FILTER deny 20
  match ip address prefix-list AS_1200
```

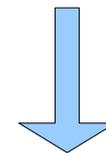
```
route-map IN_FILTER permit 50
```

```
ip prefix-list AS_1100 deny 192.168.14.0/24
ip prefix-list AS_1100 permit 0.0.0.0/0 le 32
ip prefix-list AS_1200 permit 192.168.11.0/22
ip prefix-list AS_1200 deny 0.0.0.0/0 le 32
```

## Логика работы route-map

```
router bgp 100
  neighbor 192.168.1.1 remote-as 500
  neighbor 192.168.1.1 route-map in_filter in
```

```
192.168.10.0/24
192.168.11.0/24
192.168.12.0/24
192.168.14.0/24
192.168.55.0/24
```



```
route-map in_filter permit 10
  match ip address prefix-list AS_1100
    deny 192.168.14.0/24
    permit 192.168.12.0/24
    permit 192.168.10.0/24
```

```
route-map in_filter deny 20
  match ip address prefix-list AS_1200
    permit 192.168.11.0/22
    deny 0.0.0.0/0 le 32
```

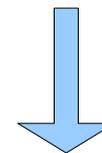
```
route-map in_filter permit 50
```

# Логика работы route-map

```
router bgp 100  
neighbor 192.168.1.1 remote-as 500  
neighbor 192.168.1.1 route-map in_filter in
```

```
route-map in_filter permit 10  
match ip address prefix-list AS_1100  
deny 192.168.14.0/24  
permit 192.168.12.0/24  
permit 192.168.10.0/24
```

192.168.10.0/24



Так как сеть совпала с permit в prefix-list, то это совпадение с Правилем 10 route-map.

```
route-map in_filter deny 20  
match ip address prefix-list AS_1200  
permit 192.168.11.0/22  
deny 0.0.0.0/0 le 32
```

В правиле route-map стоит permit, поэтому маршрут разрешен

```
route-map in_filter permit 50
```

# Логика работы route-map

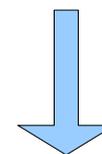
```
router bgp 100
  neighbor 192.168.1.1 remote-as 500
  neighbor 192.168.1.1 route-map in_filter in
```

```
route-map in_filter permit 10
  match ip address prefix-list AS_1100
    deny 192.168.14.0/24
    permit 192.168.12.0/24
    permit 192.168.10.0/24
```

```
route-map in_filter deny 20
  match ip address prefix-list AS_1200
    permit 192.168.11.0/22
    deny 0.0.0.0/0 le 32
```

```
route-map in_filter permit 50
```

192.168.11.0/24



Так как сеть совпала с permit в prefix-list, то это совпадение с Правилom 20 route-map.

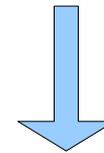
В правиле route-map стоит deny, поэтому маршрут запрещен

# Логика работы route-map

```
router bgp 100
  neighbor 192.168.1.1 remote-as 500
  neighbor 192.168.1.1 route-map in_filter in
```

```
route-map in_filter permit 10
  match ip address prefix-list AS_1100
    deny 192.168.14.0/24
    permit 192.168.12.0/24
    permit 192.168.10.0/24
```

192.168.12.0/24



Так как сеть совпала с permit в prefix-list, то это совпадение с Правилем 10 route-map.

```
route-map in_filter deny 20
  match ip address prefix-list AS_1200
    permit 192.168.11.0/22
    deny 0.0.0.0/0 le 32
```

В правиле route-map стоит permit, поэтому маршрут разрешен

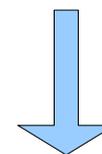
```
route-map in_filter permit 50
```

# Логика работы route-map

```
router bgp 100  
neighbor 192.168.1.1 remote-as 500  
neighbor 192.168.1.1 route-map in_filter in
```

```
route-map in_filter permit 10  
match ip address prefix-list AS_1100  
deny 192.168.14.0/24  
permit 192.168.12.0/24  
permit 192.168.10.0/24
```

192.168.14.0/24



Так как сеть совпала с deny в prefix-list, то это НЕ совпадение с правилом 10 route-map.

```
route-map in_filter deny 20  
match ip address prefix-list AS_1200  
permit 192.168.11.0/22  
deny 0.0.0.0/0 le 32
```

В 20 правиле route-map маршрут попадает на deny. Это НЕ совпадение с 20.

```
route-map in_filter permit 50
```

В правиле 50 route-map стоит permit и нет match, поэтому маршрут совпадает и разрешен

# Логика работы route-map

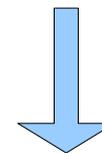
```
router bgp 100  
neighbor 192.168.1.1 remote-as 500  
neighbor 192.168.1.1 route-map in_filter in
```

```
route-map in_filter permit 10  
match ip address prefix-list AS_1100  
deny 192.168.14.0/24  
permit 192.168.12.0/24  
permit 192.168.10.0/24
```

```
route-map in_filter deny 20  
match ip address prefix-list AS_1200  
permit 192.168.11.0/22  
deny 0.0.0.0/0 le 32
```

```
route-map in_filter permit 50
```

192.168.55.0/24

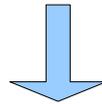


В 20 правиле route-map маршрут попадает на deny. Это НЕ совпадение с 20.

В правиле 50 route-map стоит permit и нет match, поэтому маршрут совпадает и разрешен

# Логика работы route-map

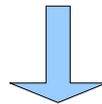
```
192.168.10.0/24  
192.168.11.0/24  
192.168.12.0/24  
192.168.14.0/24  
192.168.55.0/24
```



```
route-map in_filter permit 10  
match ip address prefix-list AS_1100  
    deny 192.168.14.0/24  
    permit 192.168.12.0/24  
    permit 192.168.10.0/24
```

```
route-map in_filter deny 20  
match ip address prefix-list AS_1200  
    permit 192.168.11.0/22  
    deny 0.0.0.0/0 le 32
```

```
route-map in_filter permit 50
```



```
192.168.10.0/24  
192.168.12.0/24  
192.168.14.0/24  
192.168.55.0/24
```

# Критерии match в route-map

Network/mask	match ip address prefix-list
AS-path	match as-path
BGP community	match community
Route originator	match ip route-source
BGP next-hop address	match ip next-hop

# Параметры set в route-map

AS path prepend

set as-path prepend

Weight

set weight

Local preference

set local-preference

BGP community

set community

MED

set metric

Origin

set origin

BGP next-hop

set next-hop

# Управление исходящим трафиком

# Управление исходящим трафиком

Управление исходящим трафиком:

- Атрибут `weight` (локально на маршрутизаторе)
- Атрибут `Local Preference` (локально в AS)
- Балансировка трафика
- Распределение трафика

**Атрибут weight**

# Атрибут weight

- Проприетарный “атрибут” Cisco.
- Локальное значение для маршрутизатора, которое не передается вместе с префиксом.
- Чем больше значение атрибута, тем более предпочтителен путь **выхода**.

Применение ко всем маршрутам полученным от соседа:

```
neighbor 192.168.1.1 weight 500
```

Применение через route-map:

```
route-map set_weight permit 10
```

```
set weight 500
```

```
sh ip bgp route-map set_weight
```

```
sh route-map
```

# Атрибут weight

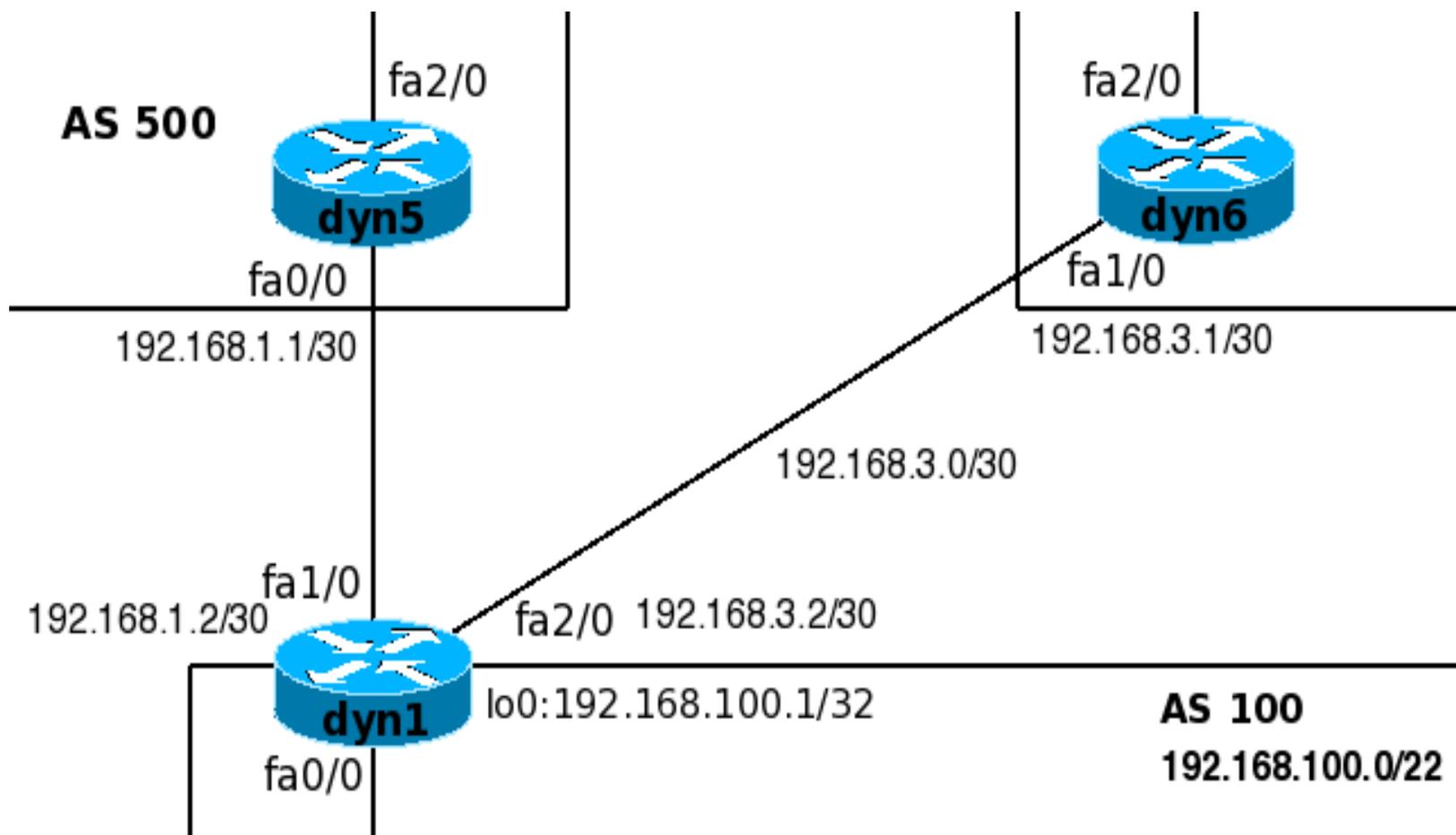
```
route-map set_weight permit 10
```

```
  set weight 500
```

```
!
```

```
router bgp 100
```

```
  neighbor 192.168.1.1 route-map set_weight in
```



# Атрибут Local Preference

# Атрибут Local Preference

- Передается только внутри автономной системы.
- Чем больше значение атрибута, тем более предпочтителен путь **выхода**.
- По умолчанию значение 100

Изменить значение по умолчанию:

```
router bgp 100
  bgp default local-preference <0-4294967295>
```

Применение через route-map:

```
route-map set_LP permit 10
  set local-preference 500
```

# Атрибут Local Preference

```
router bgp 100  
neighbor 192.168.1.1 remote-as 500  
neighbor 192.168.1.1 route-map in_filter in
```

```
route-map in_filter permit 10  
match ip address prefix-list AS_1100  
    deny 192.168.14.0/24  
    permit 192.168.12.0/24  
    permit 192.168.10.0/24  
set local-preference 200
```

```
route-map in_filter deny 20  
match ip address prefix-list AS_1200  
    permit 192.168.11.0/22  
    deny 0.0.0.0/0 le 32
```

```
route-map in_filter permit 50
```

# Балансировка трафика для eBGP

# Load sharing vs Load balancing

**Распределение нагрузки** – возможность распределять трафик (входящий или исходящий) по нескольким маршрутам.

**Балансировка нагрузки** – возможность распределять нагрузку между несколькими маршрутами для трафика передающегося в одну сеть

Load sharing is the ability to distribute outgoing traffic (or influence the flow of incoming traffic) over multiple paths.

Load balancing is the ability to split the load toward the same destination (host or IP prefix) over multiple paths.

## Балансировка исходящего трафика с помощью BGP multipath

BGP Multipath позволяет использовать в таблице маршрутизации несколько маршрутов BGP к одному и тому же получателю.

BGP всё равно выбирает один путь как лучший и анонсирует соседям только его.

Для маршрутов должны выполняться такие критерии:

- \* Должны быть одинаковыми атрибутами weight, local preference, AS path (весь атрибут, а не только длина), origin code, MED, метрика IGP.

- \* next hop маршрутизатор для каждого маршрута должен быть разным

## Балансировка исходящего трафика с помощью BGP multipath

Так как BGP Multipath требует совпадения AS path, то балансировка нагрузки может выполняться в том случае, если клиент подключен к одному и тому же провайдеру несколькими линками.

```
router bgp 100
  maximum-paths 2
```

Для того чтобы балансировать нагрузку между различными провайдерами, необходимо использовать скрытую команду IOS:

```
router bgp 100
  bgp bestpath as-path multipath-relax
```

AS path должен быть одинаковой длины, но не обязательно совпадать по перечню AS

# Балансировка исходящего трафика с помощью BGP multipath

Хотя в таблице BGP маршрут выбирается по-прежнему только один, в таблице маршрутизации будут оба:

```
dyn1#sh ip bgp 192.168.14.0
BGP routing table entry for 192.168.14.0/24, version 43
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Multipath: eBGP
Flag: 0x1820
  Not advertised to any peer
  500 1100
    192.168.1.1 from 192.168.1.1 (192.168.6.5)
      Origin IGP, localpref 100, valid, external, multipath
  600 1200
    192.168.3.1 from 192.168.3.1 (192.168.7.6)
      Origin IGP, localpref 100, valid, external, multipath, best
```

:

```
dyn1# sh ip route
B      192.168.14.0/24 [20/0] via 192.168.3.1, 01:23:19
      [20/0] via 192.168.1.1, 01:23:19
```

# Балансировка исходящего трафика с помощью BGP multipath

Подробная информация о маршруте в таблице маршрутизации:

```
dyn1# sh ip route 192.168.14.0
Routing entry for 192.168.14.0/24
  Known via "bgp 100", distance 20, metric 0
  Tag 600, type external
  Last update from 192.168.1.1 00:02:29 ago
  Routing Descriptor Blocks:
  * 192.168.3.1, from 192.168.3.1, 00:03:25 ago
    Route metric is 0, traffic share count is 1
    AS Hops 2
    Route tag 600
  192.168.1.1, from 192.168.1.1, 00:02:29 ago
    Route metric is 0, traffic share count is 1
    AS Hops 2
    Route tag 600
```

## Балансировка исходящего трафика с учетом пропускной способности каналов

BGP Multipath балансирует нагрузку без учета пропускной способности канала.

Функция BGP Link Bandwidth позволяет балансировать нагрузку в соответствии с пропускной способностью.

**EBGP соседи должны быть непосредственно соединены для того чтобы использовать эту возможность!**

```
router bgp 100
  bgp dmzlink-bw
  neighbor 192.168.1.1 dmzlink-bw
  neighbor 192.168.3.1 dmzlink-bw
```

# Балансировка исходящего трафика с учетом пропускной способности каналов

```
dyn1#sh ip bgp 0.0.0.0
BGP routing table entry for 0.0.0.0/0, version 123
Paths: (2 available, best #2, table Default-IP-Routing-Table)
Multipath: eBGP
Flag: 0x800
  Advertised to update-groups:
    1
  600
    192.168.3.1 from 192.168.3.1 (192.168.7.6)
      Origin IGP, metric 0, localpref 100, valid, external,
multipath
      DMZ-Link Bw 250 kbytes
  500
    192.168.1.1 from 192.168.1.1 (192.168.6.5)
      Origin IGP, metric 0, localpref 100, valid, external,
multipath, best
      DMZ-Link Bw 1000 kbytes
```

# Балансировка исходящего трафика с учетом пропускной способности каналов

```
dyn1#sh ip route 0.0.0.0
Routing entry for 0.0.0.0/0, supernet
  Known via "bgp 100", distance 20, metric 0, candidate
default path
  Tag 500, type external
  Last update from 192.168.1.1 00:01:02 ago
Routing Descriptor Blocks:
  192.168.3.1, from 192.168.3.1, 00:01:02 ago
    Route metric is 0, traffic share count is 1
    AS Hops 1
    Route tag 500
* 192.168.1.1, from 192.168.1.1, 00:01:02 ago
  Route metric is 0, traffic share count is 4
  AS Hops 1
  Route tag 500
```

# Балансировка трафика для iBGP

# Балансировка исходящего трафика с учетом пропускной способности для IBGP

dyn1:

```
router bgp 100
  bgp dmzlink-bw
  neighbor 192.168.1.1 dmzlink-bw
  neighbor 192.168.3.1 dmzlink-bw
  neighbor 192.168.100.2 send-community both
  neighbor 192.168.100.3 send-community both
  neighbor 192.168.100.4 send-community both
  maximum-paths 4
```

dyn2:

```
router bgp 100
  bgp dmzlink-bw
  neighbor 192.168.1.5 dmzlink-bw
  neighbor 192.168.3.5 dmzlink-bw
  neighbor 192.168.100.1 send-community both
  neighbor 192.168.100.3 send-community both
  neighbor 192.168.100.4 send-community both
  maximum-paths 4
```

# Балансировка исходящего трафика с учетом пропускной способности для IBGP

dyn3:

```
router bgp 100
  bgp dmzlink-bw
  neighbor 192.168.100.1 send-community both
  neighbor 192.168.100.2 send-community both
  neighbor 192.168.100.4 send-community both
  maximum-paths ibgp 4
```

dyn4:

```
router bgp 100
  bgp dmzlink-bw
  neighbor 192.168.100.1 send-community both
  neighbor 192.168.100.2 send-community both
  neighbor 192.168.100.3 send-community both
  maximum-paths ibgp 4
```

# **Управление входящим трафиком**

# Управление входящим трафиком

Способы управления входящим трафиком:

- AS path prepend
- MED (подключение к одной и той же AS)
- Community (если поддерживает провайдер)
- Анонс разных префиксов через разных ISP

# AS path prepend

# AS path prepend

- “Ухудшение” маршрута засчет удлинения AS path.
- Не всегда дает необходимый результат
- Чем длиннее путь, тем хуже он считается для входа.

Выполняется через route-map:

```
route-map set_prepend permit 10
  set as-path prepend 35 35 35
```

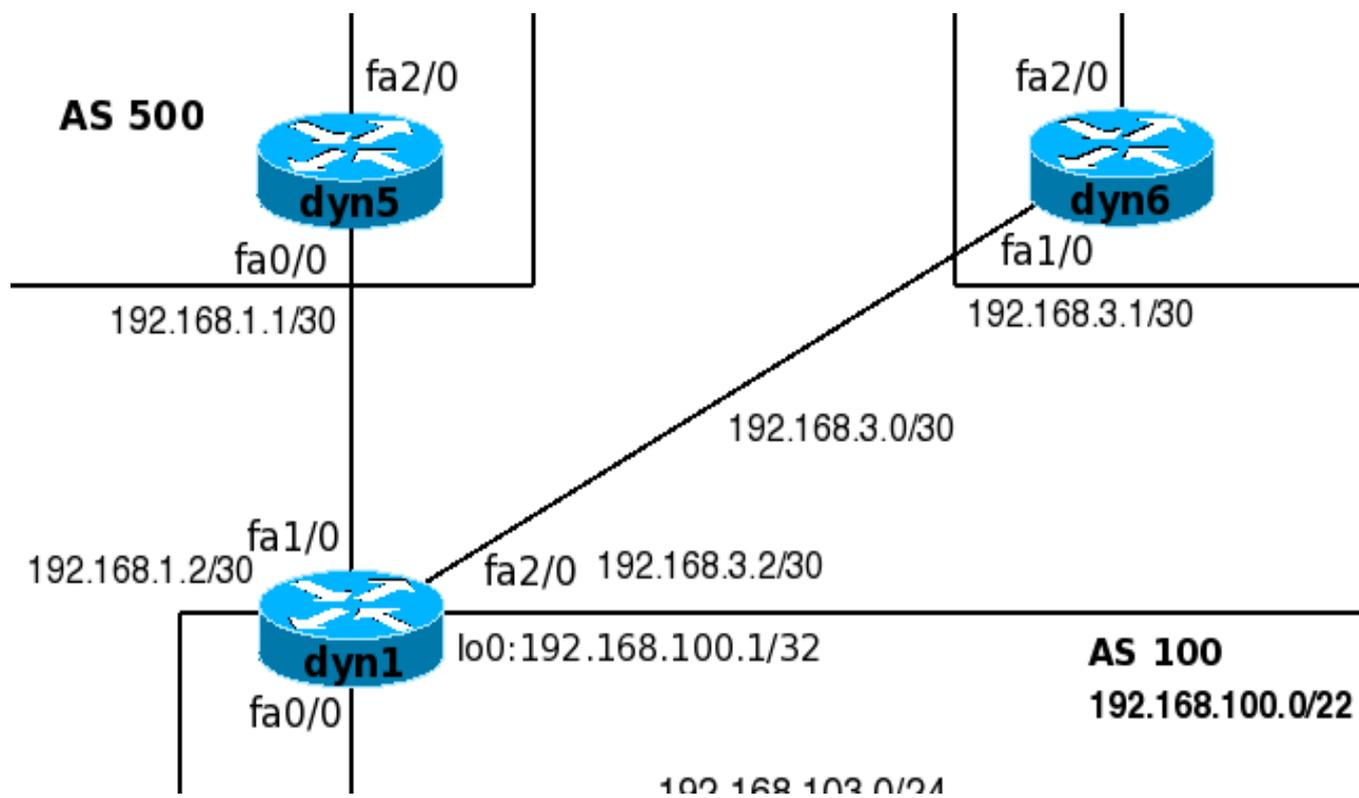
Проверить prepend можно на соседе:

```
sh ip bgp
```

Локально можно проверить настройки route-map:

```
sh route-map
```

# Использование AS path prepend



Настройки AS path prepend на dyn1:

```
route-map local_route permit 10
 match as-path 100
 set as-path prepend 100 100 100
!
router bgp 100
 neighbor 192.168.3.1 route-map local_route out
```

**Атрибут MED**

# Атрибут MED

- Передается через eBGP соседней AS.
- В Cisco называется metric
- Чем меньше значение атрибута, тем более предпочтителен путь входа в локальную AS.
- По умолчанию значение 0

Применение через route-map:

```
route-map set_MED permit 10
  set metric 500
```

# Атрибут MED

```
router bgp 100
  neighbor 192.168.1.1 remote-as 500
  neighbor 192.168.1.1 route-map MED_out out
```

```
route-map MED_out permit 10
  match ip address prefix-list 12
      permit 192.168.101.0/24
      permit 192.168.102.0/24

  set metric 200
```

```
route-map MED_out permit 20
  match ip address prefix-list 100
      permit 192.168.100.0/22

  set metric 100
```

**Community**

# Атрибут Community

- Передается соседней AS для управления входящим трафиком (одно из применений).
- Тегирование маршрутов
- Существуют predetermined значения
- По умолчанию не пересылаются соседям

Применение через route-map:

```
route-map set_C permit 10
  set community 500:100
```

Использование community как критерия через route-map:

```
route-map set_C permit 10
  match community LP_200
  set local-preference 200
```

# Атрибут Community

Предопределенные значения communities:

- no-export (0xFFFFFFFF01) — маршруты не анонсируются EBGP-соседям, но анонсируются внешним соседям в конфедерации,
- no-advertise (0xFFFFFFFF02) — маршруты не должны анонсироваться другим BGP-соседям,
- no-export-subconfed (0xFFFFFFFF03) — маршруты не должны анонсироваться внешним BGP-соседям (ни внешним в конфедерации, ни настоящим внешним соседям). В Cisco это значение встречается и под названием local-as.

# Пример политики ISP с использованием community

## Time Warner Telecom AS4323 BGP Community String

### Local Preference

<b>BGP Community String</b>	<b>Description</b>
<b>4323:80</b>	<b>Set Local Preference in AS4323 to 80</b>
<b>4323:100</b>	<b>Set Local Preference in AS4323 to 100</b>
<b>4323:120</b>	<b>Set Local Preference in AS4323 to 120</b>
<b>4323:187</b>	<b>Blackhole BGP Community -- Used only for /32 hosts</b>
<b>4323:555</b>	<b>Advertise to External Peers and Backbone Customers</b>
<b>4323:666</b>	<b>Keep with in TWTC backbone</b>

### AS prepend

<b>BGP Community String</b>	<b>Description</b>
4323:1	Prepend AS4323 once
4323:2	Prepend AS4323 twice
4323:3	Prepend AS4323 three times

# Настройка community со стороны клиента

dyn1:

```
ip bgp-community new-format
!
route-map LP_200 permit 10
  match as-path 100
  set community 600:200
!
route-map LP_500 permit 10
  match as-path 100
  set community 500:500
!
router bgp 100
  neighbor 192.168.1.1 send-community
  neighbor 192.168.1.1 route-map LP_500 out
  neighbor 192.168.3.1 send-community
  neighbor 192.168.3.1 route-map LP_200 out
```

# Настройка community

## Стандартный community-list:

```
ip community-list <1-99> <permit | deny> <value>
```

## Пример:

```
ip community-list 1 deny 100:37  
ip community-list 1 permit internet
```

## Расширенный community-list:

```
ip community-list <100-199> <permit | deny> <regex>
```

## Именованный community-list:

```
ip community-list <standard|expanded> <name> <permit |  
deny> <value|regex>
```

# Настройка community со стороны провайдера

dyn5:

```
ip bgp-community new-format
!
ip community-list standard LP_200 permit 500:200
ip community-list standard LP_500 permit 500:500
!
route-map client_LP_in permit 10
  match community LP_200
  set local-preference 200
route-map client_LP_in permit 20
  match community LP_500
  set local-preference 500
route-map client_LP_in permit 30
  set local-preference 150
!
router bgp 500
  neighbor 192.168.1.2 route-map client_LP_in in
  neighbor 192.168.1.6 route-map client_LP_in in
```

# Настройка community со стороны провайдера

```
dyn5#sh ip bgp 192.168.100.0
BGP routing table entry for 192.168.100.0/22, version 720
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to update-groups:
    1      3
100
  192.168.1.2 from 192.168.1.2 (192.168.100.1)
    Origin IGP, metric 0, localpref 500, valid, external, best
    Community: 500:500
100
  192.168.1.6 from 192.168.1.6 (192.168.100.2)
    Origin IGP, metric 0, localpref 200, valid, external
    Community: 500:200
```

**Анонс разных префиксов  
через разных ISP**

# Анонс разных префиксов через разных ISP

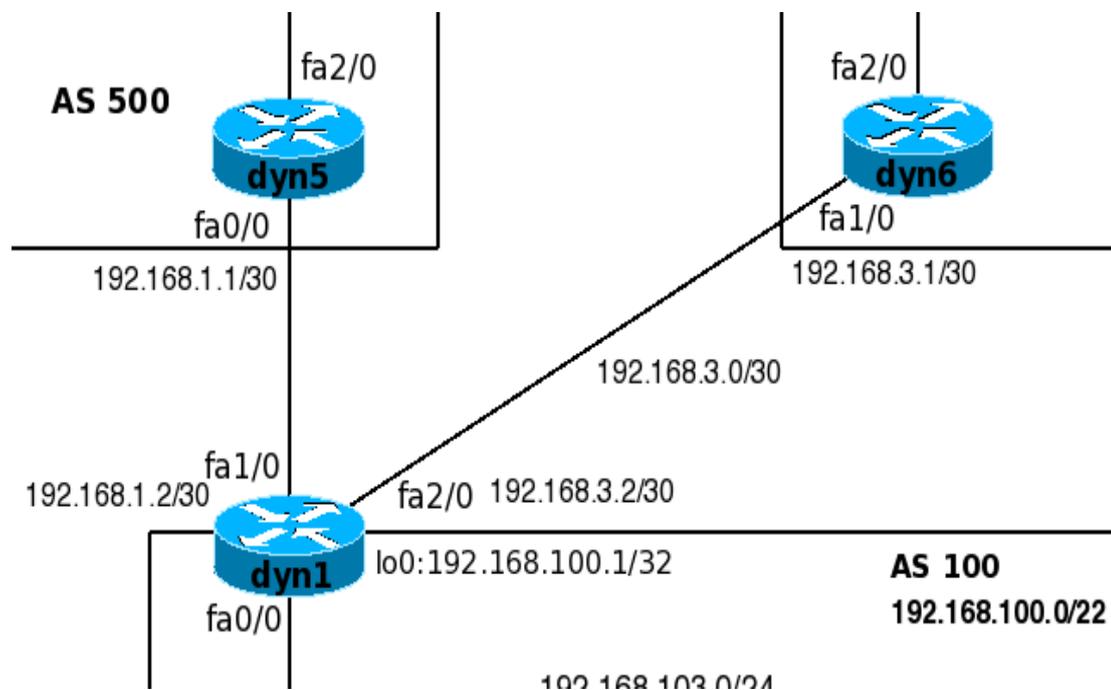
- Блок адресов можно разбить на подсети
  - Подсети с маской не длиннее /24
- Часть подсетей анонсируется через одного провайдера, вторая – через второго
- Суммарный диапазон анонсируется через оба ISP
- Не всегда дает необходимый результат
- Влияет на входящий трафик

# Анонс разных префиксов через разных ISP

```
router bgp 100
 network 192.168.100.0
 network 192.168.101.0
 network 192.168.102.0
 network 192.168.103.0
 network 192.168.100.0 mask 255.255.252.0
 neighbor 192.168.1.1 prefix-list dyn5 out
 neighbor 192.168.3.1 prefix-list dyn6 out
```

```
ip prefix-list dyn5 permit 192.168.100.0/22
ip prefix-list dyn5 permit 192.168.101.0/24
ip prefix-list dyn5 permit 192.168.102.0/24
```

```
ip prefix-list dyn6 permit 192.168.100.0/22
ip prefix-list dyn6 permit 192.168.100.0/24
ip prefix-list dyn6 permit 192.168.103.0/24
```



# **Применение изменений в настройках политик BGP**

# Hard reset

Hard reset:

```
clear ip bgp *
```

Результат выполнения команды `clear ip bgp *`:

- \* Сброс всех BGP-соединений с этим маршрутизатором
- \* Очищается таблица BGP
- \* Сессии BGP переходят из состояния `established` в состояние `idle`
- \* Вся информация должна быть заново выучена (сосед должен её заново отправить)

Hard reset для соседа:

```
clear ip bgp <neighbor-address>
```

Результат выполнения команды `clear ip bgp <neighbor>`:

- \* Сброс BGP-соединений только с соседом
- \* Сессия BGP, установленная с этим соседом, переходит из состояния `established` в состояние `idle`
- \* Вся информация от соседа должна быть заново выучена

# Outbound soft reset

Outbound soft reset:

```
clear ip bgp <neighbor-address> out
```

Результат выполнения команды:

- \* Маршруты выученные от указанного соседа не теряются
- \* Версия таблицы (table version number) для соседа выставляется равной 0. При наступлении следующего интервала для отправки обновлений, маршрутизатор проверяет таблицу BGP и отправляет соседу все маршруты, так как у них версия больше чем ноль
- \* Локальный маршрутизатор отправляет заново всю информацию BGP соседу, не разрывая соединения
- \* Соединение не разрывается
- \* Это команда нужна для случаев, когда обновляется исходящая политика
- \* При изменении входящей политики, команда soft out не помогает

# Dynamic inbound soft reset

Dynamic inbound soft reset:

```
clear ip bgp <neighbor-address> in
```

Результат выполнения команды `clear ip bgp <neighbor-address> in`:

- \* Маршруты отправленные соседу не убираются
- \* Соединение не разрывается
- \* Обновления не сохраняются локально
- \* Сосед заново отправляет маршруты
- \* Обновляет маршруты от соседа в таблице маршрутизации

Локально сохранить маршруты полученные от соседа:

```
neighbor <neighbor> soft-reconfiguration inbound
```

```
show ip bgp neighbour <neighbor> received-routes
```

# **Поиск неисправностей (troubleshooting)**

# Команды отладки

```
debug ip tcp transactions
```

```
debug ip bgp events
```

```
debug ip bgp keepalives
```

Информация об обновлениях BGP:

```
debug ip bgp updates
```

Отобразить все входящие и исходящие обновления для маршрутов совпадающих с ACL:

```
debug ip bgp updates <acl>
```

Отобразить все обновления полученные от или отправленные соседу (можно добавлять ACL):

```
debug ip bgp <ip-address> updates [acl]
```

**Object Tracking**

**IP SLA**

**Policy based routing (PBR)**

**Embedded Event Manager (EEM)**

**Object Tracking**

**IP SLA**

**Policy based routing (PBR)**

**Embedded Event Manager (EEM)**

# Object Tracking

# Object Tracking (Enhanced Object Tracking)

Объекты, состояние которых может отслеживаться:

- Состояние Line-Protocol интерфейса
- Доступность маршрута
- Метрика маршрута
- Состояние IP SLA

Условия можно комбинировать

# Object Tracking (Enhanced Object Tracking)

```
ip sla 1
  icmp-echo 192.168.6.8 source-interface
FastEthernet1/0
  timeout 500
  frequency 3
ip sla schedule 1 life forever start-time now

ip route 192.168.6.8 255.255.255.255 192.168.1.5

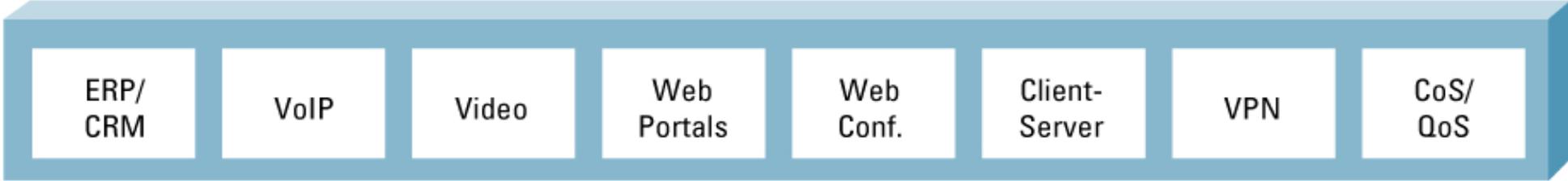
track 1 ip sla 1 reachability

track 1 ip sla 1 reachability
  delay down 10 up 5

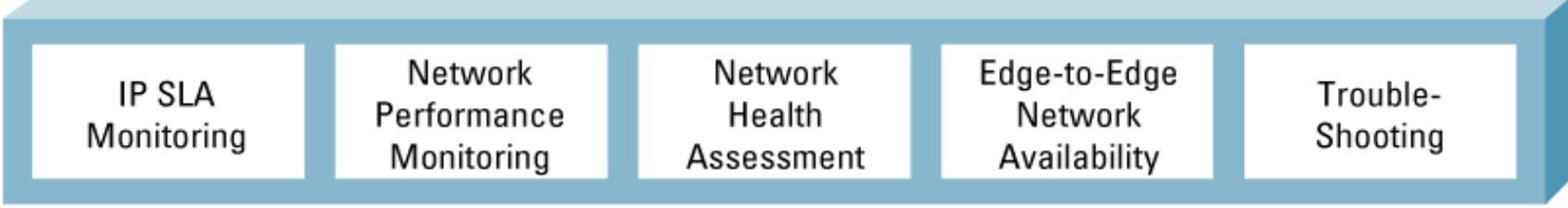
ip route 0.0.0.0 0.0.0.0 192.168.1.5 track 1
ip route 0.0.0.0 0.0.0.0 192.168.2.6 250
```

**IP SLA**

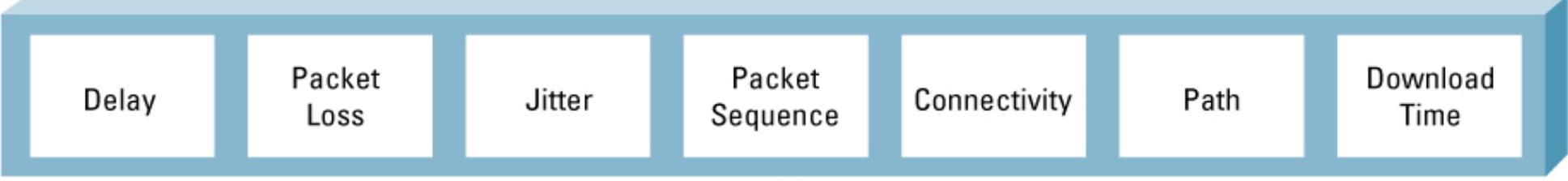
# Applications and Solutions



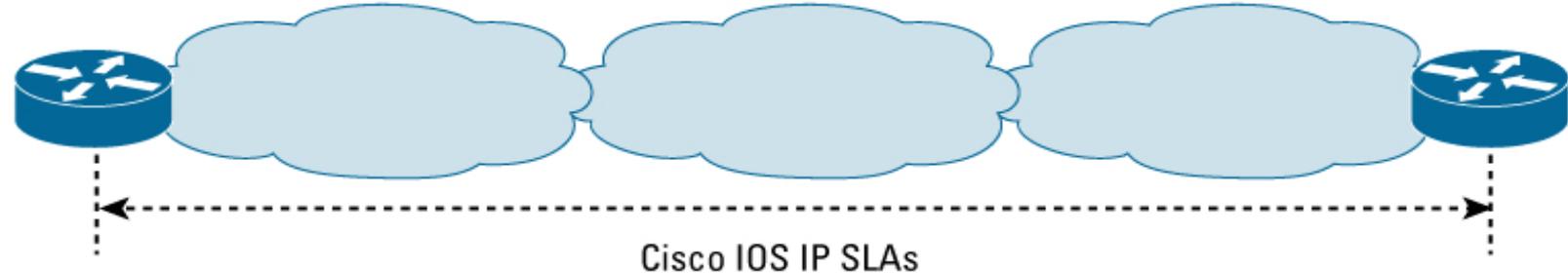
## Cisco IOS SLAs Functions



## Cisco IOS IP SLAs Metrics



## Cisco IOS IP SLAs Operations



# Доступные измерения

```
router(config)#ip sla monitor 1
```

```
router(config-sla-monitor)#type ?
```

dhcp	Perform DHCP Operation
dls	Perform DLSw Keepalive Operation
dns	Perform DNS Query
echo	Perform Point to Point Echo Operations
ftp	Perform ftp operation
http	Perform HTTP Operations
jitter	Perform Jitter Operation
pathEcho	Perform Path Discovered Echo Operations
tcpConnect	Perform TCP Connect Operations
udpEcho	Perform UDP Echo Operations

# Проверка настроек

```
dyn1#sh track 1
```

```
Track 1
```

```
IP SLA 1 reachability
```

```
Reachability is Up
```

```
8 changes, last change 00:23:30
```

```
Delay up 5 secs, down 10 secs
```

```
Latest operation return code: OK
```

```
Latest RTT (milliseconds) 56
```

```
Tracked by:
```

```
STATIC-IP-ROUTING 0
```

```
dyn1# sh ip route track-table
```

```
ip route 0.0.0.0 0.0.0.0 192.168.1.5 track 1 state is [up]
```

```
dyn1#sh ip sla statistics
```

```
IPSLAs Latest Operation Statistics
```

```
IPSLA operation id: 1
```

```
Latest RTT: 144 milliseconds
```

```
Latest operation start time: *08:42:03.787 UTC Fri Dec 10 2010
```

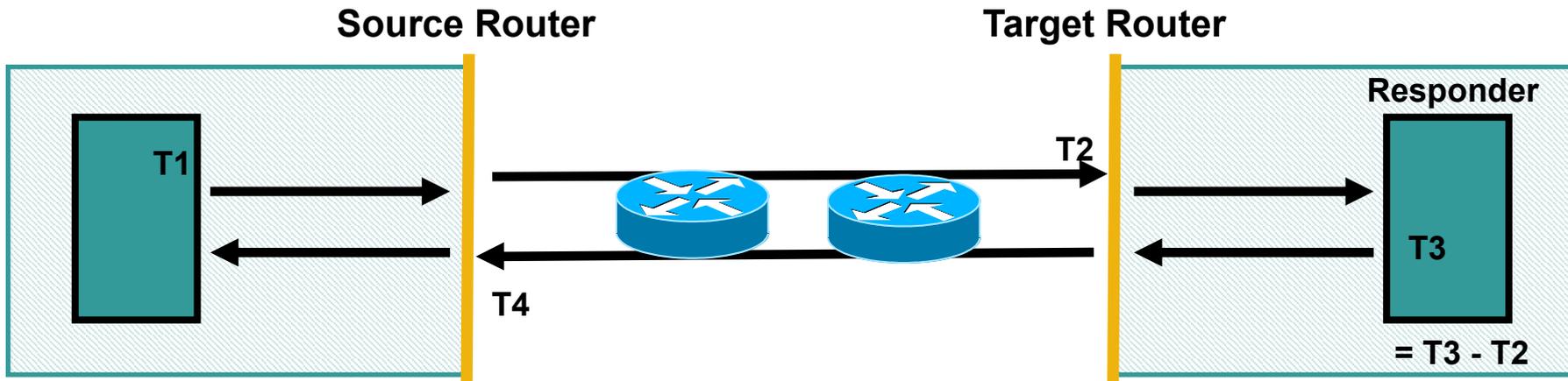
```
Latest operation return code: OK
```

```
Number of successes: 517
```

```
Number of failures: 1
```

```
Operation time to live: Forever
```

# IP SLA responder



Responder делает два замера времени (T2 & T3)

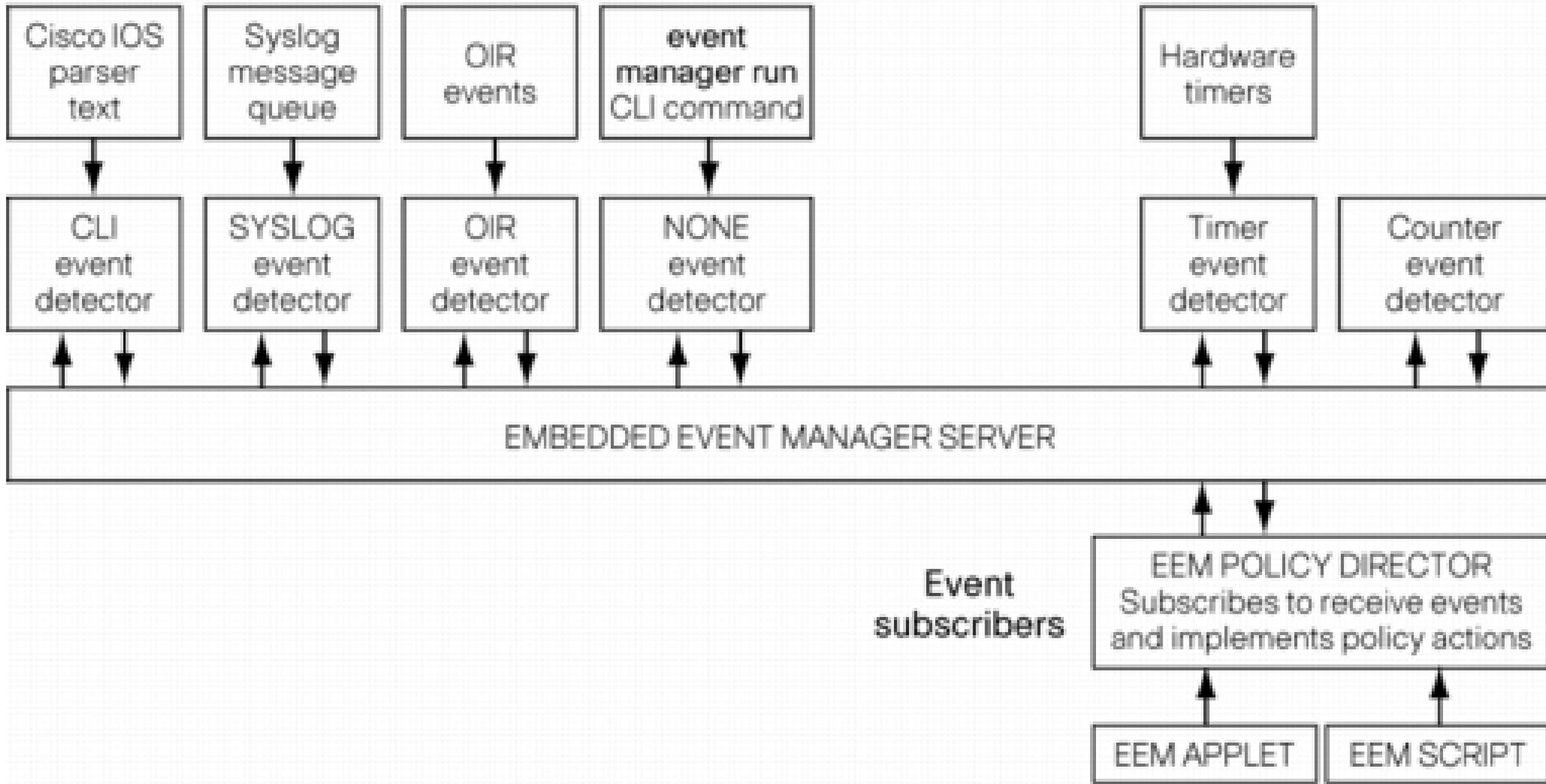
Responder учитывает время на обработку пакетов маршрутизатором

Responder позволяет выполнять односторонние измерения latency, jitter, packet loss

# **Embedded Event Manager (EEM)**

# Embedded Event Manager (EEM)

## Core event publishers



# Настройка правил для автоматической очистки таблицы трансляций

```
event manager applet track_ISP
  event track 1 state any
  action 1.0 cli command "enable"
  action 1.1 cli command "clear ip nat trans *"
  action 1.2 syslog msg "Track state is changed"
```

```
event manager applet ISP_1_UP
  event track 1 state up
  action 1.0 cli command "enable"
  action 1.1 cli command "clear ip nat trans *"
  action 1.2 syslog msg "ISP 1 is UP"
event manager applet ISP_1_DOWN
  event track 1 state down
  action 1.0 cli command "enable"
  action 1.1 cli command "clear ip nat trans *"
  action 1.2 syslog msg "ISP 1 is DOWN"
```

## Пример правила

```
event manager applet TXload
  event interface name FastEthernet1/0 parameter txload
entry-op gt entry-val 80 entry-type value poll-
interval 10
  action 1 track set 10 state up
  action 2 track read 10
event manager applet TXload_low
  event interface name FastEthernet1/0 parameter txload
entry-op lt entry-val 50 entry-type value poll-
interval 10
  action 1 track set 10 state down
  action 2 track read 10

track 10 stub-object
  default-state down
  delay down 10 up 5

ip route 0.0.0.0 0.0.0.0 192.168.3.1 track 10
```

# **Policy-based Routing (PBR)**

# Policy Based Routing

Policy Based Routing – маршрутизация пакетов, основанная на наборе определенных правил.

Предоставляет более гибкий механизм для обработки пакетов на маршрутизаторах, дополняя стандартную маршрутизацию по адресу получателя.

Может быть применена как к сквозному трафику, так и к пакетам сгенерированным самим маршрутизатором

# Policy Based Routing

Policy Based Routing – позволяет управлять маршрутизацией трафика, на основании таких параметров:

- Адрес отправителя (или комбинации адрес отправителя-адрес получателя)
- Информация 7 уровня (приложений) OSI
- Интерфейс, в который пришел пакет
- QoS-метки
- любая информация, используемая в расширенном ACL

Настраивается с помощью **route-map**

# Пример правила

## Создание правила:

```
router(config)#event manager applet int_down
```

## Настройка события на которое ЕЕМ будет реагировать:

```
router(config-applet)# event syslog pattern "Interface  
GigabitEthernet0/0, changed state to down"
```

## Настройка действий, которые будут выполнены, после того как произошло событие:

```
router(config-applet)# action 1.0 cli command "enable"  
router(config-applet)# action 1.1 cli command "configure term"  
router(config-applet)# action 1.2 cli command "interface g0/1"  
router(config-applet)# action 1.3 cli command "no shut"  
router(config-applet)#
```

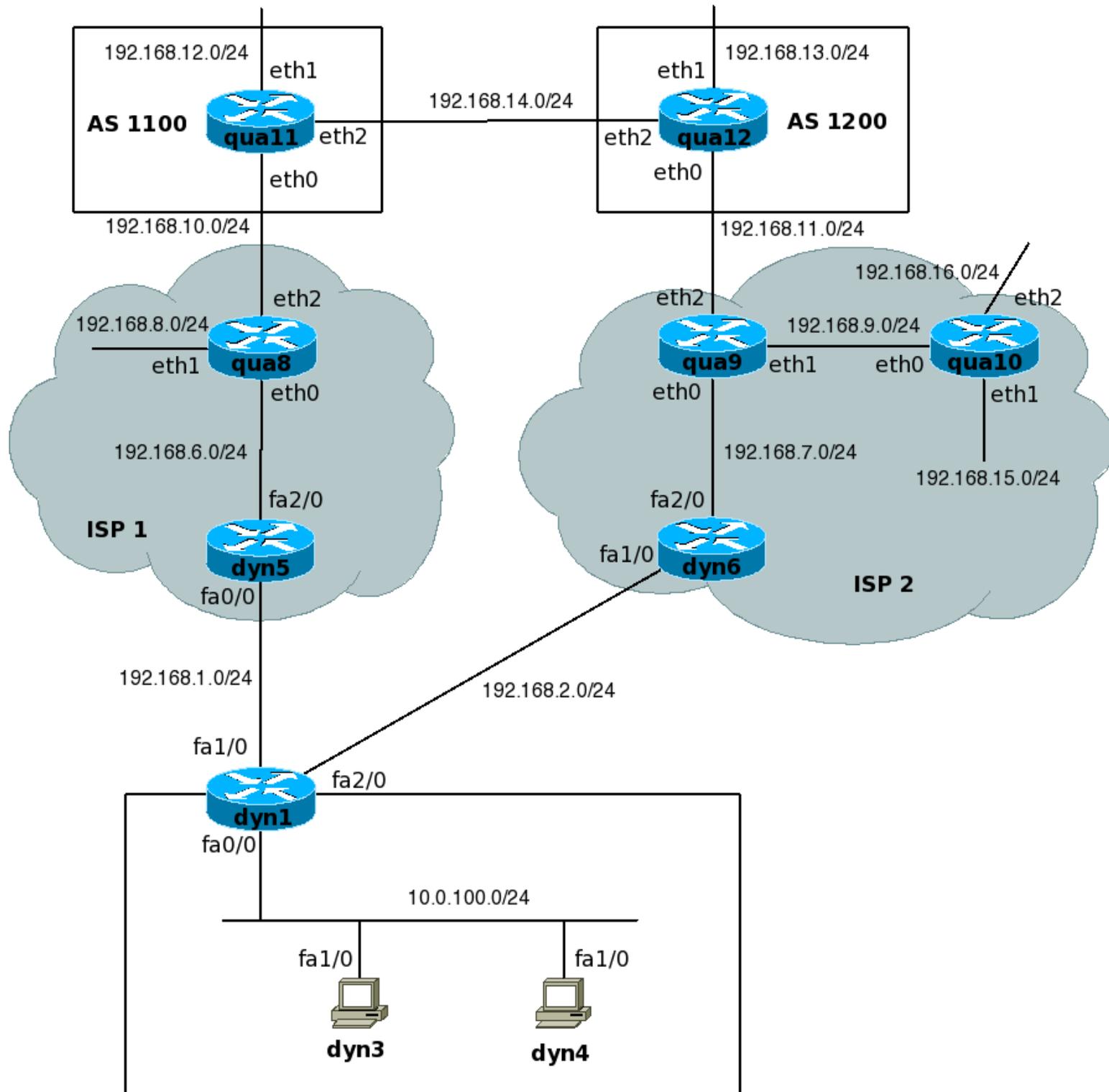
# Настройка правил для автоматической очистки таблицы трансляций

```
event manager applet track_ISP
  event track 1 state any
  action 1.0 cli command "enable"
  action 1.1 cli command "clear ip nat trans *"
  action 1.2 syslog msg "Track state is changed"
```

```
event manager applet ISP_1_UP
  event track 1 state up
  action 1.0 cli command "enable"
  action 1.1 cli command "clear ip nat trans *"
  action 1.2 syslog msg "ISP 1 is UP"
event manager applet ISP_1_DOWN
  event track 1 state down
  action 1.0 cli command "enable"
  action 1.1 cli command "clear ip nat trans *"
  action 1.2 syslog msg "ISP 1 is DOWN"
```

**Пример настройки резервирования  
каналов и распределения нагрузки  
между каналами.  
Без использования BGP**

# Топология сети



# Этапы конфигурации

1. Настройка NAT
2. Настройка IP SLA для мониторинга доступности провайдера
3. Привязка результатов IP SLA через track к статическим маршрутам по умолчанию
4. Настройка балансировки нагрузки между провайдерами.
5. Настройка правил для автоматической очистки таблицы трансляций при переключении на другого провайдера с помощью EEM

# Настройка трансляции адресов

```
interface FastEthernet0/0
  ip address 10.0.100.1 255.255.255.0
  ip nat inside
!
interface FastEthernet1/0
  ip address 192.168.1.1 255.255.255.0
  ip nat outside
!
interface FastEthernet2/0
  ip address 192.168.2.1 255.255.255.0
  ip nat outside
!
ip access-list extended LAN
  permit ip 10.0.100.0 0.0.0.255 any
!
route-map ISP_1 permit 10
  match ip address LAN
  match interface FastEthernet1/0

route-map ISP_2 permit 10
  match ip address LAN
  match interface FastEthernet2/0
!
ip nat inside source route-map ISP_1 interface Fa1/0 overload
ip nat inside source route-map ISP_2 interface Fa2/0 overload
```

# Настройка IP SLA для мониторинга доступности провайдера

```
ip sla 1
  icmp-echo 192.168.6.8 source-interface
FastEthernet1/0
  timeout 500
  frequency 3
ip sla schedule 1 life forever start-time now
!
ip sla 2
  icmp-echo 192.168.7.9 source-interface
FastEthernet2/0
  timeout 500
  frequency 3
ip sla schedule 2 life forever start-time now

ip route 192.168.6.8 255.255.255.255 192.168.1.5
ip route 192.168.7.9 255.255.255.255 192.168.2.6
```

# Привязка результатов IP SLA через track к статическим маршрутам по умолчанию

```
track 1 ip sla 1 reachability  
  delay down 10 up 5
```

```
track 2 ip sla 2 reachability  
  delay down 10 up 5
```

```
ip route 0.0.0.0 0.0.0.0 192.168.1.5 track 1  
ip route 0.0.0.0 0.0.0.0 192.168.2.6 track 2
```

# Распределение трафика с использованием PBR

ACL для выделения нечетных IP-адресов

```
ip access-list standard odd
 permit 10.0.100.1 0.0.0.254
```

ACL для выделения четных IP-адресов

```
ip access-list standard even
 permit 10.0.100.0 0.0.0.254
```

```
route-map POLICY permit 10
```

```
 match ip address odd
```

```
 set ip next-hop verify-availability 192.168.1.5 1 track 1
```

```
route-map POLICY permit 20
```

```
 match ip address even
```

```
 set ip next-hop verify-availability 192.168.2.6 2 track 2
```

```
interface FastEthernet0/0
```

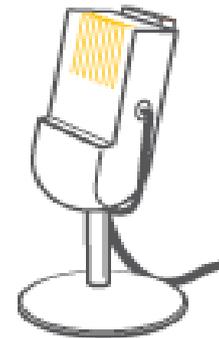
```
 ip policy route-map POLICY
```

# Настройка правил для автоматической очистки таблицы трансляций

```
event manager applet Balagan_UP
  event track 1 state up
  action 1.0 cli command "enable"
  action 1.1 cli command "clear ip nat trans *"
  action 1.2 syslog msg "Balagan is UP"

event manager applet Balagan_DOWN
  event track 1 state down
  action 1.0 cli command "enable"
  action 1.1 cli command "clear ip nat trans *"
  action 1.2 syslog msg "Balagan is DOWN"
```

**Сайт проекта ЛинкМиАп: <http://linkmeup.ru>**



# **Сети для самых маленьких Часть восьмая. BGP и IP SLA**

**Наташа Самойленко <http://xgu.ru/wiki/Участник:Nata>**