

Использование BGP для резервирования Интернет-каналов

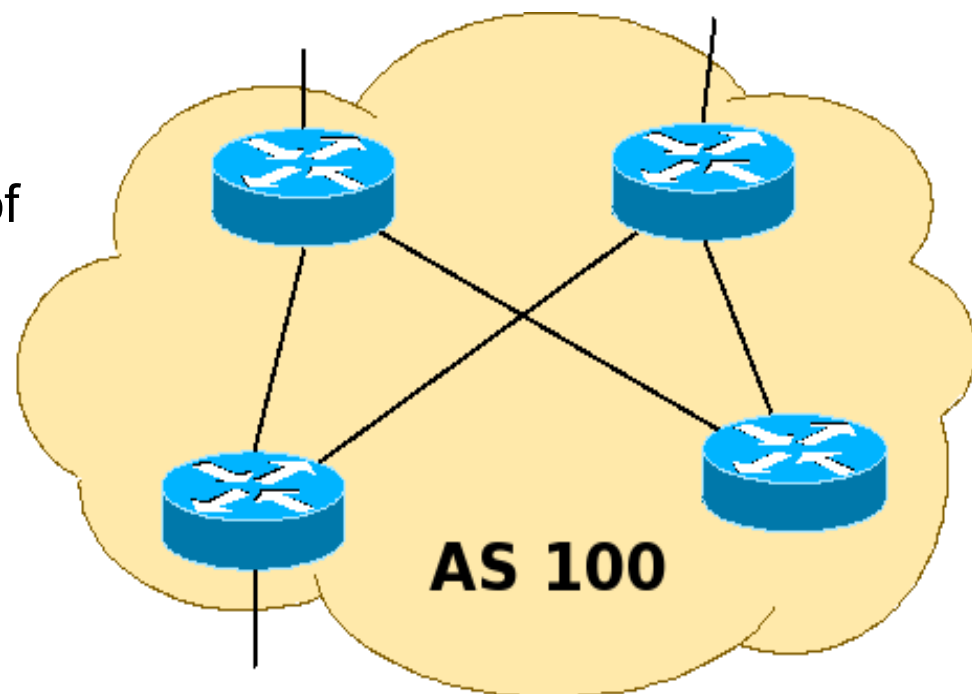
Наташа Самойленко

Сетевые Дни

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)



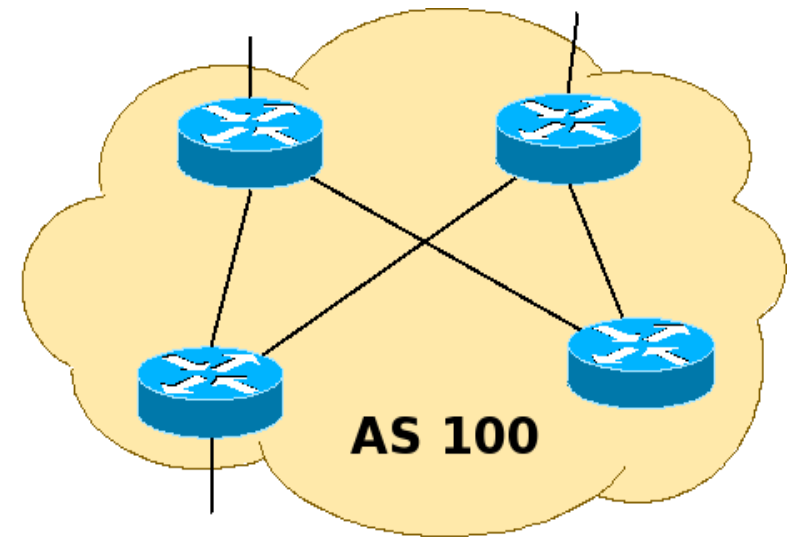
Номер автономной системы (ASN)

Два диапазона:

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

Использование:

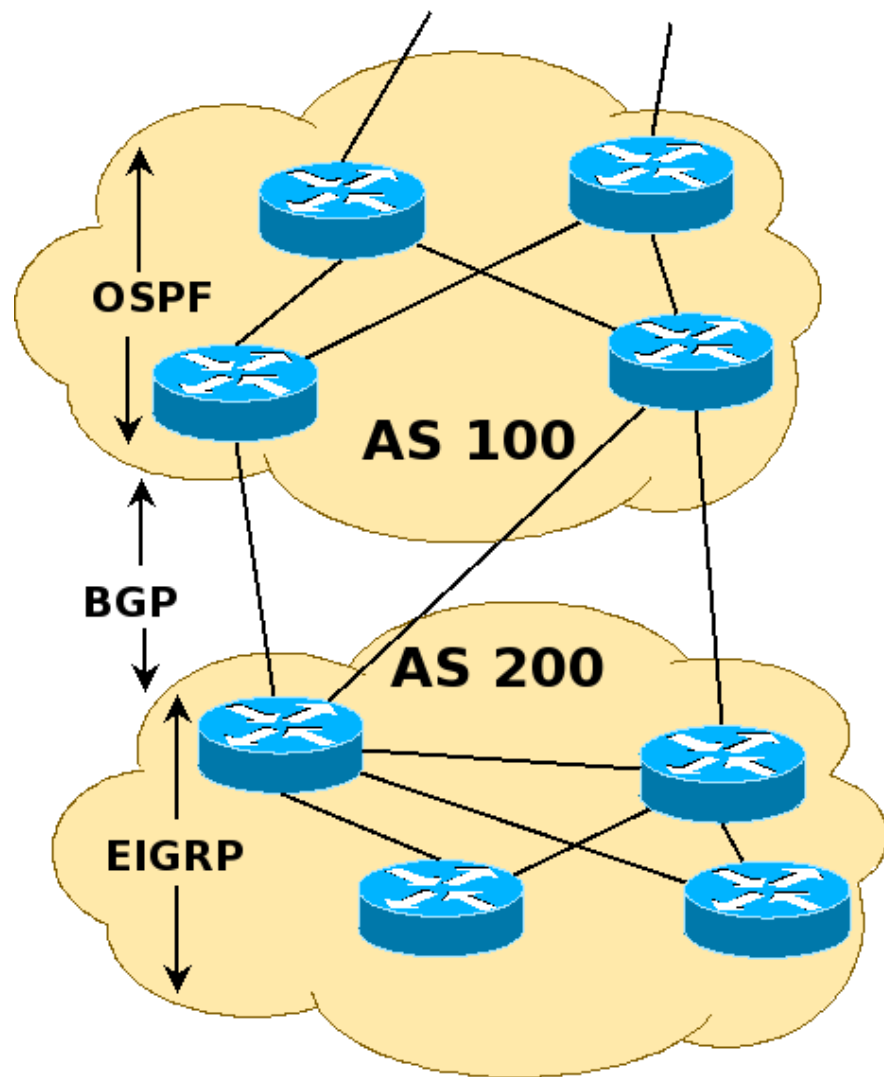
- 0 и 65535 (зарезервированы)
- 1-64495 (публичные номера)
- 65552-4294967295 (публичные номера)
- 64512-65534 (приватные номера)
- 23456 (представляет 32-bit диапазон на устройствах,
- которые работают с 16-bit диапазоном)



Протоколы маршрутизации

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

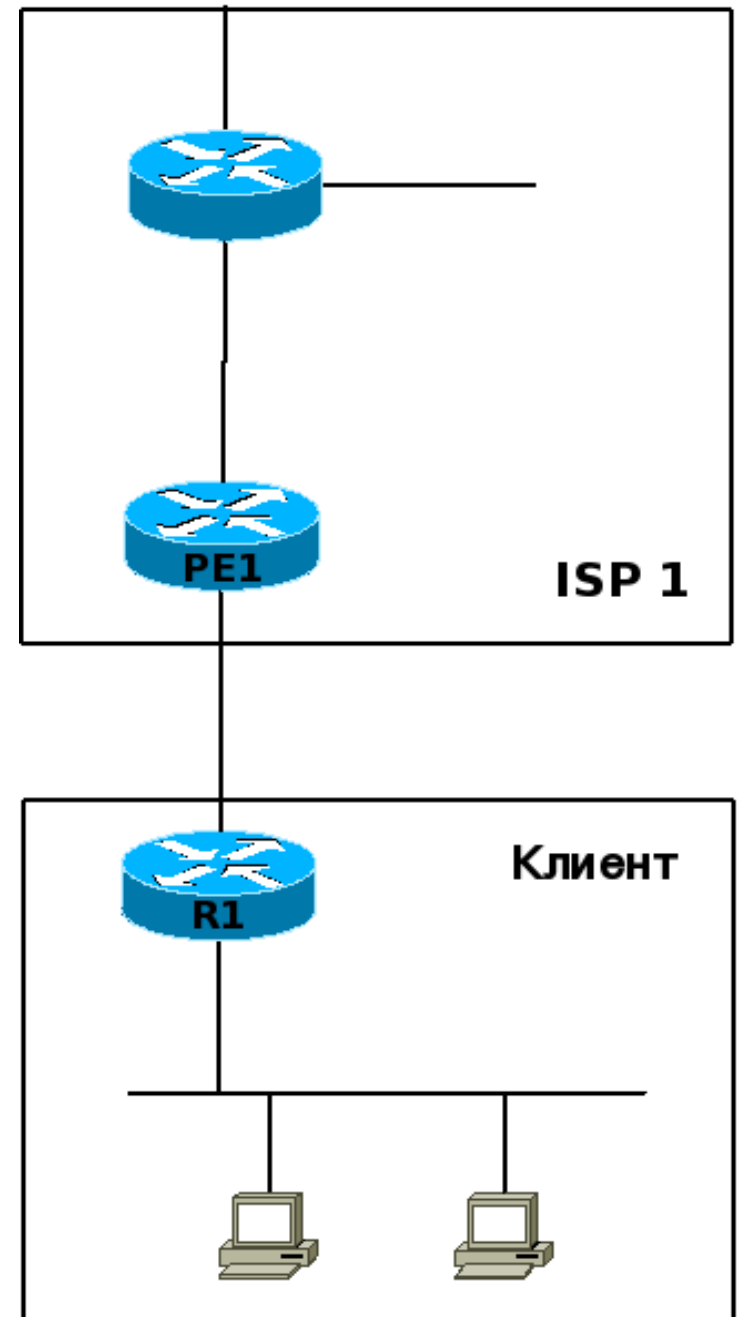
Exterior gateway protocol – протокол, который используется для передачи информации о маршрутах, между автономными системами.



Варианты подключения к ISP

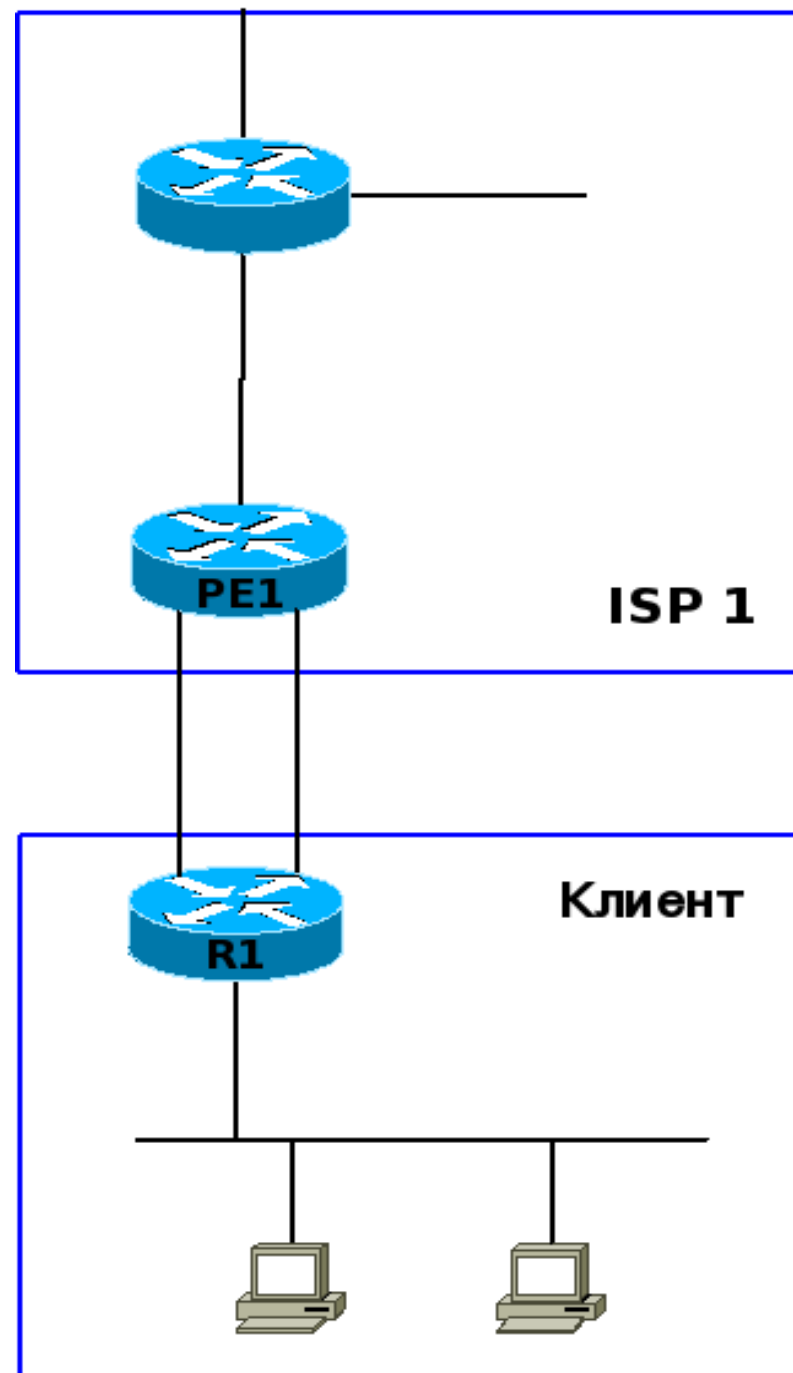
Подключение к одному провайдеру (single-homed customer)

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



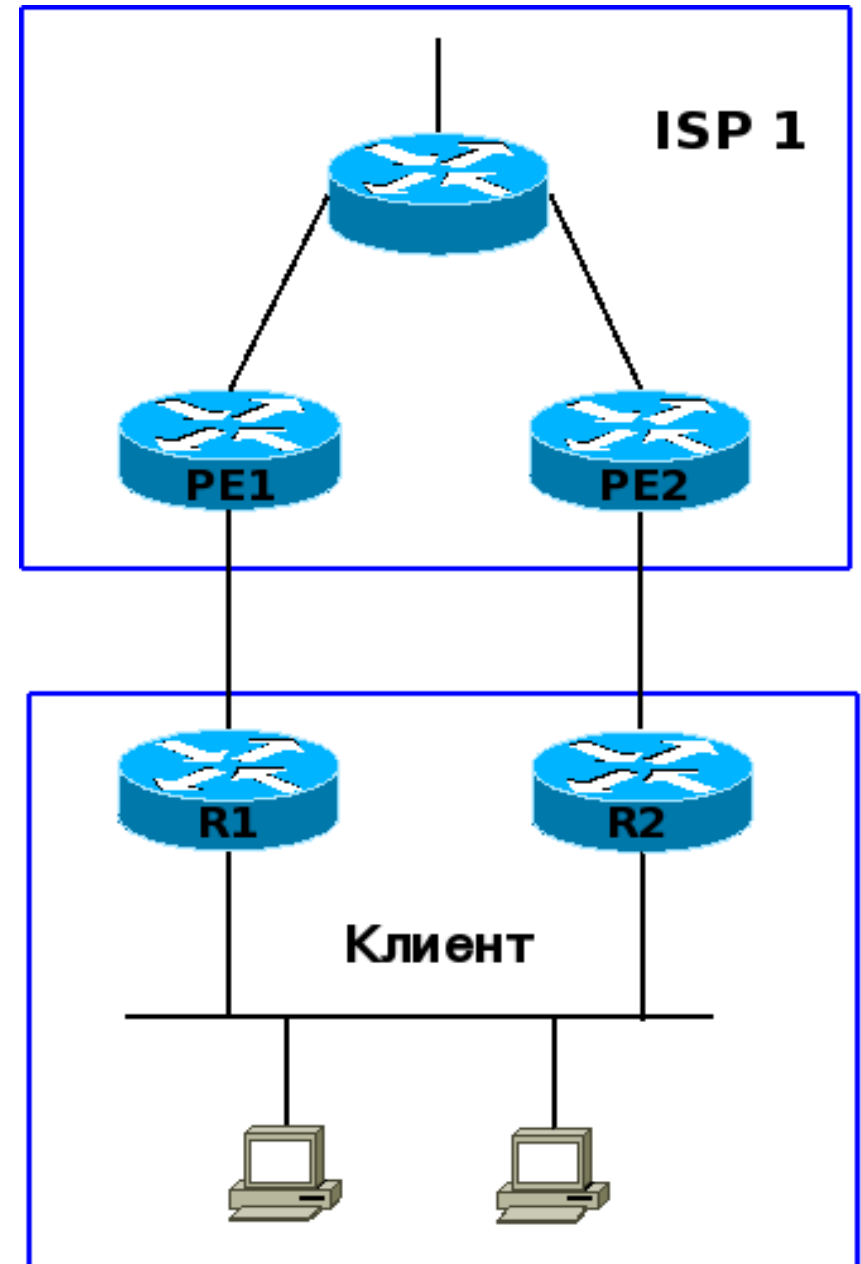
Избыточное подключение к одному провайдеру

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



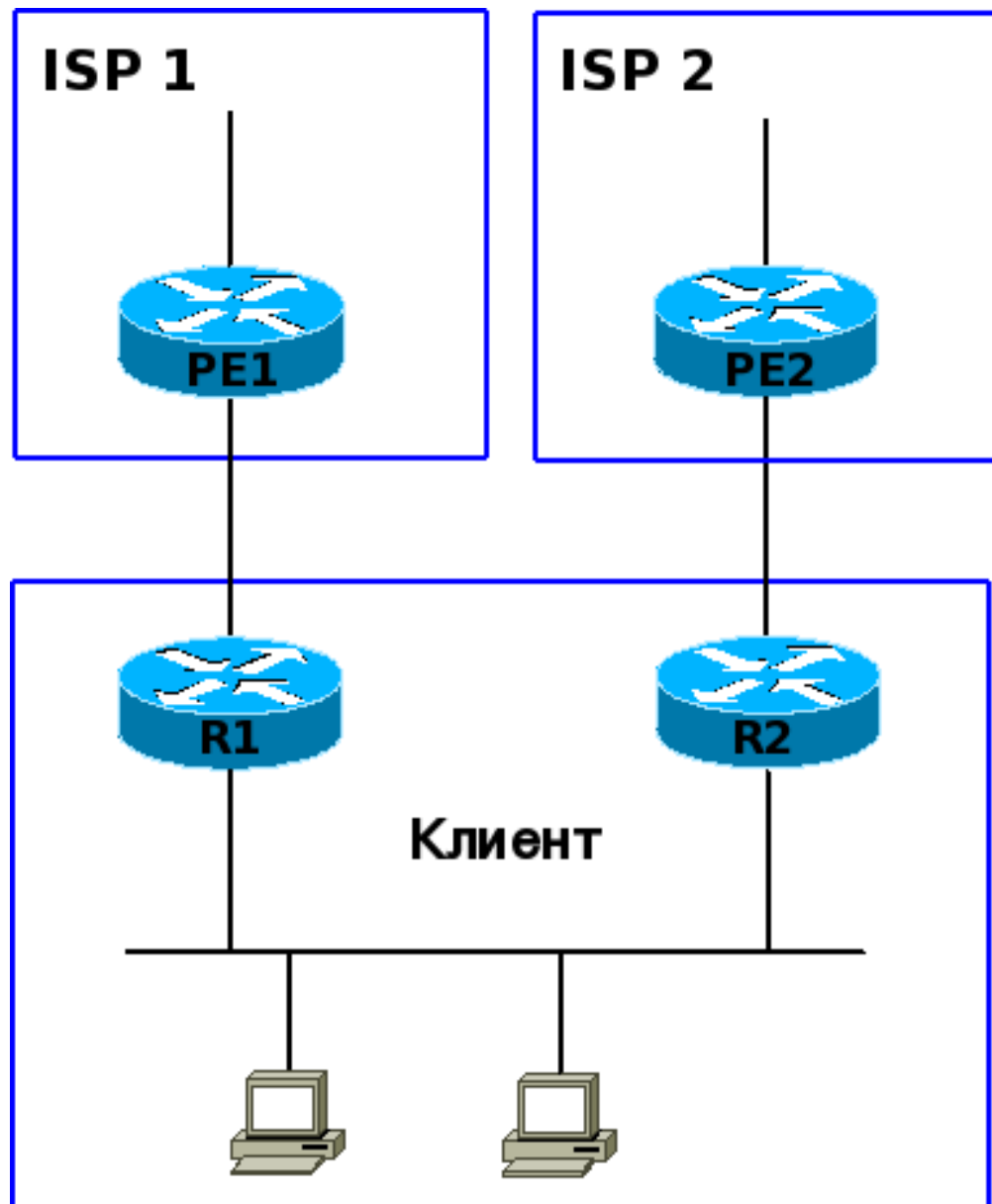
Избыточное (по устройствам) подключение к одному провайдеру

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



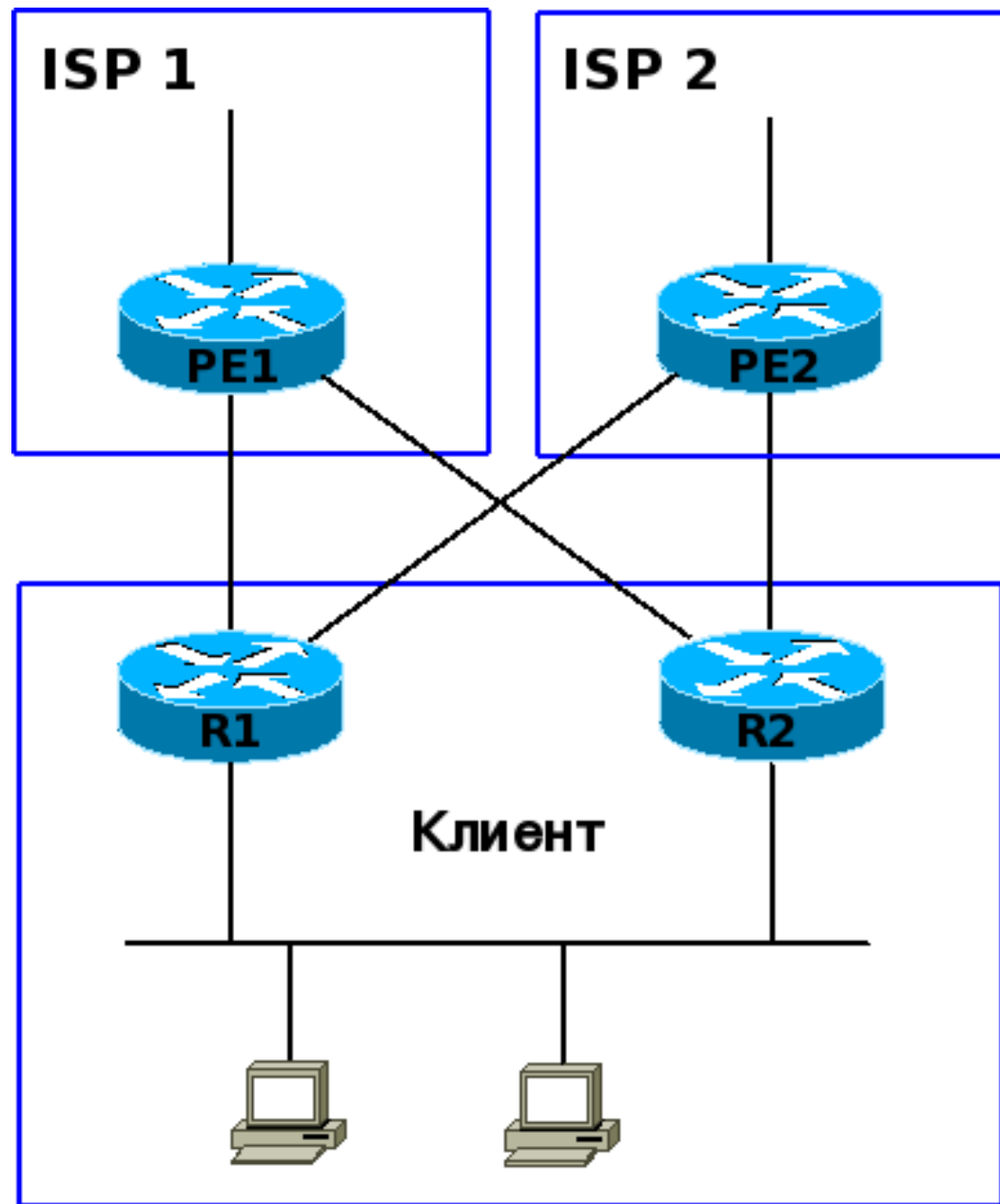
Избыточное подключение к двум провайдерам (multi-homed customer)

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



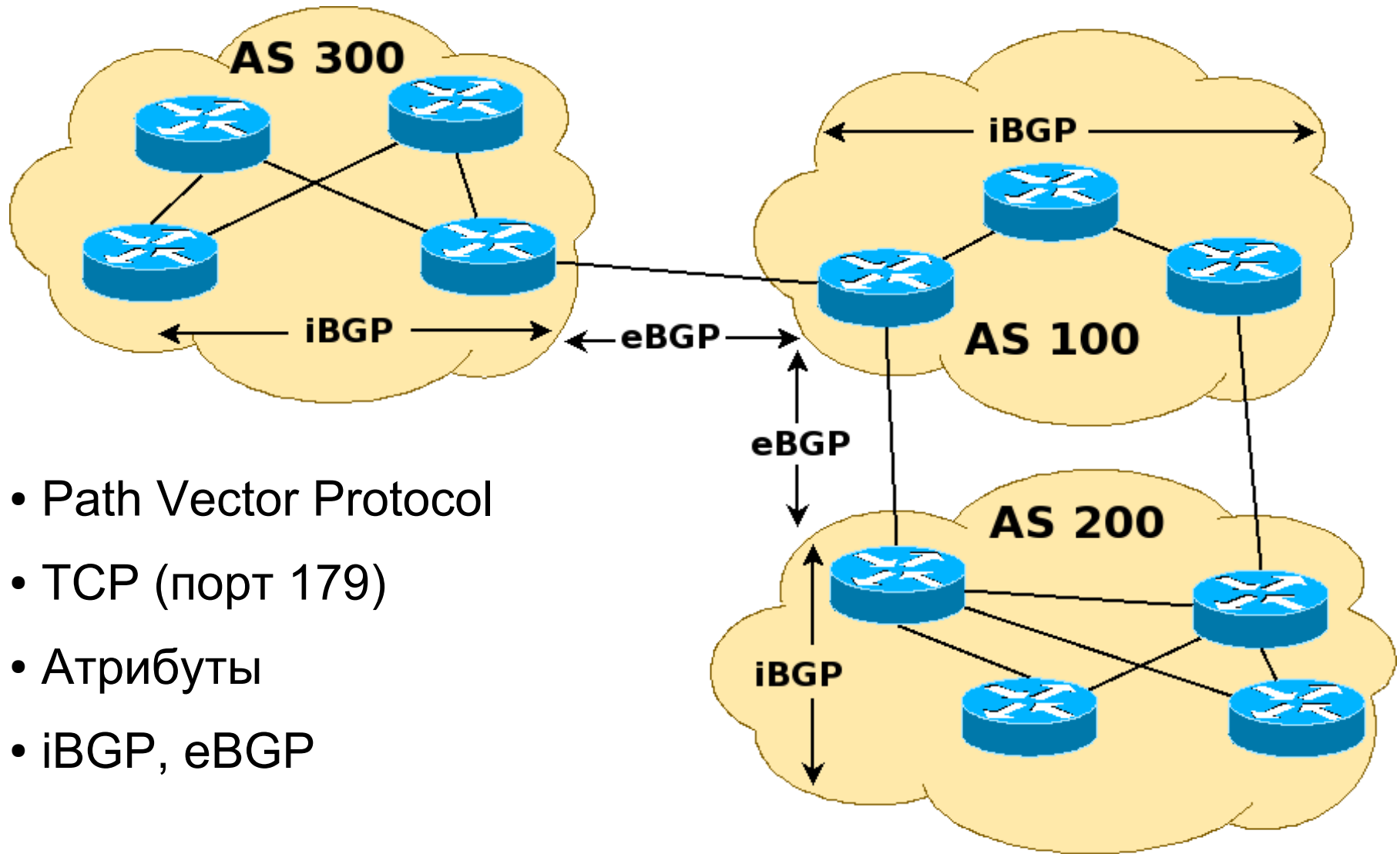
Избыточное подключение к двум провайдерам (multi-homed customer)

- 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

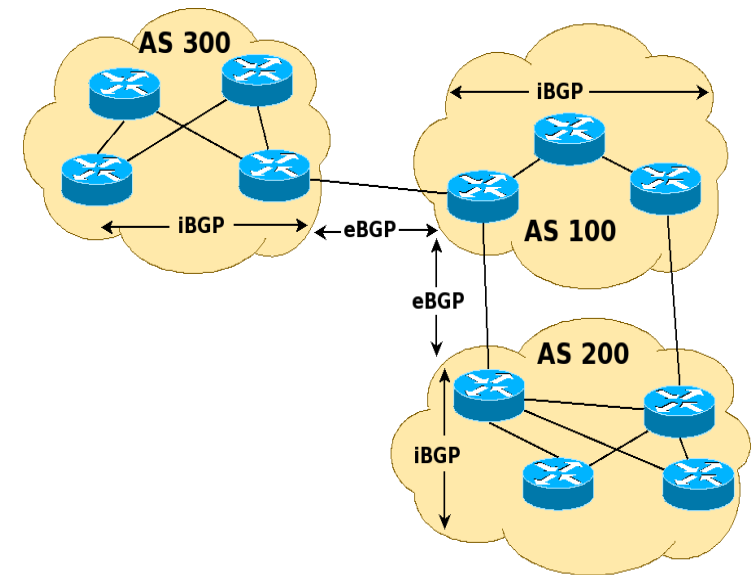
Open	используется для установки отношений соседства и обмена базовыми параметрами. Отправляется сразу после установки TCP-соединения.
Update	используется для обмена информацией о маршрутах.
Notification	отправляется когда возникают ошибки BGP. После отправки сообщения сессия с соседом разрывается.
Keepalive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

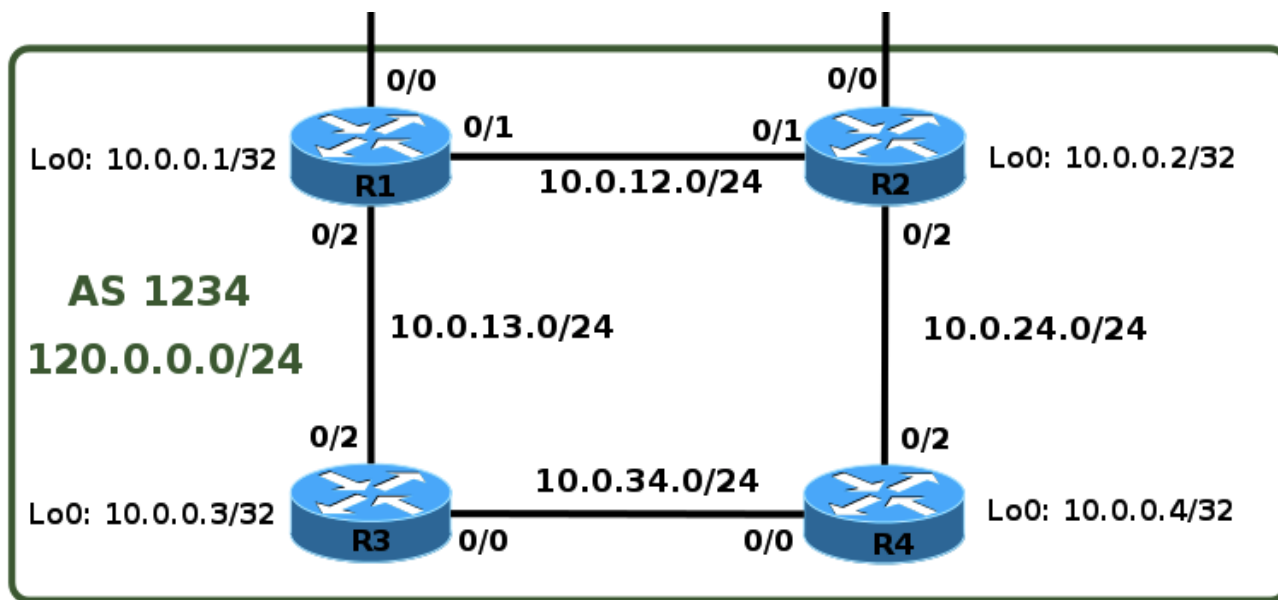
- Idle
- Connect
- Open sent
- Open confirm
- Active
- Established

Состояние	Ожидание ТСР?	Инициация ТСР?	Установлено ТСР?	Отправлен о 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.

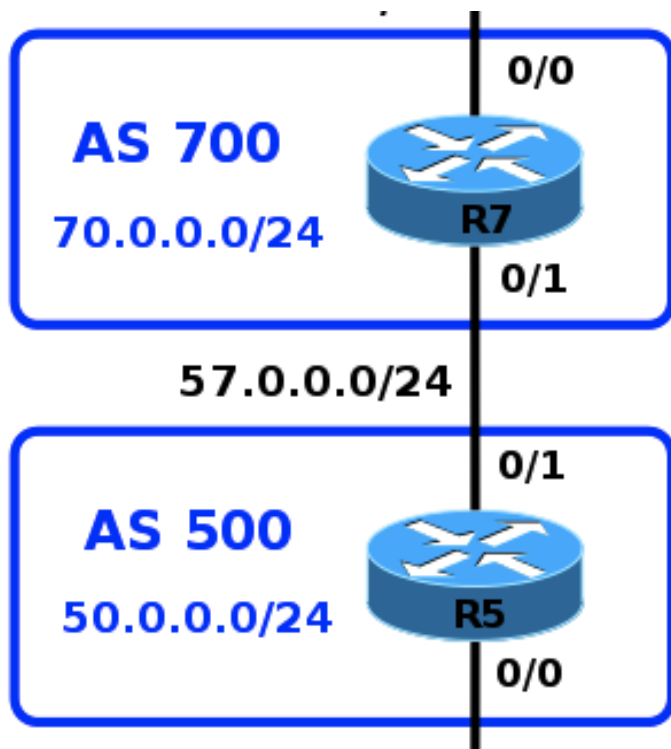
- Все iBGP-соседи внутри автономной системы должны быть соединены полносвязной топологией.
- iBGP-соседи не обязательно должны быть непосредственно соединены.



еBGP-соседи

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

По умолчанию, еBGP-соседи должны быть непосредственно соединены.

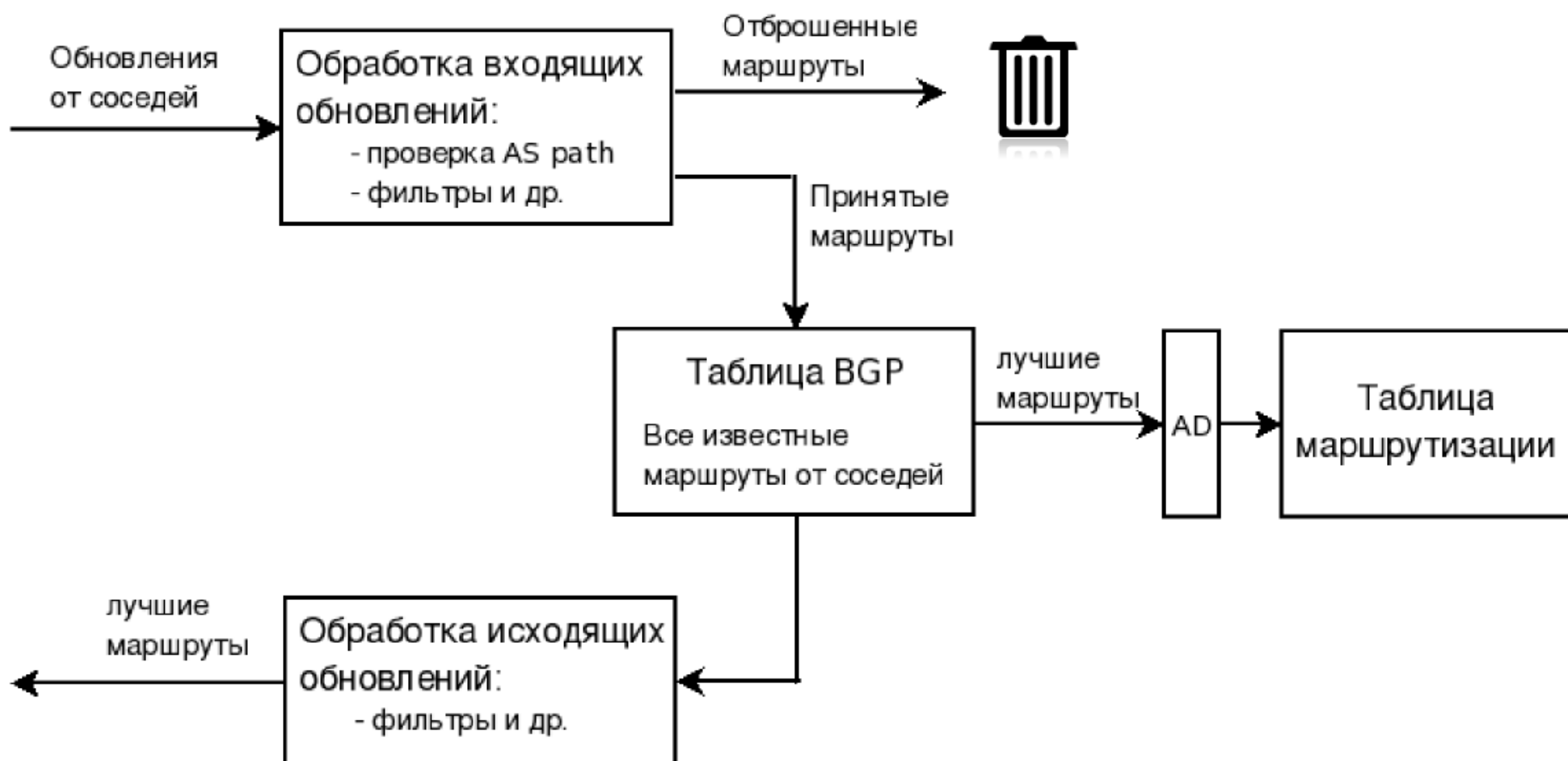


Информация BGP

- 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)
 - o Originator ID
 - o Cluster list

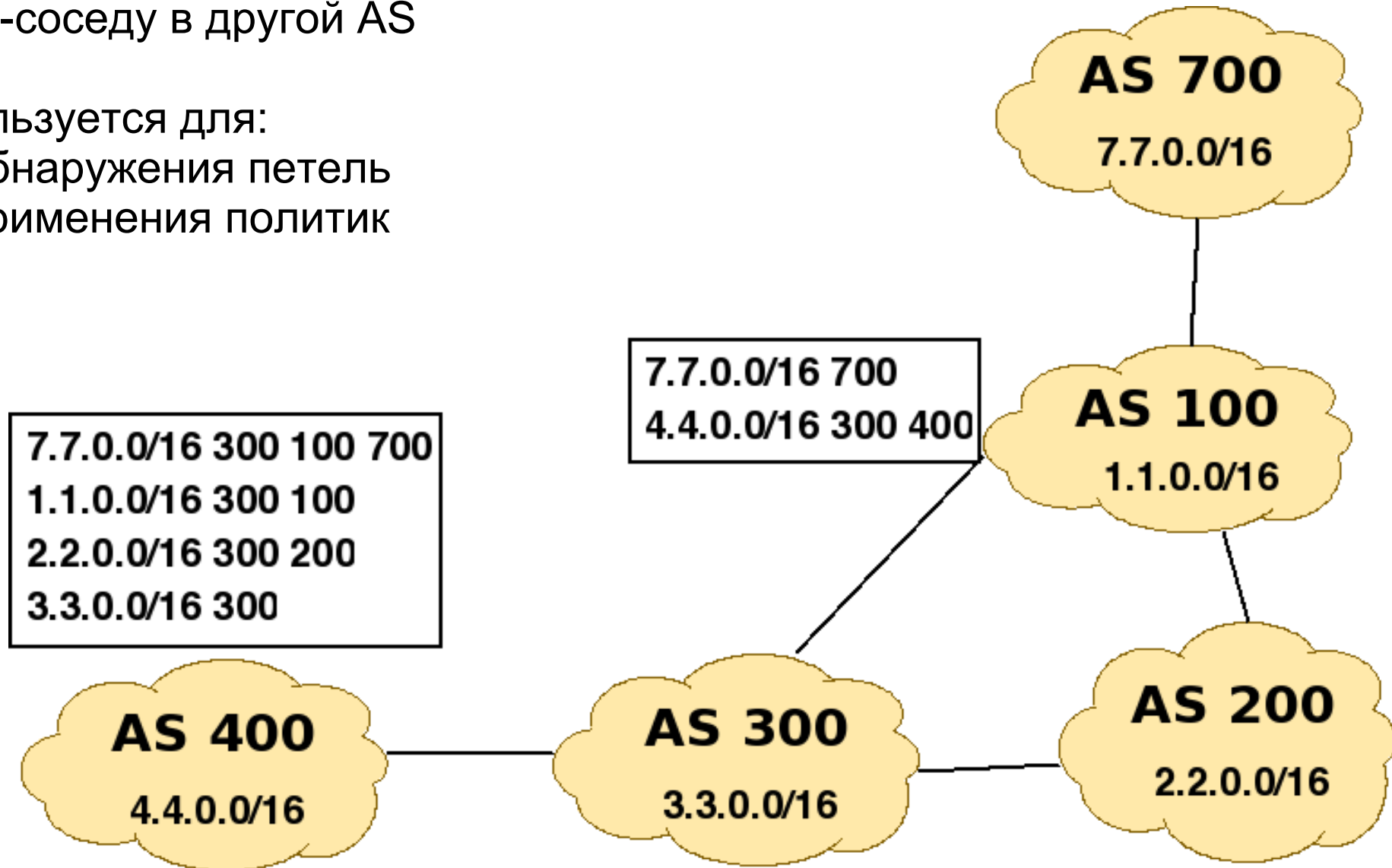
Autonomous system path

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:

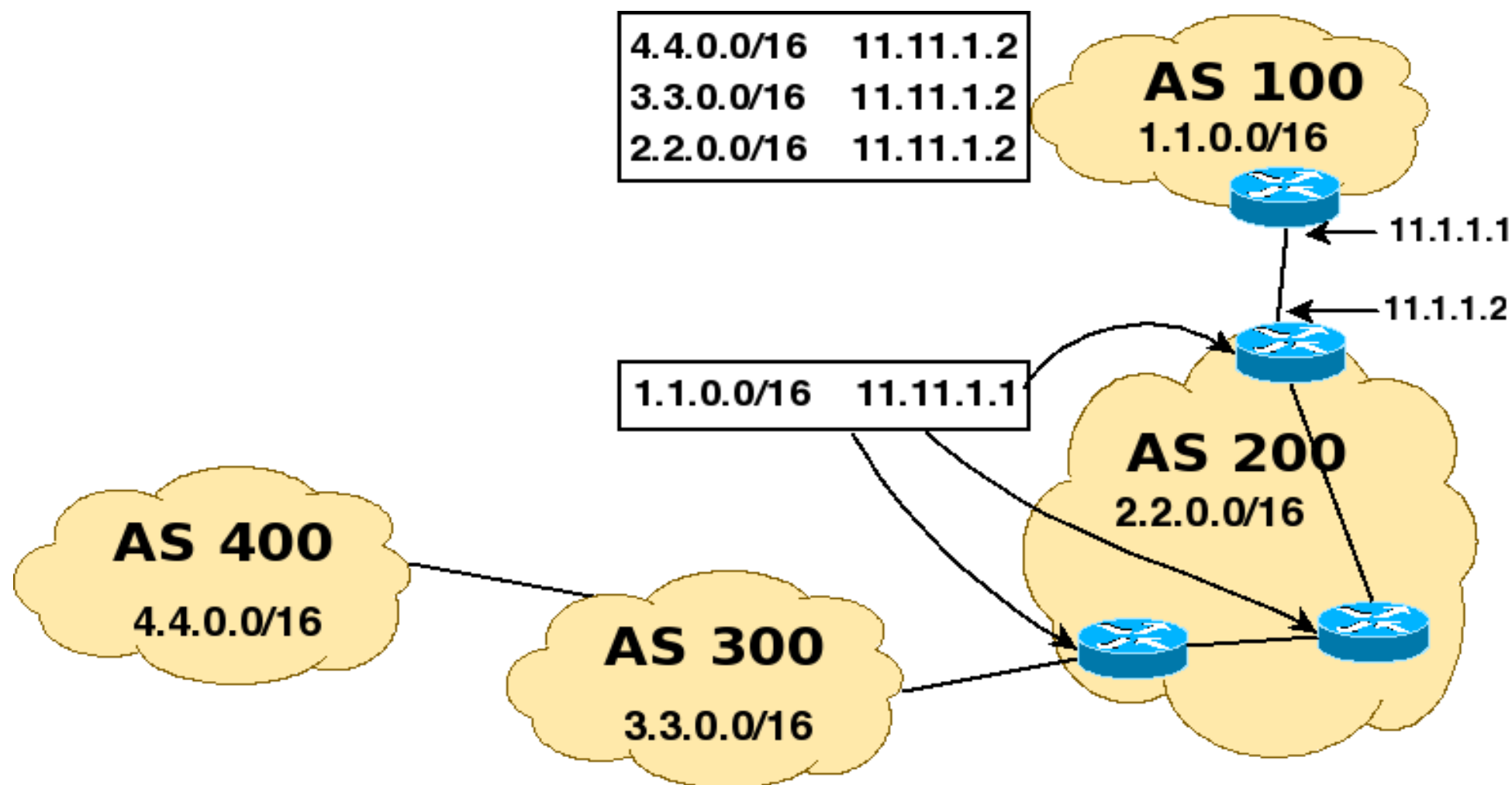
- * обнаружения петель
- * применения политик



Next hop

Атрибут Next hop:

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

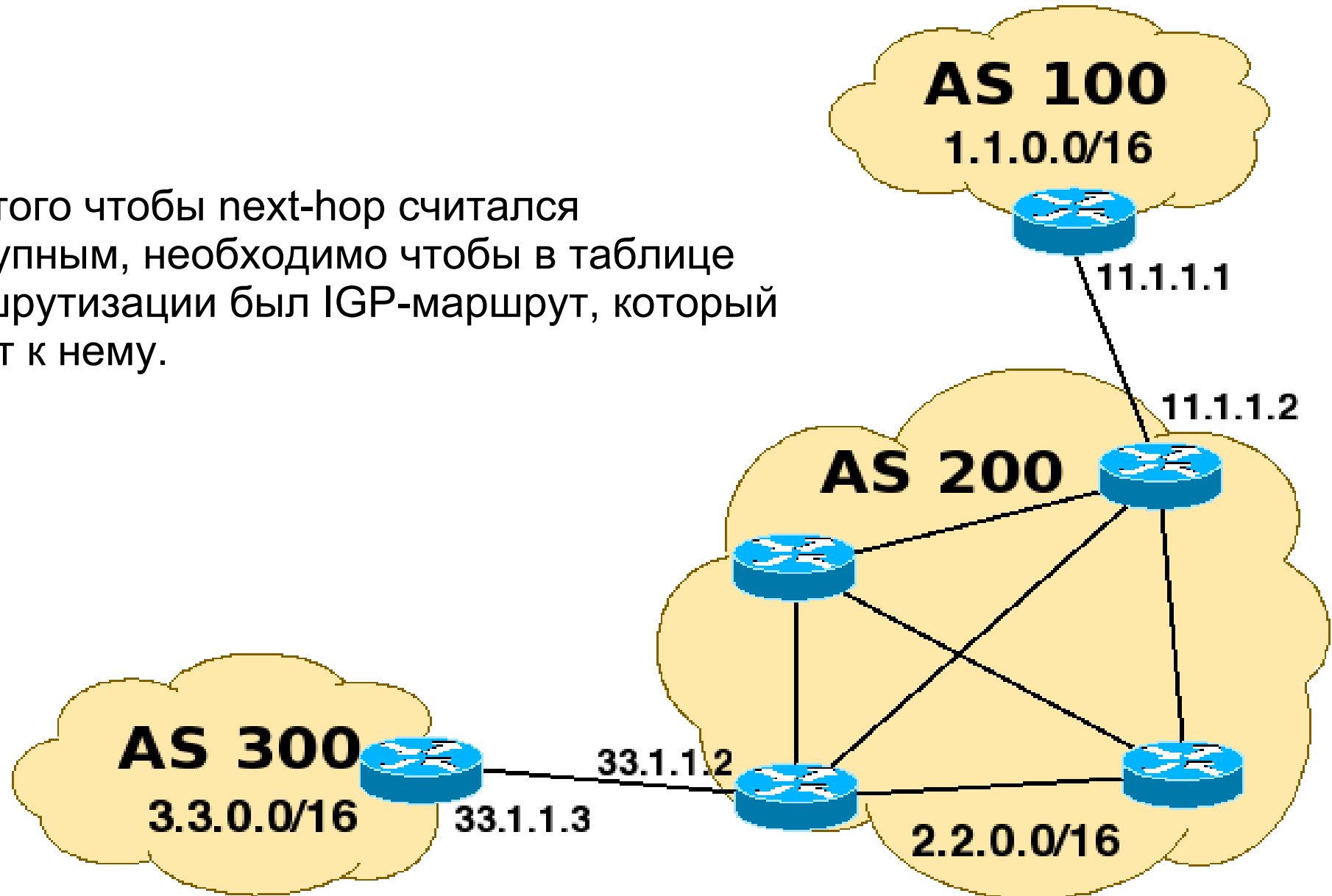


iBGP и next hop

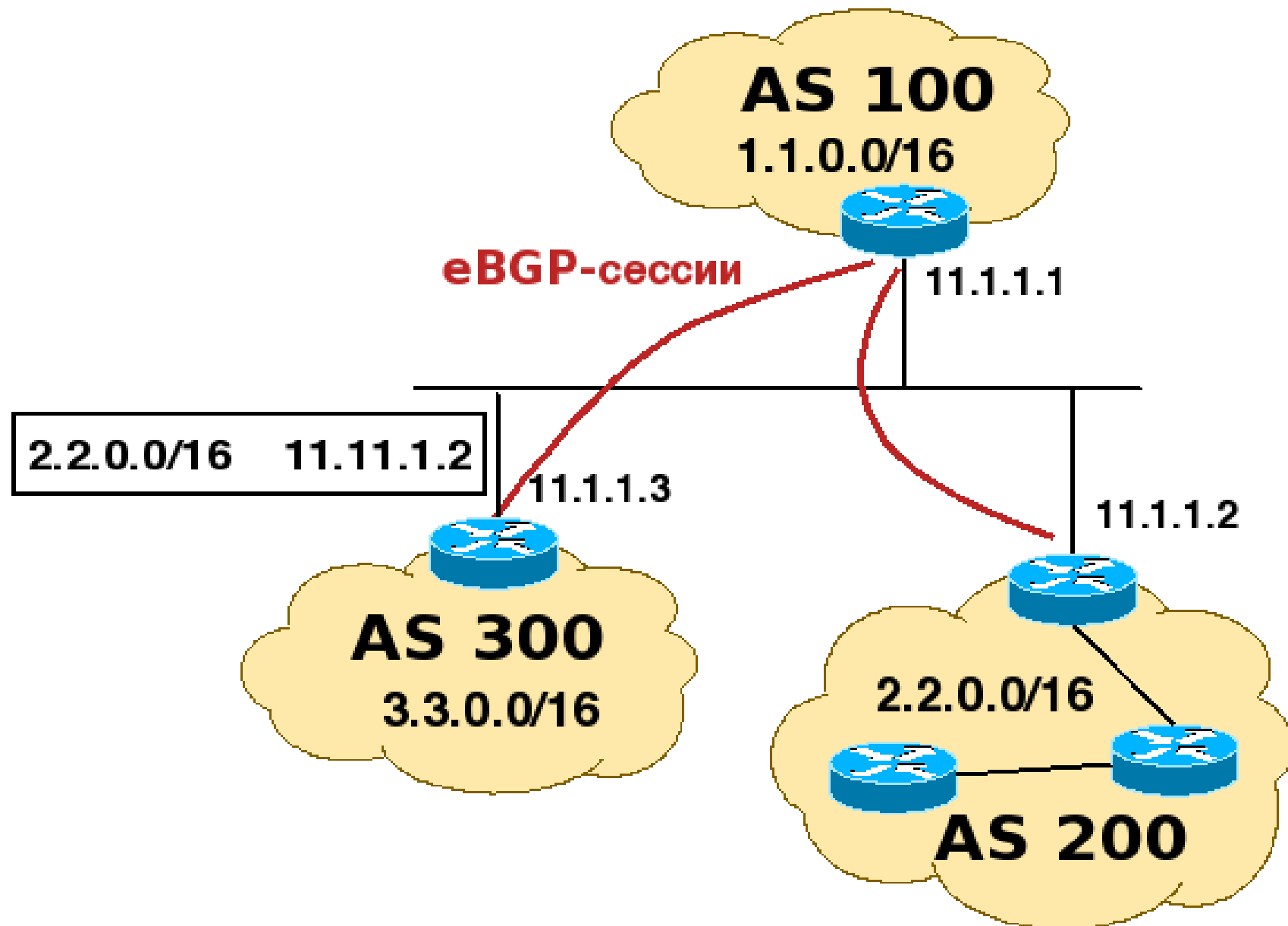
Первый шаг при выборе лучшего маршрута:

- 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.

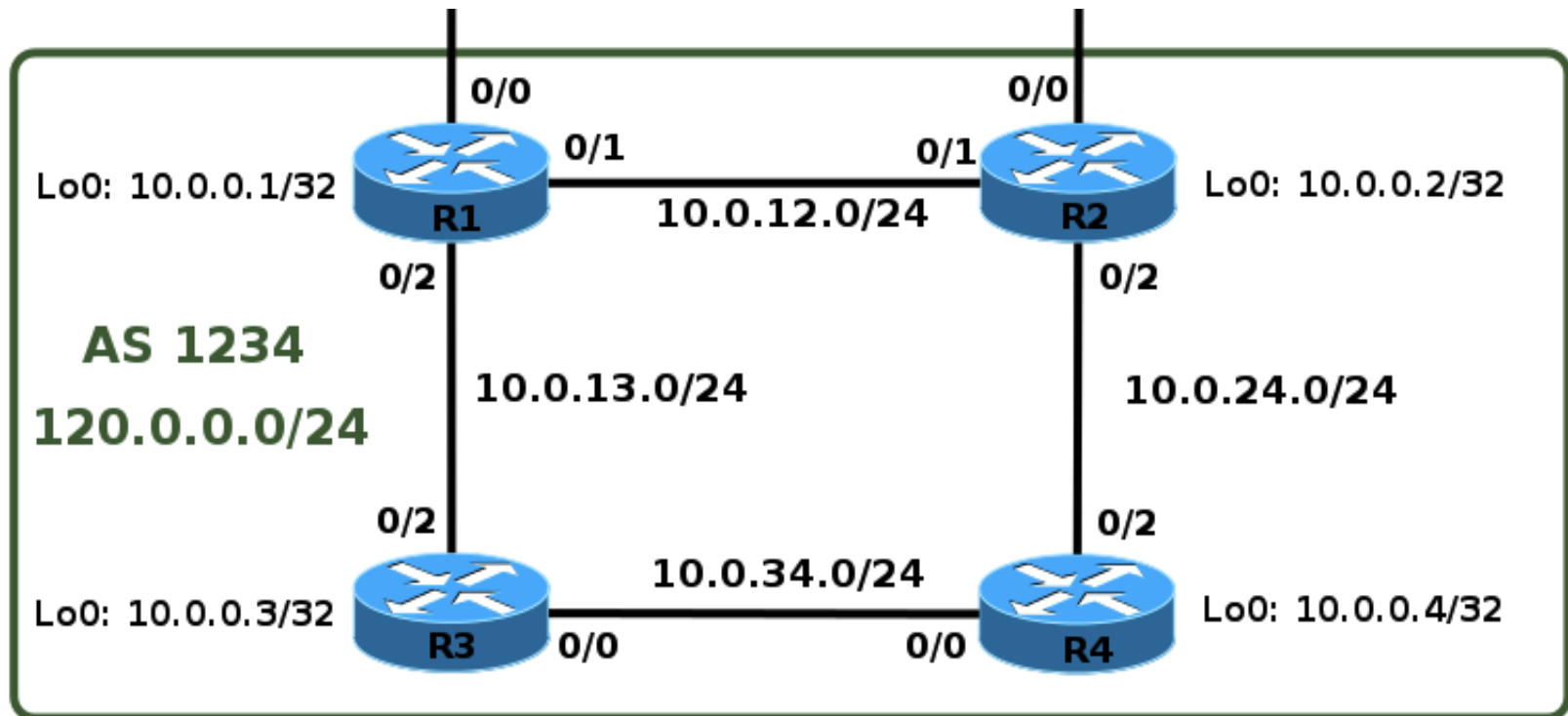
Базовые настройки BGP

Создание процесса BGP

R1 из AS 1234:

```
R1(conf)# router bgp 1234  
R1(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

R3 из AS 1234:

```
router bgp 1234
```

```
neighbor 10.0.0.1 remote-as 1234
```

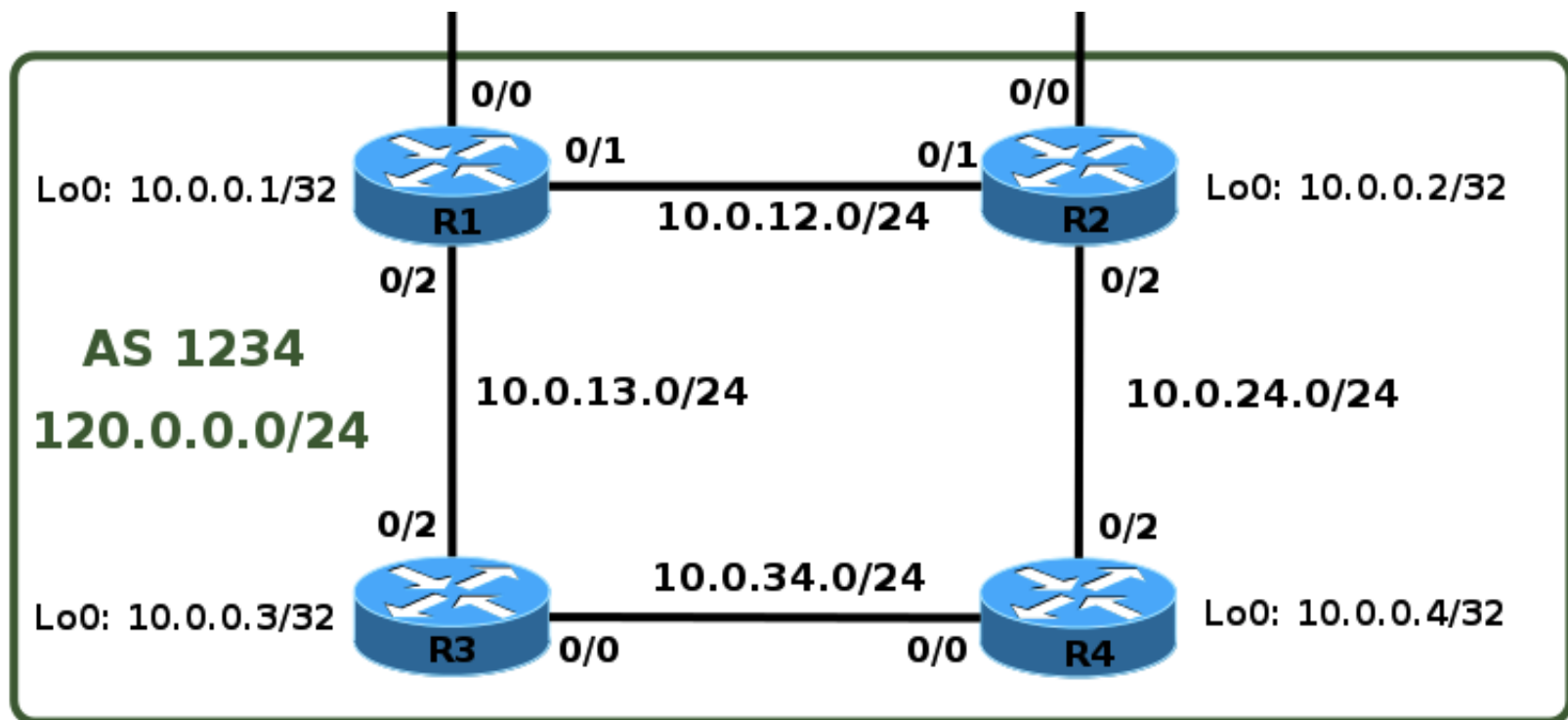
```
neighbor 10.0.0.1 update-source Loopback0
```

```
neighbor 10.0.0.2 remote-as 1234
```

```
neighbor 10.0.0.2 update-source Loopback0
```

```
neighbor 10.0.0.4 remote-as 1234
```

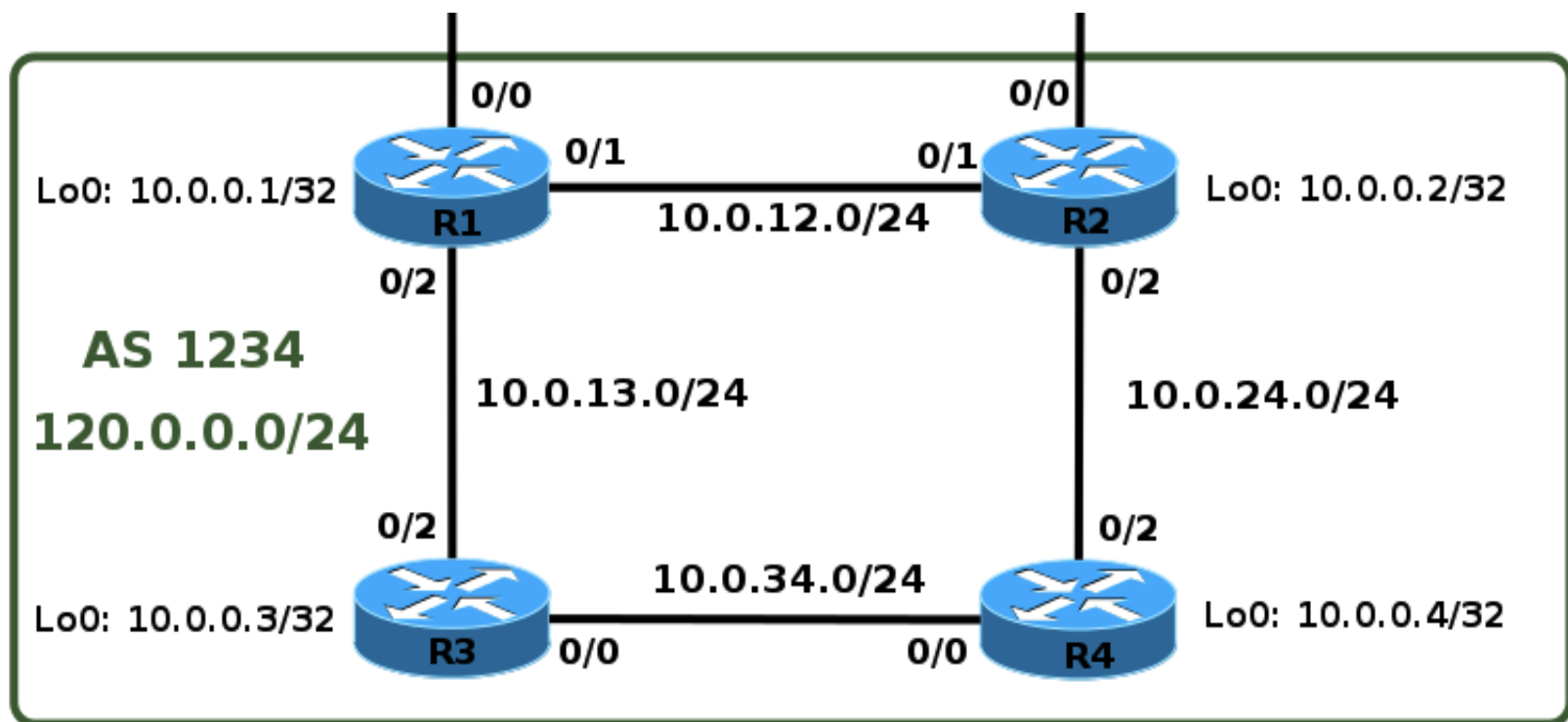
```
neighbor 10.0.0.4 update-source Loopback0
```



Использование next-hop-self

R1 из AS 1234:

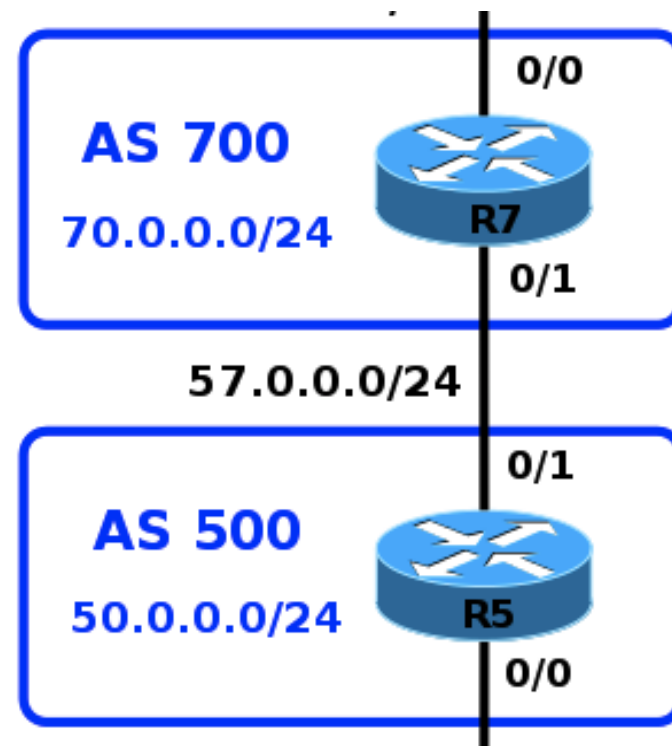
```
router bgp 1234
neighbor 10.0.0.2 remote-as 1234
neighbor 10.0.0.2 update-source Loopback0
neighbor 10.0.0.2 next-hop-self
neighbor 10.0.0.3 remote-as 1234
neighbor 10.0.0.3 update-source Loopback0
neighbor 10.0.0.3 next-hop-self
neighbor 10.0.0.4 remote-as 1234
neighbor 10.0.0.4 update-source Loopback0
neighbor 10.0.0.4 next-hop-self
```



Настройка внешних соседей BGP

```
router bgp 700  
  neighbor 57.0.0.5 remote-as 500
```

```
router bgp 500  
  neighbor 57.0.0.7 remote-as 700
```



Настройка внешних соседей BGP

```
router bgp 100
  neighbor 190.16.1.1 remote-as 500
  neighbor 190.16.1.1 ebgp-multihop 2
  neighbor 190.16.1.1 update-source Loopback0
```

```
router bgp 500
  neighbor 190.16.100.1 remote-as 100
  neighbor 190.16.100.1 ebgp-multihop 2
```

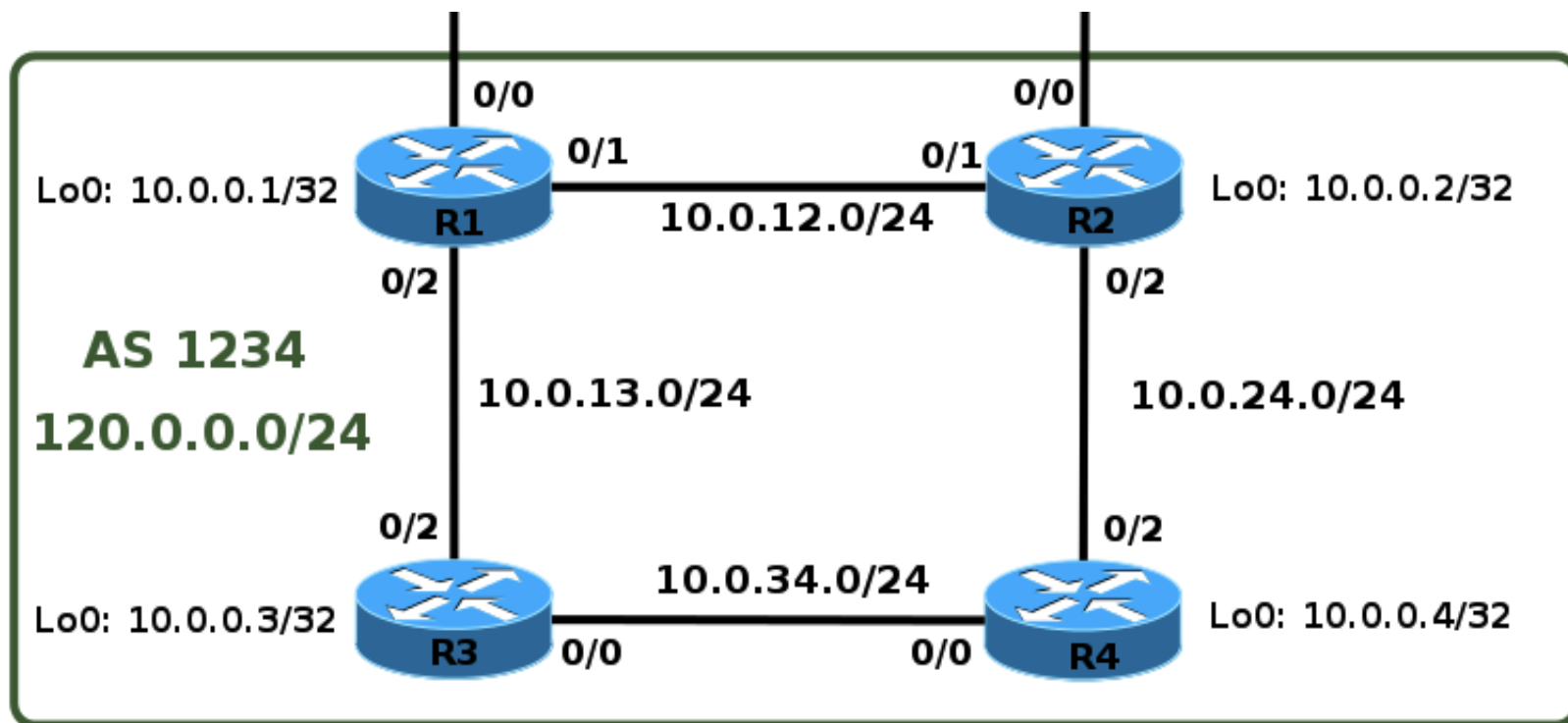
```
ip route 190.16.100.1 255.255.255.255 190.16.1.2
```

Анонсирование префиксов в BGP

Анонсирование сетей в BGP

Два механизма анонсирования сетей:

- команда network
- команда redistribute



Анонсирование сетей командой network

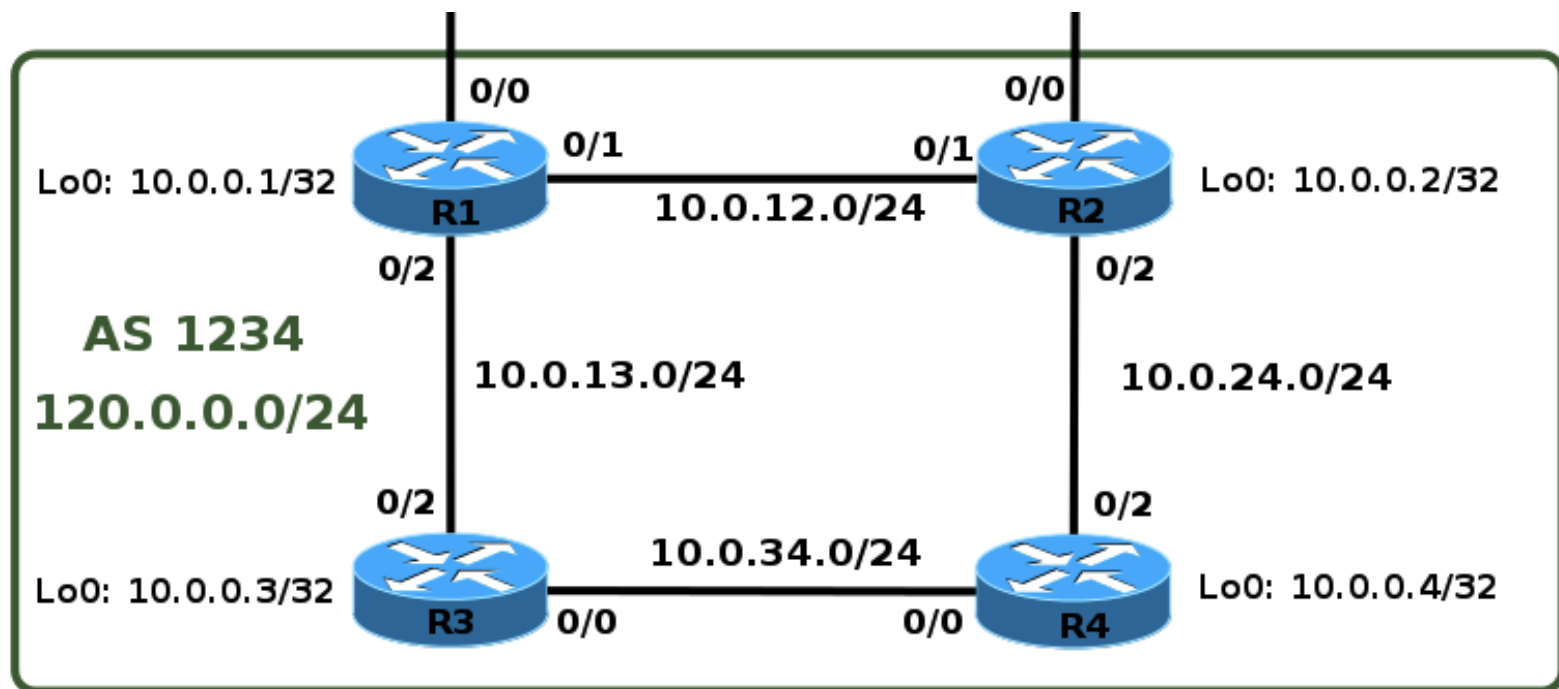
Соответствующий маршрут должен существовать в таблице маршрутизации, прежде чем сеть будет анонсирована

Значение атрибута Origin будет "IGP"

R1:

```
router bgp 1234
 network 120.0.0.0 mask 255.255.255.0

ip route 120.0.0.0 255.255.255.0 Null0
```



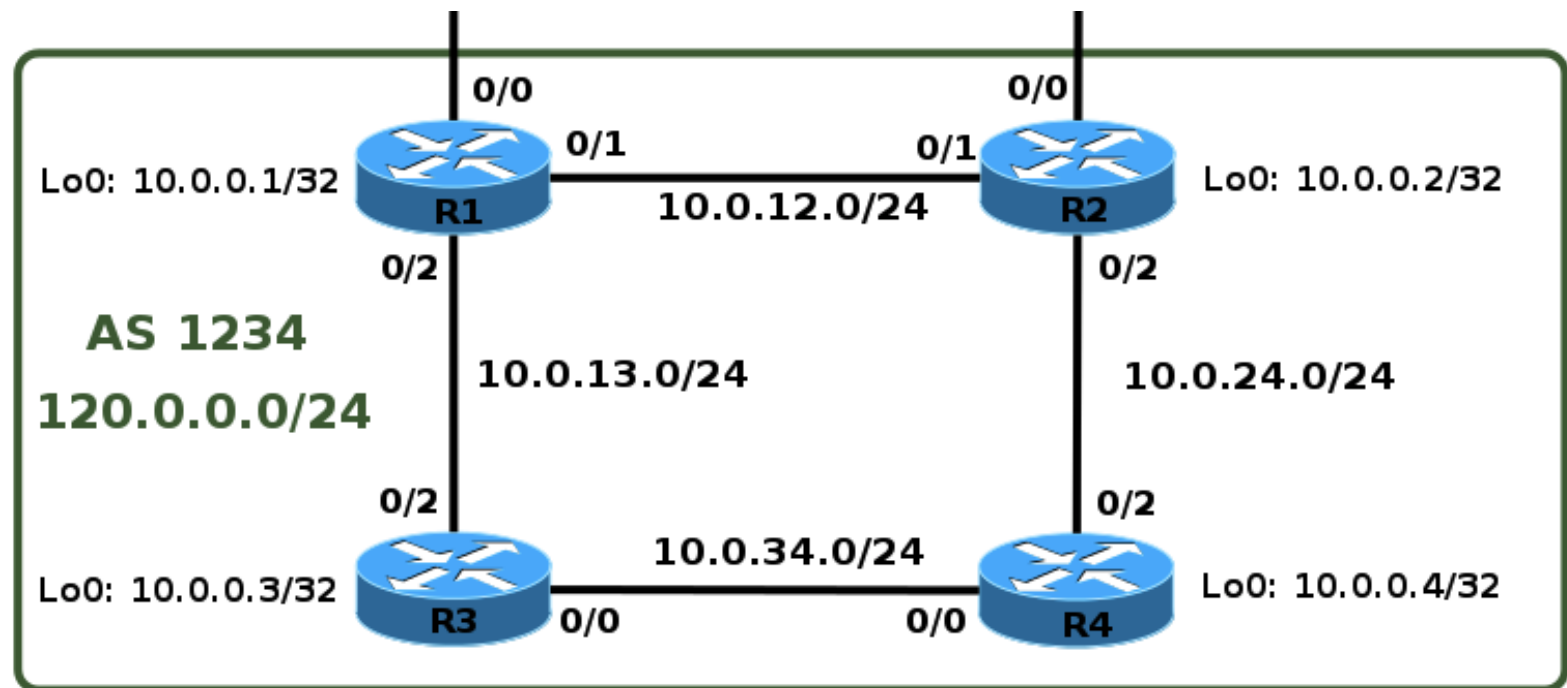
Анонсирование сетей командой redistribute

Помещает маршруты протокола IGP или статические маршруты в процесс BGP.

Значение атрибута Origin будет “Incomplete”

```
router bgp 1234  
  redistribute static
```

```
ip route 120.0.0.0 255.255.255.0 Null0
```



Просмотр информации BGP

Настройки BGP

```
R1#sh run | s ^router bgp
router bgp 1234
  network 120.0.0.0 mask 255.255.255.0
  neighbor 10.0.0.2 remote-as 1234
  neighbor 10.0.0.2 update-source Loopback0
  neighbor 10.0.0.2 next-hop-self
  neighbor 10.0.0.3 remote-as 1234
  neighbor 10.0.0.3 update-source Loopback0
  neighbor 10.0.0.3 next-hop-self
  neighbor 10.0.0.4 remote-as 1234
  neighbor 10.0.0.4 update-source Loopback0
  neighbor 10.0.0.4 next-hop-self
  neighbor 15.0.0.5 remote-as 500
```

Соседи BGP

```
R1#sh ip bgp summary
BGP router identifier 120.0.0.1, local AS number 1234
BGP table version is 13, main routing table version 13
12 network entries using 1728 bytes of memory
19 path entries using 1140 bytes of memory
17/8 BGP path/bestpath attribute entries using 2312 bytes of memory
12 BGP AS-PATH entries using 384 bytes of memory
5 BGP route-map cache entries using 180 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP using 5744 total bytes of memory
6 received paths for inbound soft reconfiguration
BGP activity 12/0 prefixes, 21/2 paths, scan interval 60 secs
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
10.0.0.2	4	1234	16	9	13	0	0	00:00:08	10
10.0.0.3	4	1234	44	47	13	0	0	00:37:34	0
10.0.0.4	4	1234	6	8	13	0	0	00:00:05	0
15.0.0.5	4	500	50	45	13	0	0	00:38:18	2

Таблица BGP

```
R1#sh ip bgp
```

```
BGP table version is 13, local router ID is 120.0.0.1
```

```
Status codes: s suppressed, d damped, h history, * valid, > best,  
i - internal, r RIB-failure, S Stale, m multipath,  
b backup-path, f RT-Filter, x best-external,  
a additional-path, c RIB-compressed,
```

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

```
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
r>i	10.0.0.2/32	10.0.0.2	0	100	0	?
r>i	10.0.12.0/24	10.0.0.2	0	100	0	?
r>i	10.0.24.0/24	10.0.0.2	0	100	0	?
*>i	26.0.0.0/24	10.0.0.2	0	100	0	?
*>	50.0.0.0/24	15.0.0.5	0	500	0	500 i
*>i	60.0.0.0/24	10.0.0.2	0	100	0	600 i
*>	70.0.0.0/24	15.0.0.5		500	0	500 700 i
*>i	80.0.0.0/24	10.0.0.2	0	100	0	600 800 i
*>i	90.0.0.0/24	10.0.0.2	0	100	0	600 800 1000 900 i
*>i	100.0.0.0/24	10.0.0.2	0	100	0	600 800 1000 i
* i	120.0.0.0/24	10.0.0.2	0	100	0	i
*>		0.0.0.0	0		32768	i
r>i	120.0.0.2/32	10.0.0.2	0	100	0	?

Подробная информация о маршруте в таблице BGP

```
R1#sh ip bgp 70.0.0.0/24
BGP routing table entry for 70.0.0.0/24, version 4
Paths: (2 available, best #1, table default)
  Advertised to update-groups:
    2
  Refresh Epoch 1
  500 700
    15.0.0.5 from 15.0.0.5 (50.0.0.5)
      Origin IGP, localpref 500, valid, external, best
  Refresh Epoch 1
  500 700, (received-only)
    15.0.0.5 from 15.0.0.5 (50.0.0.5)
      Origin IGP, localpref 100, valid, external
```

Маршруты с кодом r>:

```
R1#sh ip bgp rib-failure
```

Network	Next Hop	RIB-failure	RIB-NH	Matches
10.0.0.2/32	10.0.0.2	Higher admin distance		n/a
10.0.12.0/24	10.0.0.2	Higher admin distance		n/a
10.0.24.0/24	10.0.0.2	Higher admin distance		n/a
120.0.0.2/32	10.0.0.2	Higher admin distance		n/a

Таблица маршрутизации

```
R1#sh ip route bgp
```

```
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP  
D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area  
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2  
E1 - OSPF external type 1, E2 - OSPF external type 2  
i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2  
ia - IS-IS inter area, * - candidate default, U - per-user static route  
o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP  
+ - replicated route, % - next hop override
```

```
Gateway of last resort is not set
```

```
26.0.0.0/24 is subnetted, 1 subnets  
B      26.0.0.0 [200/0] via 10.0.0.2, 00:06:55  
50.0.0.0/24 is subnetted, 1 subnets  
B      50.0.0.0 [20/0] via 15.0.0.5, 00:43:54  
60.0.0.0/24 is subnetted, 1 subnets  
B      60.0.0.0 [200/0] via 10.0.0.2, 00:06:55  
70.0.0.0/24 is subnetted, 1 subnets  
B      70.0.0.0 [20/0] via 15.0.0.5, 00:43:31  
80.0.0.0/24 is subnetted, 1 subnets  
B      80.0.0.0 [200/0] via 10.0.0.2, 00:06:55  
90.0.0.0/24 is subnetted, 1 subnets  
B      90.0.0.0 [200/0] via 10.0.0.2, 00:06:55  
100.0.0.0/24 is subnetted, 1 subnets  
B      100.0.0.0 [200/0] via 10.0.0.2, 00:06:55
```

Фильтрация таблицы BGP

```
R1#sh ip bgp 120.0.0.0/24 longer-prefixes
```

```
BGP table version is 13, local router ID is 120.0.0.1
```

```
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,  
               r RIB-failure, S Stale, m multipath, b backup-path, f RT-Filter,  
               x best-external, a additional-path, c RIB-compressed,
```

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

```
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
* i	120.0.0.0/24	10.0.0.2	0	100	0	i
*>		0.0.0.0	0		32768	i
r>i	120.0.0.2/32	10.0.0.2	0	100	0	?

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:

```
R1#sh ip bgp regexp _800_
```

```
BGP table version is 13, local router ID is 120.0.0.1
```

```
Status codes: s suppressed, d damped, h history, * valid, > best,  
               i - internal, r RIB-failure, S Stale, m multipath,  
               b backup-path, f RT-Filter,  
               x best-external, a additional-path, c RIB-compressed,
```

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

```
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>i	80.0.0.0/24	10.0.0.2	0	100	0 600 800	i
*>i	90.0.0.0/24	10.0.0.2	0	100	0 600 800 1000 900	i
*>i	100.0.0.0/24	10.0.0.2	0	100	0 600 800 1000	i

Маршруты, которые анонсируются соседу BGP

```
R1#sh ip bgp neighbors 15.0.0.5 advertised-routes
BGP table version is 13, local router ID is 120.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
                r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,
                x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
r>i	10.0.0.2/32	10.0.0.2	0	100	0	?
r>i	10.0.12.0/24	10.0.0.2	0	100	0	?
r>i	10.0.24.0/24	10.0.0.2	0	100	0	?
*>i	26.0.0.0/24	10.0.0.2	0	100	0	?
*>i	60.0.0.0/24	10.0.0.2	0	100	0	600 i
*>i	80.0.0.0/24	10.0.0.2	0	100	0	600 800 i
*>i	90.0.0.0/24	10.0.0.2	0	100	0	600 800 1000 900 i
*>i	100.0.0.0/24	10.0.0.2	0	100	0	600 800 1000 i
*>	120.0.0.0/24	0.0.0.0	0		32768	i
r>i	120.0.0.2/32	10.0.0.2	0	100	0	?

Total number of prefixes 10

Маршруты, полученные от соседа BGP

```
R1#sh ip bgp neighbors 15.0.0.5 routes
BGP table version is 13, local router ID is 120.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
                r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,
                x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

	Network	Next Hop	Metric	LocPrf	Weight	Path
*>	50.0.0.0/24	15.0.0.5	0	500	0 500	i
*>	70.0.0.0/24	15.0.0.5		500	0 500 700	i

```
Total number of prefixes 2
```

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

Объекты для работы с префиксами и политиками BGP

Объекты для работы с префиксами и политиками BGP

Объекты для фильтрации маршрутов:

- 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

Объект для изменения атрибутов:

- Route-map

AS path access list

AS-path access-list

Синтаксис:

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

```
router bgp 1234
 network 120.0.0.0 mask 255.255.255.0
 neighbor 15.0.0.5 filter-list 1 out
 neighbor 15.0.0.5 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 в AS100

Без настройки фильтрации dyn1 анонсирует соседним маршрутизаторам все сети, которые он получил по BGP

```
R1#sh ip bgp neighbors 15.0.0.5 advertised-routes
BGP table version is 13, local router ID is 120.0.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i -
internal,
                r RIB-failure, S Stale, m multipath, b backup-path, f RT-
Filter,
                x best-external, a additional-path, c RIB-compressed,
Origin codes: i - IGP, e - EGP, ? - incomplete
RPKI validation codes: V valid, I invalid, N Not found
```

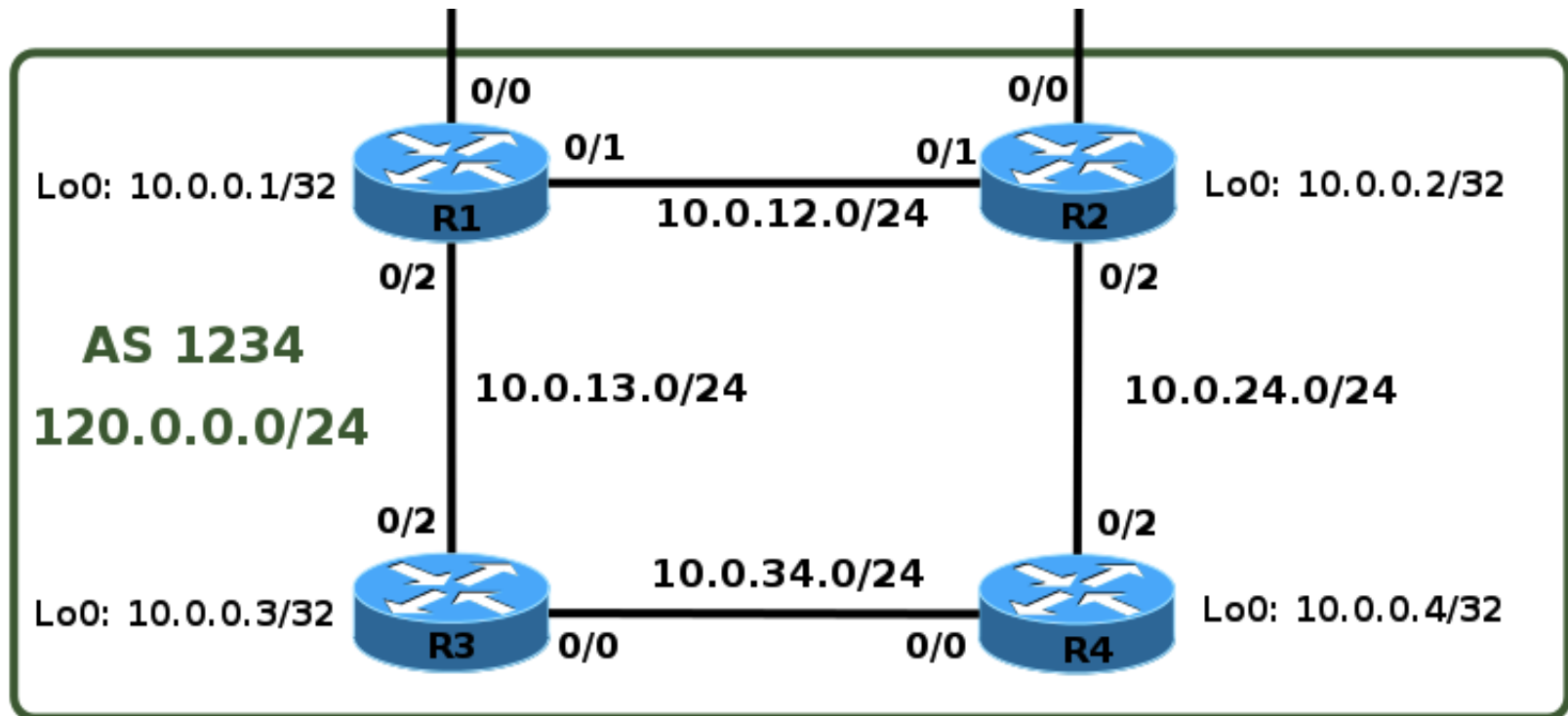
	Network	Next Hop	Metric	LocPrf	Weight	Path
*>i	26.0.0.0/24	10.0.0.2	0	100	0	?
*>i	60.0.0.0/24	10.0.0.2	0	100	0	600 i
*>i	80.0.0.0/24	10.0.0.2	0	100	0	600 800 i
*>i	90.0.0.0/24	10.0.0.2	0	100	0	600 800 1000 900 i
*>i	100.0.0.0/24	10.0.0.2	0	100	0	600 800 1000 i
*>	120.0.0.0/24	0.0.0.0	0		32768	i
r>i	120.0.0.2/32	10.0.0.2	0	100	0	?

Total number of prefixes 10

Фильтрация исходящих анонсов в AS100

R1 из AS 1234:

```
ip as-path access-list 100 permit ^$  
!  
router bgp 1234  
  neighbor 15.0.0.5 filter-list 100 out
```



Prefix list

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 90.0.0.0/8

<u>90.0.0.0/8</u>	90.0.0.0/10	90.0.0.0/16	90.0.0.0/24
-------------------	-------------	-------------	-------------

ip prefix-list NetDay permit 90.0.0.0/8 le 10

<u>90.0.0.0/8</u>	<u>90.0.0.0/10</u>	90.0.0.0/16	90.0.0.0/24
-------------------	--------------------	-------------	-------------

ip prefix-list NetDay permit 90.0.0.0/8 ge 11

90.0.0.0/8	90.0.0.0/10	<u>90.0.0.0/16</u>	<u>90.0.0.0/24</u>
------------	-------------	--------------------	--------------------

ip prefix-list NetDay permit 90.0.0.0/8 ge 10 le 20

90.0.0.0/8	<u>90.0.0.0/10</u>	<u>90.0.0.0/16</u>	90.0.0.0/24
------------	--------------------	--------------------	-------------

Prefix-list

Синтаксис:

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

```
router bgp 1234  
  network 120.0.0.0 mask 255.255.255.0  
  neighbor 15.0.0.5 prefix-list AS500_OUT out
```

```
ip prefix-list AS500_OUT permit 120.0.0.0/24
```


Проверка prefix-list

```
R1#sh ip prefix-list
ip prefix-list AS500_OUT: 1 entries
    seq 5 permit 120.0.0.0/24
ip prefix-list FILTER: 2 entries
    seq 5 permit 50.0.0.0/24
    seq 10 permit 70.0.0.0/24
```

```
R1#sh ip prefix-list detail
Prefix-list with the last deletion/insertion: AS500_OUT
ip prefix-list AS500_OUT:
    count: 1, range entries: 0, sequences: 5 - 5, refcount: 2
    seq 5 permit 120.0.0.0/24 (hit count: 0, refcount: 1)
ip prefix-list FILTER:
    count: 2, range entries: 0, sequences: 5 - 10, refcount: 2
    seq 5 permit 50.0.0.0/24 (hit count: 2, refcount: 1)
    seq 10 permit 70.0.0.0/24 (hit count: 1, refcount: 2)
```

Отобразить маршруты, которые совпадают с prefix-list AS500_OUT:

```
R1# show ip bgp prefix-list AS500_OUT
```

Route map

Route-map

Позволяет управлять атрибутами и фильтровать префиксы

```
route-map ISP_IN permit 10
  match ip address prefix-list AS500 AS600
  set local-preference 120
!
route-map ISP_IN permit 20
  match ip address prefix-list AS700
  match as-path 100
  set local-preference 80
!
route-map ISP_IN permit 30
```

AS500 ИЛИ AS600

AS700 И as-path 100

разрешить остальные

Route-map

```
router bgp 1234
  neighbor 26.0.0.6 remote-as 600
  neighbor 26.0.0.6 route-map AS600_IN in
```

```
route-map AS600_IN permit 10
  match as-path 1
```

```
route-map AS600_IN deny 20
  match as-path 2
```

```
route-map AS600_IN permit 30
```

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

Логика работы route-map

Логика работы route-map

```
router bgp 1234
  neighbor 26.0.0.6 remote-as 600
  neighbor 26.0.0.6 route-map AS600_IN in

route-map AS600_IN permit 10
  match ip address prefix-list AS_1100

route-map AS600_IN deny 20
  match ip address prefix-list AS_1200

route-map AS600_IN permit 50

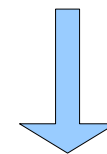
ip prefix-list AS_1100 deny 190.16.14.0/24
ip prefix-list AS_1100 permit 0.0.0.0/0 le 32

ip prefix-list AS_1200 permit 190.16.11.0/22
ip prefix-list AS_1200 deny 0.0.0.0/0 le 32
```

Логика работы route-map

```
router bgp 1234
  neighbor 26.0.0.6 remote-as 600
  neighbor 26.0.0.6 route-map AS600_IN in
```

190.16.10.0/24
190.16.11.0/24
190.16.12.0/24
190.16.14.0/24
190.16.55.0/24



```
route-map AS600_IN permit 10
  match ip address prefix-list AS_1100
    deny 190.16.14.0/24
    permit 190.16.12.0/24
    permit 190.16.10.0/24
```

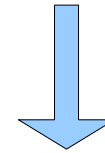
```
route-map AS600_IN deny 20
  match ip address prefix-list AS_1200
    permit 190.16.11.0/24
    deny 0.0.0.0/0 le 32
```

```
route-map AS600_IN permit 50
```

Логика работы route-map

```
router bgp 1234
  neighbor 26.0.0.6 remote-as 600
  neighbor 26.0.0.6 route-map AS600_IN in
```

190.16.10.0/24



```
route-map AS600_IN permit 10
  match ip address prefix-list AS_1100
    deny 190.16.14.0/24
    permit 190.16.12.0/24
    permit 190.16.10.0/24
```

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

```
route-map AS600_IN deny 20
  match ip address prefix-list AS_1200
    permit 190.16.11.0/24
    deny 0.0.0.0/0 le 32
```

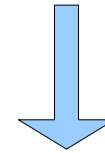
В правиле route-map стоит permit, поэтому маршрут разрешен

```
route-map AS600_IN permit 50
```


Логика работы route-map

```
router bgp 1234
  neighbor 26.0.0.6 remote-as 600
  neighbor 26.0.0.6 route-map AS600_IN in
```

190.16.11.0/24



```
route-map AS600_IN permit 10
  match ip address prefix-list AS_1100
    deny 190.16.14.0/24
    permit 190.16.12.0/24
    permit 190.16.10.0/24
```

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

```
route-map AS600_IN deny 20
  match ip address prefix-list AS_1200
    permit 190.16.11.0/24
    deny 0.0.0.0/0 le 32
```

В правиле route-map стоит deny, поэтому маршрут запрещен

```
route-map AS600_IN permit 50
```

Логика работы route-map

```
router bgp 1234
  neighbor 26.0.0.6 remote-as 600
  neighbor 26.0.0.6 route-map AS600_IN in
```

190.16.12.0/24



```
route-map AS600_IN permit 10
  match ip address prefix-list AS_1100
    deny 190.16.14.0/24
    permit 190.16.12.0/24
    permit 190.16.10.0/24
```

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

```
route-map AS600_IN deny 20
  match ip address prefix-list AS_1200
    permit 190.16.11.0/24
    deny 0.0.0.0/0 le 32
```

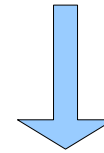
В правиле route-map стоит permit, поэтому маршрут разрешен

```
route-map AS600_IN permit 50
```

Логика работы route-map

```
router bgp 1234
  neighbor 26.0.0.6 remote-as 600
  neighbor 26.0.0.6 route-map AS600_IN in
```

190.16.14.0/24



```
route-map AS600_IN permit 10
  match ip address prefix-list AS_1100
    deny 190.16.14.0/24
    permit 190.16.12.0/24
    permit 190.16.10.0/24
```

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

```
route-map AS600_IN deny 20
  match ip address prefix-list AS_1200
    permit 190.16.11.0/24
    deny 0.0.0.0/0 le 32
```

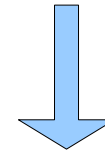
В 20 правиле route-map маршрут попадает на deny. Это НЕ совпадение с 20.

```
route-map AS600_IN permit 50
```

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

Логика работы route-map

```
router bgp 1234
  neighbor 26.0.0.6 remote-as 600
  neighbor 26.0.0.6 route-map AS600_IN in    190.16.55.0/24
```



```
route-map AS600_IN permit 10
  match ip address prefix-list AS_1100
    deny 190.16.14.0/24
    permit 190.16.12.0/24
    permit 190.16.10.0/24
```

```
route-map AS600_IN deny 20
  match ip address prefix-list AS_1200
    permit 190.16.11.0/24
    deny 0.0.0.0/0 le 32
```

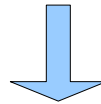
```
route-map AS600_IN permit 50
```

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

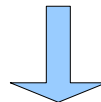
190.16.10.0/24
190.16.11.0/24
190.16.12.0/24
190.16.14.0/24
190.16.55.0/24



```
route-map AS600_IN permit 10
  match ip address prefix-list AS_1100
    deny 190.16.14.0/24
    permit 190.16.12.0/24
    Permit 190.16.10.0/24
```

```
route-map AS600_IN deny 20
  match ip address prefix-list AS_1200
    permit 190.16.11.0/24
    deny 0.0.0.0/0 le 32
```

```
route-map AS600_IN permit 50
```



190.16.10.0/24
190.16.12.0/24
190.16.14.0/24
190.16.55.0/24

Изменение атрибутов BGP с помощью route-map

Route-map

Позволяет управлять атрибутами и фильтровать префиксы

```
route-map ISP_IN permit 10
  match ip address prefix-list AS500 AS600
  set local-preference 120
!
route-map ISP_IN permit 20
  match ip address prefix-list AS700
  match as-path 100
  set local-preference 80
!
route-map ISP_IN permit 30
```

AS500 ИЛИ AS600

AS700 И as-path 100

разрешить остальные

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 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.

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

Примечание: Это не четкое разделение и термины часто используются, как взаимозаменяемые. Для четкости, в этой презентации они используются как описано выше.

Управление входящим и исходящим трафиком

Управление исходящим трафиком:

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

Управление исходящим трафиком

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 15.0.0.5 weight 500
```

Применение через route-map:

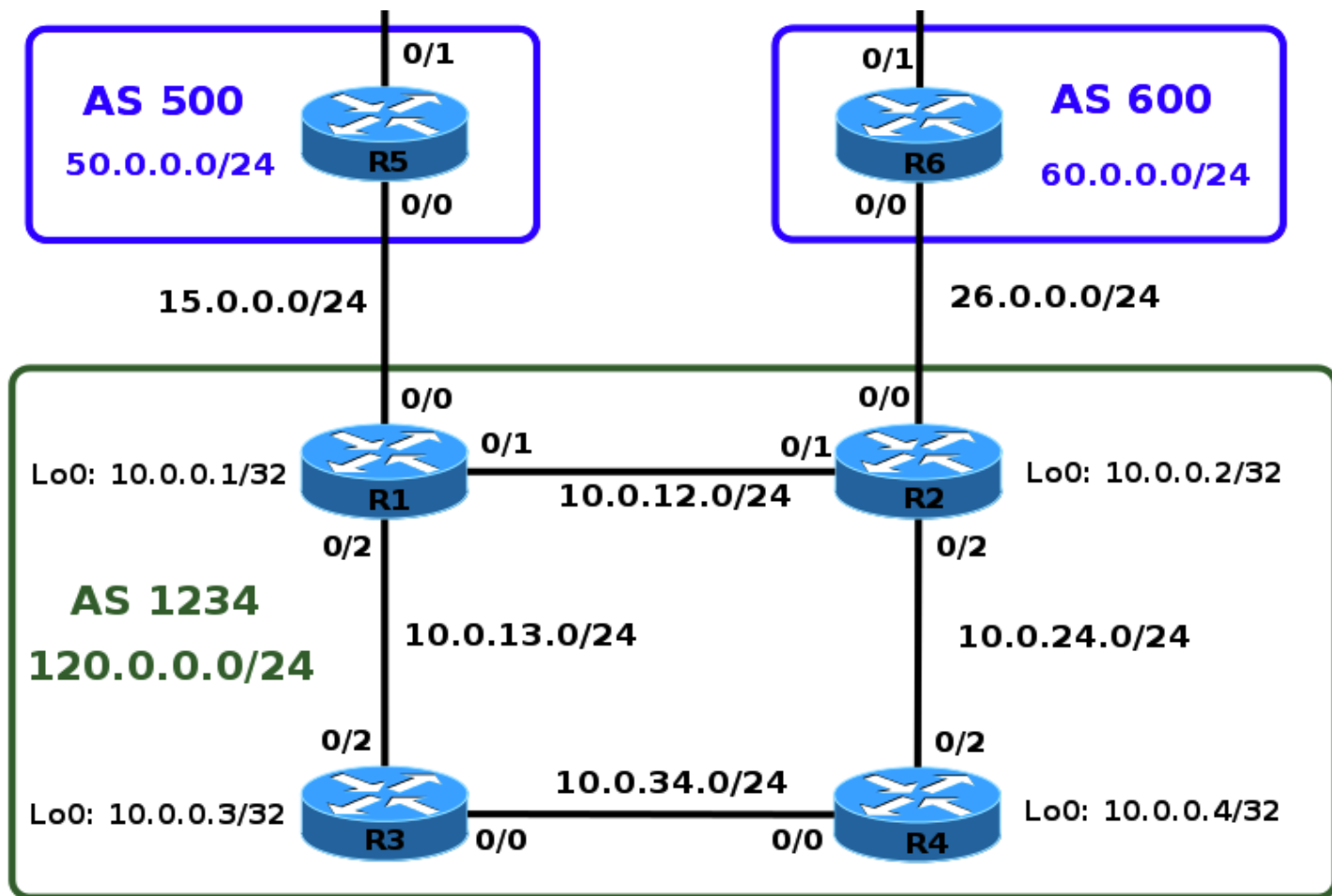
```
route-map SET_WEIGHT permit 10  
    set weight 500
```

```
sh ip bgp route-map SET_WEIGHT
```

```
sh route-map
```

Атрибут weight

```
route-map SET_WEIGHT permit 10
  set weight 500
!
router bgp 1234
  neighbor 15.0.0.5 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 1234
  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 1234
  neighbor 15.0.0.5 remote-as 500
  neighbor 15.0.0.5 route-map AS500_IN in
```

```
route-map AS500_IN permit 10
  match ip address prefix-list AS700
    permit 70.0.0.0/24
    permit 79.0.0.0/24
  set local-preference 200
```

```
route-map AS500_IN deny 20
  match ip address prefix-list AS900
    permit 90.0.0.0/24
```

```
route-map AS500_IN 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 1234  
  maximum-paths 2
```

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

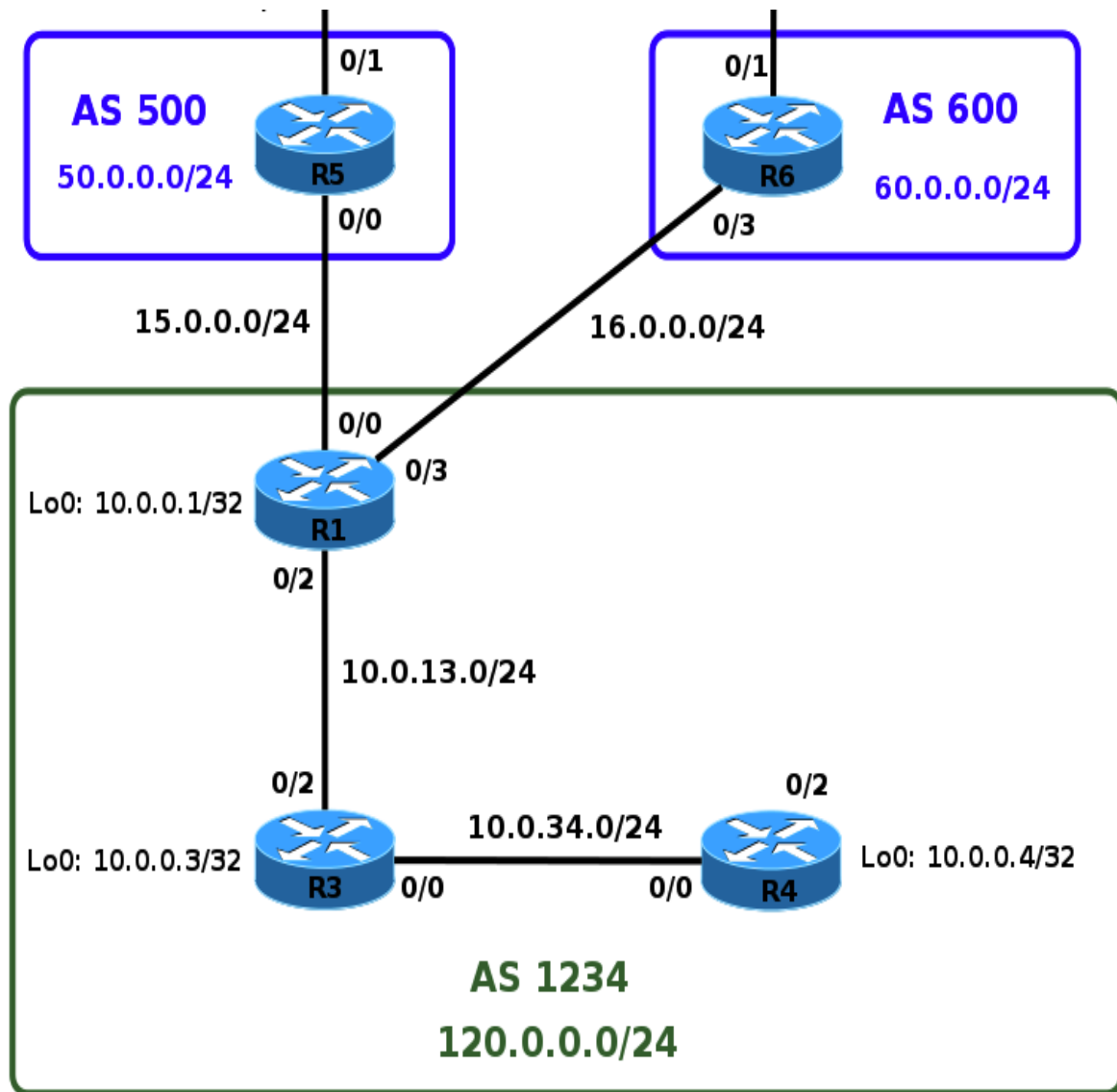
```
router bgp 1234  
  bgp bestpath as-path multipath-relax
```

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

Балансировка исходящего трафика с помощью BGP multipath

R1:

```
router bgp 1234
  bgp bestpath as-path multipath-relax
  neighbor 15.0.0.5 remote-as 500
  neighbor 16.0.0.6 remote-as 600
  maximum-paths 2
```



Балансировка исходящего трафика с помощью BGP multipath

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

```
R1#sh ip bgp 90.0.0.0/24
BGP routing table entry for 90.0.0.0/24, version 26
Paths: (2 available, best #1, table default)
Multipath: eBGP
  Advertised to update-groups:
    4          5
  Refresh Epoch 2
  500 700 900 900
    15.0.0.5 from 15.0.0.5 (50.0.0.5)
      Origin IGP, localpref 100, valid, external, multipath, best
  Refresh Epoch 2
  600 800 1000 900
    16.0.0.6 from 16.0.0.6 (60.0.0.6)
      Origin IGP, localpref 100, valid, external, multipath(oldest)

:
```



```
R1# sh ip route
B          90.0.0.0 [20/0] via 16.0.0.6, 00:03:50
           [20/0] via 15.0.0.5, 00:03:50
```

Балансировка исходящего трафика с помощью BGP multipath

Подробная информация о маршруте в таблице маршрутизации:

```
R1#sh ip route 90.0.0.0 255.255.255.0
Routing entry for 90.0.0.0/24
  Known via "bgp 1234", distance 20, metric 0
  Tag 500, type external
  Last update from 16.0.0.6 00:04:55 ago
  Routing Descriptor Blocks:
    16.0.0.6, from 16.0.0.6, 00:04:55 ago
      Route metric is 0, traffic share count is 1
      AS Hops 4
      Route tag 500
      MPLS label: none
  * 15.0.0.5, from 15.0.0.5, 00:04:55 ago
      Route metric is 0, traffic share count is 1
      AS Hops 4
      Route tag 500
      MPLS label: none
```

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 15.0.0.5 dmzlink-bw
  neighbor 16.0.0.6 dmzlink-bw
```

Балансировка исходящего трафика с учетом пропускной способности каналов

```
R1#sh ip bgp 90.0.0.0/24
BGP routing table entry for 90.0.0.0/24, version 38
Paths: (2 available, best #1, table default)
Multipath: eBGP
  Advertised to update-groups:
    4          5
  Refresh Epoch 6
  500 700 900 900
    15.0.0.5 from 15.0.0.5 (50.0.0.5)
      Origin IGP, localpref 100, valid, external, multipath, best
      DMZ-Link Bw 1250 kbytes
  Refresh Epoch 6
  600 800 1000 900
    16.0.0.6 from 16.0.0.6 (60.0.0.6)
      Origin IGP, localpref 100, valid, external, multipath(oldest)
      DMZ-Link Bw 3750 kbytes
```

Балансировка исходящего трафика с учетом пропускной способности каналов

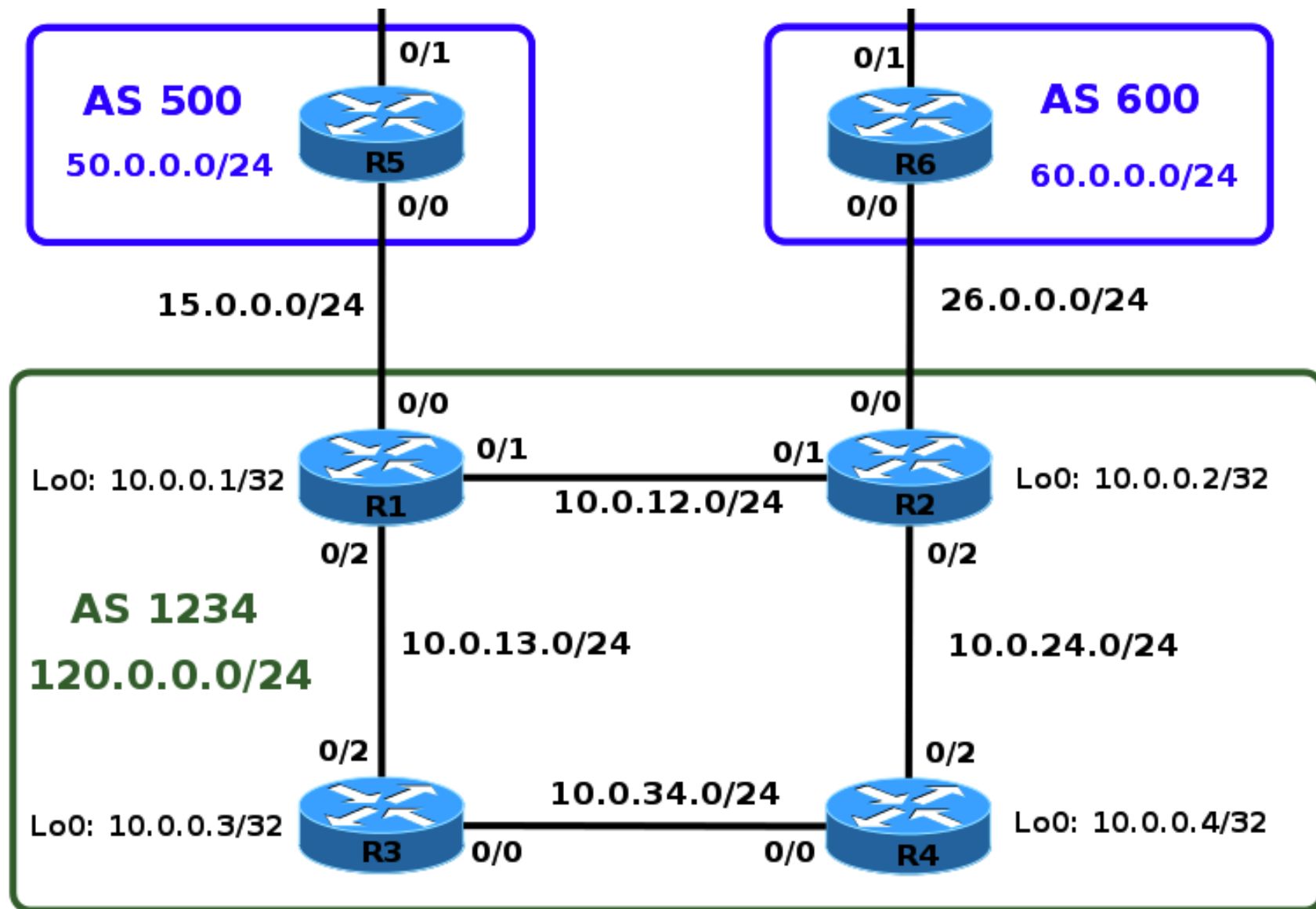
```
R1#sh ip route 90.0.0.0 255.255.255.0
Routing entry for 90.0.0.0/24
  Known via "bgp 1234", distance 20, metric 0
  Tag 500, type external
  Last update from 16.0.0.6 00:01:58 ago
  Routing Descriptor Blocks:
    16.0.0.6, from 16.0.0.6, 00:01:58 ago
      Route metric is 0, traffic share count is 3
      AS Hops 4
      Route tag 500
      MPLS label: none
  * 15.0.0.5, from 15.0.0.5, 00:01:58 ago
    Route metric is 0, traffic share count is 1
    AS Hops 4
    Route tag 500
    MPLS label: none
```

Балансировка исходящего трафика с учетом пропускной способности каналов

```
R1#sh ip cef 90.0.0.0/24 internal
90.0.0.0/24, epoch 0, flags rib only nolabel, rib defined all
labels, RIB[B], refcount 5, per-destination sharing
...
  output chain:
    loadinfo F0B1A5A4, per-session, 2 choices, flags 0003, 5
locks
  flags: Per-session, for-rx-IPv4
  16 hash buckets
    < 0 > IP adj out of Ethernet0/0, addr 15.0.0.5 F26F38B0
    < 1 > IP adj out of Ethernet0/3, addr 16.0.0.6 F26F3780
    < 2 > IP adj out of Ethernet0/0, addr 15.0.0.5 F26F38B0
    < 3 > IP adj out of Ethernet0/3, addr 16.0.0.6 F26F3780
    < 4 > IP adj out of Ethernet0/0, addr 15.0.0.5 F26F38B0
    < 5 > IP adj out of Ethernet0/3, addr 16.0.0.6 F26F3780
    < 6 > IP adj out of Ethernet0/0, addr 15.0.0.5 F26F38B0
    < 7 > IP adj out of Ethernet0/3, addr 16.0.0.6 F26F3780
    < 8 > IP adj out of Ethernet0/3, addr 16.0.0.6 F26F3780
    < 9 > IP adj out of Ethernet0/3, addr 16.0.0.6 F26F3780
    <10 > IP adj out of Ethernet0/3, addr 16.0.0.6 F26F3780
    <11 > IP adj out of Ethernet0/3, addr 16.0.0.6 F26F3780
    <12 > IP adj out of Ethernet0/3, addr 16.0.0.6 F26F3780
    <13 > IP adj out of Ethernet0/3, addr 16.0.0.6 F26F3780
    <14 > IP adj out of Ethernet0/3, addr 16.0.0.6 F26F3780
    <15 > IP adj out of Ethernet0/3, addr 16.0.0.6 F26F3780
```


Балансировка трафика для iBGP

Балансировка исходящего трафика с учетом пропускной способности для IBGP



Балансировка исходящего трафика с учетом пропускной способности для IBGP

R1:

```
router bgp 1234
  bgp bestpath as-path multipath-relax
  bgp dmzlink-bw
  neighbor 10.0.0.2 send-community both
  neighbor 10.0.0.3 send-community both
  neighbor 10.0.0.4 send-community both
  neighbor 15.0.0.5 remote-as 500
  neighbor 15.0.0.5 dmzlink-bw
  maximum-paths 2
```

R2:

```
router bgp 1234
  bgp bestpath as-path multipath-relax
  bgp dmzlink-bw
  neighbor 10.0.0.1 send-community both
  neighbor 10.0.0.3 send-community both
  neighbor 10.0.0.4 send-community both
  neighbor 26.0.0.6 remote-as 600
  neighbor 26.0.0.6 dmzlink-bw
  maximum-paths 2
```

Балансировка исходящего трафика с учетом пропускной способности для IBGP

R3:

```
router bgp 1234
  bgp bestpath as-path multipath-relax
  bgp dmzlink-bw
  neighbor 10.0.0.1 remote-as 1234
  neighbor 10.0.0.1 update-source Loopback0
  neighbor 10.0.0.2 remote-as 1234
  neighbor 10.0.0.2 update-source Loopback0
  neighbor 10.0.0.4 remote-as 1234
  neighbor 10.0.0.4 update-source Loopback0
  maximum-paths ibgp 2
```

Балансировка исходящего трафика с учетом пропускной способности для IBGP

```
R3#sh ip bgp 90.0.0.0/24
BGP routing table entry for 90.0.0.0/24, version 67
Paths: (2 available, best #2, table default)
Multipath: iBGP
  Not advertised to any peer
  Refresh Epoch 14
  600 800 1000 900
    10.0.0.2 (metric 21) from 10.0.0.2 (120.0.0.2)
      Origin IGP, metric 0, localpref 100, valid, internal,
multipath(oldest)
      DMZ-Link Bw 3750 kbytes
  Refresh Epoch 14
  500 700 900 900
    10.0.0.1 (metric 21) from 10.0.0.1 (120.0.0.1)
      Origin IGP, metric 0, localpref 100, valid, internal,
multipath, best
      DMZ-Link Bw 1250 kbytes
```

Балансировка исходящего трафика с учетом пропускной способности для IBGP

```
R3#sh ip route 90.0.0.0 255.255.255.0
Routing entry for 90.0.0.0/24
  Known via "bgp 1234", distance 200, metric 0
  Tag 500, type internal
  Last update from 10.0.0.1 00:10:05 ago
  Routing Descriptor Blocks:
    * 10.0.0.2, from 10.0.0.2, 00:10:05 ago
      Route metric is 0, traffic share count is 3
      AS Hops 4
      Route tag 500
      MPLS label: none
    10.0.0.1, from 10.0.0.1, 00:10:05 ago
      Route metric is 0, traffic share count is 1
      AS Hops 4
      Route tag 500
      MPLS label: none
```

Балансировка исходящего трафика с учетом пропускной способности для IBGP

```
R3#sh ip route 90.0.0.0 255.255.255.0
Routing entry for 90.0.0.0/24
  Known via "bgp 1234", distance 200, metric 0
  Tag 500, type internal
  Last update from 10.0.0.1 00:10:05 ago
  Routing Descriptor Blocks:
    * 10.0.0.2, from 10.0.0.2, 00:10:05 ago
      Route metric is 0, traffic share count is 3
      AS Hops 4
      Route tag 500
      MPLS label: none
    10.0.0.1, from 10.0.0.1, 00:10:05 ago
      Route metric is 0, traffic share count is 1
      AS Hops 4
      Route tag 500
      MPLS label: none
```

Управление входящим трафиком

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 (в лабораторной среде):

```
sh ip bgp
```

Проверить prepend в реальной жизни можно с помощью:

Looking Glass

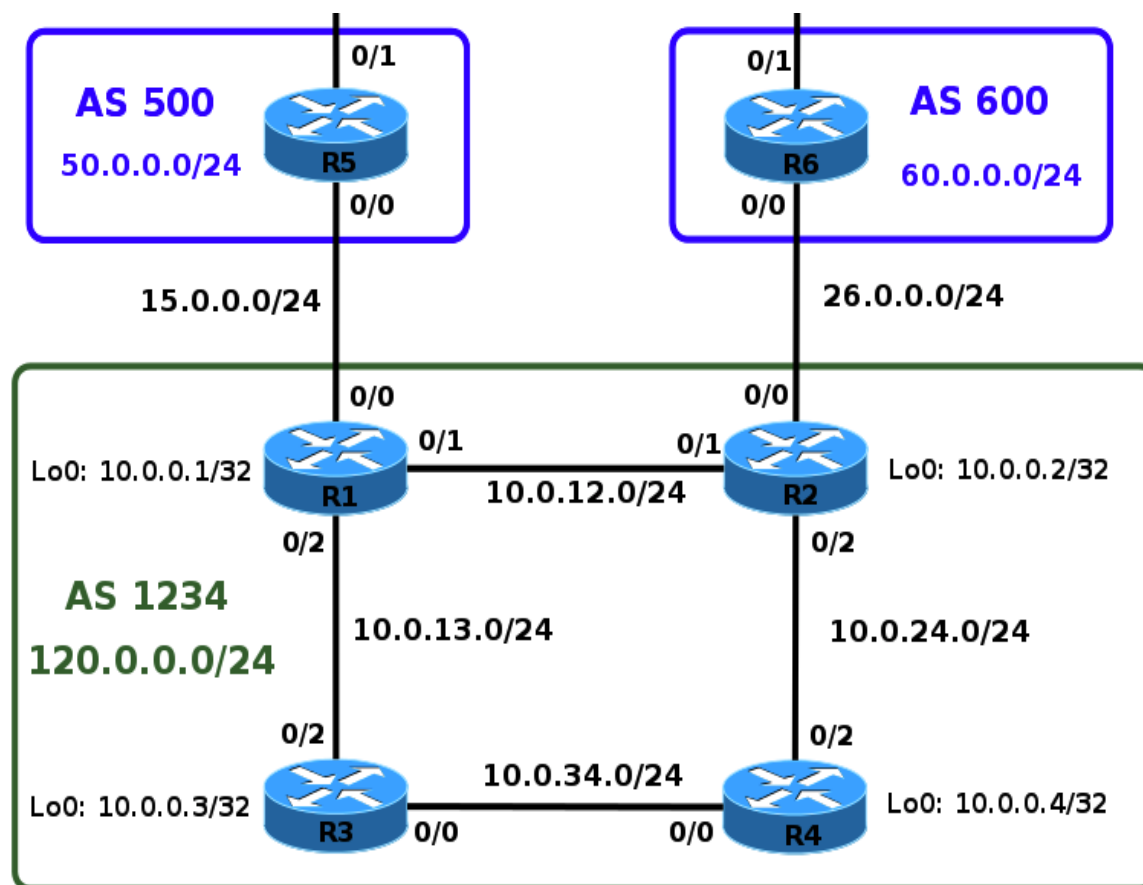
Локально можно проверить настройки route-map:

```
sh route-map
```

Использование AS path prepend

Настройки AS path prepend на R1:

```
route-map AS600_OUT permit 10
  set as-path prepend 1234 1234 1234
!
router bgp 1234
  neighbor 26.0.0.6 route-map AS600_OUT 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 190.16.1.1 remote-as 500
  neighbor 190.16.1.1 route-map MED_OUT out

route-map MED_OUT permit 10
  match ip address prefix-list AS600
  set metric 200

route-map MED_OUT permit 20
  match ip address prefix-list AS700
  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 AS600_OUT permit 10  
  set community 600:200
```

Использование community как критерия через route-map:

```
route-map AS1234_IN permit 10  
  match community LP_200  
  set local-preference 200
```


Атрибут Community

Предопределенные значения communities:

- no-export (0xFFFFFFF01) — маршруты не анонсируются EBGP-соседям, но анонсируются внешним соседям в конфедерации,
- no-advertise (0xFFFFFFF02) — маршруты не должны анонсироваться другим BGP-соседям,
- no-export-subconfed (0xFFFFFFF03) — 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

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

AS prepend

BGP Community String	Description
4323:1	Prepend AS4323 once
4323:2	Prepend AS4323 twice
4323:3	Prepend AS4323 three times

Настройка community со стороны клиента

R2:

```
ip bgp-community new-format
!  
route-map AS600_OUT permit 10  
  match ip address prefix-list LOCAL  
  set community 600:3 600:150  
!  
router bgp 100  
  neighbor 26.0.0.6 remote-as 600  
  neighbor 26.0.0.6 send-community  
  neighbor 26.0.0.6 route-map AS600_OUT 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

```
R6:
ip bgp-community new-format
!
ip community-list standard LP_50 permit 600:50
ip community-list standard LP_150 permit 600:150
ip community-list standard PREPEND_1 permit 600:1
ip community-list standard PREPEND_2 permit 600:2
ip community-list standard PREPEND_3 permit 600:3
!
```

Настройка community со стороны провайдера

R6:

```
route-map CLIENT_IN permit 40
  match community LP_50
  set local-preference 50
  continue 60
route-map CLIENT_IN permit 50
  match community LP_150
  set local-preference 150
  continue 60
route-map CLIENT_IN permit 60
  match community PREPEND_1
  set as-path prepend 600
route-map CLIENT_IN permit 70
  match community PREPEND_2
  set as-path prepend 600 600
route-map CLIENT_IN permit 80
  match community PREPEND_3
  set as-path prepend 600 600 600
route-map CLIENT_IN permit 100
!
router bgp 600
  neighbor 26.0.0.2 route-map CLIENT_IN in
```

Настройка community со стороны провайдера

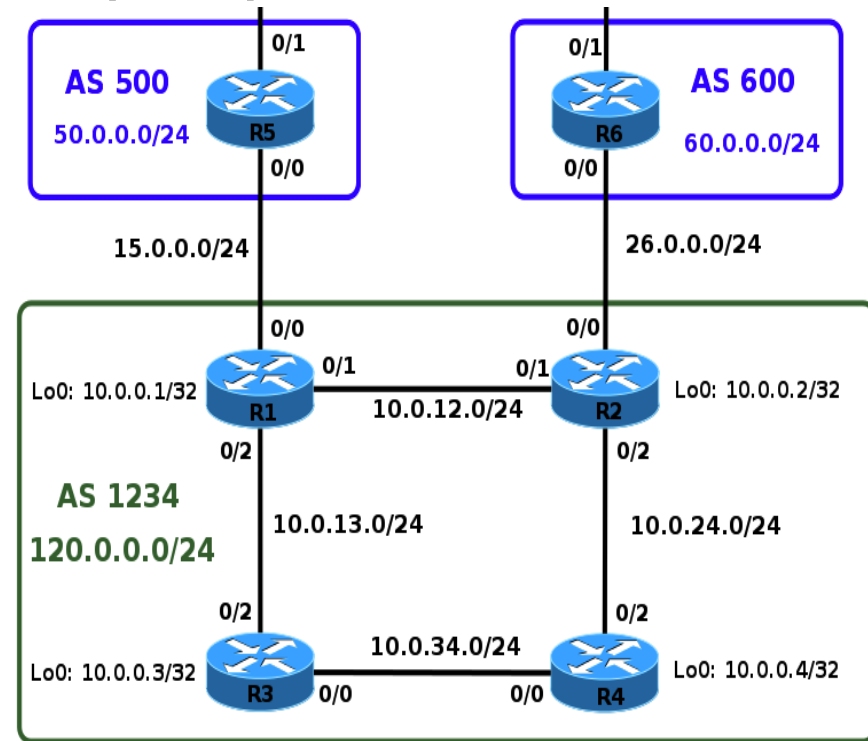
```
R6#sh ip bgp 120.0.0.0/24
BGP routing table entry for 120.0.0.0/24, version 8
Paths: (3 available, best #2, table default)
  Advertised to update-groups:
    1
  Refresh Epoch 2
    800 1000 900 700 500 1234
      68.0.0.8 from 68.0.0.8 (80.0.0.8)
        Origin IGP, localpref 100, valid, external
  Refresh Epoch 1
    600 600 600 1234
      26.0.0.2 from 26.0.0.2 (120.0.0.2)
        Origin IGP, metric 0, localpref 150, valid, external, best
        Community: 600:3 600:150
  Refresh Epoch 1
    1234, (received-only)
      26.0.0.2 from 26.0.0.2 (120.0.0.2)
        Origin IGP, metric 0, localpref 100, valid, external
        Community: 600:3 600:150
```

**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



R1:
router bgp 1234
 network 120.0.0.0 mask 255.255.255.0
 network 120.0.0.0 mask 255.255.255.128
 neighbor 15.0.0.5 prefix-list AS500_OUT out

ip prefix-list AS500_OUT permit 120.0.0.0/24
ip prefix-list AS500_OUT permit 120.0.0.0/25

R2:
router bgp 1234
 network 120.0.0.0 mask 255.255.255.0
 network 120.0.0.128 mask 255.255.255.128
 neighbor 26.0.0.6 prefix-list AS600_OUT out

ip prefix-list AS600_OUT permit 120.0.0.0/24
ip prefix-list AS600_OUT permit 120.0.0.128/25

Управление входящим и исходящим трафиком

Управление входящим и исходящим трафиком

Управление исходящим трафиком:

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

Применение изменений в настройках политик BGP

Применение изменений в настройках политик BGP

В Cisco IOS при изменении политик BGP, например, выполнение AS path prepend, изменения не применяются автоматически к префиксам.

Для того чтобы изменения применились, нужно сбросить сессию. Есть два способа применить обновления:

- Hard reset
- Soft reset

Soft reset может быть в направлениях in и out:

- В направлении out нет никаких проблем с отправкой обновлений, так как локальный маршрутизатор просто заново отправляет свою таблицу BGP соседу
- В направлении in есть два варианта:
 - Сохранять локально на маршрутизаторе префиксы, которые прислал сосед и при необходимости заново локально их пересылать
 - Функционал Route Refresh

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 не помогает

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
```

Route Refresh (Dynamic inbound soft reset)

Обновление маршрутов (команда улучшает механизм inbound soft reset):

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

Маршрутизатор отправляет соседу запрос на повторную отправку всех маршрутов. При этом не происходит разрыва сессии с соседом. По сравнению с inbound soft reset, когда требовалось сохранить все маршруты полученные от соседа, этот метод требует меньшей затраты ресурсов маршрутизатора.

Для использования route refresh, оба маршрутизатора должны поддерживать эту функцию. Информация о поддержке функции анонсируется в сообщениях Open.

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

Группы соседей, шаблоны

BGP peer group, dynamic update peer-group

При настройке соседей в процессе BGP, часто большое количество настроек повторяется для нескольких соседей. Для того чтобы оптимизировать настройку и не повторять одни и те же параметры для каждого соседа, используются группы соседей:

- BGP peer group

Ранее, группы соседей BGP использовались также для оптимизации отправки обновлений:

- Для всех соседей, которые объединены в одну группу, генерируется одно исходящее обновление

Позже появился функционал Dynamic Update Peer-Groups:

- Соседи, для которых настроены одинаковые исходящие политики, автоматически объединяются в группы обновлений

BGP peer group

BGP peer group

Группы позволяют сократить количество настроек, которые относятся к соседям.

Объединение соседей в группы не добавляет новый функционал.

После объединения соседей в группу:

- Параметры можно задавать для группы соседей
- Для группы должна быть одинаковая исходящая политика (outbound policy)
- Входящая политика (inbound policy) для группы может быть различной
- Обновления генерируются один раз для всей группы
- Соседний маршрутизатор может принадлежать только одной группе соседей
- Группа соседей имеет локальное значение, эта настройка никак не передается соседям
- В группе нельзя смешивать iBGP и EBGP соседей.

BGP peer group

Соседи, которые принадлежат группе, всегда наследуют такие параметры группы (настройка этих параметров для конкретного соседа не перебивает настройки сделанные для группы):

- remote-as
- version
- update-source
- out-route-map
- out-filter-list
- out-dist-list
- minimum-advertisement-interval
- next-hop-self

Создание группы соседей:

```
R1(config-router)# neighbor IBGP peer-group
```

Добавить соседа в созданную группу:

```
R1(config-router)# neighbor 10.0.0.1 peer-group IBGP
```

BGP peer group

R1:

```
router bgp 1234
  network 120.0.0.0 mask 255.255.255.0
  neighbor IBGP peer-group
  neighbor IBGP remote-as 1234
  neighbor IBGP update-source Loopback0
  neighbor IBGP next-hop-self
  neighbor IBGP send-community both
  neighbor 10.0.0.2 peer-group IBGP
  neighbor 10.0.0.3 peer-group IBGP
  neighbor 10.0.0.4 peer-group IBGP
  neighbor 15.0.0.5 remote-as 500
```


BGP Dynamic Update Peer-Groups

BGP Dynamic Update Peer-Groups

```
router bgp 1234
 network 120.0.0.0 mask 255.255.255.0
 neighbor 10.0.0.2 remote-as 1234
 neighbor 10.0.0.2 update-source Loopback0
 neighbor 10.0.0.2 next-hop-self
 neighbor 10.0.0.2 send-community both
 neighbor 10.0.0.3 remote-as 1234
 neighbor 10.0.0.3 update-source Loopback0
 neighbor 10.0.0.3 next-hop-self
 neighbor 10.0.0.3 send-community both
 neighbor 10.0.0.4 remote-as 1234
 neighbor 10.0.0.4 update-source Loopback0
 neighbor 10.0.0.4 next-hop-self
 neighbor 10.0.0.4 send-community both
 neighbor 15.0.0.5 remote-as 500
```

BGP Dynamic Update Peer-Groups

```
R1#sh ip bgp update-group
```

```
BGP version 4 update-group 1, external, Address Family: IPv4 Unicast
```

```
BGP Update version : 1/0, messages 0
```

```
Topology: global, highest version: 1, tail marker: 1
```

```
Format state: Current working (OK, last not in list)
```

```
Refresh blocked (not in list, last not in list)
```

```
Update messages formatted 0, replicated 0, current 0, refresh 0, limit 1000
```

```
Number of NLRIs in the update sent: max 0, min 0
```

```
Minimum time between advertisement runs is 30 seconds
```

```
Has 1 member:
```

```
15.0.0.5
```

```
BGP version 4 update-group 2, internal, Address Family: IPv4 Unicast
```

```
BGP Update version : 1/0, messages 0
```

```
NEXT_HOP is always this router for eBGP paths
```

```
Community attribute sent to this neighbor
```

```
Extended-community attribute sent to this neighbor
```

```
Topology: global, highest version: 1, tail marker: 1
```

```
Format state: Current working (OK, last not in list)
```

```
Refresh blocked (not in list, last not in list)
```

```
Update messages formatted 0, replicated 0, current 0, refresh 0, limit 1000
```

```
Number of NLRIs in the update sent: max 0, min 0
```

```
Minimum time between advertisement runs is 0 seconds
```

```
Has 3 members:
```

```
10.0.0.2
```

```
10.0.0.3
```

```
10.0.0.4
```

BGP Dynamic Update Peer-Groups

```
R1#sh ip bgp update-group summary
```

```
Summary for Update-group 1, Address Family IPv4 Unicast
```

```
BGP router identifier 120.0.0.1, local AS number 1234
```

```
BGP table version is 1, main routing table version 1
```

```
7 network entries using 1008 bytes of memory
```

```
14 path entries using 840 bytes of memory
```

```
14/0 BGP path/bestpath attribute entries using 1904 bytes of memory
```

```
12 BGP AS-PATH entries using 400 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
```

```
BGP using 4152 total bytes of memory
```

```
BGP activity 7/0 prefixes, 14/0 paths, scan interval 60 secs
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
15.0.0.5	4	500	13	2	1	0	0	00:00:55	6

```
Summary for Update-group 2, Address Family IPv4 Unicast
```

```
BGP router identifier 120.0.0.1, local AS number 1234
```

```
BGP table version is 1, main routing table version 1
```

```
7 network entries using 1008 bytes of memory
```

```
14 path entries using 840 bytes of memory
```

```
14/0 BGP path/bestpath attribute entries using 1904 bytes of memory
```

```
12 BGP AS-PATH entries using 400 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
```

```
BGP using 4152 total bytes of memory
```

```
BGP activity 7/0 prefixes, 14/0 paths, scan interval 60 secs
```

Neighbor	V	AS	MsgRcvd	MsgSent	TblVer	InQ	OutQ	Up/Down	State/PfxRcd
10.0.0.2	4	1234	11	2	1	0	0	00:00:53	7
10.0.0.3	4	1234	4	3	1	0	0	00:00:53	0
10.0.0.4	4	1234	5	2	1	0	0	00:00:56	0

BGP peer templates

BGP peer templates

B Session Templates:

- allowas-in
- description
- disable-connected-check
- ebgp-multihop
- fall-over
- local-as
- password
- remote-as
- shutdown
- timers
- translate-update
- transport
- ttl-security
- update-source
- version

B Policy Templates:

- advertisement-interval
- allowas-in
- as-override
- capability
- default-originate
- distribute-list
- dmzlink-bw
- filter-list
- maximum-prefix
- next-hop-self
- next-hop-unchanged
- prefix-list
- remove-private-as
- route-map
- route-reflector-client
- send-community
- send-label
- soft-reconfiguration
- soo
- unsuppress-map
- weight

BGP peer templates

```
router bgp 2334
  template peer-policy HQ
    route-reflector-client
    send-community extended
  exit-peer-policy
  !
  template peer-session HQ
    remote-as 2334
    update-source Loopback20
  exit-peer-session
  !
  bgp router-id 21.34.0.1
  no bgp default ipv4-unicast
  neighbor 21.34.0.2 inherit peer-session HQ
  neighbor 21.34.0.2 description HQ-r2
  neighbor 21.34.1.1 inherit peer-session HQ
  neighbor 21.34.1.1 description HQ1-r1
  !
  address-family vpnv4
    neighbor 21.34.0.2 activate
    neighbor 21.34.0.2 inherit peer-policy HQ
    neighbor 21.34.0.2 route-map LP_100 in
    neighbor 21.34.1.1 activate
    neighbor 21.34.1.1 inherit peer-policy HQ
    neighbor 21.34.1.1 route-map LP_200 out
  exit-address-family
```

Масштабируемость iBGP

iBGP-соседи

По умолчанию, все iBGP-соседи внутри автономной системы должны быть соединены полносвязной топологией.

Это требование связано с правилом предотвращения петель в AS:

Обновления, которые были получены от iBGP-соседа, не передаются другим iBGP-соседям.

Два механизма позволяют обойти правило:

- Route reflector
- Confederation

Route reflector

Route reflector

Route reflectors (RR) позволяет:

- избежать необходимости создания полносвязной топологии между всеми iBGP-соседами,
- всем iBGP-соседам выучить все iBGP-маршруты в AS,
- предотвратить образование петель.

При использовании RR для маршрутизаторов в AS определяются такие три роли:

- RR сервер (RR Server, RR)
- Клиент (Client)
- Не клиент (Nonclient)

Только маршрутизатор работающий как RR использует логику отличную от обычного iBGP-маршрутизатора. Другие маршрутизаторы (клиент и не клиент) не изменяют правила работы.

Route reflector

RR сервер (или несколько серверов) и его клиенты формируют один RR кластер (RR cluster). AS в которой используются RR может состоять из:

- кластера с несколькими RR
- нескольких кластеров

Когда используется несколько кластеров, то хотя бы один RR из кластера должен быть соседом с хотя бы одним RR в каждом из других кластеров.

Как правило, все RR являются непосредственно присоединенными соседями и формируют между собой полносвязную топологию.

Предотвращение петель, при работе с RR

Для того чтобы предотвратить образование петель RR используют такие механизмы:

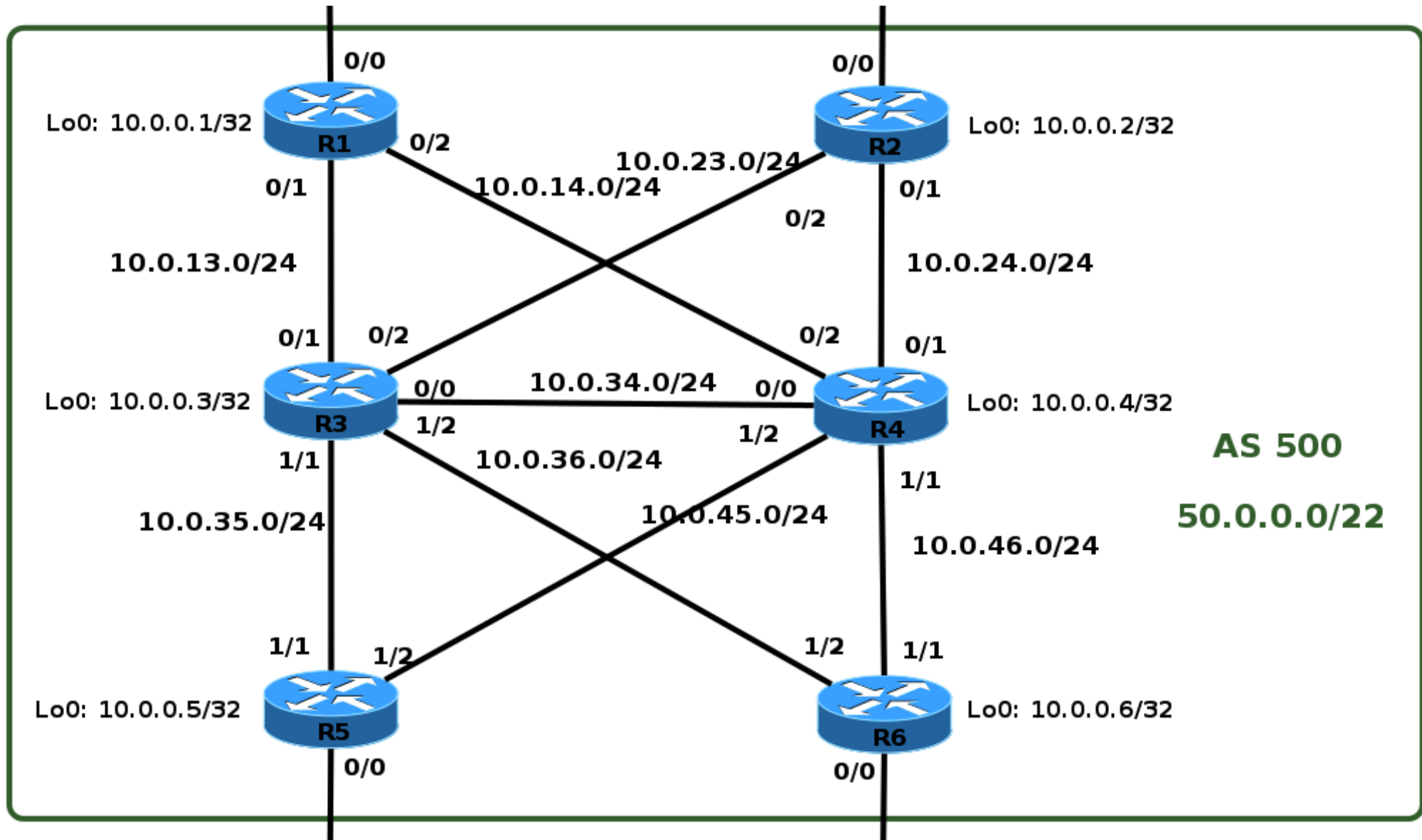
- **CLUSTER_LIST** — RR добавляет идентификатор кластера (cluster ID) в атрибут CLUSTER_LIST прежде чем отправить обновление. При получении обновления RR отбрасывает те префиксы для которых идентификатор его кластера уже указан в атрибуте. Это предотвращает образование петель между кластерами.
- **ORIGINATOR_ID** — этот атрибут указывает RID того маршрутизатора (не RR), который анонсировал маршрут внутри локальной AS. Если маршрутизатор получает обновление в котором указан его RID, то этот маршрут не используется и не передается далее соседям.
- анонсируются только лучшие маршруты — RR передает маршрут далее только если маршрут считается лучшим в его таблице BGP.

Изменение процедуры выбора лучшего маршрута в AS с RR

В тех случаях, когда маршрутизаторы могут получить один и тот же маршрут от RR и от обычного маршрутизатора, правила выбора лучшего маршрута BGP немного изменяются:

- Первые шаги, в которых учитываются атрибуты weight, local preference, origin и MED — остаются без изменений,
- Если все предыдущие параметры одинаковы, то маршруты полученные от EBGP-соседей предпочтительней, чем маршруты полученные от IBGP-соседей,
- Выбрать путь через ближайшего IGP-соседа
- Выбрать путь через соседа с наименьшим BGP router ID.
Если у префикса есть атрибут Originator ID, то вместо router ID сравнивается Originator ID.
- Reflected-маршруты с более коротким cluster-list предпочтительнее (длина cluster-list равна нулю, если маршрут без атрибута cluster-list),
- Выбрать путь через соседа с наименьшим IP-адресом

Настройка route reflector



Настройка route reflector

RR-сервер (R3):

```
router bgp 500
  neighbor IBGP peer-group
  neighbor IBGP remote-as 500
  neighbor IBGP update-source Loopback0
  neighbor IBGP route-reflector-client
  neighbor IBGP next-hop-self
  neighbor IBGP send-community both
  neighbor 10.0.0.1 peer-group IBGP
  neighbor 10.0.0.2 peer-group IBGP
  neighbor 10.0.0.4 peer-group IBGP
  neighbor 10.0.0.5 peer-group IBGP
  neighbor 10.0.0.6 peer-group IBGP
```

RR-сервер (R4):

```
router bgp 500
  neighbor IBGP peer-group
  neighbor IBGP remote-as 500
  neighbor IBGP update-source Loopback0
  neighbor IBGP route-reflector-client
  neighbor IBGP next-hop-self
  neighbor IBGP send-community both
  neighbor 10.0.0.1 peer-group IBGP
  neighbor 10.0.0.2 peer-group IBGP
  neighbor 10.0.0.3 peer-group IBGP
  neighbor 10.0.0.5 peer-group IBGP
  neighbor 10.0.0.6 peer-group IBGP
```


Настройка route reflector

RR-клиент:

R1:

```
router bgp 500
  neighbor 10.0.0.3 remote-as 500
  neighbor 10.0.0.4 remote-as 500
```

RR-клиент:

R2:

```
router bgp 500
  neighbor 10.0.0.3 remote-as 500
  neighbor 10.0.0.4 remote-as 500
```

RR-клиент:

R5:

```
router bgp 500
  neighbor 10.0.0.3 remote-as 500
  neighbor 10.0.0.4 remote-as 500
```

RR-клиент:

R6:

```
router bgp 500
  neighbor 10.0.0.3 remote-as 500
  neighbor 10.0.0.4 remote-as 500
```

Настройка route reflector

```
R1#sh ip bgp 50.0.1.0/24
BGP routing table entry for 50.0.1.0/24, version 26
Paths: (2 available, best #1, table default)
  Not advertised to any peer
  Refresh Epoch 1
  65000
    10.0.0.6 (metric 21) from 10.0.0.3 (50.0.0.3)
      Origin IGP, metric 0, localpref 100, valid, internal, best
      Originator: 50.0.0.6, Cluster list: 50.0.0.3
  Refresh Epoch 6
  65000
    10.0.0.6 (metric 21) from 10.0.0.4 (50.0.0.4)
      Origin IGP, metric 0, localpref 100, valid, internal
      Originator: 50.0.0.6, Cluster list: 50.0.0.4
```

Просмотр информации о route reflector

Routing Protocol is "bgp 1234"

Outgoing update filter list for all interfaces is not set

Incoming update filter list for all interfaces is not set

Route Reflector for address family IPv4 Unicast, 5 clients

IGP synchronization is disabled

Automatic route summarization is disabled

Neighbor(s):

Address	FiltIn	FiltOut	DistIn	DistOut	Weight
RouteMap					
1.1.1.1					
3.3.3.3					
4.4.4.4					
5.5.5.5					
7.7.7.7					

Maximum path: 1

Routing Information Sources:

Gateway	Distance	Last Update
7.7.7.7	200	00:04:26
1.1.1.1	200	00:04:26
5.5.5.5	200	00:07:20

Distance: external 20 internal 200 local 200

Confederation

Confederation

Конфедерации (confederation) это механизм, который позволяет обойти необходимость полной связности внутренних соседей BGP.

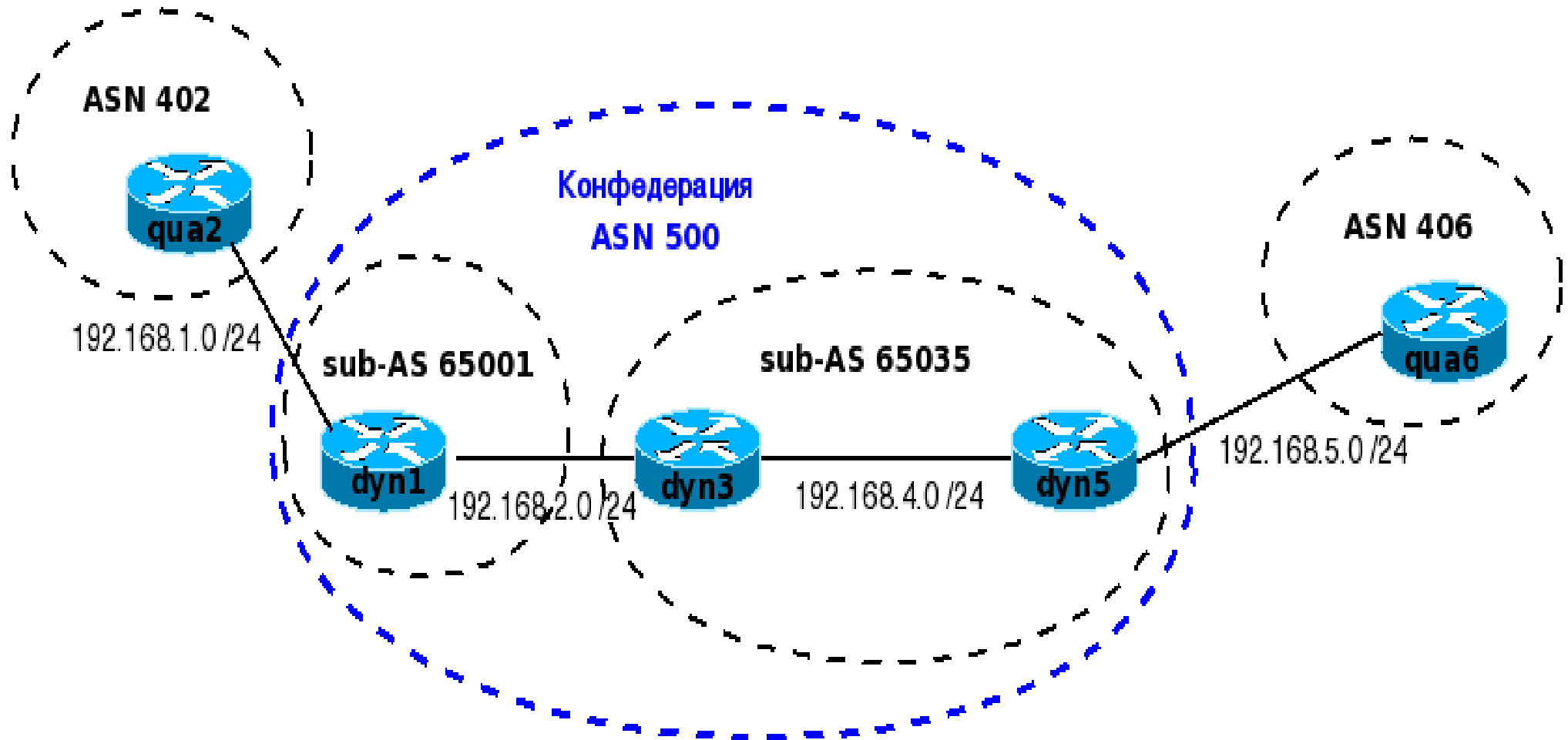
При использовании конфедераций, исходная автономная система разбивается на подавтономные системы (sub-AS), внутри которых соседи должны быть соединены друг с другом в полносвязной топологии.

Маршрутизаторы, которые находятся в одной sub-AS называются **confederation iBGP-соседи**, а маршрутизаторы в разных sub-AS называются **confederation eBGP-соседи**.

Правила работы маршрутизаторов в конфедерации

- iBGP-соседи в конфедерации должны быть соединены в полносвязную топологию. Они, как и обычные iBGP-соседи, не передают iBGP-маршруты друг другу.
- eBGP-соседи в конфедерации:
 - как и eBGP-соседи анонсируют iBGP-маршруты выученные внутри sub-AS конфедерации в другую sub-AS,
 - как и eBGP-соседи по умолчанию используют для пакетов TTL 1 (изменяется `neighbor ebgp-multihop`),
 - во всех остальных случаях работают как обычные iBGP-соседи (например, `next-hop` по умолчанию не изменяется).
- Внутри конфедераций для предотвращения петель используется атрибут AS Path. Маршрутизаторы, которые находятся в конфедерации добавляют в атрибут сегменты `AS_CONFED_SEQ` и `AS_CONFED_SET`.
- Когда маршрутизатор выбирает лучший маршрут на основании атрибута AS Path, номера автономных систем конфедераций не учитываются.
- Когда обновление отправляется маршрутизатору, который не находится в конфедерации, то номера конфедераций удаляются.

Настройка конфедерации



Настройка конфедерации

BGP dyn1:

```
router bgp 65001
  bgp confederation identifier 500
  bgp confederation peers 65035
  neighbor 192.168.1.2 remote-as 402
  neighbor 192.168.2.3 remote-as 65035
```

BGP dyn3:

```
router bgp 65035
  bgp confederation identifier 500
  bgp confederation peers 65001
  neighbor 192.168.2.1 remote-as 65001
  neighbor 192.168.4.5 remote-as 65035
```

BGP dyn5:

```
router bgp 65035
  bgp confederation identifier 500
  bgp confederation peers 65001
  neighbor 192.168.4.3 remote-as 65035
  neighbor 192.168.5.6 remote-as 406
```

BGP qua6:

```
router bgp 406
  neighbor 192.168.5.5 remote-as 500
```


Использование BGP для резервирования Интернет-каналов

**Наташа Самойленко:
nataliya.samoylenko @ gmail.com**