



Networking  
For everyone

# Фильтрация в BGP

---



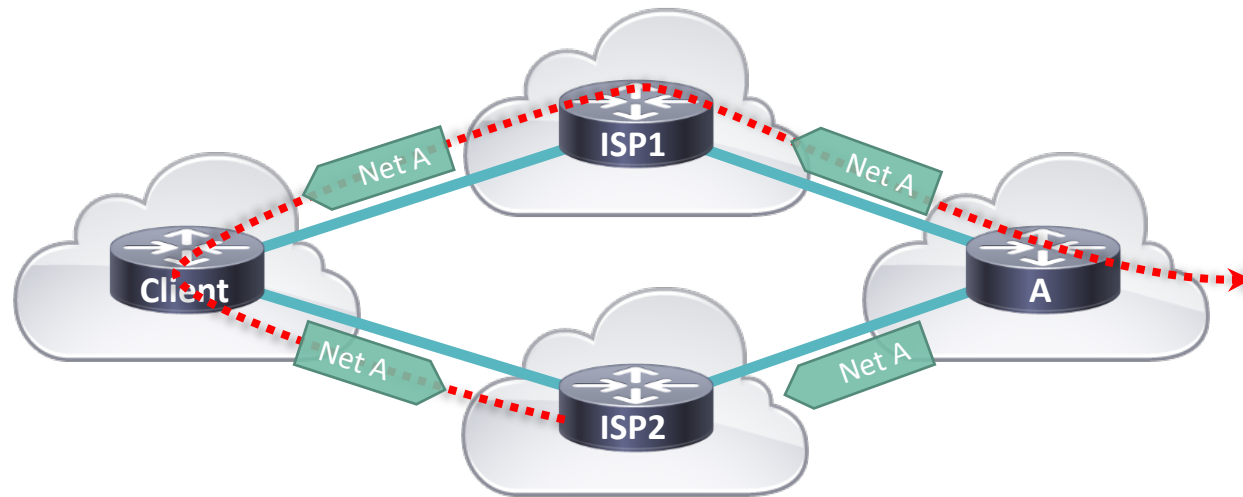
# Фильтрация входящих BGP Update

- Не все маршруты нужно принимать от EBGP-пиров:
  - Некоторые префиксы провайдер в принципе не должен вам анонсировать
  - Ваше устройство, как правило, ограничено по ресурсам
- Что можно принять от провайдера, если вы – предприятие?
  - Только маршрут по умолчанию
  - Маршрут по умолчанию и отдельные specific routes
    - Маршруты клиентов, подключенных к тому же провайдеру
    - Маршруты крупных контент-провайдеров для балансировки трафика
  - Вообще все маршруты в Интернете (>700000 префиксов)
    - Прекрасный способ выстрелить себе в ногу и бездарно потратить деньги на железо



# Фильтрация исходящих BGP Update

- Не все маршруты нужно отправлять EBGP-пирам:
  - По сути, вашему провайдеру не нужно отправлять ничего, кроме ваших PI IP
  - Особенно плохая идея – отправлять вообще все маршруты без разбора





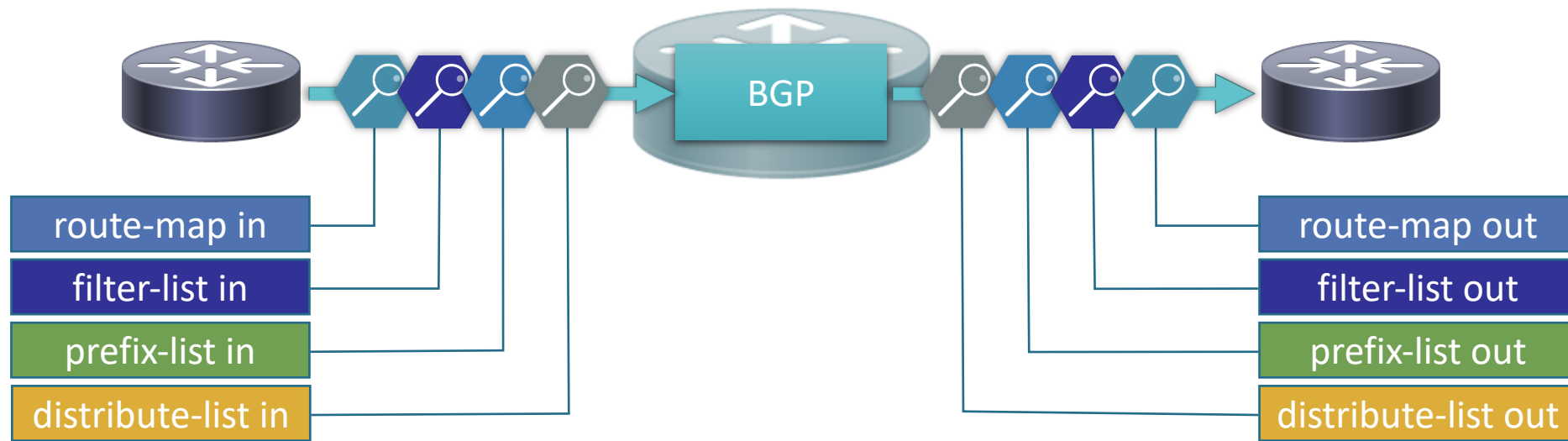
Networking  
For everyone

BGP фильтрация в IOS



# Механизмы фильтрации в Cisco IOS

- Cisco IOS позволяет фильтровать маршруты BGP-пира с помощью:
  - distribute-list – фильтрация на основе ACL
  - prefix-list – фильтрация на основе prefix-list
  - filter-list – фильтрация на основе AS-Path ACL
  - route map – фильтрация на основе чего угодно плюс изменение атрибутов





# Фильтрация с помощью ACL/Prefix List

- Фильтрация исходящих префиксов по ACL:

```
R1(config)#ip access-list standard ACL_PI_ADDRESSES
R1(config-std-nacl)#permit 203.0.113.0 0.0.0.255
R1(config-std-nacl)#router bgp 65001
R1(config-router)#neighbor 10.1.1.1 distribute-list ACL_PI_ADDRESSES out
```

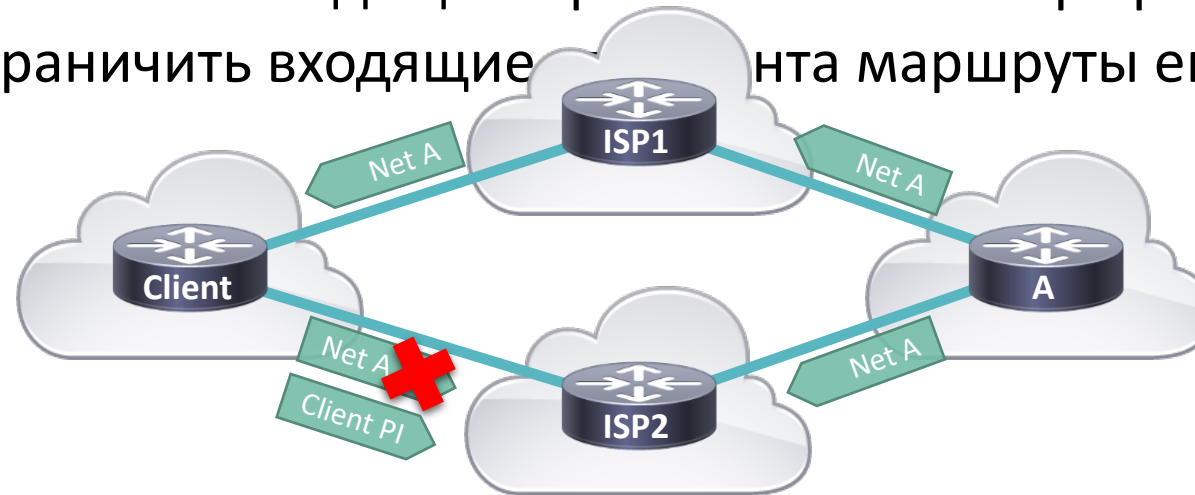
- Фильтрация входящих префиксов по Prefix-list

```
R1(config)#ip prefix-list PL_DEFAULT_AND_8-24 permit 0.0.0.0/0
R1(config)#ip prefix-list PL_DEFAULT_AND_8-24 permit 0.0.0.0/0 ge 8 le 24
R1(config)#router bgp 65001
R1(config-router)#neighbor 10.1.1.1 prefix-list PL_DEFAULT_AND_8-24 in
```



# Фильтрация с помощью AS-Path ACL

- AS-Path ACL используют фильтрацию на основе AS-Path
  - Строковое представление AS-Path анализируется регулярным выражением
  - AS-Path ACE состоят из вердикта и шаблона регулярного выражения
    - Вердикт выносится при соответствии части AS-Path шаблону регулярного выражения
- Удобные сценарии использования AS-Path
  - Компании: ограничить исходящие Update своими PI-префиксами
  - Провайдеру: ограничить входящие маршруты его PI-префиксами





# Символы регулярного выражения

Символы	Что попадает
Строка из цифр	Указанная подстрока
строка1 строка2	Любая из указанных подстрок
[диапазон]	Любой символ из диапазона
.	Любой символ, включая пробел
^	Начало строки
\$	Конец строки
_	Любой разделитель, включая пробел, начало и конец строки
(строка)	Группировка символов в выражение
* ? +	"Ноль или более", "ноль или один", "один или более"
\1	Значение первого выражения





# Примеры совпадения

Regexp	AS-Path	Match
31	27 31 23 317 223 2316	27 <b>31</b> 23 <b>317</b> 223 2 <b>316</b>
21 31	213 317 2316 31 12	<b>213 317 2316 31</b> 12
[1-3][45]	214 317 35 567	<b>214</b> 317 <b>35</b> 567
1.3	321 333 66 123	<b>321 333 66 123</b>
^12	123 123 123	<b>123</b> 123 123
31\$	131 31 310	no match
_31_	131 31 10 31	131 <b>31</b> 10 <b>31</b>
(213 218)_31	213 317 1218 316 31	<b>213 317 1218 316</b> 31
_23(_78)?_45_	213 23 45 345	213 <b>23 45</b> 345
_23(_78)+_45_	213 23 78 78 78 45 345	213 <b>23 78 78 78 45</b> 345



# Некоторые удобные регулярки

Regexp	Смысл
_100_	AS-Path содержит AS 100
^100\$	Маршрут импортирован в напрямую подключенной AS 100
_100\$	Маршрут импортирован в BGP в AS 100
^100_	Маршрут получен из AS 100
^\$	Маршрут импортирован в BGP в локальной AS
^[0-9]+\$	Маршрут получен из внешней AS
^([0-9]+)(_1)*_	Соседняя AS выполнила AS Prepending
.*	Любой AS-Path



# Фильтрация с помощью AS-Path ACL

- Фильтрация исходящих префиксов:

```
R1(config)#ip as-path access-list 1 permit $^  
R1(config)#router bgp 65001  
R1(config-router)#neighbor 10.1.1.1 filter-list 1 out
```

- Вероятный AS-Path ACL со стороны провайдера:

```
ISP1(config)#ip as-path access-list 1 permit $65001^  
ISP1(config)#router bgp 65500  
ISP1(config-router)#neighbor 10.1.1.2 filter-list 1 in
```



Networking  
For everyone

BGP фильтрация в IOS-  
XR



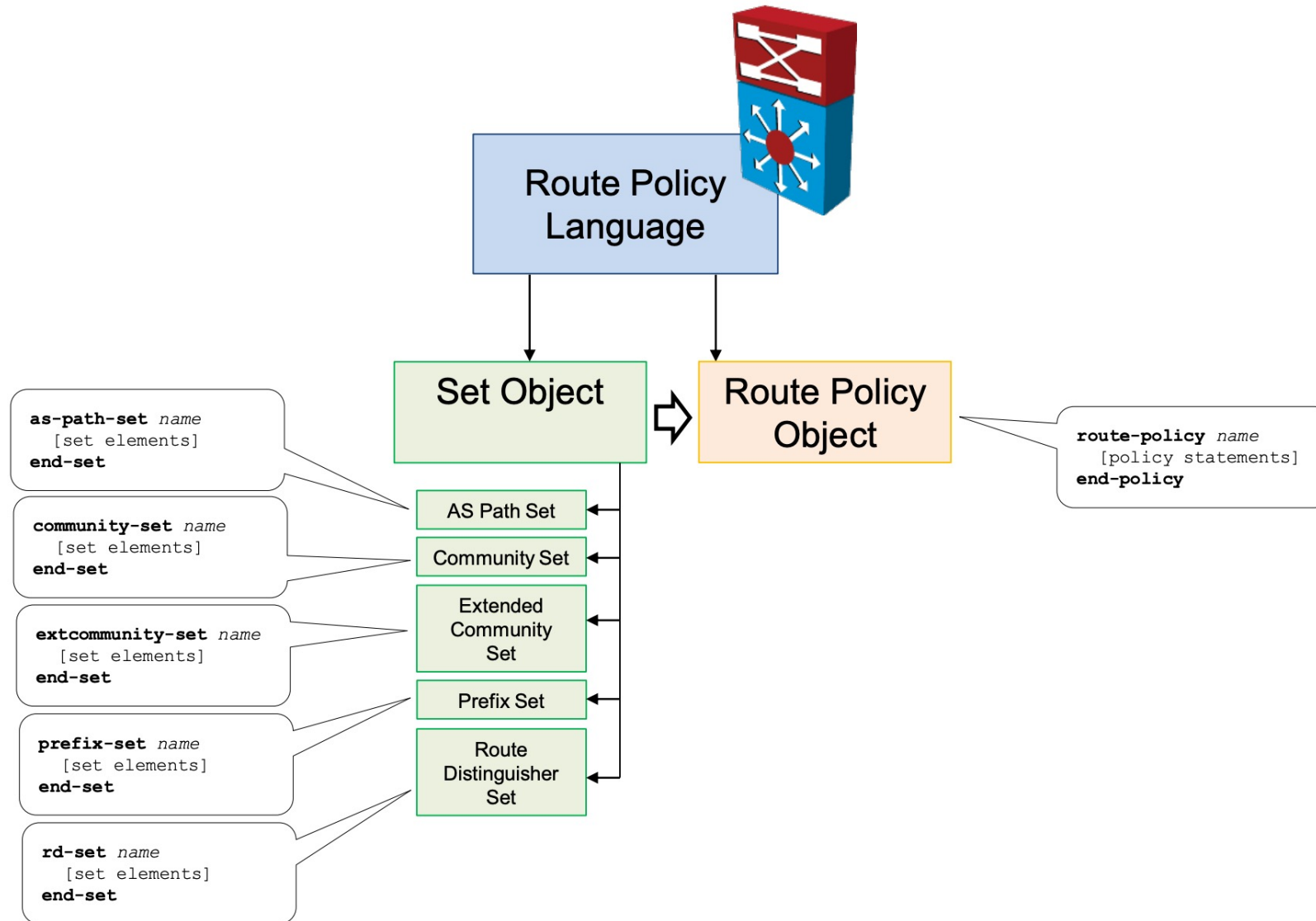
# Routing Policy Language

- IOS XR использует Routing Policy Language
- RPL всегда требуется для eBGP
  - В простейшем случае – политика “pass”

# Элементы RPL



Networking  
For everyone





# Элементы RPL

- К каждому Policy Object можно применить одно из нескольких действий:

- Pass

- Отловленный префикс будет пропущен без изменений

- Set

- К префиксу будут применены действия, указанные в ключевых словах

- Done

- Отловленный префикс будет пропущен без изменений

- Drop

- Отловленный префикс будет отброшен
    - В конце политики всегда стоит implicit drop



# Пример RPL

- Попробуем вместе разобраться, что же здесь сказано?

```
route-policy myPolicy
  if med eq 50 and destination in (10.0.0.0/8) then
    set local-preference 50
  else
    pass
  endif
end-policy
```

```
prefix-set siteBprefixes
  2001:db8:b:10::/64,
  2001:db8:b:11::/64
end-set
!
route-policy SiteA-in-R1
  if destination in siteBprefixes then
    set local-preference 150
  else
    set local-preference 200
  endif
end-policy
```





# Применение RPL в BGP

- С т.з BGP RPL применяются на конкретного соседа
  - Одна политика в каждом направлении In/Out

```
router bgp 64500
neighbor 10.1.1.1
remote-as 65000
address-family ipv4 unicast
route-policy myPolicy_in in
route-policy myPolicy_out out
```



Networking  
For everyone