

BGP Roles

Alexander Azimov

Qrator Labs

BGP Policy is made by...

- Local Pref
- Prepends
- RPSL
- RPKI
- Communities

And other XX config options

Ha! Easy! Love flexibility!

But is it simple?

140 RFC have already **been** written!

388 RFC drafts

Numerous BCPs...

KISS is not for BGP?

It could be quite simple...

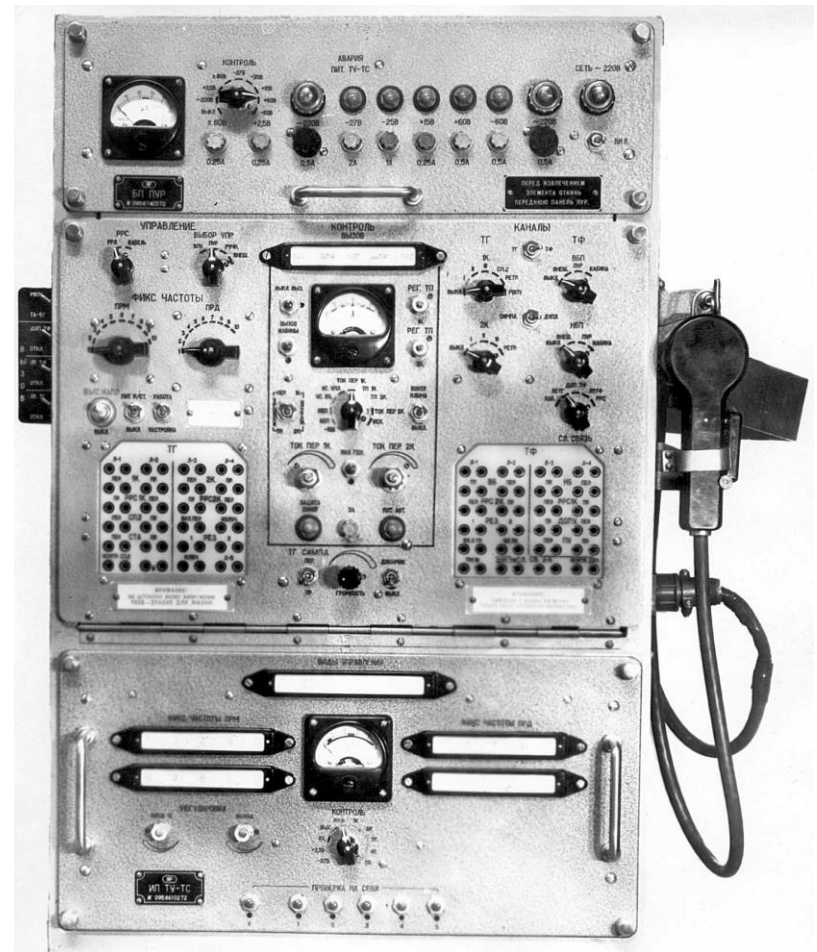
```
protocol bgp IAMOPERATOR {  
    local as MY_AS;  
    neighbor X.X.X.X as AS_PROVIDER;  
}
```

The Great Gap

Expectation

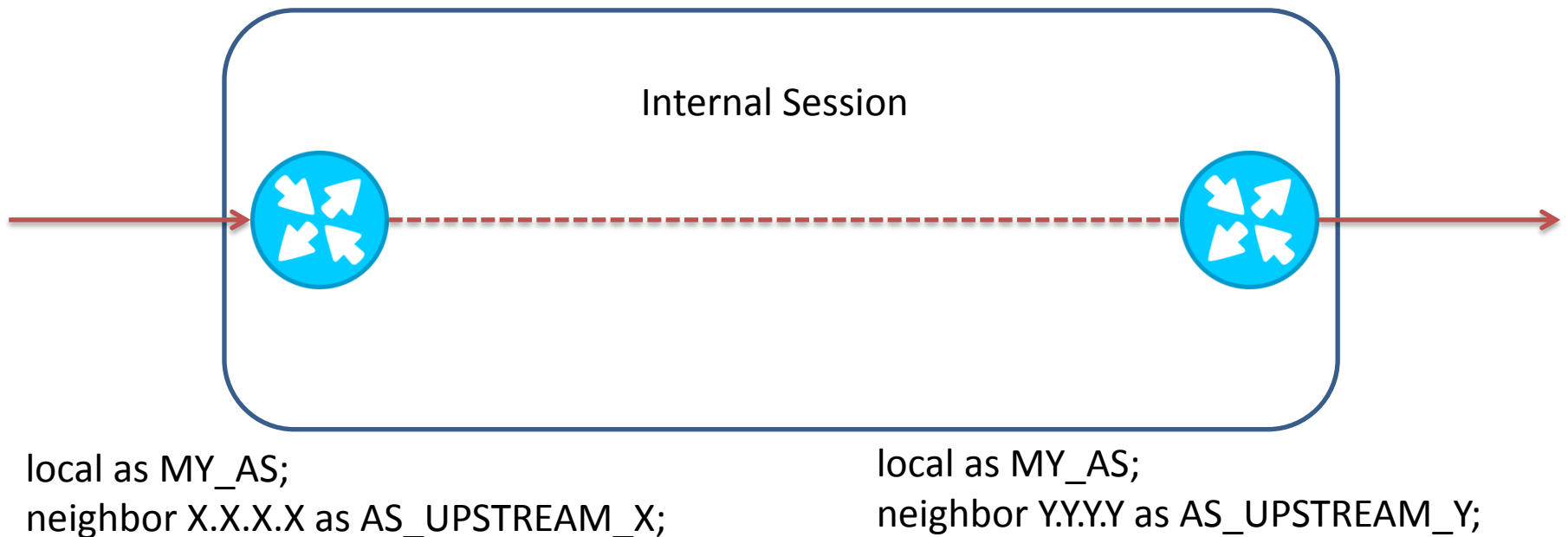


BGP Configuration

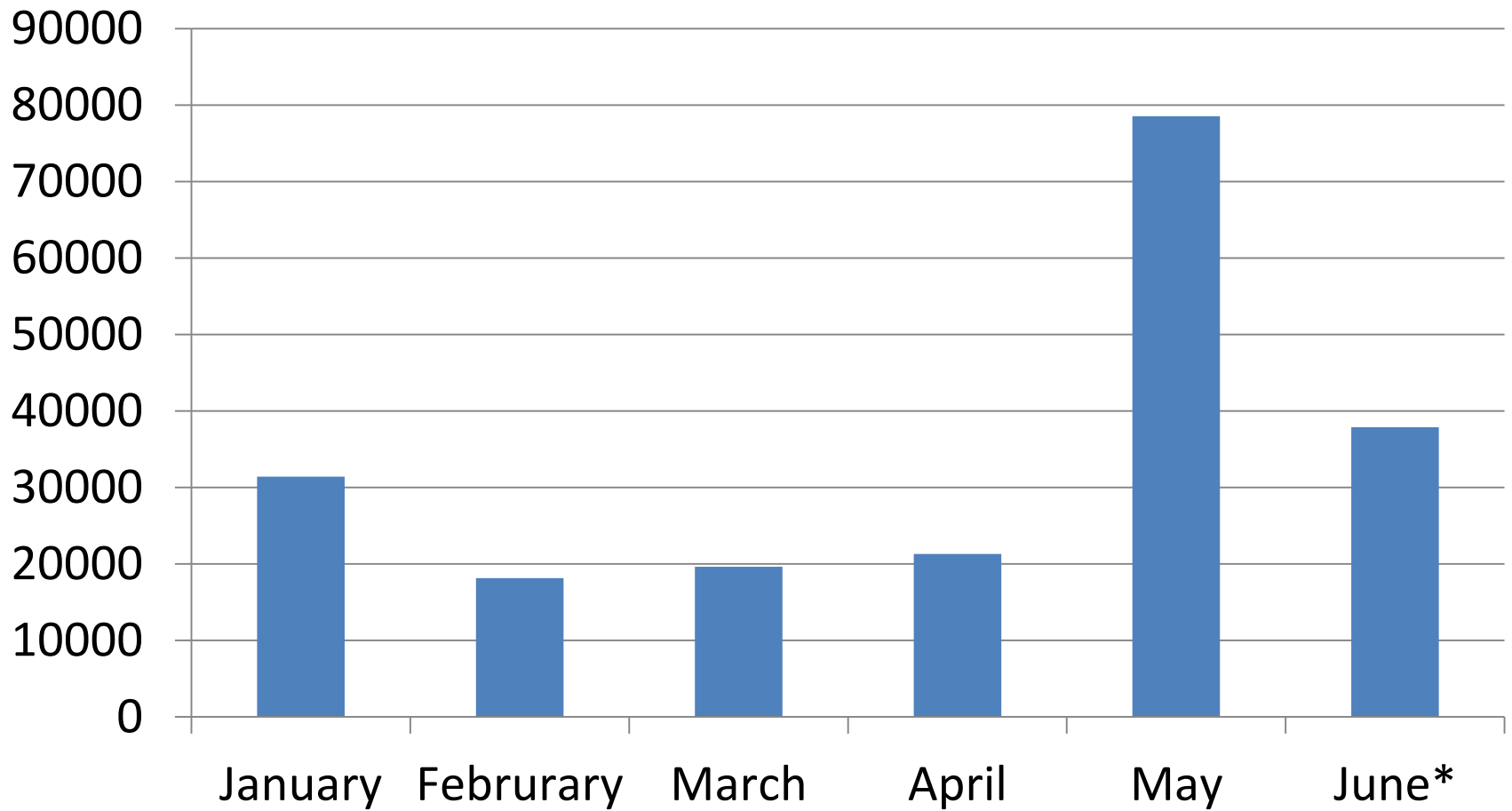


A Simple Result

A simple route leak!



Route Leak: stats

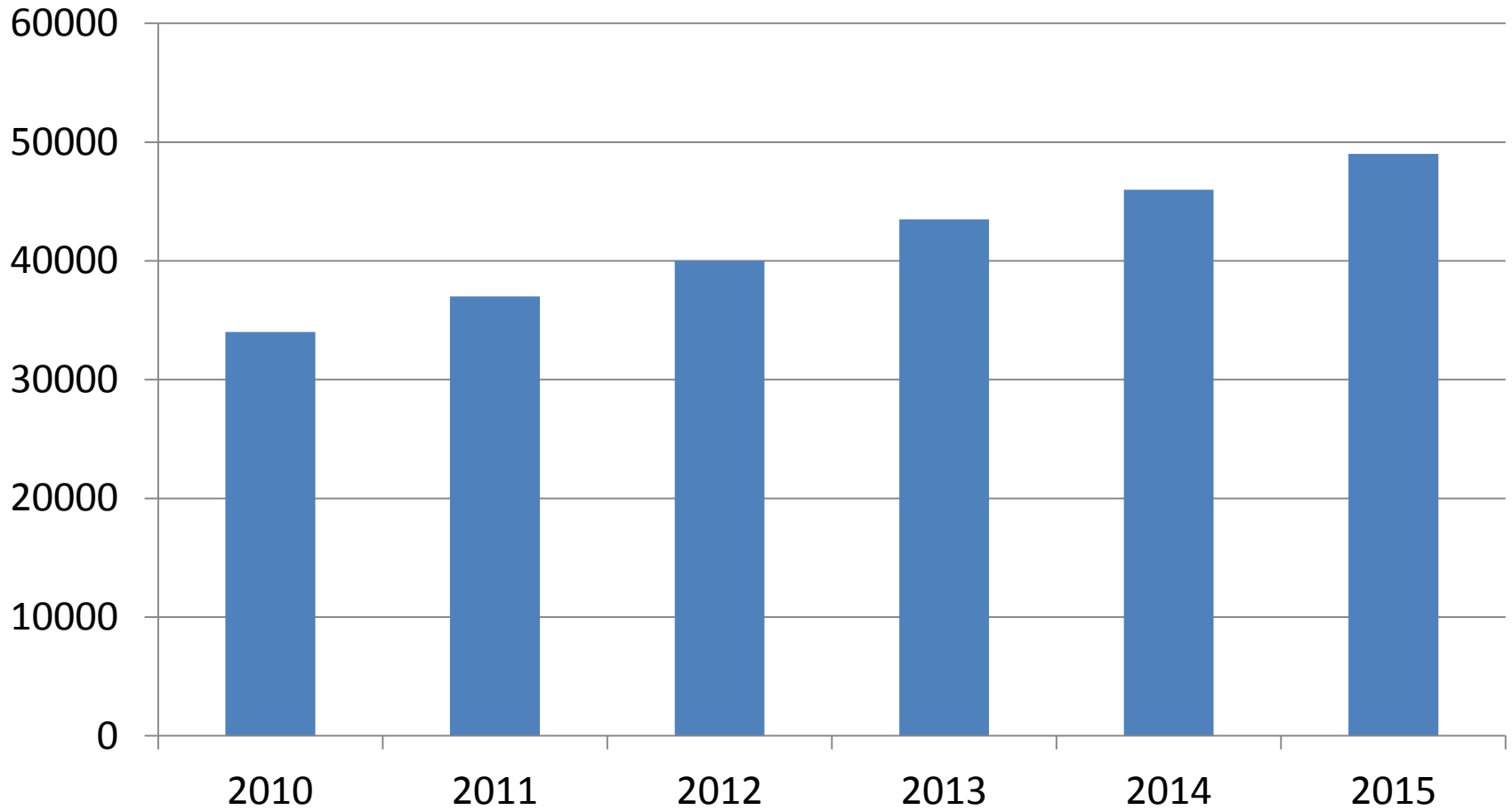


Who is the *leaker*?



About ~1000 leakers affect... Everybody!

Newcomers



Imagine yourself on a highway...



Option 1: Regulation



bgpbusters!
Are you afraid of leaks?

Option 2: A new BGP extension

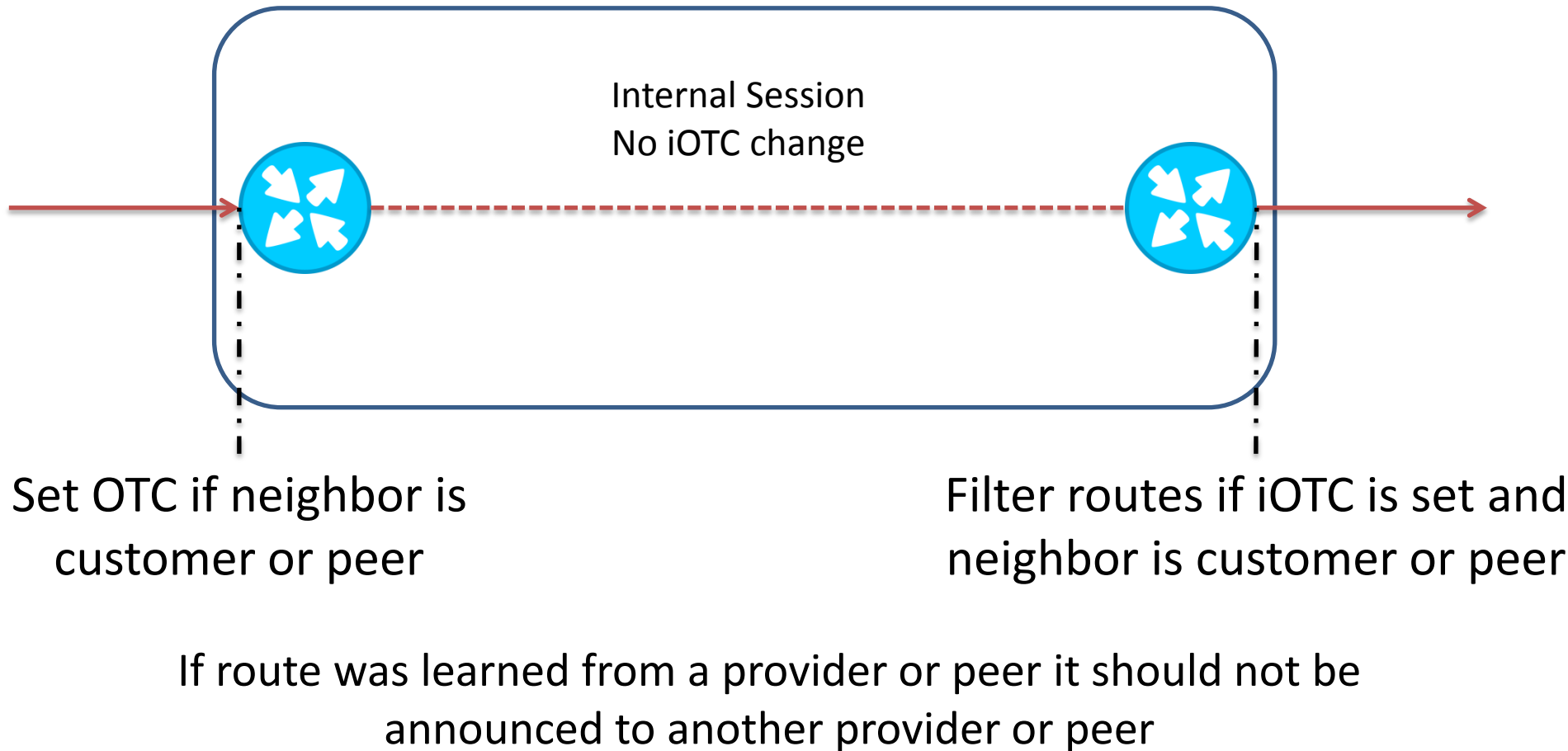


Avoiding Route Leaks

Optional non-transit attribute – Internal Only To Customer Attribute (iOTC):

- Flag is not set – announce in all directions
- Flag is set – announce **only** to internal and customer links

Preventing Route Leaks

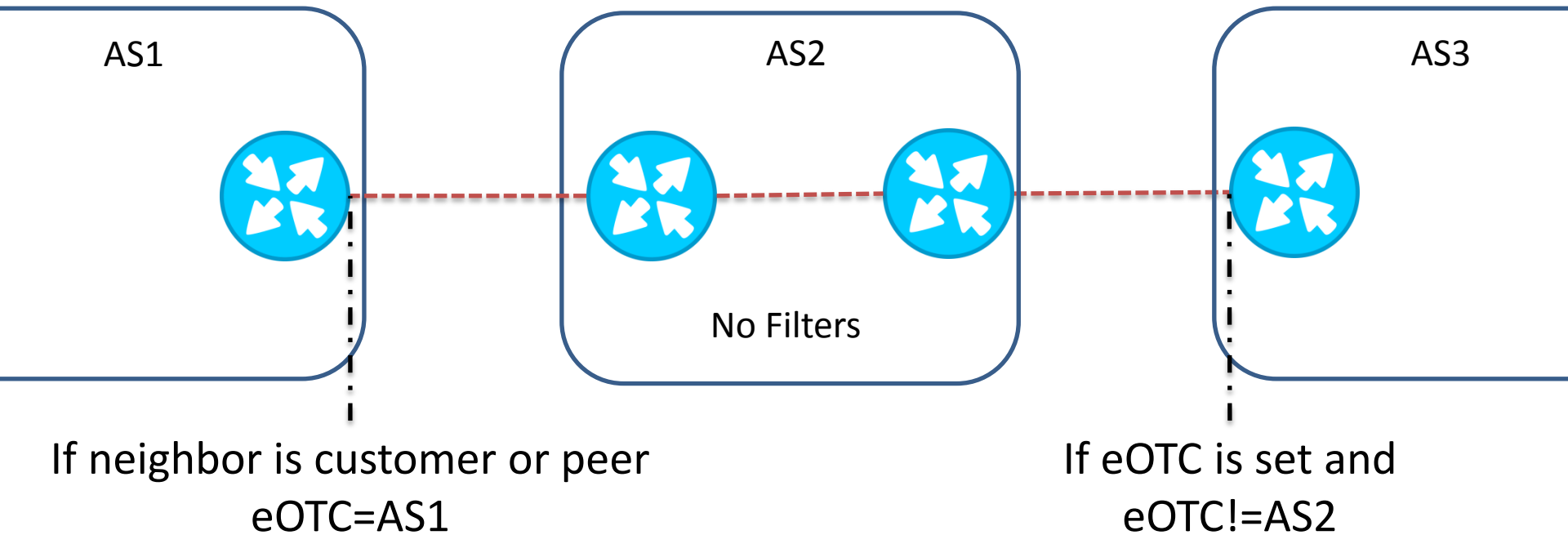


Detecting Route Leaks

Optional transit attribute – External Only To Customer Attribute (eOTC):

- Attribute is not set – no info
- Attribute is set and equals to neighbor AS – ok
- Otherwise – route leak

Detecting Route Leaks



If route was learned from a customer or peer and eOTC is set
and eOTC != neighbor AS then **route was leaked**

What should we do with Route Leak?



Security Considerations: eOTC

Mistake/violation in eOTC flag



It could have significantly impact on route propagation

Local Preference Values

Preset values:

- From customer (+\$) links: 300;
- From peer (0\$) links: 200;
- From provider (-\$) links: 100;
- Route Leaks (-\$\$\$) links: 0.

Route Leak Mitigation

- iOTC – route leak prevention
- eOTC – route leak detection

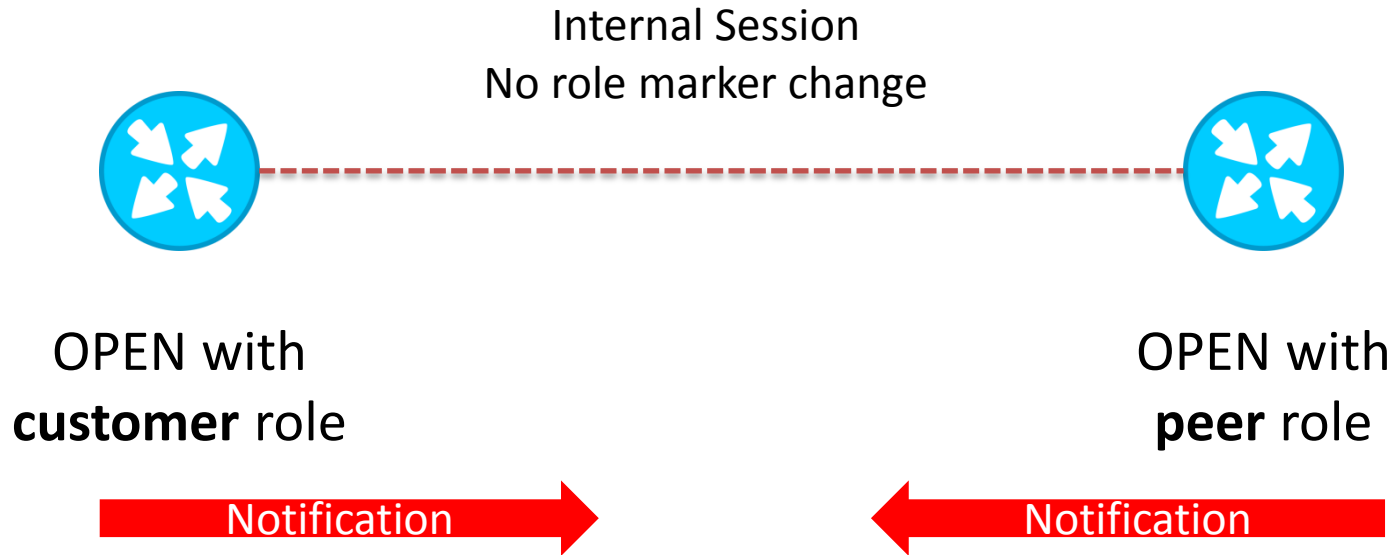
BINGO?

Meet The Neighbor Role

4 Roles: customer, provider, peer, internal

```
mitradir@ubuntu:~/workspace/hll/bird$ ./bird -p -c bird.conf
bird: bird.conf, line 11: Role must be set for each BGP protocol
mitradir@ubuntu:~/workspace/hll/bird$ █
```

Meet The Neighbor Role



3 pairs of non-conflict roles:

1. Peer <---> Peer
2. Customer <---> Provider
3. Internal <---> Internal

Strict Mode



Notification if the role is not set in OPEN from the neighbor

Route Leak Mitigation

- iOTC – route leak prevention
- eOTC – route leak detection
- Roles controlling both attributes
- Roles controlled by neighbors

BINGO!

A Simple Config

```
protocol bgp IAMOPERATOR {  
    local as MY_AS;  
    neighbor X.X.X.X as AS_PROVIDER;  
    role client;  
}
```

Benefits

Backward compatibility

- Unknown optional non-transit attributes are just ignored
- Unknown capabilities should be just ignored!

Route leak extinction:

- No **mistake** leaks

Roles:

- Opportunity to control neighbor configuration
- *And a set of other applications*

Future Work

- Roles as AS boundaries
 - Keep non-transit attributes between internal BGP sessions
 - external $>$ internal in BGP decision process
- Roles as preset Local Preference Values
- Roles in anti-spoofing

Useful Links

Overview of protocol change:

radar.qrator.net/tools/simple-bgp/

Fork of BIRD routing daemon:

github.com/QRatorLabs/bird/

IETF draft:

tools.ietf.org/html/draft-ymbk-idr-bgp-open-policy-00

Alexander Azimov

<aa@qrator.net>