

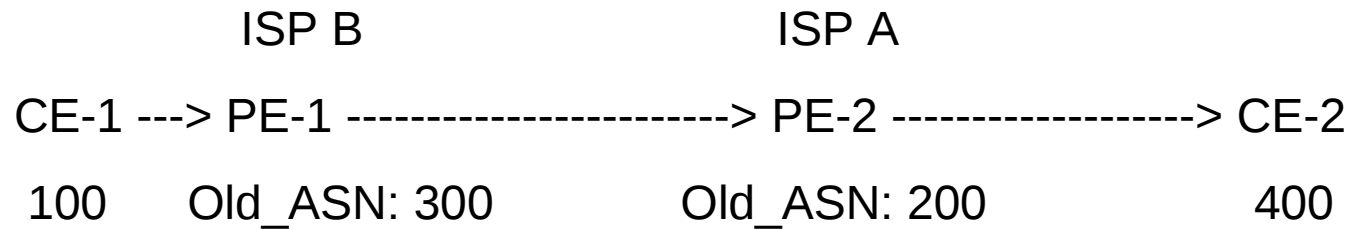
Migratory ASs

(my own interpretation of
draft-ga-idr-as-migration &
draft-george-sidr-as-migration)

Sandra Murphy

sandra.murphy@sparta.com

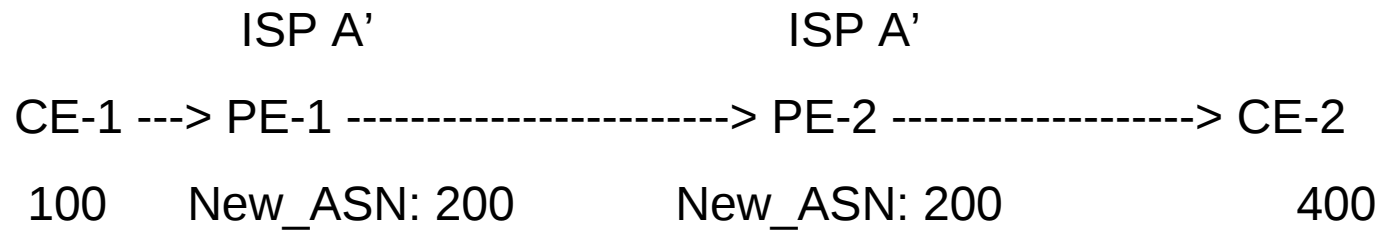
Before Merger



AS_PATHs in UPDATES

100 → 300,100 → 200,300,100 →

Reconfig Before Migration

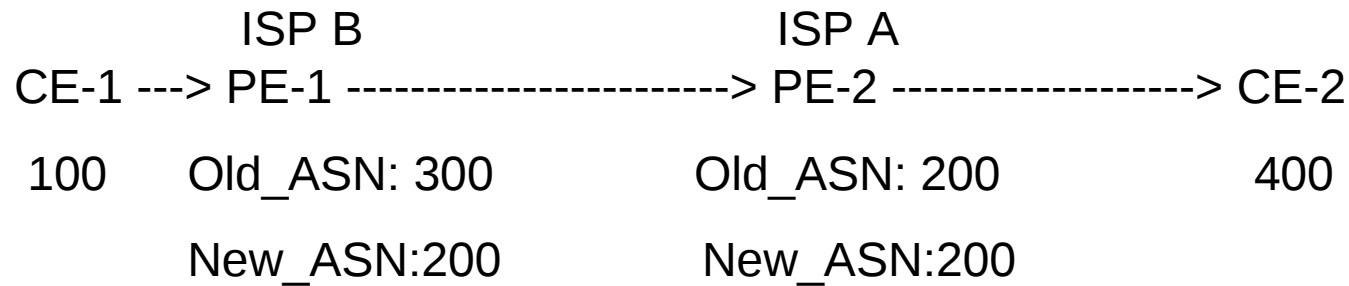


AS_PATHs in UPDATES

100 → 100 → 200,100 →

But this requires reconfiguration of CE-1 with the new ASN for PE-1

Migration

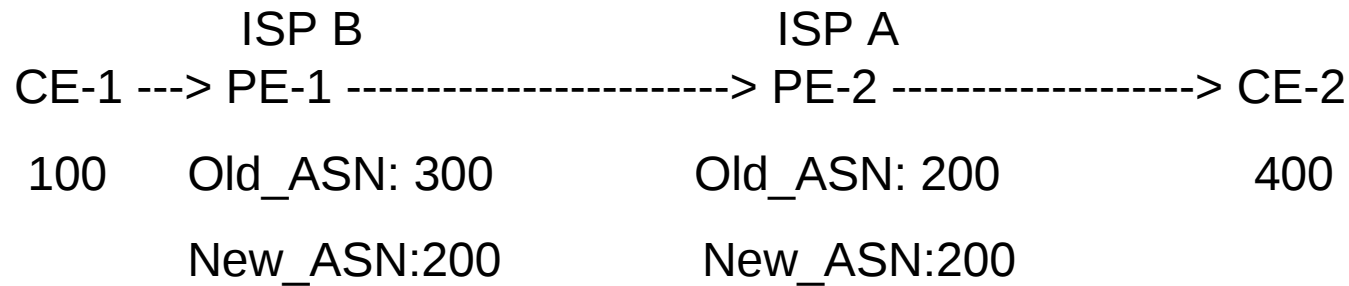


AS_PATHs in UPDATES

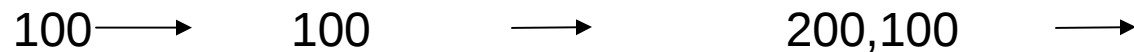
100 → 300,100 → 200,300,100 →

Use Local_AS configuration option
CE-1 does not need to be upgraded
But 300 gets inserted into AS_PATH

Migration

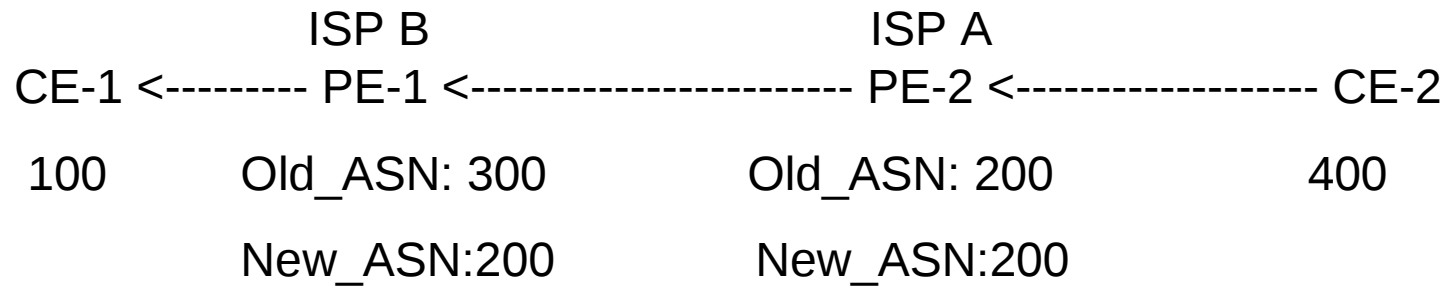


AS_PATHs in UPDATES



Use Local_AS and no-prepend configuration options
CE-1 does not need to be upgraded
300 does not get inserted into AS_PATH

Migration

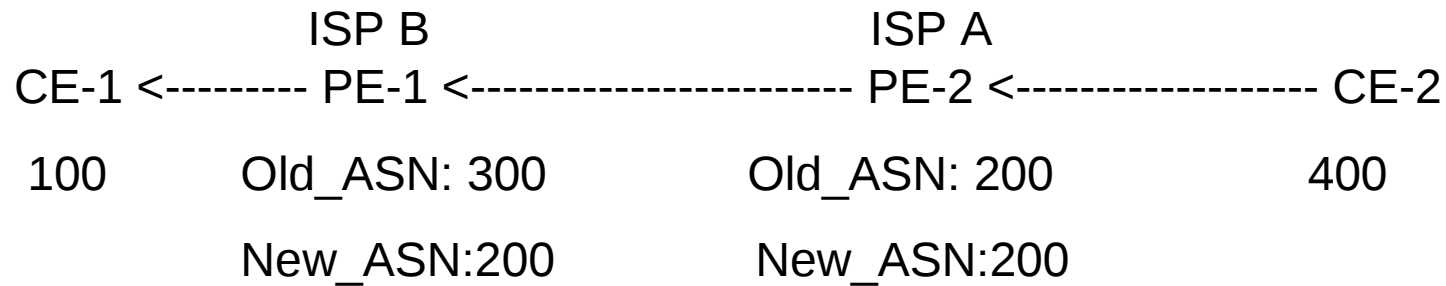


AS_PATHs in UPDATES

← 300,200,400 ← 400 ← 400

Use Local_AS and no-prepend configuration options
CE-1 does not need to be upgraded
300 gets inserted into AS_PATH toward CE-1

Migration



AS_PATHs in UPDATES

← 200,400 ← 400 ← 400

Use Local_AS and no-prepend and replace-AS configuration options
CE-1 does not need to be upgraded
300 does not get inserted into AS_PATH toward CE-1

How to Do That in BGPSEC

Notation

In the following, the notation means:

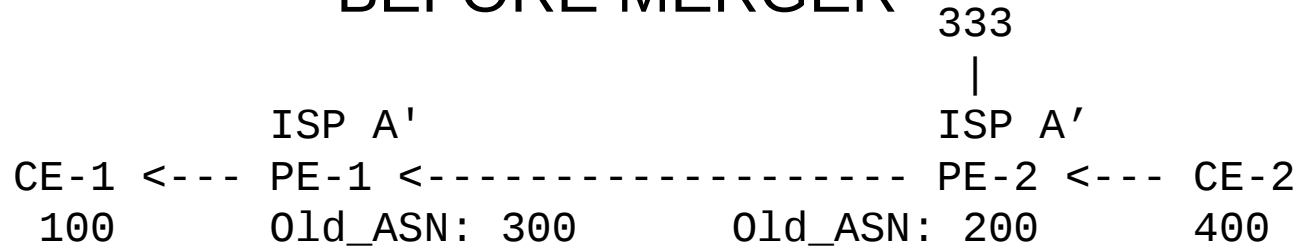
for the origin BGPSEC signature attribute:

sig(originprefix, <target>, AS of signer, pcount) key

for intermediate BGPSEC signature attributes:

sig(<target>, AS of signer, pcount) key

BEFORE MERGER



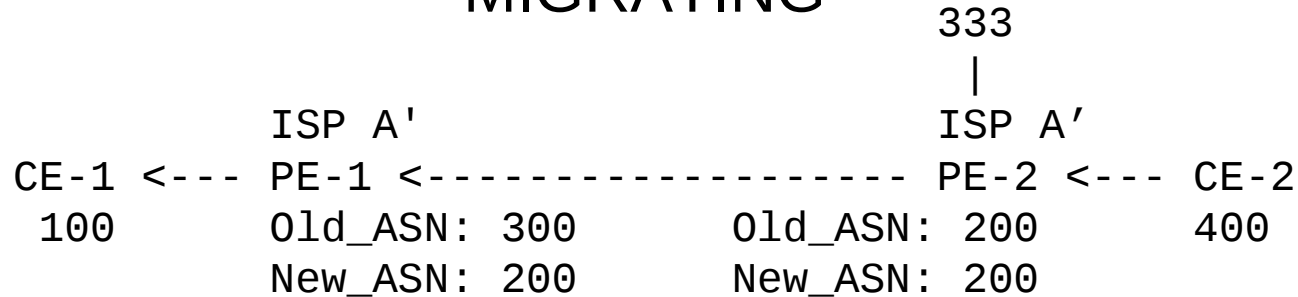
CE-2 to PE-2: sig(origin(N), <200>, 400, pcount=1)K_400-CE2 [sig1]
 AS_PATH=(400)
 length=sum(pcount)=1

PE-2 to 333: sig(<333>, 200, sig1, pcount=1)K_200-PE2 [sig2]
 sig(origin(N), <200>, 400, pcount=1)K_400-CE2 [sig1]
 AS_PATH=(200, 400)
 length=sum(pcount)=2

PE-2 to PE-1: sig(<300>, (200), sig1, pcount=1)K_200-PE2 [sig3]
 sig(origin(N), <200>, (400), pcount=1)K_400-CE2 [sig1]
 AS_PATH=(200, 400)
 length=sum(pcount)=2

PE-1 to CE-1: sig(<100>, 300, sig3, pcount=1)K_300-PE1 [sig4]
 sig(<300>, 200, sig1, pcount=1)K_200-PE2 [sig3]
 sig(origin(N), <200>, 400, pcount=1)K_400-CE2 [sig1]
 AS_PATH = 300, 200, 400
 length=sum(pcount)=3

MIGRATING



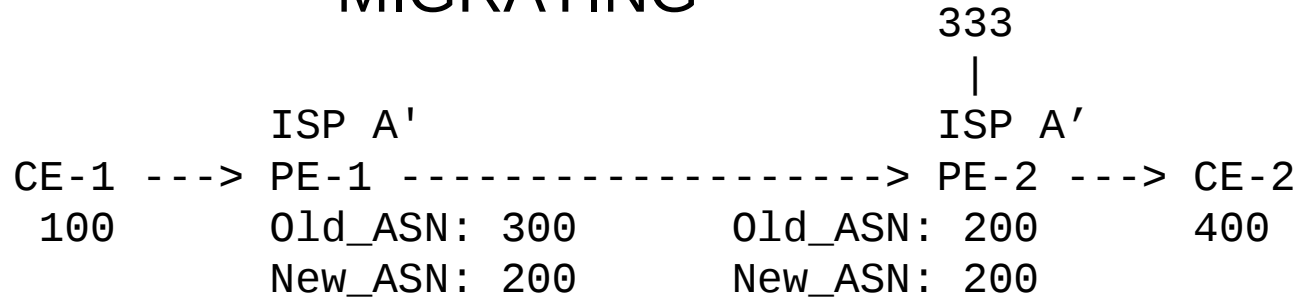
CE-2 to PE-2: sig(origin(N), <200>, 400, pcount=1)K_400-CE2 [sig11]
 AS_PATH=(400)
 length=sum(pcount)=1

PE-2 to 333: sig(<333>, 200, sig11, pcount=1, sig11)K_200-PE2 [sig12]
 sig(origin(N), <200>, 400, pcount=1)K_400-CE2 [sig11]
 AS_PATH=(200, 400)
 length=sum(pcount)=2

PE-2 to PE-1: sig11

PE-1 to CE-1: sig(<100>, 300, pcount=1, sig12)K_300-PE1 [sig13]
 sig(<300>, 200, pcount=0, sig11)K_200-PE2 [sig12]
 sig(origin(N), <200>, 400, pcount=1)K_400-CE2 [sig11]
 AS_PATH=(300, 400)
 length=sum(pcount)=2 (length is NOT 3)

MIGRATING

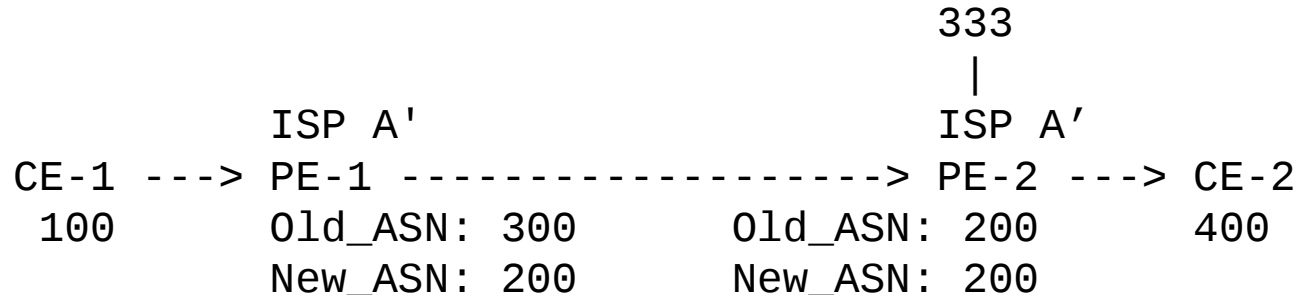


```

PE-1 to CE-1:  sig(<100>,300,pcount=1,sig12)K_300-PE1 [sig13]
                sig(<300>,200,pcount=0,sig11)K_200-PE2 [sig12]
                sig(origin(N),<200>,400,pcount=1)K_400-CE2 [sig11]
                AS_PATH=(300,400)
                length=sum(pcount)=2 (length is NOT 3)
  
```

Note that PE-1 is adding two signatures to the list.
 It is as if PE-1 had an AS hop internally. NOTE: **“AS IF”**
 PE-1 adds sig12 and sig13.
 sig12 is added as if PE-1 is AS200 and adds the pcount=0.
 sig13 is added as if PE-1 is AS300. So “AS300” authorizes
 “AS200” to add the pcount=0.
 Note that the neighbor of the AS that adds pcount=0 is the one
 authorizing – further ASs can not check the authorization.
 (For original route server motivation, neighbor **is** authority.)

MIGRATING in Other Direction – Alternative 1

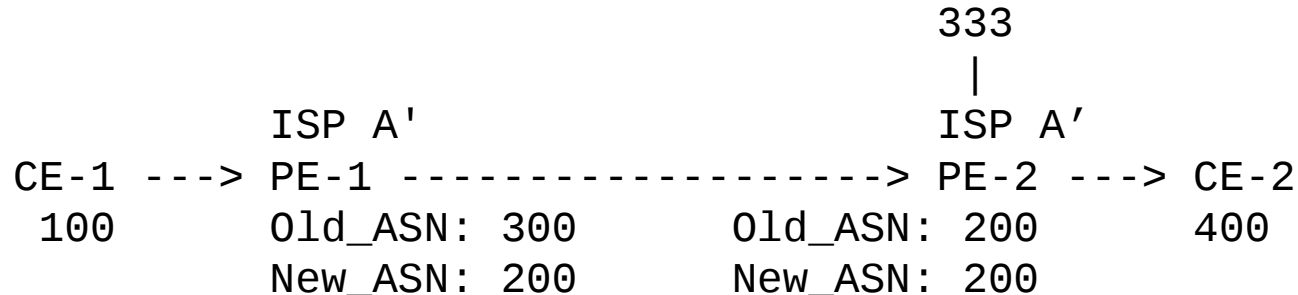


CE-1 to PE-1: sig(origin(N2), <300>, 100, pcount=1)K_100-CE1 [sig21]
 AS_PATH=(100)
 length=sum(pcount)=1

PE-1 to PE-2: sig(<200>, 300, pcount=0, sig21)K_300-PE1 [sig22]
 sig(origin(N2), <300>, 100, pcount=1)K_100-CE1 [sig21]
 AS_PATH=(300, 100)
 length=sum(pcount)=1

PE-2 to CE-2: sig(<400>, 200, pcount=1, sig22)K_200-PE2 [sig23]
 sig(<200>, 300, pcount=0, sig21)K_300-PE1 [sig22]
 sig(origin(N2), <300>, 100, pcount=1)K_100-CE1 [sig21]
 AS_PATH=(200, 300, 100)
 length=sum(pcount)=2

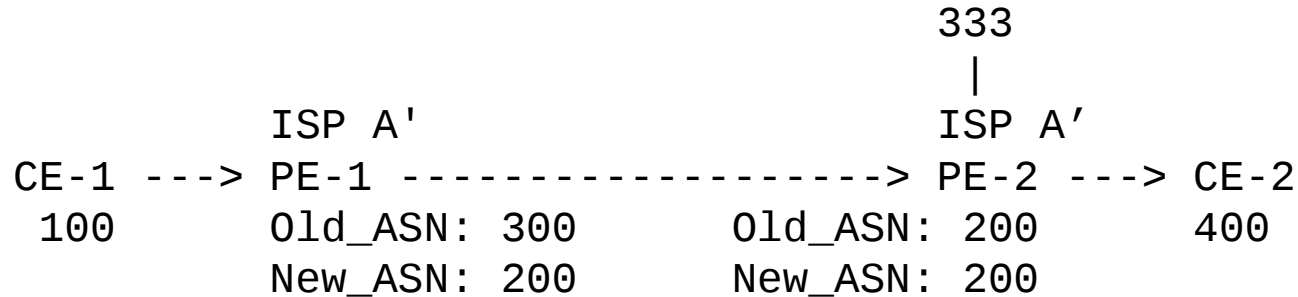
MIGRATING in Other Direction – Alternative 1



```
PE-1 to PE-2:  sig(<200>,300,pcount=0,sig21)K_300-PE1  [sig22]
                sig(origin(N2),<300>,100,pcount=1)K_100-CE1  [sig21]
                AS_PATH=(300,100)
                length=sum(pcount)=1
```

PE-1 is the one doing the migration, so it makes sense that it is the one that has to take special action. but this is an ibgp session and not normal to be adding bgpsec attrb on ibgp session. Again, this is working as if there was a ebgp hop internal to PE1, at least as far as the bgpsec attributes

MIGRATING in Other Direction – Alternative 2

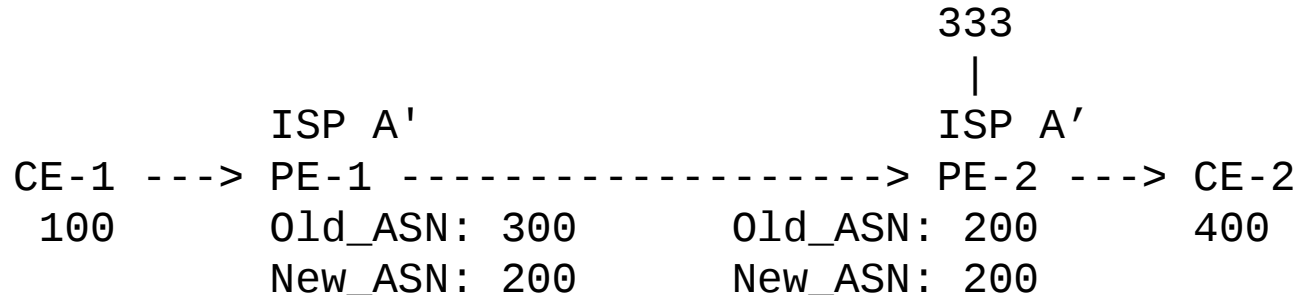


CE-1 to PE-1: sig(origin(N2), <300>, 100, pcount=1)K_100-CE1 [sig21]
 AS_PATH=(100)
 length=sum(pcount)=1

PE-1 to PE-2: sig21

PE-2 to CE-2: sig(<400>, 200, pcount=1, sig22)K_200-PE2 [sig23]
 sig(<200>, 300, pcount=0, sig21)K_300-PE1 [sig22]
 sig(origin(N2), <300>, 100, pcount=1)K_100-CE1 [sig21]
 AS_PATH=(200, 300, 100)
 length=sum(pcount)=2

MIGRATING in Other Direction – Alternative 2



CE-1 to PE-1: sig(origin(N2), <300>, 100, pcount=1)K_100-CE1 [sig21]
 AS_PATH=(100)
 length=sum(pcount)=1

PE-1 to PE-2: sig21

PE-2 to CE-2: sig(<400>, 200, pcount=1, sig22)K_200-PE2 [sig23]
 sig(<200>, 300, pcount=0, sig21)K_300-PE1 [sig22]
 sig(origin(N2), <300>, 100, pcount=1)K_100-CE1 [sig21]
 AS_PATH=(200, 300, 100)
 length=sum(pcount)=2

Defined behavior is that bgpsec attribute is added on ebgp sessions only.
 But this means that PE-2 has to know what other PE's are migrating. (Non-starter)