BGP Add-Paths

Hundreds of proposals hiding behind one...

Pierre.Francois@UCLouvain.be

ToC

- draft-ietf-idr-add-paths
  Why doing Add-paths
- draft-ietf-idr-add-paths-guidelines
  (draft-uttaro-idr-add-paths-guidelines)
  Why only a small subset of proposals will be supported

Motivation for Add-paths

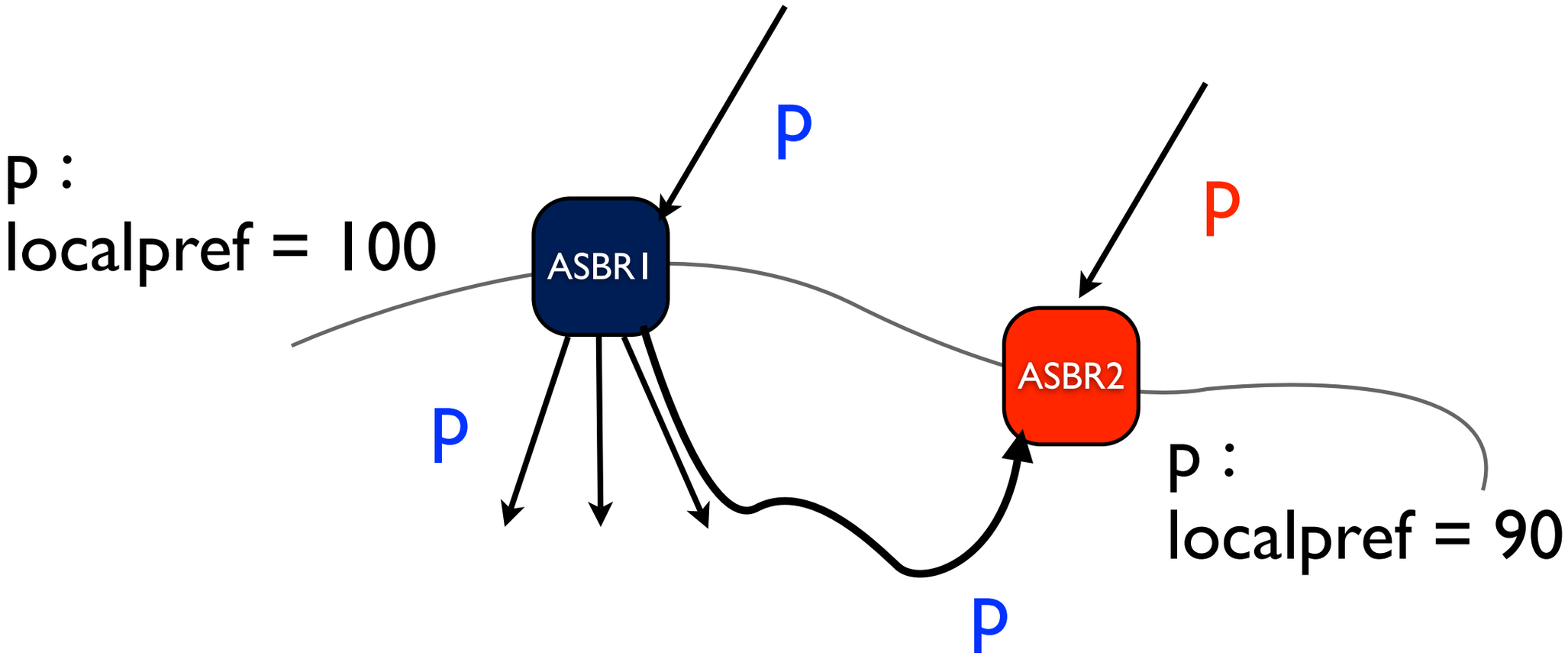
- Initial "motivation" was MED oscillation avoidance
- Emergence of new IDR requirements a few years ago
  - Fast recovery upon peering link / ASBR failure
  - Load balancing among multiple primary BGP NHs
  - Hitless planned maintenance

- "Optimal" hot-potato routing
- (Churn reduction / convergence concealment)

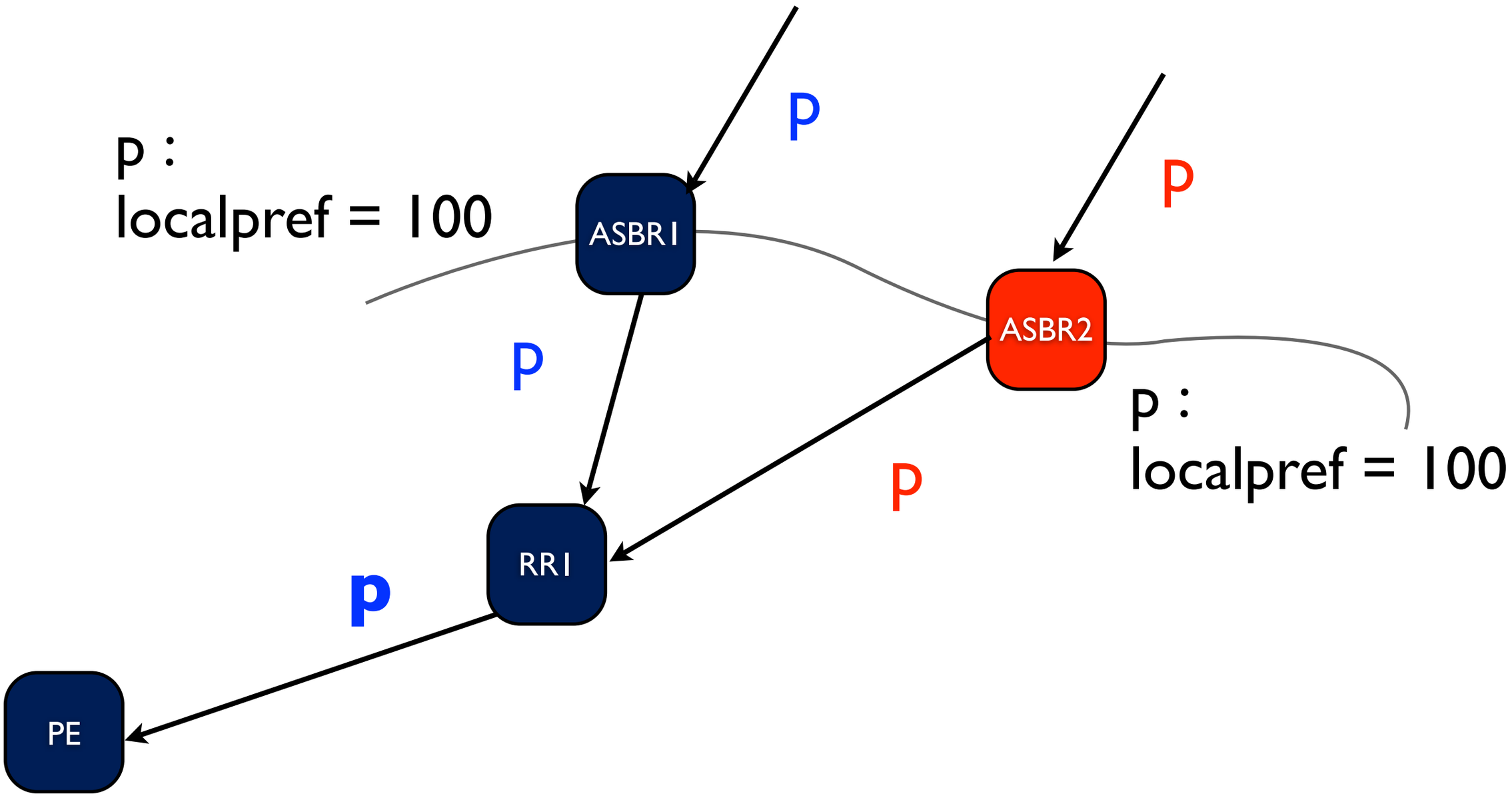
iBGP Path hiding

- Lack of path diversity in iBGP deployments
  - Policies
  - Route Reflection

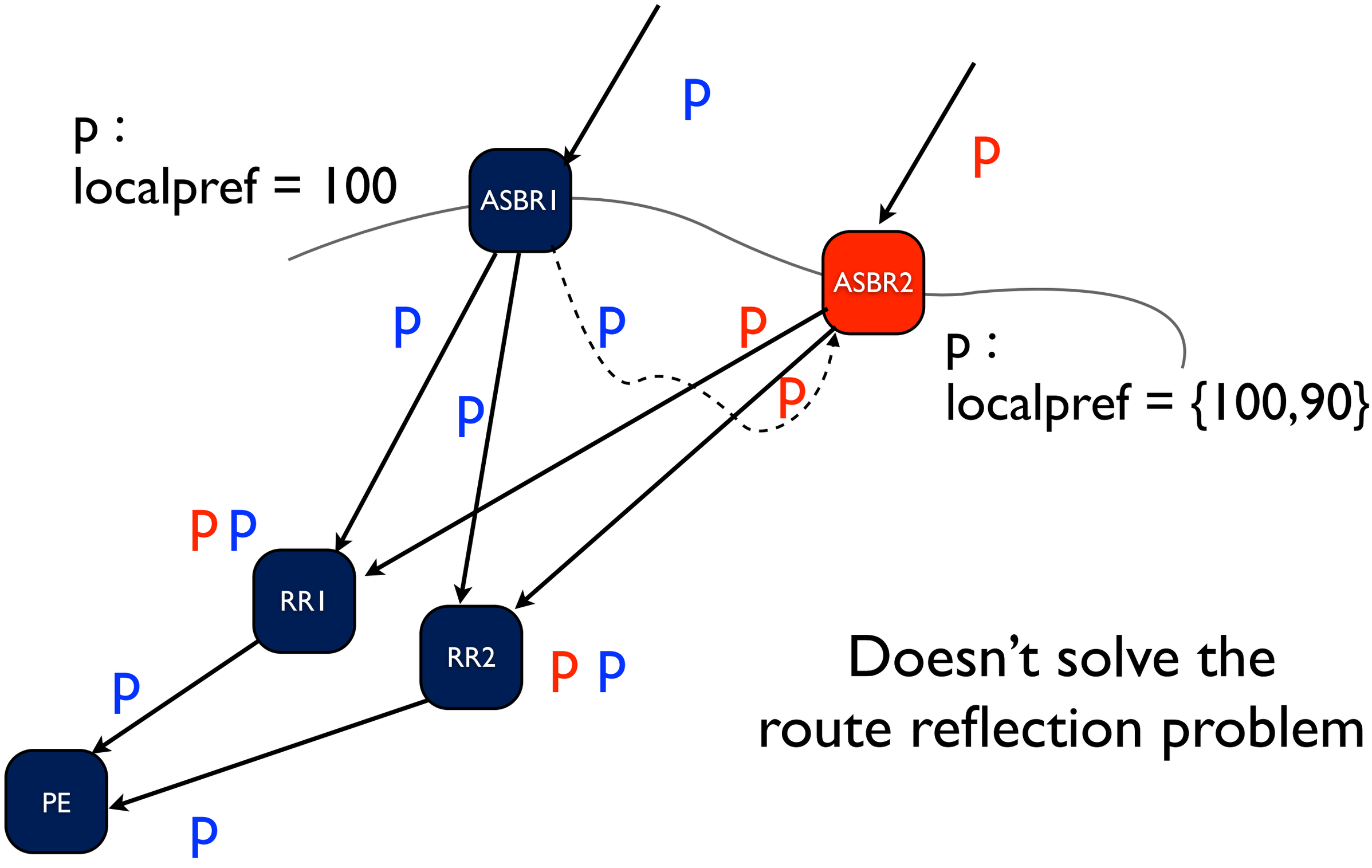
Policies let paths be hidden



Route Reflection hides paths

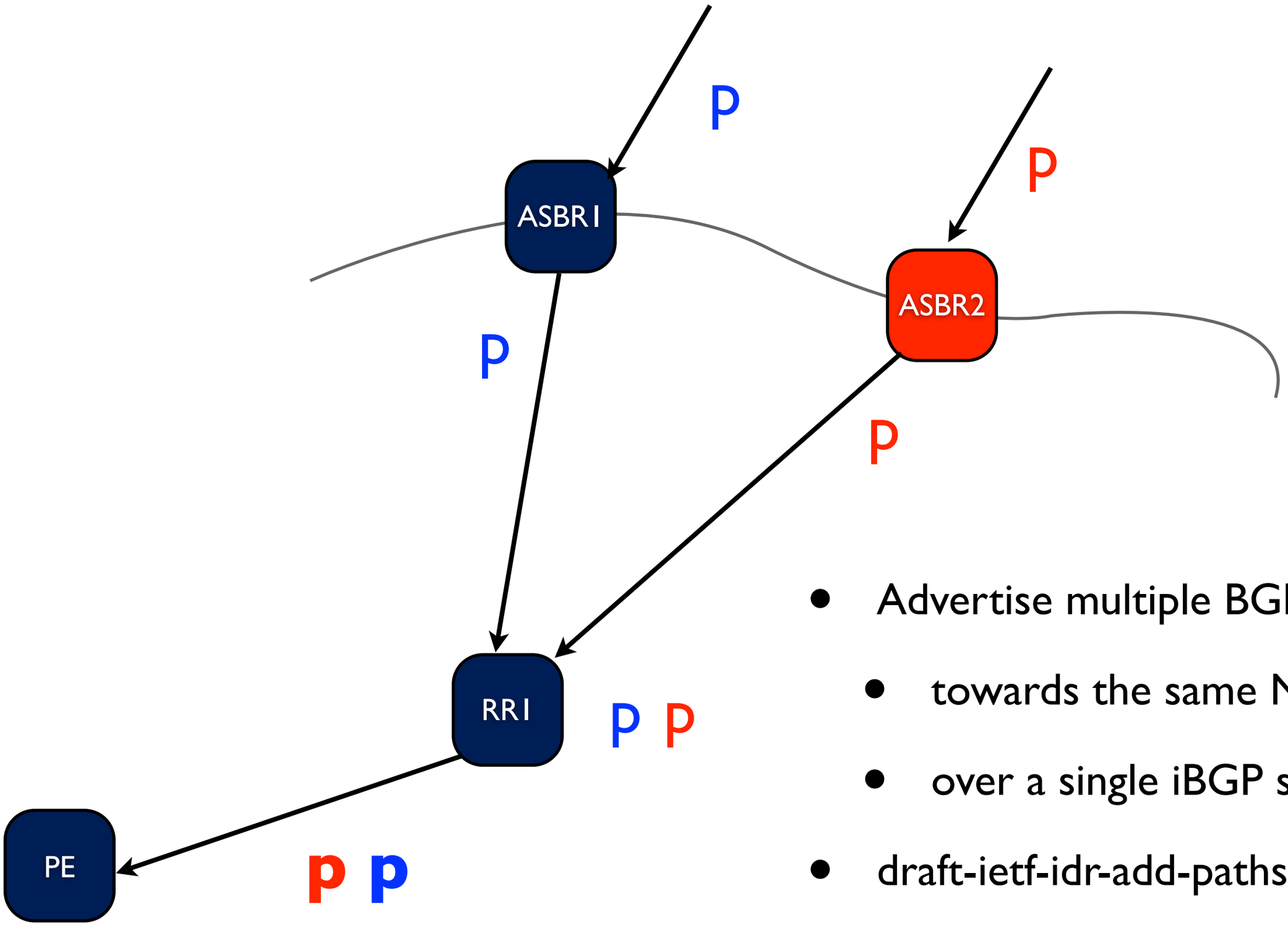


Can’t we just turn adv-best-external on ?



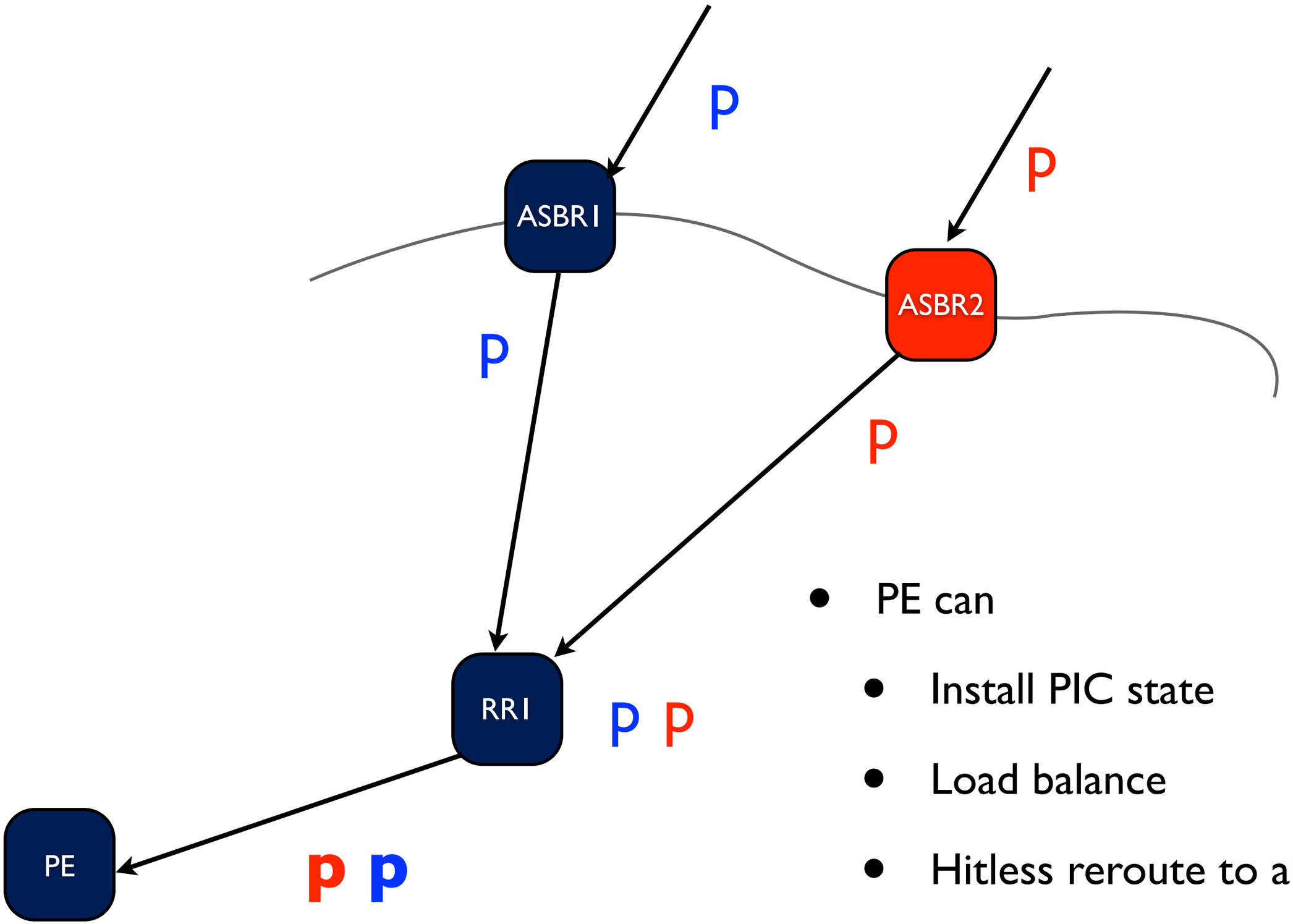
Doesn’t solve the
route reflection problem

BGP Add paths



- Advertise multiple BGP paths
  - towards the same NLRI
  - over a single iBGP session
- draft-ietf-idr-add-paths

BGP Add paths



- PE can
  - Install PIC state
  - Load balance
  - Hitless reroute to alternate

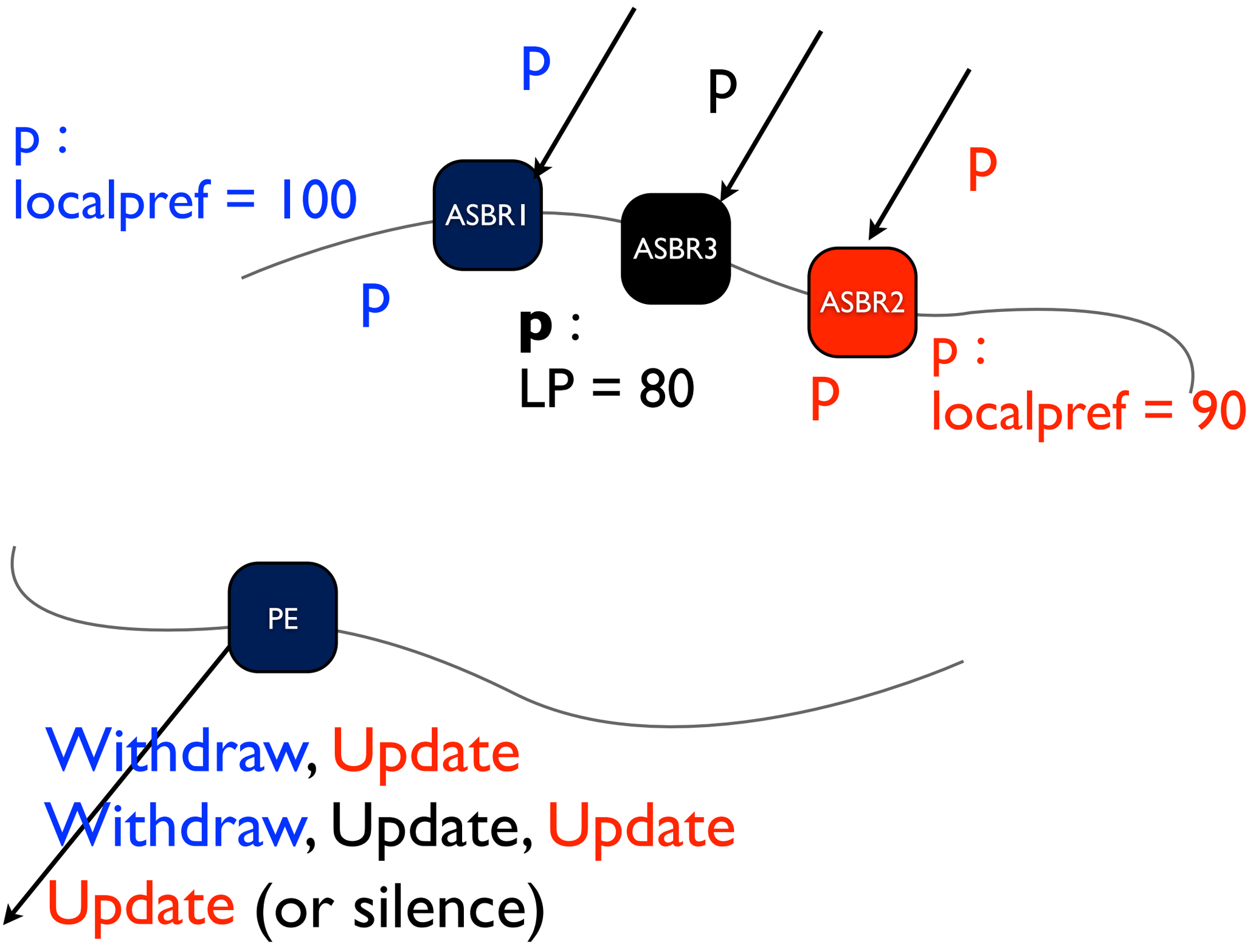
Optimal Hot Potato

- RRs may perform different IGP tie-breaking
  - Clients don’t get the path that they would pick
  - Add-paths enabled RRs let the IGP tie-break to clients
    - Depending on which paths it advertises

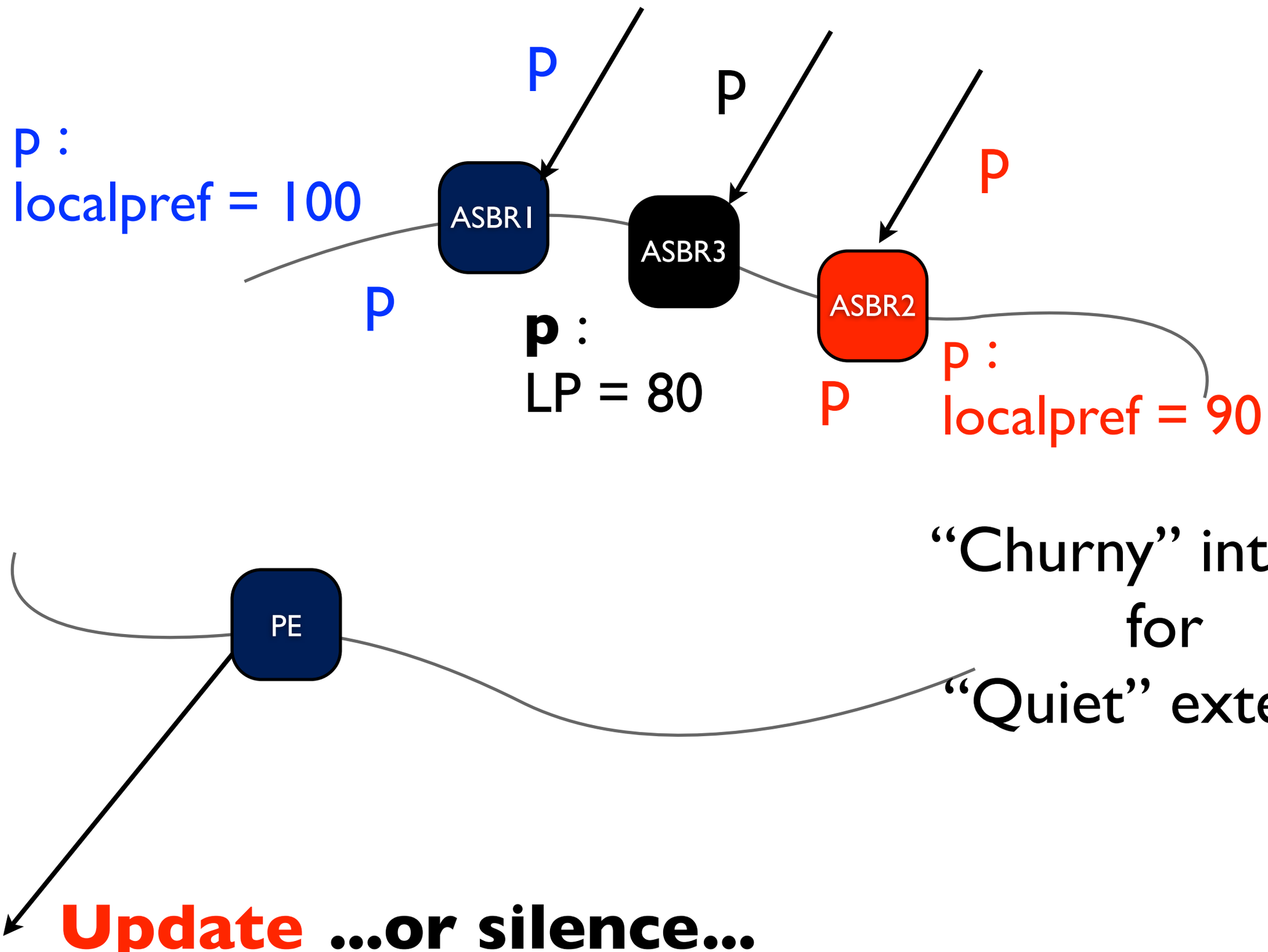
Churn reduction ???

- Churn reduction for primary paths...
- ...with internal churn increase for non-primary ones

Churn Reduction



Churn Reduction



"Churny" internal
for
"Quiet" external

Update ...or silence...

draft-ietf-idr-add-paths

- Adds an identifier to paths
- Identifier only has session meaning

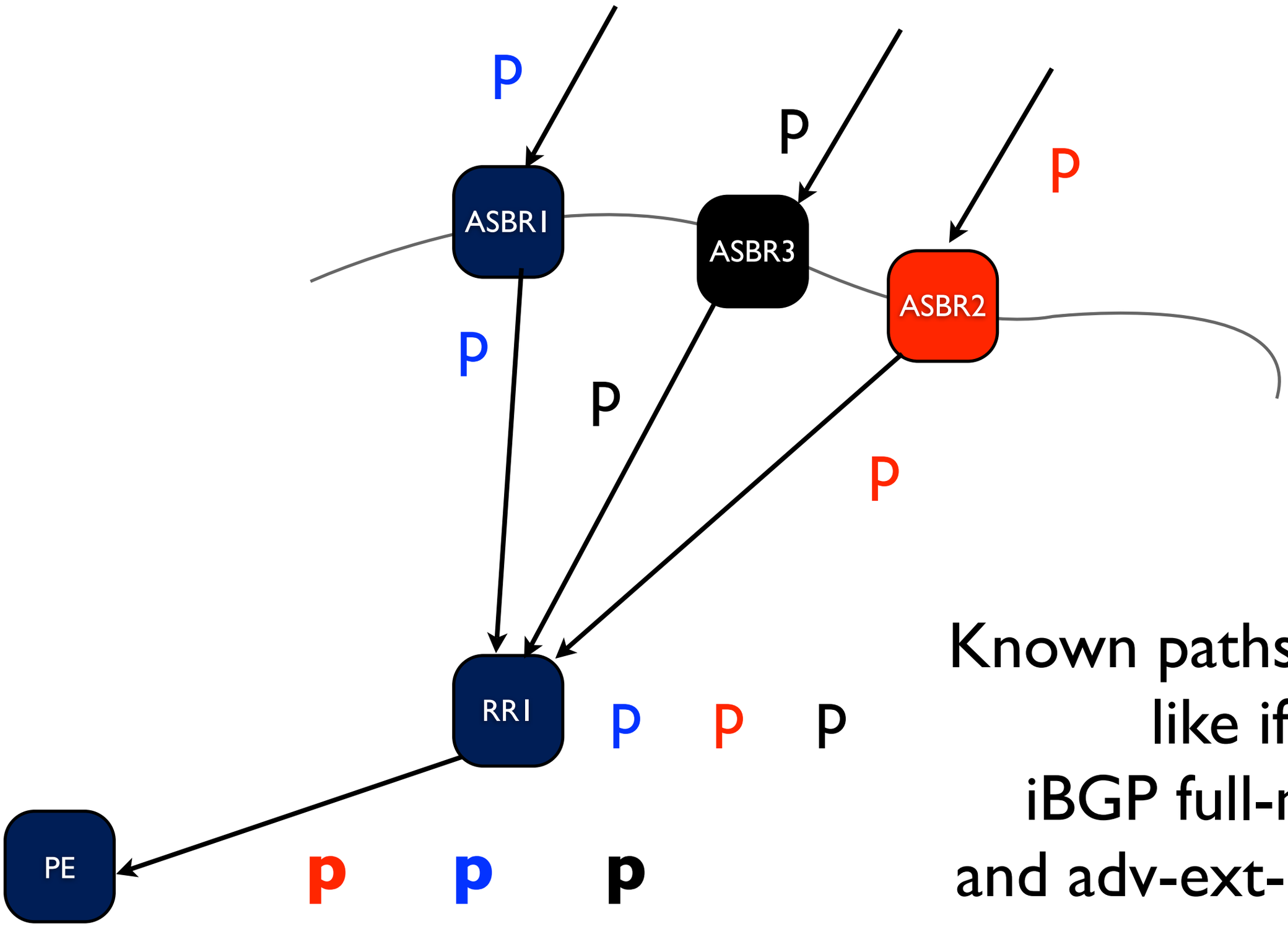
draft-ietf-idr-add-paths-guidelines

- draft-ietf-idr-add-paths doesn't tell which paths to select
- Multiple motivations lead to different "selection modes"
  - Evaluate them (what they give, at which cost)
    - analytical
    - *"numbers"*

Modes

- All paths
- N paths
- AS-Wide best paths (and variants)
- Best Loc Pref / Second best Loc Pref paths
- Decisive step -1 paths
- Neighbor-AS group best paths

All Paths

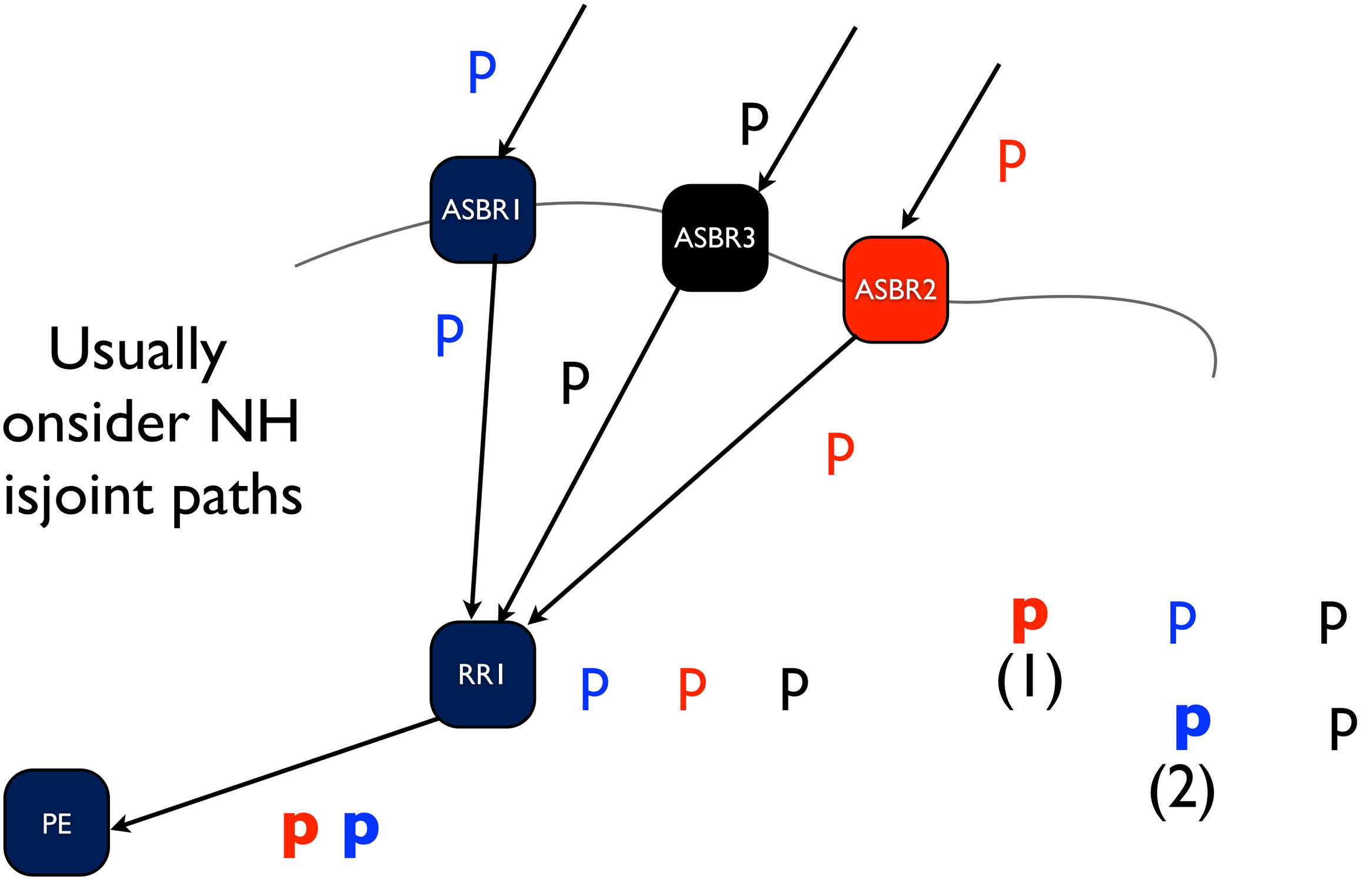


Known paths almost
like if
iBGP full-mesh
and adv-ext-best on

Add-All

- Easiest Decision Process algorithm
- Nice mode to turn on towards a BGP monitor
- Memory/internal update churn monster
  - Depending on how many paths for each p

N paths (N is configured)



Usually consider NH disjoint paths

Add-N-Paths

- Most practical use cases
  - Set N to 2 for basic PIC support
  - Set N to desired number of NHs for LB
- Memory hit kept under control through configuration of N

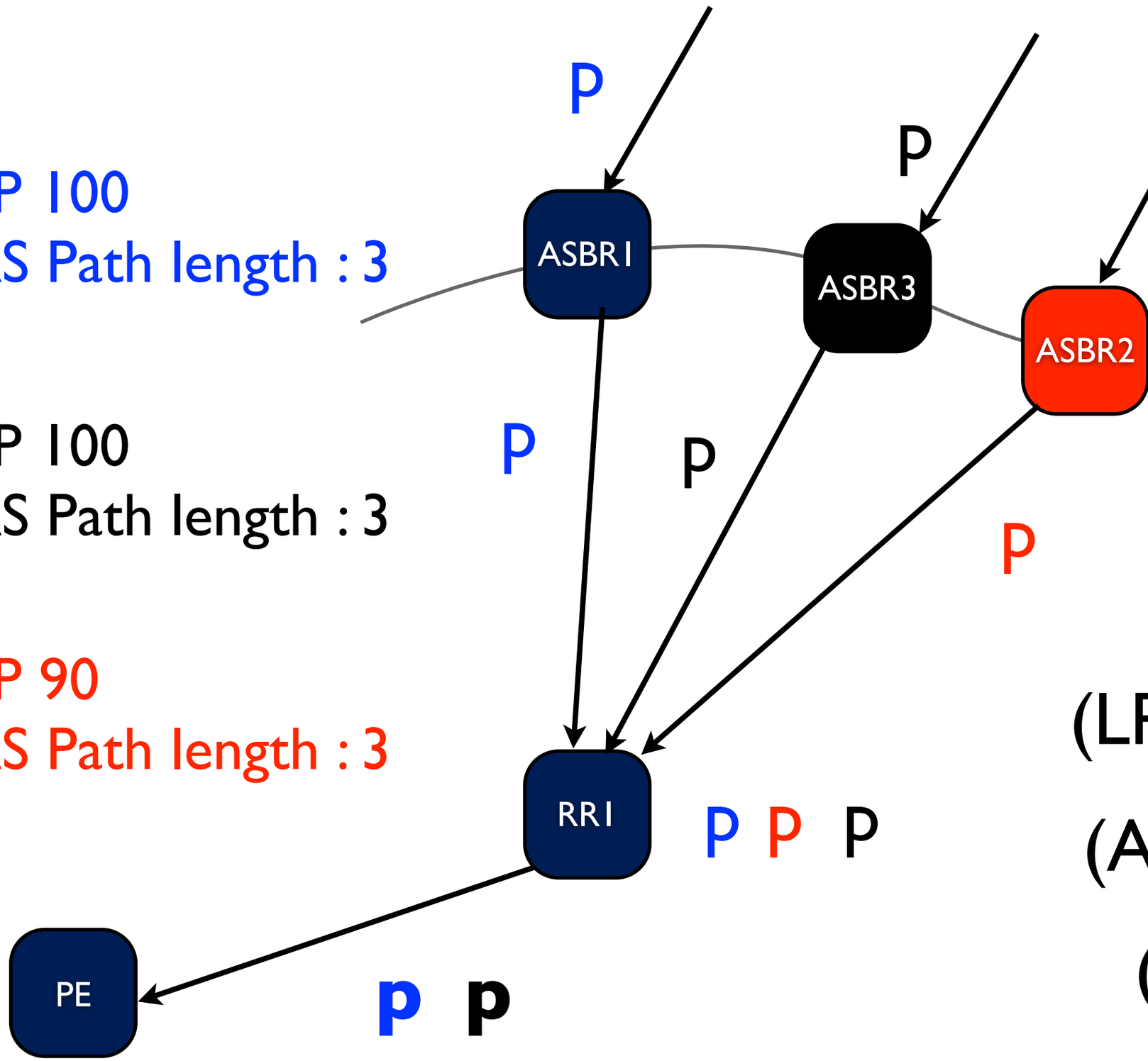
- Doesn’t solve MED oscillations
- Developers tend to implement it as N*DP

AS-Wide Best paths

P
LP 100
AS Path length : 3

P
LP 100
AS Path length : 3

P
LP 90
AS Path length : 3



Not hiding paths that another node would have preferred

(LP)	P	**p**	**p**
(AS Path)		**p**	**p**
(MED)		**p**	**p**

AS-Wide Best paths

- “The router doesn’t make local decisions”
- DP complexity < not running add-paths
- Provides routing optimality and max LB potential
- Provides MED oscillation avoidance

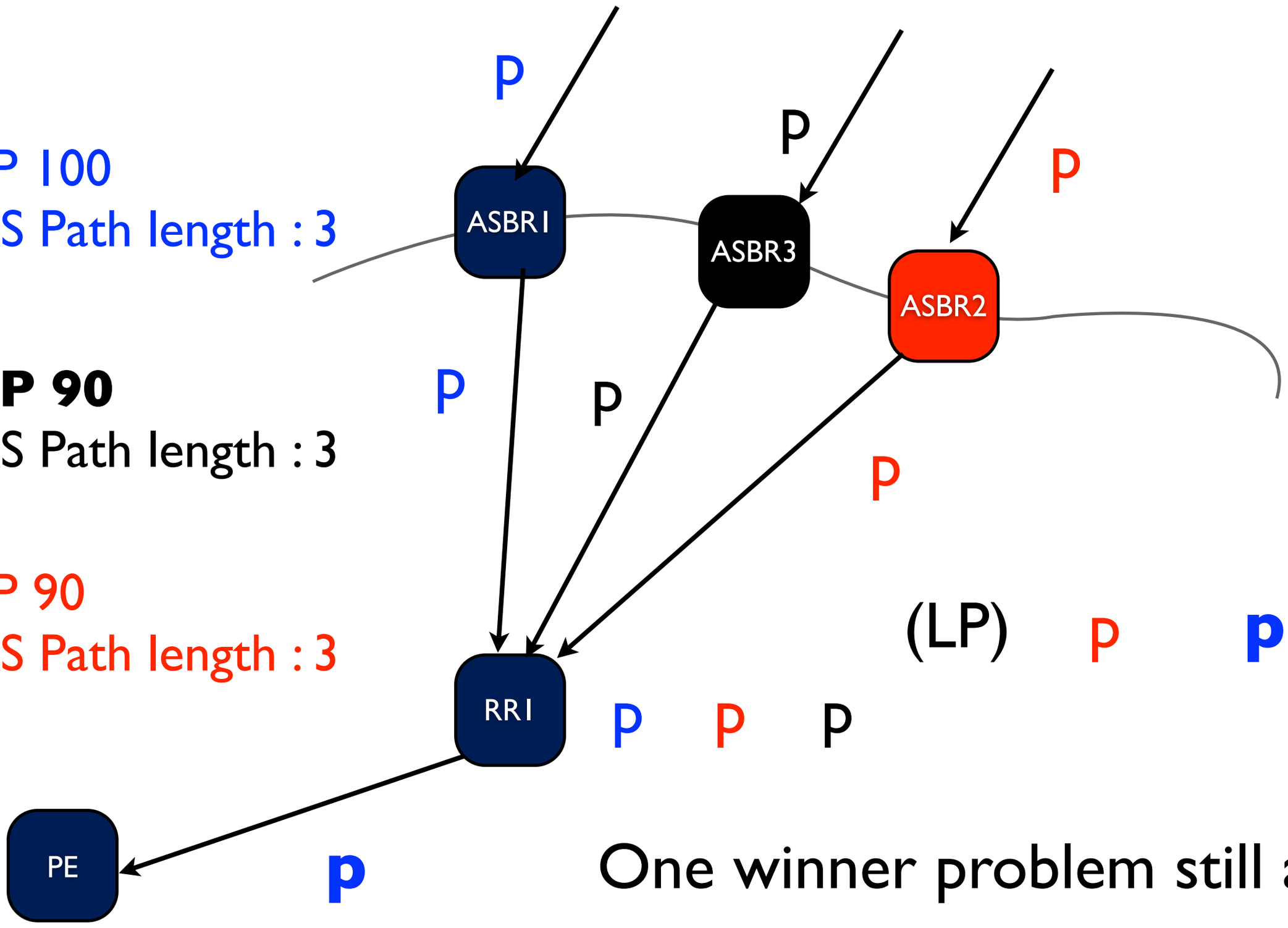
- !!! Doesn’t feed PIC !!!

AS-Wide Best paths

P
LP 100
AS Path length : 3

P
LP 90
AS Path length : 3

P
LP 90
AS Path length : 3



(LP) P p P

P P P

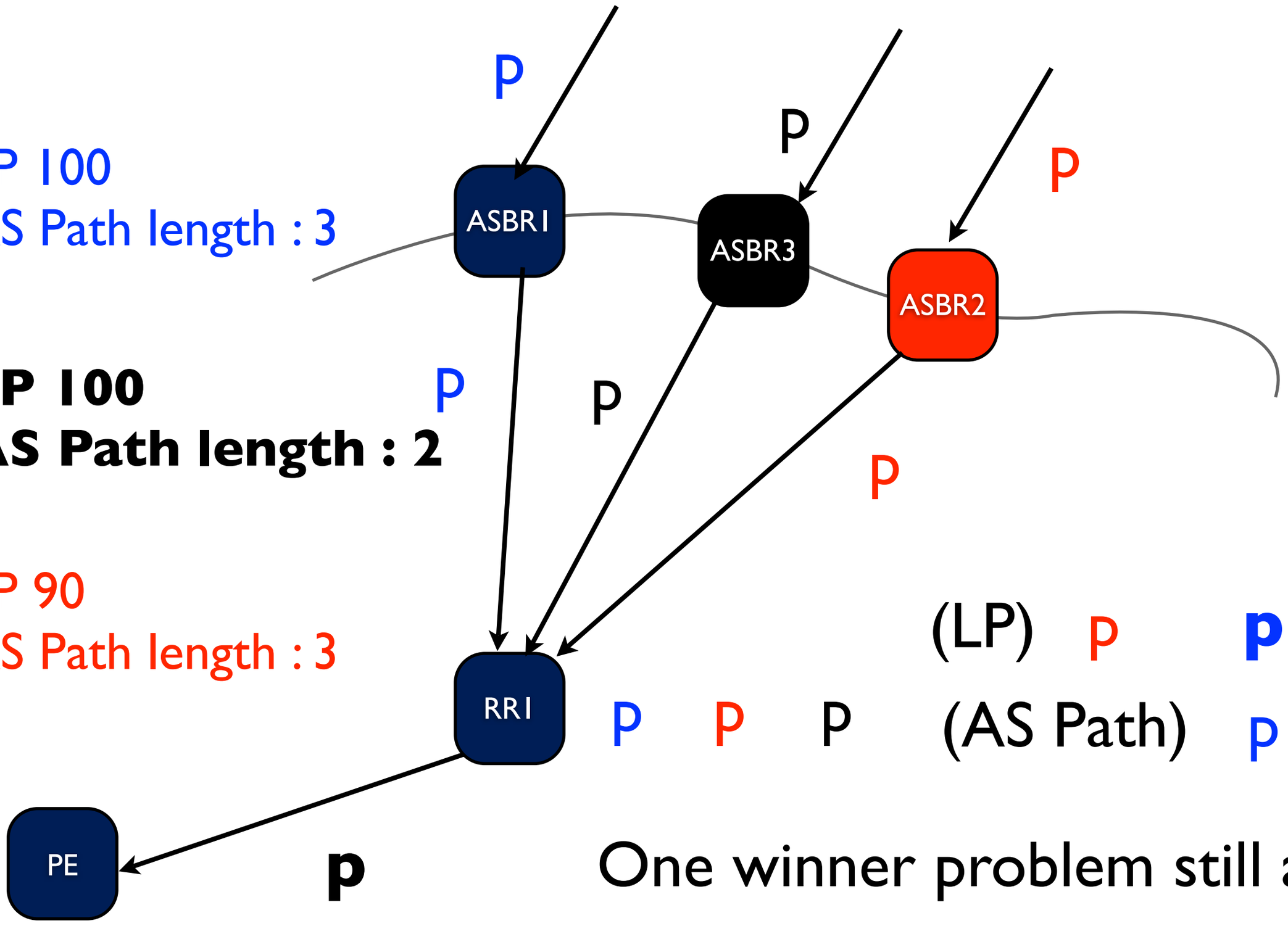
One winner problem still applies

AS-Wide Best paths

P
LP 100
AS Path length : 3

P
LP 100
AS Path length : 2

P
LP 90
AS Path length : 3



(LP) P **p** **p**

P P P (AS Path) p **p**

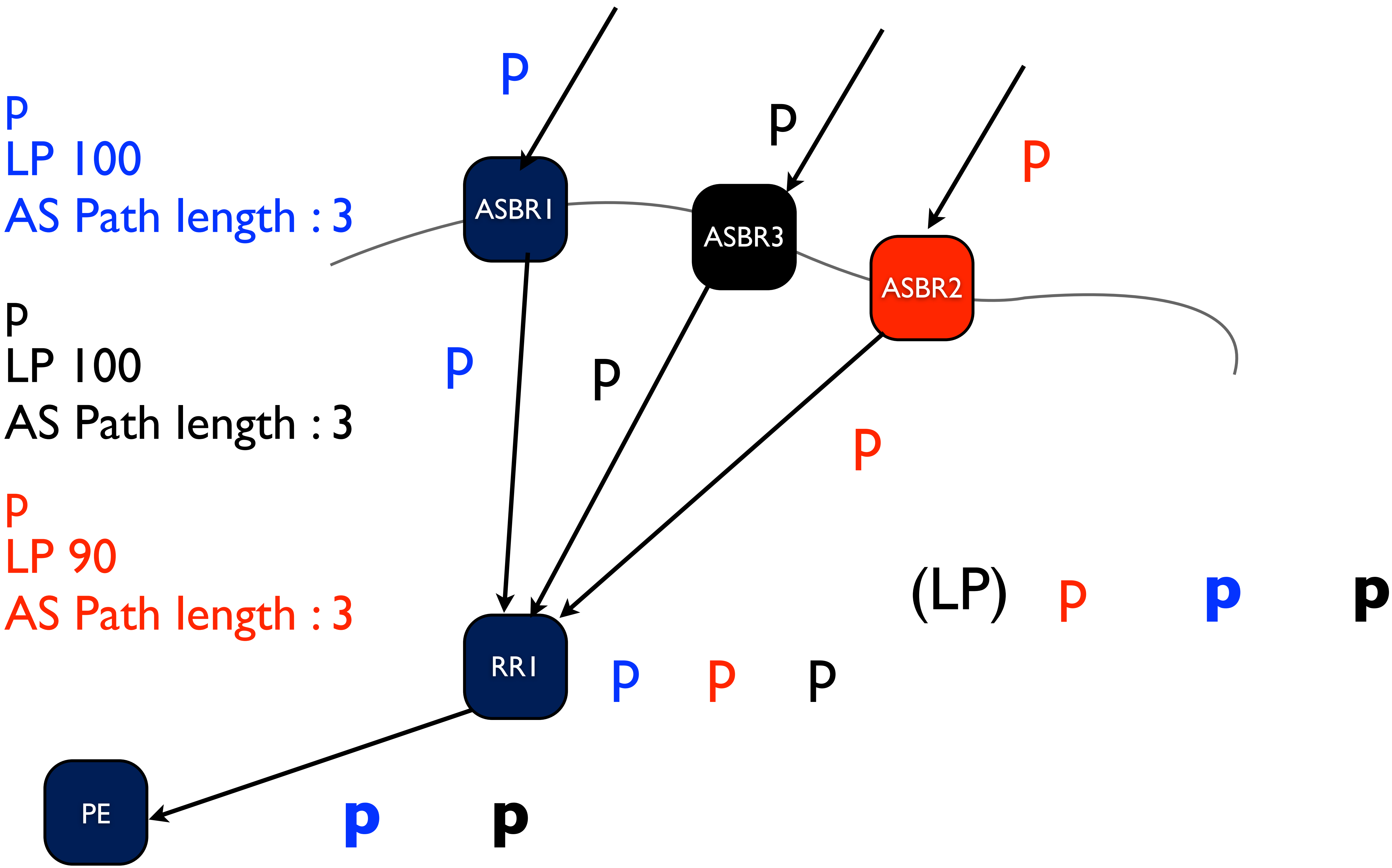
p

One winner problem still applies

Best LP/Second Best LP

- If #(paths with highest LP) > 1
  - advertise paths with highest LP
- else
  - advertise the path with highest LP
  - advertise the paths with second highest LP

Best LP/Second Best LP

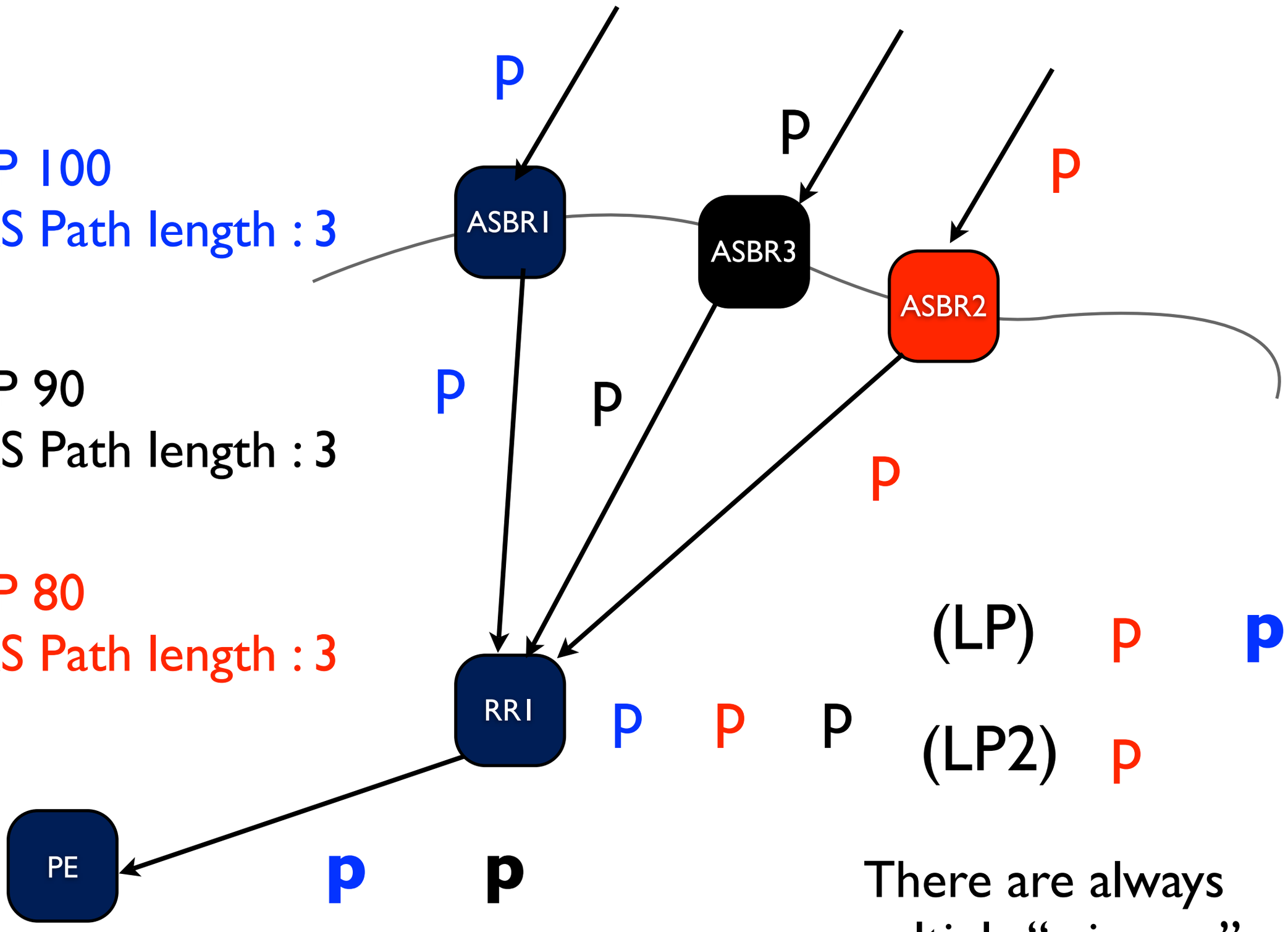


Best LP/Second Best LP

P
LP 100
AS Path length : 3

P
LP 90
AS Path length : 3

P
LP 80
AS Path length : 3



(LP) P p P

(LP2) P p

There are always
multiple “winners”

Best LP / Second Best LP

- Adj-Rib-In optimized for this mode contains two or three sets of paths per NLRI
  - Best bin
  - Second best bin if required
  - Others
- Decision Process :
  Select what's in first and second bin

Decisive step - 1

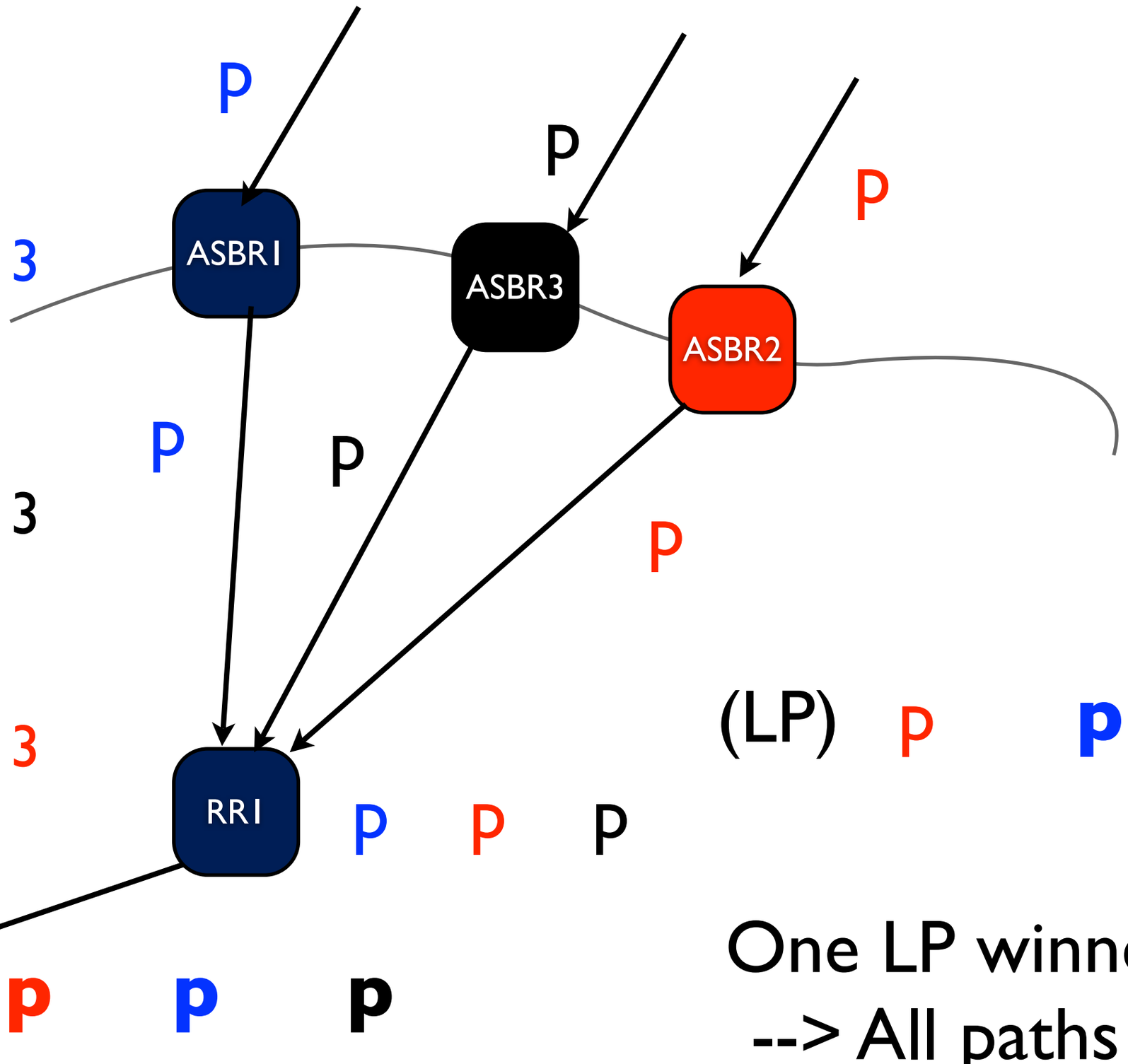
- Apply normal BGP selection process, but
  - If IGP tie-break rule is reached, advertise what remains
    - If best path is found at a preceding rule i, advertise what remained when applying rule i-1
- Tries to obtain diversity while advertising as few paths as possible

Decisive step - 1

P
LP 100
AS Path length : 3

P
LP 90
AS Path length : 3

P
LP 90
AS Path length : 3



(LP) P **p** P

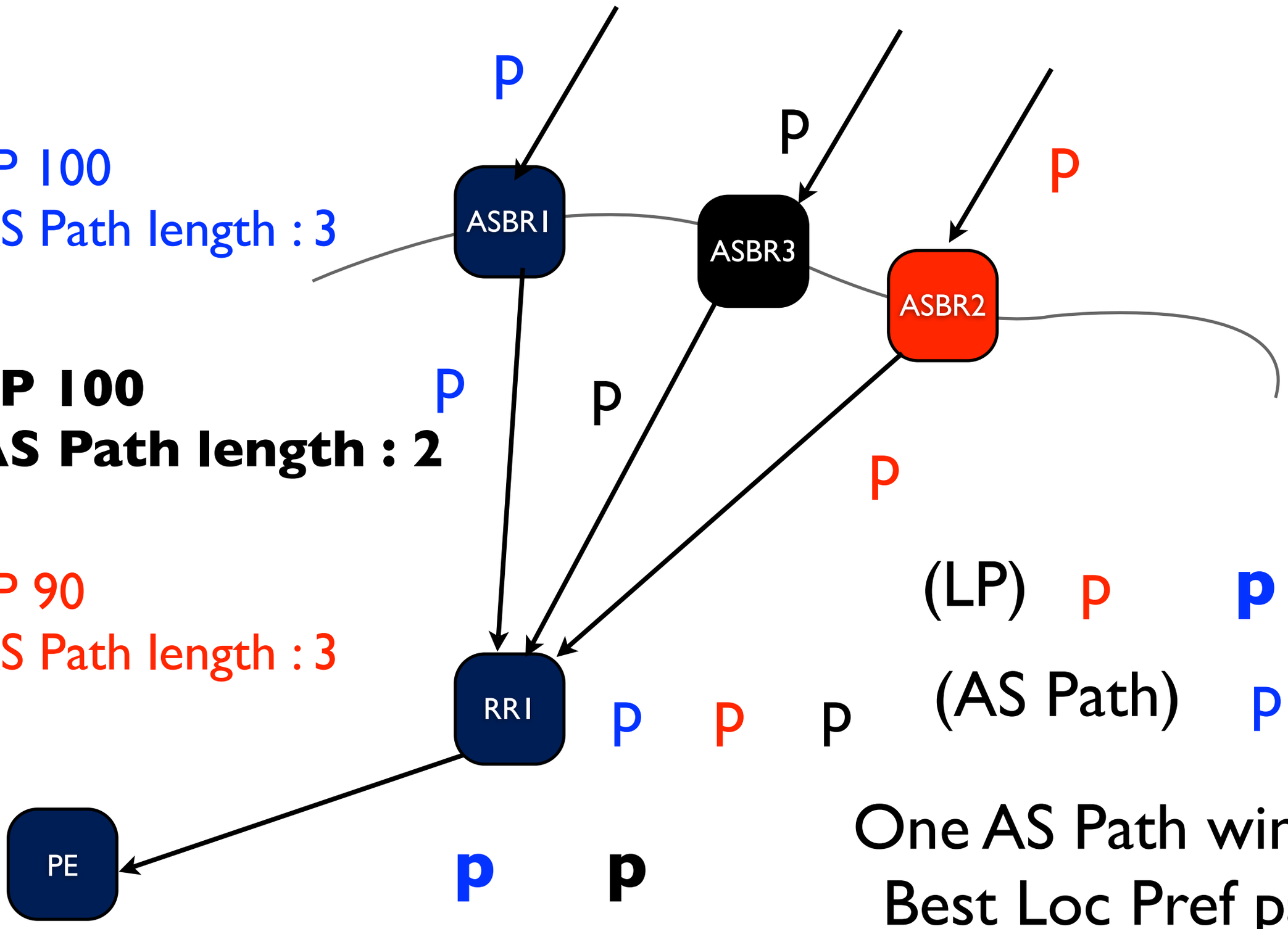
One LP winner
--> All paths !

Decisive step - 1

P
LP 100
AS Path length : 3

P
LP 100
AS Path length : 2

P
LP 90
AS Path length : 3



P P P

P P P

(LP) P **p** **p**

(AS Path) P **p**

p **p**

One AS Path winner :
Best Loc Pref paths

Neighbor-AS group best

- Avoids MED oscillations
  - draft-walton-bgp-route-oscillation-stop
  - Advertise the best path from each neighboring AS
    - No ASBR picks as best a non-lowest MED path

Neighbor-AS group best

- Provides paths from different neighboring ASes, but
  - their existence is not guaranteed
  - nothing to deal with post-convergence paths

Summary

	Path optimality	Backup availability / optimality	Control plane load and stress	DP Complexity	MED osc. avoidance
All	OK	OK	Max	EASIEST	OK
N	?	OK / ?	Bounded	Depends on N can be optimized	?
AS-Wide	OK	KO / ~OK	~MAX	EASY	OK
LP1/LP2	OK	OK	~MAX	EASIER	OK
Decisive-1	OK	OK	~MAX	Easy but "spaghetti"	OK
Group best	KO ...	KO	~MAX	?	OK

Summary

	Path optimality	Backup availability / optimality	Control plane load and stress	DP Complexity	MED osc. avoidance
All	OK	OK	**Max**	EASIEST	OK
N	?	OK / ?	**Bounded**	Depends on N can be optimized	?
AS-Wide	OK	KO / ~OK	**~MAX**	EASY	OK
LP1/LP2	OK	OK	**~MAX**	EASIER	OK
Decisive-1	OK	OK	**~MAX**	Easy but "spaghetti"	OK
Group best	KO ...	KO	**~MAX**	?	OK

Summary

	Path optimality	Backup availability / optimality	Control plane load and stress	DP Complexity	MED osc. avoidance
All	OK	OK	Max	EASIEST	**OK**
N	?	OK / ?	Bounded	Depends on N can be optimized	**?**
AS-Wide	OK	KO / ~OK	~MAX	EASY	**OK**
LP1/LP2	OK	OK	~MAX	EASIER	**OK**
Decisive-1	OK	OK	~MAX	Easy but "spaghetti"	**OK**
Group best	KO ...	KO	~MAX	?	**OK**

Summary

	Path optimality	Backup availability / optimality	Control plane load and stress	DP Complexity	MED osc. avoidance
All	OK	OK	Max	EASIEST	**OK**
N	?	OK / ?	Bounded	Depends on N can be optimized	**?**
AS-Wide	**OK**	KO / ~OK	~MAX	EASY	**OK**
LP1/LP2	OK	OK	~MAX	EASIER	**OK**
Decisive-1	OK	OK	~MAX	Easy but “spaghetti”	**OK**
Group best	**KO ...**	KO	~MAX	?	**OK**

Summary

	Path optimality	Backup availability / optimality	Control plane load and stress	DP Complexity	MED osc. avoidance
All	OK	OK	Max	EASIEST	OK
N	?	OK / ?	Bounded	Depends on N can be optimized	?
AS-Wide	OK	KO / ~OK	~MAX	EASY	OK
LP1/LP2	OK	OK	~MAX	EASIER	OK
Decisive-1	OK	OK	~MAX	Easy but "spaghetti"	OK
Group best	KO ...	KO	~MAX	?	OK

Summary

	Path optimality	Backup availability / optimality	Control plane load and stress	DP Complexity	MED osc. avoidance
All	OK	OK	Max	EASIEST	OK
N	?	OK / ?	Bounded	Depends on N can be optimized	?
AS-Wide	OK	KO / ~OK	~MAX	EASY	OK
LP1/LP2	OK	OK	~MAX	EASIER	OK
Decisive-1	OK	OK	~MAX	**Easy but “spaghetti”**	OK
Group best	KO ...	KO	~MAX	?	OK

Summary

	Path optimality	Backup availability / optimality	Control plane load and stress	DP Complexity	MED osc. avoidance
All	OK	OK	Max	EASIEST	OK
N	?	OK / ?	Bounded	Depends on N can be optimized	?
AS-Wide	OK	KO / ~OK	~MAX	EASY	OK
LP1/LP2	OK	OK	~MAX	EASIER	OK
Decisive-1	OK	OK	~MAX	Easy but "spaghetti"	OK
Group best	KO ...	KO	~MAX	?	OK

Summary

	Path optimality	Backup availability / optimality	Control plane load and stress	DP Complexity	MED osc. avoidance
All	OK	OK	Max	EASIEST	OK
N	?	OK / ?	Bounded	Depends on N can be optimized	?
AS-Wide	OK	KO / ~OK	~MAX	EASY	OK
LP1/LP2	OK	OK	~MAX	EASIER	OK
Decisive-1	OK	OK	~MAX	Easy but "spaghetti"	OK
Group best	KO ...	KO	~MAX	?	OK

Summary

	Path optimality	Backup availability / optimality	Control plane load and stress	DP Complexity	MED osc. avoidance
All	OK	OK	Max	EASIEST	OK
N	?	OK / ?	Bounded	Depends on N can be optimized	?
AS-Wide	OK	KO / ~OK	~MAX	EASY	OK
LP1/LP2	OK	OK	~MAX	EASIER	OK
Decisive-1	OK	OK	~MAX	Easy but “spaghetti”	OK
Group best	KO ...	KO	~MAX	?	OK

Current Recommendations

- MUST: Add-N
  - Default MUST be 2
    - N MUST be configurable
    - Option to not limit N (Add-All)
- OPTIONAL: AS-Wide best variants
- OPTIONAL-: All others

Tool

- input: BGP config, IGP config, as many *show ip bgp all* as possible, priority on RRs, please adv-ext-best
- output: for each mode
  - number of paths in Rib-in
  - optimality of paths
  - iBGP churn upon nexthop failure / single update
  - generated eBGP churn upon nexthop failure

Deployment

- Session wide upgrade required
- Add-path easily converted to diverse-paths
- As for all solutions
  - Forget about deployments w/o Ingress-Egress encap
  - Transient forwarding loops if naïve PIC implementation

Next Steps

- Add-path for eBGP
- Route Server implementation
- draft-jasinska-ix-bgp-route-server
  +Add-All
  +Filtering
  +Pick one for clients not supporting add-paths

Thanks !