BGP Add-Paths

Pierre Francois
Institute IMDEA Networks
Pierre.Francois@imdea.org

ToC

- Data-plane evolution : BGP PIC
- Control-plane evolution : BGP Add-paths

BGP PIC

Sub-second data-plane convergence

- Fast switchover to pre-installed alternate paths
- Convergence, not Fast ReRoute
- BGP converges when IGP converges

FIB design

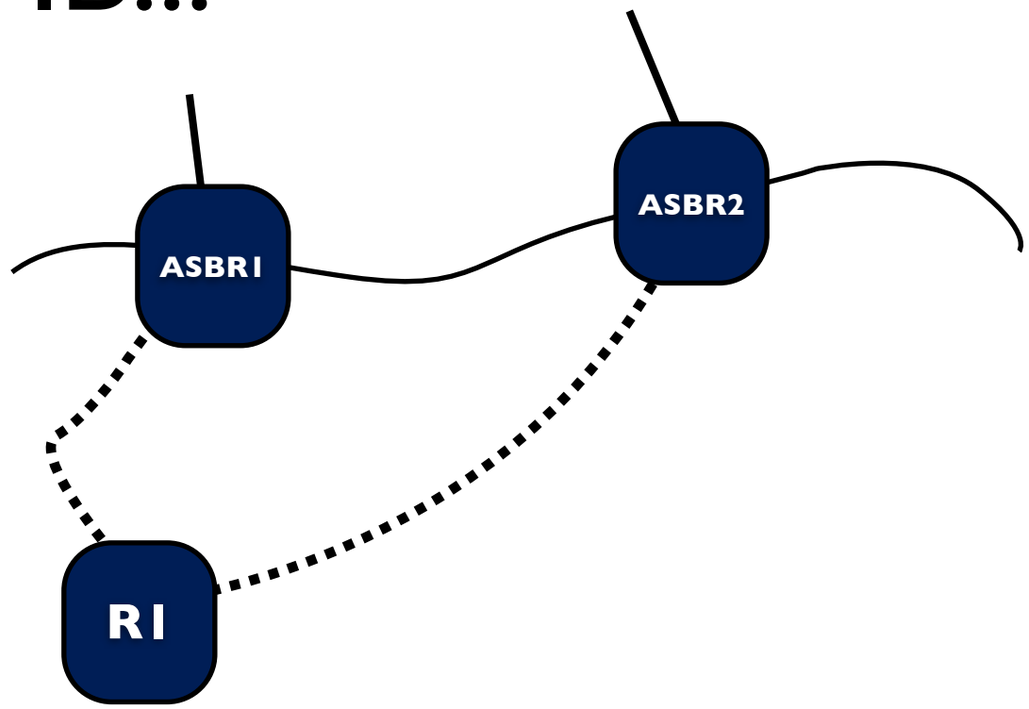
From flat FIB...

RIB

Prefix	NH or oif
p/P	ASBR1
q/Q	ASBR1
r/R	ASBR2
...	...
...	...
ASBR1	**north oif**
ASBR2	**east oif**

} BGP

} IGP



FIB

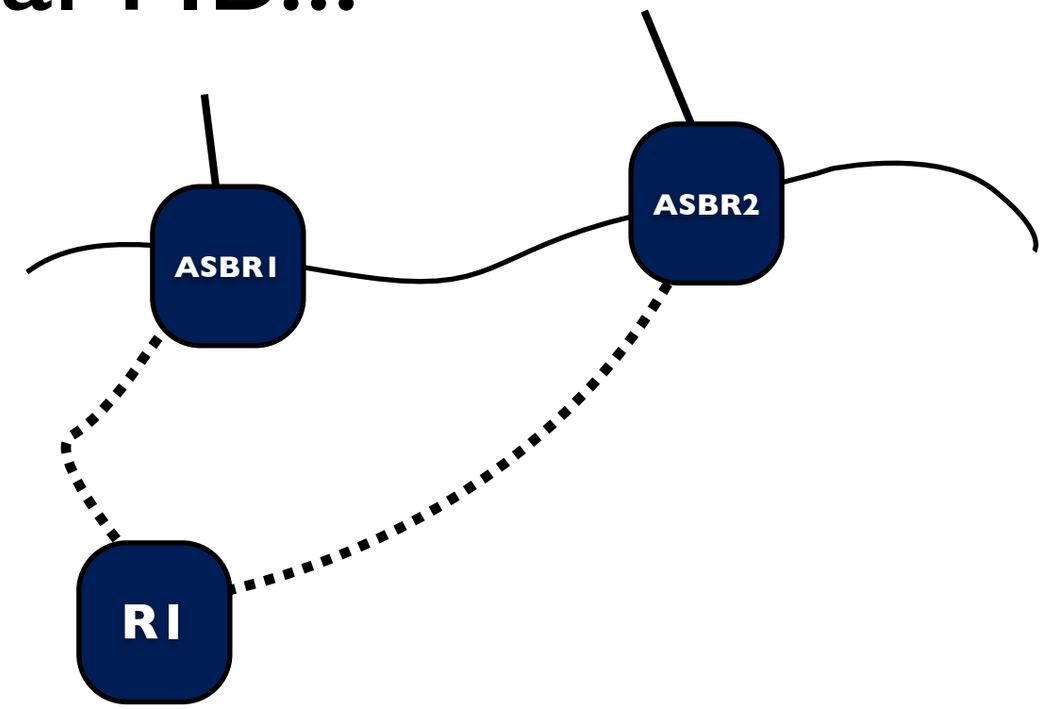
IGP change leads to FIB Updates for BGP prefixes !

Prefix	oif
p/P	**north oif**
q/Q	**north oif**
r/R	**east oif**
ASBR1	**north oif**
ASBR2	**east oif**

FIB design
to hierarchical FIB...

FIB

Prefix	NH or oif
p/P	**ASBR1**
q/Q	**ASBR1**
r/R	**ASBR2**
...	**...**
...	**...**
ASBR1	north oif
ASBR2	east oif

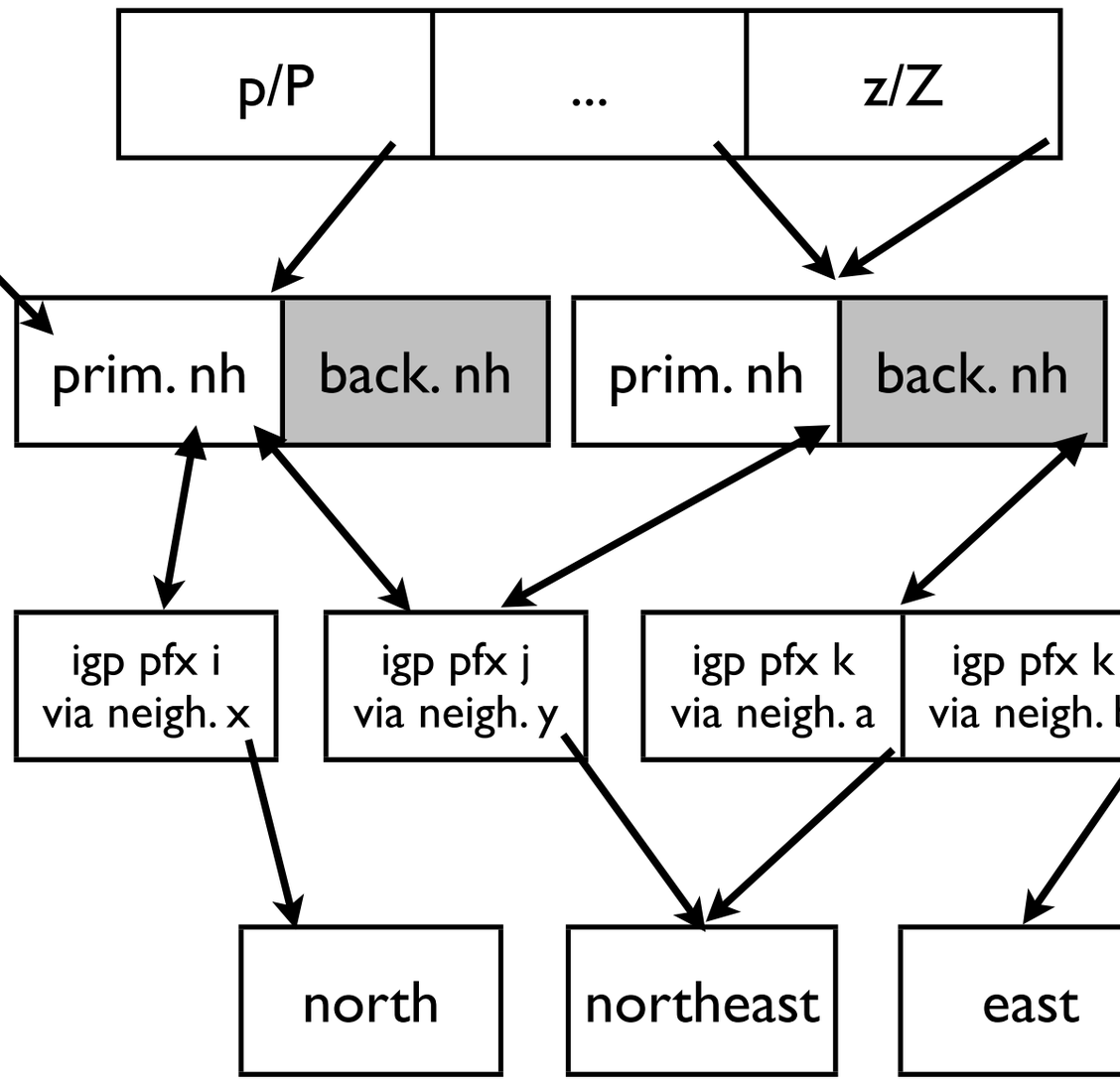


PIC “core”

Update of the outgoing interface for a BGP nexthop impacts all prefixes tracked by the nexthop

FIB design
...to generalized FIB

Becomes a list when BGP Multipath is used



BGP Path Lists
Primary nh / Backup nh

IGP (ECMP) PLs

oifs

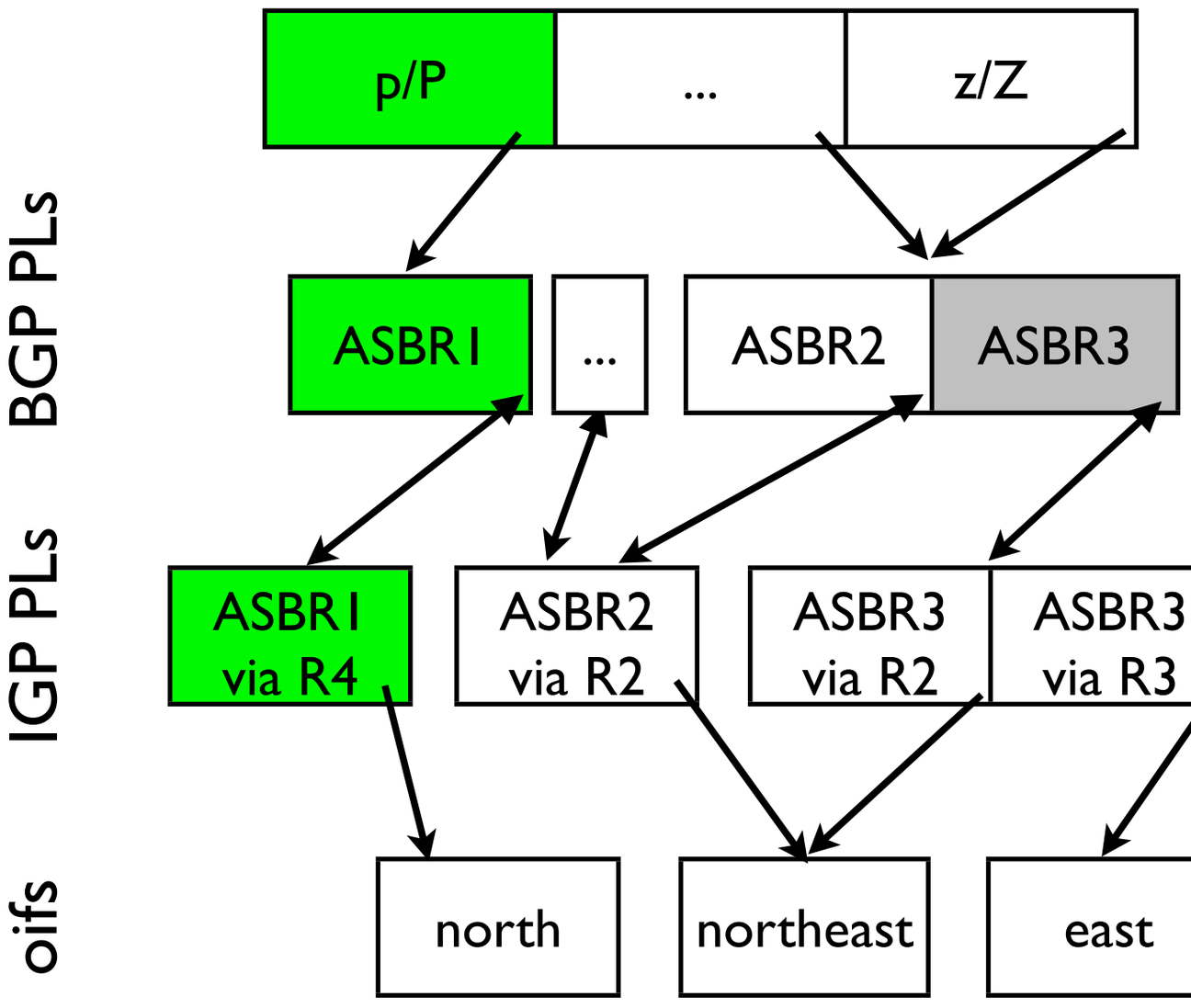
BGP prefixes share members of the BGP PL set
Small PLs # in practice

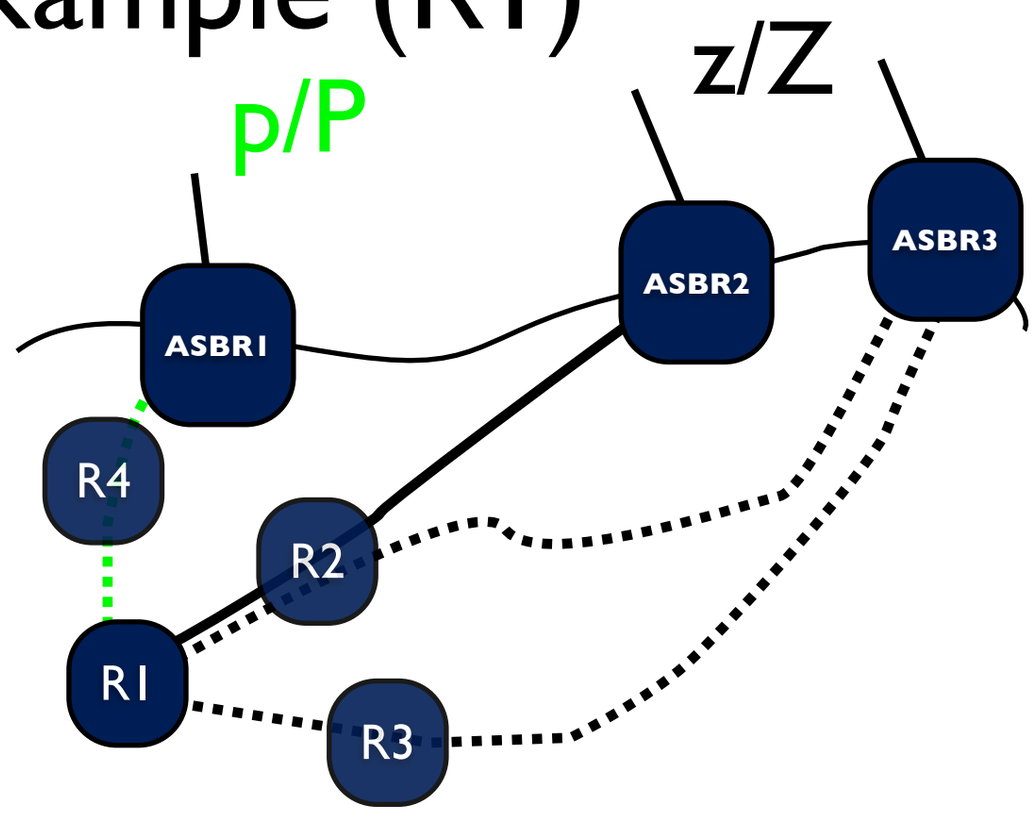
A given IGP PL is shared by BGP PLs
 # IGP PLs = n

Support for PIC “core”/“edge”

FIB design

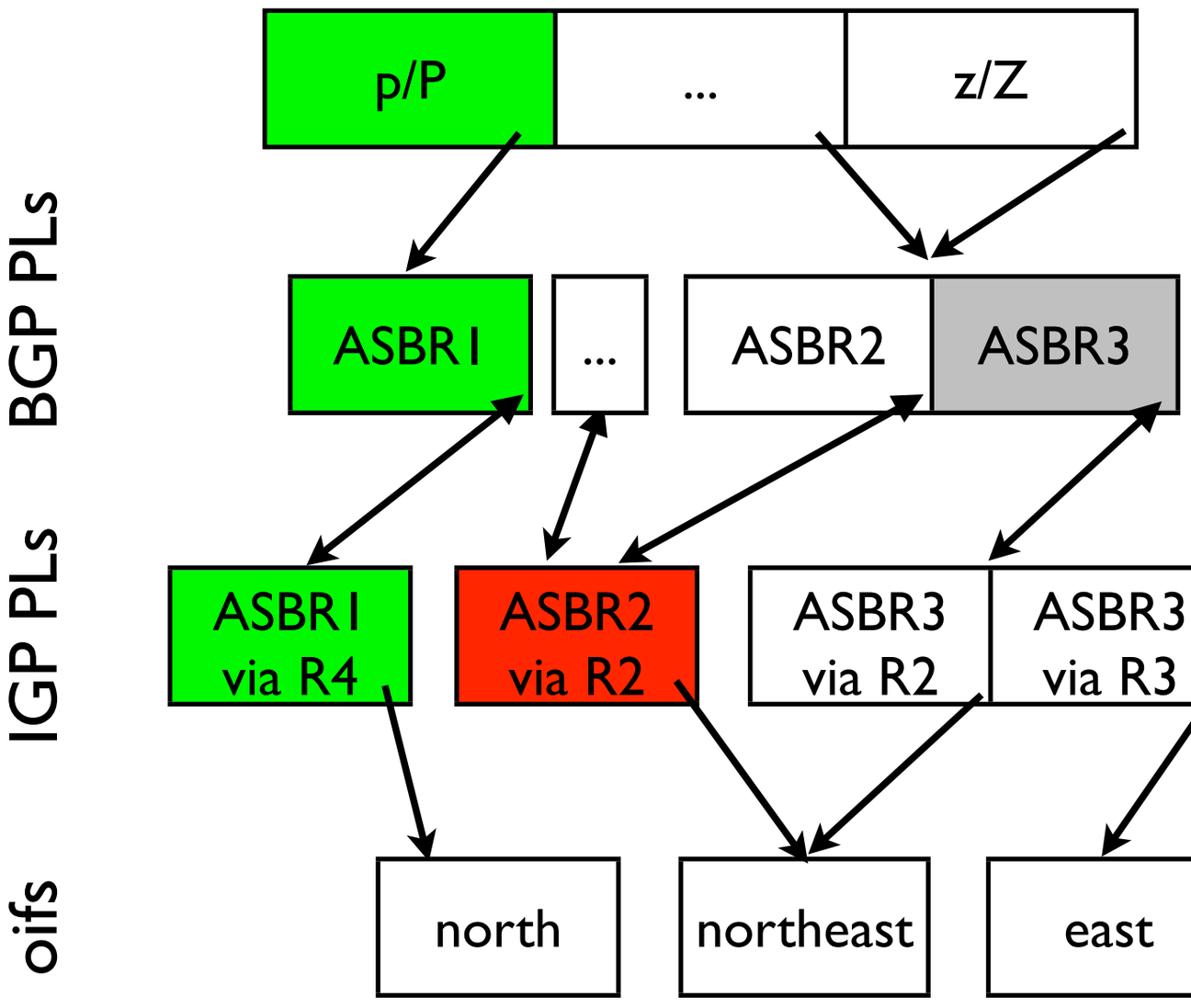
generalized FIB example (R1)

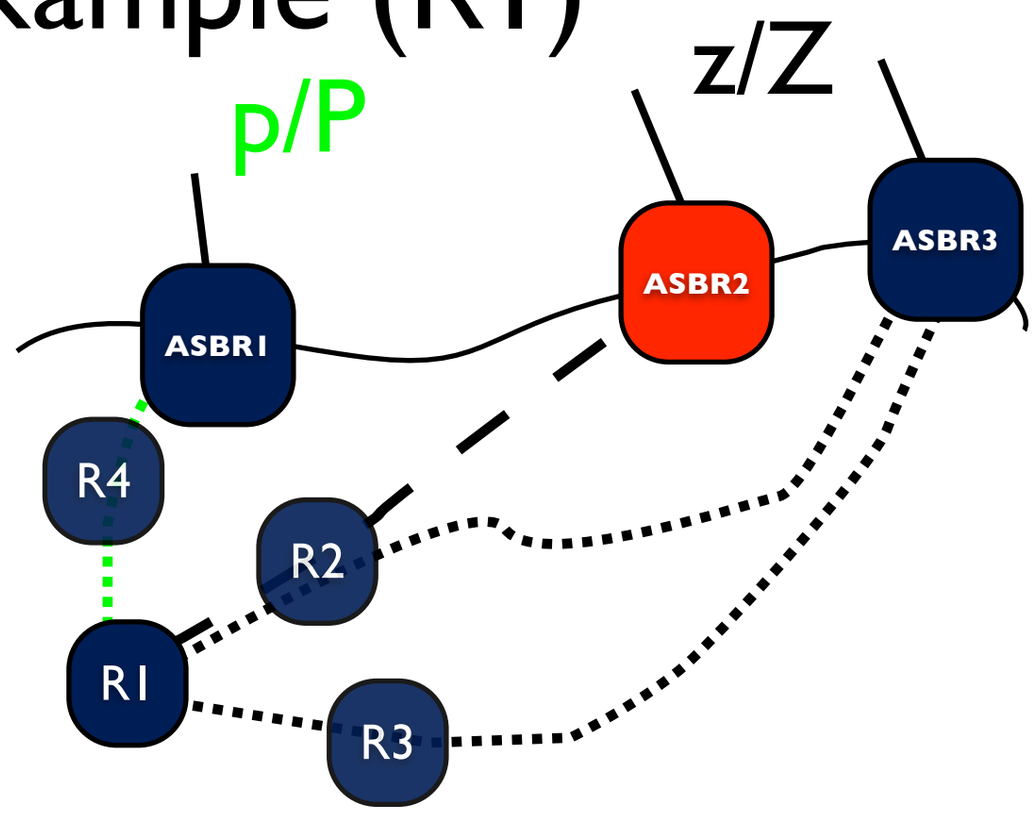




FIB design

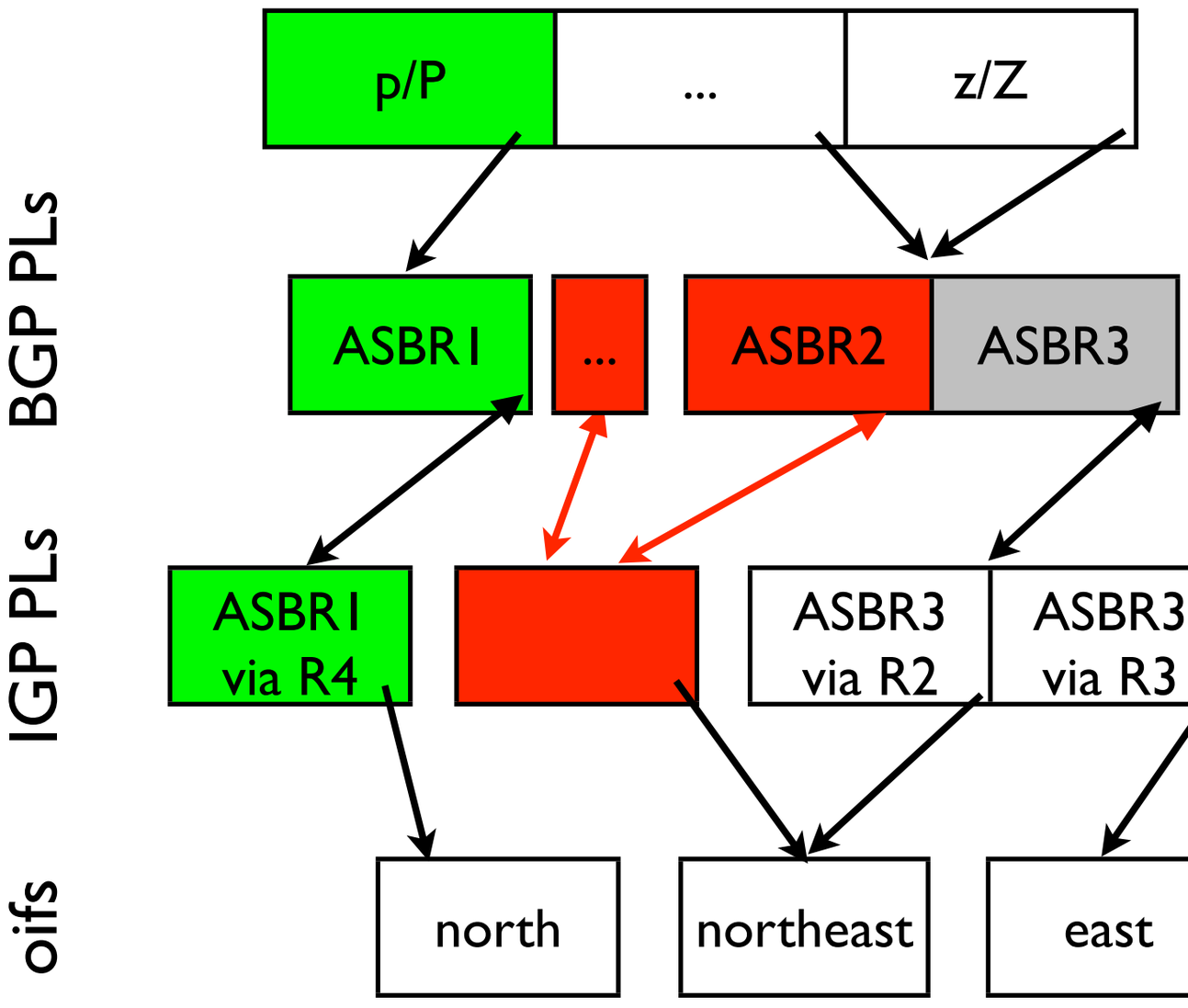
generalized FIB example (R1)

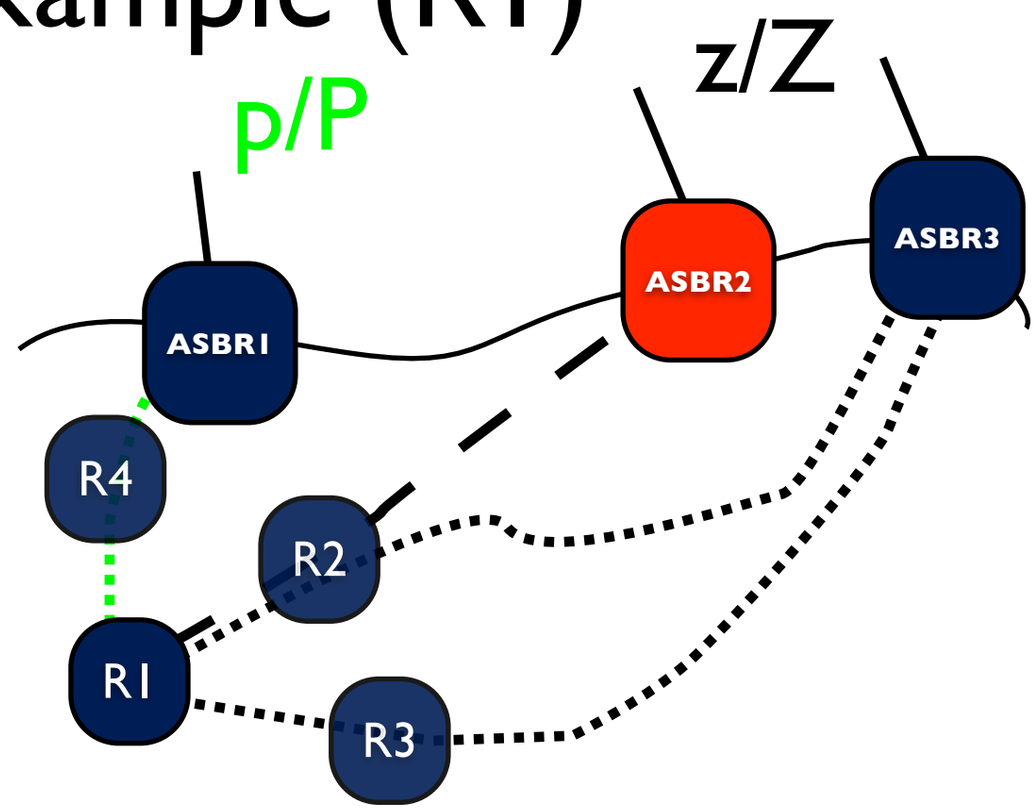




FIB design

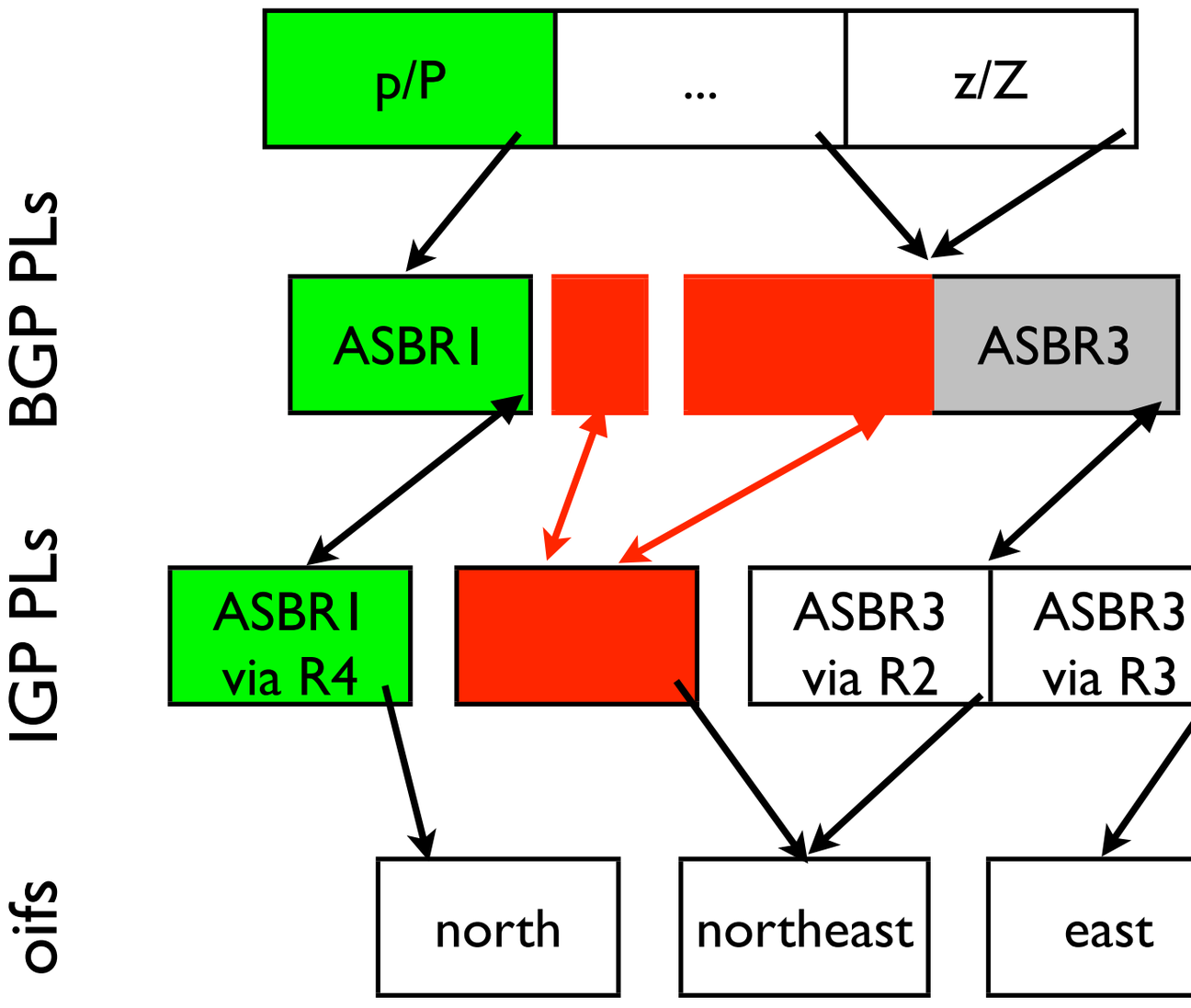
generalized FIB example (R1)

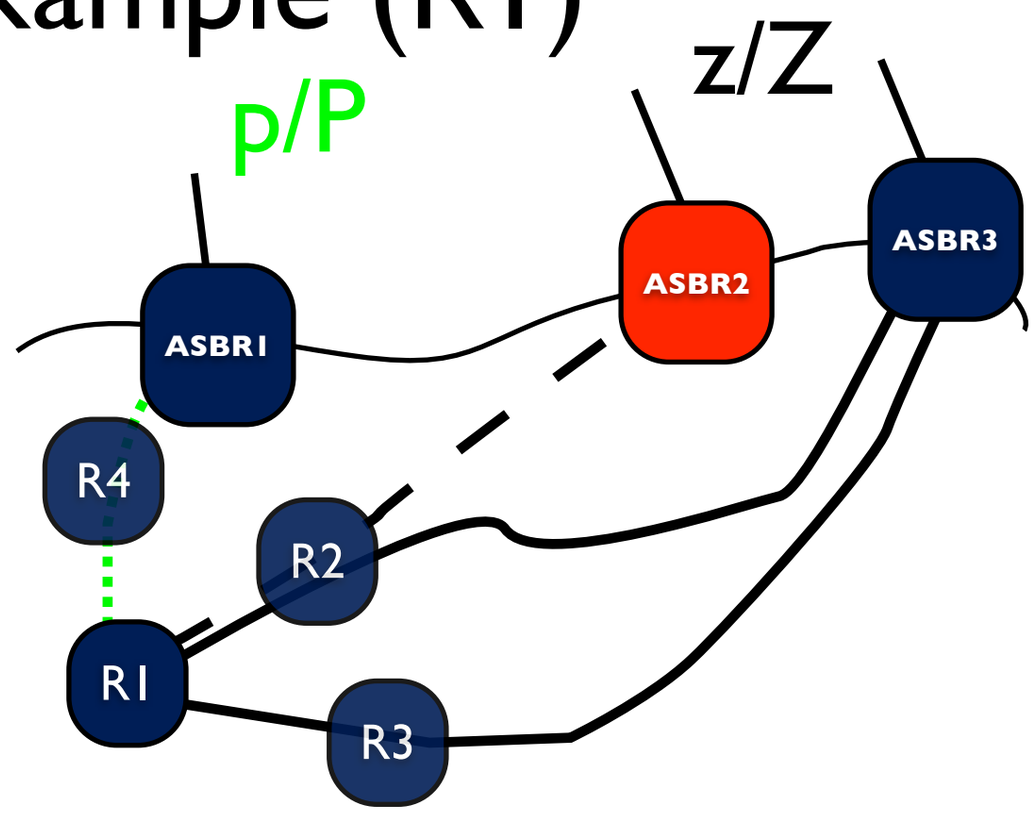




FIB design

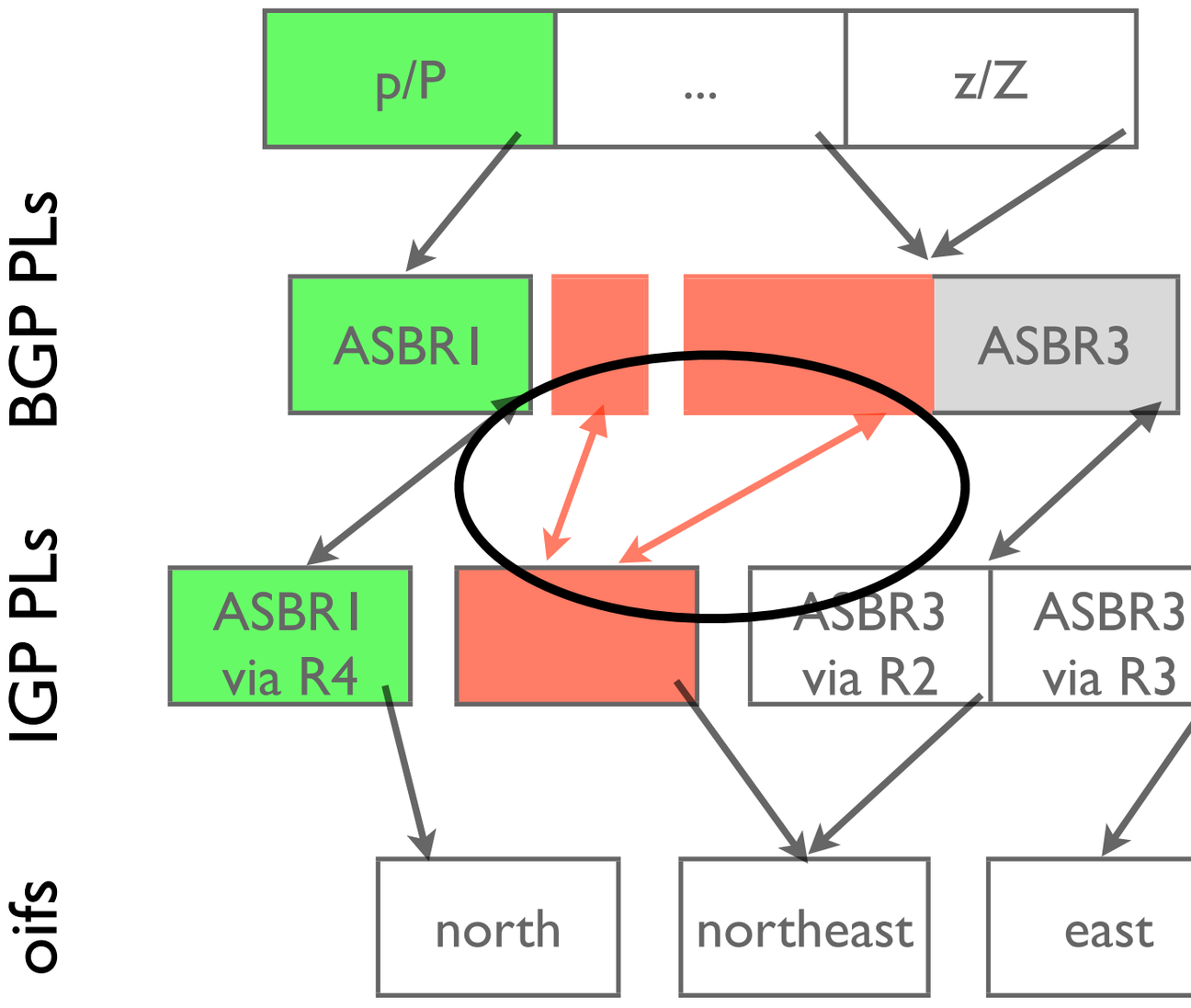
generalized FIB example (R1)

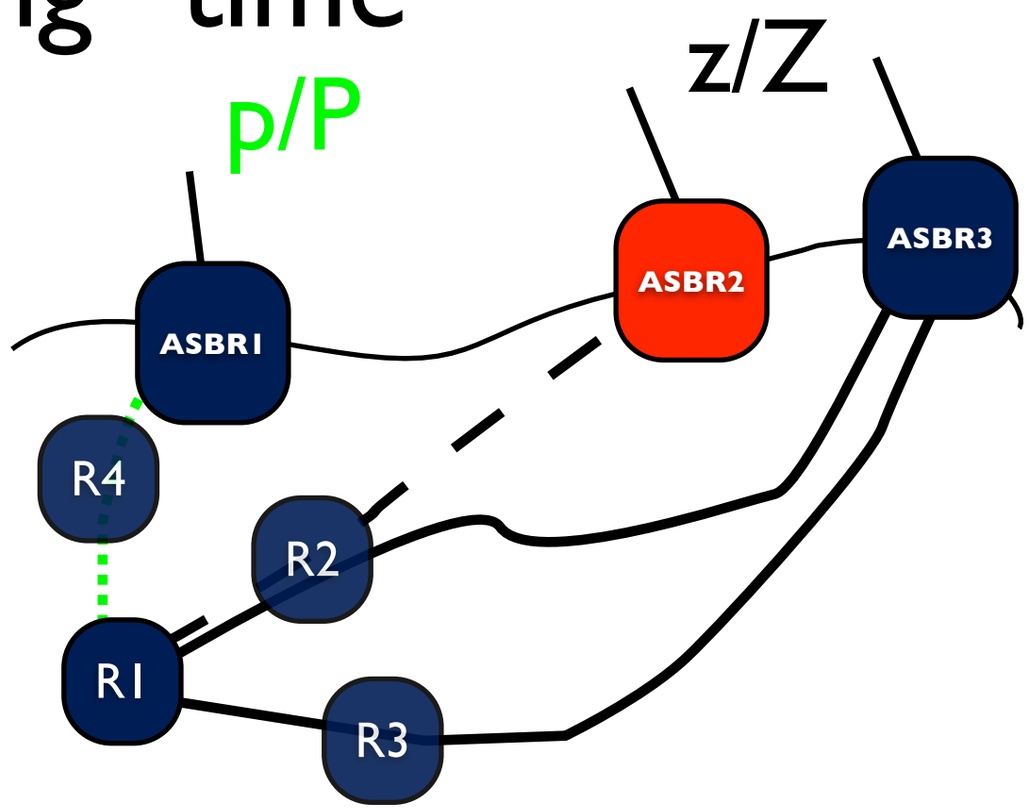




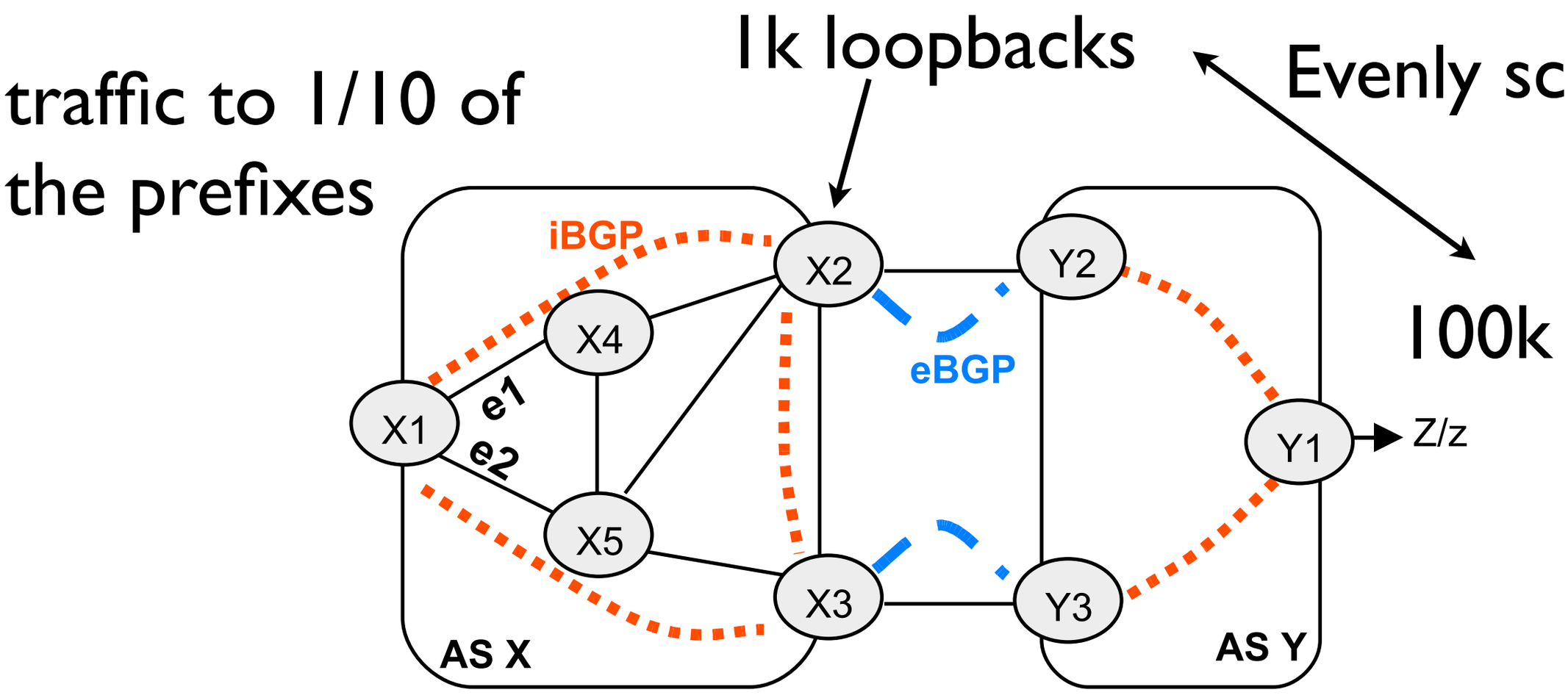
FIB design

PL "Backwalking" time



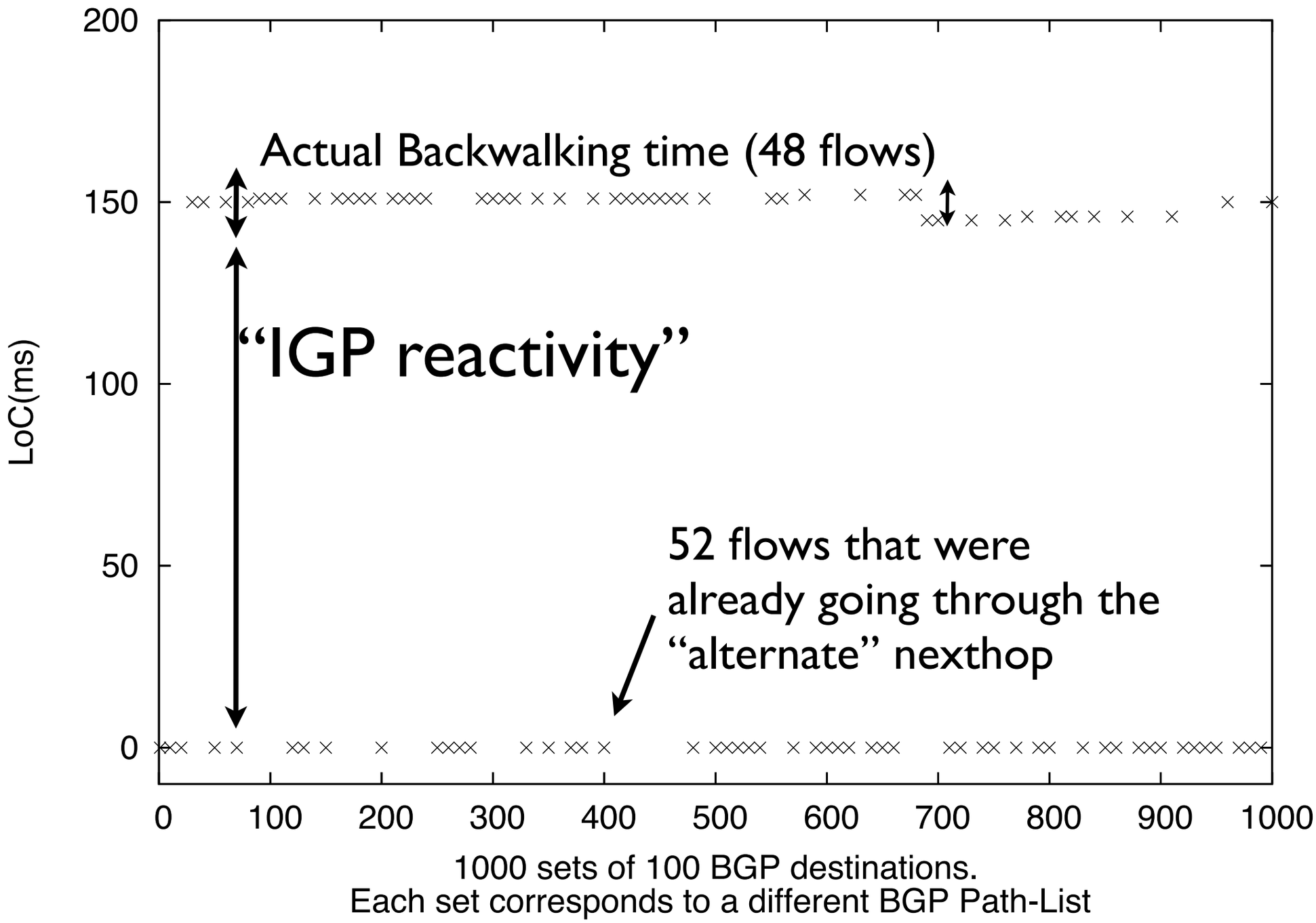


Backwalking time Testbed



FIB design

"PL Backwalking" time in a CRS-1



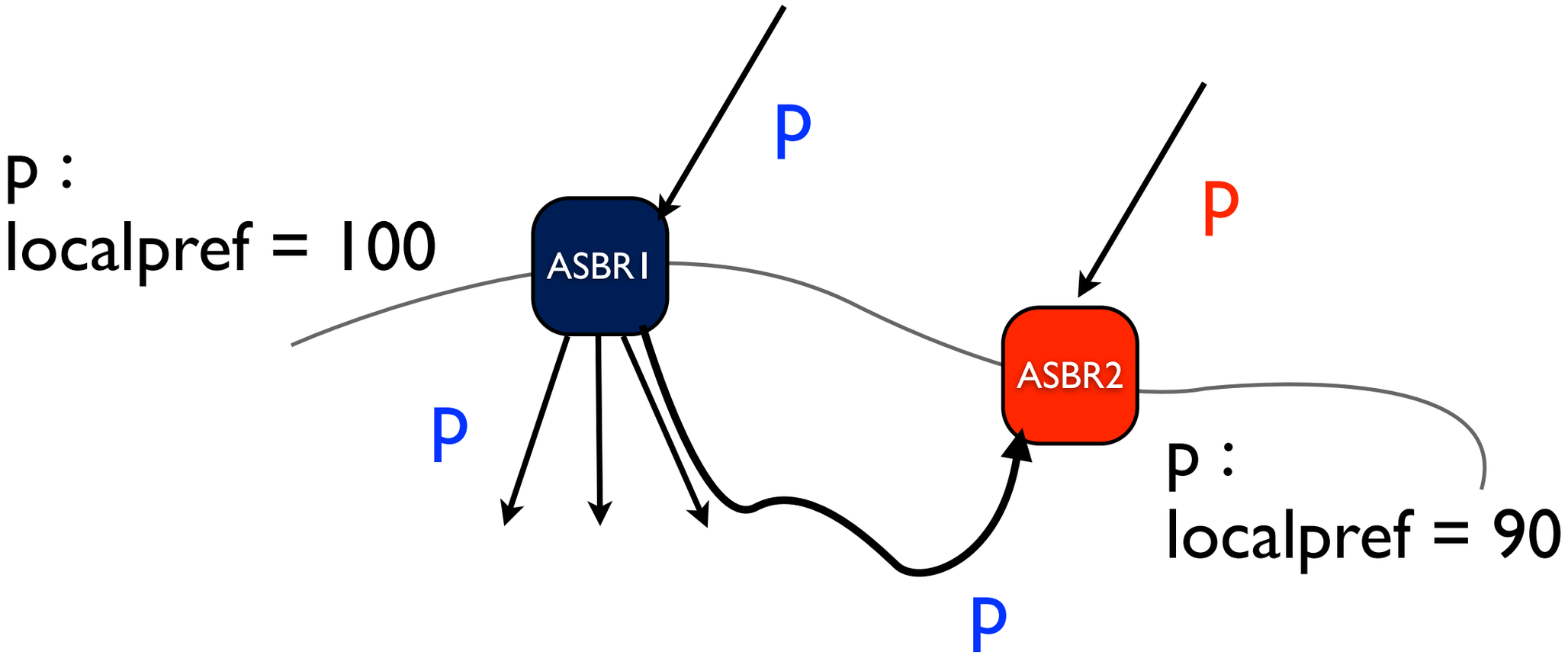
Not a futuristic talk

- BGP PIC Core/Edge is available

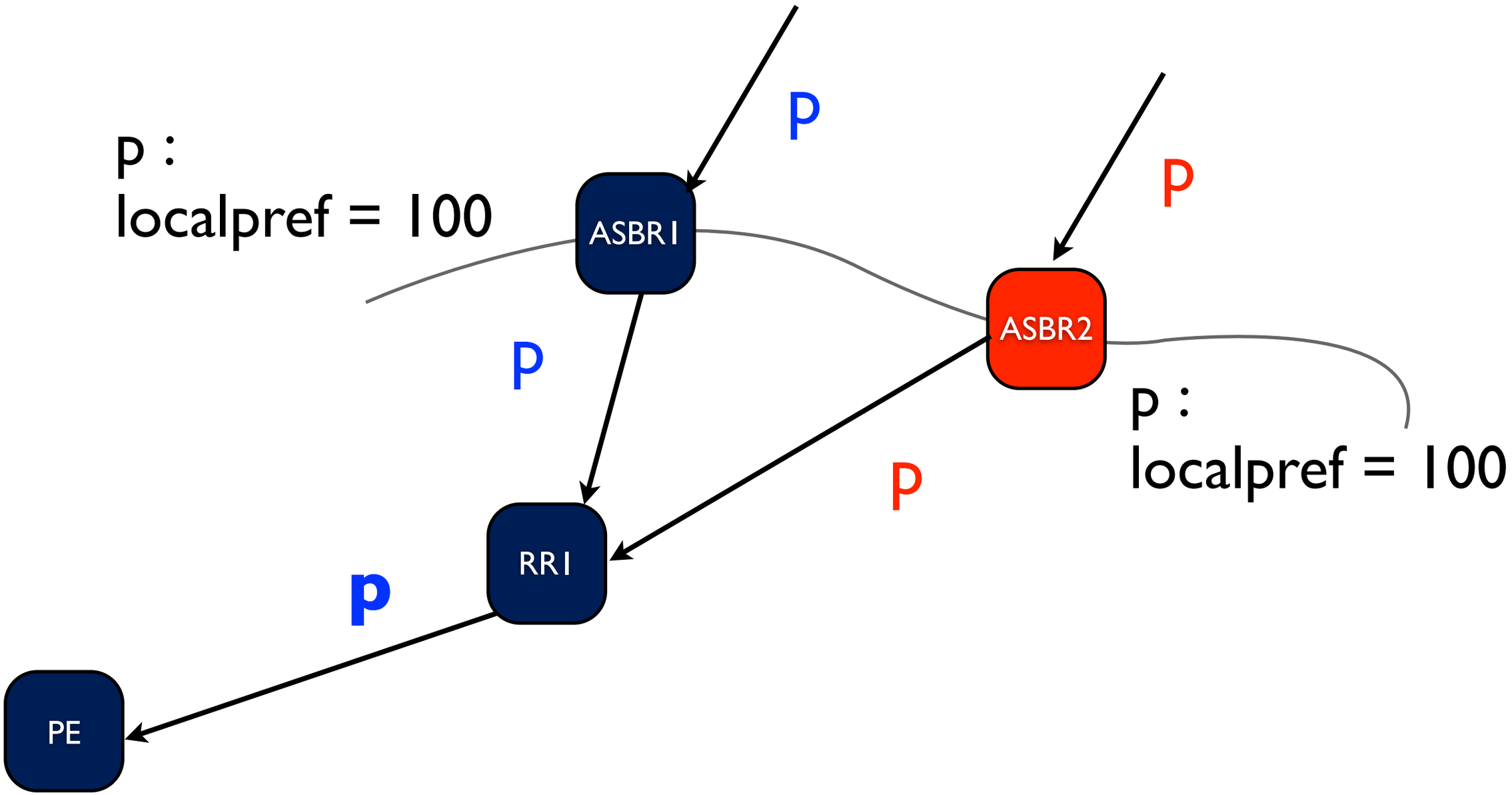
All our problems solved ?

- PIC makes data-plane convergence really fast
- Useless if no alternates to converge to...

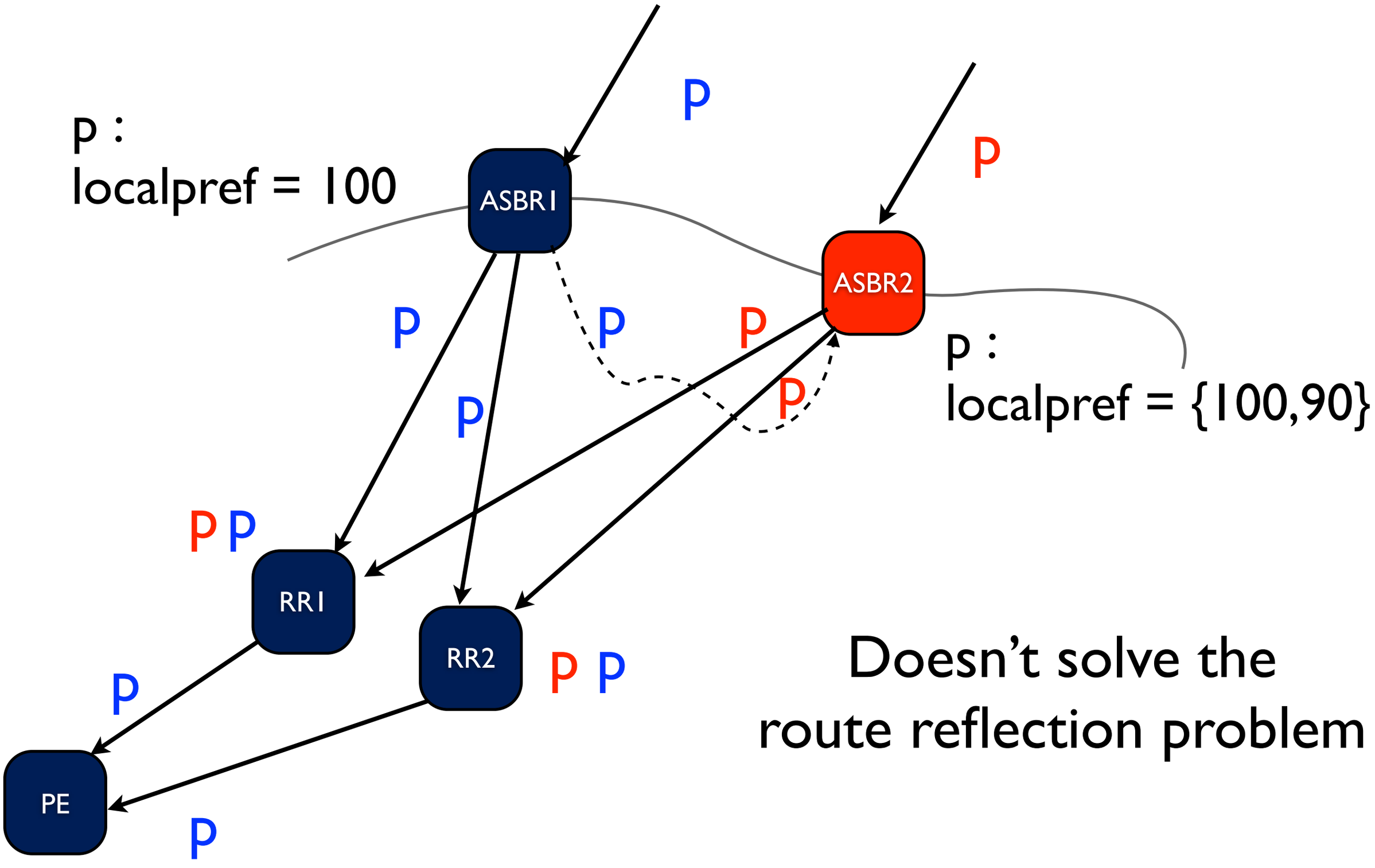
Policies let paths be hidden



Route Reflection hides paths



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

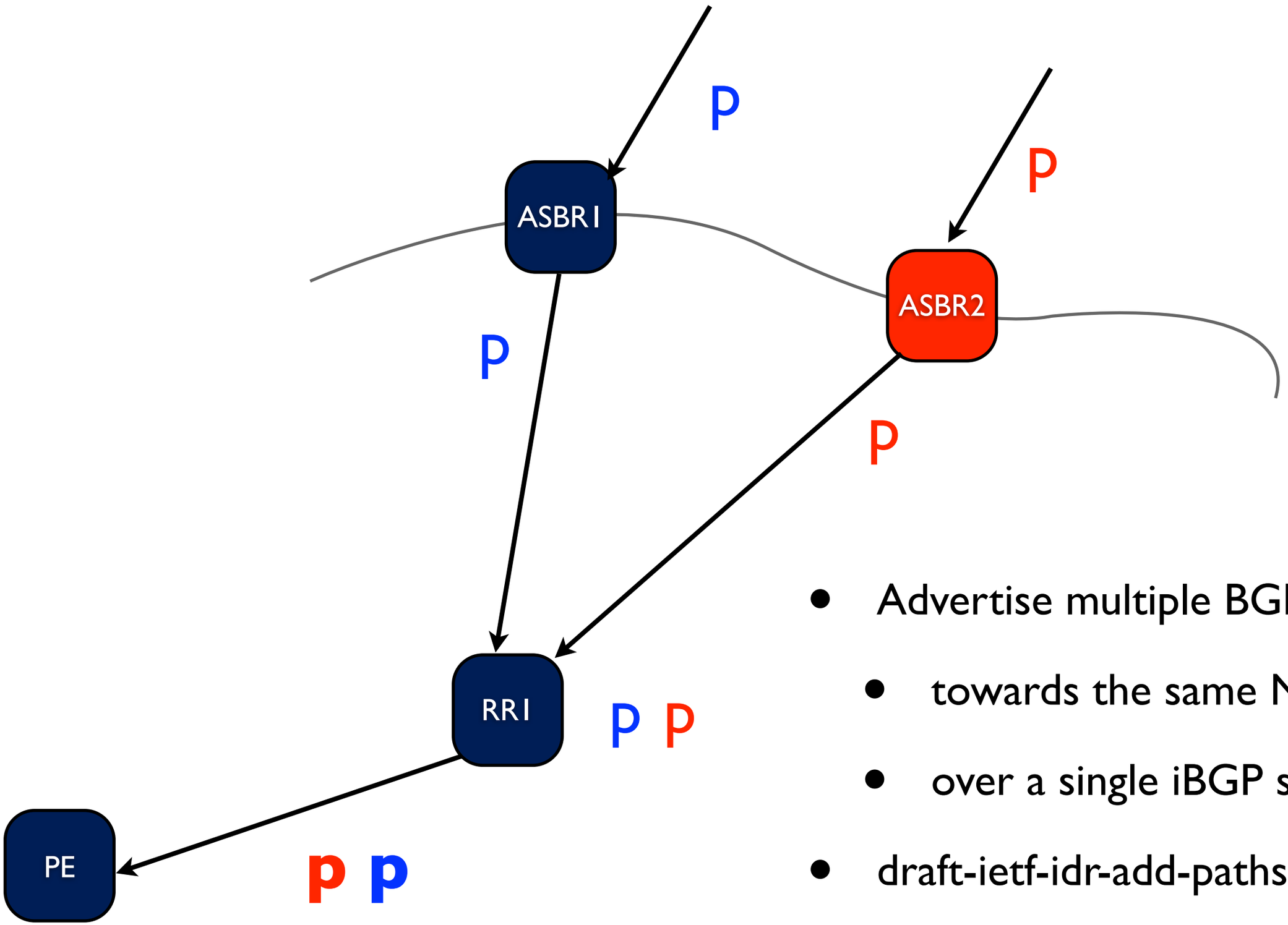


Doesn’t solve the
route reflection problem

Motivation for Add-paths

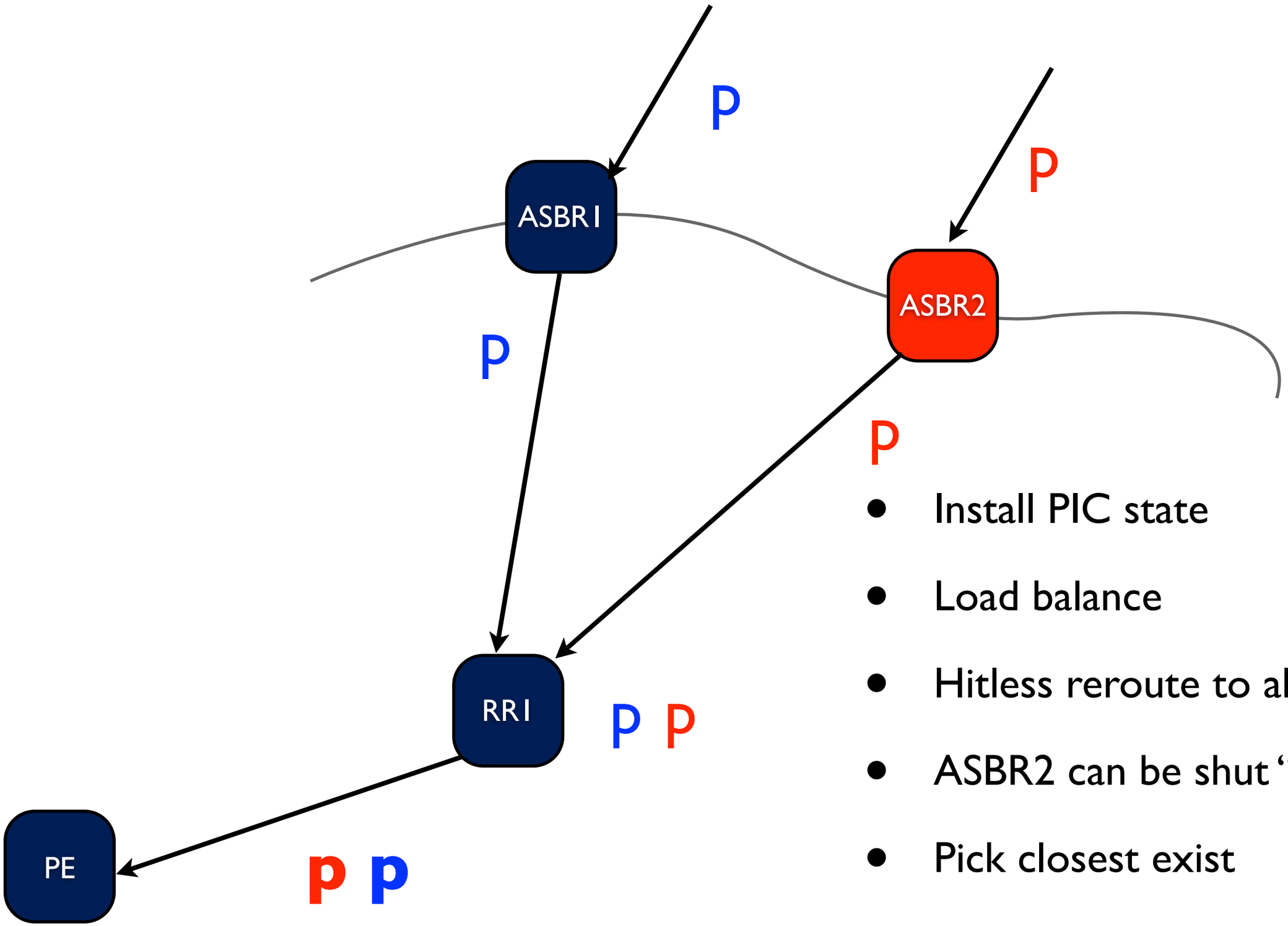
- Initial "motivation" was MED oscillation avoidance
- Emergence of new IDR requirements a few years ago
  - Fast recovery upon peering link / ASBR failure (PIC)
  - Load balancing among multiple primary BGP NHs
  - Hitless planned maintenance
  - "Optimal" hot-potato routing
  - (Churn reduction / convergence concealment)

BGP Add paths



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

BGP Add paths

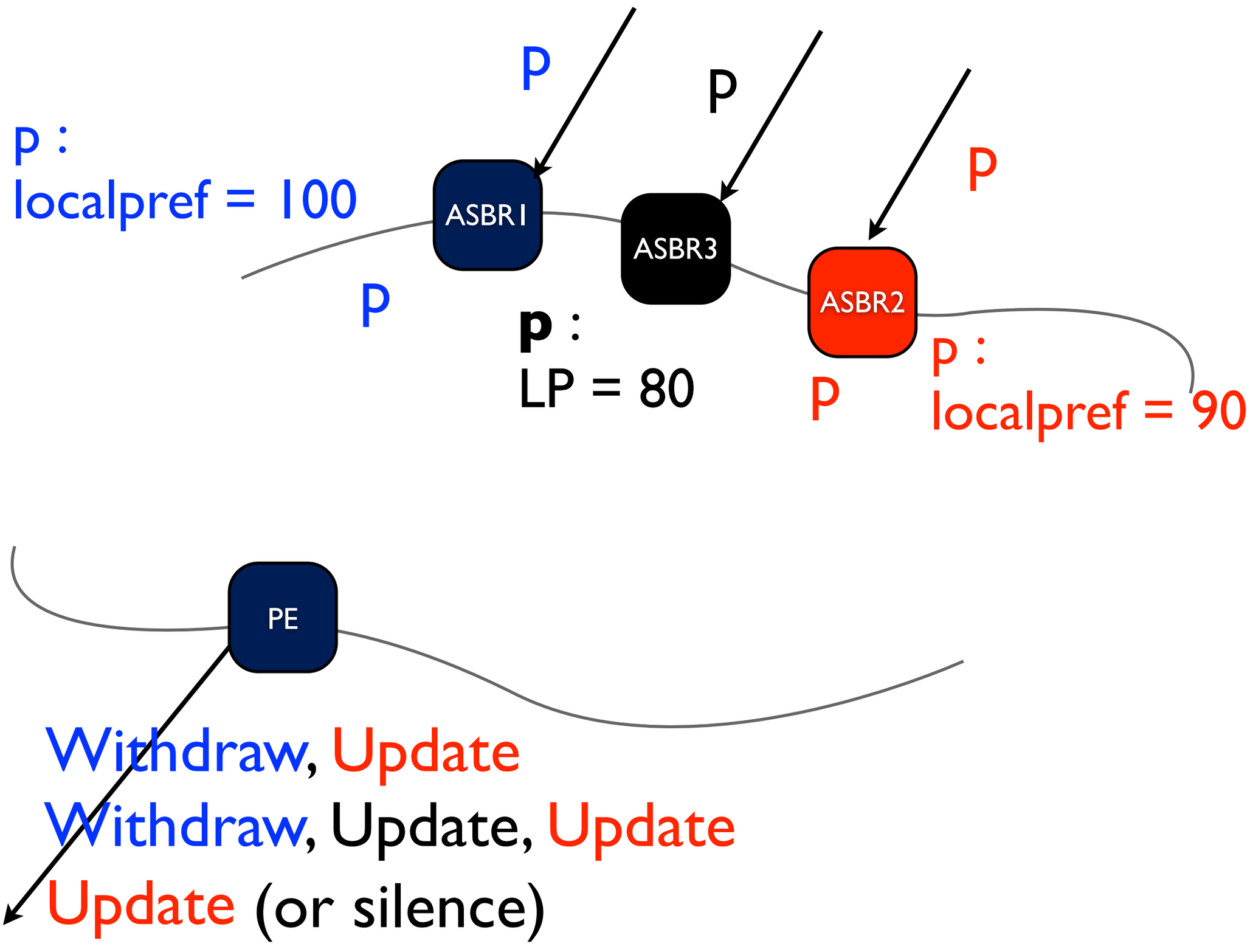


- Install PIC state
- Load balance
- Hitless reroute to alternate
- ASBR2 can be shut “hitlessly”
- Pick closest exist
- Reduced churn upon loss

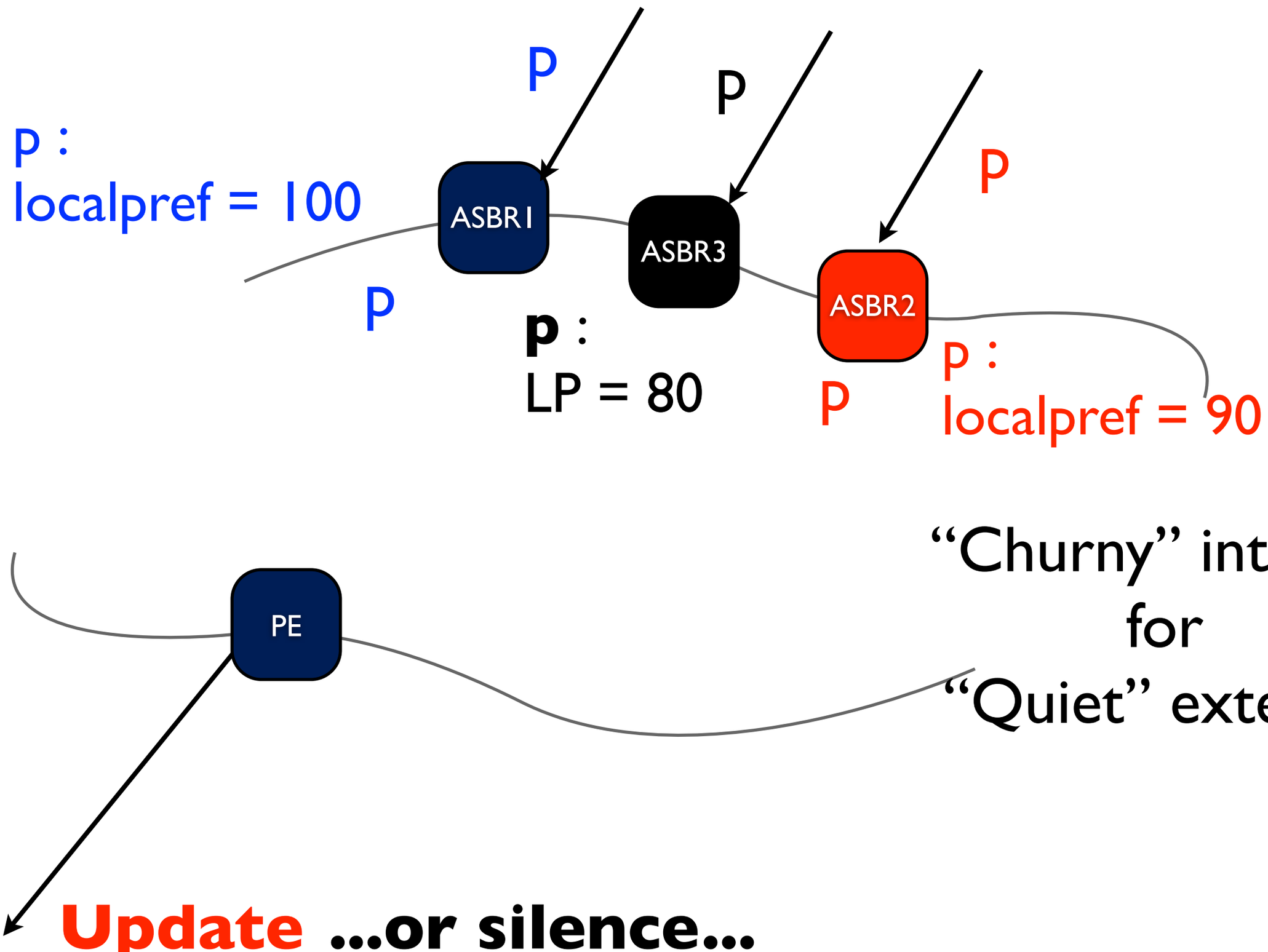
Churn reduction

- 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...

What to send ?

Application dependent

- MED oscillation ?
  - Avoid hiding lowest MED paths from neighboring ASes
- PIC ?
  - Feed all PE nodes with 2 paths
- Load-balancing ?
  - Feed all PE nodes with N paths
- Hitless maintenance
  - Never have 0 paths
- Churn reduction
  - Favor dissemination of post-convergence paths

draft-ietf-idr-add-paths

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

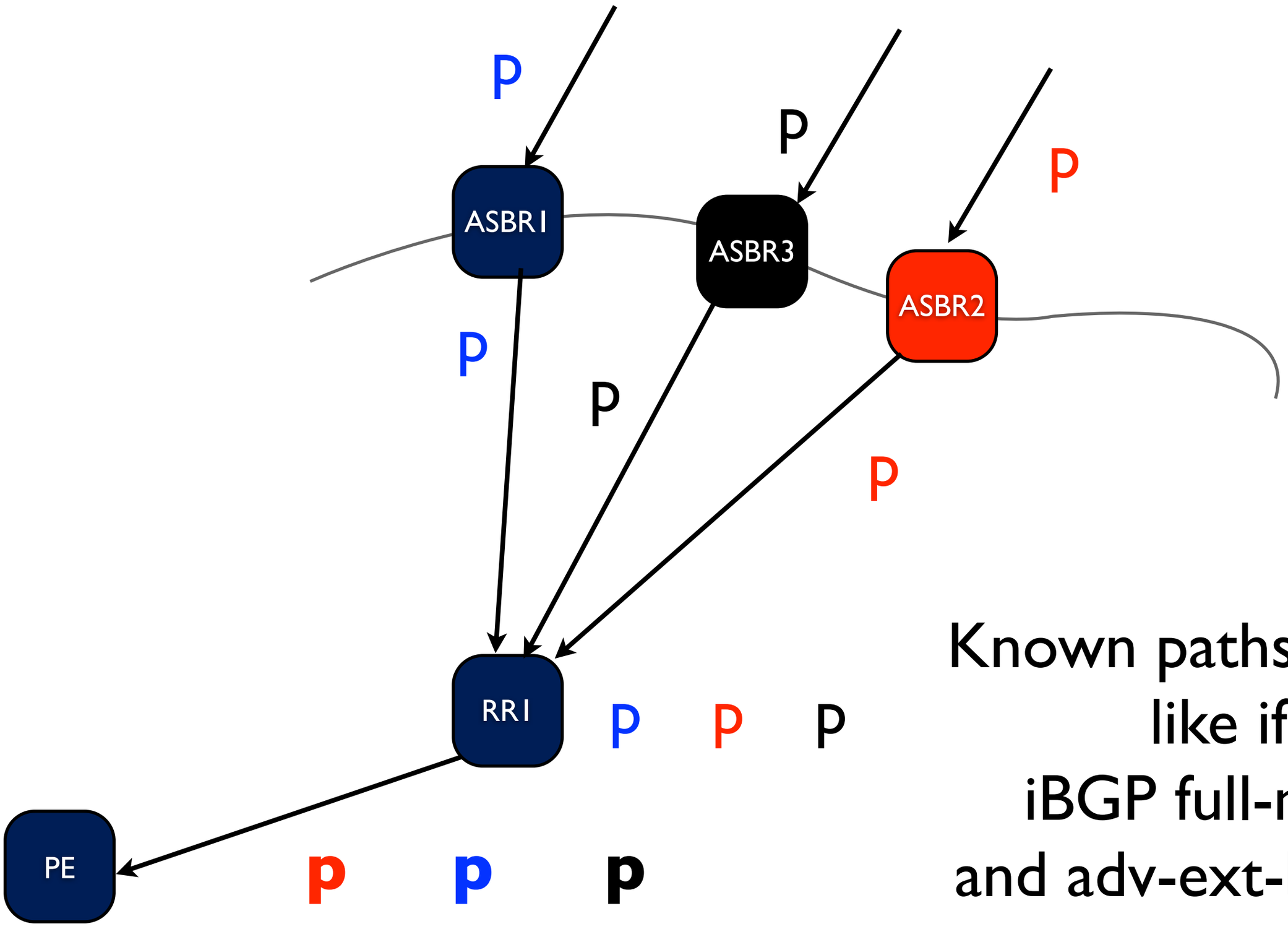
draft-ietf-idr-add-paths-guidelines

- 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)
- Neighbor-AS group best paths
- (Best Loc Pref / Second best Loc Pref paths)
- (Decisive step -1 paths)

All Paths

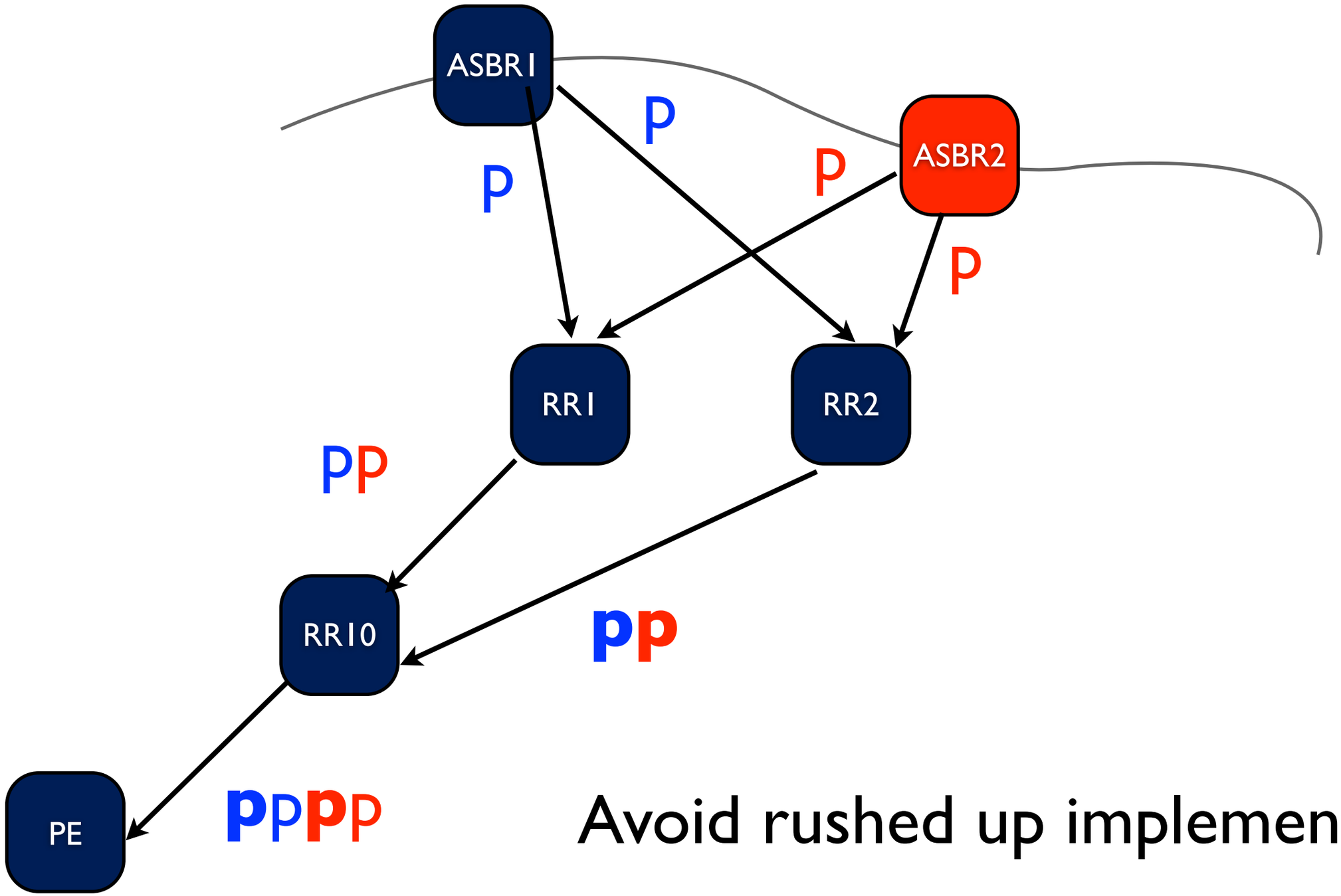


Known paths almost
like if
iBGP full-mesh
and adv-ext-best on

Add-All

- Easiest Decision Process algorithm
- Nice mode to turn on towards a BGP monitor
- Depending on how many paths for each p
  - Memory monster (control-plane)
  - Internal update churn monster

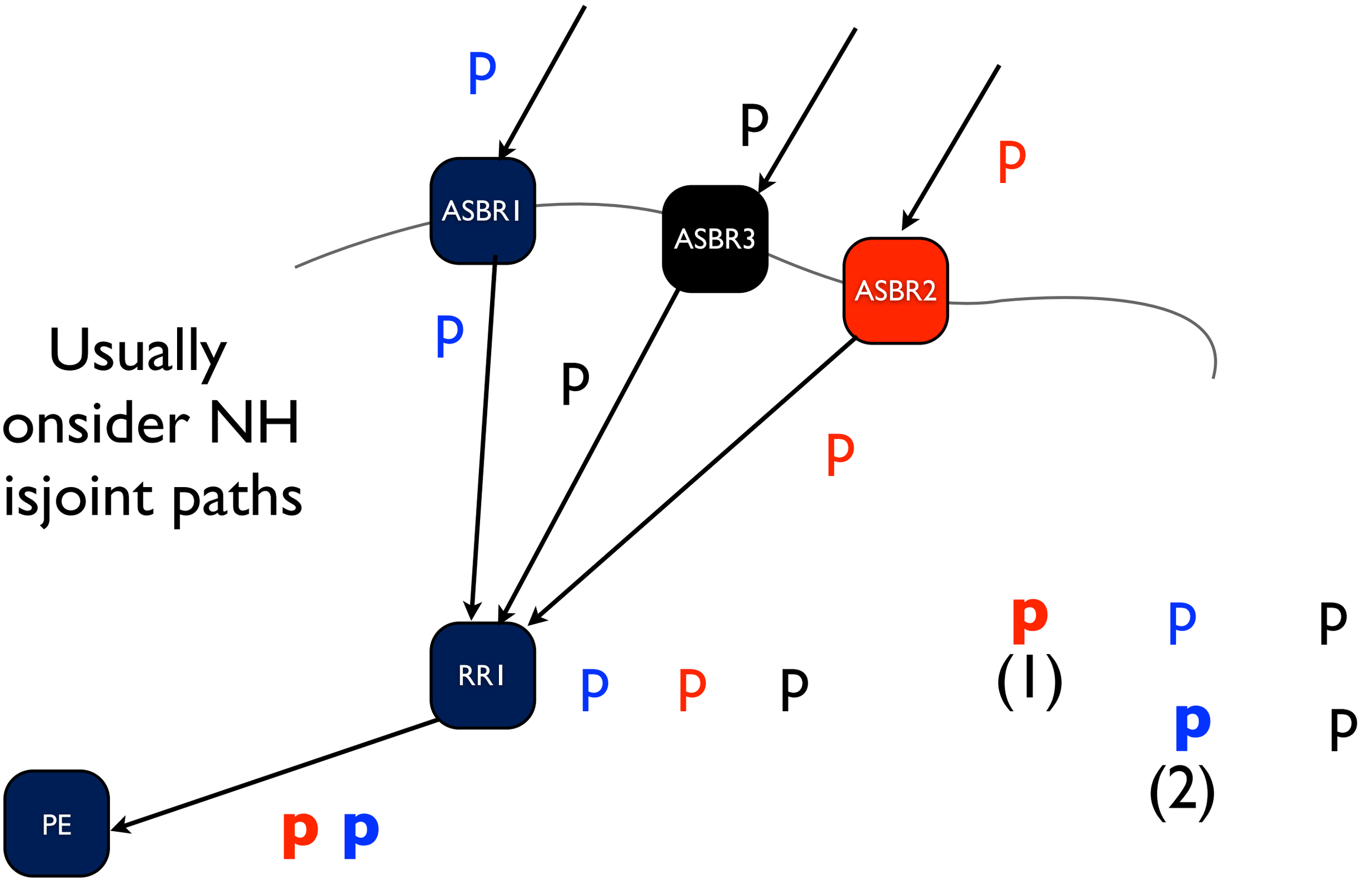
Please...



Avoid rushed up implementations

Add-N paths

Usually consider NH disjoint paths



Add-N-Paths

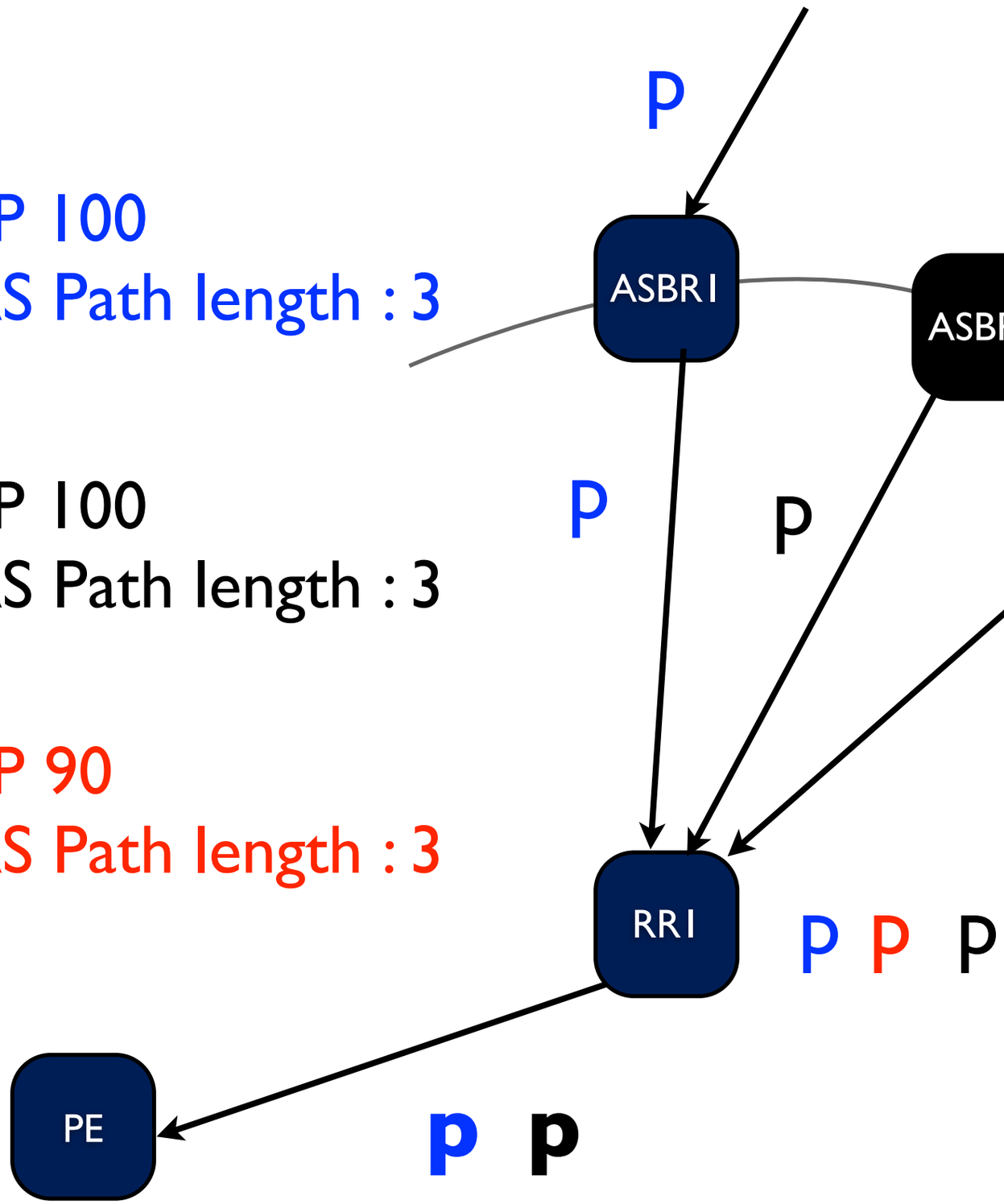
- Most practical use cases
  - Set N to 2 for basic PIC support
  - Set N to desired number of NHs for LB
- Memory hit kept under control through configuration of N
  - Configurable per AFI/SAFI (some may provide finer tuning)
- Doesn’t solve MED oscillations
- Developers tend to implement it as N*DP

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

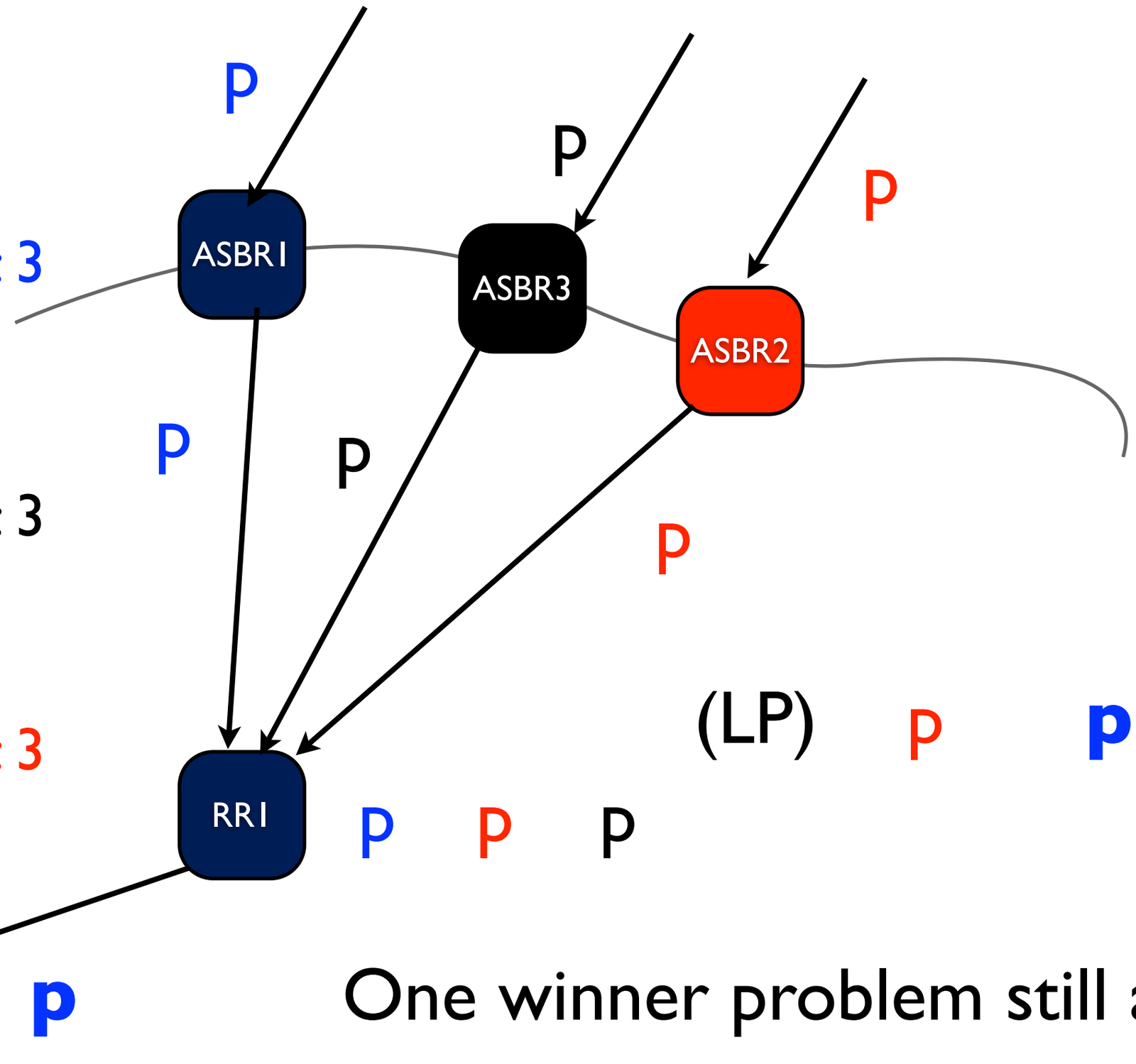
- !!! Doesn’t feed PIC !!!

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

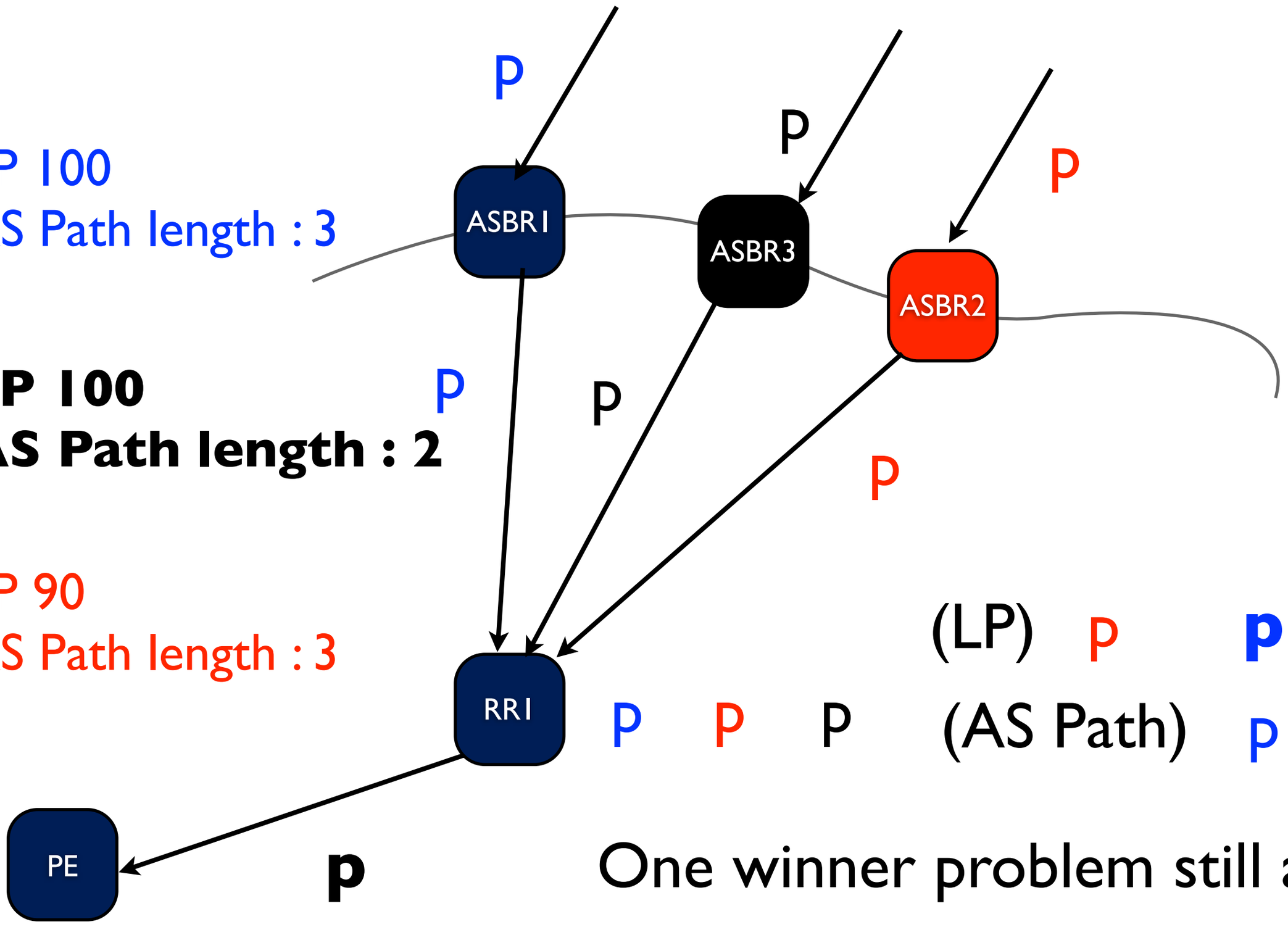
One winner problem still applies

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

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

Deployment

- Session wide upgrade required (unlike Robert's Diverse-paths)
- As for all solutions
  - Forget about deployments w/o Ingress-Egress encap
  - Transient forwarding loops if naïve PIC implementation

Tool

- Evaluate the behavior of Add-paths in YOUR network
- Analytical
- Numbers

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

Next Steps

- Improvement of capability negotiation ?
  - Currently send/receive bits
  - Might want to make this a bit more expressive
    - Announce N to route reflector ?
    - Community based modes
    - Advertise which mode the RR should run ???

Thanks !