The Case for Separating Routing from Routers

Nick Feamster, Hari Balakrishnan

M.I.T. Computer Science and Artificial Intelligence Laboratory

Jennifer Rexford, Aman Shaikh, Kobus van der Merwe

AT&T Labs -- Research

You’ve Probably Heard the News

"BGP is broken."

- It might not converge.
- When it converges, it does so slowly.
- It causes routing loops inside an AS.
- It’s misconfigured frequently.
- Routing tables are getting huge!

"We can’t fix the problems."

- BGP is hard-coded into routers.
- It’s dictated by slow-moving standards.
- No flag days!

BGP's Problems Have Scared Us Away

"BGP is broken."

- It might not converge.
- When it converges, it does so slowly.
- It causes routing loops inside an AS.
- It's misconfigured frequently.
- Routing tables are getting huge!

What to do?

- Delve into BGP-specific, esoteric arcana
  - Discover more negative results
  - Incremental fixes that make BGP even harder to understand!
- Design idealistic architectures
-

These Problems Can Be Fixed

"BGP is broken."

- It might not converge.
- When it converges, it does so slowly.
- It causes routing loops inside an AS.
- It’s misconfigured frequently.
- Routing tables are getting huge!

What’s causing these problems?

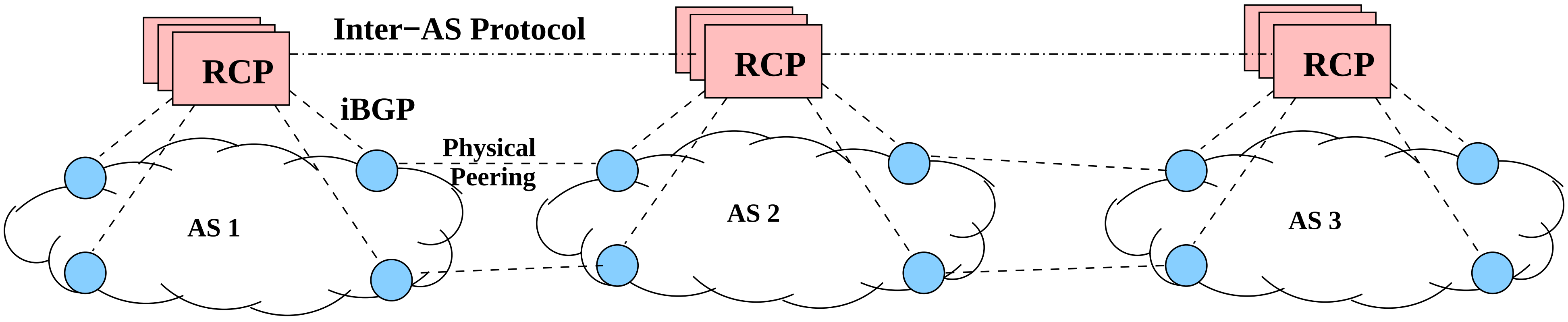
- Each router has limited, inconsistent state
- BGP interacts in odd ways with other protocols

Problems result from placing too much logic in the routers.

Our Vision: A "Routing Control Platform"

Routers do not compute routes!

- Route computation for an AS is offloaded to a system with a complete view of network state.
- Each AS has a "server" that exchanges consistent routing information with other ASes



The rest of this talk: The Case for RCP

Principles for interdomain routing:

- Compute consistent routes using complete state.
  - Example: high-level policy expression
- Control routing protocol interactions.
  - Example: interactions between BGP and lower-level protocols

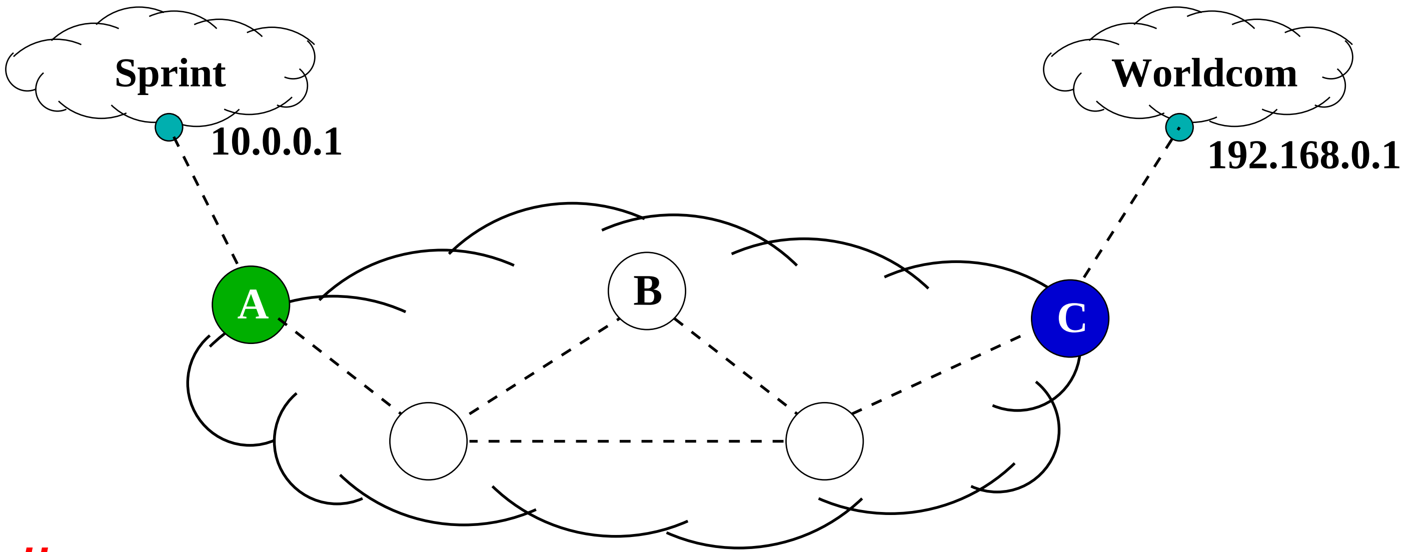
Potential dealbreakers:

- Backwards compatibility and incentives
- Scalability and reliability goals

Related work (or..."haven't we seen this before?"):

- Route reflection and route servers
- Overlay networks

Routers have inconsistent configuration state

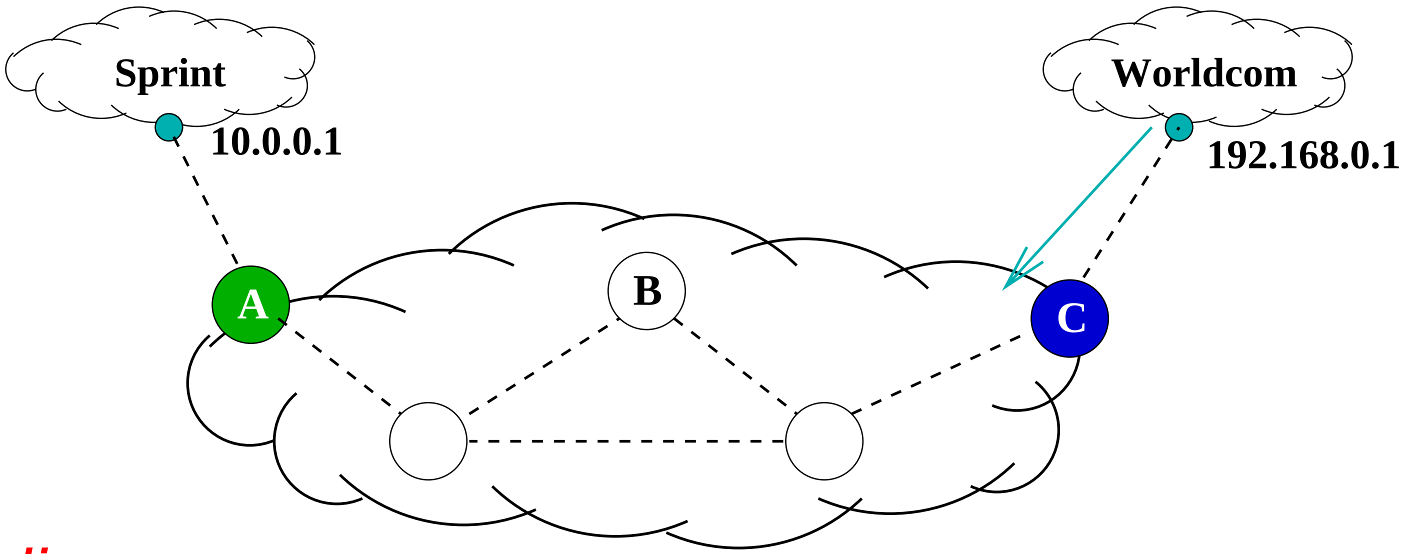


Simple Policy:

"Don’t advertise routes learned from Worldcom to Sprint."

Configuration is decomposed, so the route must carry state!

Configuration decomposed across routers



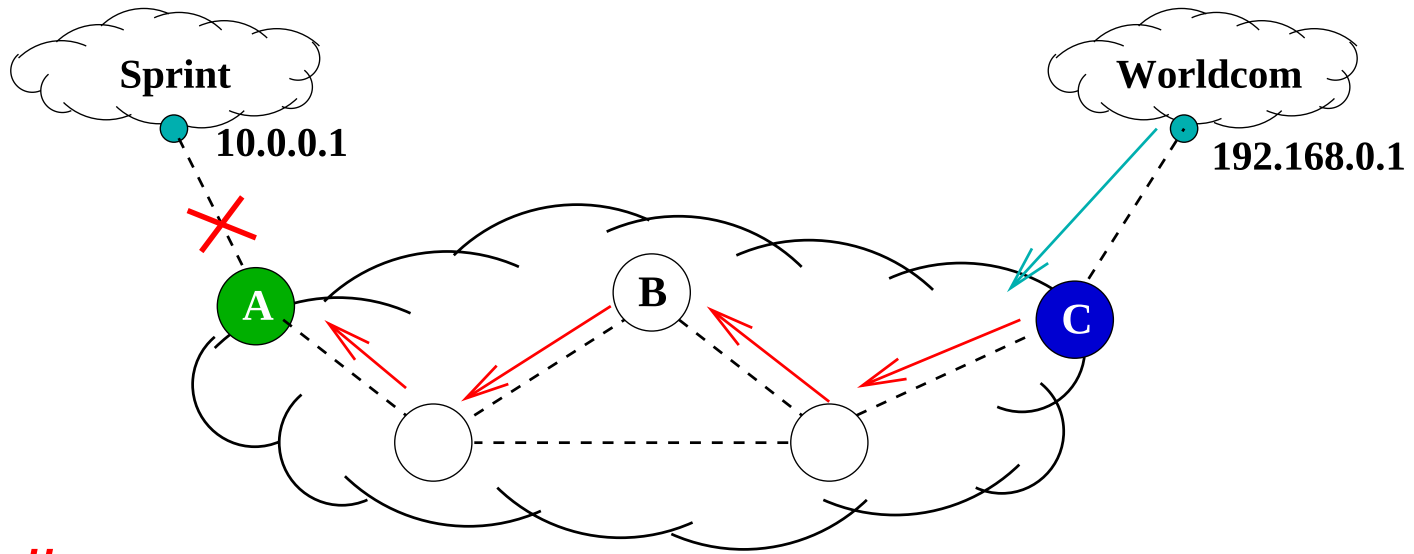
Simple Policy:

"Don't advertise routes learned from Worldcom to Sprint."

Configuration is decomposed, so the route must carry state!

```
neighbor 192.168.0.1 route-map IMPORT-C in
route-map IMPORT-C permit 10
  set community 0:1000
!
```

Configuration decomposed across routers



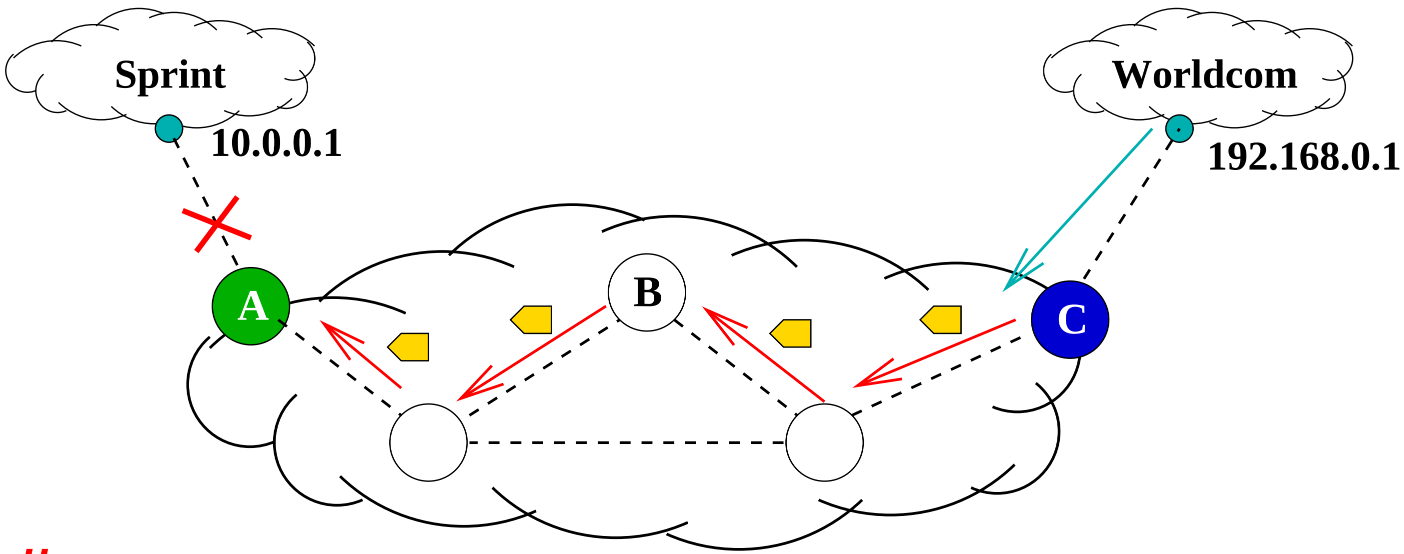
Simple Policy:

"Don't advertise routes learned from Worldcom to Sprint."

Configuration is decomposed, so the route must carry state!

```
neighbor 192.168.0.1 route-map IMPORT-C in
route-map IMPORT-C permit 10
  set community 0:1000
!..
ip community-list 1 permit 0:1000
neighbor 10.0.0.1 route-map EXPORT-A out
route-map EXPORT-A deny 10
  match community 1
!..
```

Configuration decomposed across routers



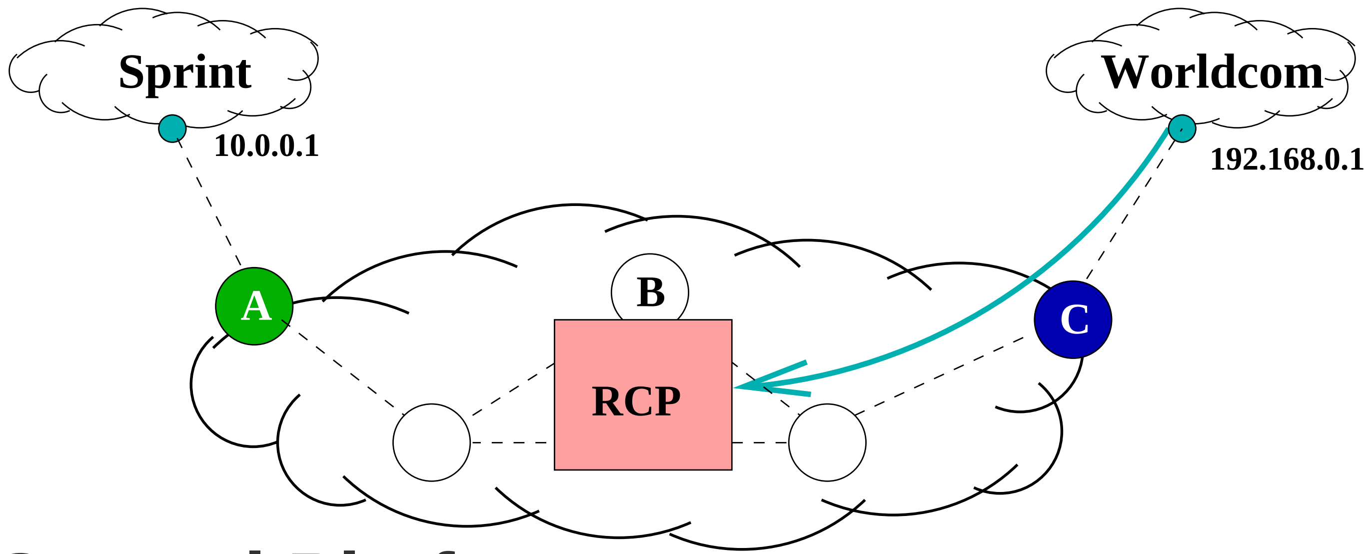
Simple Policy:

"Don't advertise routes learned from Worldcom to Sprint."

Configuration is decomposed, so the route must carry state!

```
neighbor 192.168.0.1 route-map IMPORT-C in
route-map IMPORT-C permit 10
  set community 0:1000
!..
ip community-list 1 permit 0:1000
neighbor 10.0.0.1 route-map EXPORT-A out
route-map EXPORT-A deny 10
  match community 1
!..
```

Centralize configuration state



Routing Control Platform:

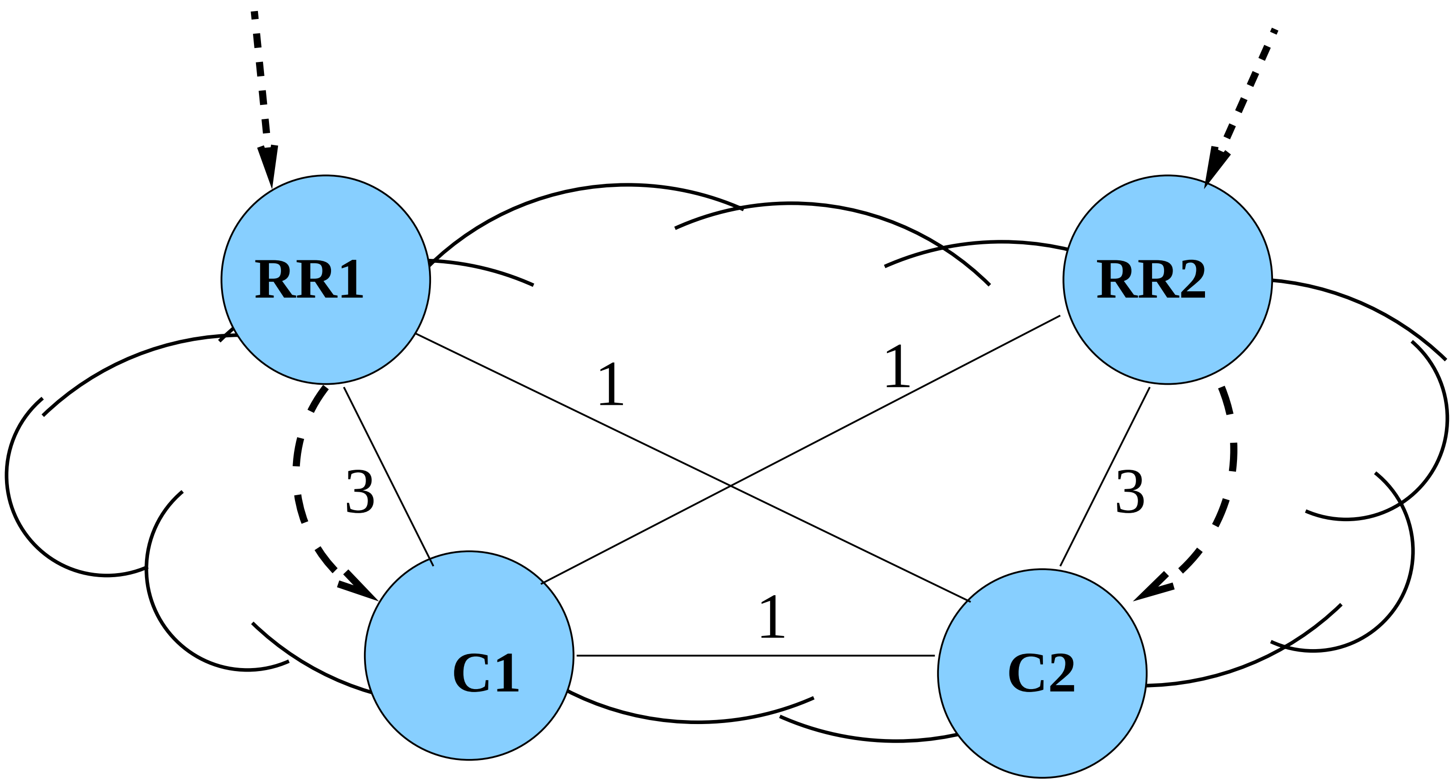
- Has views of all sessions to other ASes.
- Implements policy in terms of AS relationship

(RCP has policy configuration that expresses the constraint directly.)

Benefits

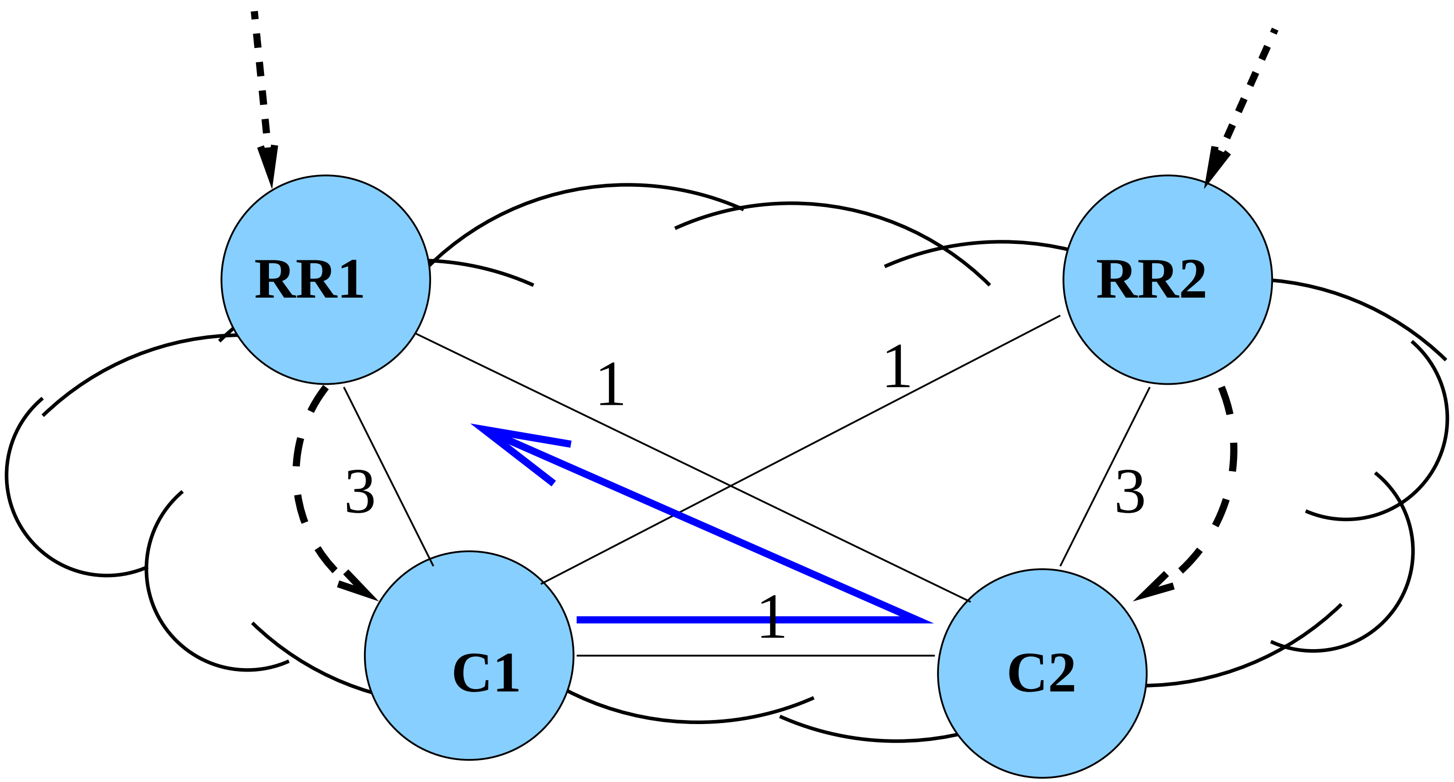
- Simpler configuration
  - separates policy and mechanism
- Don’t have to "tag" routes with state

BGP interacts with underlying protocols



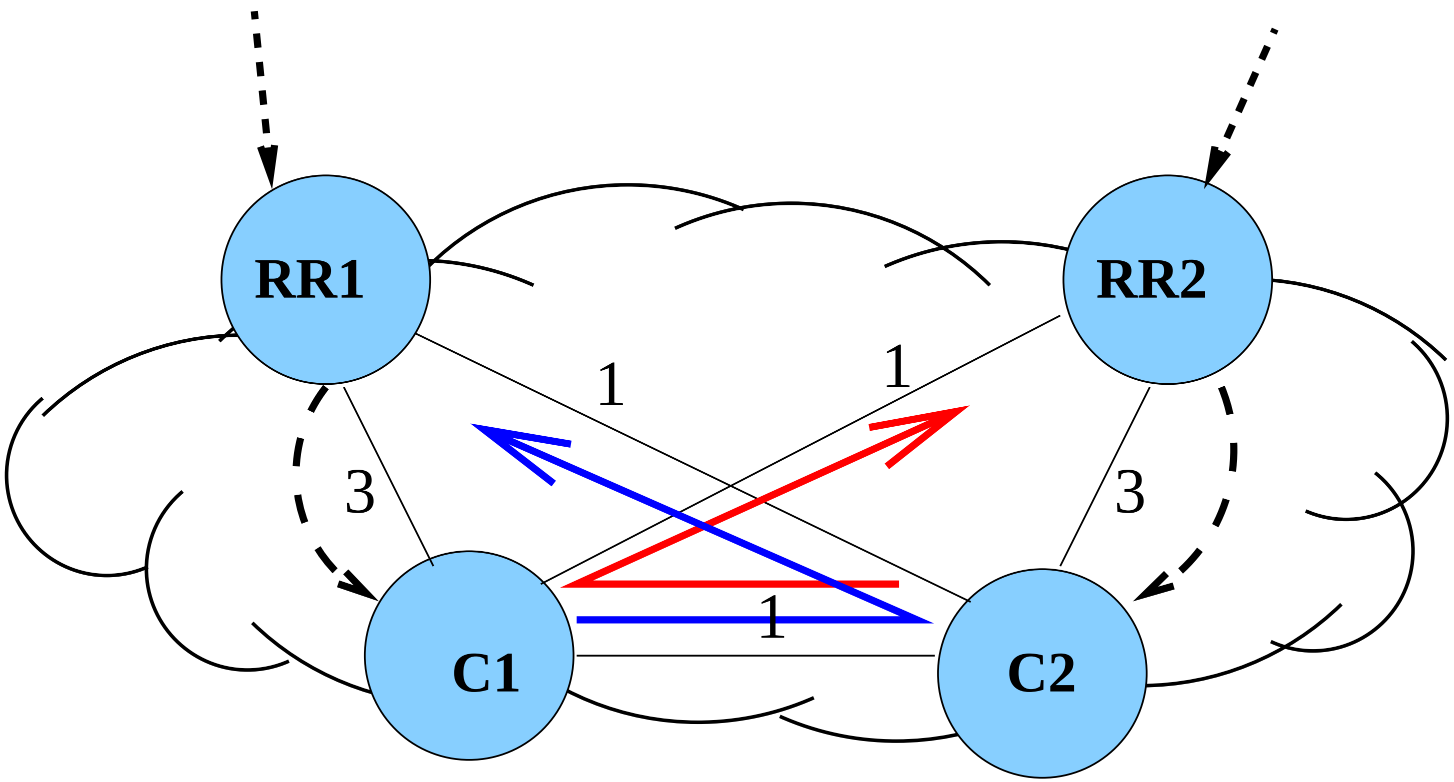
C1 learns BGP route to destination from RR1.
C2 learns BGP route to destination from RR2.

BGP interacts with underlying protocols



C1 sends packets to RR1 via its shortest path.
That path traverses C2.

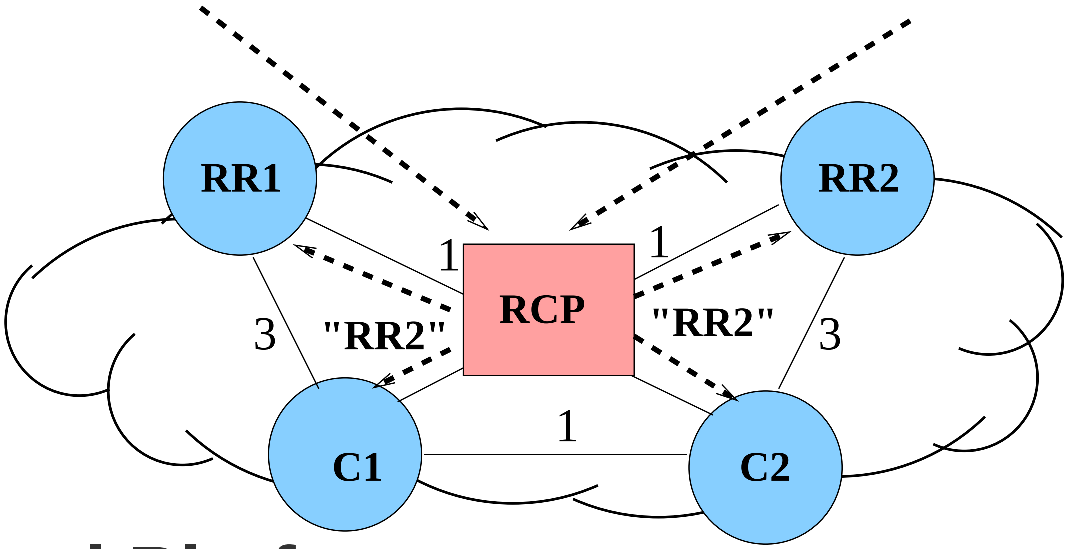
BGP interacts with underlying protocols



C2 sends packets to RR2 via its shortest path.
That path traverses C1.

Persistent forwarding loop!

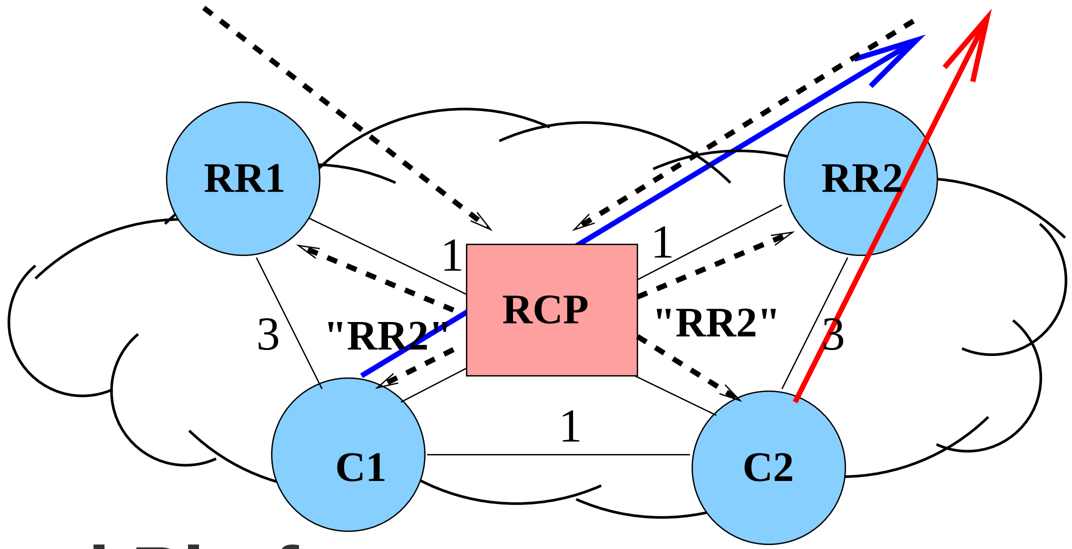
Compute routes with complete information



Routing Control Platform:

- Learns all externally learned routes
- Computes consistent router-level *paths*

Compute routes with complete information



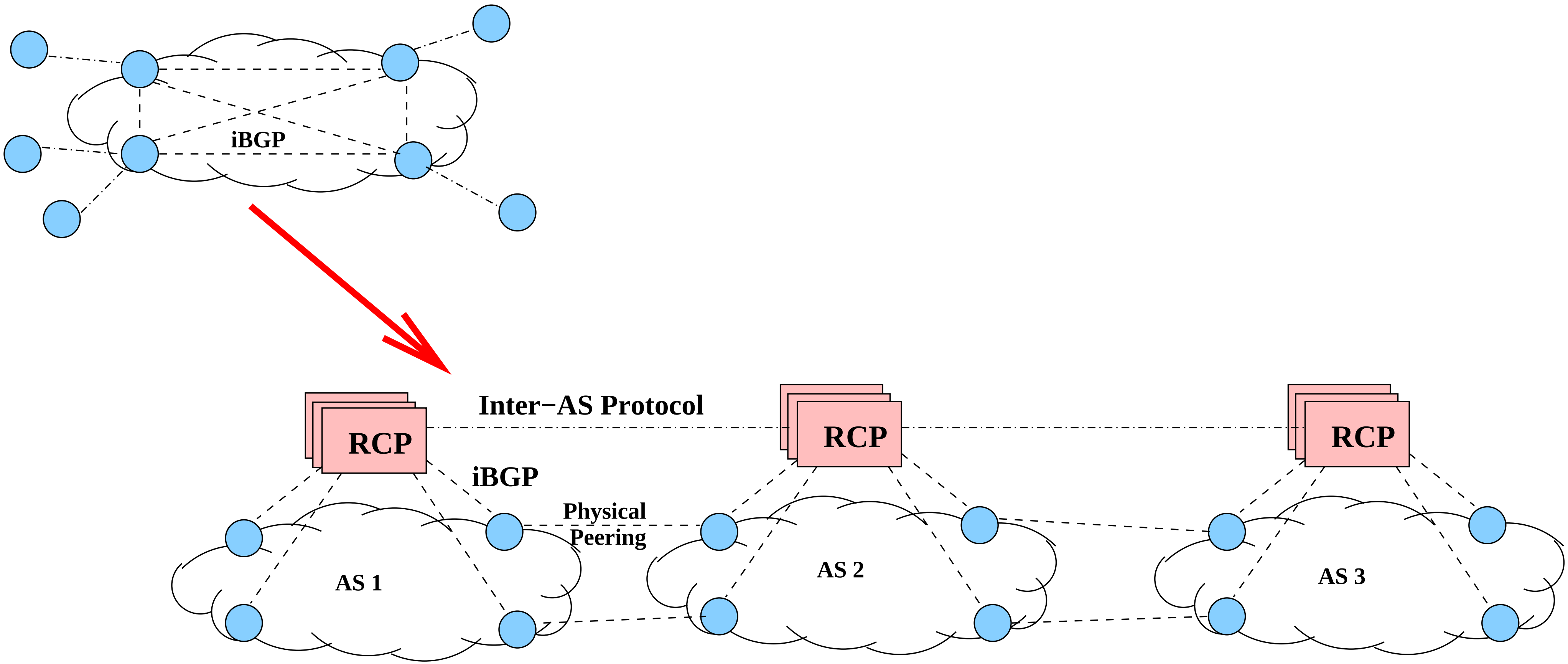
Routing Control Platform:

- Learns all externally learned routes
- Computes consistent router-level *paths*

Benefits

- Intrinsic loop freedom and convergence
- Path selection dictated by RCP
  - Need not abide by BGP-specific decision process
  - Can "pin" paths

Getting from here to there in three easy steps

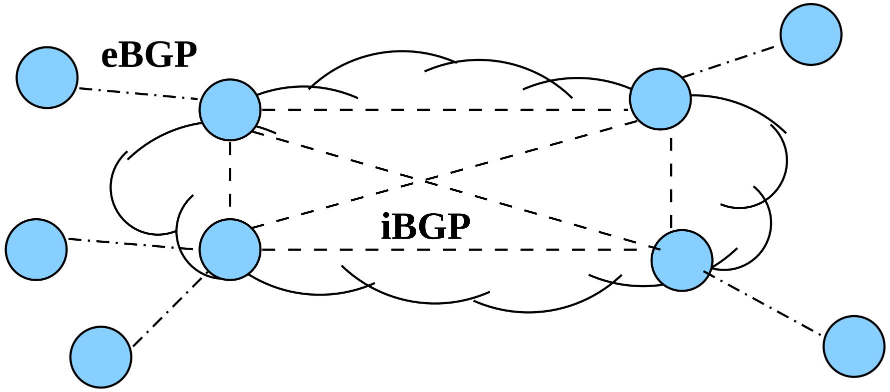


Two issues:

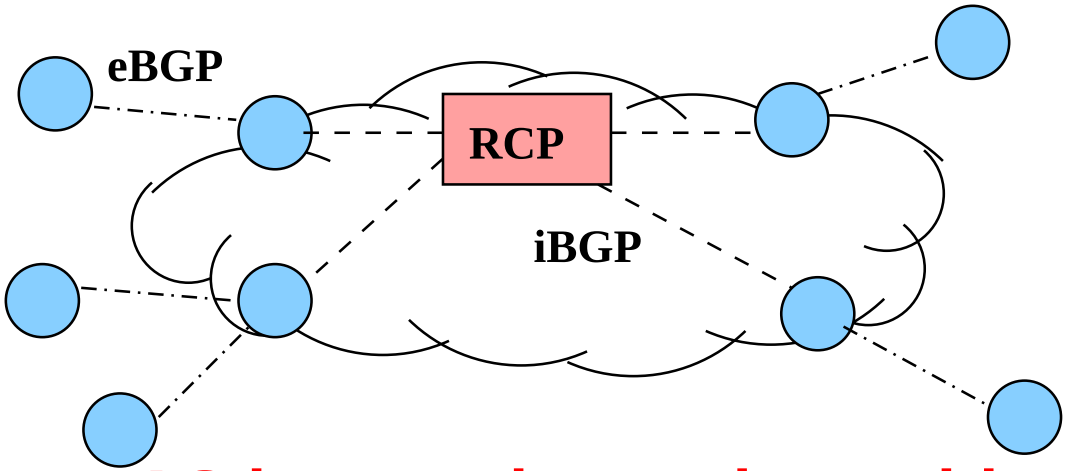
- Backwards compatibility
- Deployment incentives

Phase 1: Control Over Protocol Interactions

Before: Conventional iBGP



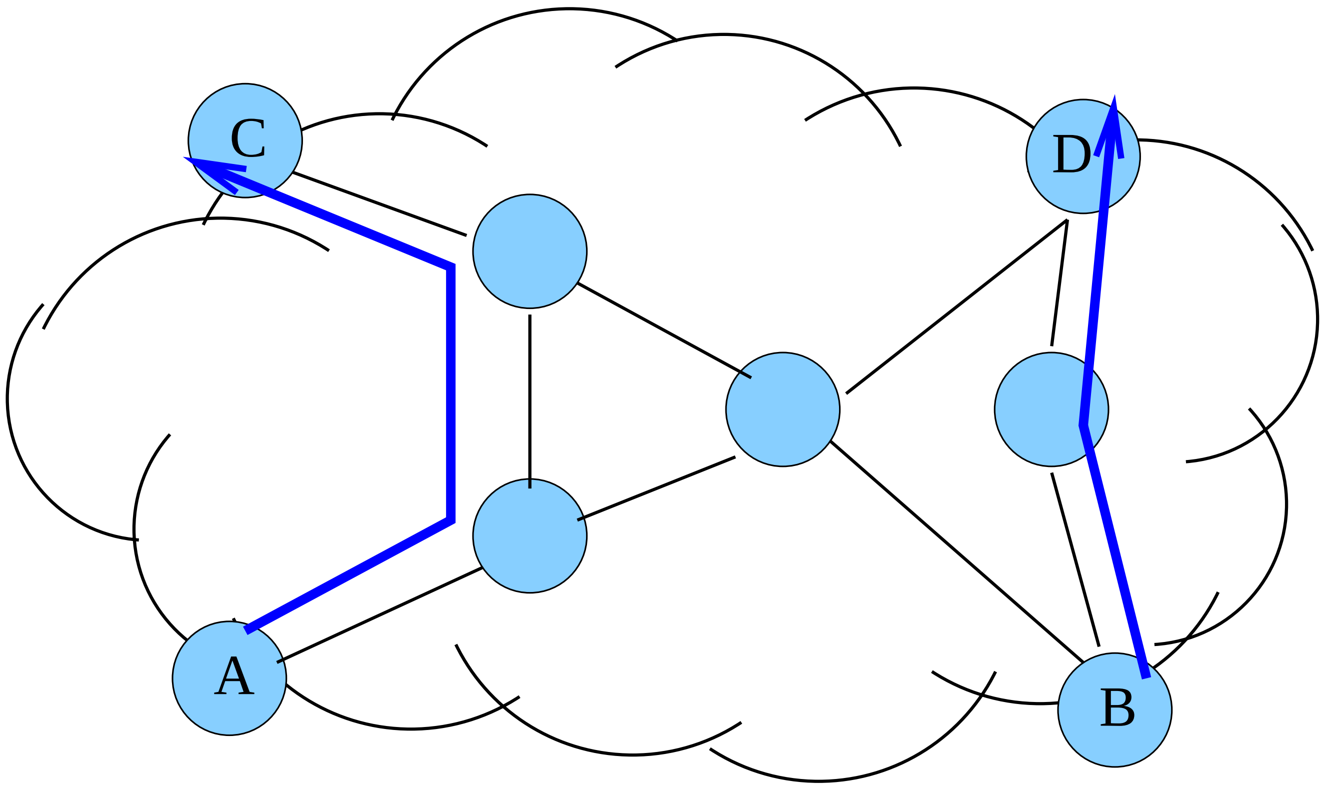
After: RCP gets "best" iBGP routes (and IGP topology)



Only one AS has to change its architecture!

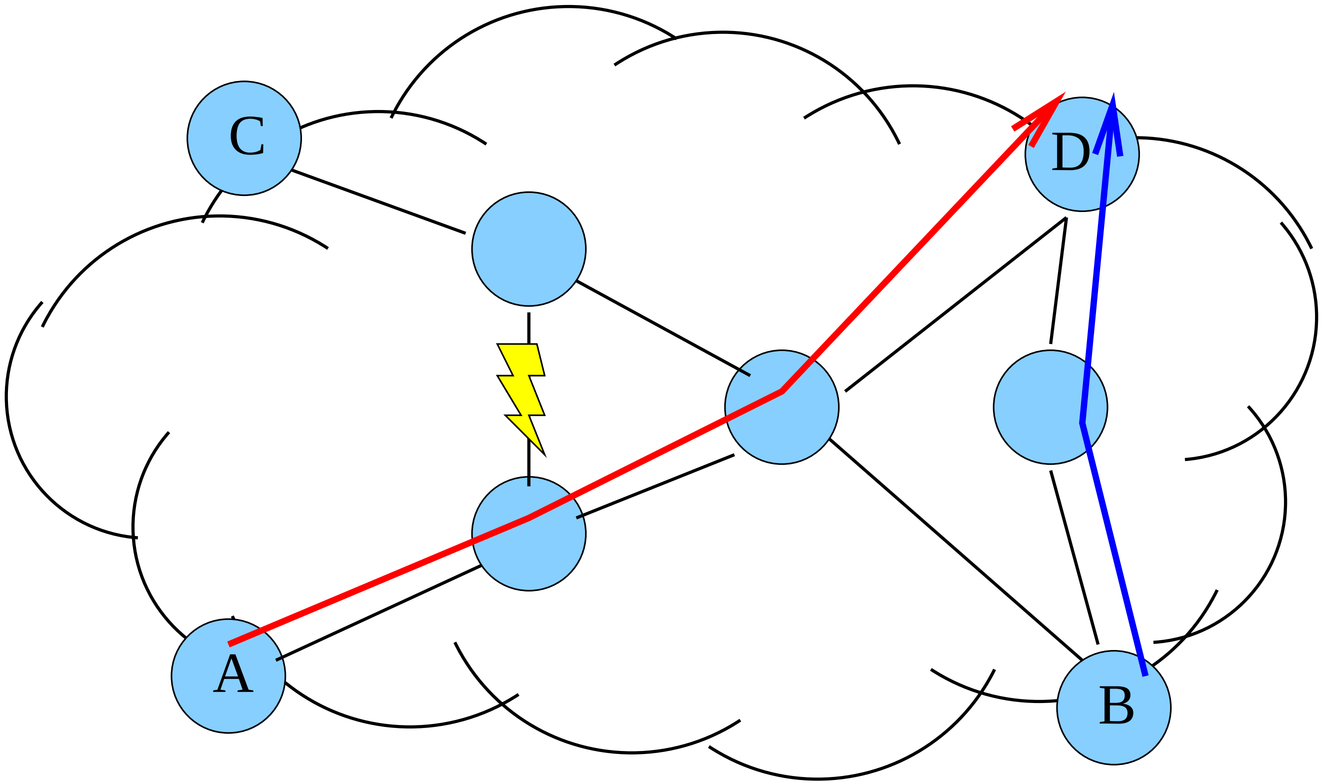
Application: Controlling Path Changes

BGP routes take "nearest exit" (shortest IGP path).



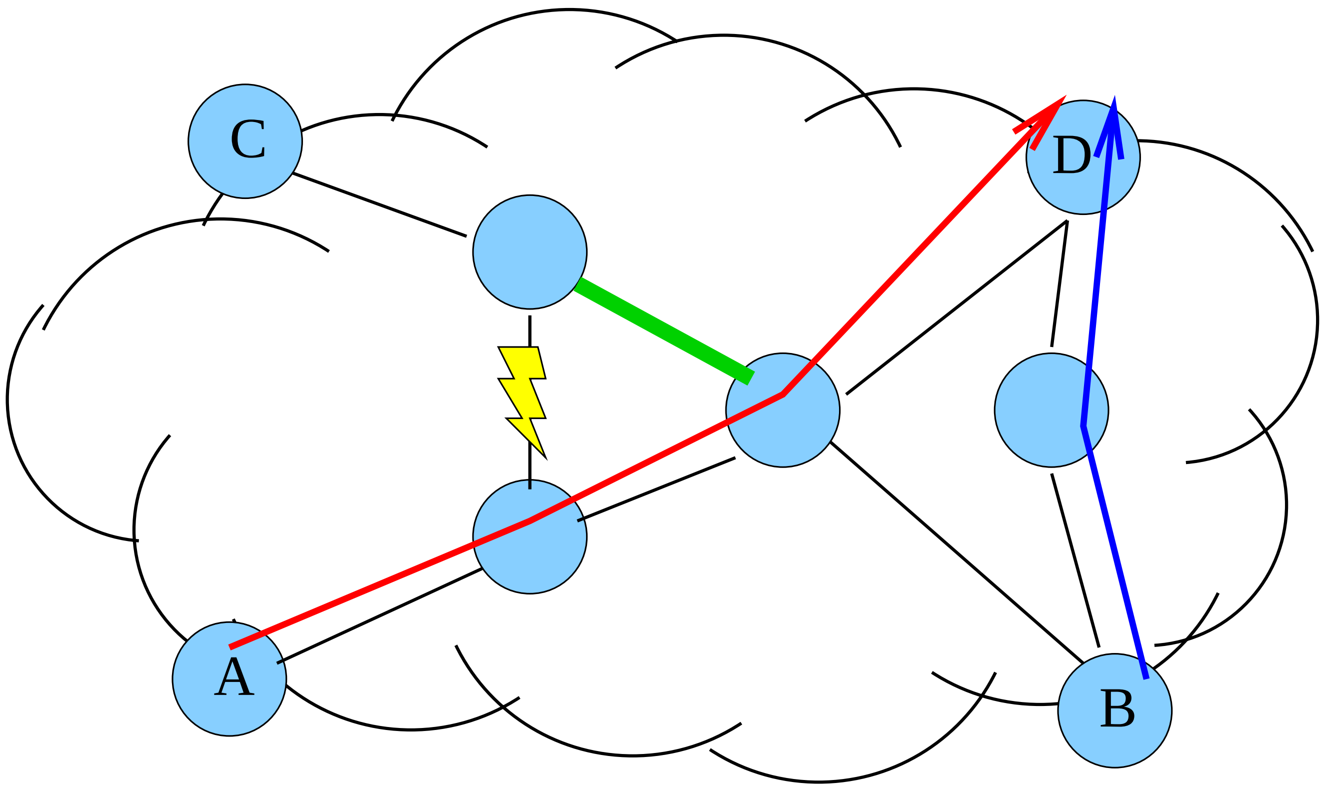
Application: Controlling Path Changes

BGP routes take "nearest exit" (shortest IGP path).
Failures or maintenance change internal weights.
Exit point can also change.
Traffic shifts, convergence delay, congestion in downstream AS.



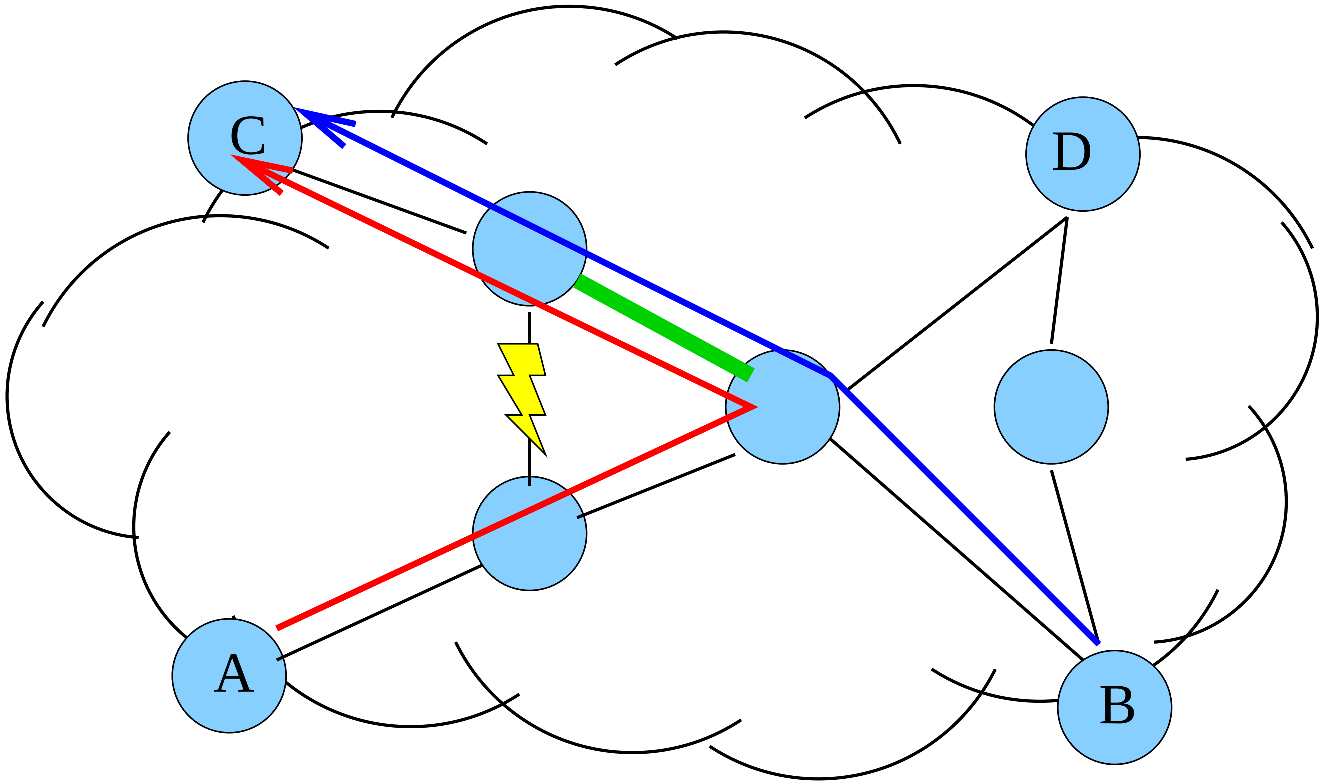
Application: Controlling Path Changes

BGP routes take "nearest exit" (shortest IGP path).
Failures or maintenance change internal weights.
Exit point can also change.
Traffic shifts, convergence delay, congestion in downstream AS.



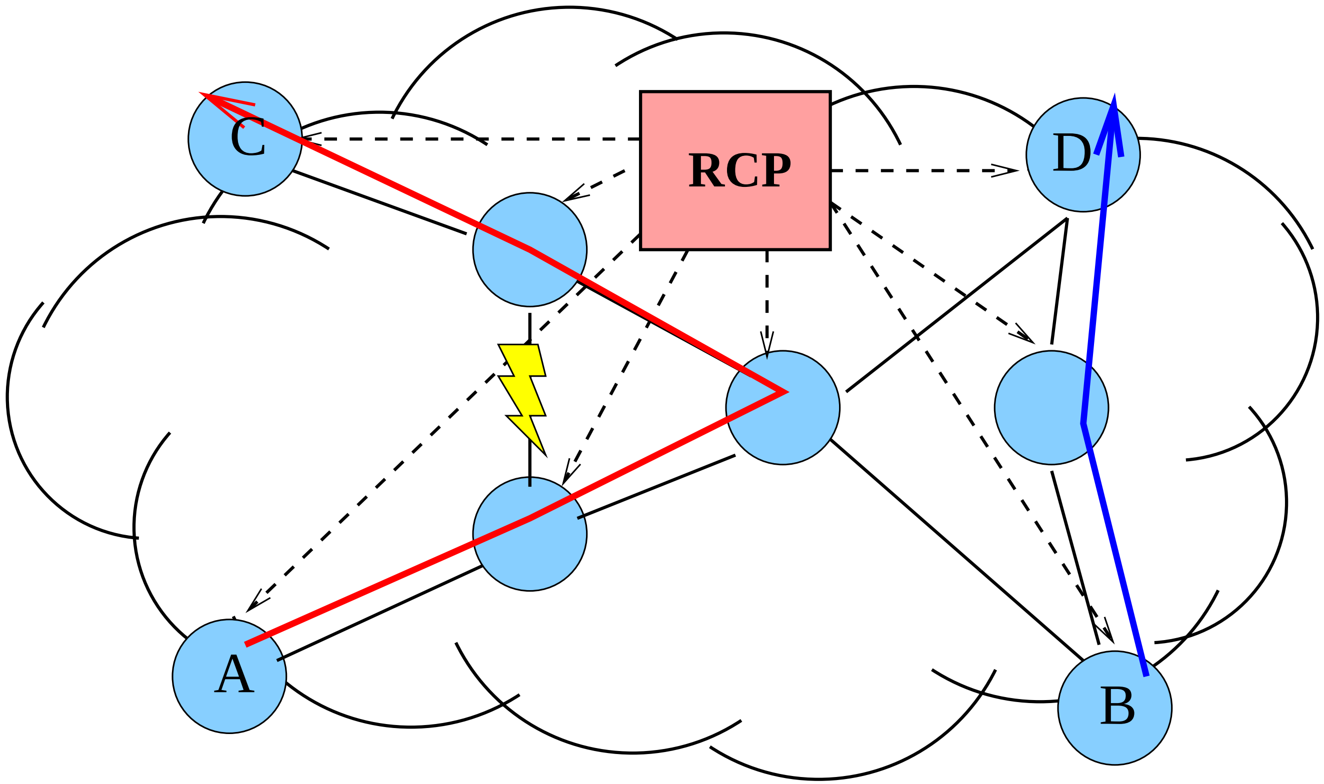
Application: Controlling Path Changes

BGP routes take "nearest exit" (shortest IGP path).
Failures or maintenance change internal weights.
Exit point can also change.
Traffic shifts, convergence delay, congestion in downstream AS.



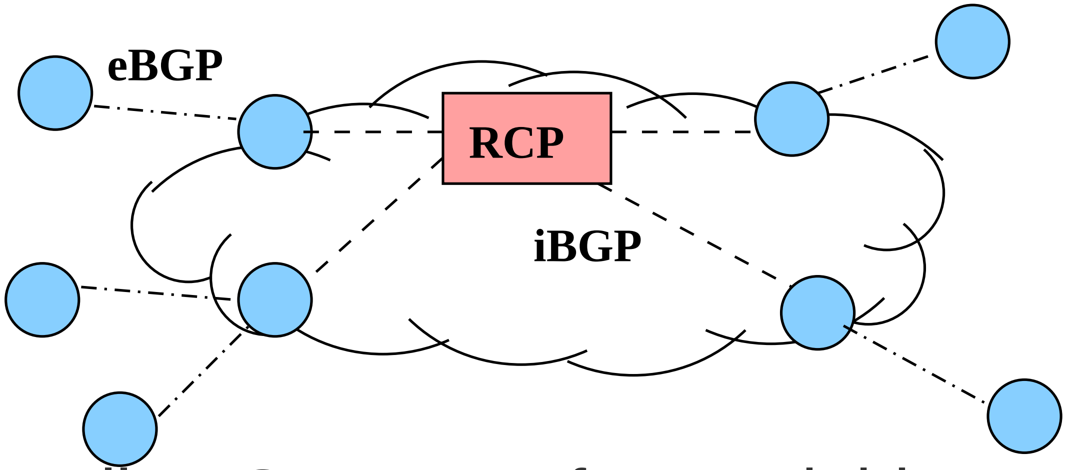
Application: Controlling Path Changes

BGP routes take "nearest exit" (shortest IGP path).
Failures or maintenance change internal weights.
RCP can "pin" exit points as IGP weights change.

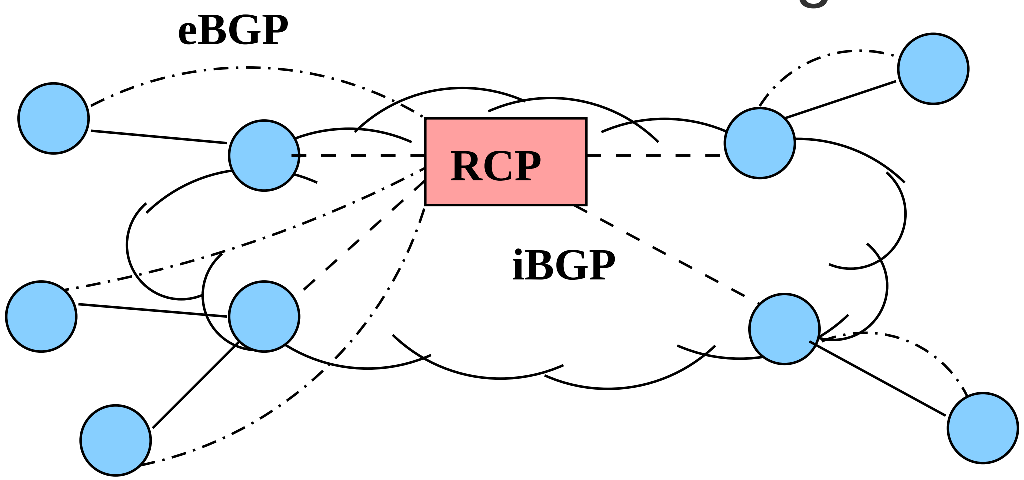


Phase 2: AS-Wide Selection and Policy

Before: RCP gets "best" iBGP routes (and IGP topology)

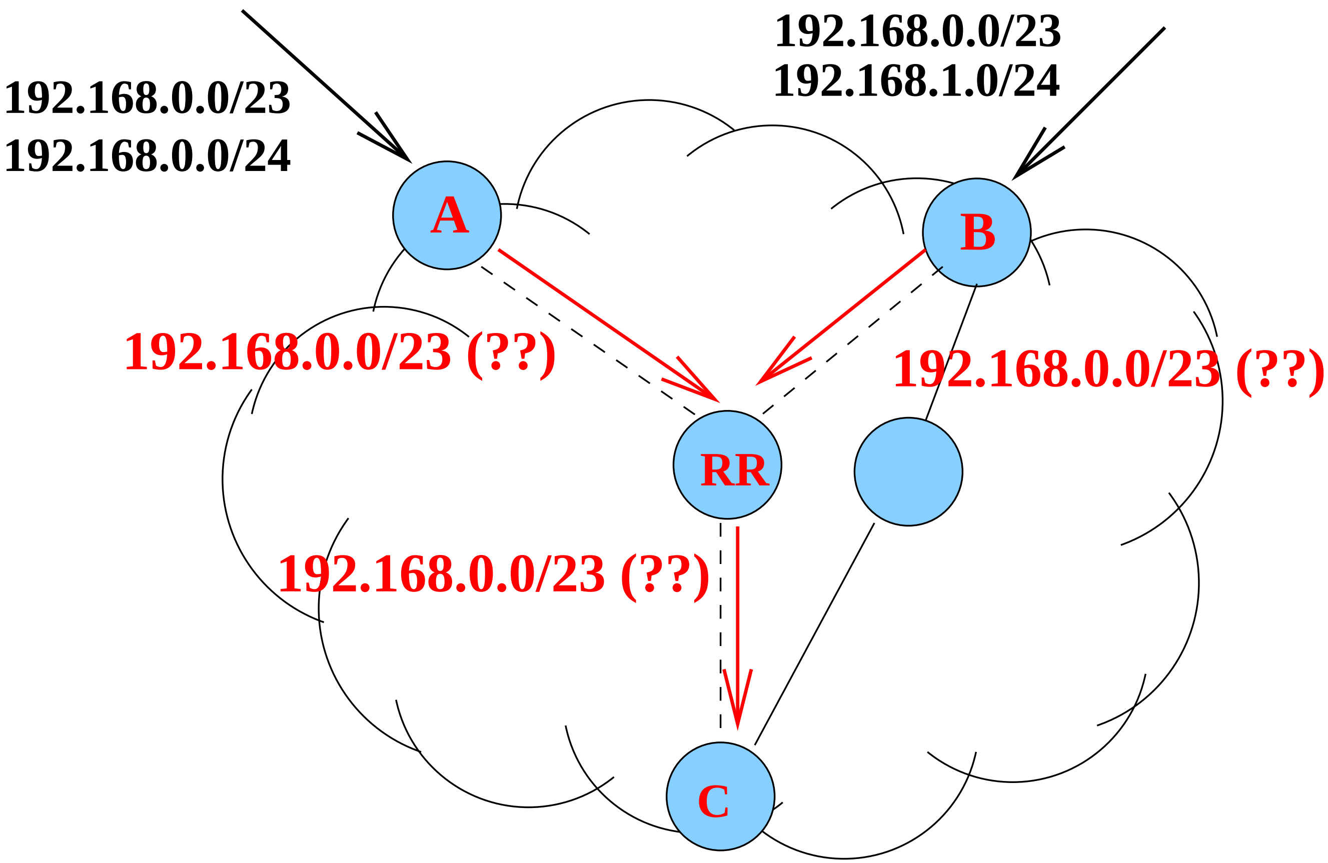


After: RCP gets all eBGP routes from neighbors



Phase 2 Application: Efficient Aggregation

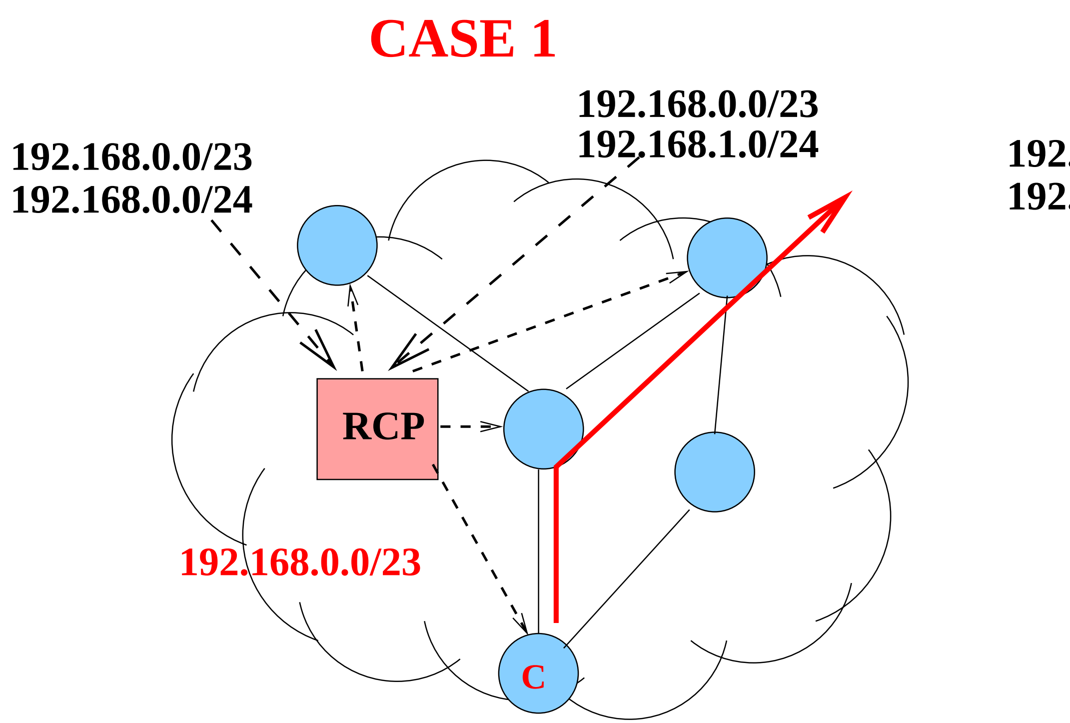
Aggregation curbs routing table growth.
Routers can't know which routers need more specific routes.



Phase 2 Application: Efficient Aggregation

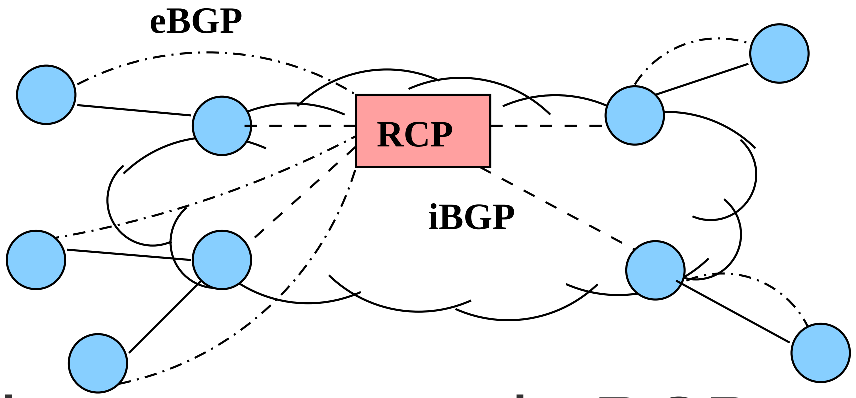
Aggregation curbs routing table growth.

Policy at RCP determines whether routers need separate routes.
RCP can always pass two subnets to downstream ASes.

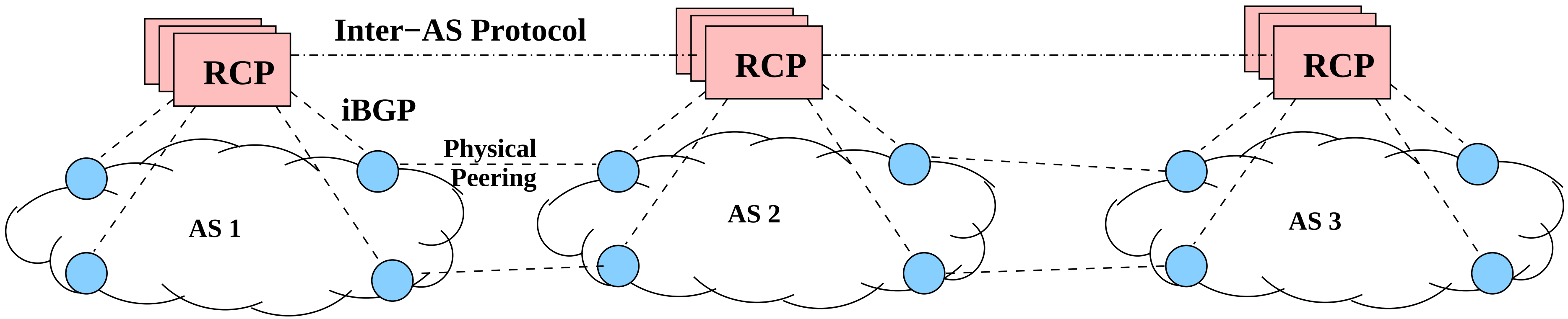


Phase 3: Routing Has Left the Routers

Before: RCP gets all eBGP routes from neighbors



After: ASes exchange routes via RCP



Phase 3 Application: More flexible routing

Better management:

- Diagnostics and troubleshooting
- Routing co-located with traffic information, etc.
- Ability to reason about the AS as a single entity

Protocol improvements:

- Attaching prices to routes
- Inter-AS negotiation of exit points
- Overlay routing informed by IP-layer information

- Your application here

  (Without worrying about BGP-specific arcana)

Scalability and Robustness

- Will it scale? Will it be fast enough?

  Maybe. We believe we can build the RCP on a single box. We're building a prototype.

  The RCP is doing less work than N routers

  - Cisco PRP-2 is 1.3 GHz, 1GB RAM

  (Note: centralized != inability to scale)

- Is that a single point of failure I see?

  No. Safe to replicate.

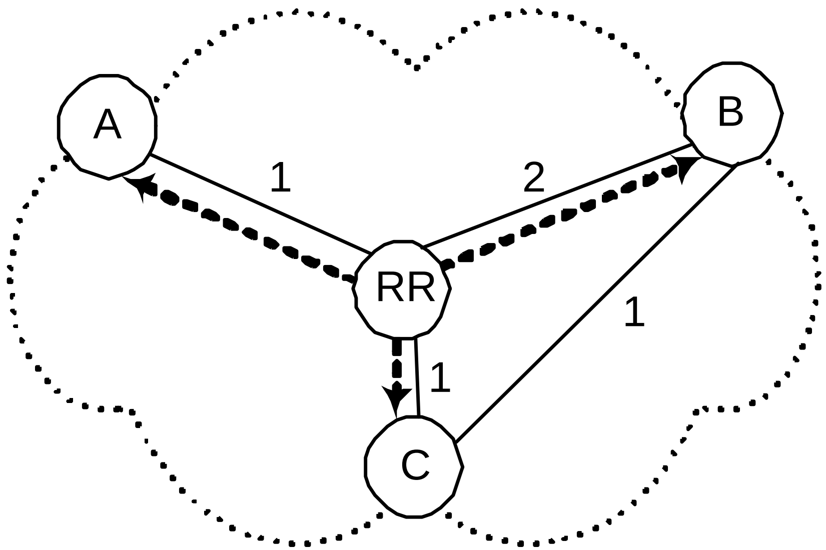
  - RCP can be replicated using distributed systems insights.
  - Consistency (mostly) a non-issue: OSPF guarantees clean partitions
  - Today's BGP was not designed with robustness in mind.

    (e.g., must replicate route reflectors PoP-by-PoP)

"RCP is basically a route reflector."

Yes, but it’s better.

- "Customized" routing decisions for clients.
  - Router reflectors do not compute routes from client’s perspective.
  - Route reflectors do not emulate a "full mesh". RCP can, though.



- Routing decisions based on complete visibility.
  - Guaranteed correct routes.
  - Replication can be dictated by systems issues.

"RCP also looks a lot like..."

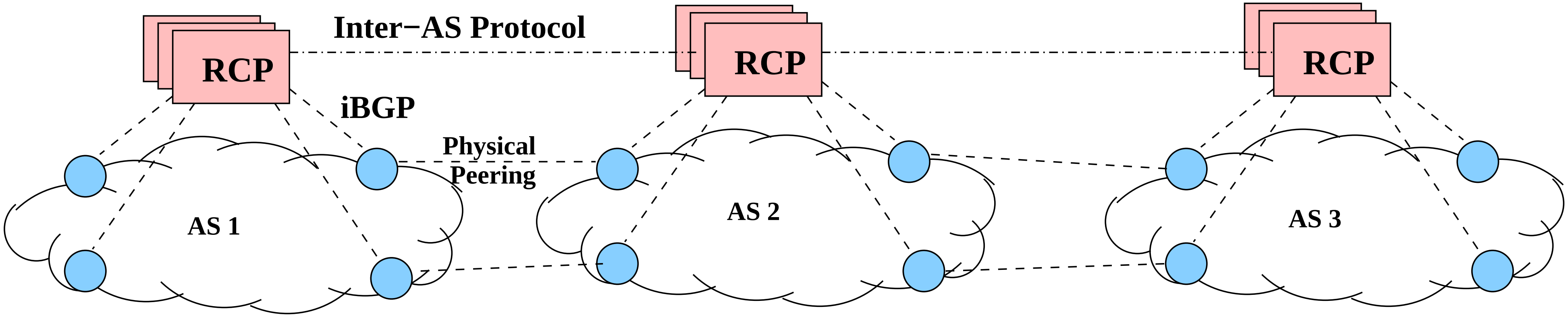
- A "route server"
  - Route arbiter: looked at applying policy at exchange points
  - AS agents: RCP answers questions like "What should these policy agents be doing?"

- An overlay network
  - Most previous work is in data overlays.
  - RCP is a **control overlay***(no data packets)*.
  - RCP could give data overlays more information and control.
    - RCP has more fine-grained information directly from the network (e.g., topology, traffic).
    - Can also make changes to the IP layer.

Conclusion: "Routing Control Platform"

Principles for interdomain routing:

- Compute consistent routes using complete state.
- Control routing protocol interactions.



Benefits:

- Simpler, more expressive configuration
- Intrinsic robustness: no loops, convergence, etc.
- More stable routing
- Enables new applications