Improving Internet-wide routing convergence with MRPC timers: bringing order to routing dynamics

Steve Uhlig
TU Berlin/Deutsche Telekom Labs
steve@net.t-labs.tu-berlin.de

Anthony Lambert
Orange Labs

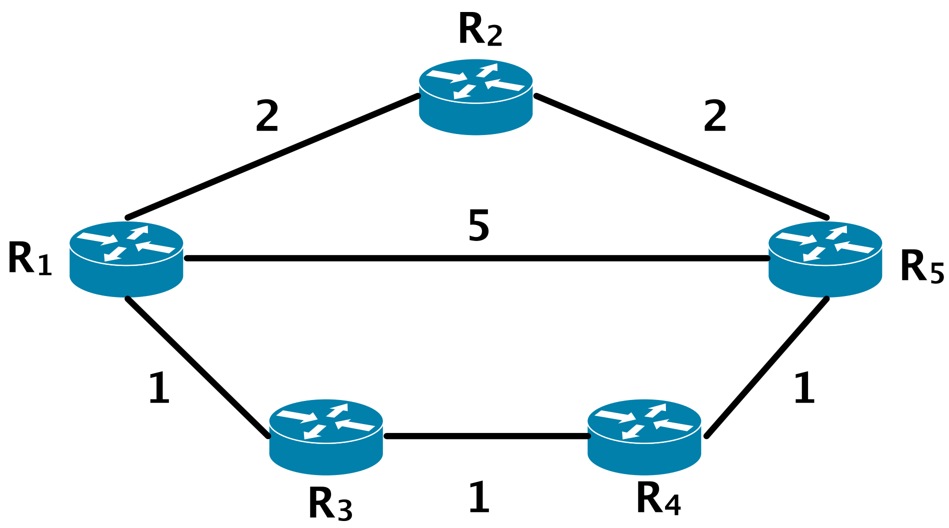
Marc-Olivier Buob
Orange Labs

Outline

- Motivation
- Routing algebras
- MRPC timers
- Evaluation
- Arbitrary dynamics
- Conclusions

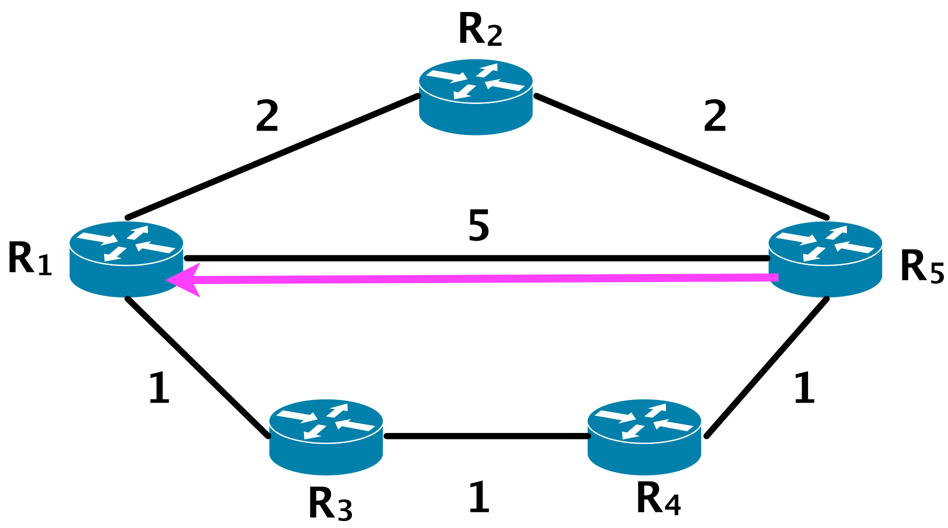
Motivation

Path exploration



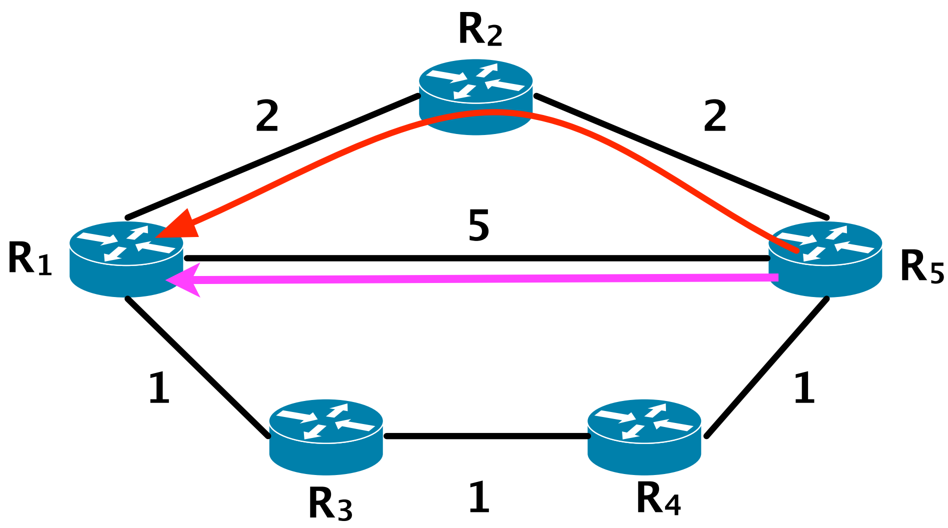
- Path exploration = paths selected as best before final best
- Path exploration is common in the Internet [Bürkle'03, Oliveira'06]
- Leads to poor convergence [Labovitz'99,Mao'02]

Motivation
Path exploration



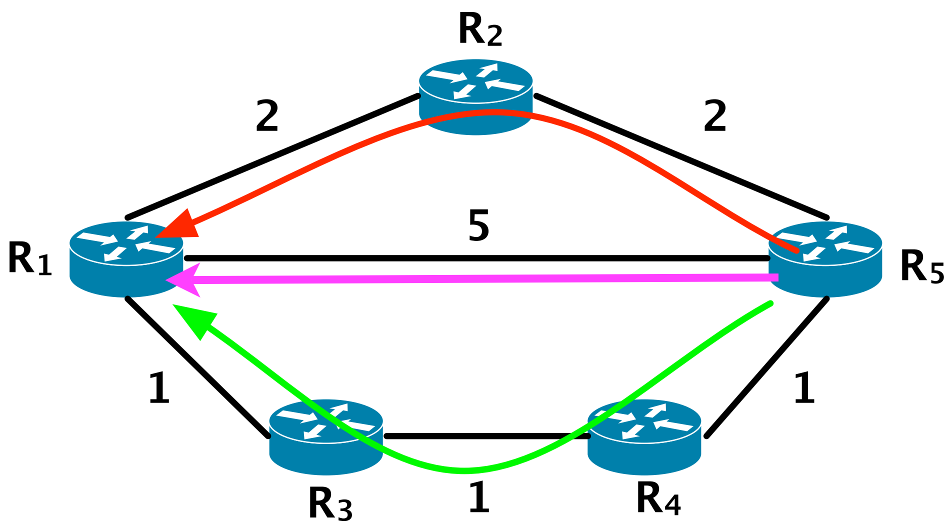
- Path exploration = paths selected as best before final best
- Path exploration is common in the Internet [Bürkle'03, Oliveira'06]
- Leads to poor convergence [Labovitz'99,Mao'02]

Motivation
Path exploration



- Path exploration = paths selected as best before final best
- Path exploration is common in the Internet [Bürkle'03, Oliveira'06]
- Leads to poor convergence [Labovitz'99,Mao'02]

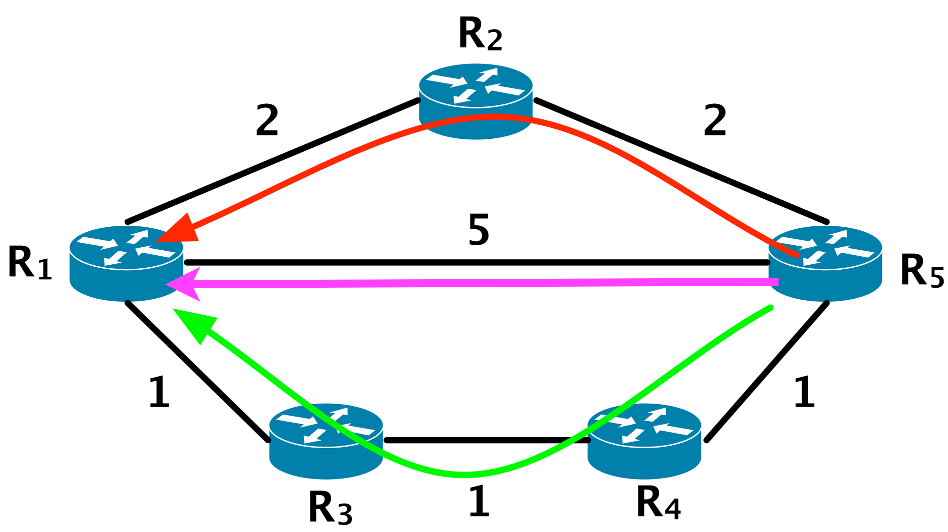
Motivation
Path exploration



- Path exploration = paths selected as best before final best
- Path exploration is common in the Internet [Bürkle'03, Oliveira'06]
- Leads to poor convergence [Labovitz'99,Mao'02]

Motivation

Path exploration

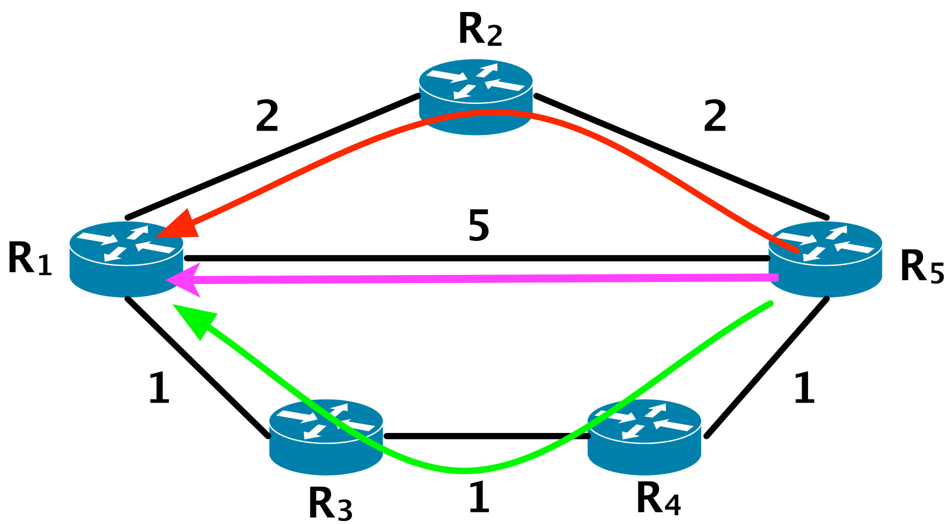


- Path exploration = paths selected as best before final best
- Path exploration is common in the Internet [Bürkle'03, Oliveira'06]
- Leads to poor convergence [Labovitz'99,Mao'02]

- Paths explored by R_5 towards R_1:

  1. R_5 – R_1
  2. R_5 – R_2 – R_1

Motivation
Path exploration

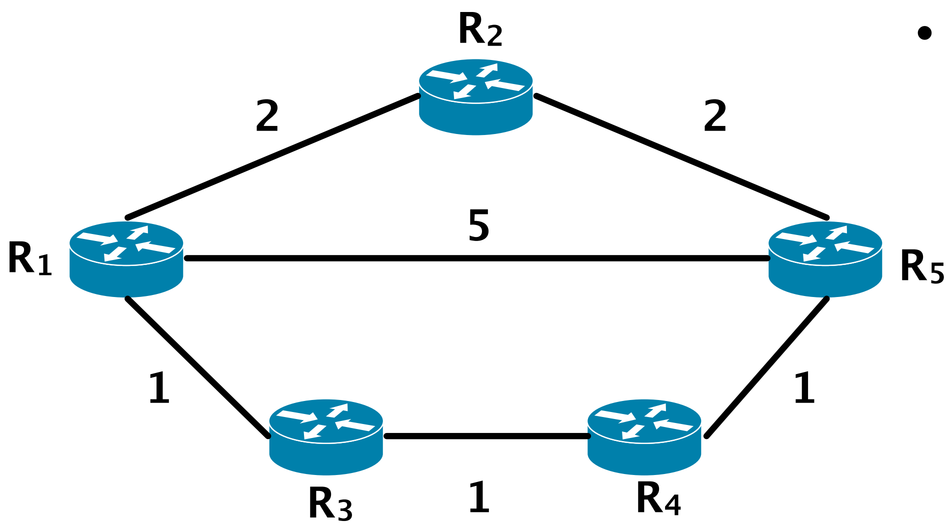


- Path exploration = paths selected as best before final best
- Path exploration is common in the Internet [Bürkle'03, Oliveira'06]
- Leads to poor convergence [Labovitz'99,Mao'02]

- Paths explored by R_5 towards R_1:
  1. $R_5 - R_1$
  2. $R_5 - R_2 - R_1$

Path exploration is the consequence of the real problem: lack of proper routing updates ordering!

Motivation
Today's solution

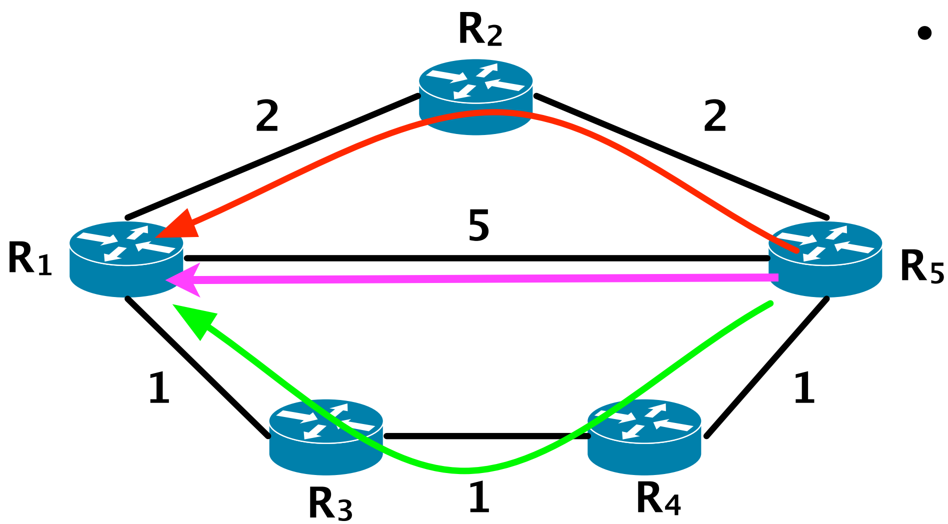


- Impact of uniform MRAI on propagation
  - ? $R_5 - R_1$
  - ? $R_5 - R_2 - R_1$
  - ? $R_5 - R_4 - R_3 - R_1$

- MRAI timers delay BGP updates announcements on BGP sessions
  - Implementation-dependent behavior
  - Typical values: [0,5] seconds on iBGP sessions, [0,30] seconds on eBGP sessions [RFC4271]
  - All messages are delayed indiscriminately
  - No value fits all situations [Griffin'01]

Motivation

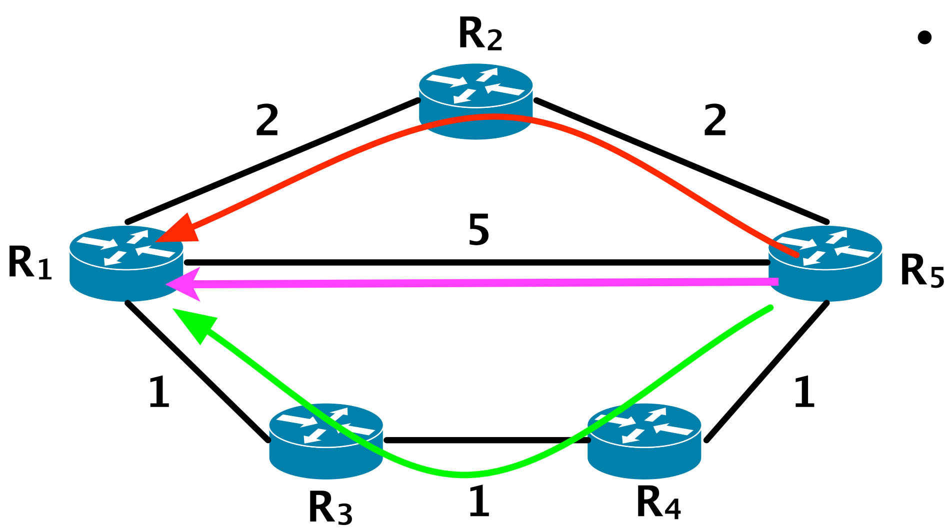
Today’s solution



- Impact of uniform MRAI on propagation
  - ? $R_5 - R_1$
  - ? $R_5 - R_2 - R_1$
  - ? $R_5 - R_4 - R_3 - R_1$

- MRAI timers delay BGP updates announcements on BGP sessions
  - Implementation-dependent behavior
  - Typical values: [0,5] seconds on iBGP sessions, [0,30] seconds on eBGP sessions [RFC4271]
  - All messages are delayed indiscriminately
  - No value fits all situations [Griffin’01]

Motivation
Today's solution



- Impact of uniform MRAI on propagation
  - ? $R_5 - R_1$
  - ? $R_5 - R_2 - R_1$
  - ? $R_5 - R_4 - R_3 - R_1$

- MRAI timers delay BGP updates announcements on BGP sessions
  - Implementation-dependent behavior
  - Typical values: [0,5] seconds on iBGP sessions, [0,30] seconds on eBGP sessions [RFC4271]
  - All messages are delayed indiscriminately
  - No value fits all situations [Griffin'01]

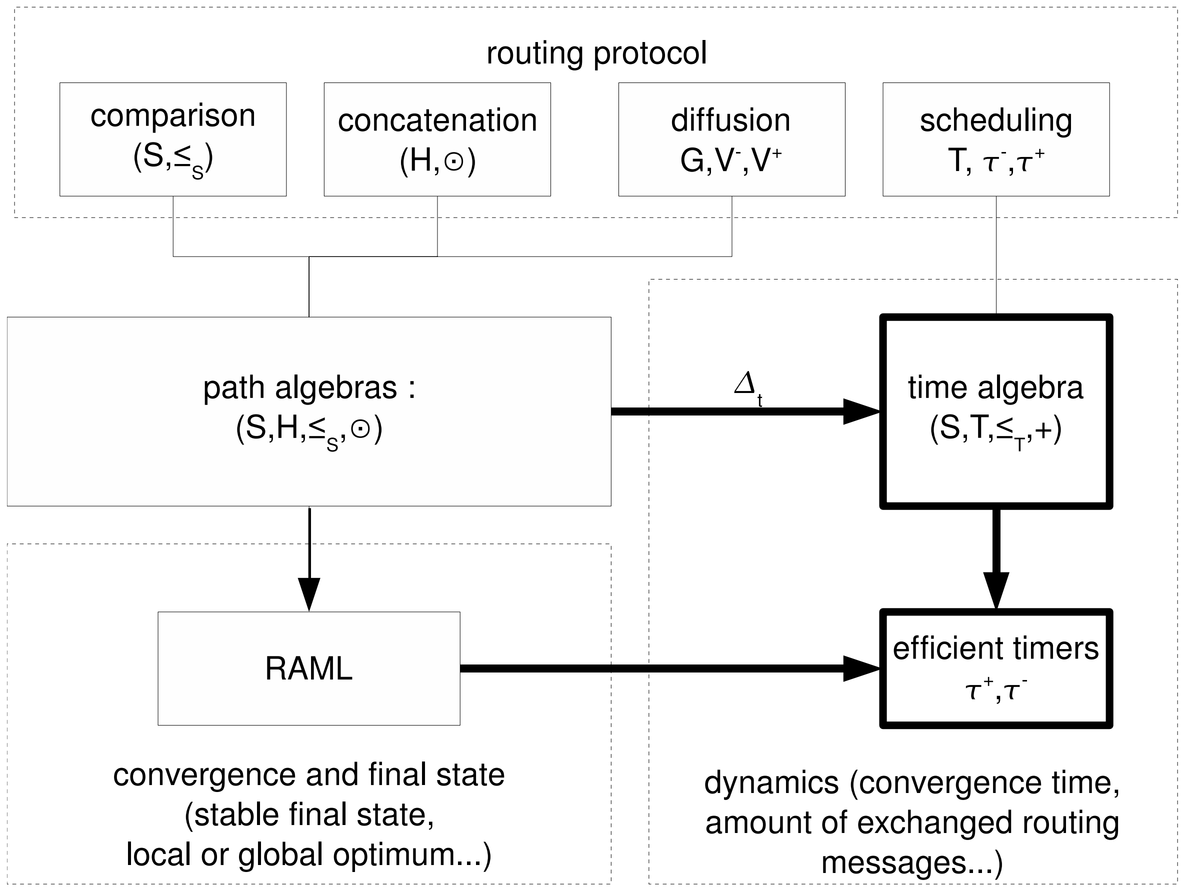
MRAI = delaying blindly

Outline

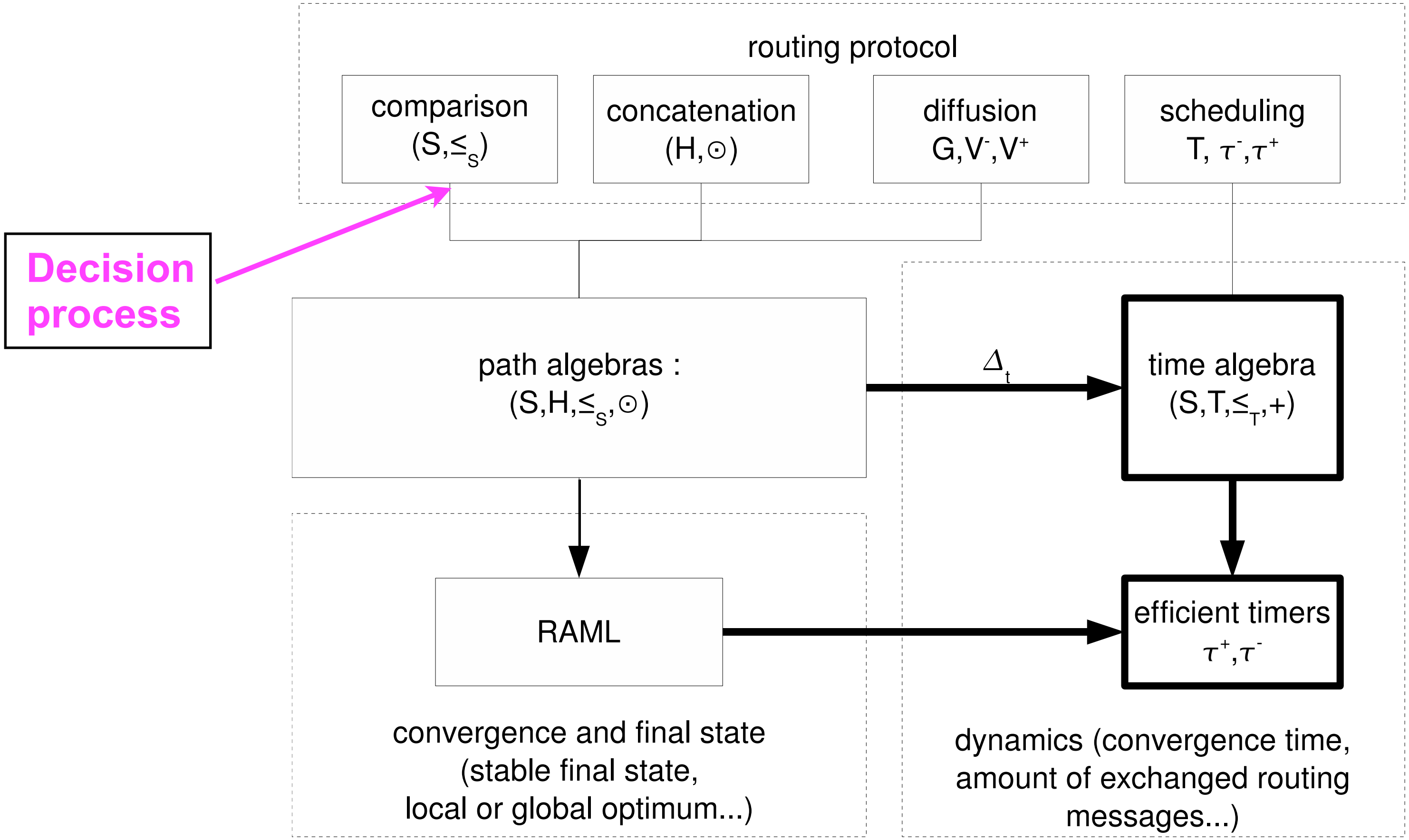
- Motivation
- Routing algebras
- MRPC timers
- Evaluation
- Arbitrary dynamics
- Conclusions

Routing algebras

What is a routing protocol?

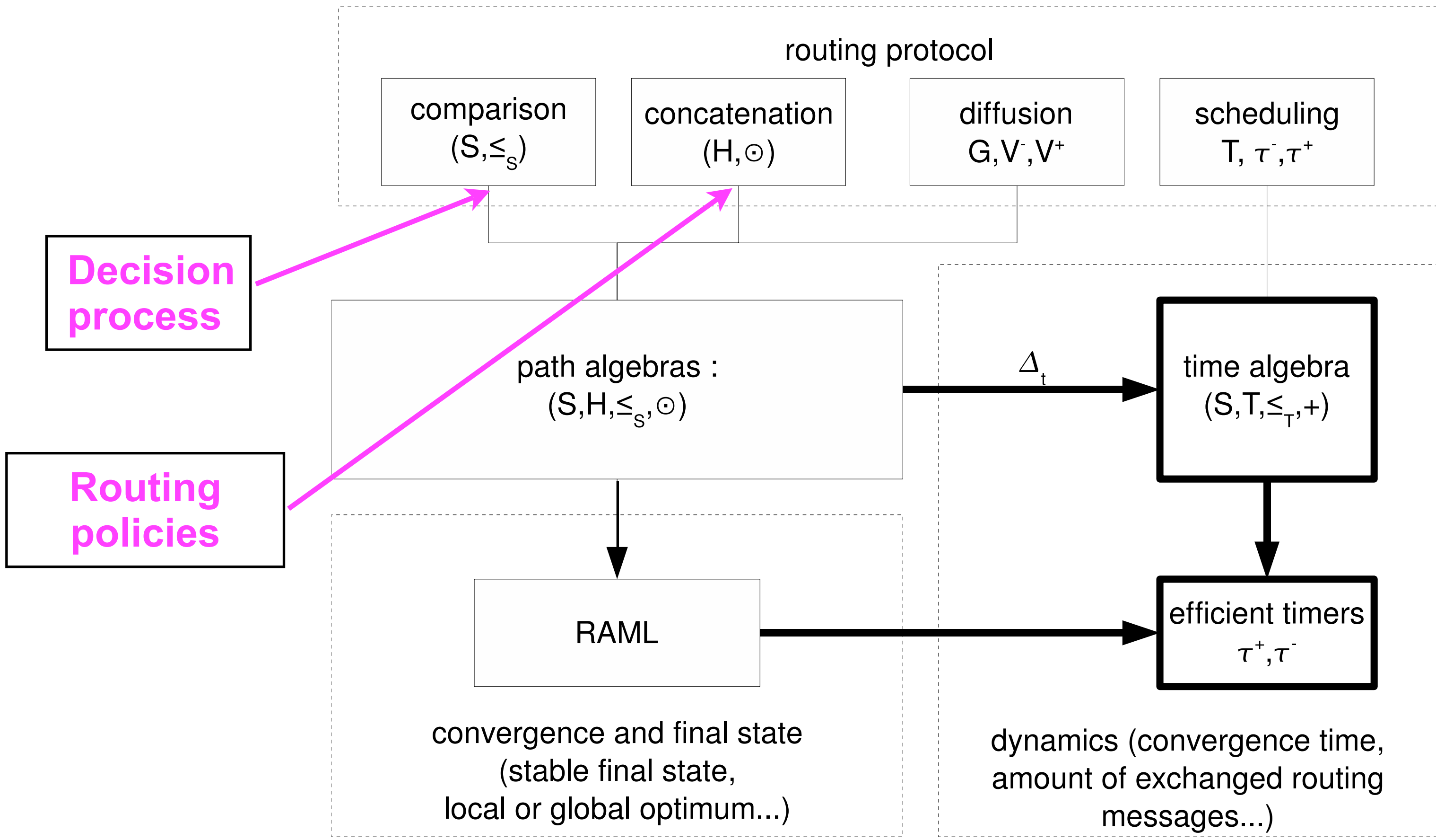


Routing algebras
What is a routing protocol?



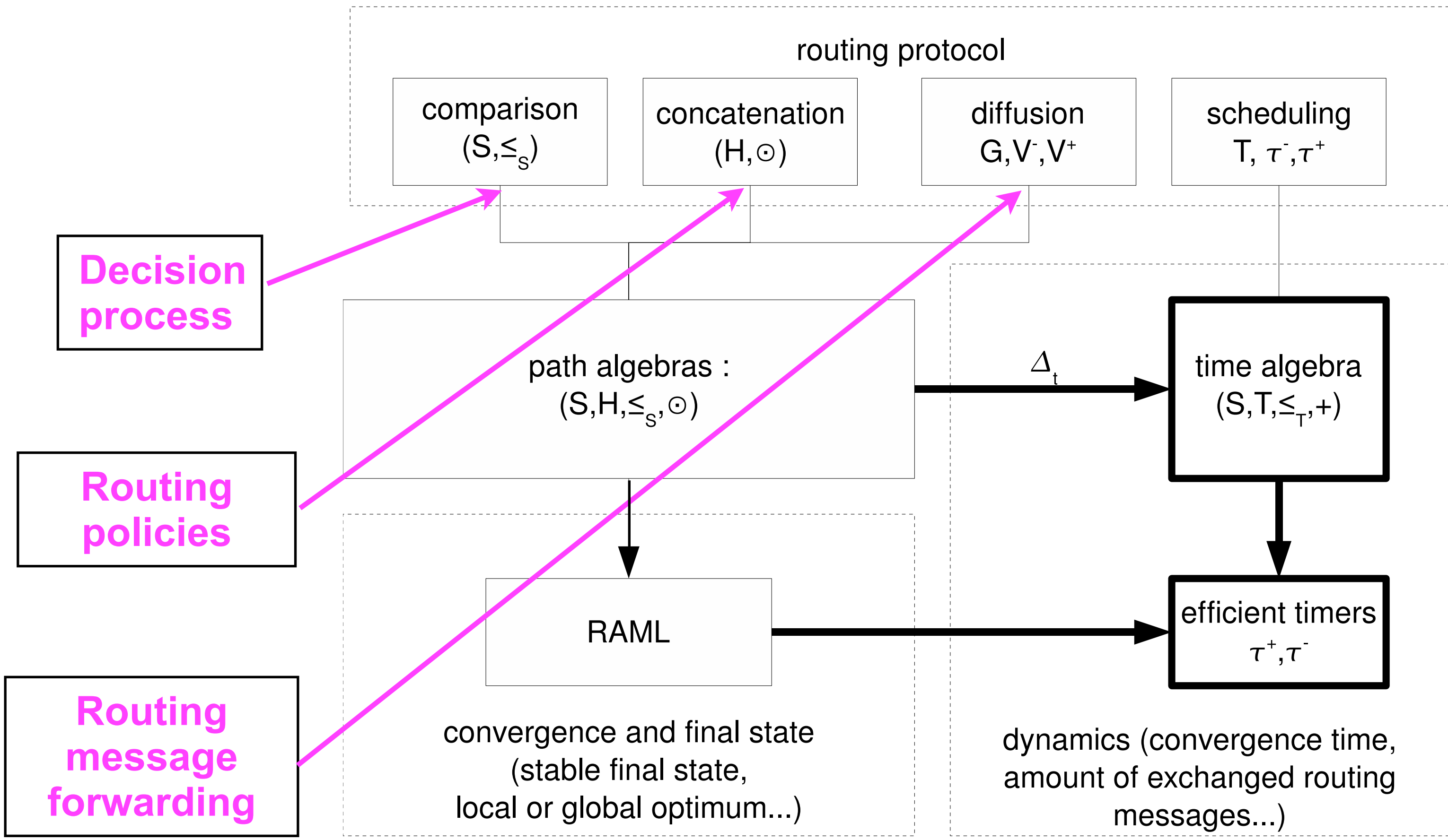
Routing algebras

What is a routing protocol?

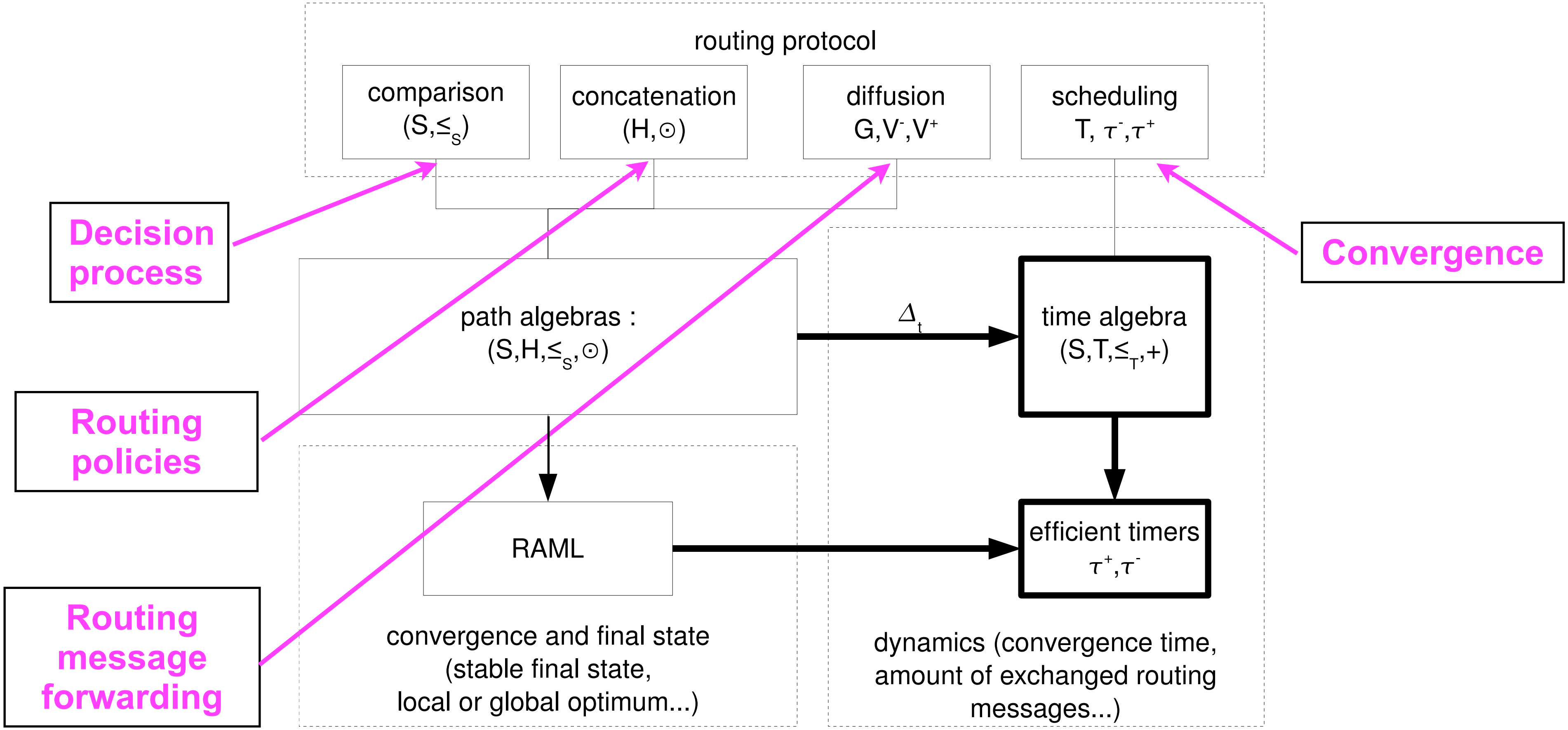


Routing algebras

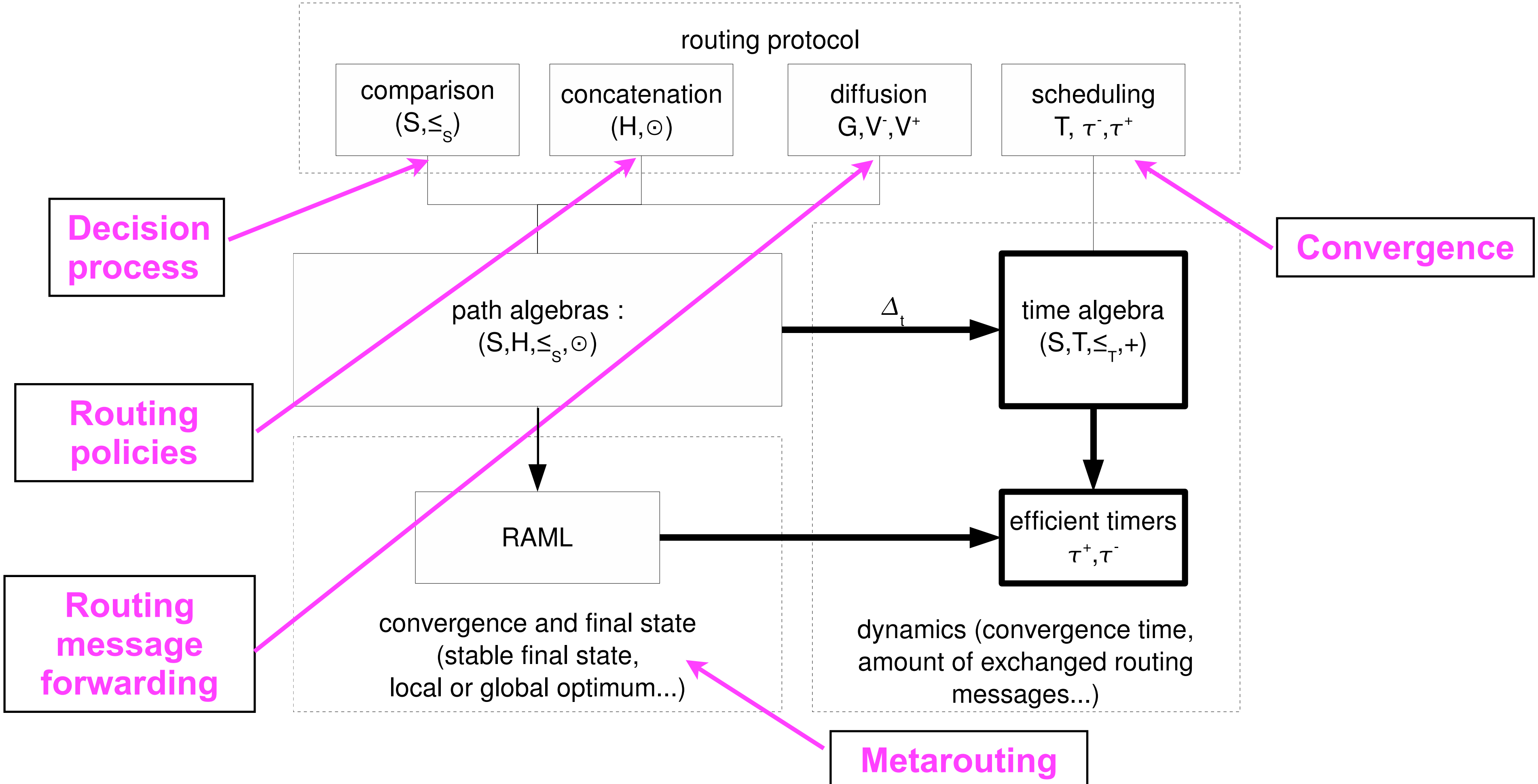
What is a routing protocol?



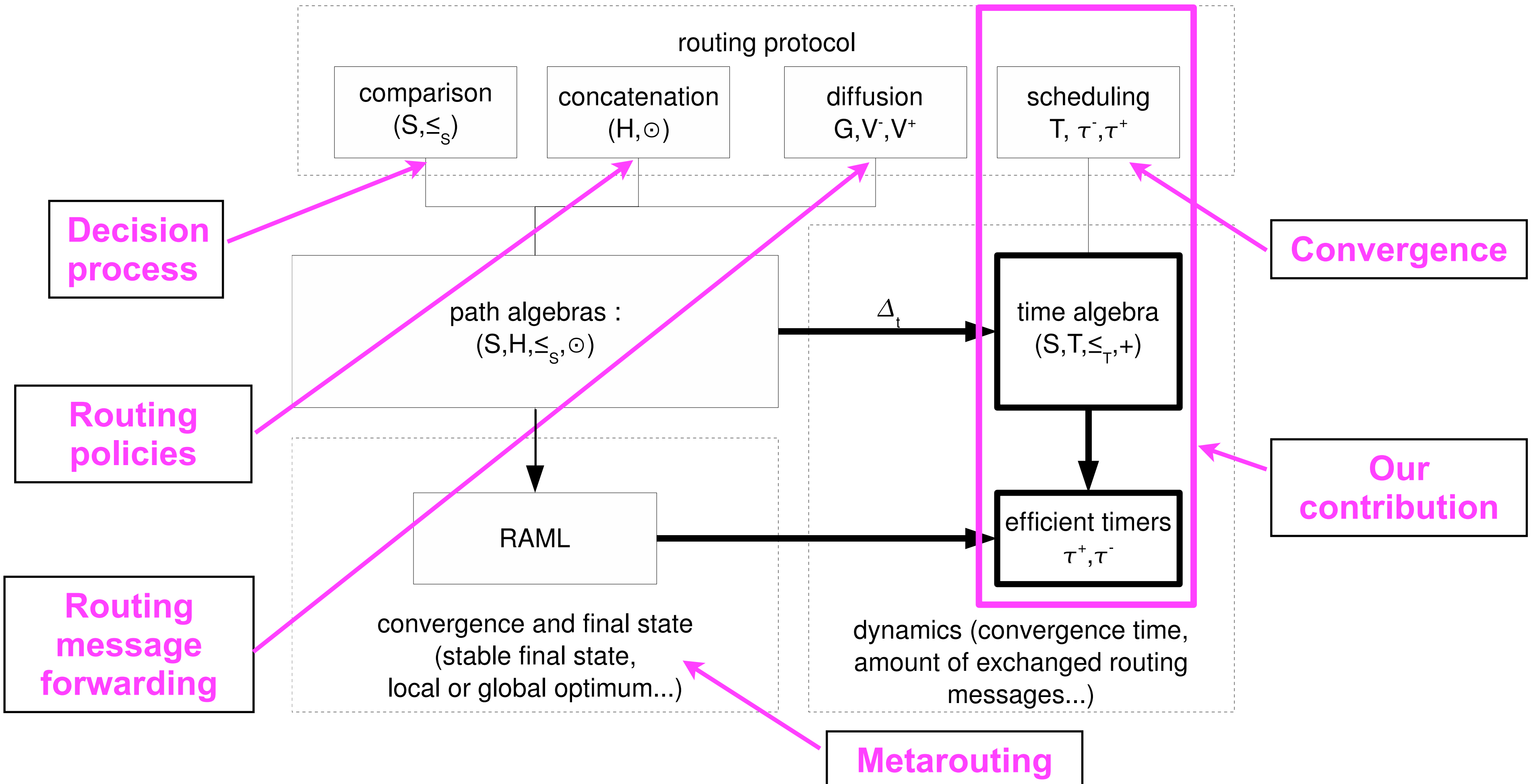
Routing algebras
What is a routing protocol?



Routing algebras
What is a routing protocol?



Routing algebras
What is a routing protocol?

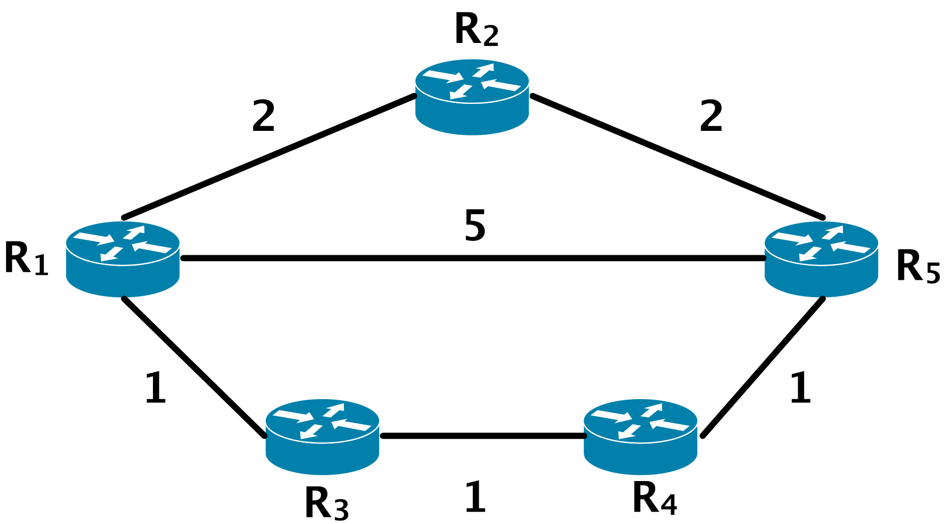


Outline

- Motivation
- Routing algebras
- MRPC timers
- Evaluation
- Arbitrary dynamics
- Conclusions

MRPC timers

Shortest path routing



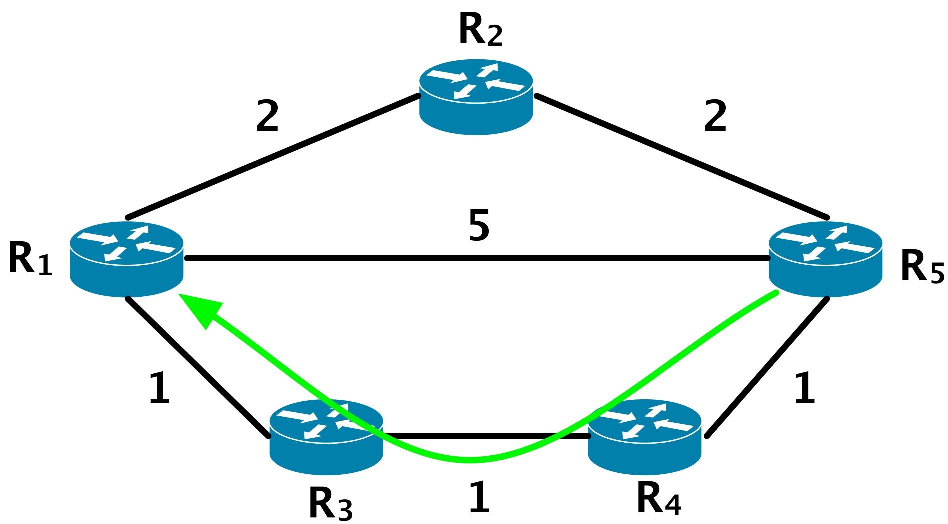
- How to ensure that each router will learn its preferred path first?
- Delay announcements to enforce a given updates propagation

For instance for R_5 toward R_1:

- R_5–R_4–R_3–R_1 is learned after 3k time units
- R_5–R_2–R_1 is learned after 4k time units
- R_5–R_1 is learned after 5k time units

MRPC timers

Shortest path routing



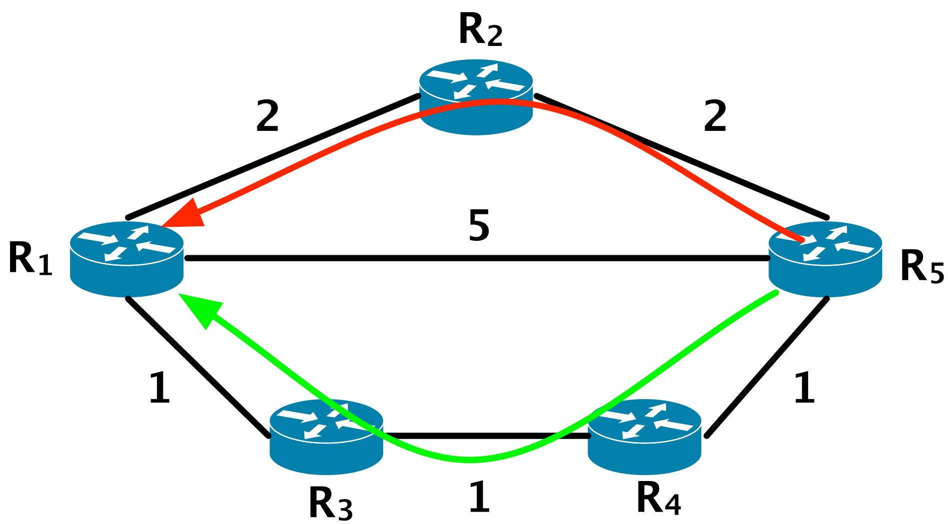
- How to ensure that each router will learn its preferred path first?
- Delay announcements to enforce a given updates propagation

For instance for R_5 toward R_1:

- R_5–R_4–R_3–R_1 is learned after 3k time units
- R_5–R_2–R_1 is learned after 4k time units
- R_5–R_1 is learned after 5k time units

MRPC timers

Shortest path routing

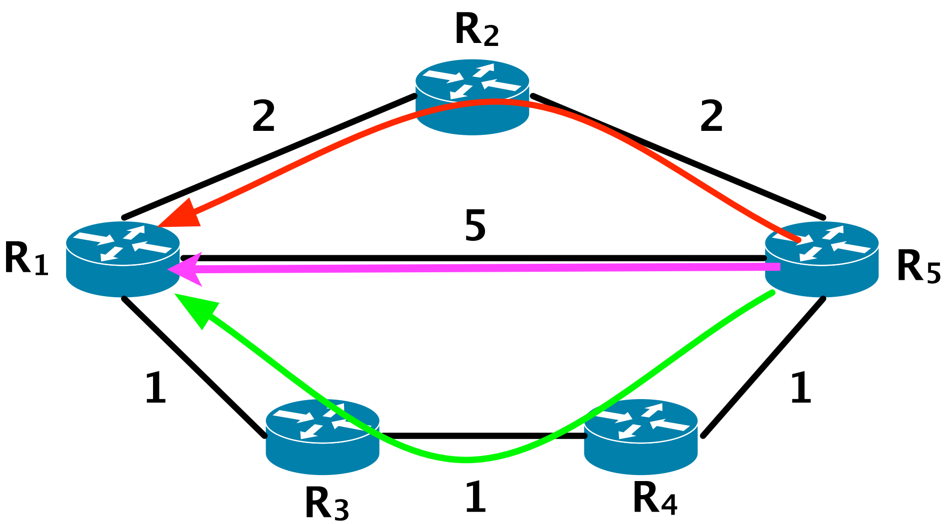


- How to ensure that each router will learn its preferred path first?
- Delay announcements to enforce a given updates propagation

For instance for R_5 toward R_1:
- R_5–R_4–R_3–R_1 is learned after 3k time units
- R_5–R_2–R_1 is learned after 4k time units
- R_5–R_1 is learned after 5k time units

MRPC timers

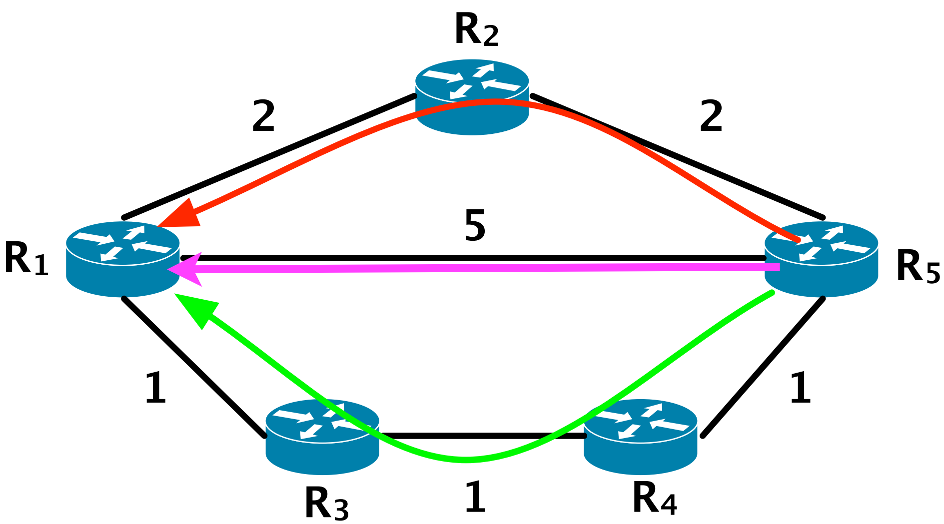
Shortest path routing



- How to ensure that each router will learn its preferred path first?
- Delay announcements to enforce a given updates propagation

For instance for R_5 toward R_1:
- R_5–R_4–R_3–R_1 is learned after 3k time units
- R_5–R_2–R_1 is learned after 4k time units
- R_5–R_1 is learned after 5k time units

MRPC timers
Shortest path routing



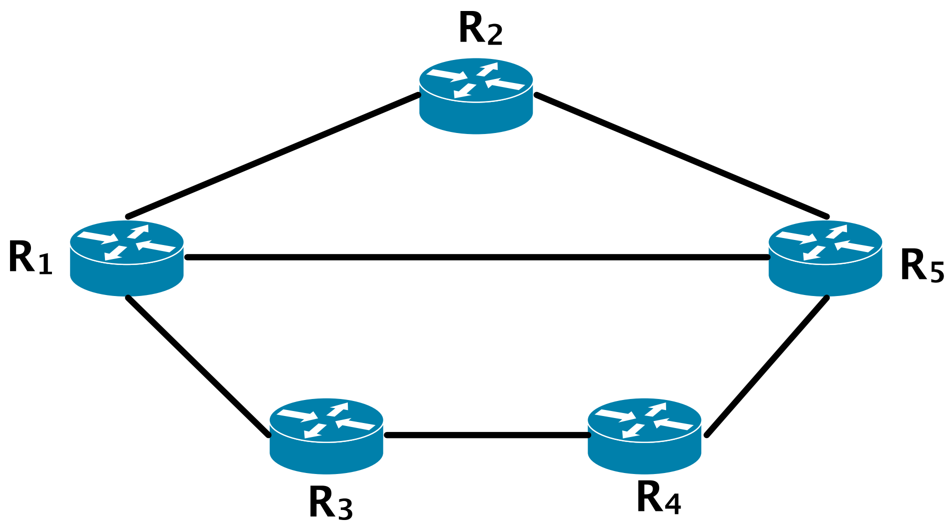
- How to ensure that each router will learn its preferred path first?
- Delay announcements to enforce a given updates propagation

For instance for R_5 toward R_1:
- R_5–R_4–R_3–R_1 is learned after 3k time units
- R_5–R_2–R_1 is learned after 4k time units
- R_5–R_1 is learned after 5k time units

Guiding principle: A path should be learned earlier if it is preferred to another

MRPC timers

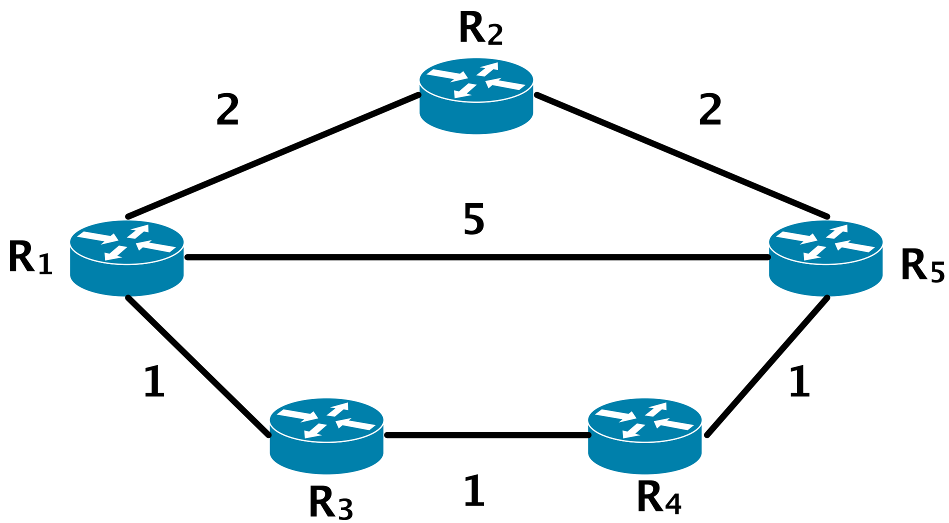
From metrics to functions



- Metrics are actually functions applied by routers on paths received
- Functions are defined on arcs and can be asymmetric
- Functions consider Metrics and Routing Policies ⇒ MRPC

MRPC timers

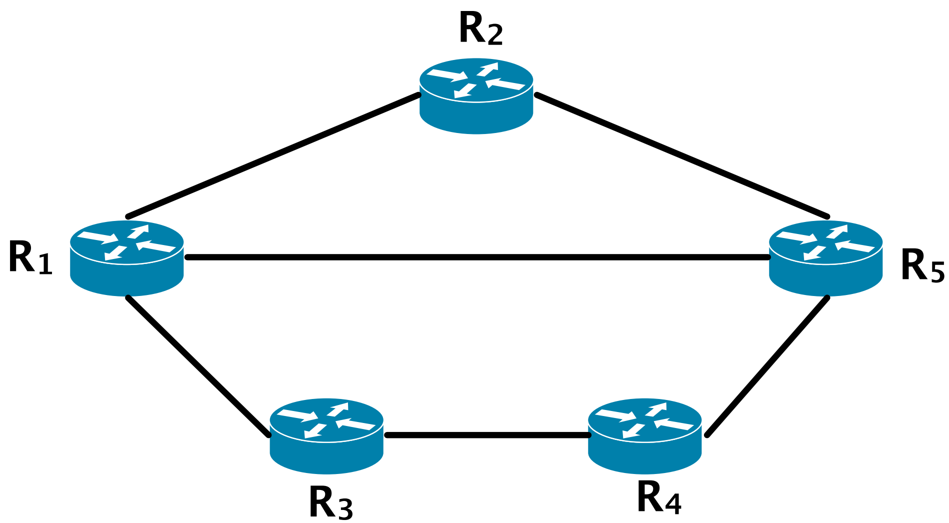
From metrics to functions



- Metrics are actually functions applied by routers on paths received
- Functions are defined on arcs and can be asymmetric
- Functions consider Metrics and Routing Policies ⇒ MRPC

MRPC timers

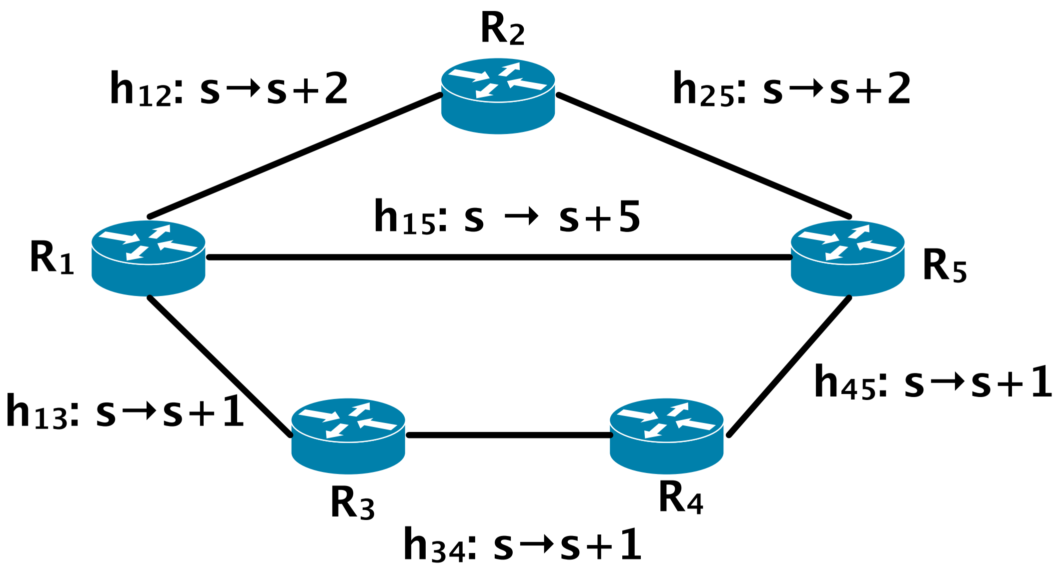
From metrics to functions



- Metrics are actually functions applied by routers on paths received
- Functions are defined on arcs and can be asymmetric
- Functions consider Metrics and Routing Policies ⇒ MRPC

MRPC timers

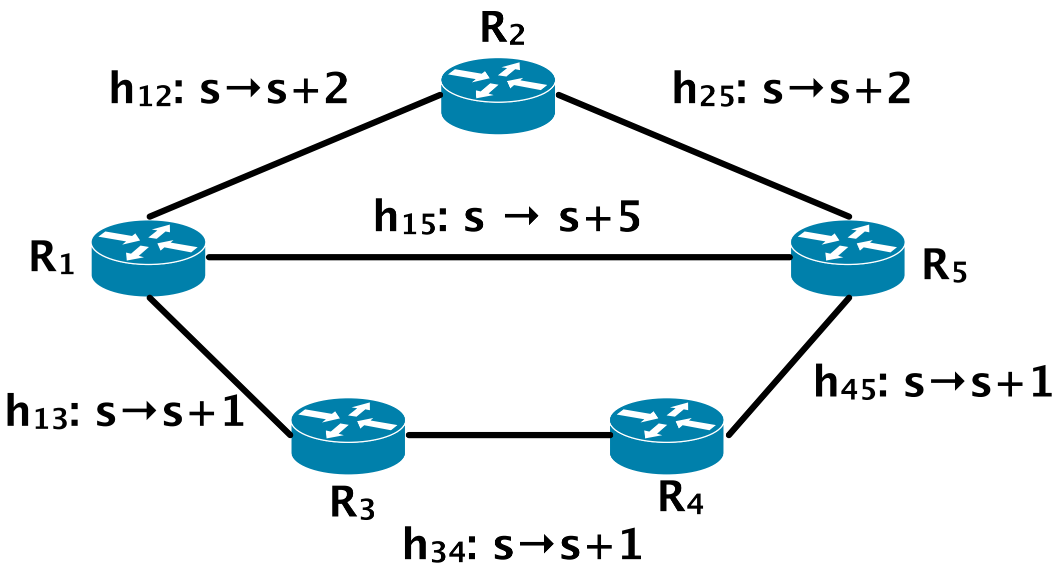
From metrics to functions



- Metrics are actually functions applied by routers on paths received
- Functions are defined on arcs and can be asymmetric
- Functions consider Metrics and Routing Policies ⇒ MRPC

MRPC timers

From metrics to functions

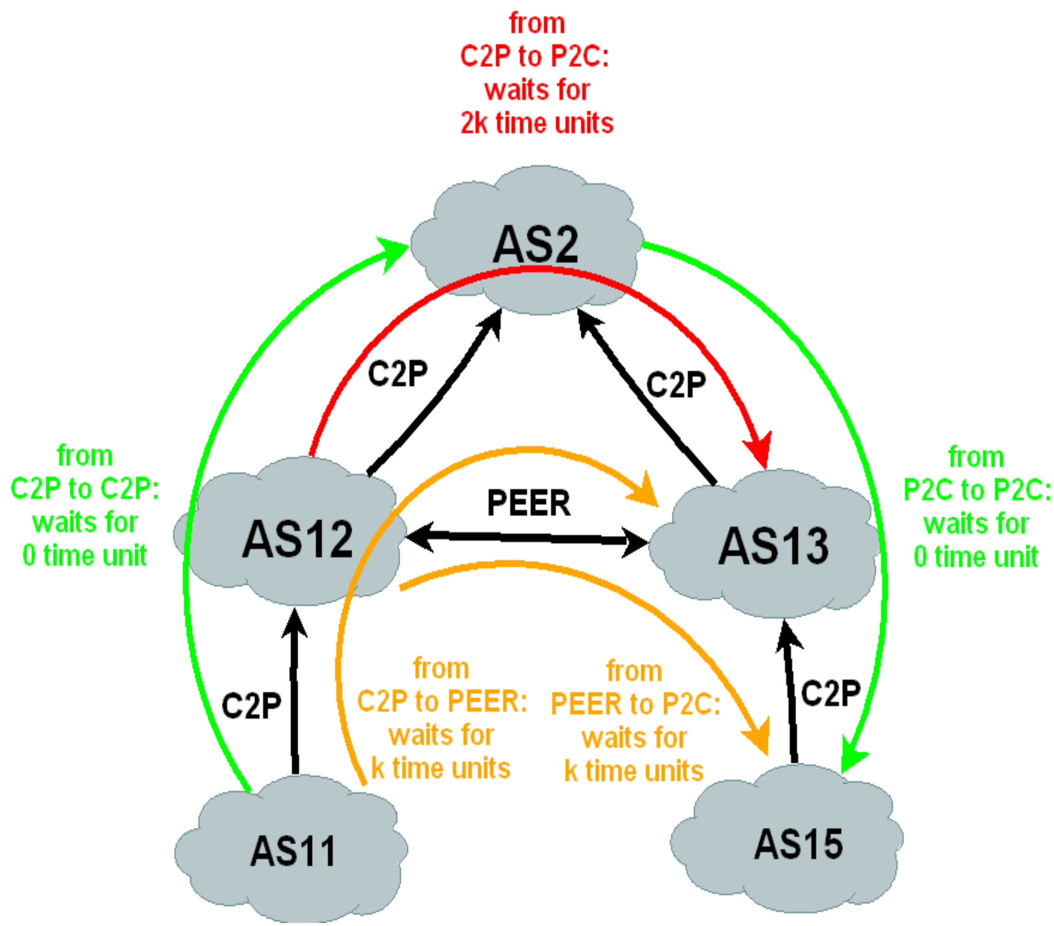


- Metrics are actually functions applied by routers on paths received
- Functions are defined on arcs and can be asymmetric
- Functions consider Metrics and Routing Policies ⇒ MRPC

- Contribution of the paper:
  - Study how to transform timer functions into real timers
  - Functional system to compute timers
  - Proof of correctness
- Simple metrics: section 4.1
- BGP: section 4.2

MRPC timers

Timers for BGP: local preference



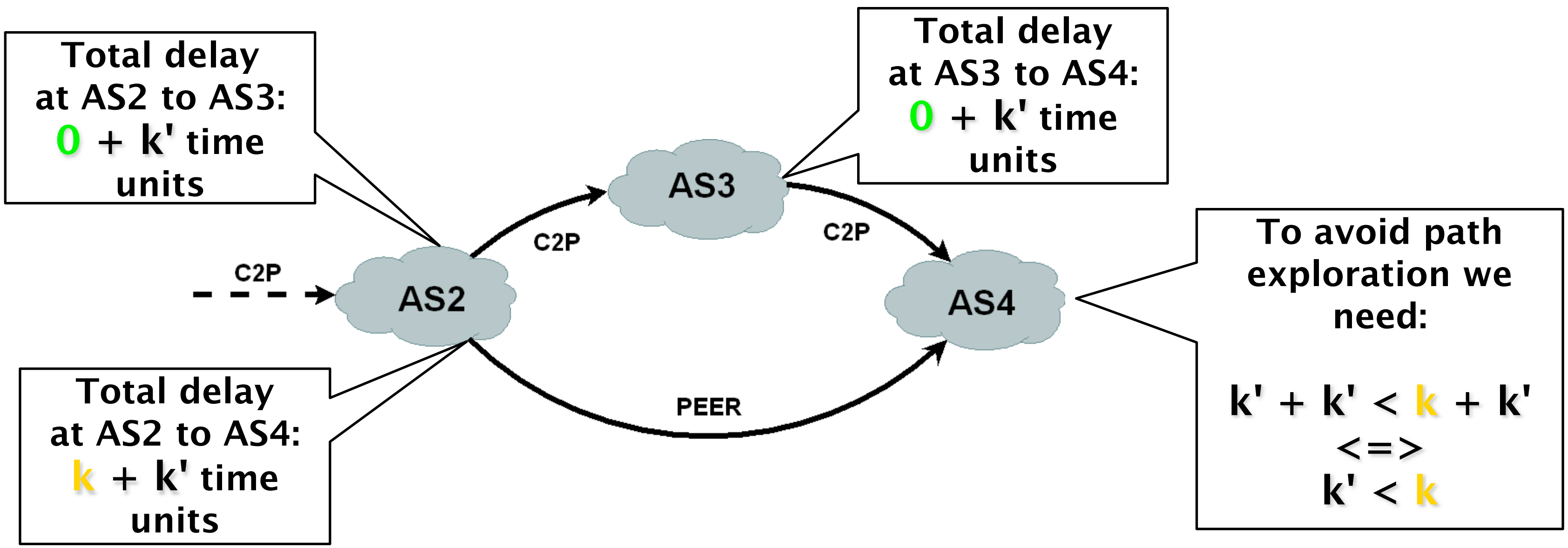
- Example of local preference model: C2P > PEER > P2C
- An AS delays routing updates according to the type of its incoming/outgoing neighbors:
  - C2P/C2P or P2C/P2C: 0
  - C2P/PEER or PEER/P2C: k
  - C2P/P2C: 2k
- Note: This is just one example. MRPC timers work with generic local-pref classes

MRPC timers

Timers for BGP: AS path

- In case of local preference tie-break, AS path length decides

➡ Delay a route by k' time units if you increase its length by k'



Outline

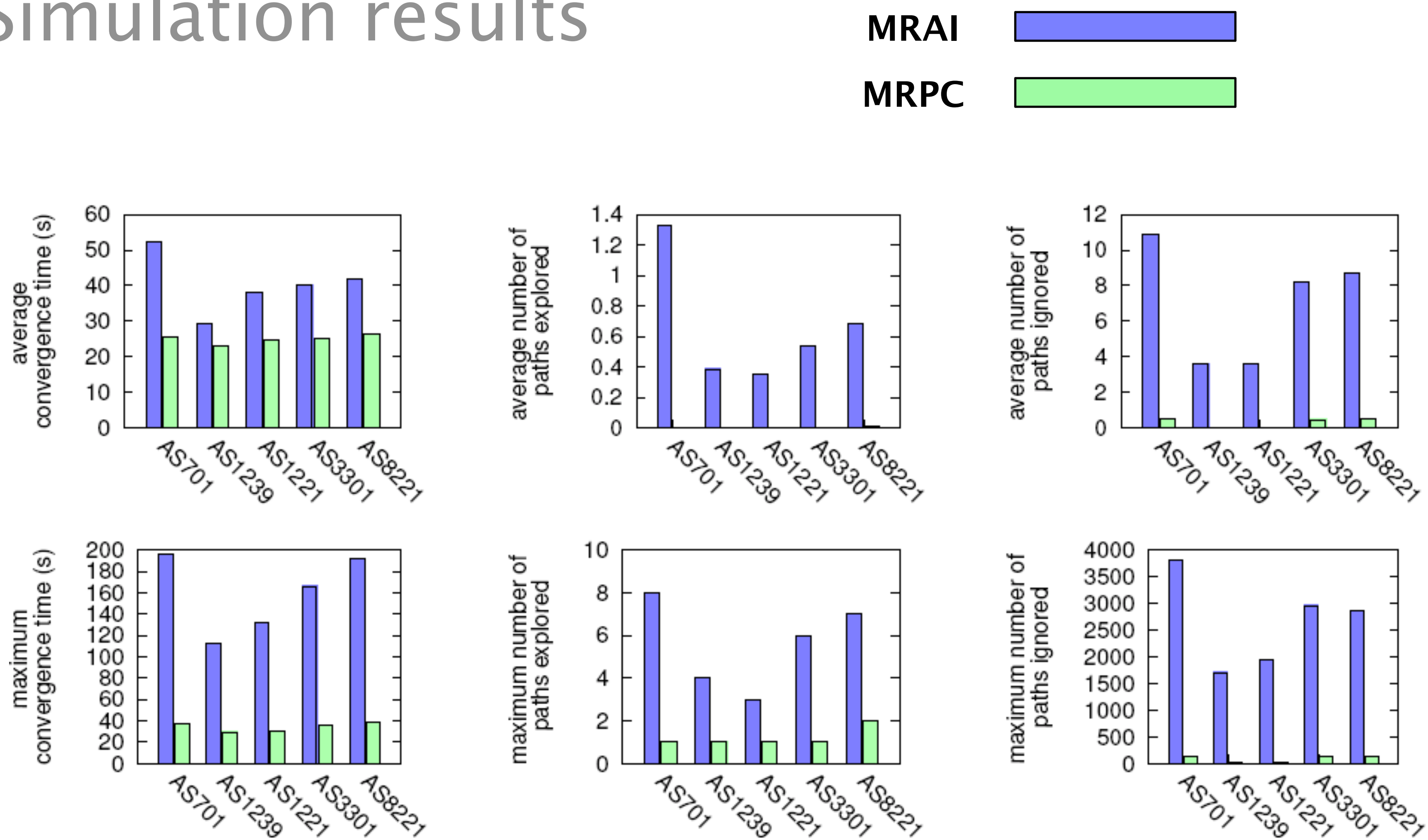
- Motivation
- Routing algebras
- MRPC timers
- Evaluation
- Arbitrary dynamics
- Conclusions

Evaluation
Setting

- AS-level topology from RouteViews (29,146 ASs and 78,934 edges)
- eBGP delay:
  - MRAI: [0,30[seconds
  - MRPC:
    - local-pref: 10 seconds for (c2p,peer) or (peer,p2c) and 20 seconds for (c2p,p2c)
    - AS path: 0.1 second per AS hop
- iBGP delay: [0,1[second
- Advertise a prefix from different ASs (tier-1, tier-2, stub) and measure propagation properties

Evaluation

Simulation results



Outline

- Motivation
- Routing algebras
- MRPC timers
- Evaluation
- Arbitrary dynamics
- Conclusions

Arbitrary dynamics

1. Ghost flushing
   - Do not delay withdraws of obsolete paths
   - Make sure timers give enough time to flush obsolete paths
2. Originator synchronization
   - If path to be installed is worse the previous best
   - Apply MRPC delay on new path metric
   - Wait for this delay before installing the route in the RIB (or FIB for the IGP)

- 1 + 2 ⇒ path exploration and loops are avoided in all situations

Outline

- Motivation
- Routing algebras
- MRPC timers
- Evaluation
- Arbitrary dynamics
- Conclusions

Conclusions

- BGP propagation is arbitrary today
- Bringing order to BGP is possible: enforce a proper ordering of routing messages during propagation
- We proposed new timers, called MRPC:
  - No need to reveal routing policies
  - Down-scaling possible to reduce convergence time
  - Drastically reduce path exploration
  - Backward compatible: wider deployment means more gain in terms of convergence properties
  - Guarantees of proper forwarding behavior