Improving Message Throughput in Akka.NET

Improving Message Throughput With Routers

Jason Roberts

@robertsjason | dontcodetired.com

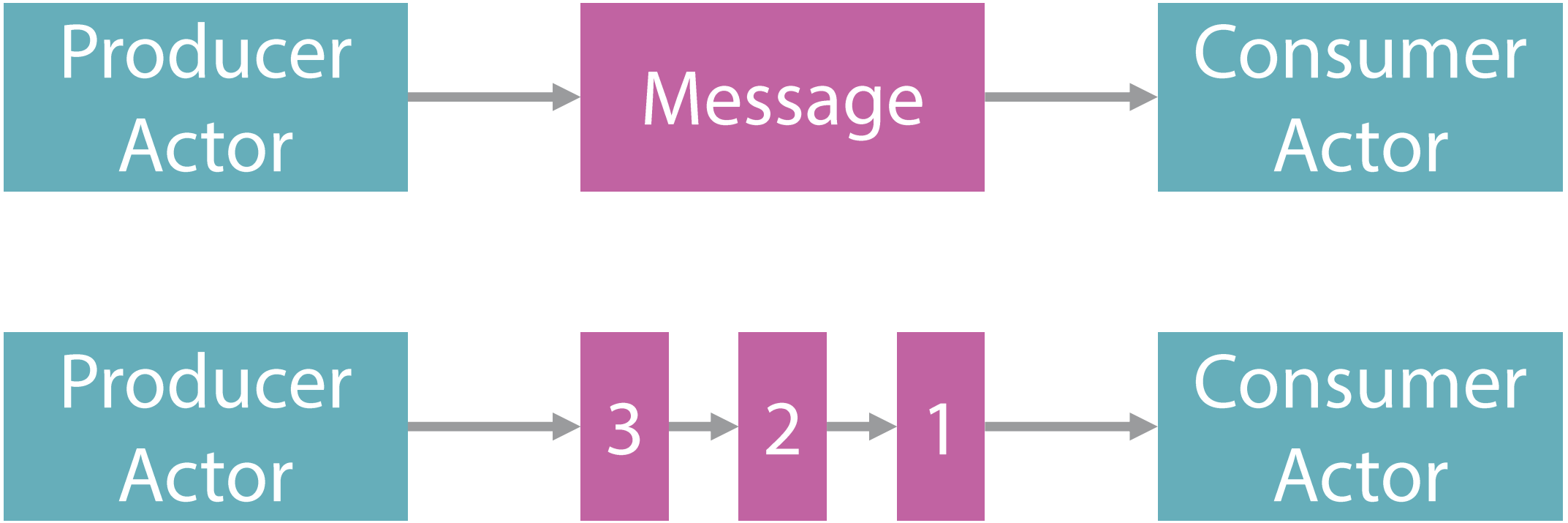
Overview

- What routers do and why use them?
- Two types of routers
- Overview of routing strategies
- Supplied routing strategies
- Demo application (no router)
- Add a router to improve concurrency
- Configuring routers using HOCON
- Auto dynamically resizing pools

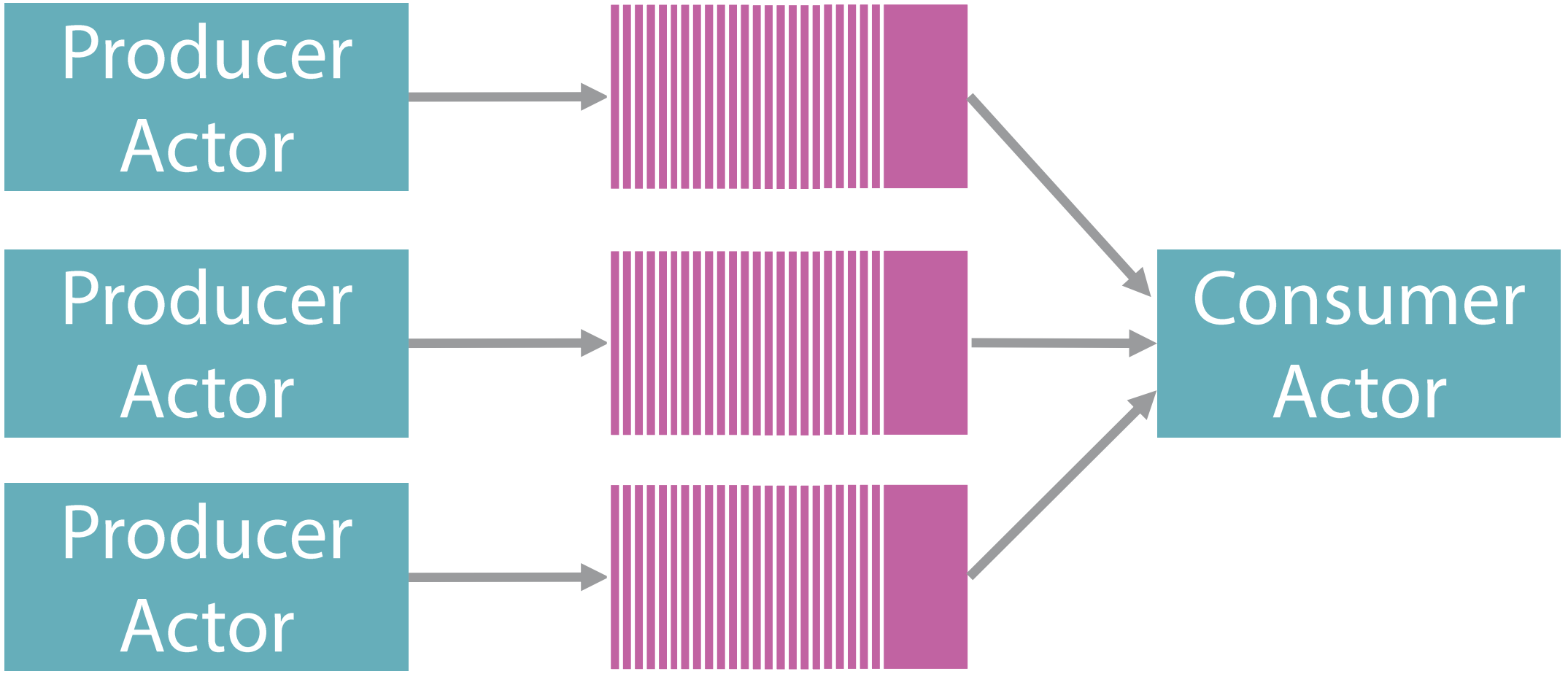
Suggested Prerequisites

- Akka.NET
  - Defining actors
  - Sending/receiving messages
  - Supervision hierarchies
  - "Building Concurrent Applications with the Actor Model in Akka.NET" course
- Dependency Injection
  - General understanding of DI (e.g. via constructor parameters)
  - DI in Akka.NET
  - "Implementing Logging and Dependency Injection in Akka.NET" course

Overview of Routers



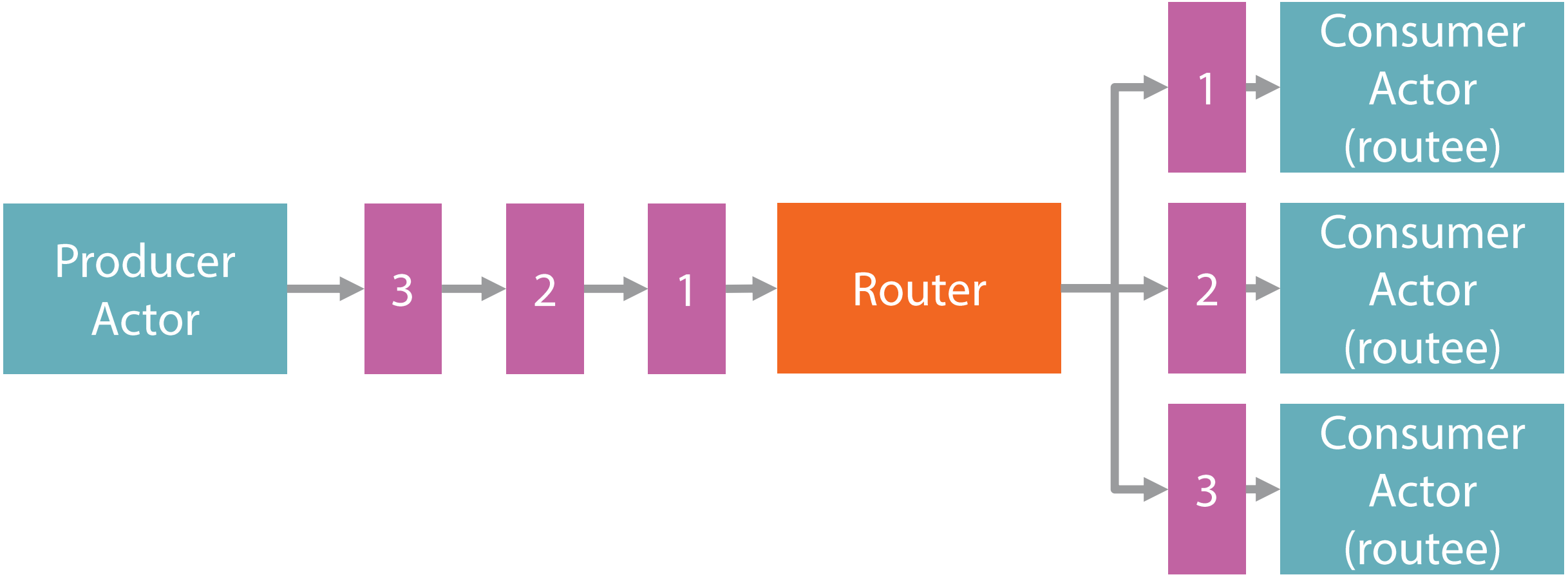
Overview of Routers



A router is a special type of actor whose job is to route messages to other actors called routees.

— getakka.net

Overview of Routers



Types of Routers

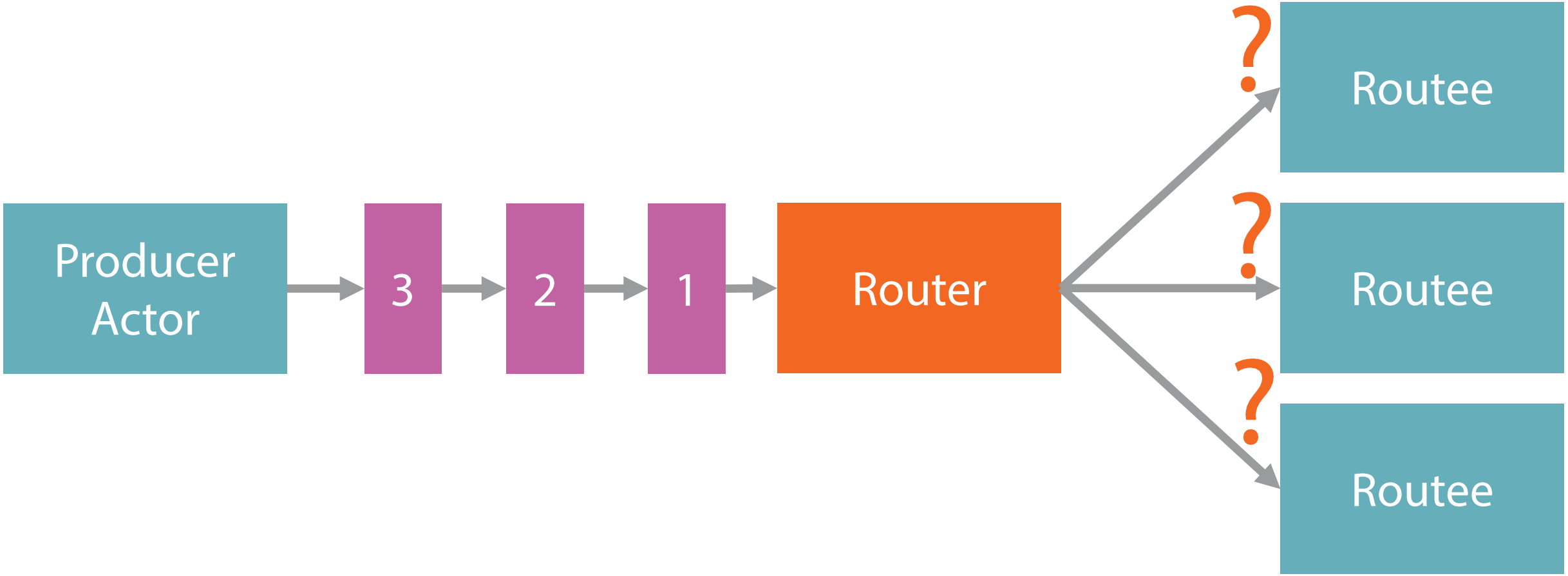
Pool Routers

- Create own routees
- Supervise routees
- Only one type of routee actor
- Automatic dynamic pool resizing
- Generally recommended (know if routees die)

Group Routers

- Routees created elsewhere
- Routees supervised elsewhere
- May have different routee types
- No automatically created pool

Overview of Routing Strategies



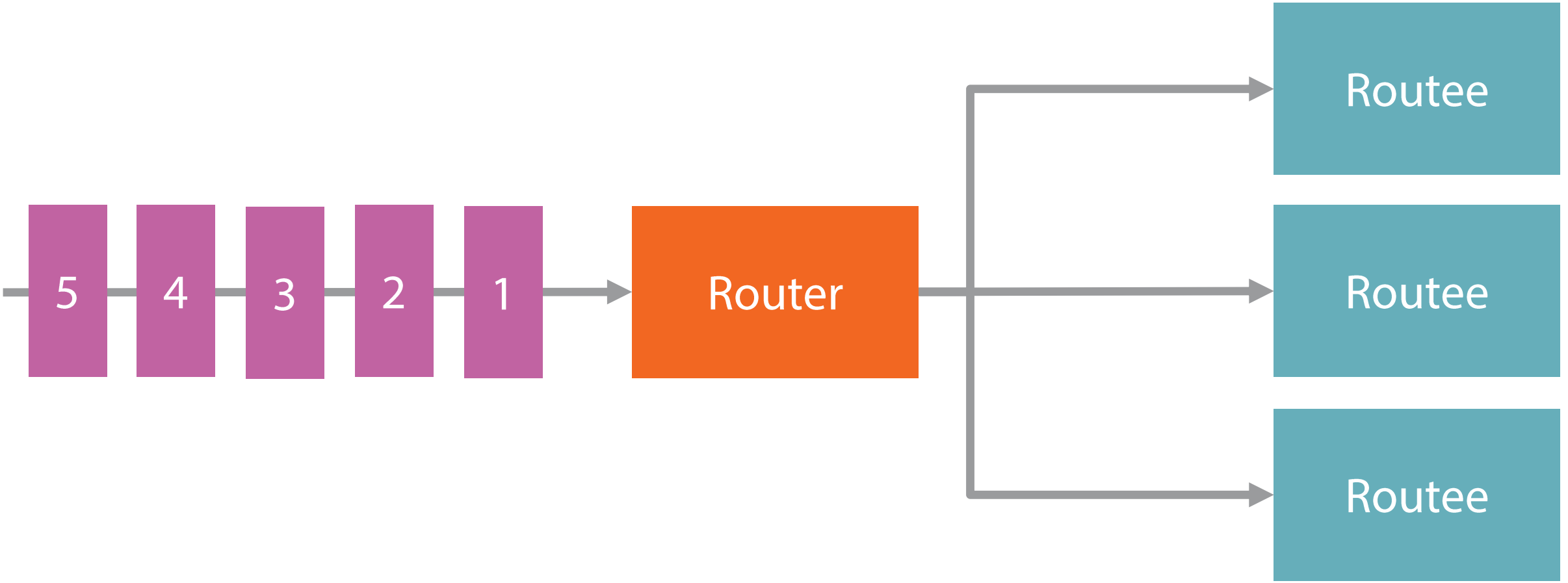
Routing Strategy

The logic that determines how the router decides which routee or routees an incoming message will be forwarded on to.

Some routing strategies are not available in both Pool and Group router types

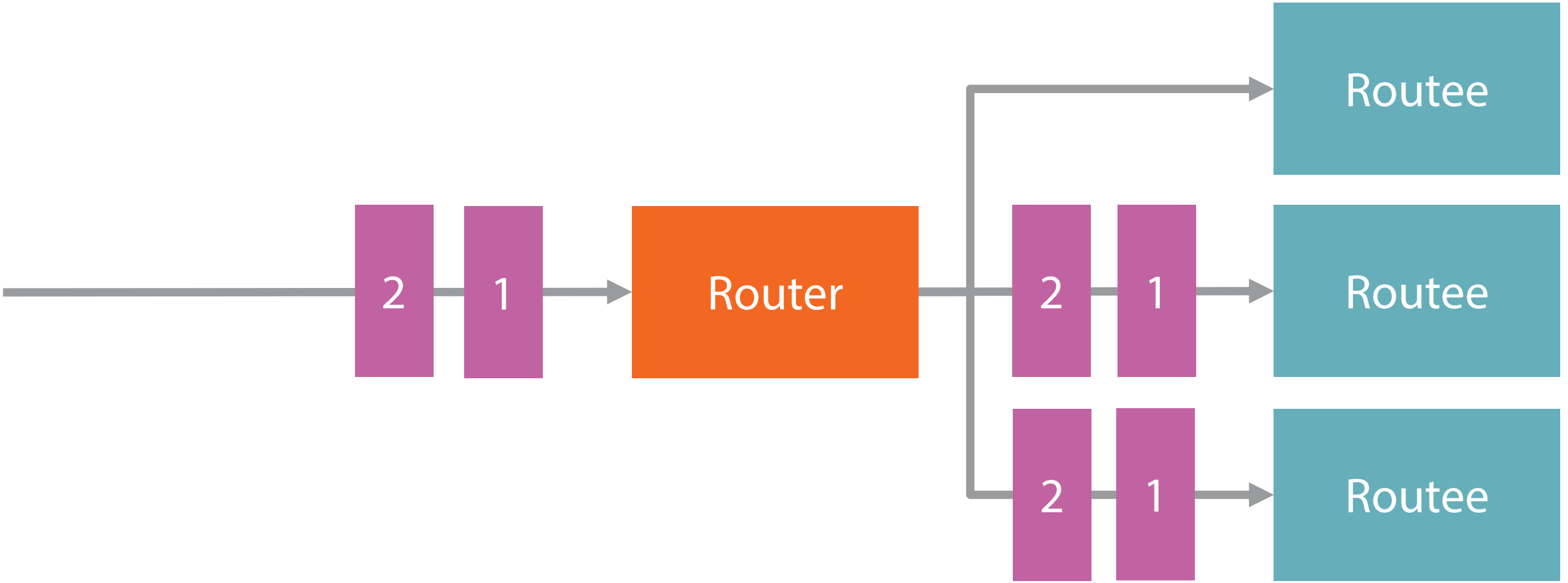
Supplied Routing Strategies

RoundRobin



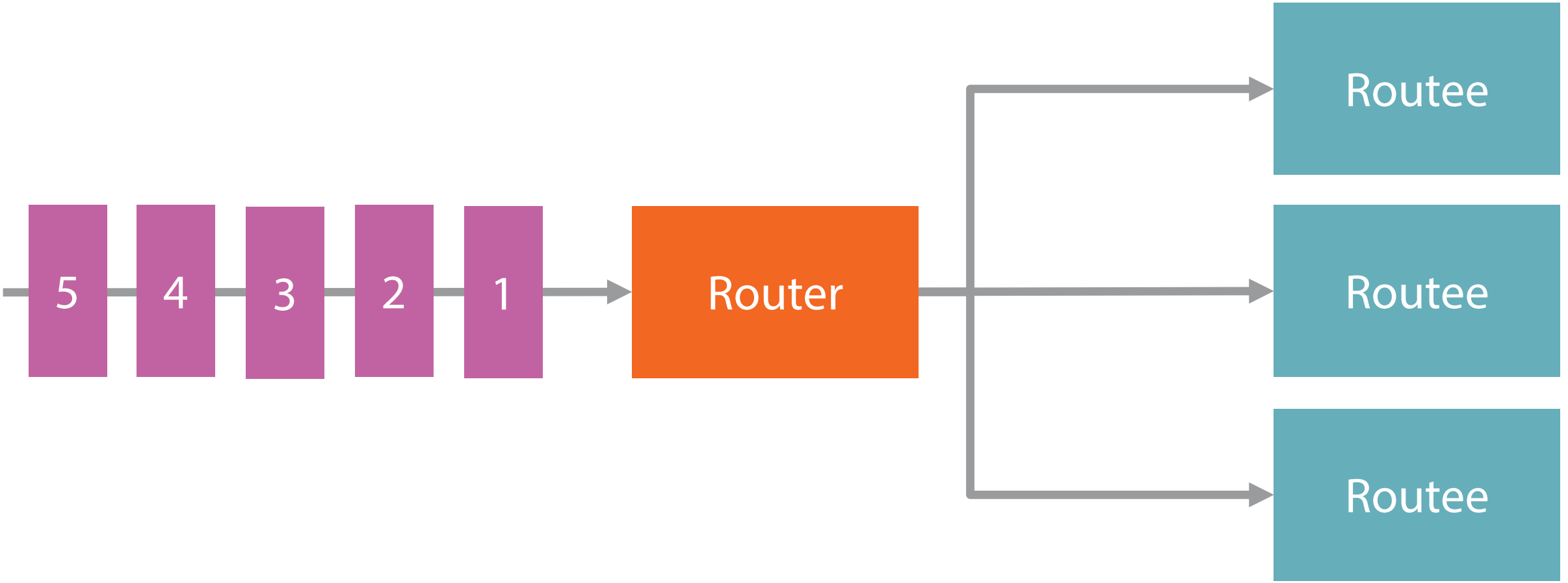
Distribute messages in round robin order

Broadcast



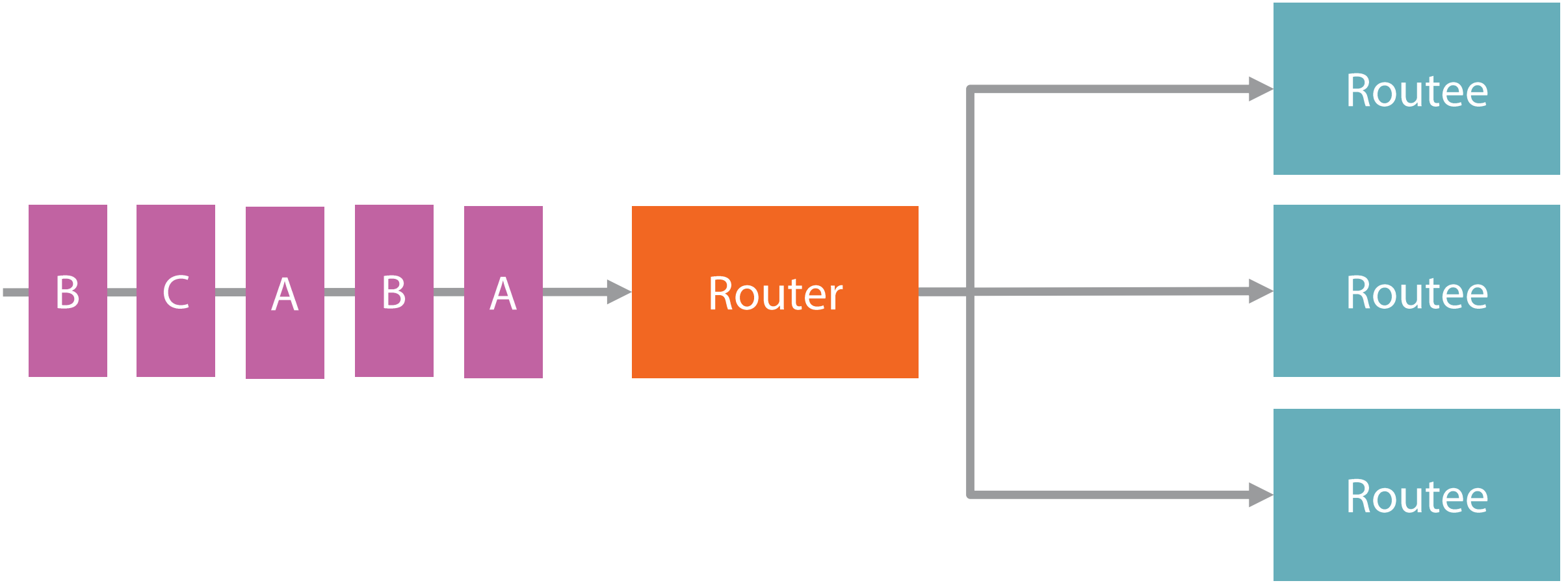
Distribute every message to every routee

Random



Distribute messages to random routee

ConsistentHashing



Distribute messages with the same key to same routee

Advanced Routing Strategies

- TailChopping
  - Send message to random routee
  - If routee does not respond, send to another random routee
  - Forward reply from routee that responds first back to the sender
  - Ignore any subsequent routee replies
  - Use to decrease latency in responding to the sender (e.g. queries)
- ScatterGatherFirstCompleted
  - Send message to all routees at same time
  - Forward reply from routee that responds first back to the sender
  - Ignore any subsequent routee replies

Advanced Routing Strategies

- SmallestMailbox
  - Send message to the routee that has the smallest number of inbox messages
  - Pool routers only (i.e. not Group)
  - Order of precedence for routee selection:
    1. Routee with empty mailbox and not currently processing a message
    2. Routee with empty mailbox
    3. Routee with smallest number of inbox messages
    4. Routee in remote process / system

Demo Application Overview

Using a Group Router

Create 3 PaymentWorkerActors

Modify JobCoordinatorActor

RoundRobin Group

Using a Pool Router

Remove 3 PaymentWorkerActors

Modify JobCoordinatorActor

RoundRobin Pool

Pool size

Configuring Routers with HOCON

Automatic Dynamic Pool Resizing

Summary

- Increase message throughput
- Pool and Group routers
- Supplied routing strategies
- Demo application (no router – 5 seconds)
- RoundRobin Group/Pool routers (2 seconds)
- Configuring routers using HOCON
- Auto dynamically resizing pools

Next:

Regulating Message Processing with Stashing