Latency-Aware Strategies for Placing Data Stream Analytics onto Edge Computing

Alexandre da Silva Veith, Marcos Dias de Assunção, Laurent Lefèvre
Avalon, LIP, ENS Lyon, University of Lyon - Lyon, France
alexandre.veith@ens-lyon.fr

Scenarios

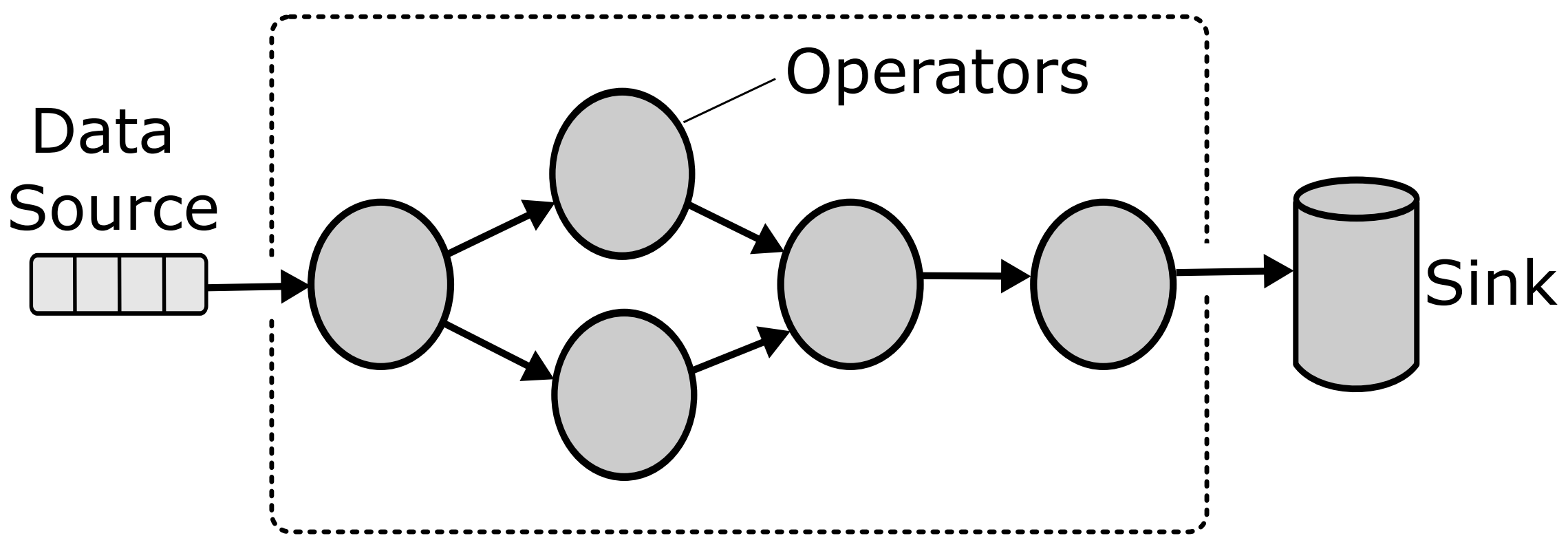
Monitoring of operational infrastructure

Anomaly detection, fraud detection

Social networks

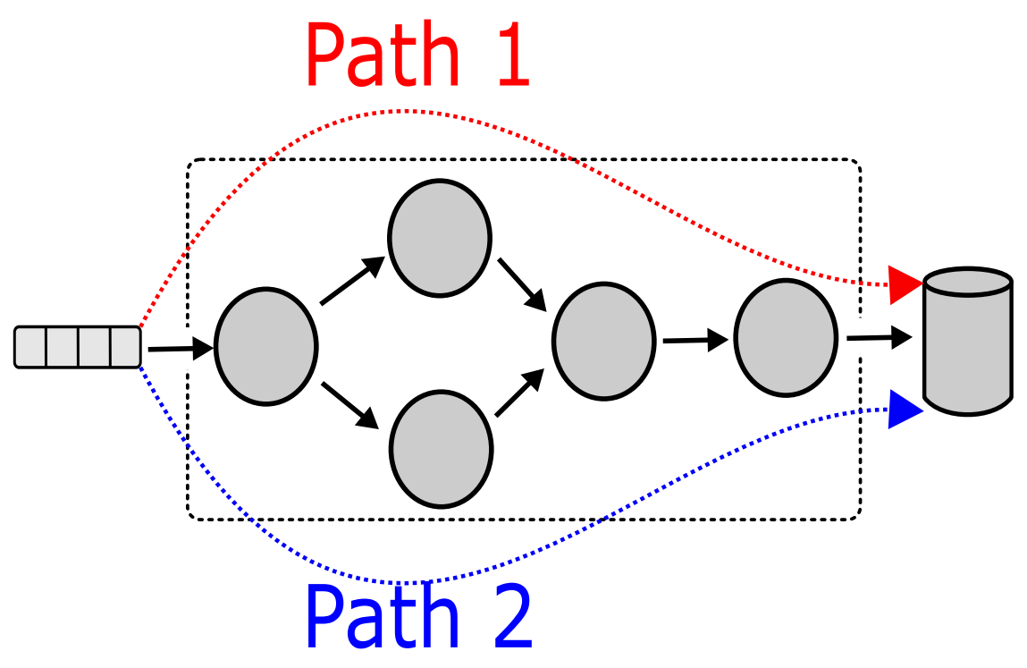
Internet of Things (IoT)

Applications



Directed dataflows whose vertices are operators that execute a function over the incoming data and edges define how data flows between them

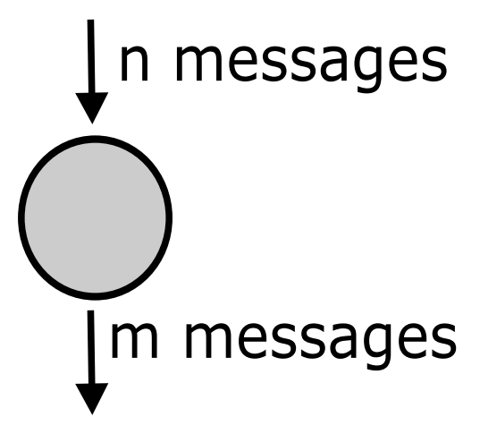
Operator Characteristics



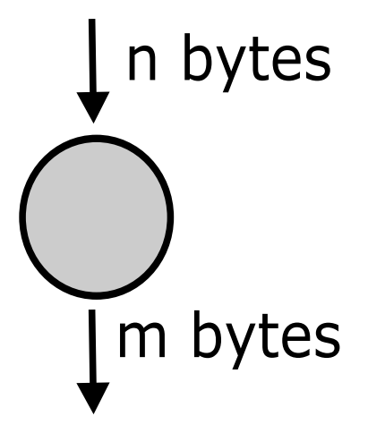
Paths and location: multiple sources and sinks distributed across cloud and/or edge

Fork/ Split: messages can be replicated or scheduled to downstream operators

Merge/ Join: merges the outcome of upstream operators



Selectivity: The ratio of number of input messages to output messages



Data compression/ expansion factor: The ratio of the size of input events to the size of output events

Response time: the total time taken for a message to traverse a path

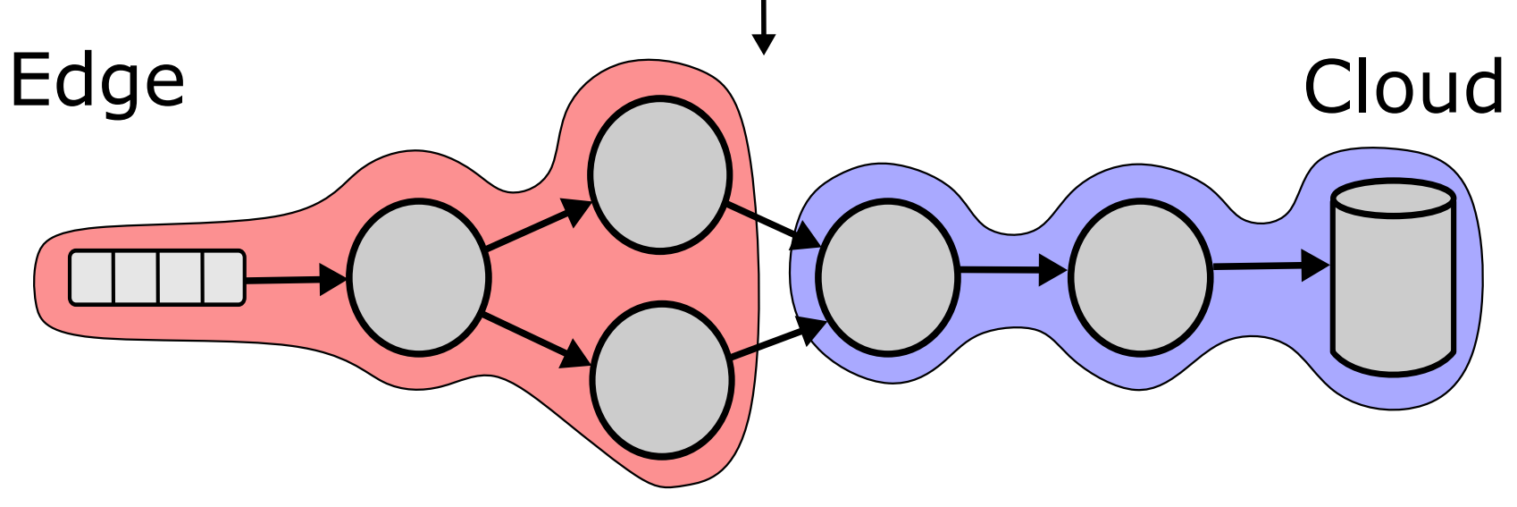
Latency-Aware Decisions

Operator Characteristics and Requirements (CPU, Memory, Bandwidth)

Minimize the response times in all paths

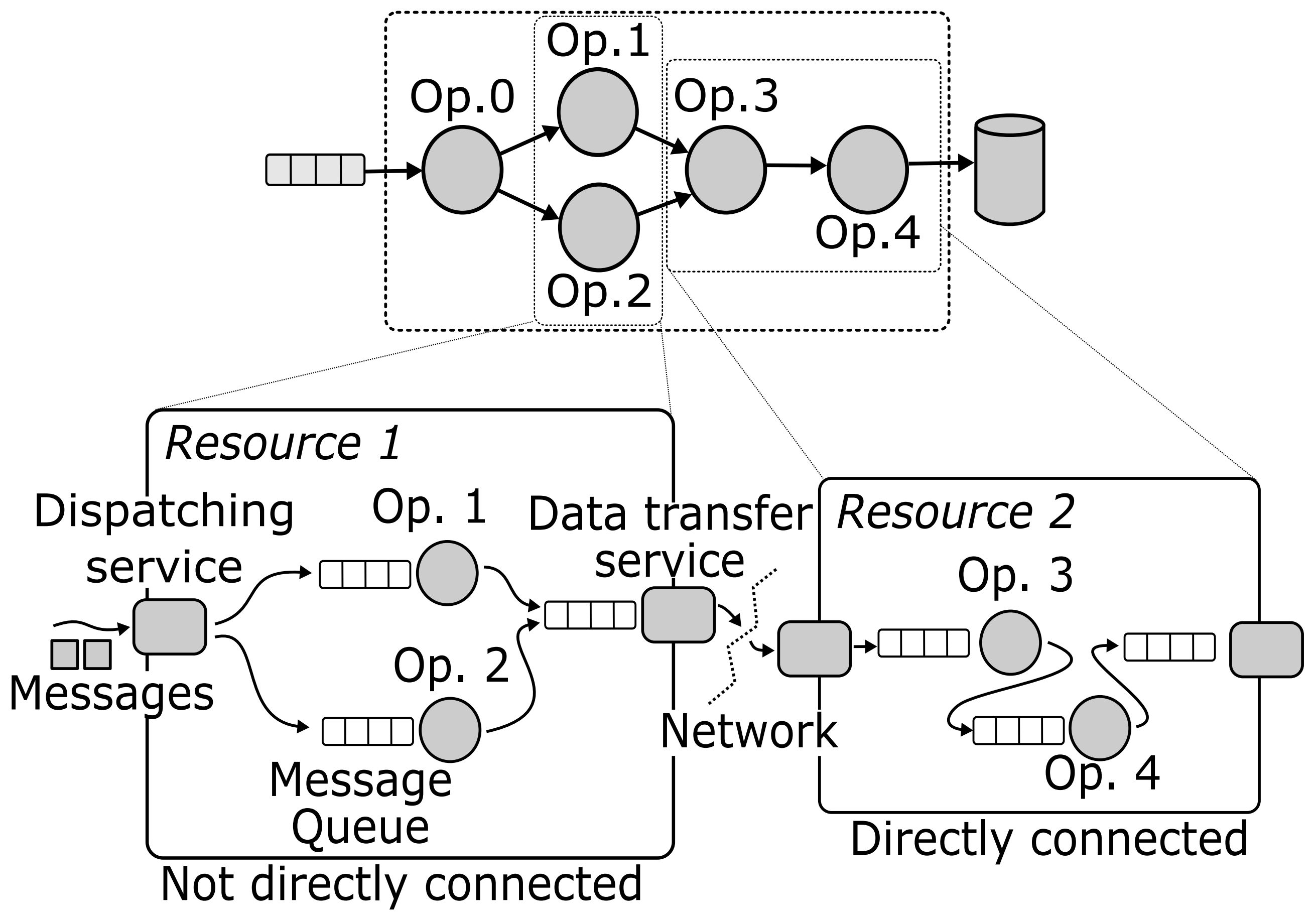
Edge and Cloud Capabilities (CPU, Memory, Bandwidth, Latency)

Application Splitting Across Edge and Cloud



Resource and Application Models

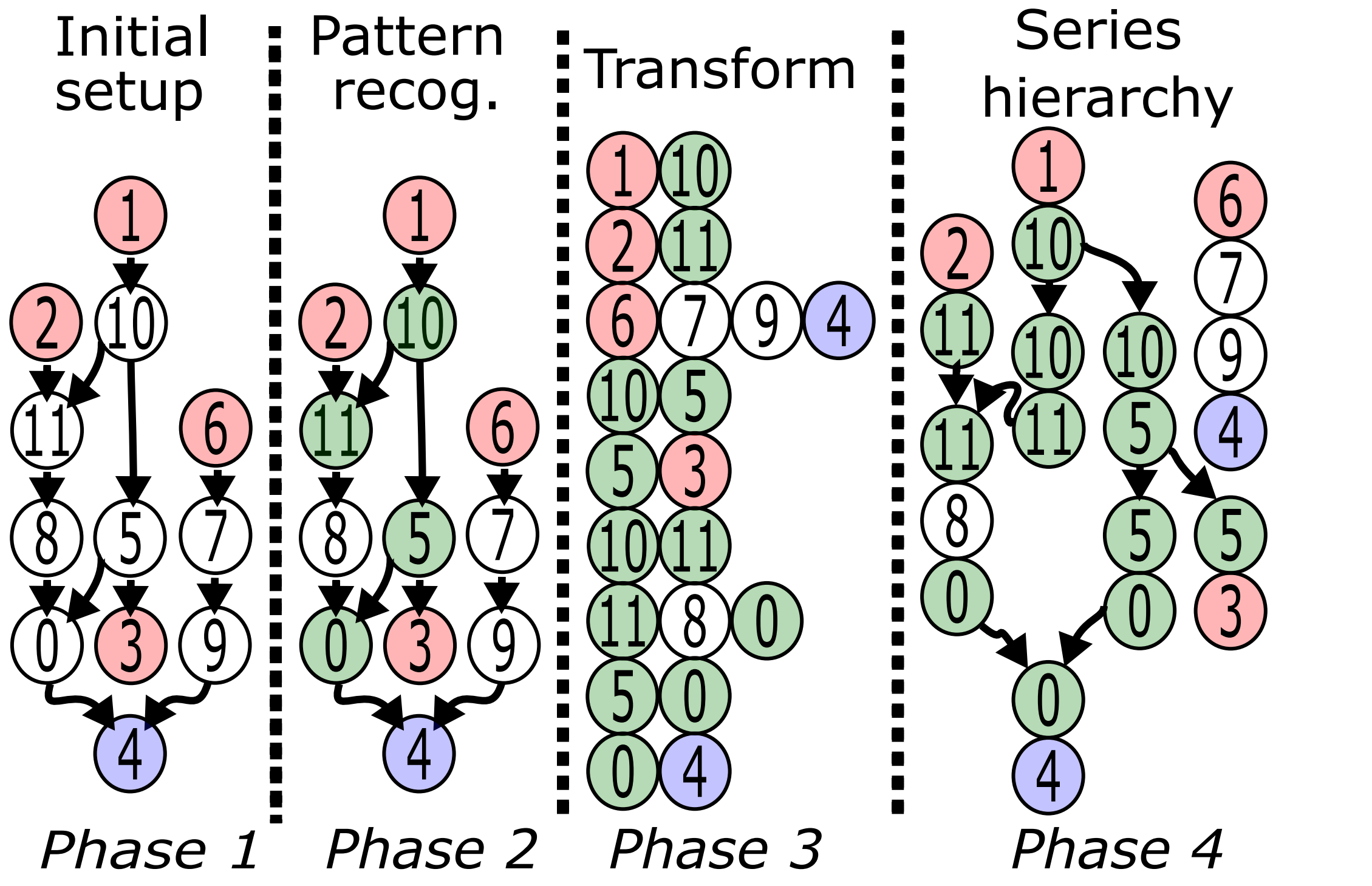
Example of operator placement



Operators - communication and computation services

Services - M/M/1 queues

Finding Application Patterns



- Find the dataflow split points (green)
- Edge (red) and cloud (blue) placements

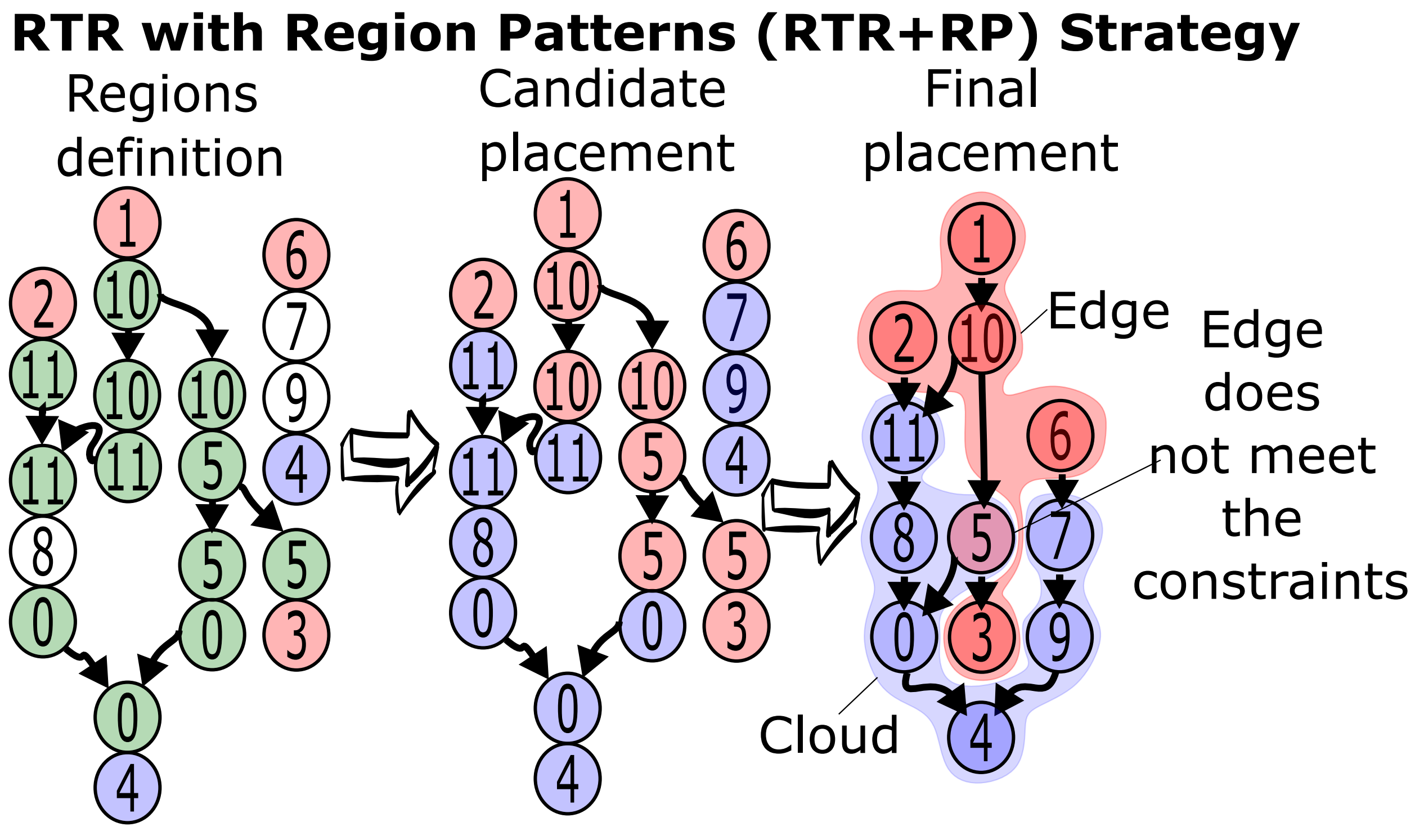
Response Time Rate (RTR) Strategy

1) BFS-Traversal algorithm = operator sequence

2) For each operator in the sequence

- Response time estimation for all resources
- Host with shortest response time is elected to host the operator

RTR with Region Patterns (RTR+RP) Strategy



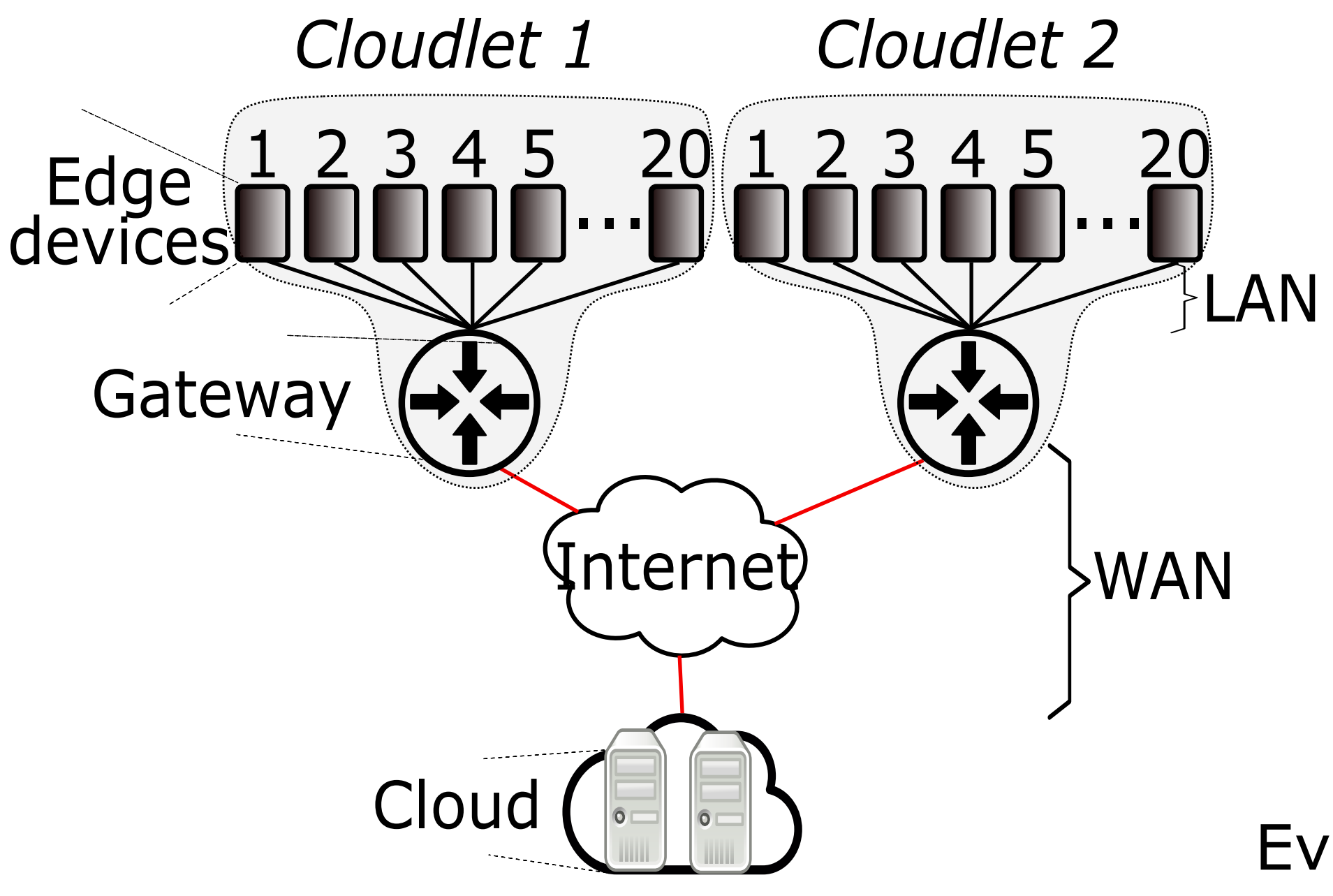
Operator sequence = RTR

Candidate placement = destination infrastructure of the message

- Only cloud = cloud
- Edge and/or cloud = edge

Response time estimation - only for edge candidates

Experimental setup



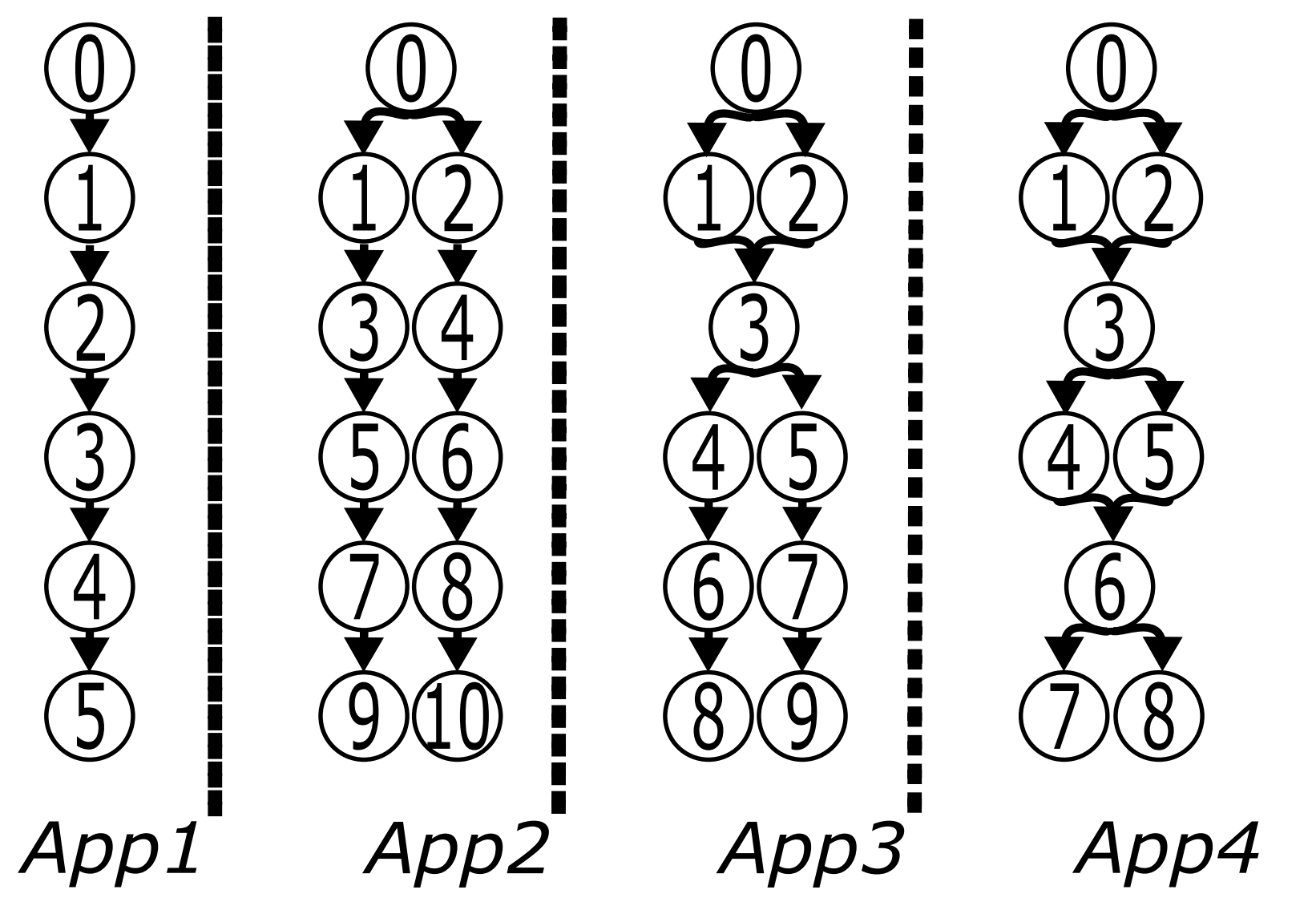
Edge device: Two cloudlets with 20 Raspberry PI 2 (4,74 MIPS at 1GHz and 1GB of RAM)

Cloud: Two AMD RYZEN 7 1800x (304,51 MIPS at 3.6GHz and 1TB of RAM)

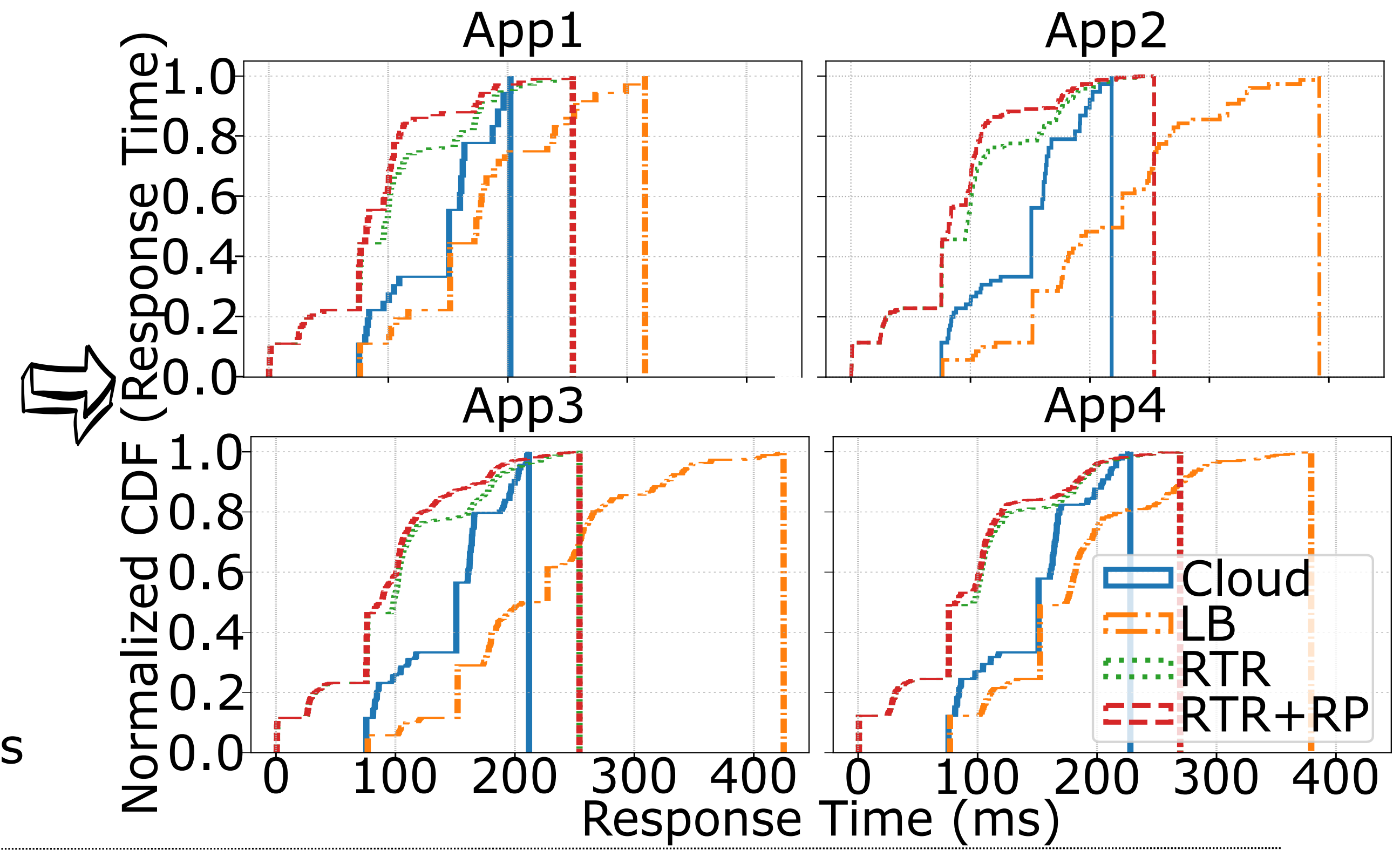
LAN:
- Latency: $U(0.015\text{-}0.8)$ms
- Bandwidth: 100 Mbps

WAN:
- Latency: $U(65\text{-}85)$ms
- Bandwidth: 1 Gbps

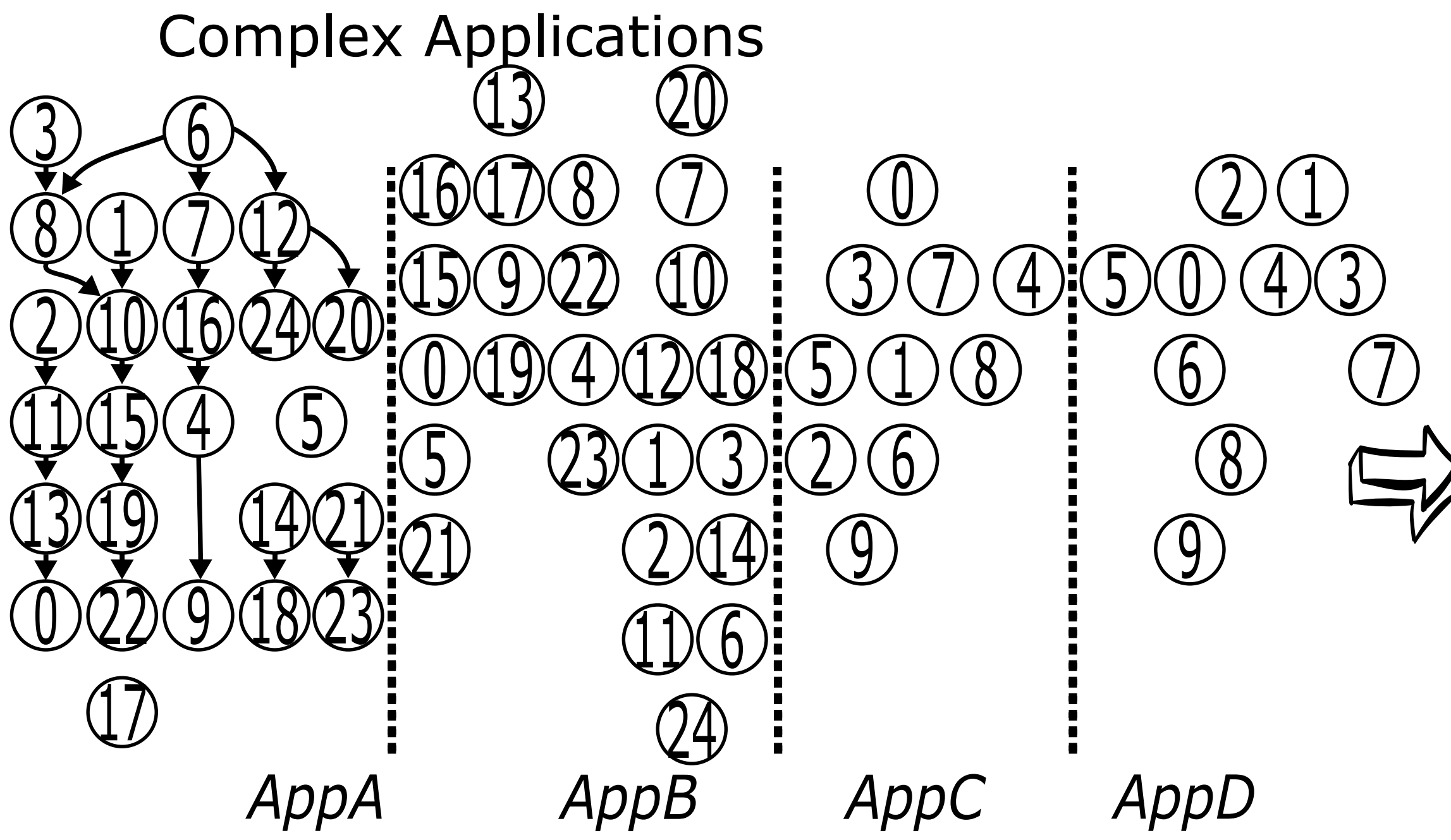
Microbenchmarks



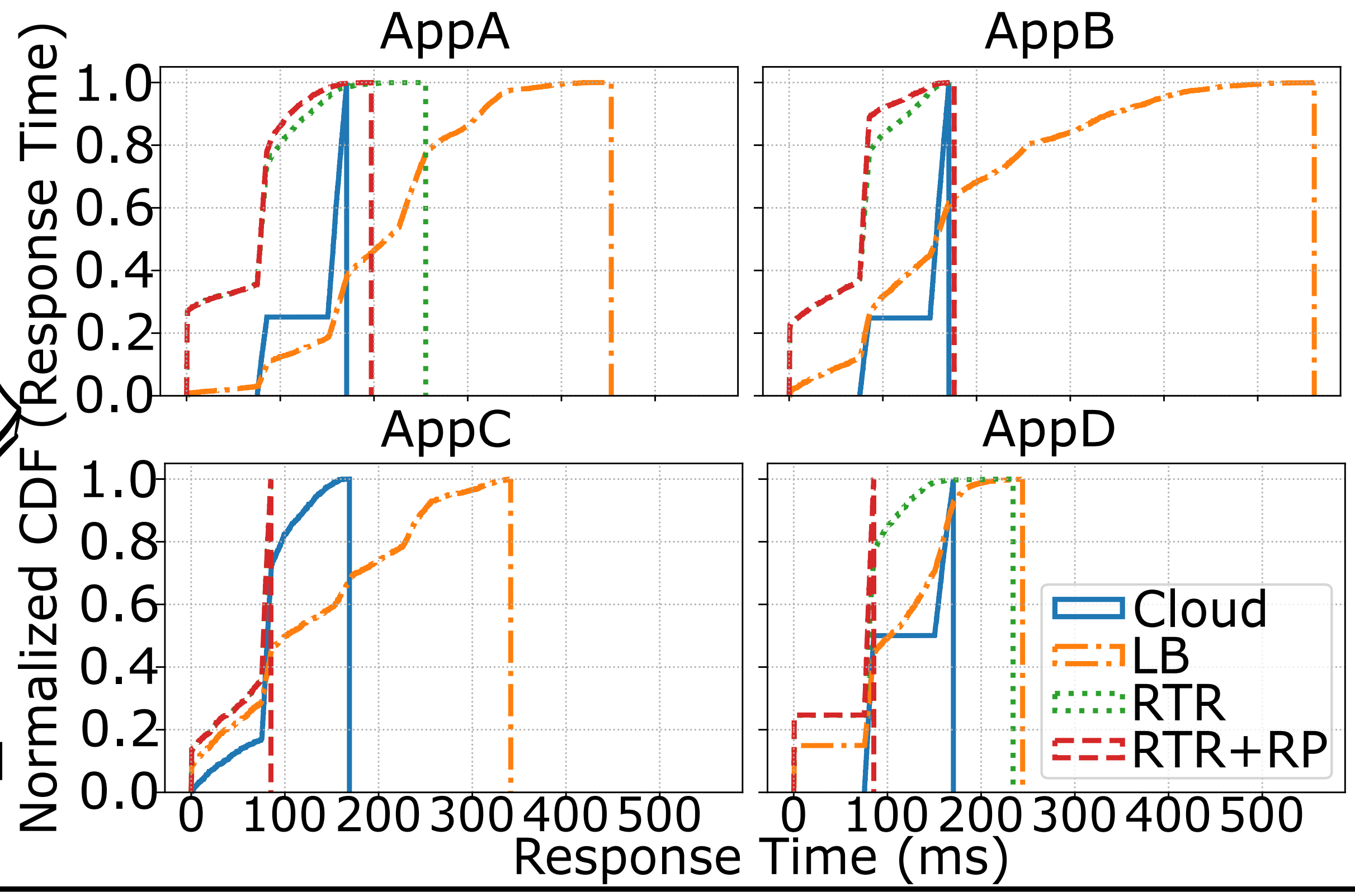
Evaluate the impact of fork/join operators



Complex Applications



Investigate the outcomes for generic and multiple path applications



Analysis

Our approach:

Microbenchmarks
- 95% better

Complex Applications
- 50% faster in average
- 52% of reduction in the communication for sinks located in the edge

Conclusions

- Dynamically movement of operators across edge and cloud

- 50% faster in generic and complex datafows

Reference

[1] Taneja, M., Davy, A.: Resource aware placement of IoT application modules in fog-cloud computing paradigm. 2017

Acknowledgment: This work was performed within the framework of the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).