CAF: A Computing-aware Adaptive Forwarding Strategy in Named Data Networking

Jian Yang[1], Teng Liang[1], Beichuan Zhang[2], Yu Zhang[1], Weizhe Zhang[1]

[1]Peng Cheng Lab, [2]The University of Arizona

NDNComm'25
UCLA, Los Angeles, CA

April 2025

Remote Computation In NDN

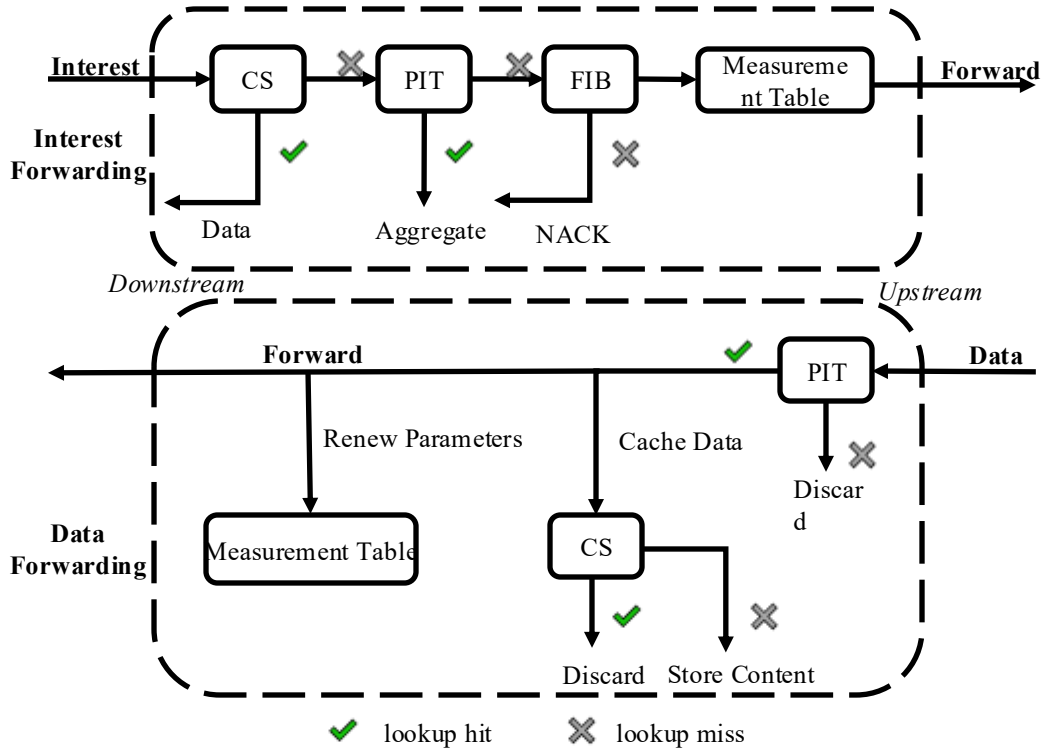
- Multiple use-cases to prove that remote computation is important
  - E.g., edge/fog computing, IoT, VR/AR
- In NDN, multiple frameworks for remote computation exist
  - E.g., NFN, NFaaS, CFN, RICE
- What architectural benefits can we get from NDN to better support remote computation?

What is this research about?

- We investigate NDN's **adaptive forwarding** in remote computation
- One potential benefit: in-network load balance

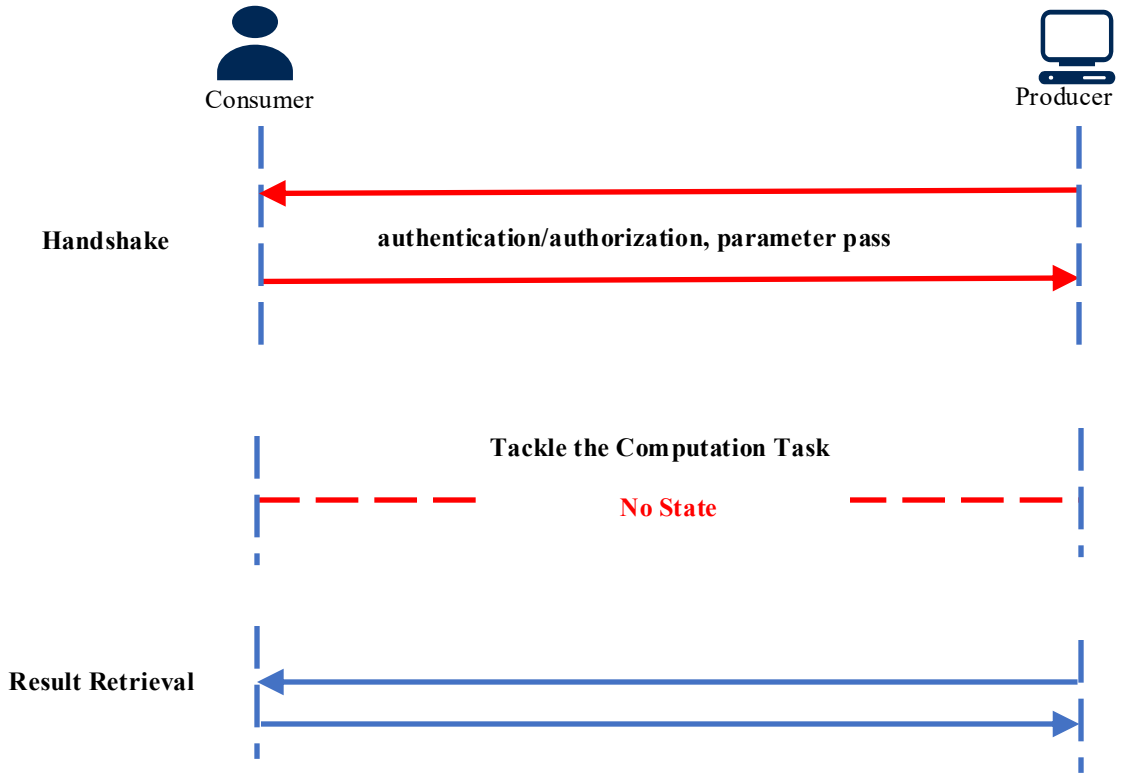
NDN Adaptive Forwarding

- **NDN forwarding is stateful and has adaptability**
  - Observe past data retrieval measurement on multiple paths
  - use it to improve forwarding decisions for future Interests



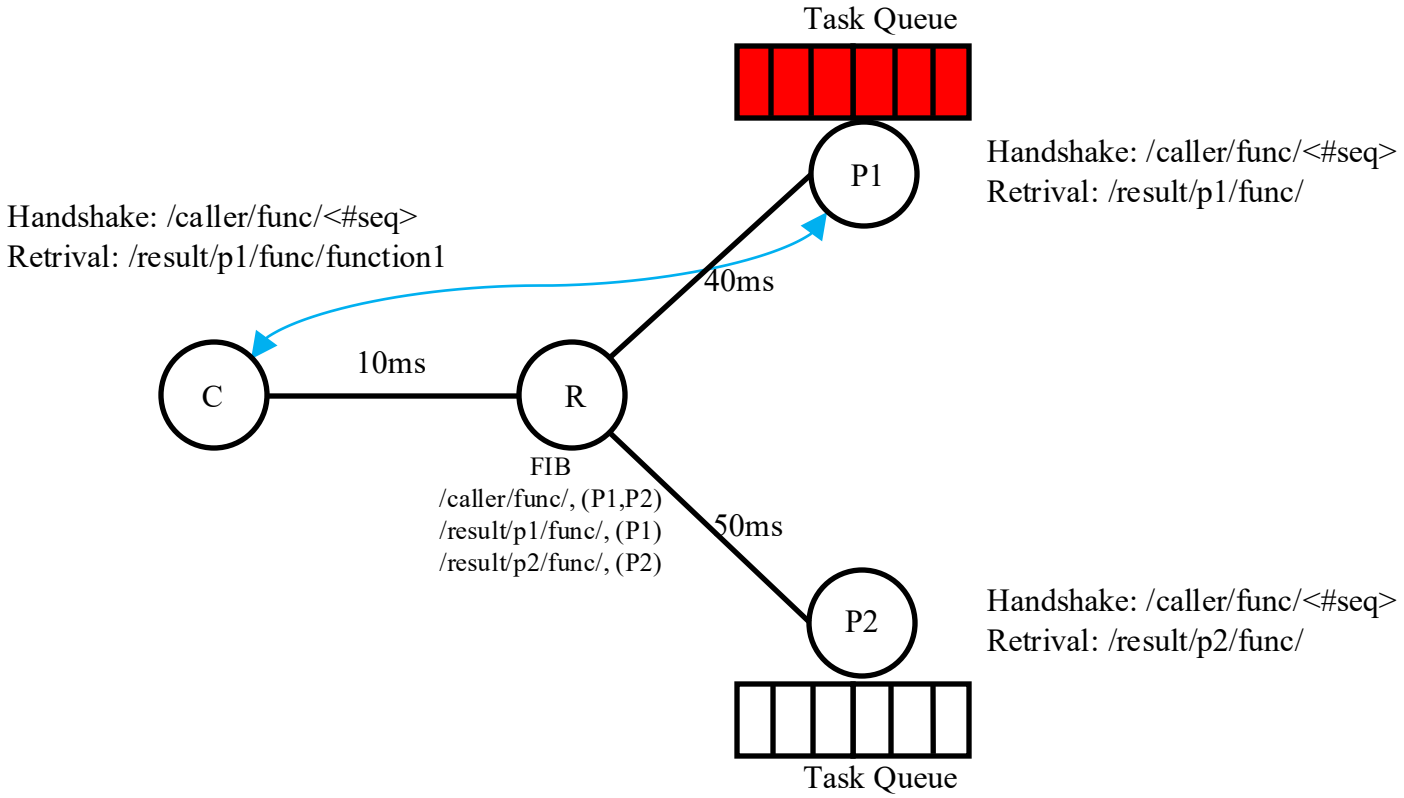
RPC Framework (RICE, NSC, etc.)

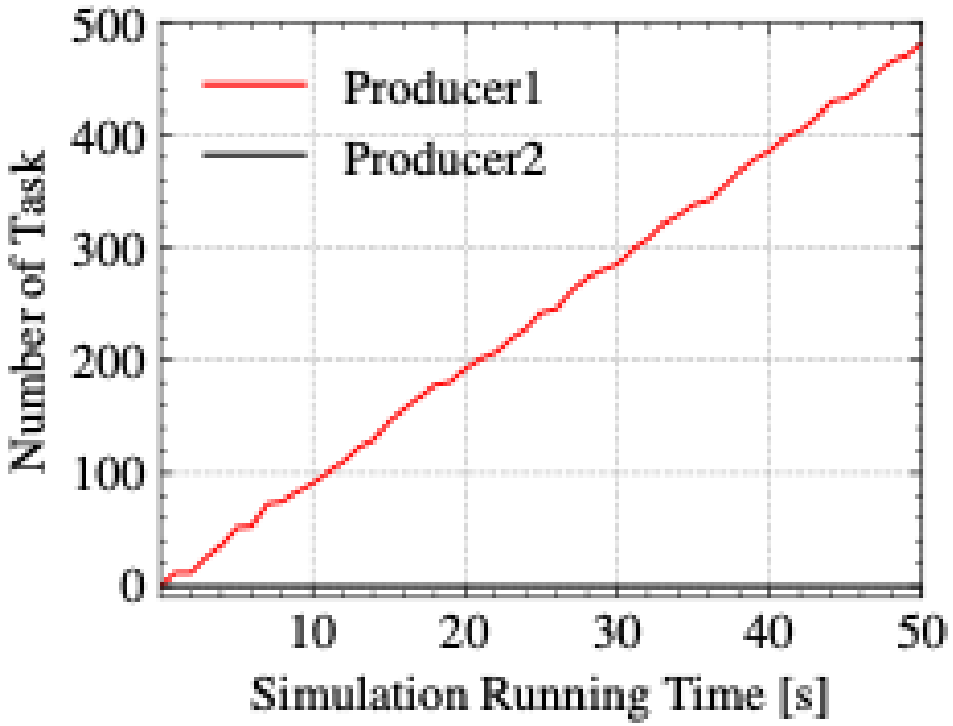
- Decouple computation stage from communication
  (**two stage communication partten**)
- Multiple Interest-Data exchange
- Different name prefixes



The Path Load Aggregation Problem

- Existing forwarding strategies do not understand NDN-based RPC frameworks
- Put them together leads to The Path Load Aggregation Problem





simulation result

The Path Load Aggregation Problem

- NDN adaptive forwarding measure the metric by a Interest-Data exchange
- RPC framework decouple a remote computation request into two stages
  - the adaptive forwarding plane cannot observe the change in producer
- Problem: The adaptive forwarding plane cannot observe the change in producer

We define it as the **path load aggregation problem**

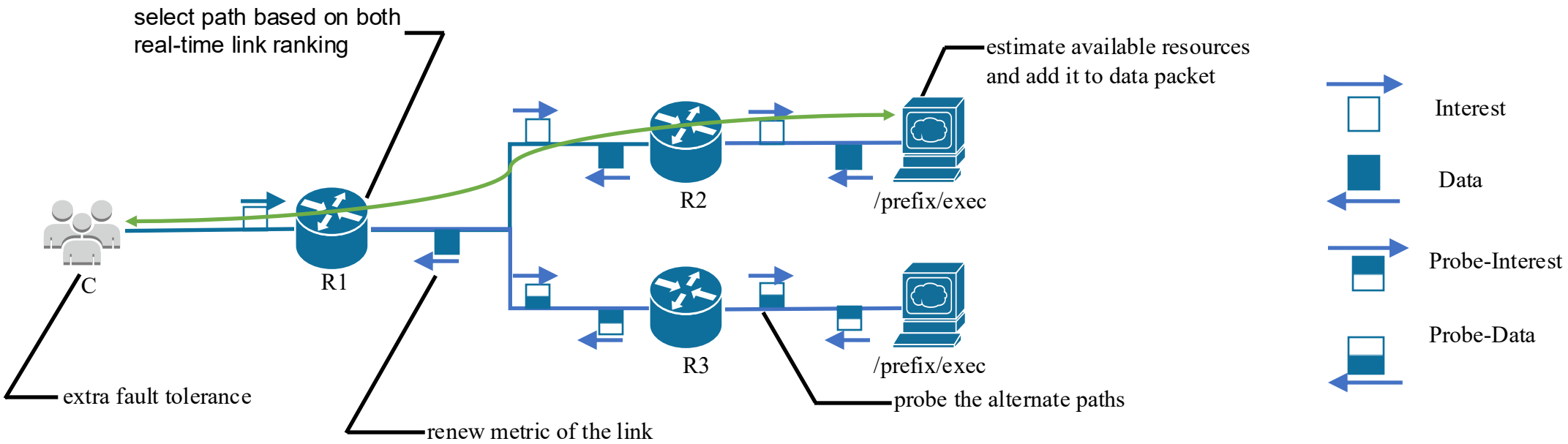
Design Goals

- Work for RPC frameworks.
- Improve efficiency.
- Use NDN features.
- Flexible.

OUR DESIGN

CAF Overview

- CAF dynamically make forwarding decisions based on real-time performance



Integrate Computation Info

- Redesign measurement table (MT)
  - Add computation Info into MT

Structure of new MT

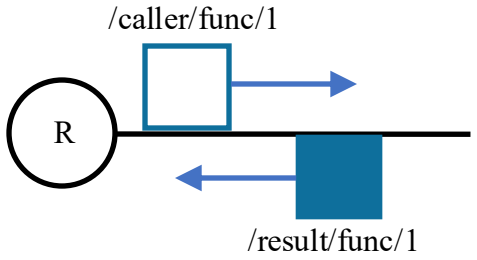
name	next hop	Network Delay	Producer Delay
/caller/func/a	P1	….	….
/caller/func/a	P2	….	….

- **Network Delay**: Round-trip time (affected by topology, bandwidth, and congestion)
- **Producer Delay**: Time for the producer to process the request (queuing and execution latency)

Local Measurement VS Collaborative Measurement

Local Measurement Scheme

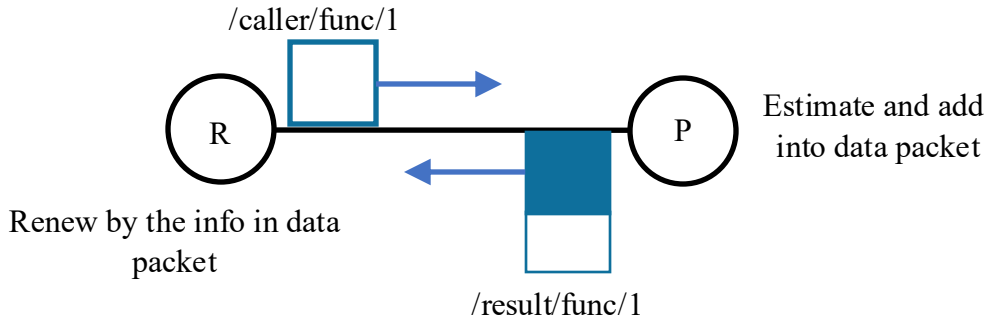
- Router locally records metric based On Interest-Data exchange



Measured Cost = $t_2 - t_1$

Collaborative Measurement Scheme

- Producer estimate metric
- Router renew based on info in Data



- Decoupled forwarding & computation → direct metrics are hard
- Persistent probing needed → slow, inefficient updates
- RTT reflects *past* queue, not current task state

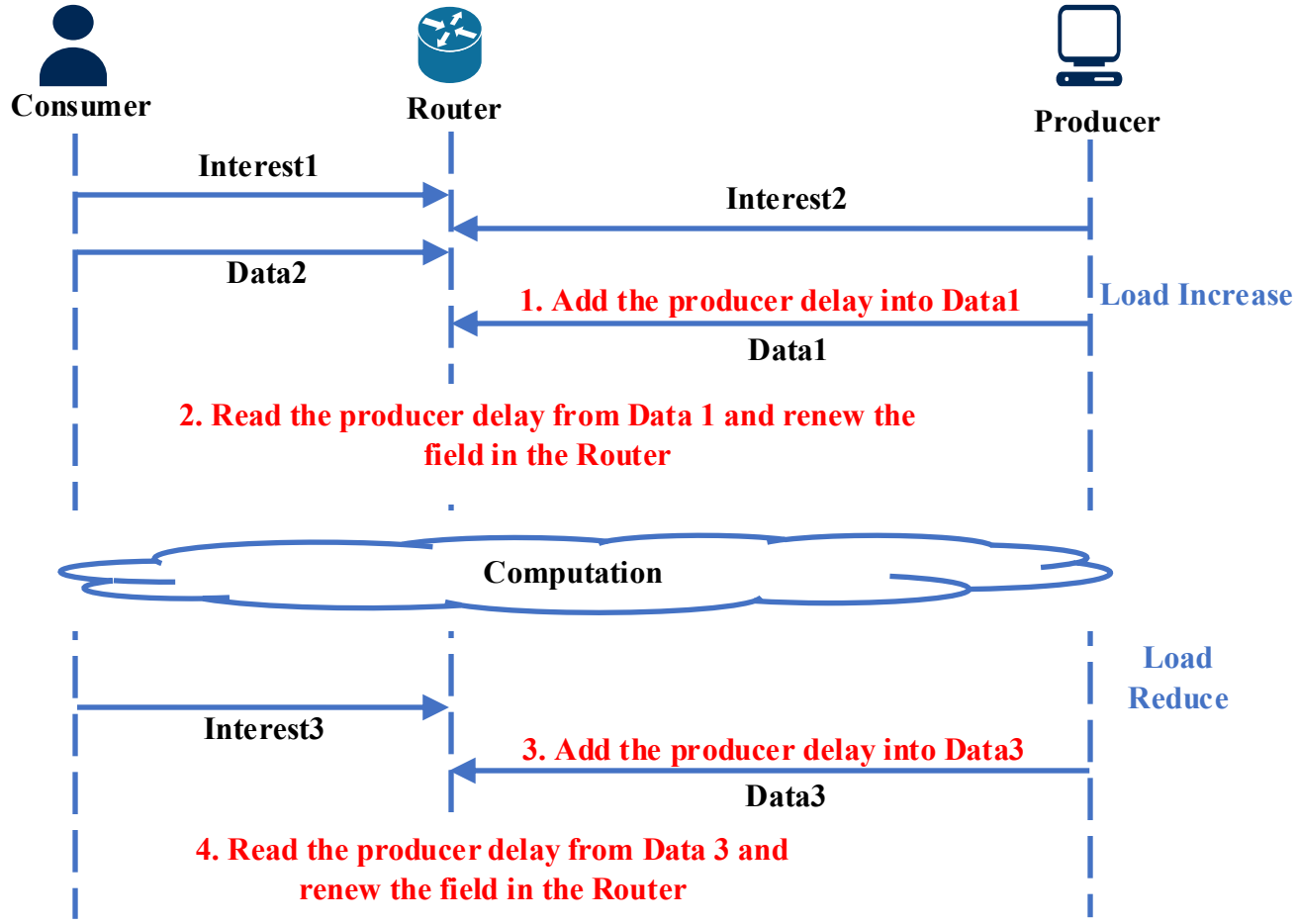
- Fits naturally into RPC workflows
- No protocol changes required
- Provides real-time load insights from the producer

Collaborative Measurement Scheme

Collaborative Measurement Sampling

- **Two key sampling points**:
  (a) After handshake between consumer and producer
  (b) When computation completes and result is returned

- **Purpose**:
  Ensures downstream nodes quickly learn updated producer load states



Mechanism to ensure robustness

- A redesigned path probe scheme

Problems

- producers **cannot distinguish** user vs. probe Interests
- **Redundant computation**

Solutions

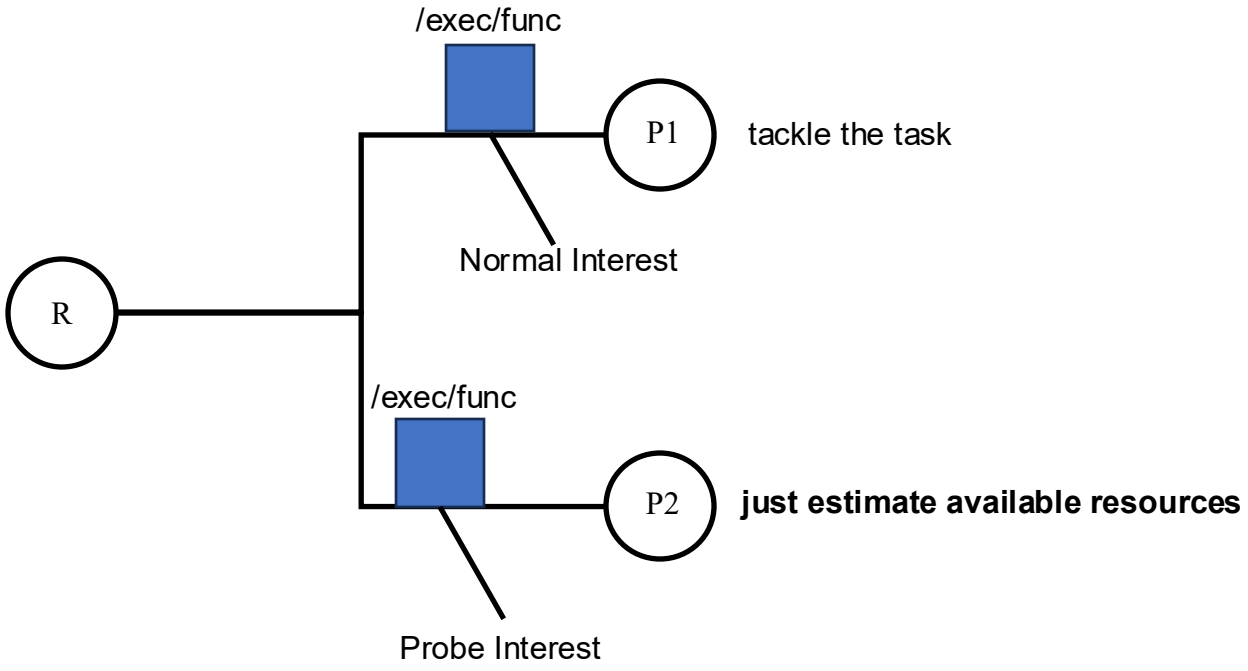
- Use **ApplicationParameters** to mark probe Interests
- Producer skips handshake & returns **resource status only**

Probe Interest Pakcet

Content Name	/caller/function/1
Selector	……
Nonce	……
Application Parameters	Type: probe

Probe Data Pakcet

Content Name	/result/function/1
Signature	……
SignatureInfo	……
Content	$Time_{estd} = \cdots$

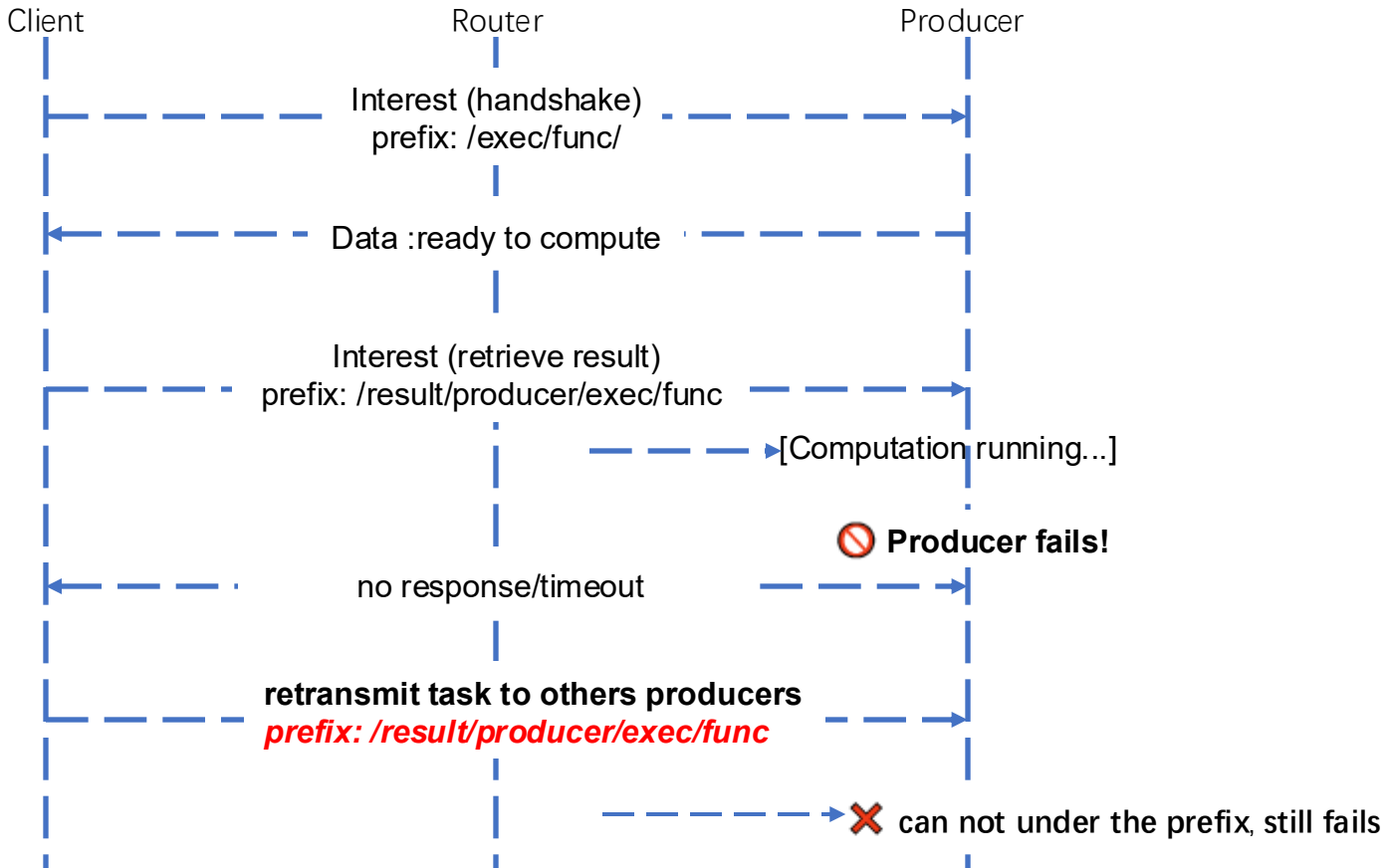


Mechanism to ensure robustness

- A extra fault tolerance mechanism

  Why need this?

  Task can't be reassigned if producer fails during execution
  - Prefix is different in handshake & retrieval
  - Prefix relation unmapped within a session



Mechanism to ensure robustness

- A extra fault tolerance mechanism

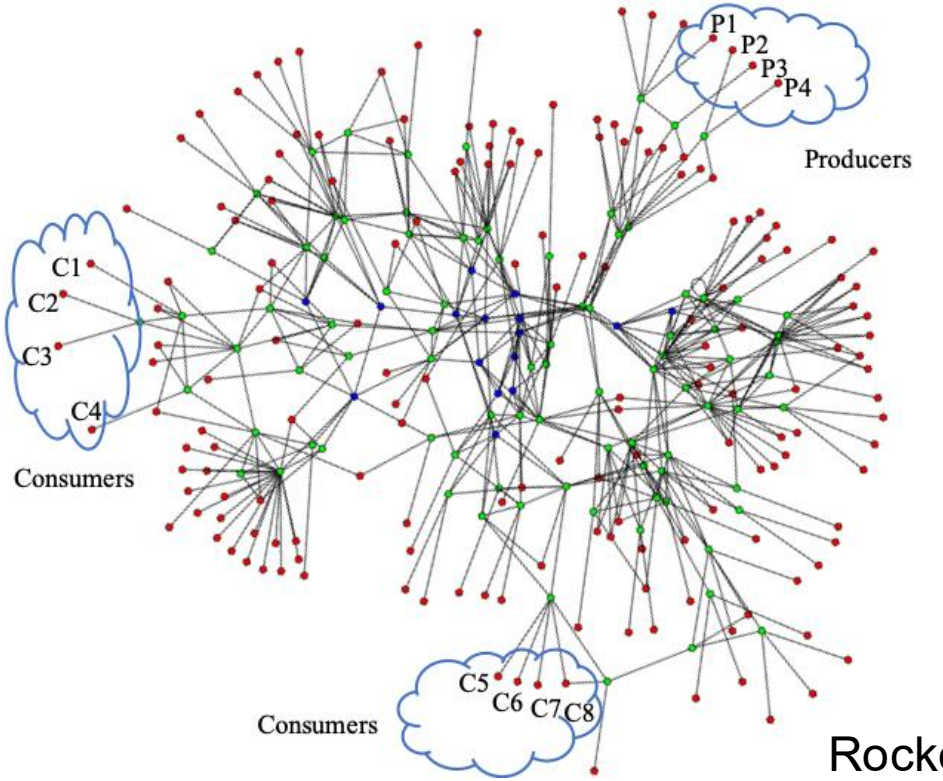
Solutions:

- Store handshake name after initial negotiation
- Track failed result retrievals with a counter
- If retries exceed threshold, re-initiate handshake
- Task is reassigned to a new available producer

EVALUATION

Evaluation details

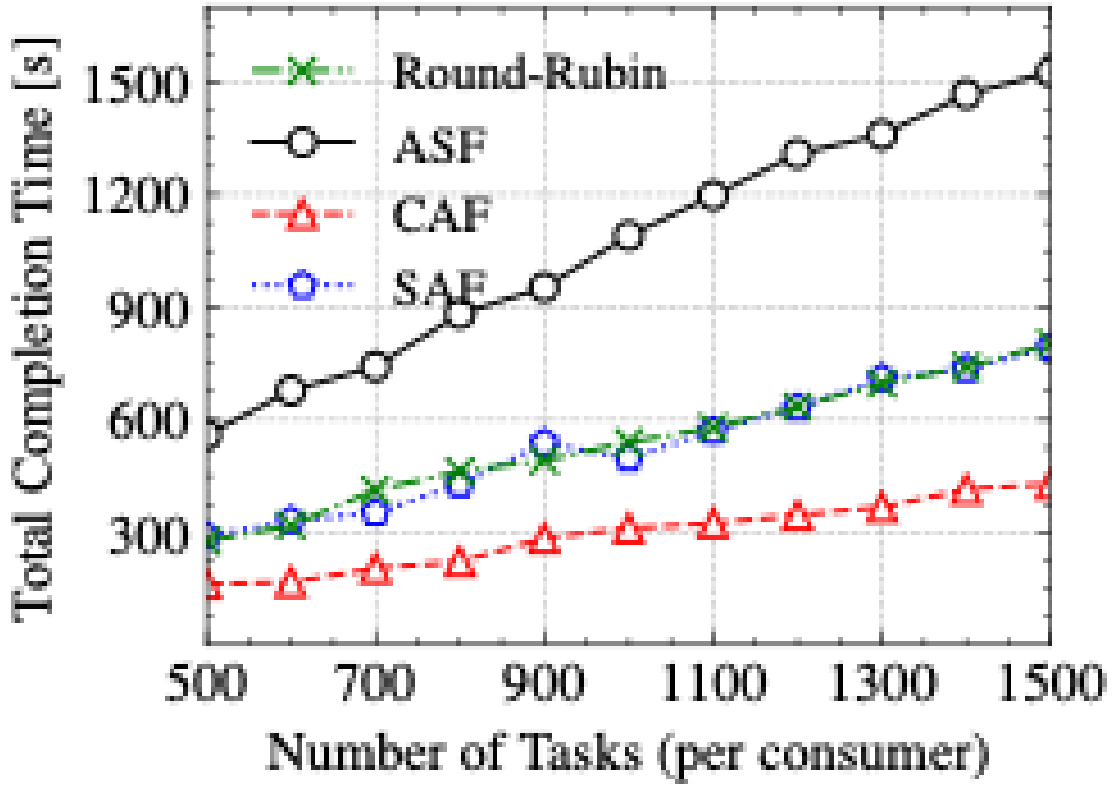
- Use real-world topology
- Use applications based on RPC frameworks
- Randomly send the Interests and record the whole completed time



Rocketfuel-1225

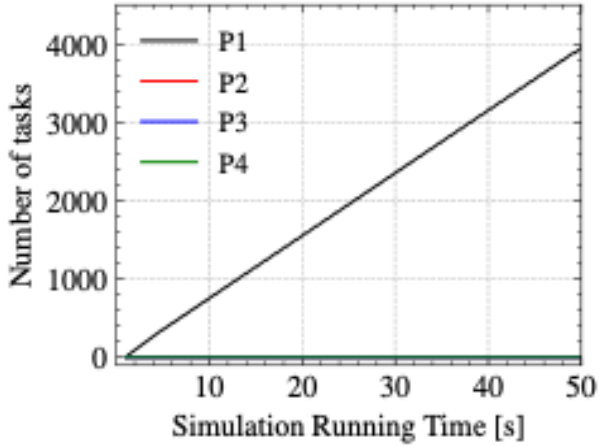
Metric 1: Performance

- Simulate remote computation with varying task loads and use completion time to evaluate strategy performance

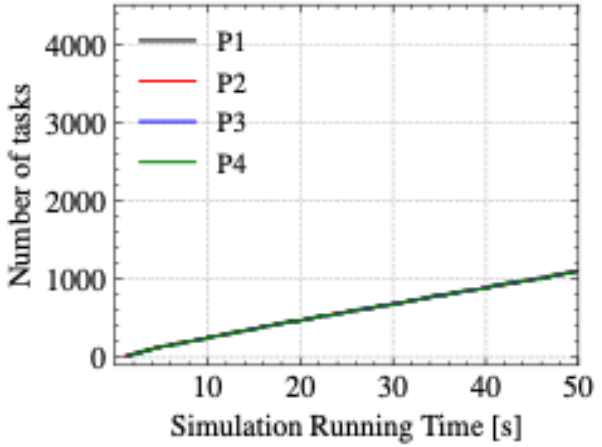


Metric 2: Server Load

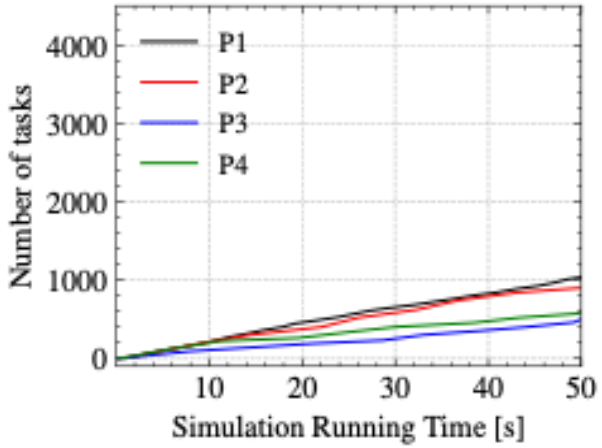
- Assign 4,000 tasks (500 per consumer) record the number of producer-side load



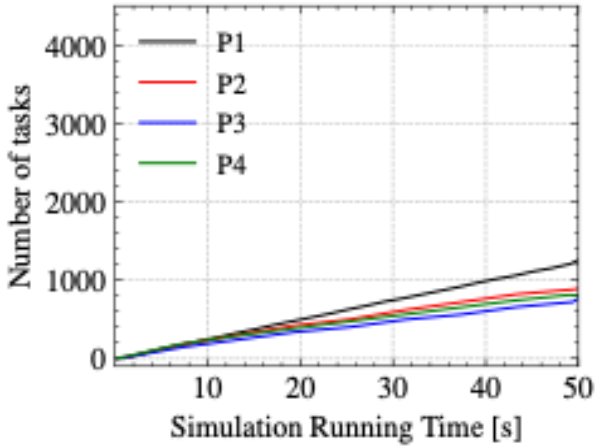
ASF



CAF



SAF



Round-Rubin

SUMMARY

Summary

- **Limitation of current adaptive forwarding designs in Remote Computation:**
  - How to integrate the producer's state into forwarding plane and consider the state while making forwarding decisions

- **Key ideas:**
  - Integrate the state into measurement table
  - Collaborative measurement scheme

Thanks for your listening!

Q/A