Fault Tolerant Service Function Chaining

M. GHAZNAVI, E. JALALPOUR, B. WONG, A. MASHTIZADEH, R. BOUTABA

UNIVERSITY OF WATERLOO

Fault Tolerant Service Function Chaining

Keep a service function chain running after **f ≥ 1** number of its service functions fail

Extend IETF network service header (NSH) to support fault tolerance

Introduction
Fault Tolerant Chaining
NSH for Fault Tolerant Chaining
Conclusion

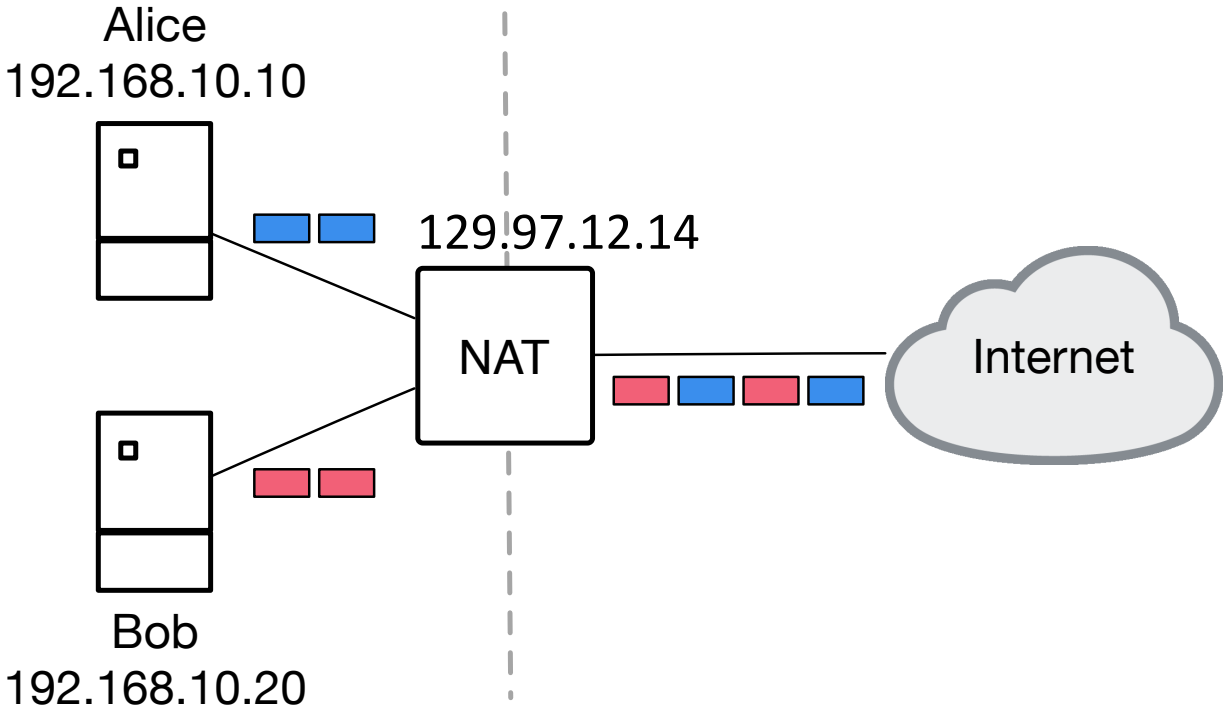
→ **Introduction**

Fault Tolerant Chaining

NSH for Fault Tolerant Chaining

Conclusion

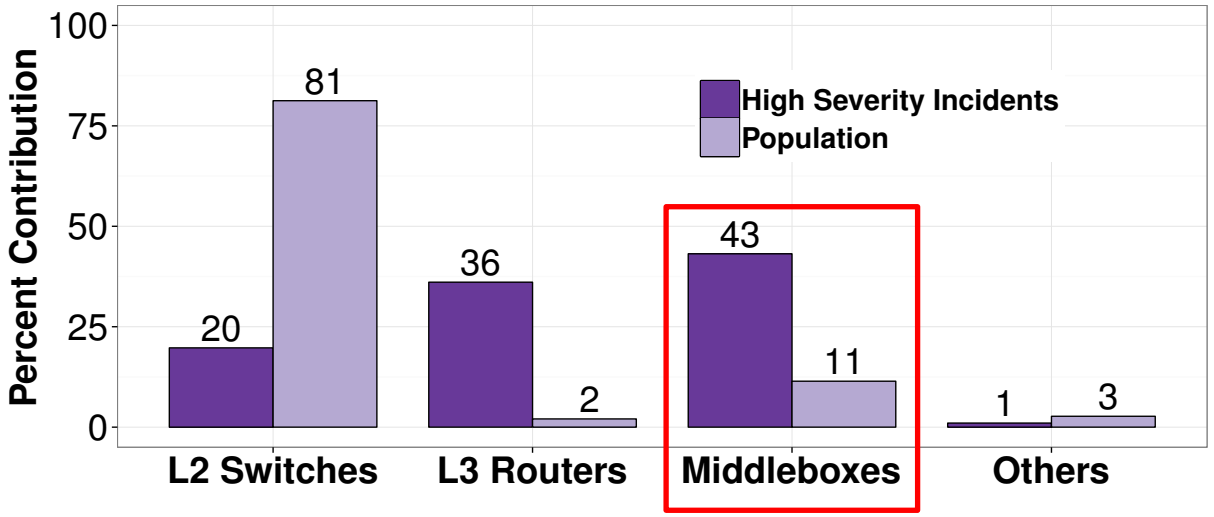
Service Functions (Middleboxes)



Network Address Translator (NAT)

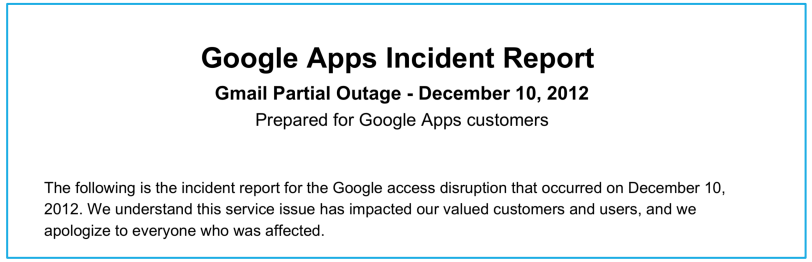
Firewall

Service Function Failures



Contributing to **43%** of high- severity incidents

"Demystifying the dark side of the middle: a field study of middlebox failures in datacenters." IMC 2013.

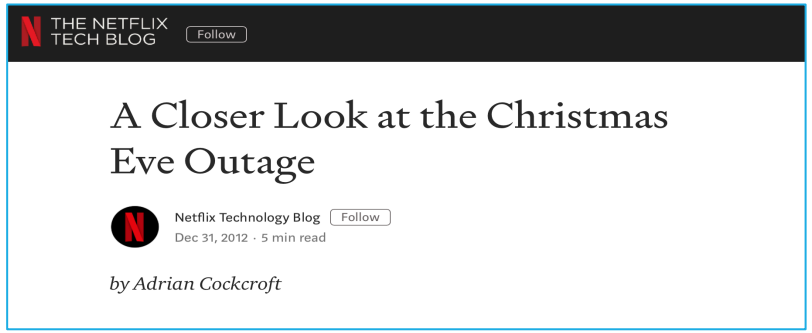


Google Apps Incident Report

Gmail Partial Outage - December 10, 2012

Prepared for Google Apps customers

The following is the incident report for the Google access disruption that occurred on December 10, 2012. We understand this service issue has impacted our valued customers and users, and we apologize to everyone who was affected.



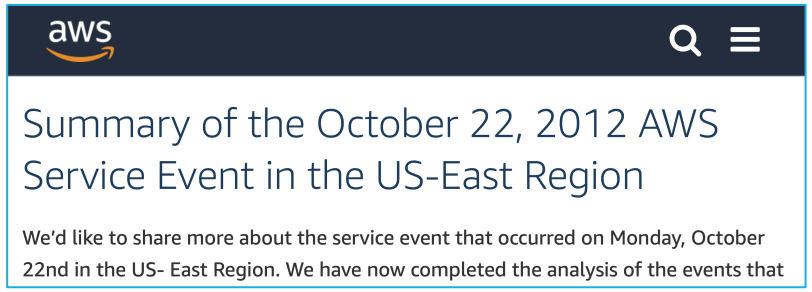
THE NETFLIX TECH BLOG Follow

A Closer Look at the Christmas Eve Outage

Netflix Technology Blog Follow

Dec 31, 2012 · 5 min read

by Adrian Cockcroft



aws

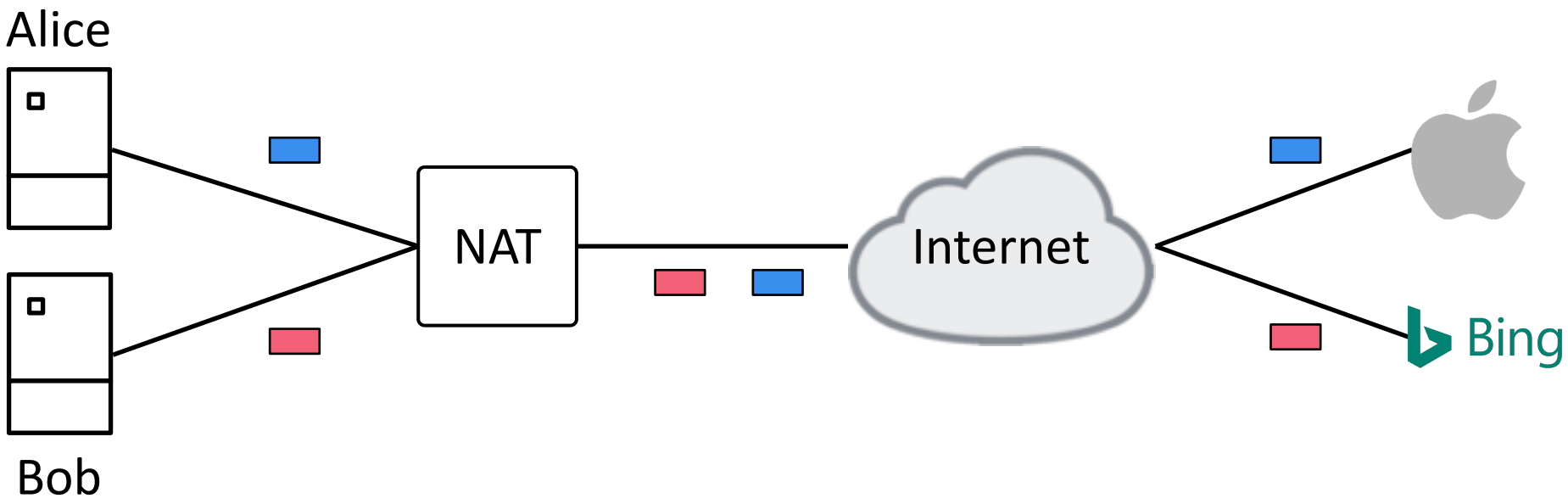
Summary of the October 22, 2012 AWS Service Event in the US-East Region

We'd like to share more about the service event that occurred on Monday, October 22nd in the US- East Region. We have now completed the analysis of the events that

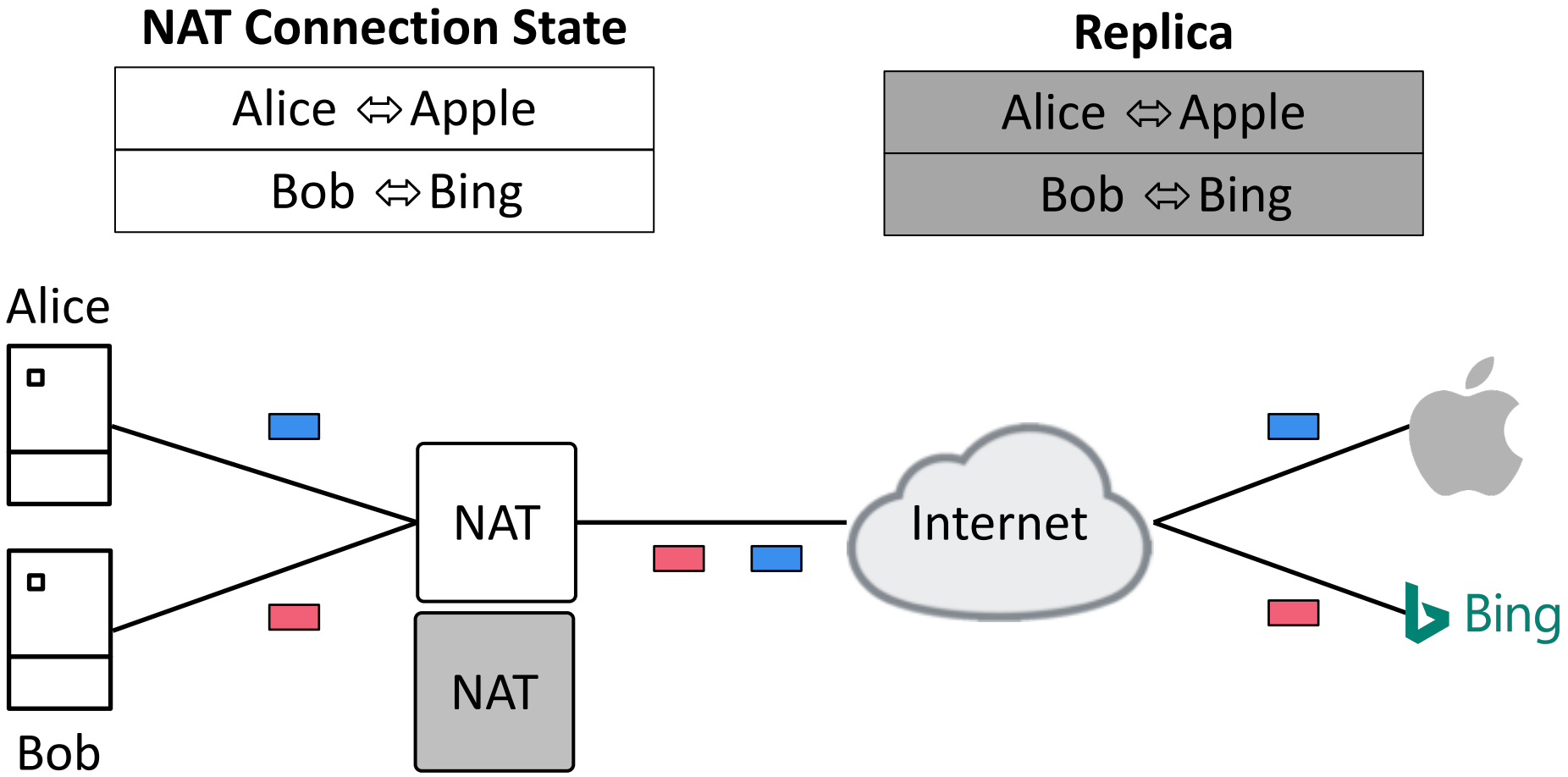
Service Function Fault Tolerance

NAT Connection State

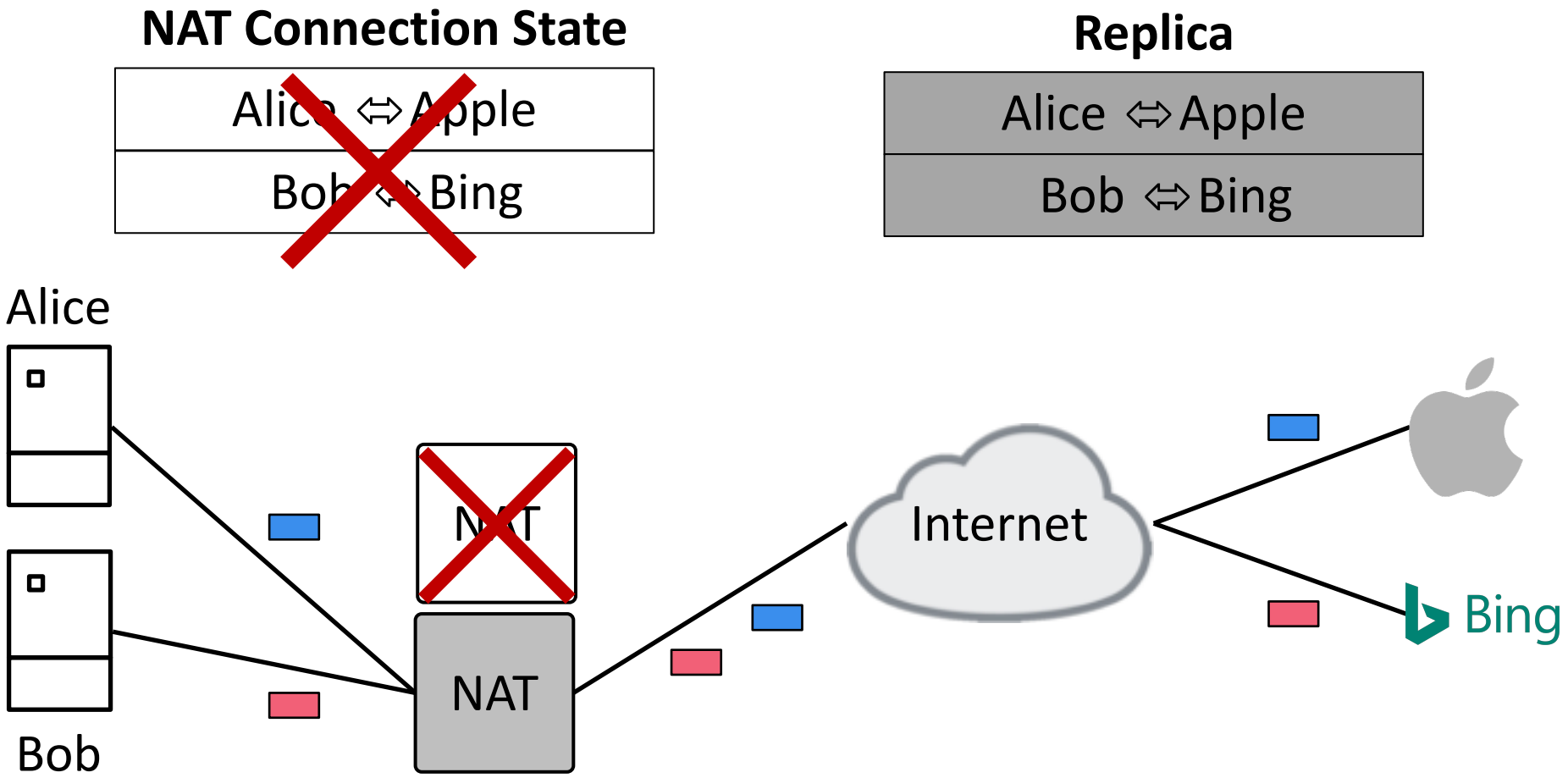
Alice ⇔ Apple
Bob ⇔ Bing



Service Function Fault Tolerance



Service Function Fault Tolerance

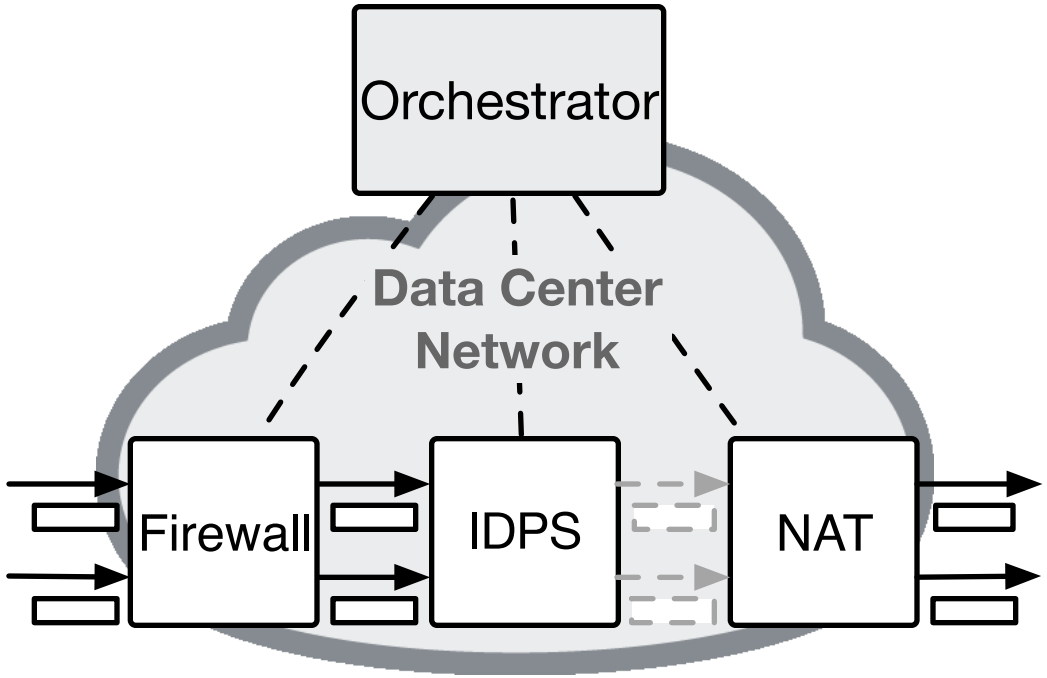


Service Function Fault Tolerance – Cont.

Most of existing solutions are snapshot based

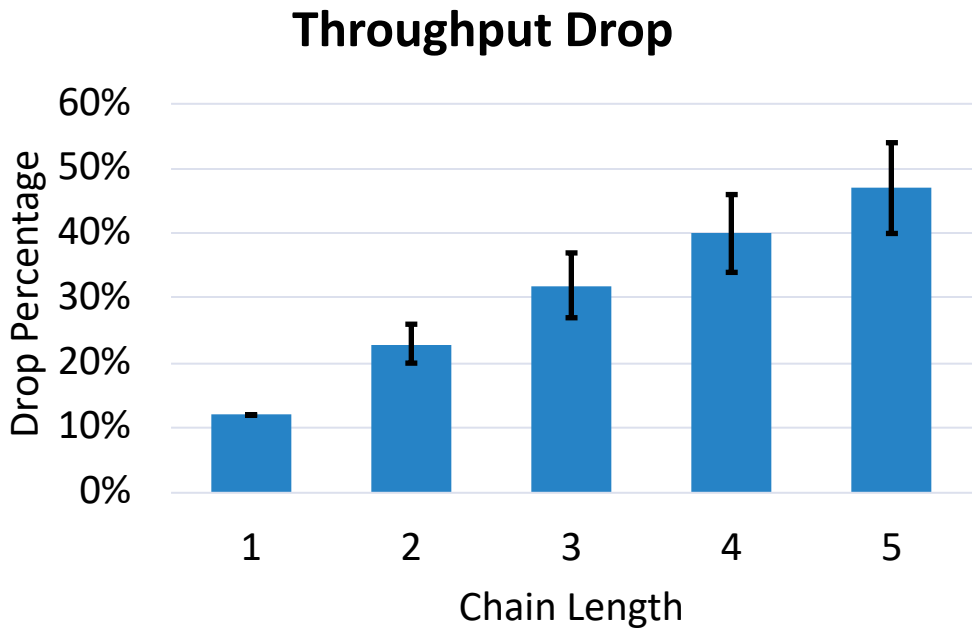
- **Pico Replication**, SoCC 2013
- **FTMB**, SIGCOMM 2015
- **REINFORCE**, CoNEXT 2018

Service Function Chains (Chains)

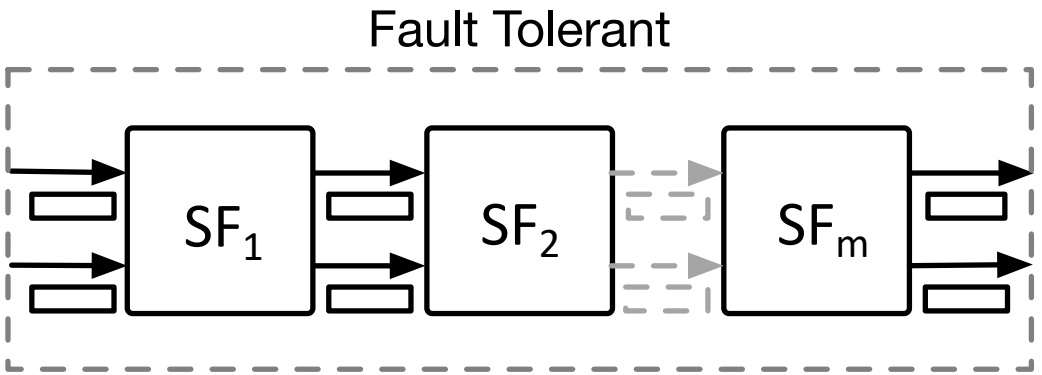


Fault Tolerance for a Chain

EXISTING SNAPSHOT-BASED APPROACHES



OUR APPROACH: FAULT TOLERANCE FOR AN ENTIRE CHAIN



Introduction

→ **Fault Tolerant Chaining**

NSH for Fault Tolerant Chaining

Conclusion

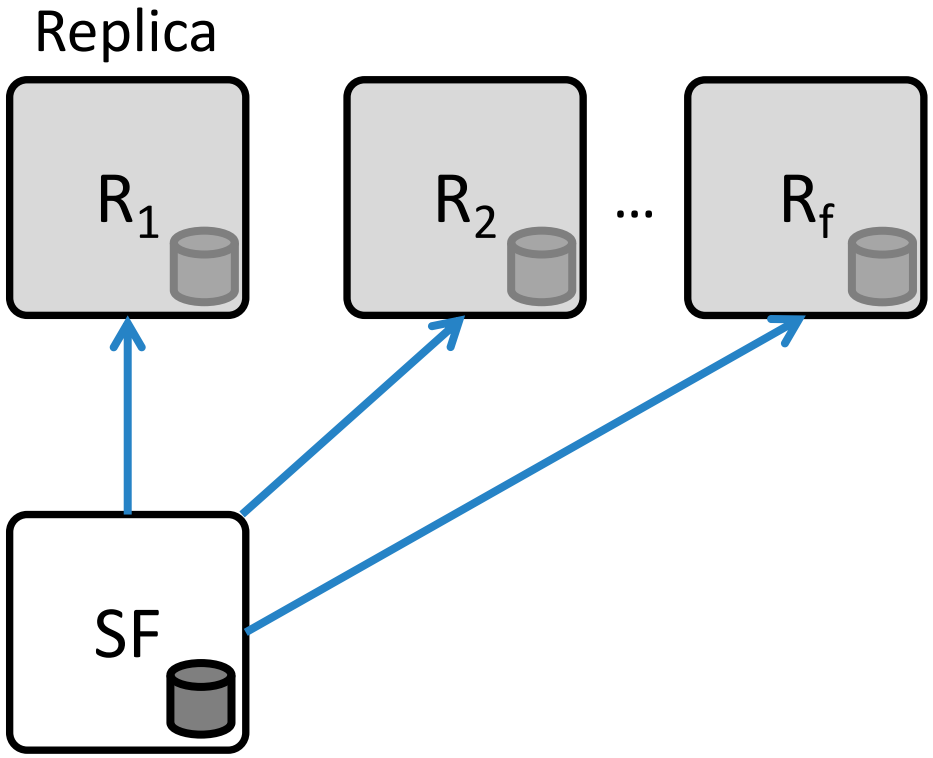
Design Choices

In-chain replication

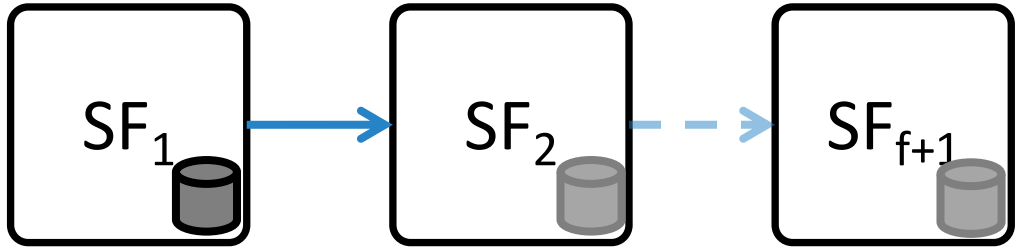
State piggybacking

Design Choices – In-Chain Replication

EXISTING APPROACHES

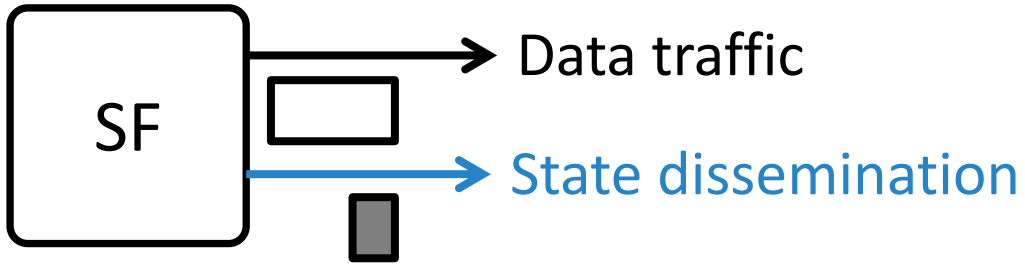


FTC'S APPROACH

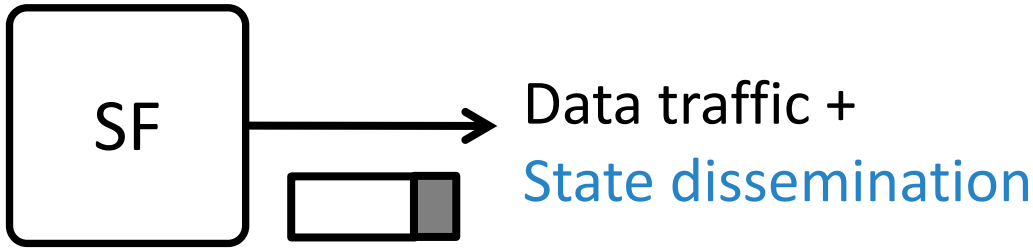


Design Choices – State Piggybacking

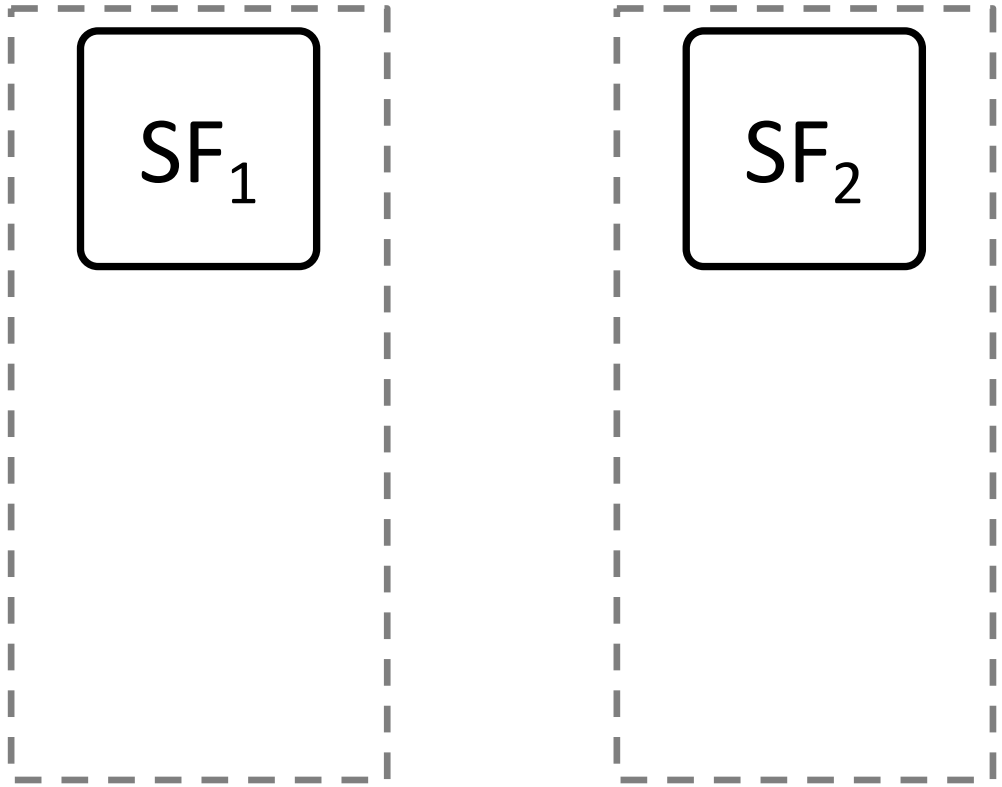
EXISTING APPROACHES



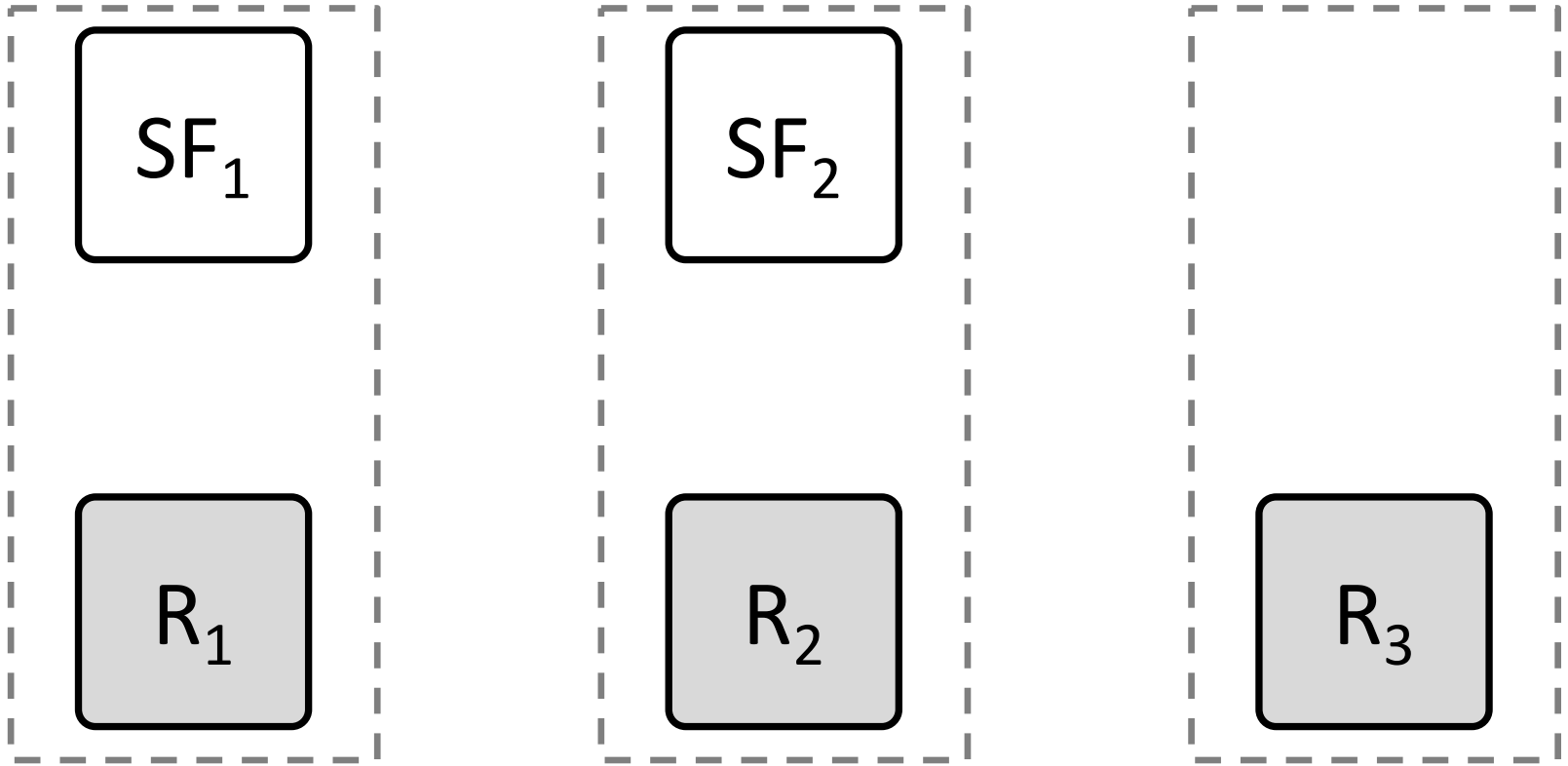
FTC'S APPROACH



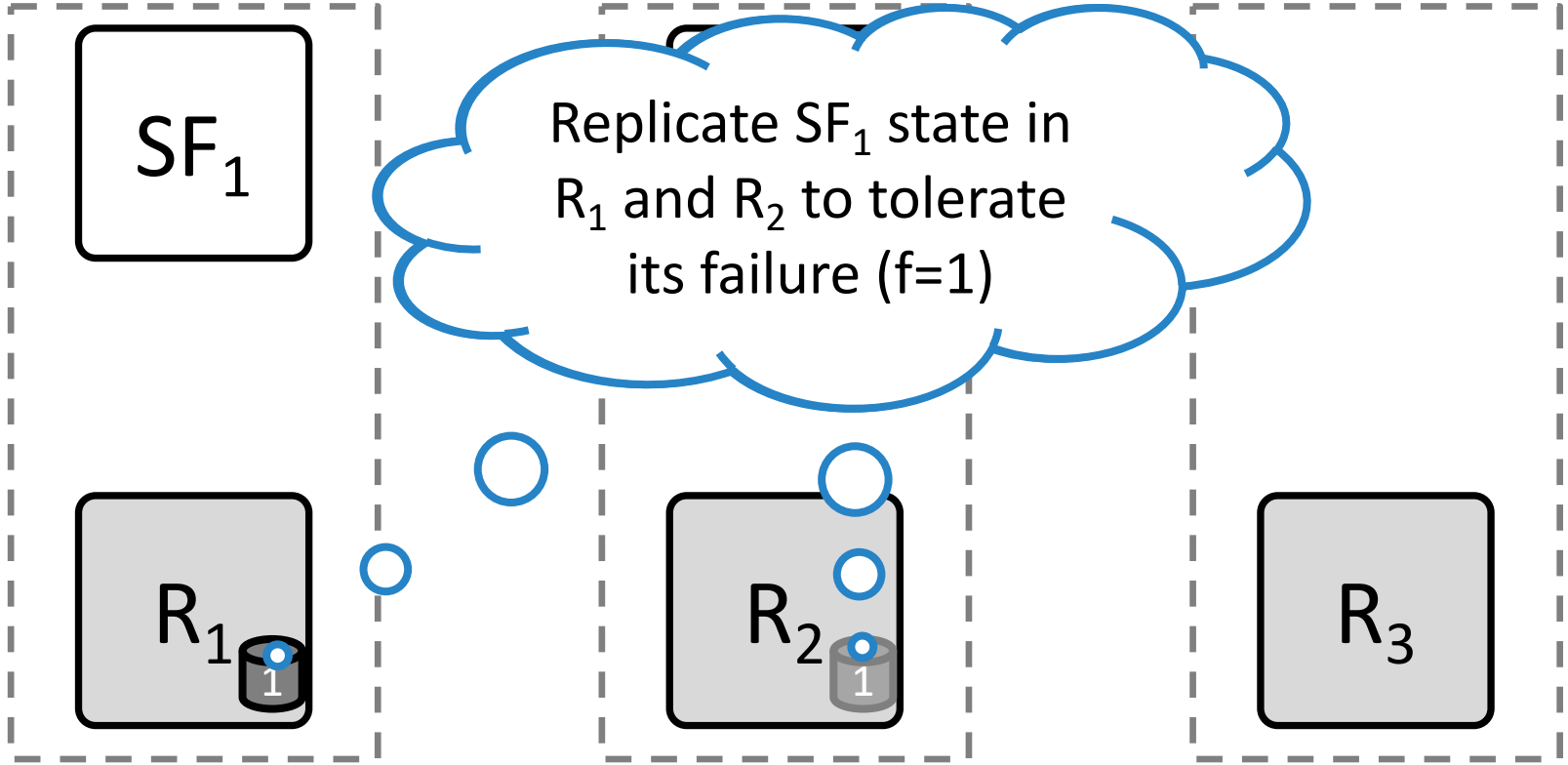
Fault Tolerant Chain Protocol



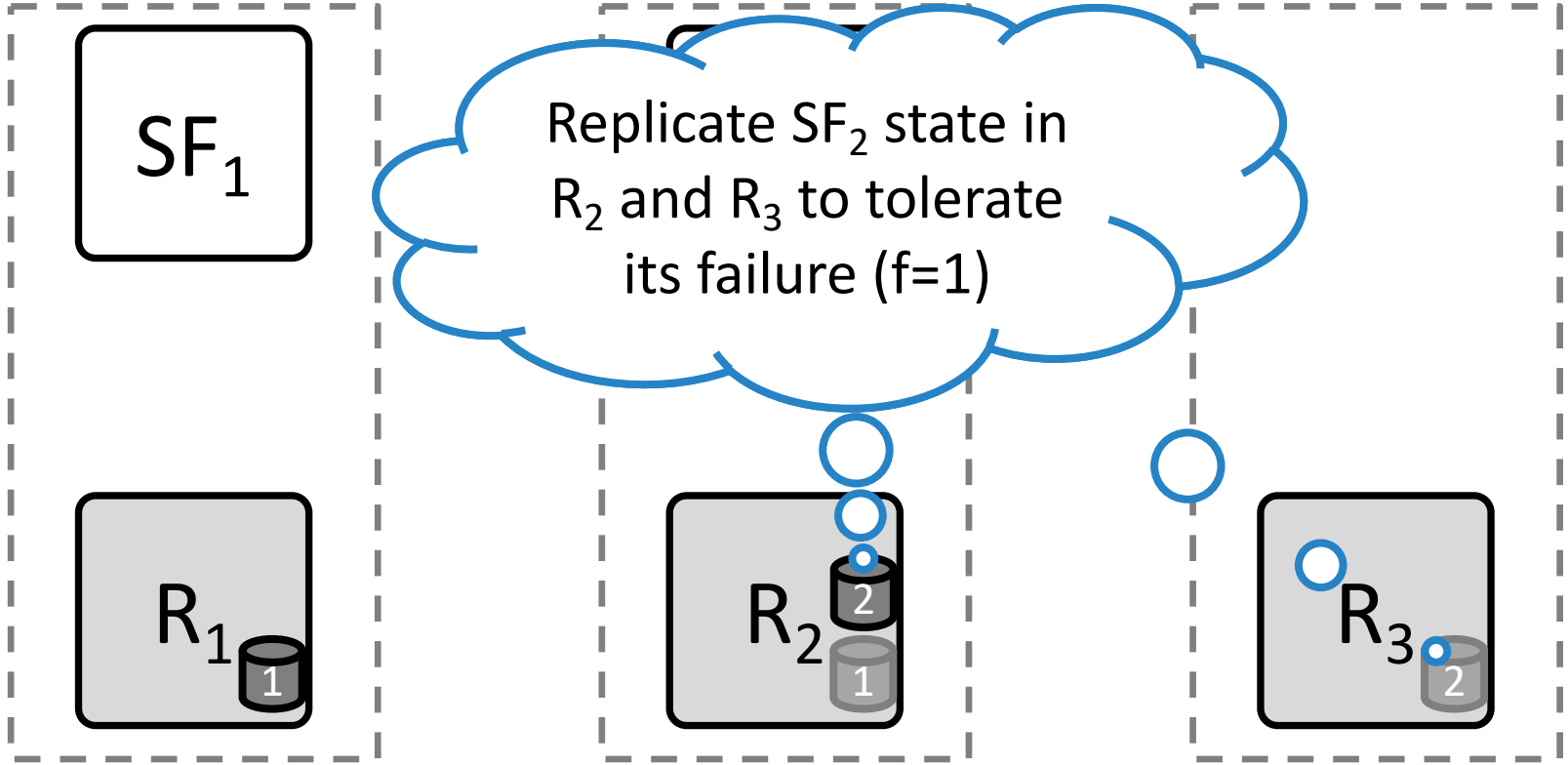
Fault Tolerant Chain Protocol



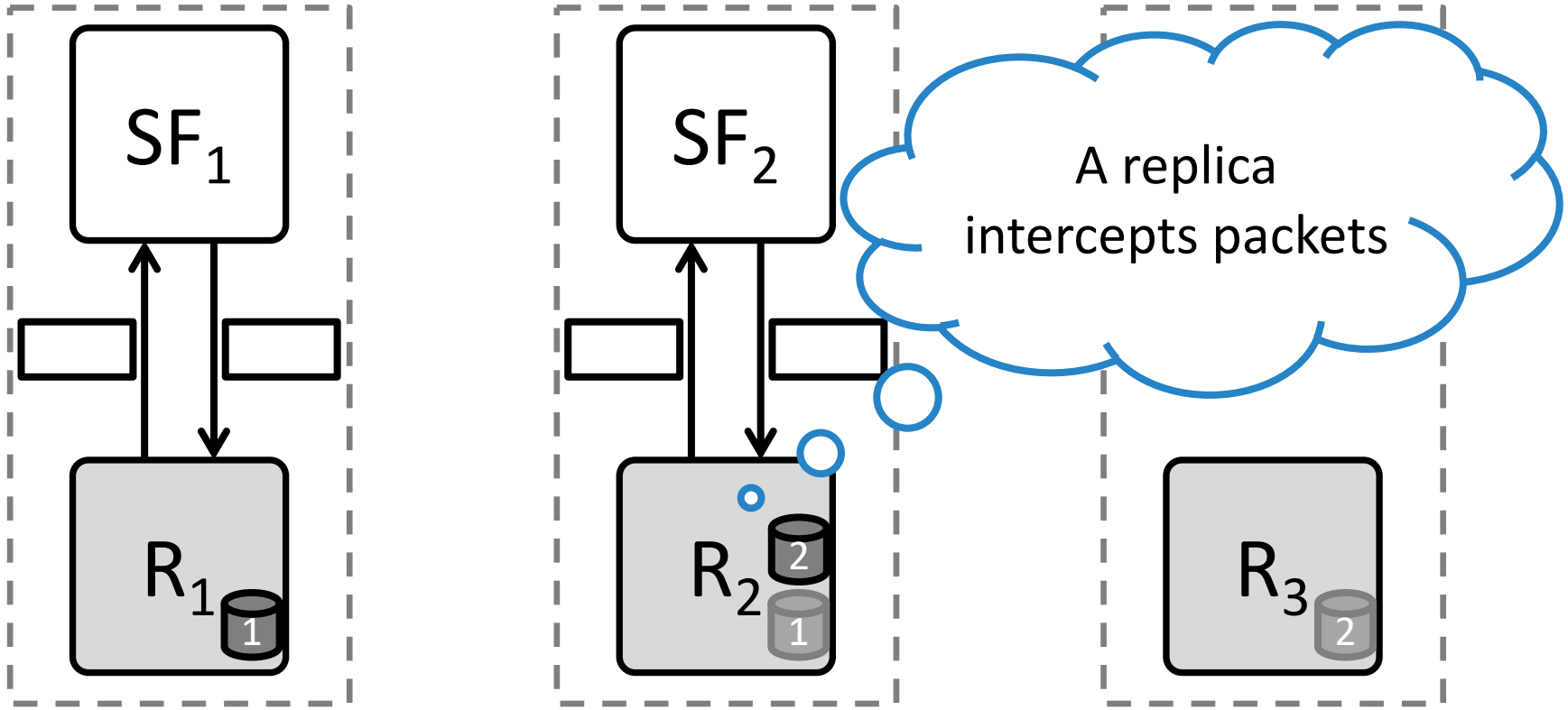
Fault Tolerant Chain Protocol



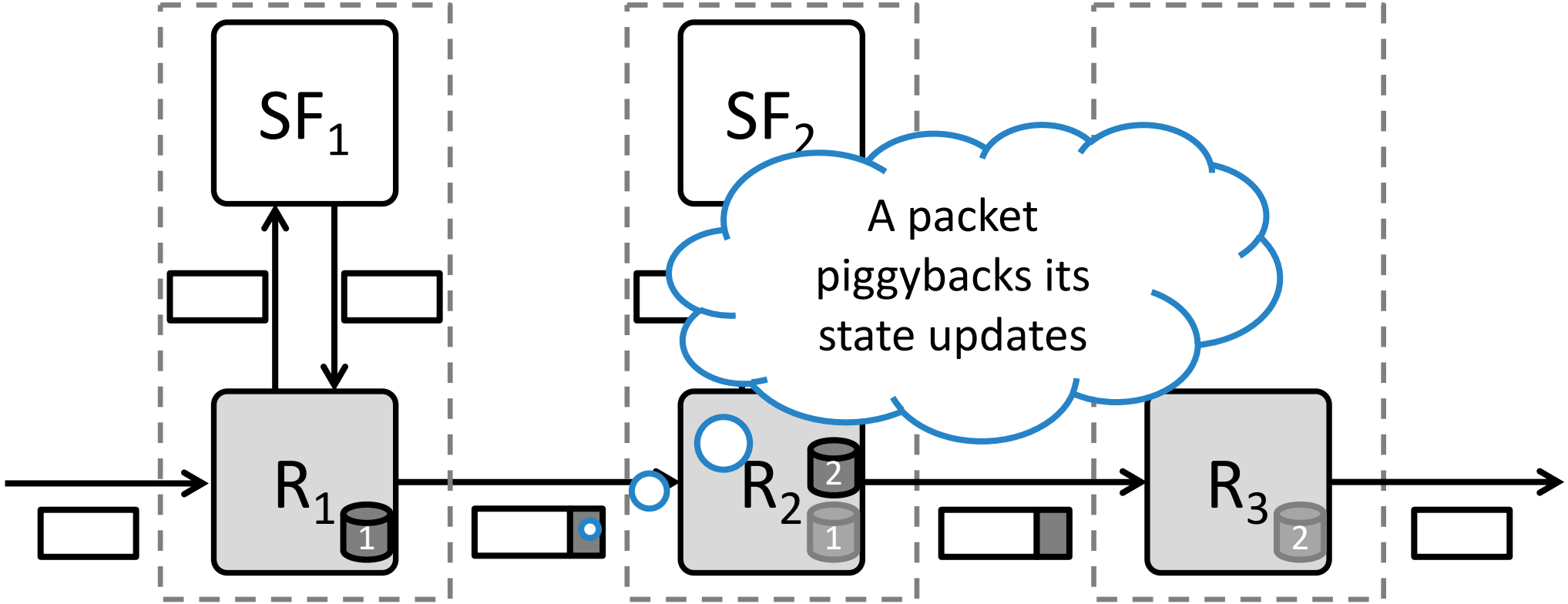
Fault Tolerant Chain Protocol



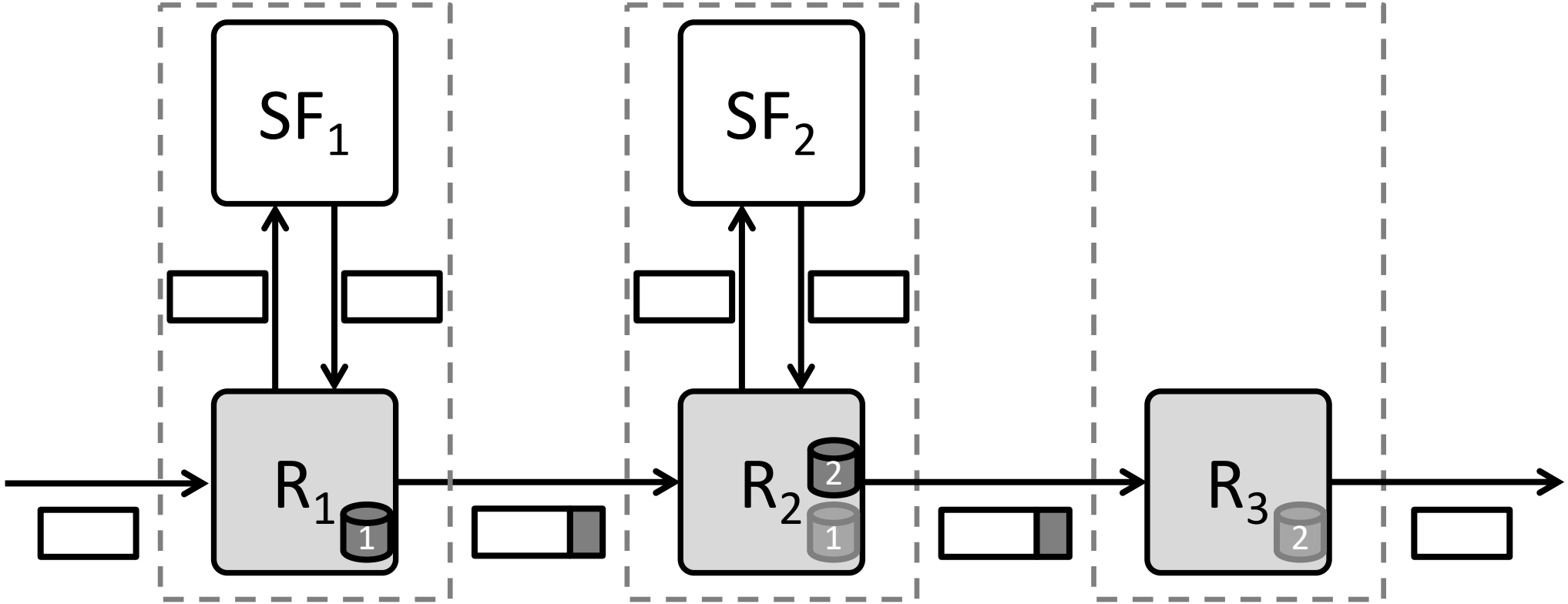
Fault Tolerant Chain Protocol



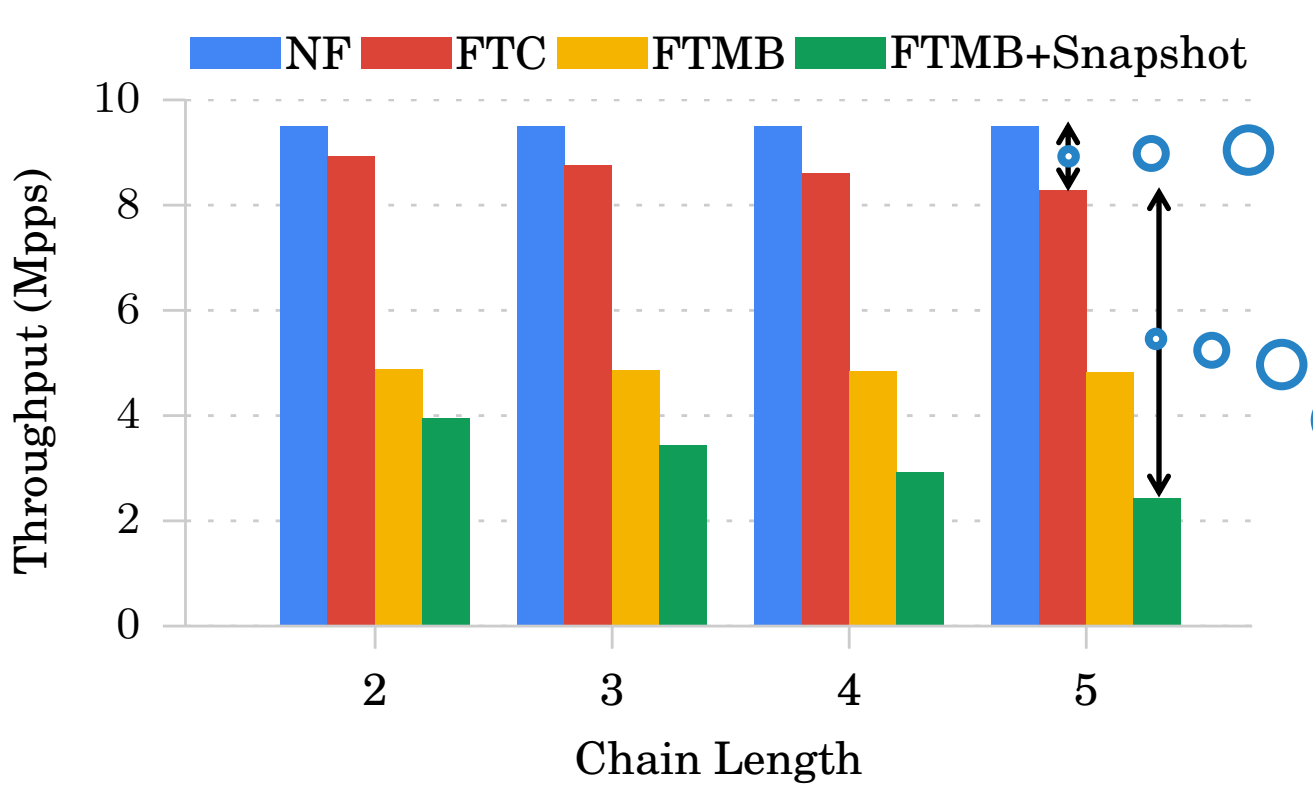
Fault Tolerant Chain Protocol



Fault Tolerant Chain Protocol



FTC's Performance



13 % less than optimal

3.5x higher throughput

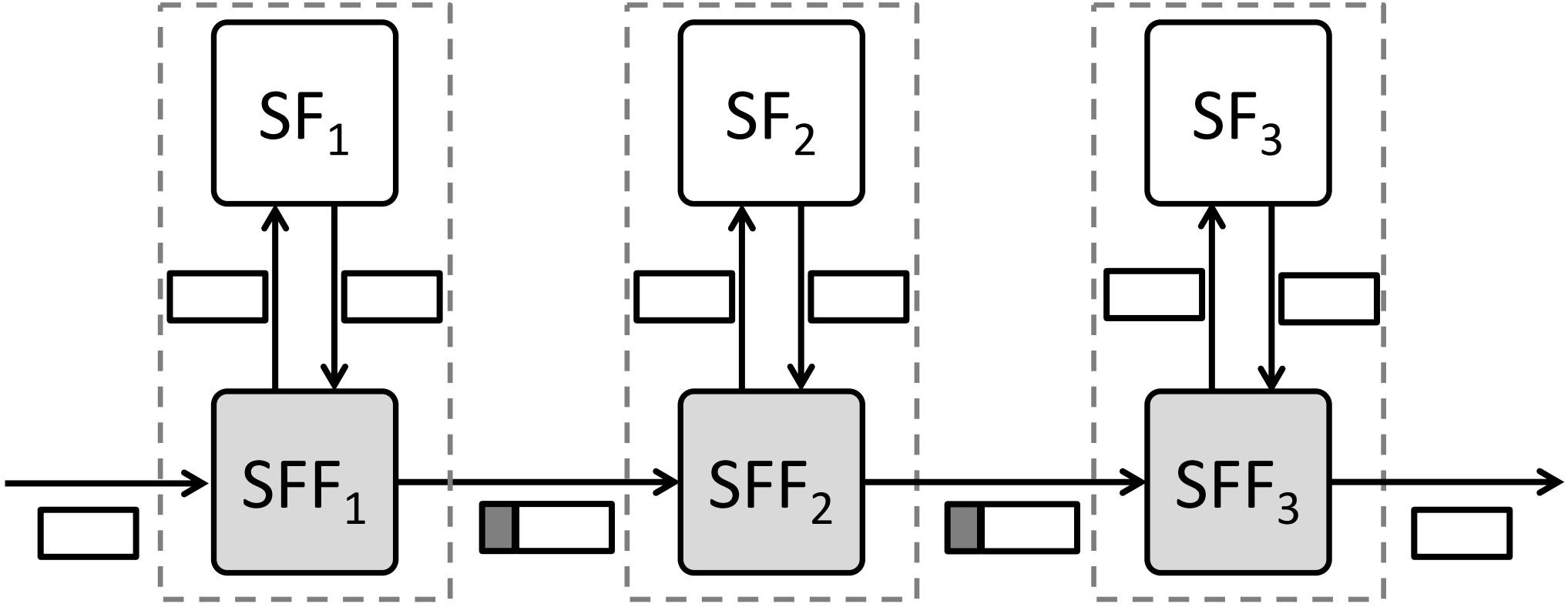
Introduction

Fault Tolerant Chaining

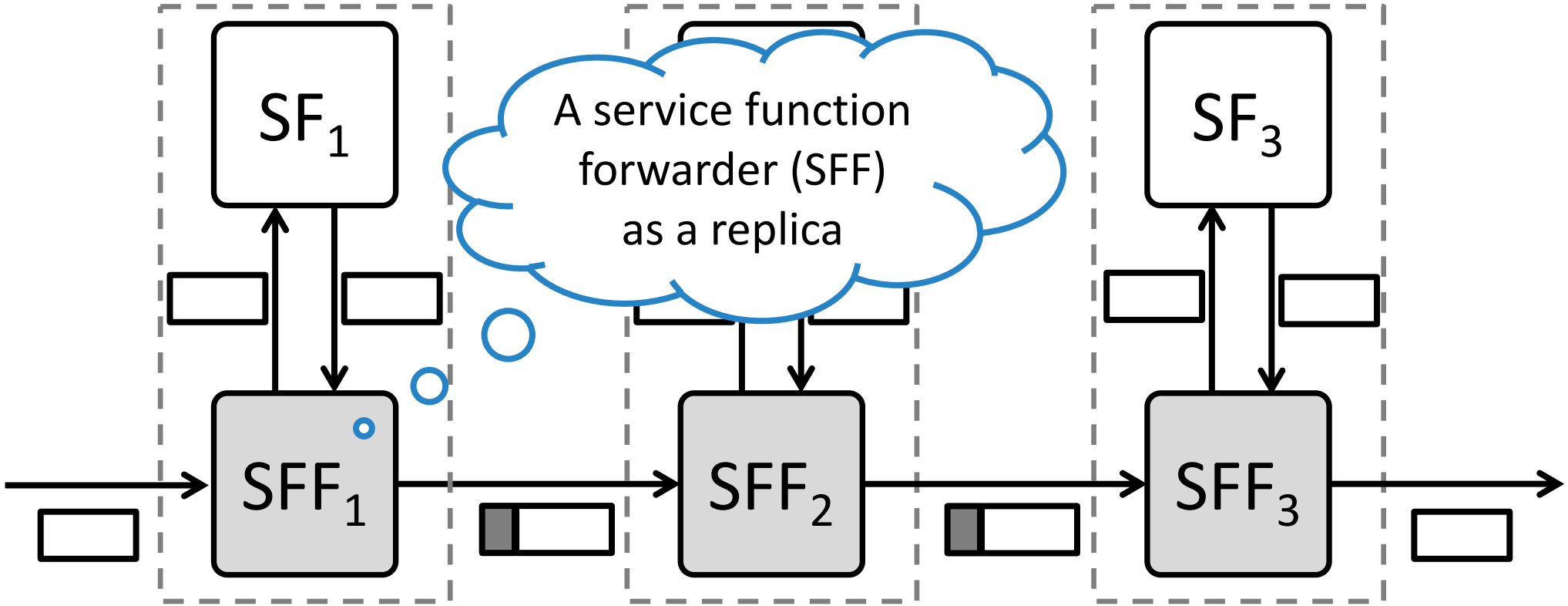
→ **NSH for Fault Tolerant Chaining**

Conclusion

Network Service Header – RFC 8300



Service Function Forwarder



Service Function Forwarder As a Replica

SUPPORTED BY ORIGINAL NSH

Packet forwarding through a chain

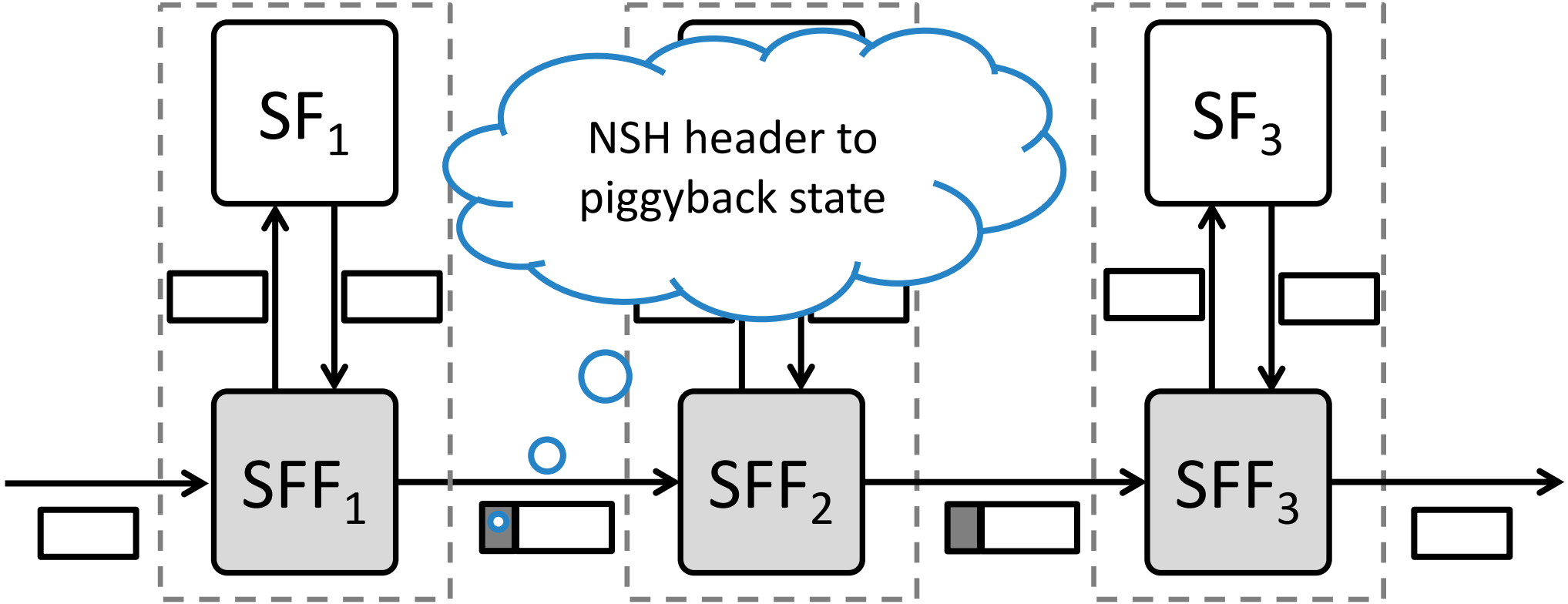
OUR CONTRIBUTIONS

Extensions to NSH

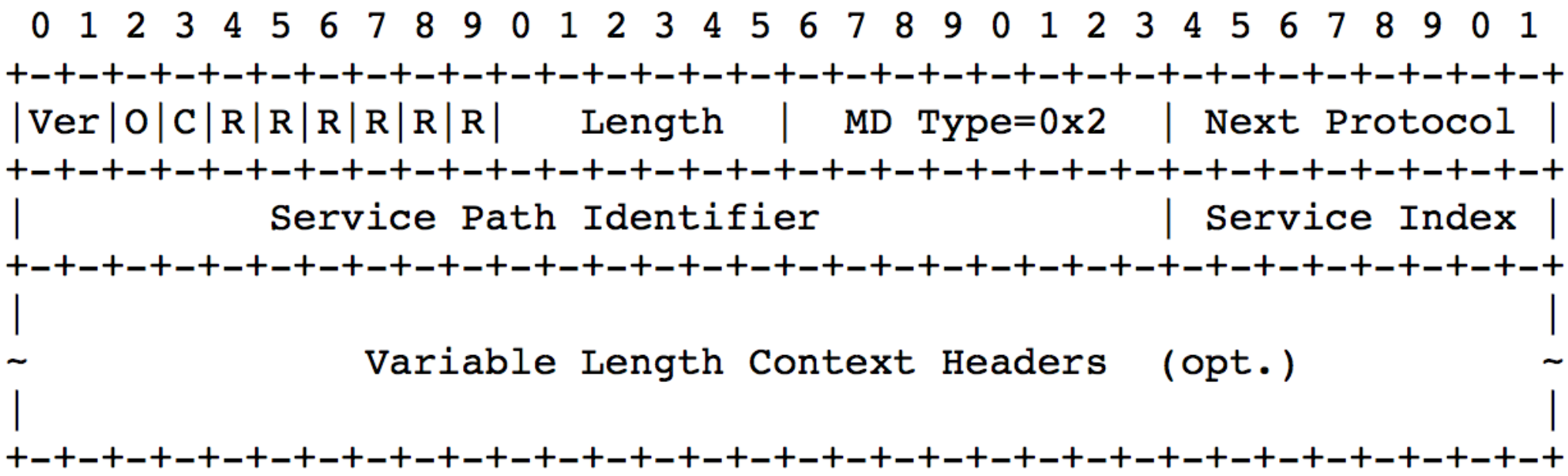
- State management API
- State replication

NSH support in Click modular router

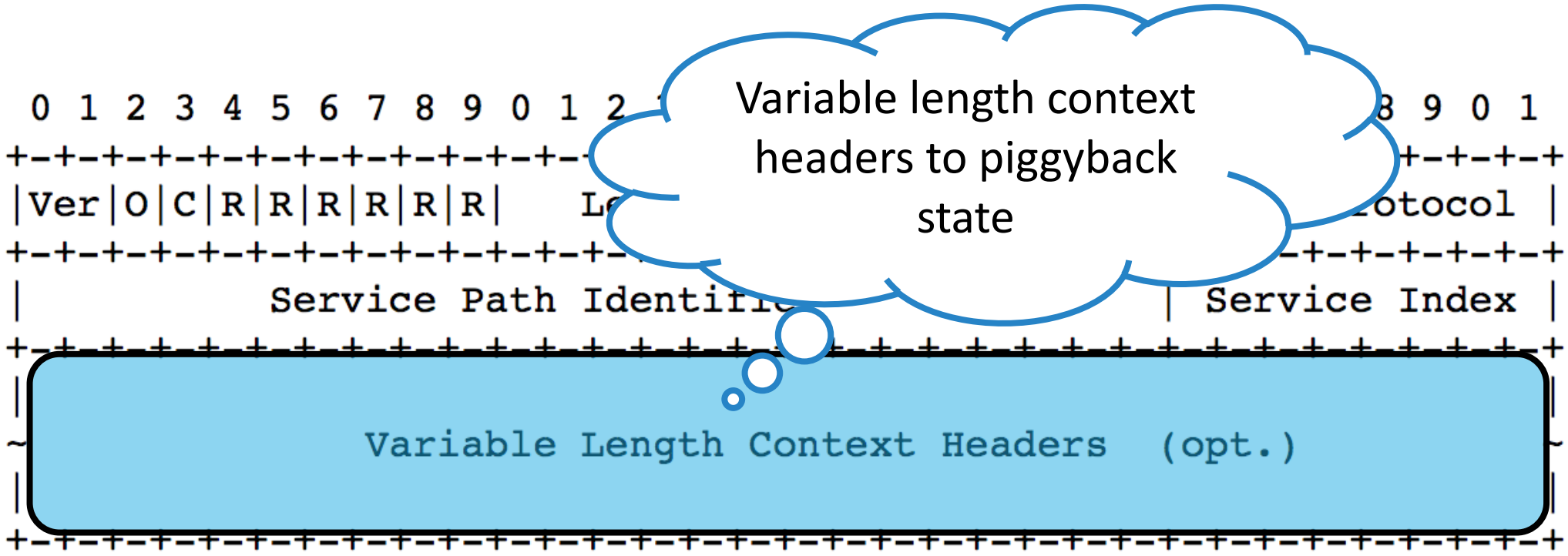
Network Service Header Format



Network Service Header Format



Network Service Header Format



Context Headers to Piggyback State

SUPPORTED BY NSH

Packet encapsulation

Variable length metadata

OUR CONTRIBUTIONS

Extensions to NSH

- State piggybacking using NSH metadata
- Defining datatypes

Introduction

Fault Tolerant Chaining

Evaluation

→ Conclusion

Summary

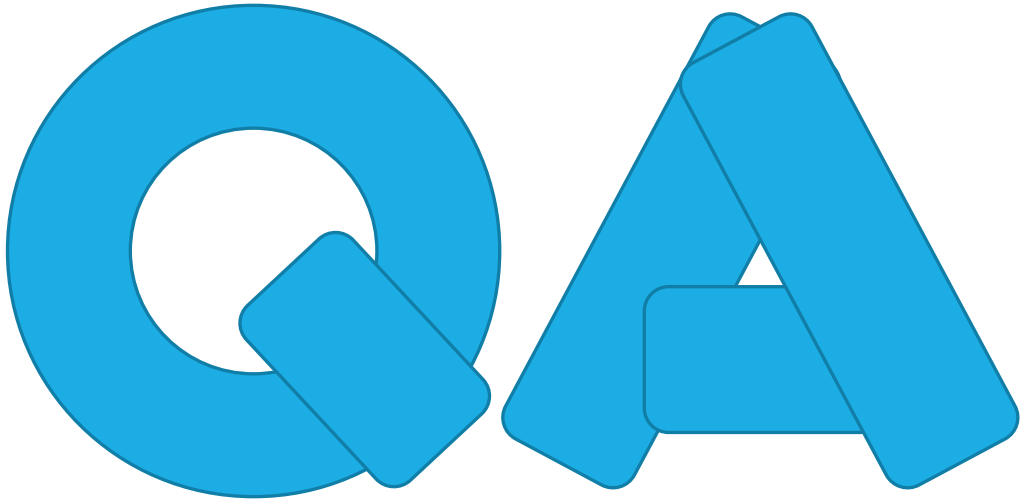
FTC keeps a service function chain running after **f ≥ 1** of its service functions fail

- State piggybacking
- In-chain replication

Extending NSH to provide fault tolerant service function chaining

Full paper available on arXiv: https://arxiv.org/pdf/2001.03321.pdf

- Transactional packet processing
- Failure recovery procedure


QA