Fault Tolerant Service Function Chaining

Keep a service function chain running after $f \geq 1$ 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

Alice
192.168.10.10

Bob
192.168.10.20

129.97.12.14
Service Function Failures

Contributing to 43% of high-severity incidents
Service Function Fault Tolerance

NAT Connection State

| Alice ↔ Apple  
| Bob ↔ Bing  

Diagram:
- Alice and Bob connected to NAT
- NAT connected to Internet
- Apple and Bing accessible through Internet
Service Function Fault Tolerance

NAT Connection State

<table>
<thead>
<tr>
<th>Alice ↔ Apple</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob ↔ Bing</td>
</tr>
</tbody>
</table>

Replica

<table>
<thead>
<tr>
<th>Alice ↔ Apple</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bob ↔ Bing</td>
</tr>
</tbody>
</table>

Alice

Bob

Internet

Apple

Bing
Service Function Fault Tolerance

NAT Connection State

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>Apple</td>
</tr>
<tr>
<td>Bob</td>
<td>Bing</td>
</tr>
</tbody>
</table>

Replica

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>Apple</td>
</tr>
<tr>
<td>Bob</td>
<td>Bing</td>
</tr>
</tbody>
</table>

Alice

Bob

NAT

Internet

Apple

Bing
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

Throughput Drop

<table>
<thead>
<tr>
<th>Chain Length</th>
<th>Drop Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>10%</td>
</tr>
<tr>
<td>2</td>
<td>20%</td>
</tr>
<tr>
<td>3</td>
<td>30%</td>
</tr>
<tr>
<td>4</td>
<td>40%</td>
</tr>
<tr>
<td>5</td>
<td>50%</td>
</tr>
</tbody>
</table>

Fault Tolerant

SF_1 \rightarrow SF_2 \rightarrow SF_m
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

Data traffic

SF

Data traffic +
State dissemination

State dissemination
Fault Tolerant Chain Protocol

$SF_1$

$SF_2$
Fault Tolerant Chain Protocol

SF_1
R_1

SF_2
R_2

R_3
Fault Tolerant Chain Protocol

Replicate SF₁ state in R₁ and R₂ to tolerate its failure (f=1)
Fault Tolerant Chain Protocol

Replicate SF$_2$ state in R$_2$ and R$_3$ to tolerate its failure (f=1)
Fault Tolerant Chain Protocol

A replica intercepts packets
Fault Tolerant Chain Protocol

A packet piggybacks its state updates
Fault Tolerant Chain Protocol
FTC’s Performance

![Graph showing performance comparison]

- **Throughput (Mpps)**
  - Chain Length: 2, 3, 4, 5
  - NF, FTC, FTMB, FTMB+Snapshot

**Key Observations**
- 13% less than optimal throughput
- 3.5x higher throughput
Introduction
Fault Tolerant Chaining
→ NSH for Fault Tolerant Chaining
Conclusion
Network Service Header – RFC 8300
Service Function Forwarder

A service function forwarder (SFF) as a replica
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

NSH header to piggyback state
Network Service Header Format

```
0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|Ver|O|C|R|R|R|R|R|R| Length | MD Type=0x2 | Next Protocol |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| Service Path Identifier | Service Index |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
| ~ Variable Length Context Headers (opt.) ~ |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```
Network Service Header Format

Variable length context headers to piggyback state

Variable Length Context Headers (opt.)
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 *\( f \geq 1 \)* of its service functions fail

- State piggybacking
- In-chain replication

Extending NSH to provide fault tolerant service function chaining


- Transactional packet processing
- Failure recovery procedure