Service Function Chaining-Enabled I2NSF Architecture

(draft-hyun-i2nsf-triggered-steering-04)

IETF 100, Singapore

November 14, 2017

S. Hyun (Presenter), J. Jeong,
J. Park and S. Hares

Contents

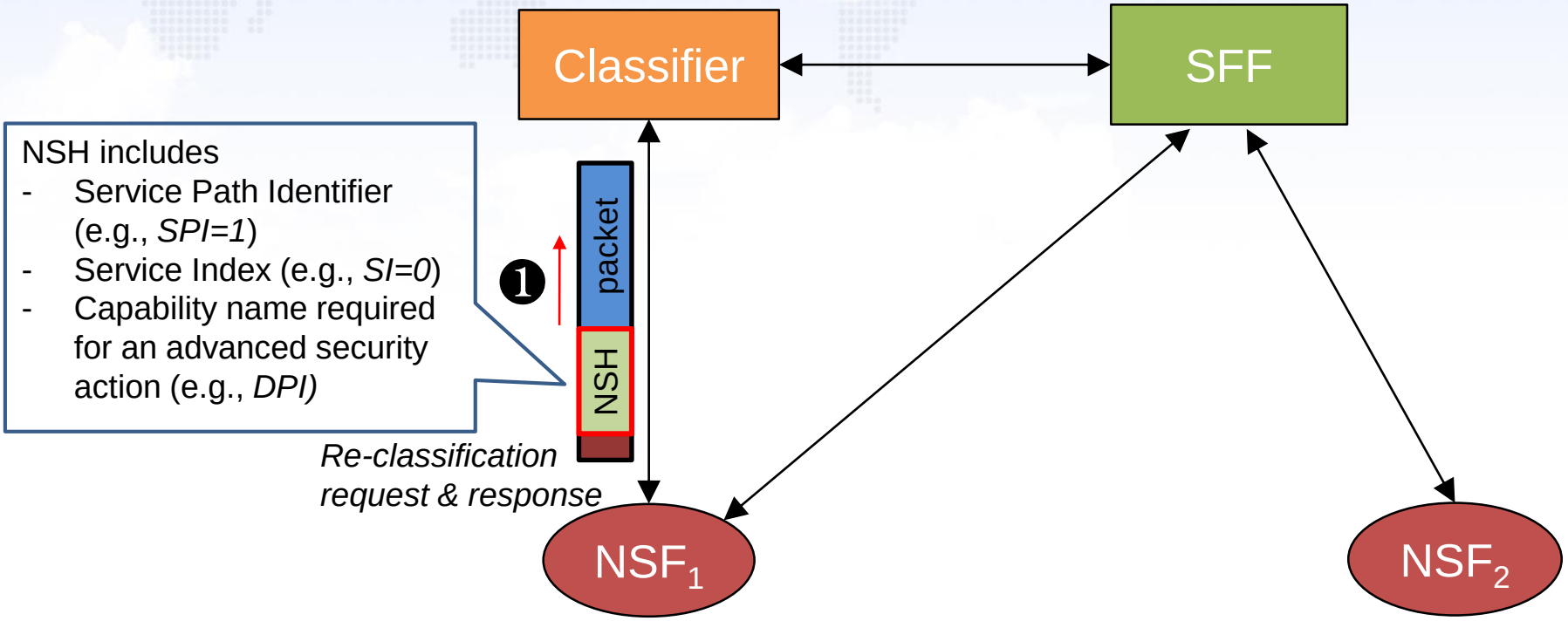
Introduction

- **Objective**
  - This document describes an architecture that integrates service function chaining (SFC) into the I2NSF framework to support packet forwarding between NSFs.

- **Motivation**
  - To support an advanced security action in the I2NSF framework that allows an NSF to call another type of NSF
  - To enable composite inspection of packets through various types of NSFs
  - To enable load balancing over multiple NSF instances combined with dynamic NSF instantiation

SFC-based Packet Forwarding in I2NSF

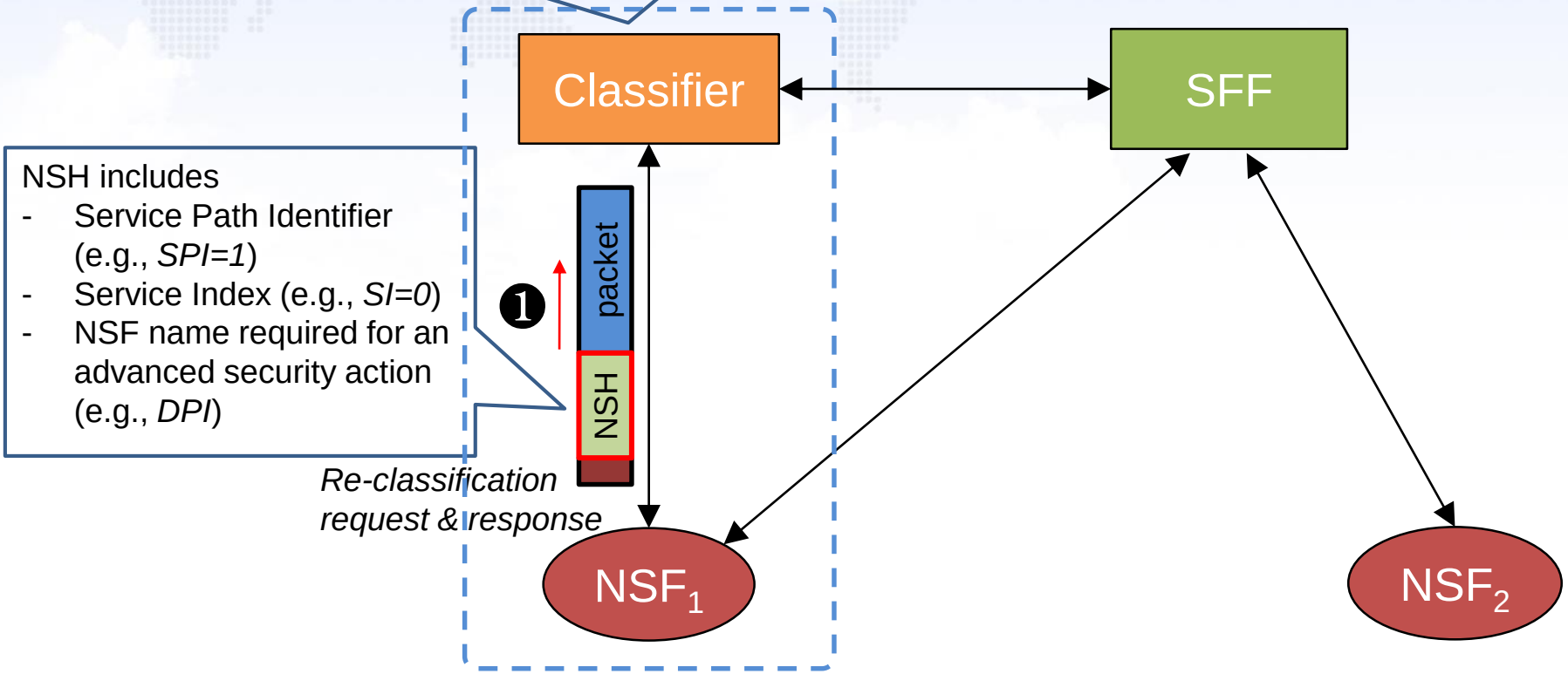
- To trigger an advanced security action, NSF_1 appends the capability name required for the advanced security action in NSH.



- SPI 1: NSF_1
- SPI 2: NSF_1 → NSF_2

SFC-based Packet Forwarding in I2NSF

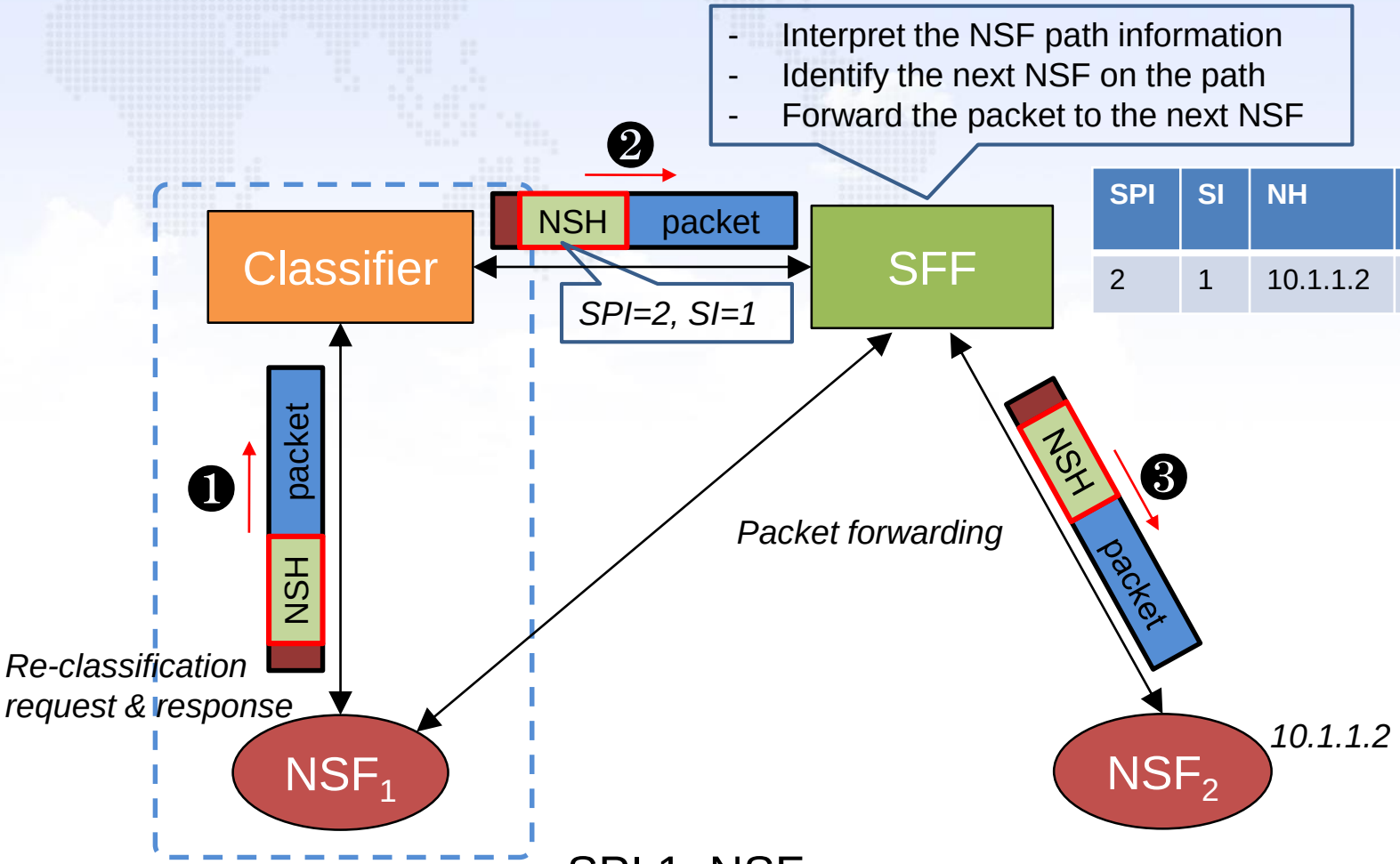
- Identify the particular NSF for DPI (NSF_2 is a DPI.) specified in NSH and determine the new NSF path of the packet
- Re-classification to change the existing path into the new one (*SPI=2, SI=1*)



The classifier may be co-resident with the NSFs.

- SPI 1: NSF_1
- SPI 2: NSF_1 → NSF_2

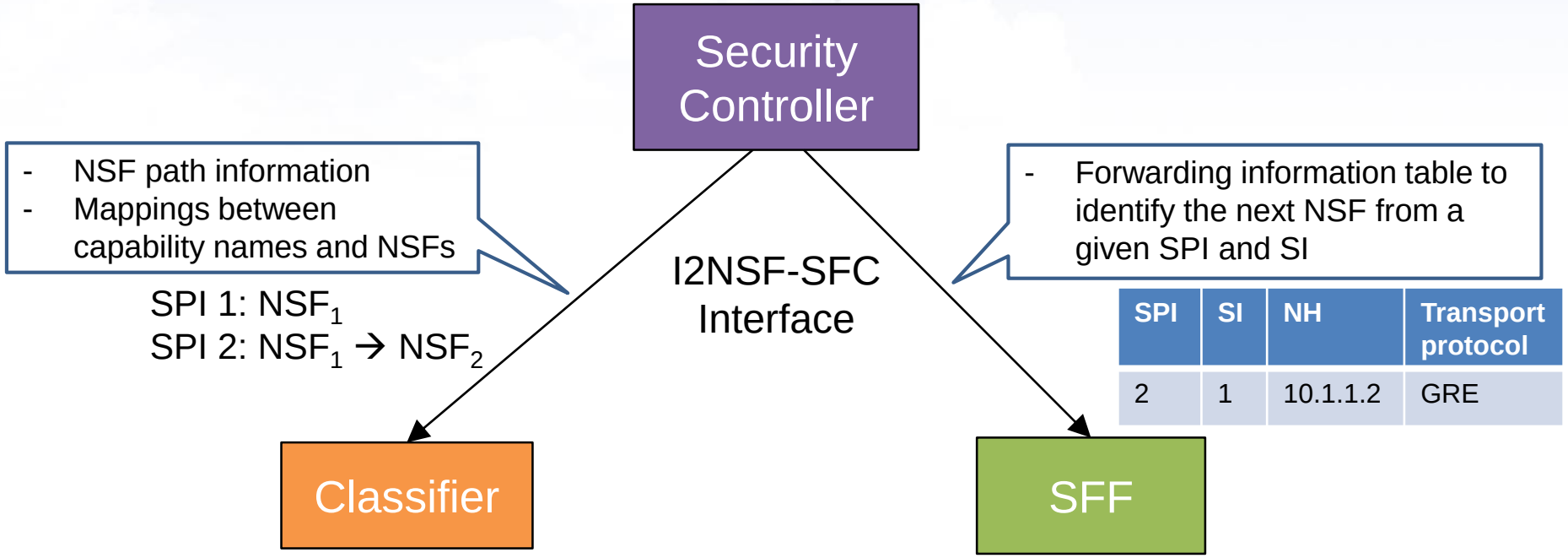
SFC-based Packet Forwarding in I2NSF



- SPI 1: NSF_1
- SPI 2: NSF_1 → NSF_2

Configuration for SFC

- The Security Controller configures the classifier with service function chain/path information.
- The Security Controller generates the forwarding information table of NSFs and configures the SFF with it.



Tunneling-based Forwarding

- Tunneling protocols can be utilized to support packet forwarding between SFF and NSF.
- We implemented network tunneling based on GRE (Generic Routing Encapsulation).

Packet format

L2 Header	**L3 Header (outer IP) Protocol=47**	**GRE header PT=0x894F**	**NSH NP=0x1 SPI=1 SI=1**	**Original packet**

Discussion

- SFC is suitable for enforcing the default (pre-determined) NSF path.

- Re-classification is required to support an advanced security action that the next NSF is determined in the I2NSF framework.
  - Introducing some overhead particularly when the classifier exists separately from an NSF

- Identifying a particular NSF for the given capability name (e.g., DPI) is required to fit into the I2NSF framework.
  - Interface between the Security Controller and SFC component (e.g., classifier, SFF) is required. → I2NSF-SFC Interface?

Update from -03 Version

- The following changes have been made from draft-hyun-i2nsf-nsf-triggered-steering-03.
  - Section 7 has been added to discuss implementation considerations of the SFC-enabled I2NSF architecture.

Next Step

- We will specify more details of what kind of information should be included in the NSH header to support packet forwarding between NSFs and the formats.

- Design of I2NSF-SFC Interface
  - Information model & data model

Thank you!

Any questions or comments?