Fast and Scalable Network Monitoring Applications based on Data Stream Processing

Presenter: Alessandra Fais

PhD student at University of Pisa

Outline

- Introduction
  - Problem and main goals

- What can be found in literature

- Proposed approach
  - Continuous monitoring as stream analytics problem
  - Architecture and challenges

- Conclusions

Introduction

- Modern networks
  - Accommodate a variety of services
  - Shared infrastructure
  - Adapt to requests for service (de-)activation
  - Stringent Quality of Service requirements

Introduction

➢ How to manage modern networks?

Introduction

➢ How to manage modern networks?

- Network operators need
  - Rapid and easy network (re-)configuration
  - Continuous real-time network monitoring for detecting
    - Security issues
    - Performance degradation

Introduction

➢ How to manage modern networks?

- Network operators need
  - Rapid and easy network (re-)configuration
  - Continuous real-time network monitoring for detecting
    - Security issues
    - Performance degradation

What can be found in literature

➢ Main tradeoff: **expressiveness** VS **performance**

What can be found in literature

➢ Main tradeoff: **expressiveness** VS **performance**

- Solutions based on
  - Stream processing only   ➡   high expressiveness
 - Programmable switches alone   ➡   high performance

What can be found in literature

➢ Main tradeoff: **expressiveness** VS **performance**

- Solutions based on
  - Stream processing only   ➡   high expressiveness
  - Programmable switches alone   ➡   high performance

- Can we provide both?
  - Some new proposals use stream processing AND programmable switches
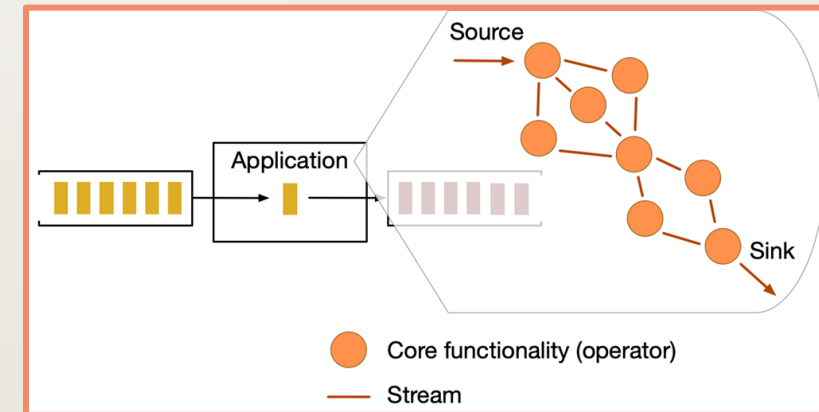 - Proposed approach based on stream processing ONLY

A new framework: main goals

- **High performance: real-time continuous data analysis**
  - Stream processing pipelines
  - Implementation through Data Stream Processing frameworks
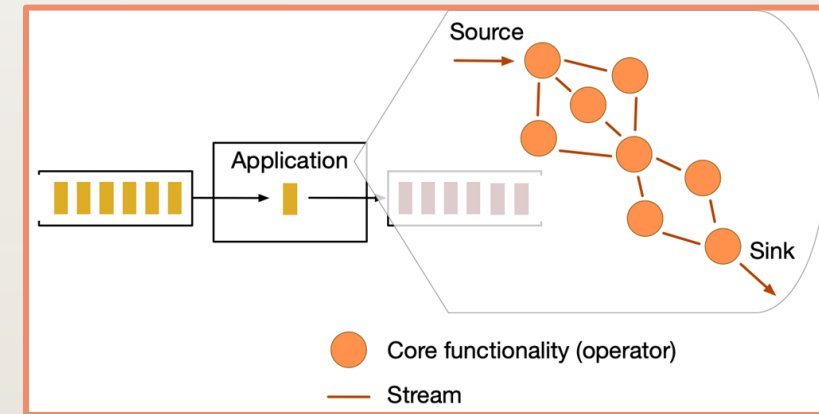
A new framework: main goals

- **High performance: real-time continuous data analysis**
  - Stream processing pipelines
  - Implementation through Data Stream Processing frameworks

- **Scalability**
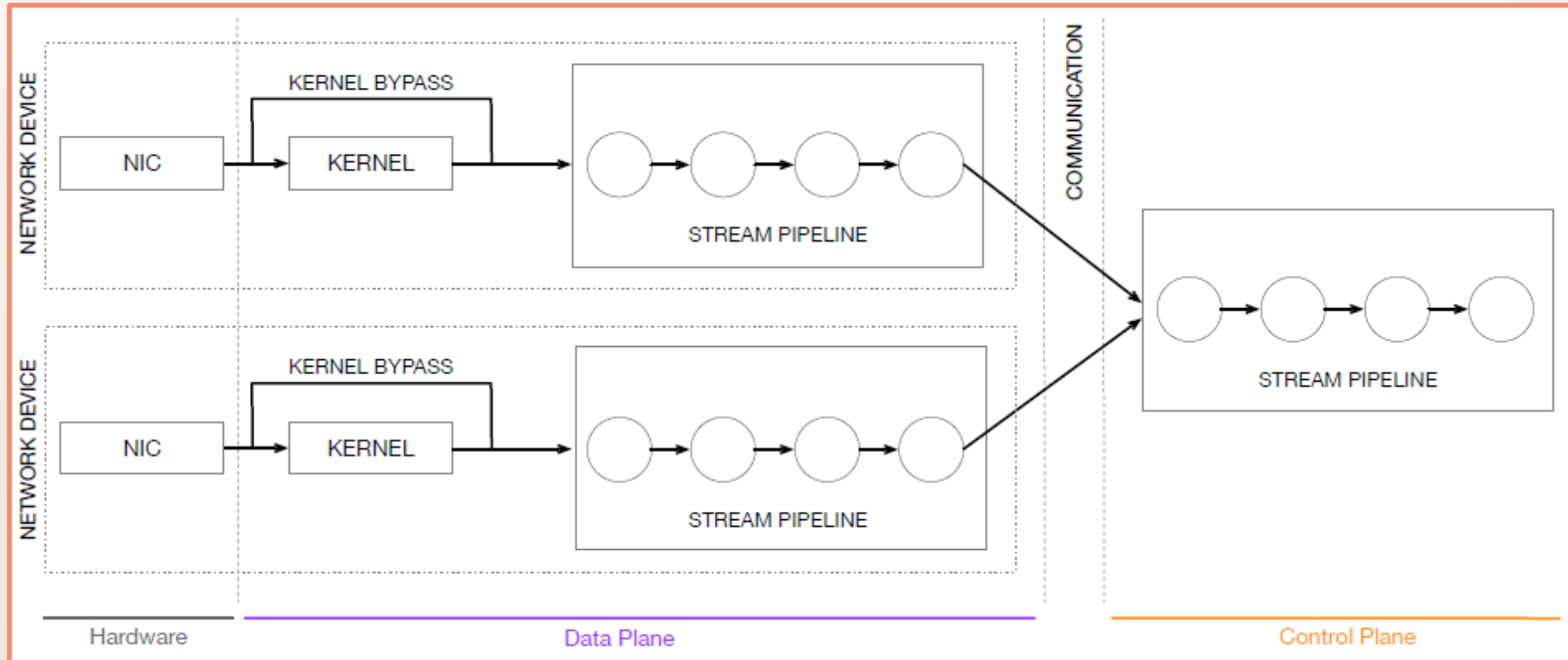 - Distributed processing over programmable network nodes

A new framework: main goals

- **High performance: real-time continuous data analysis**
  - Stream processing pipelines
  - Implementation through Data Stream Processing frameworks

- **Scalability**
  - Distributed processing over programmable network nodes

- **High expressiveness**
  - Provide high level programming abstractions
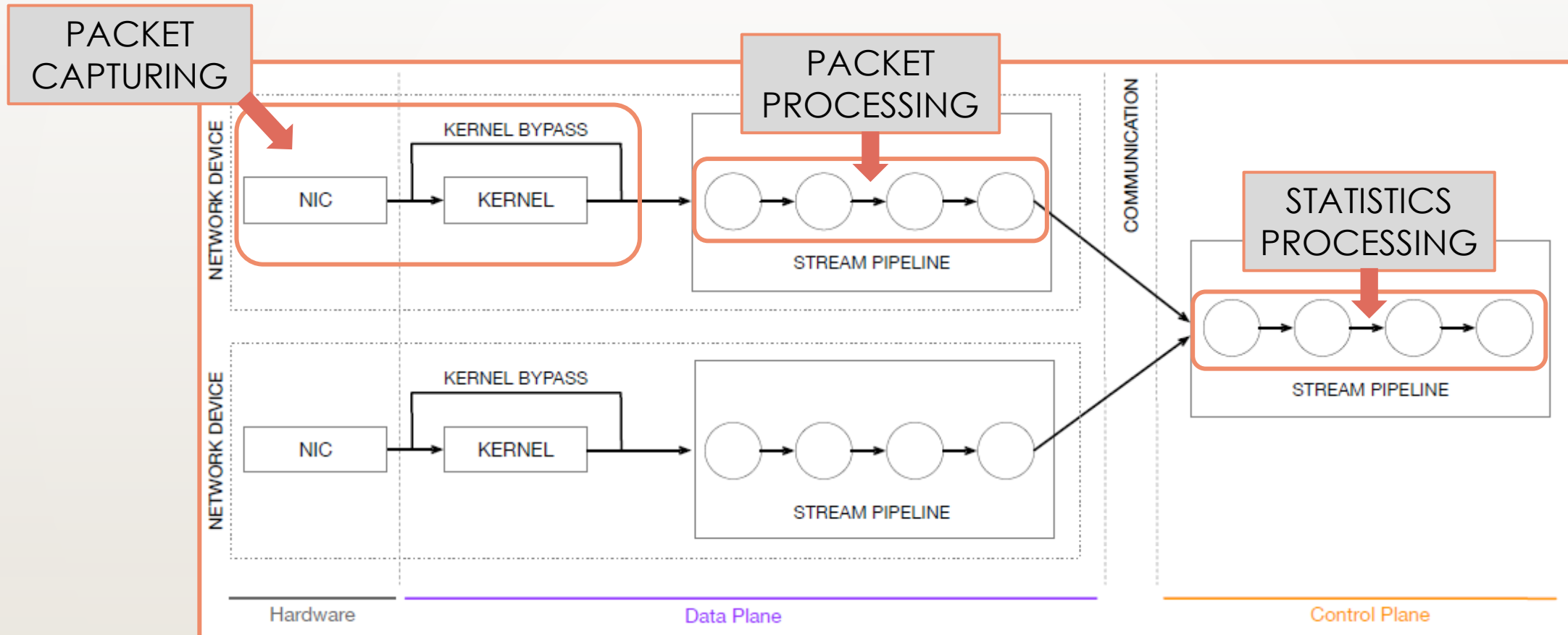 - Support for complex analysis queries
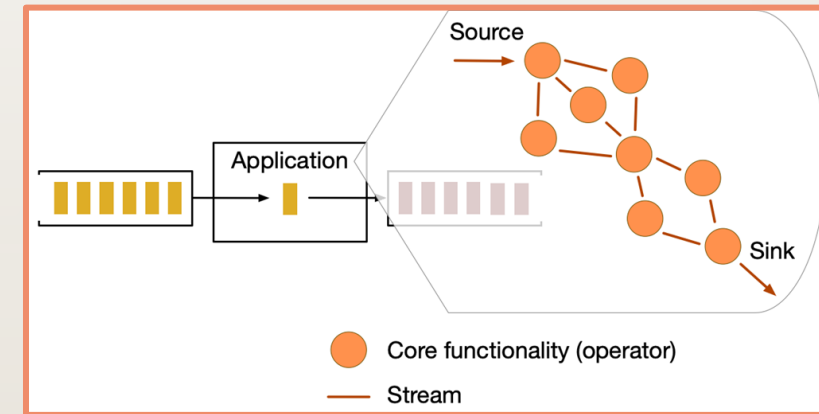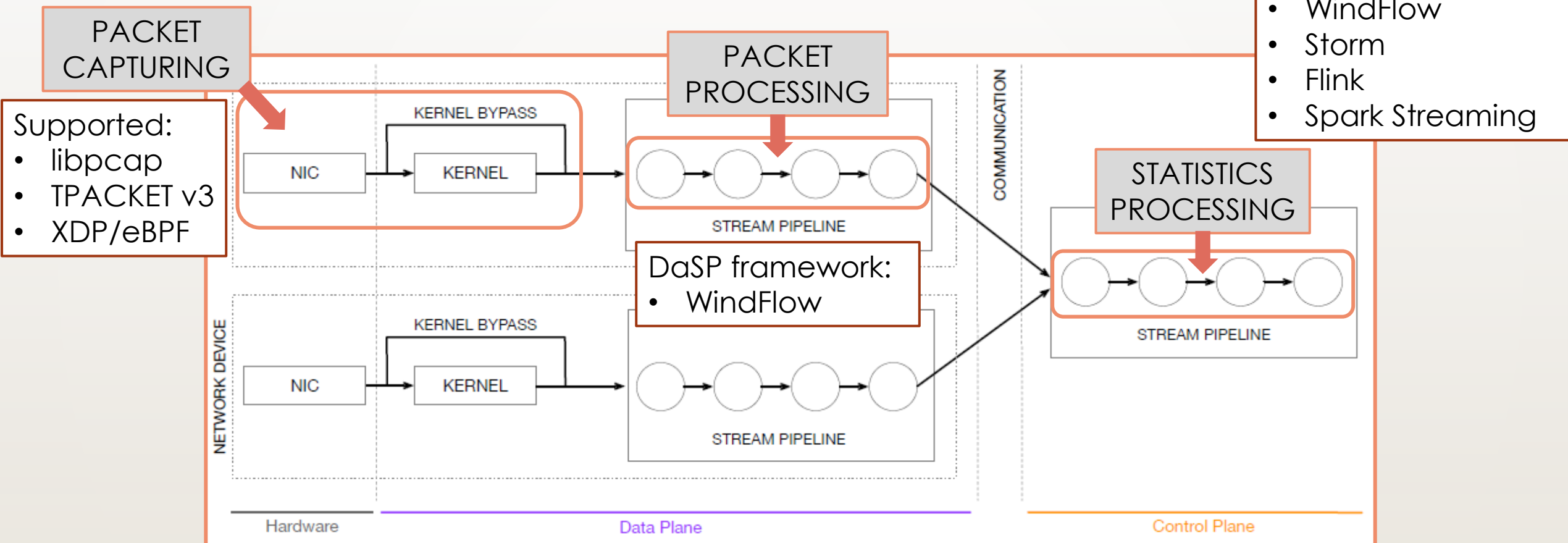
A new framework: architecture

A new framework: architecture

A new framework: architecture

PACKET CAPTURING

Supported:
- libpcap
- TPACKET v3
- XDP/eBPF

PACKET PROCESSING

DaSP framework:
- WindFlow
- Storm
- Flink
- Spark Streaming

STATISTICS PROCESSING

DaSP framework:
- WindFlow

NETWORK DEVICE

KERNEL BYPASS

NIC → KERNEL

KERNEL BYPASS

NIC → KERNEL

STREAM PIPELINE

STREAM PIPELINE

COMMUNICATION

STREAM PIPELINE

Hardware | Data Plane | Control Plane

Conclusions

- Cheaper solution
  - Programmable switches are expensive
  - Use general purpose network devices (multicores + SmartNICs, GPUs, FPGAs)

Conclusions

- Cheaper solution
  - Programmable switches are expensive
  - Use general purpose network devices (multicores + SmartNICs, GPUs, FPGAs)
- High performance
  - Optimized usage of resources
  - Parallelism in all computation stages (packet capturing, processing)

Conclusions

- Cheaper solution
  - Programmable switches are expensive
  - Use general purpose network devices (multicores + SmartNICs, GPUs, FPGAs)

- High performance
  - Optimized usage of resources
  - Parallelism in all computation stages (packet capturing, processing)

- High-level interface
  - Simplify the implementation of real-time continuous monitoring applications
  - Encourage network programmers to use the proposed framework