



# Transparent TSN for Agnostic End-hosts via P4-based Traffic Characterization at Switches

Cornelia Brühlhart ♣, Nurefşan Sertbaş Bülbül ♣,  
Nils Ole Tippenhauer ◇, Mathias Fischer ♣

♣ Universität Hamburg

◇ CISPA Helmholtz Center for Information Security

# Motivation

## Challenges in Mission-Critical Systems



# Background

Standard Ethernet is insufficient

---

## Standard Ethernet

Best-effort packet delivery

No inherent latency guarantees

Optimized for general-purpose

### **Problem:**

Standard Ethernet cannot ....

- Satisfy MCS timing and bandwidth constraints
- Inherently solve its packet loss
- Offer security for resource constrained devices

## Time-Sensitive Networking (TSN)

Deterministic delivery of data

Traffic-shaping and QoS

Low-latency and high bandwidth

### **Problem:**

TSN....

- Is built upon several standards
- Networks have compatibility issues
- Requires careful bandwidth allocation
- Needs to overcome security implications

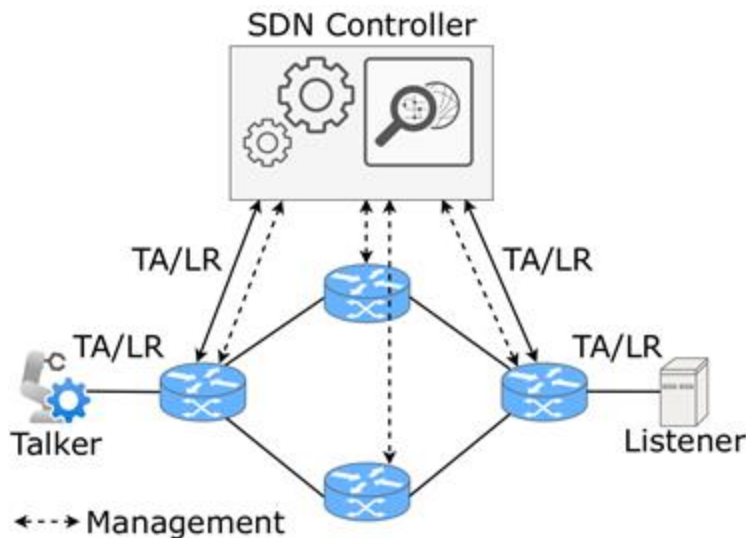
# Problem statement and Research Question

End-hosts need to know **how** to schedule their transmission (according to the network's timing and rules)

**= TSN-Awareness**

Research Question:

How can we eliminate end-host dependencies in TSN and facilitate traffic flow management?



Centralized Control via SDN

- + Simplified resource management
- + Facilitates dynamic traffic control
- Latency concerns
- Potential bottleneck as network grows

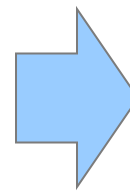
# Approach

## Transparent TSN

---

Shift TSN responsibilities from end-hosts to network infrastructure

- Traffic classification
- Traffic identification
- Network management interaction



~~SDN-Controller~~  
~~End-hosts~~  
**Switches**

- + Lower complexity
- + Lower network management overhead
- + Reduced latency
- + More scalable and robust network
- + Anomaly detection

# Approach

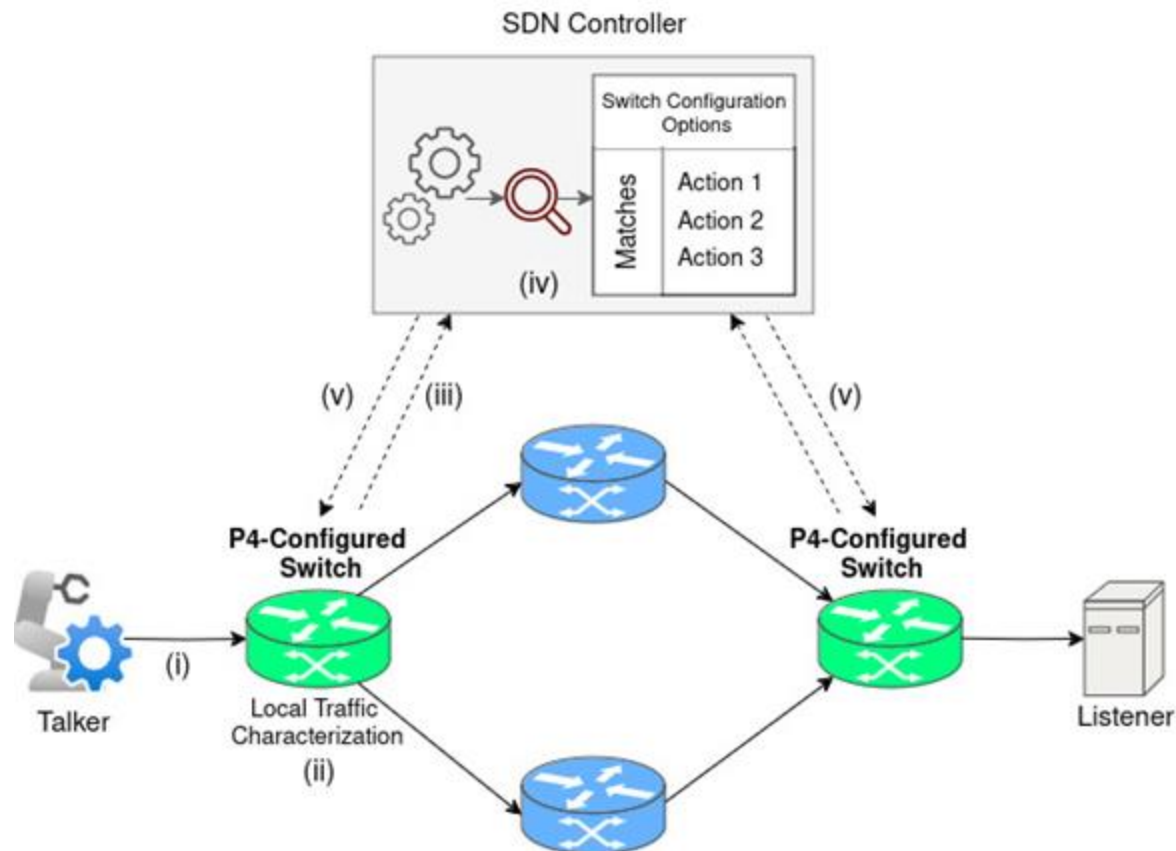
## Decentralized In-Switch Monitoring

### Edge switches:

- Disassemble packets
- Extract traffic characteristics
- Determine traffic type locally

$$R(k) = \frac{\sum_{t=1}^{N-k} (x_t - \bar{x})(x_{t+k} - \bar{x})}{\sum_{t=1}^N (x_t - \bar{x})^2} \quad \forall N, \bar{x}, k \in \mathbb{N}$$

$R(k)$  : Autocorrelation coefficient at lag  $k$ .  
 $N$  : Total number of frame inter-arrival times.  
 $x_t \geq 0$  : Inter-arrival time of the  $t$ -th frame.  
 $\bar{x}$  : Mean inter-arrival time.  
 $k$  : Lag, inter-arrival times displacement.

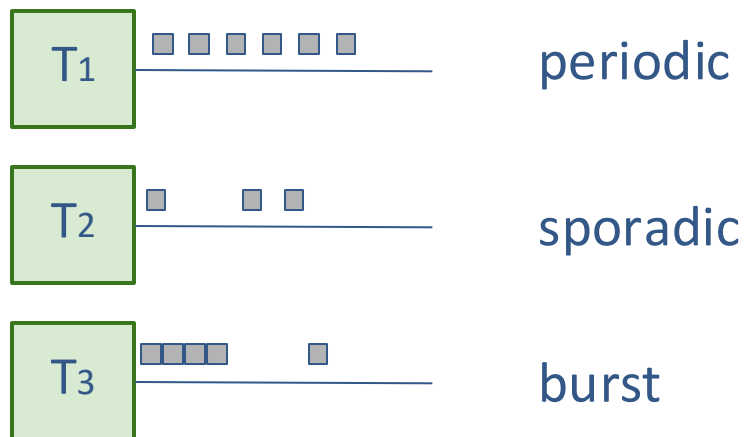


# Implementation

## System Setup

- Network setup in Mininet 2.3.1ba
- Simple Switch Behavior Model v2 (BMv2)
- Switch programming using P4
- Multiple scripts for traffic generation
- Calculate the autocorrelation to classify traffic

Mininet



# Implementation

## Traffic Characterization

### Problem No. 1:

P4 does **not maintain state** between packet processing operations.

### Problem No. 2:

Which algorithm to pick for traffic characteristics when P4 can only use **limited arithmetic operations**?

### Problem No. 3:

How much traffic should be monitored / stored for classification?

Counter Register

Flow Iteration	Inter-Arrival	Per Flow Packet	Periodic	Sporadic	Burst	...
----------------	---------------	-----------------	----------	----------	-------	-----

Flow Register

Src Address	Dst Address	Prev Timestamp	Inter-Arrival Time 1	Inter-Arrival Time 2	...	Inter-Arrival Time N
-------------	-------------	----------------	----------------------	----------------------	-----	----------------------

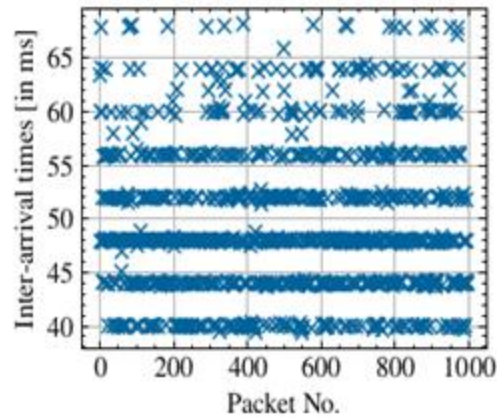
Autocorrelation Register

Src Address	Dst Address	Auto-correlation 1	Auto-correlation 2	Auto-correlation 3	...	Auto-correlation N-1
-------------	-------------	--------------------	--------------------	--------------------	-----	----------------------

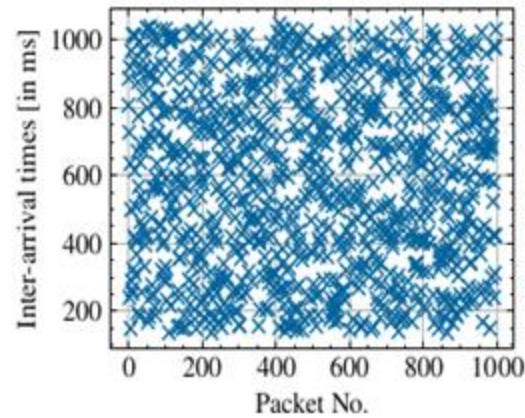


# Results

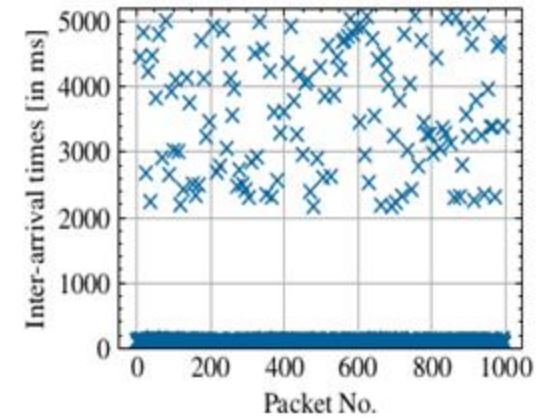
## Traffic Characterization with P4 on the Switch



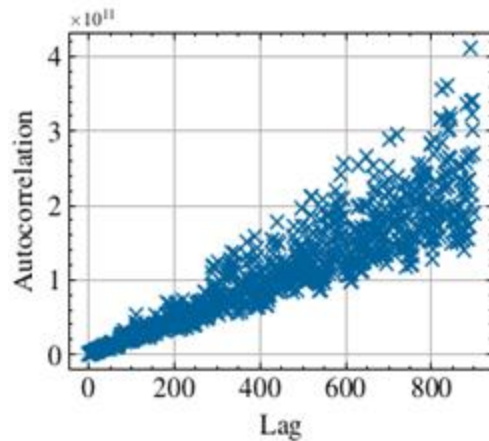
(a) periodic



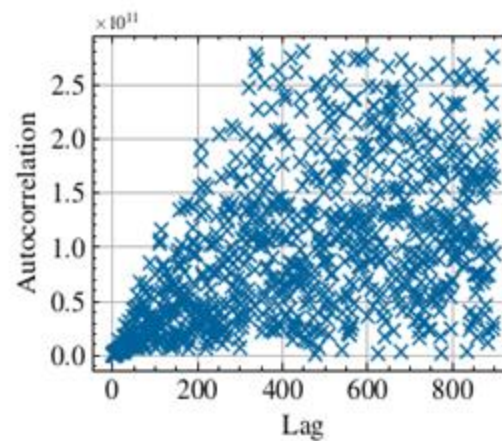
(b) sporadic



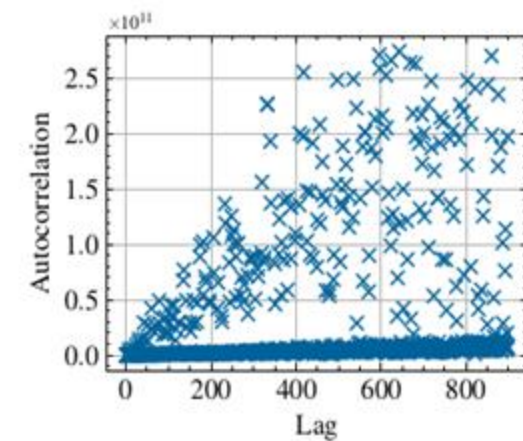
(c) burst



(a) periodic



(b) sporadic



(c) burst

# Evaluation

## Traffic Characterization with P4 on the Switch

---

### Sliding Windows Evaluation:

Register Size  $N = 500 / 200 / 100 / 50$

Precision and Recall for varying register sizes

Register Size $N$	50	100	250	500
Precision	0.7	0.67	0.73	<u>0.8</u>
Recall	0.74	0.67	0.73	1

### Confusion Matrix

200 periodic and 200 non-periodic traffic flows

Sent vs. Characterized	Periodic	Non-Periodic
Periodic (sent)	200 (TP)	0 (FN)
Non-Periodic (sent)	51 (FP)	149 (TN)

### Forwarding Latency

tshark to capture ingress and egress ports on the switch: **0.22 ms** per packet

# Conclusion

---

## Our approach:

- Reduces communication overhead / network load
- Reduces latency compared to traditional / SDN approaches
- Remove end-host dependencies for reservation requests
- In-Switch anomaly detection for traffic classes

Network participants reap the benefits of TSN

“Costs” only 0.2 ms per packet

Towards autonomous switches for a flexible network

Facilitates the switch from legacy systems to time-sensitive networks

We taught resource-constrained devices (switches) to characterize time-sensitive traffic locally!

# Thank you!

## Time for questions

### Transparent TSN for Agnostic End-hosts via P4-based Traffic Characterization at Switches

Cornelia Brühlhart<sup>1</sup>, Nurefşan Serbay Bilbil<sup>1</sup>, Nils Ole Tippenhauer<sup>1</sup>, Mathias Fischer<sup>2</sup>  
<sup>1</sup>Universität Hamburg, Germany  
{cornelia.bruehlhart, nurefshan.serbaybilbil, nils.ole.tippenhauer}@uni-hamburg.de  
<sup>2</sup>CISPA Helmholtz Center for Information Security, Germany  
tippenhauer@cispa.de

**Abstract**—Mission-critical networks currently face a transition from legacy network protocols to advanced time-sensitive networking (TSN) standards. TSN guarantees reliable and deterministic communication using off-the-shelf Ethernet equipment. However, end-hosts must be TSN-aware and may pose security risks by arbitrarily over-allocating resources. Integrating central instances like a software-defined networking (SDN) controller into TSN networks to streamline network management presents a promising solution. This raises concerns regarding latency in communication between switches and the controller, as well as among switches themselves. To address this, we propose an approach that renders TSN transparent to end-hosts, eliminating the need for their involvement in resource reservations. We embed packet processing logic in P4-enabled TSN switches to characterize network traffic intelligently. This enables switches to allocate network resources autonomously and adjust real-time traffic handling mechanisms. Leveraging P4 storage structures introduces statefulness for traffic characterization computing within the inherently stateless P4 language. Our experiments demonstrate that our P4-enhanced switches require a minimal 0.814 MB of on-chip memory to distinguish between periodic and non-periodic traffic with an 80% precision while incurring a mere 0.2 ns forwarding latency per packet.

**Index Terms**—P4 Programming, Traffic Characterization, Software-Defined Networking, Time-Sensitive Networking, Real-Time Traffic Management

#### I. INTRODUCTION

MISSION-CRITICAL networks, characterized by their deterministic and reliable communication in low-latency settings, face the challenge of adapting to modern system environments' increasing heterogeneity and agility. In response, time-sensitive networking (TSN) has emerged, offering deterministic and reliable communication in low-latency settings that align well with the demands of modern system environments. For example, network infrastructure in smart manufacturing facilities must meet stringent quality of service (QoS) and time-sensitive demands, where integrity in real-time data exchange is crucial for seamless operations. However, as there is a shift towards adopting cost-effective commercial off-the-shelf (COTS) hardware and software over specialized field boxes, orchestrating interactions among network components remains complex in real-time applications. Implementing TSN using COTS hardware and software can ensure QoS, but it requires end-hosts to be TSN-aware, thus restricting the network's flexibility.

In response, ongoing research explores the feasibility of central control in time-sensitive networks. Network management can be facilitated, e.g., through the holistic view of a centralized entity, similar to the concept of software-defined networking (SDN) [1]. SDN allows modification and specification of network behavior through software, effectively separating the control plane from the data plane. A central entity, known as the SDN controller, dynamically manages the traffic and configures the forwarding behaviour of the switches based on the specific application requirements. However, SDN also introduces challenges, especially in heterogeneous network environments, such as TSN. The central (TSN) controller must accommodate diverse device characteristics and network policies. This may result in latencies of up to 0.3 ms per packet [2] and significantly impact network operations' real-time nature. Additionally, the dynamic traffic and resource orchestration in TSN networks involve frequent communication between network components and the TSN controller [3], [4]. As network traffic increases, the controller must process high volumes of messages, which may become a bottleneck in the network.

These challenges show the urgent need for a transparent TSN mechanism that guarantees dynamic, deterministic, and low-latency communication for the desired QoS without mandating TSN-capable end-hosts. In this work, we propose to shift TSN tasks and configuration responsibilities from the network's periphery (the end-hosts) to its internal network components to address this challenge. This enables the network to autonomously manage and adapt to traffic demands and communicate responses, achieving transparent TSN. By identifying various traffic types in real-time, the network can dynamically allocate resources, reduce the burden on end-hosts, and minimize messaging frequency to the controller. Hence, our solution for transparent TSN involves transferring TSN tasks and network management responsibilities from the end-hosts and the TSN controller to the network switches. Empowering switches to manage and identify traffic locally eliminates the need for end-host TSN-awareness. Furthermore, autonomous traffic type characterization at switches reduces messaging frequency between the TSN controller and switches, minimizing network load and reducing latency.

Cornelia Brühlhart

[cornelia.bruehlhart@uni-hamburg.de](mailto:cornelia.bruehlhart@uni-hamburg.de)

Research Associate / PhD student  
Universität Hamburg