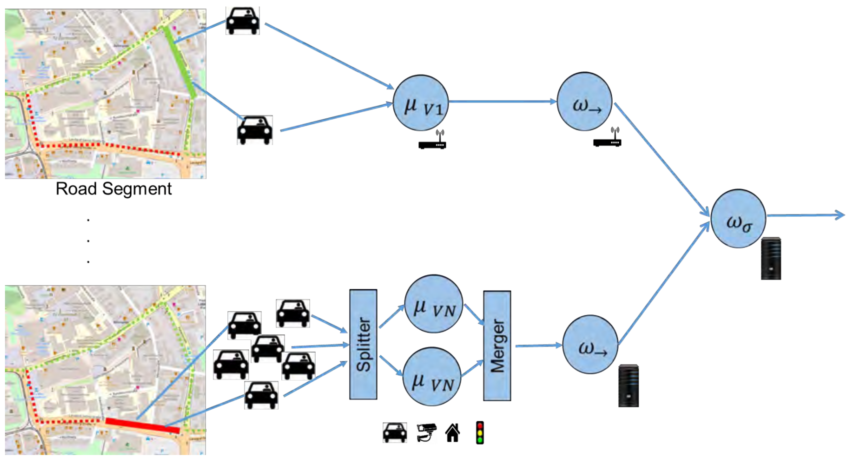
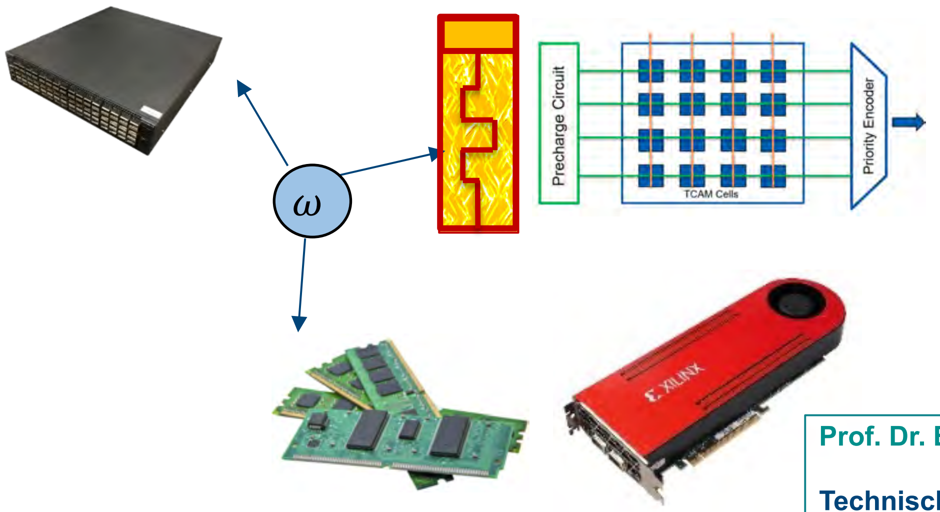
In-Network Computing: Effizientere Datenanalyse durch Netzzentrisches Rechnen?

Boris Koldehofe

VFFIA Workshop 2024





Prof. Dr. Boris Koldehofe

Technische Universität Ilmenau
Department of Computer Science and Automation
Distributed Systems and Operating Systems Group

17-Apr-24

Short Introduction
Boris Koldehofe

Distributed and Operating Systems Group

Technical University of Ilmenau

Research

- Distributed data analytics
- Computer system principles
- Reliability, Performance, Energy-Efficiency
- Information Security

Specific Focus

- Distributed Event-based systems (DEBS)
- In-Network Computing

Data Driven Applications

Nowadays everywhere!

- Autonomous driving, smart factories, smart cities, telemedicine, and many more

MAPE loop of IoT services:

- **M**onitor and **A**nalyze “Things”
- **P**lan and **E**xecute Processes

Insights into data key to adapt applications

- Billions of things
- Exabytes of context knowledge

But Performance and Low Latency is not straight forward!

Outline

Why low latency response?

The Bottleneck in Data Movements

In-Network Computing Technologies accelerating performance

Examples in the context of Distributed Data Analytics Middleware

Conclusion

Low Latency responses

Often relates to highly accurate timestamps of events

Manufacturing process

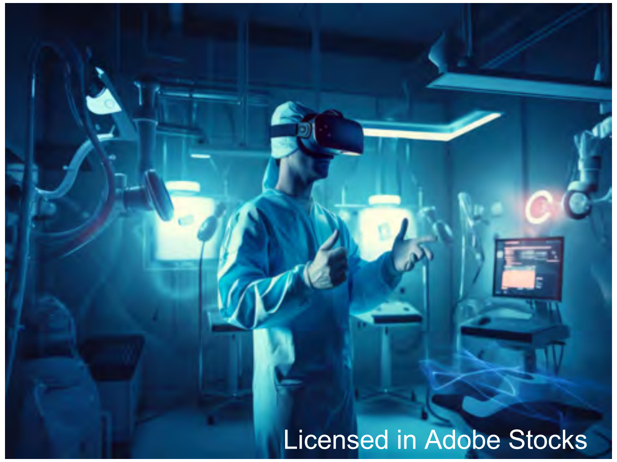
- Understand correct position over time
- Low Jitter in Communications

Telemedicine

- Understand situations with very low reaction time

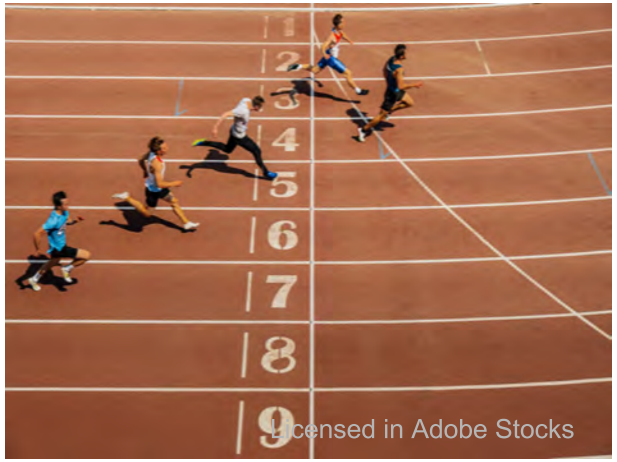
Financial applications

- Algorithmic trading
- Very low responses in detecting and analyzing packets



Timestamp inaccuracy	Location Inaccruracy
1s	10m
1ms	10cm
1µs	0.1mm

Timestamp-Location Accuracy



Moving Object of 36km/h

Improving Response & Timestamp Accuracy
Technological Developments

5G and even 6G Campus networks

- Goal interconnect processors fast
- 100µs - 1ms delays, high mobility

TSN

- Deterministic Real-time guarantees for industrial applications

Edge Computing

- Offload Computations

Accelerators

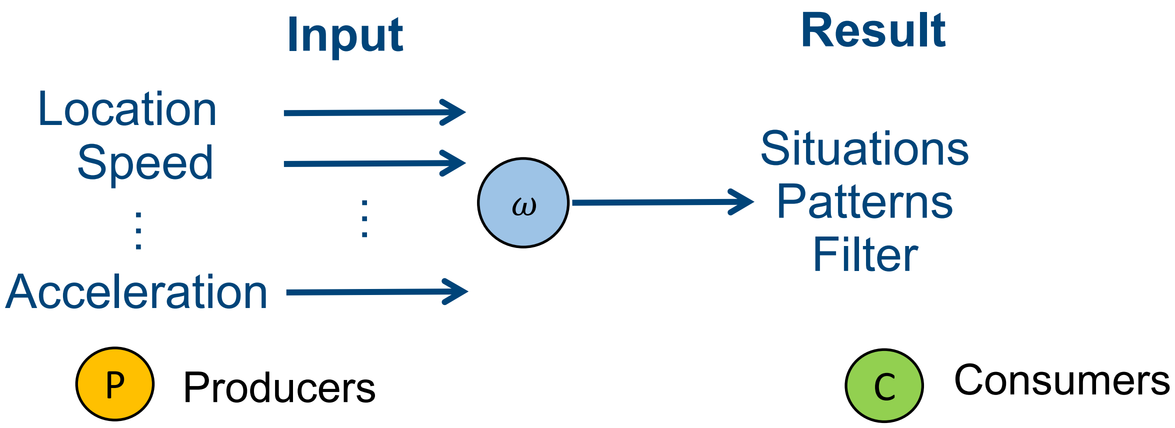
- Computation
- I/O
- Protocols / Architectures

Ilmenau 6G Campus Network

Real-Time Data Analytics

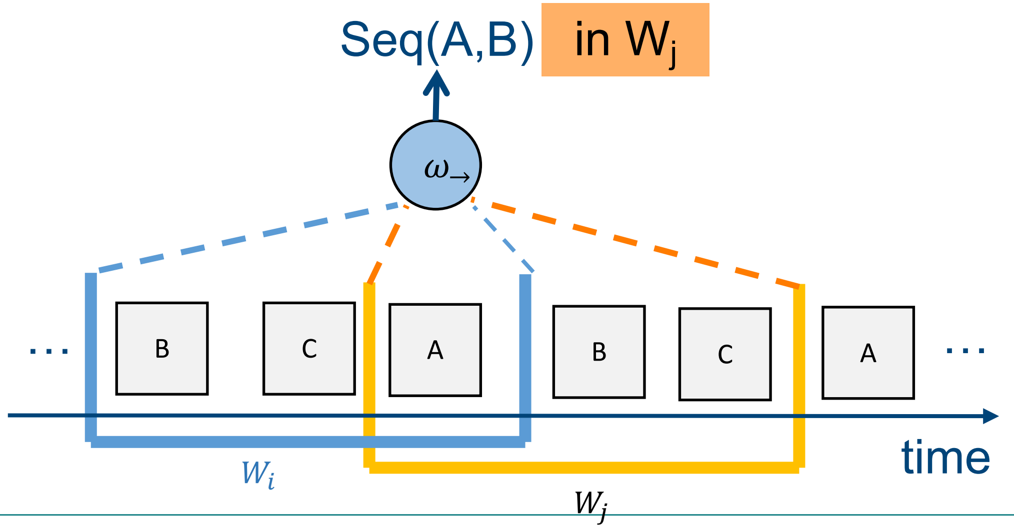
Correlations on data stream

- With low end to end delay
- High accuracy detection



Paradigm:

- *Operators* identify pattern on partial data stream: *window*
- E.g. *CEP operator*, Filter, Neural Network, Deep Learning Model



The SPIRIT of science
TECHNISCHE UNIVERSITÄT ILMENAU

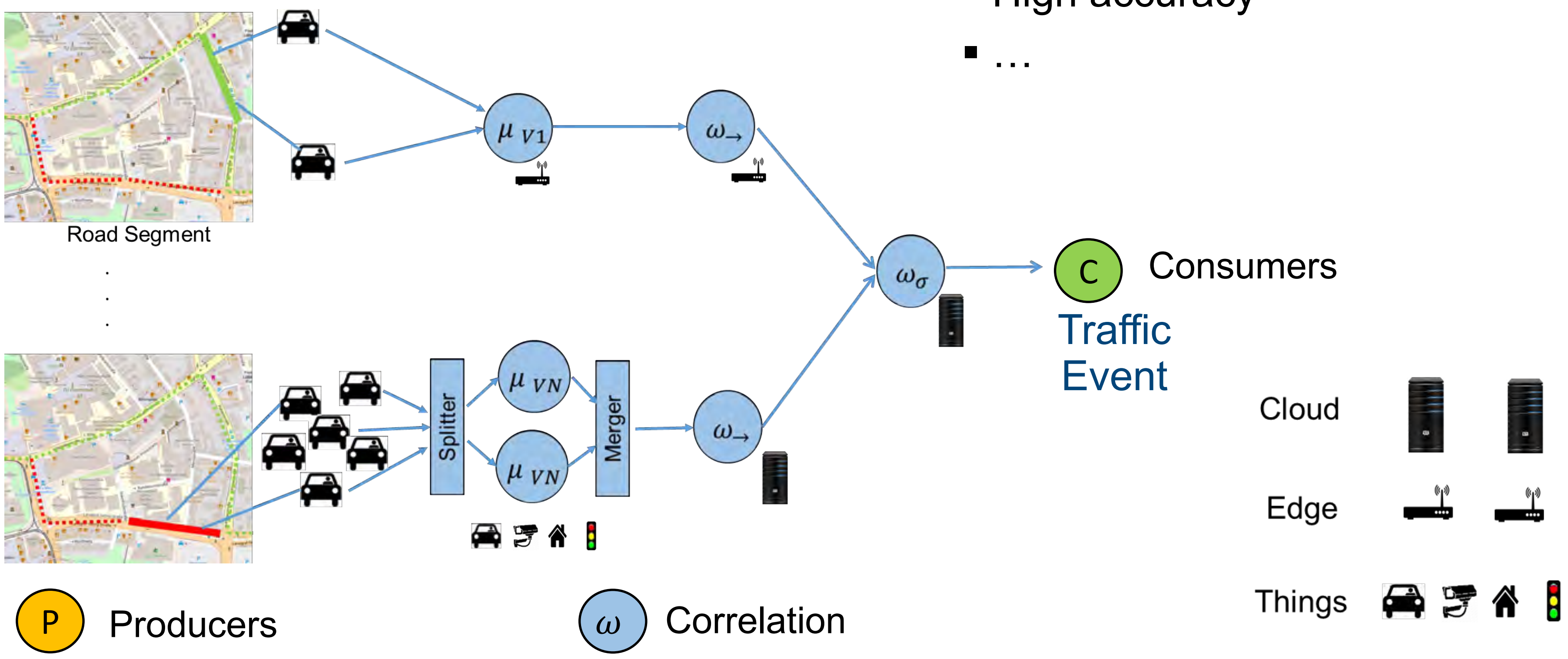
Distributed Real-Time Data Analytics

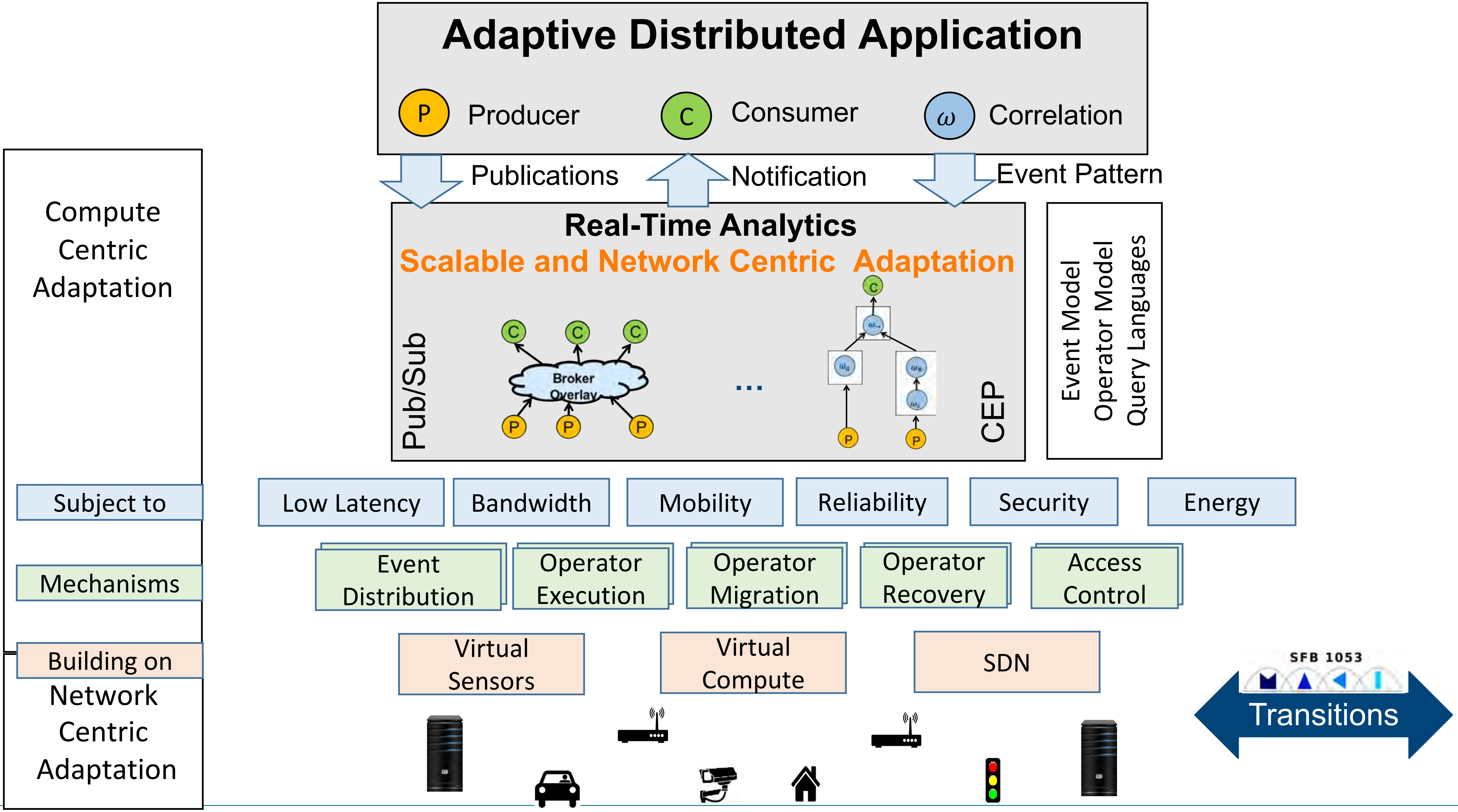
Execute operator network on a distributed infrastructure

Increase Scalability and Performance

Optimization subject to potentially conflicting goals

- Decoupling producers and consumers
- Low end-to-end delay
- High accuracy
- …



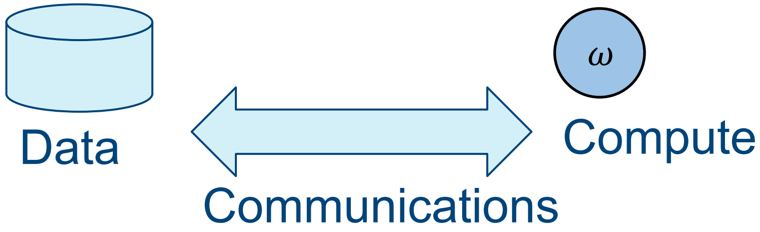

Adaptive Distributed Application
P
Producer
C
Consumer
ω
Correlation
Publications
Notification
Event Pattern
Real-Time Analytics
Scalable and Network Centric Adaptation
Pub/Sub
Broker Overlay
C
P
…
CEP
Event Model
Operator Model
Query Languages
Compute Centric Adaptation
Subject to
Low Latency
Bandwidth
Mobility
Reliability
Security
Energy
Mechanisms
Event Distribution
Operator Execution
Operator Migration
Operator Recovery
Access Control
Building on
Network Centric Adaptation
Virtual Sensors
Virtual Compute
SDN
SFB 1053
Transitions

Meeting Performance of Time Sensitive Distributed Applications

Cyberphysical application

- Low latency?
- Predictable performance?

Bottlenecks in data movement and processing



Requires much more flexibility in using mechanisms of the distributed infrastructure!

Ingredients for Increased Flexibility

Programable hardware

- P4 Switches
- NetFPGA

New networking paradigms

- Software-Defined Networking
- Network Function Virtualization

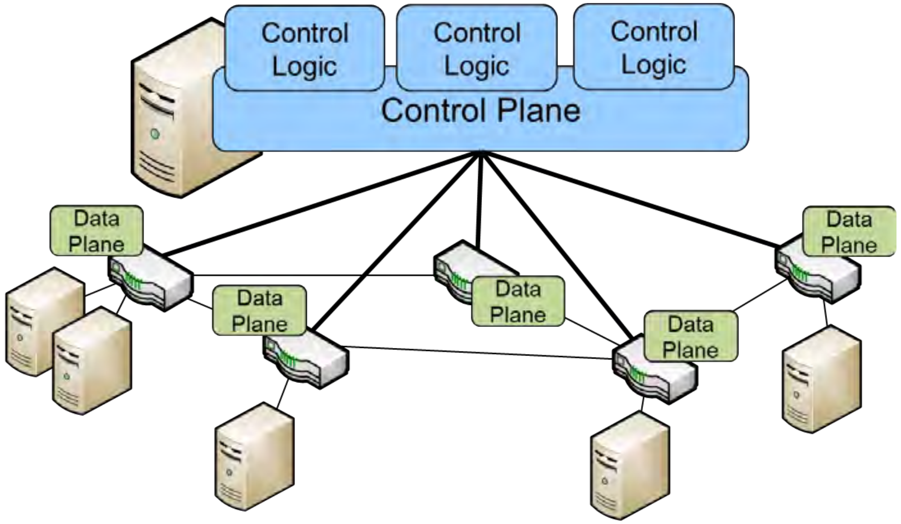
Significant changes in the infrastructure

- Edge Data Center
- Technologies & Concepts
  - DPDK, P4, OpenFlow, RDMA

Enabler for in-network computing!

Barefoot Tofino

FPGA



In-Network Computing

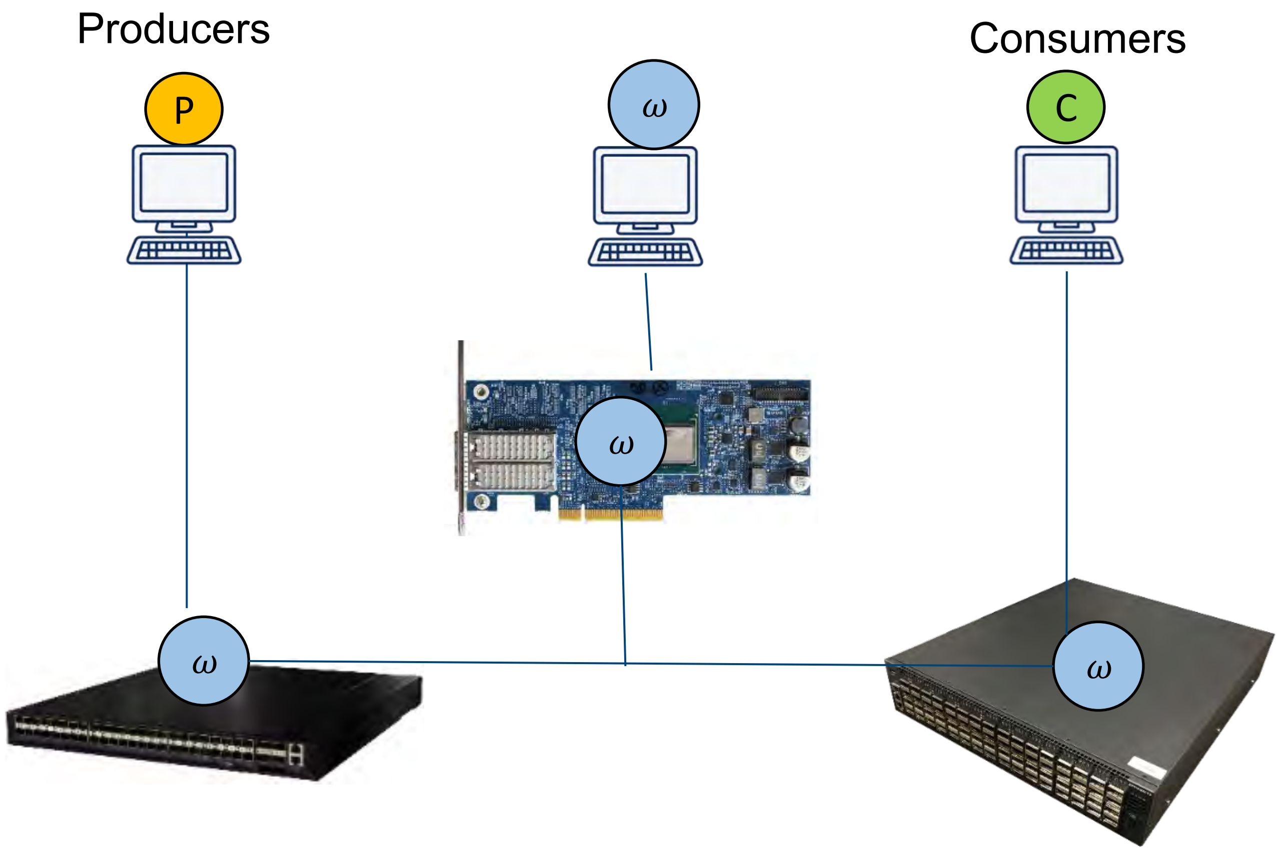
Idea enable computations on the data path

Traditionally,

- Packet header processing,
  - e.g., routing, firewall, packet classification, load balancing, deep packet inspection

Often

- Match/action pipeline model of networking hardware
- Management interface, specific programming interfaces, …



Evolution: In-Network Computing resources

Towards flexible, high performance, and energy-efficient in-network computing

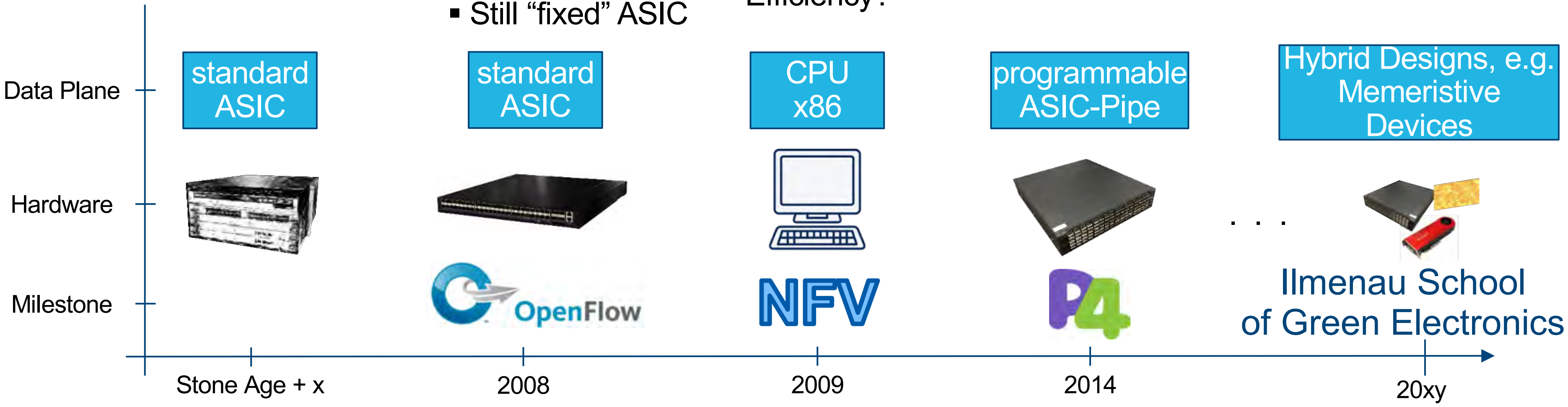
- "Blackbox"
- Vendor-lock

- "Blackbox" → "Whitebox"
- Software-defined Networking
- Controller Interface
- Still "fixed" ASIC

- "Why hardware?"
- Just do it in C!
- Standard Server
- DPDK
- Performance?
- Efficiency?

- Programmable ASIC for packet processing
- Very high bandwidth
- Limited flexibility

- Energy Efficient Switching?
- Computational Intelligence inside the network?

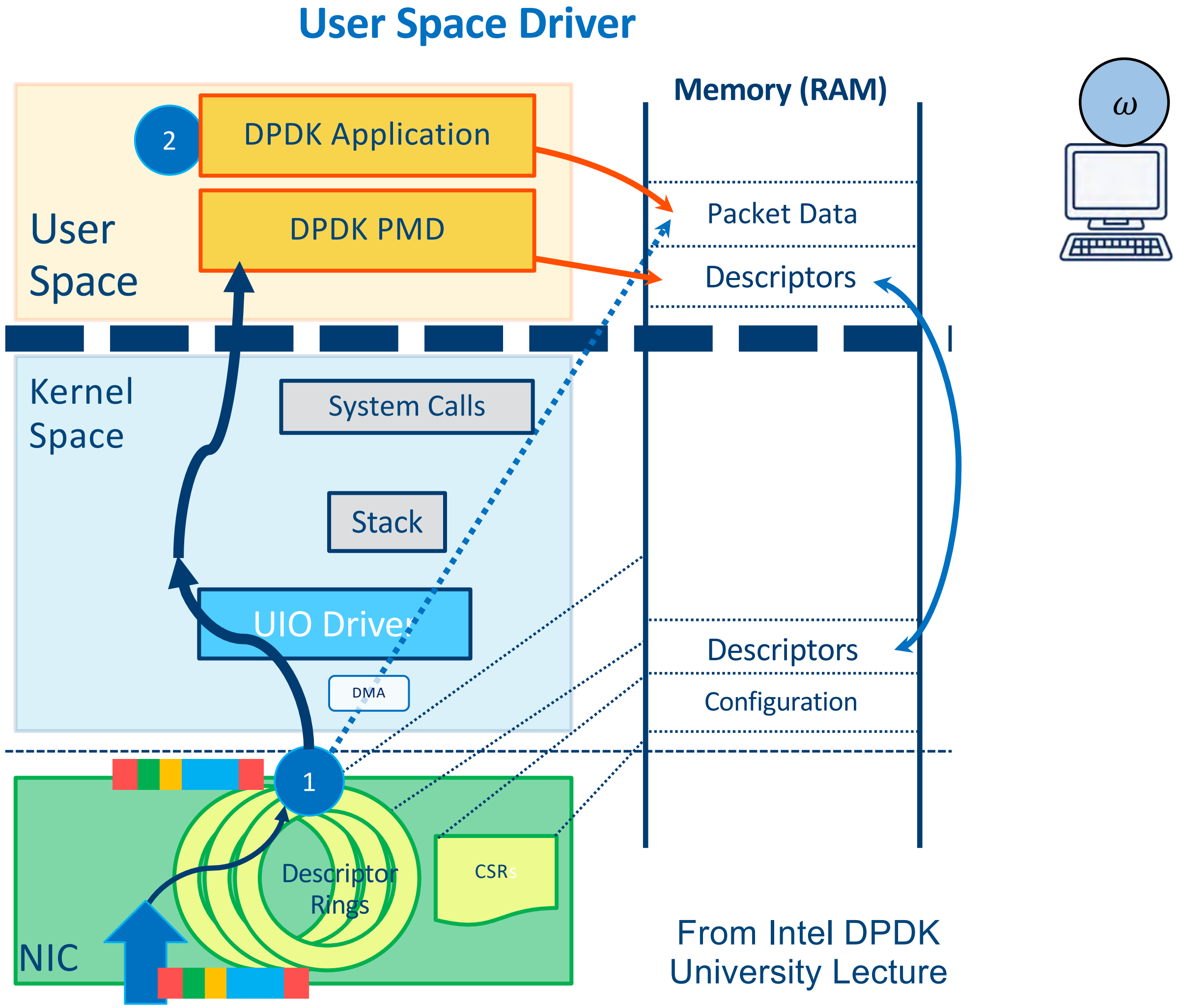


ASIC = „fixed silicon chip for special purpose, e.g. packet switching"

Performance Acceleration via INP

INP resources can reduce the time to move data, e.g.

- DPDK: circumvent OS
- OS Kernel: Enhance Communication Protocol
- NIC: process ahead of OS
- Switch : closer to producer/ consumer



From Intel DPDK University Lecture

Performance Acceleration via INP

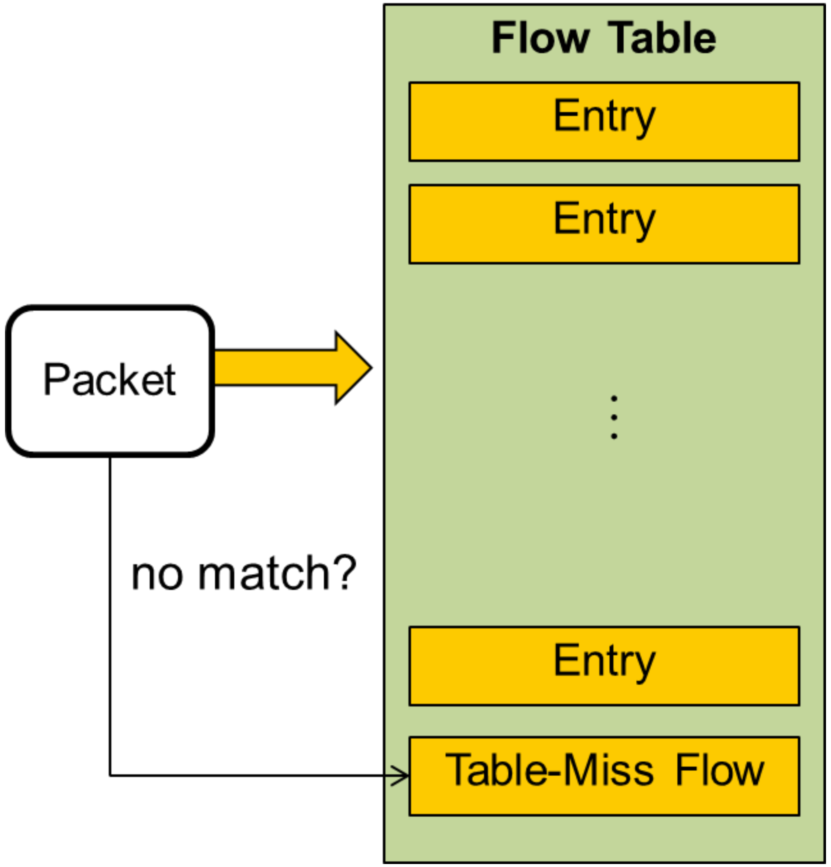
INP resources can reduce the time to move data, e.g.

- DPDK: circumvent OS
- OS Kernel: Enhance Communication Protocol
- NIC: process ahead of OS
- Switch : closer to producer/ consumer

INP resources can accelerate the processing time

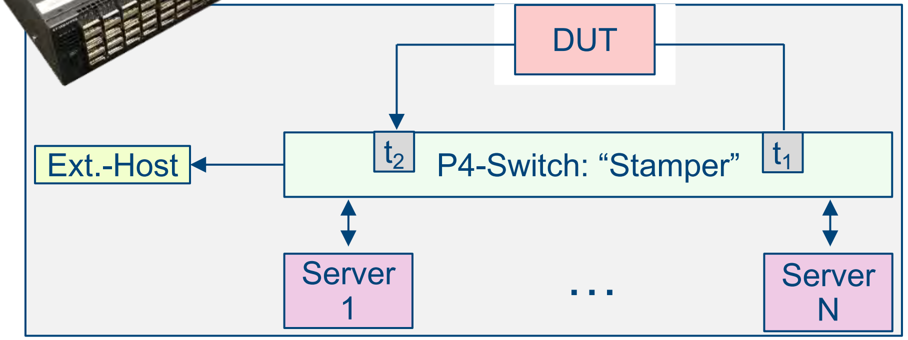
- Efficient Matching : TCAM
- Transformation and routing

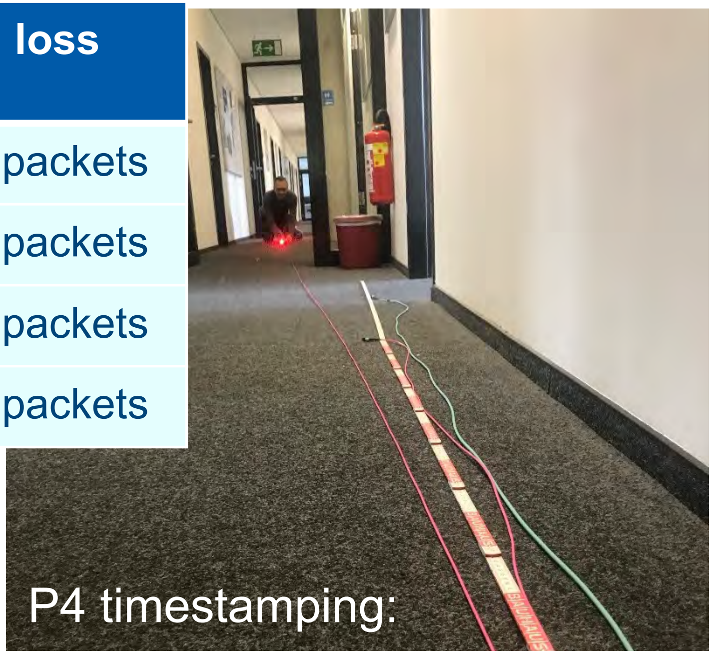
INP enables dynamic exchange of functionality

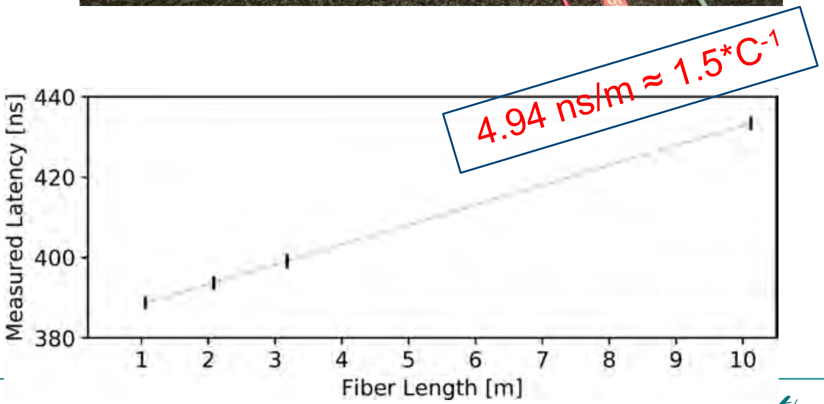


Is INP = Low Latency?
High Performance Packet Analytics in P4STA

	actual fiber length	avg. latency	std. dev.	loss
1 m	1.06 m	107.830 ns	1.46 ns	0 packets
2 m	2.08 m	112.850 ns	1.61 ns	0 packets
3 m	3.18 m	118.336 ns	1.52 ns	0 packets
10 m	10.12 m	152.883 ns	1.61 ns	0 packets







Kundel, Siegmund, Blendin, Rizk, Koldehofe**. P4STA: High Performance Packet Timestamping with Programmable Packet Processors.** In Proceedings of the IEEE/IFIP Network Operations and Management Symposium (NOMS 2020)

Challenges in using them for Real-time data analytics

Specific domain specific programming models

- OpenFlow, P4, Verilog

Breaking distribution transparency

- E.g., applications does not work on byte streams, but packets!

Increased heterogeneity

Headers may leak information on the packet content

Outline

Why low latency response?

The Bottleneck in Data Movements

In-Network Computing Technologies accelerating performance

Examples in the context of Distributed Event-Based Systems

Conclusion

Publish/Subscribe and Performance

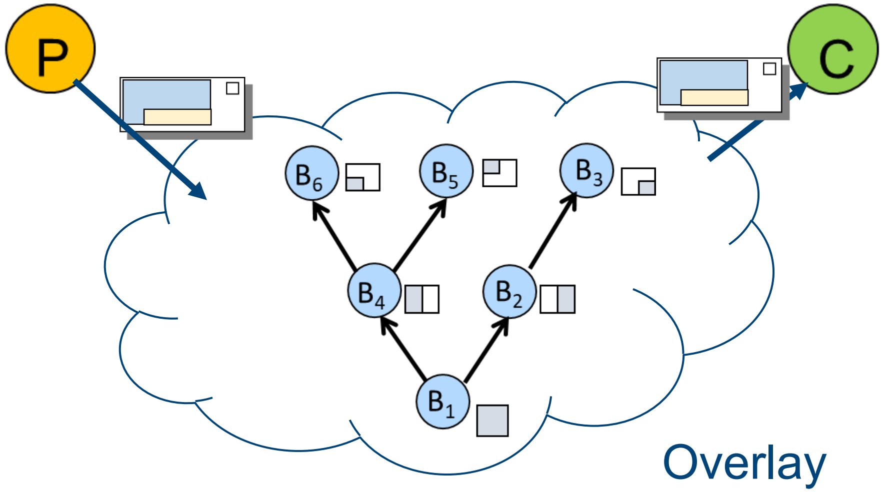
Fundamental programming paradigm for scalable distributed applications

Efficient distribution by means of overlays

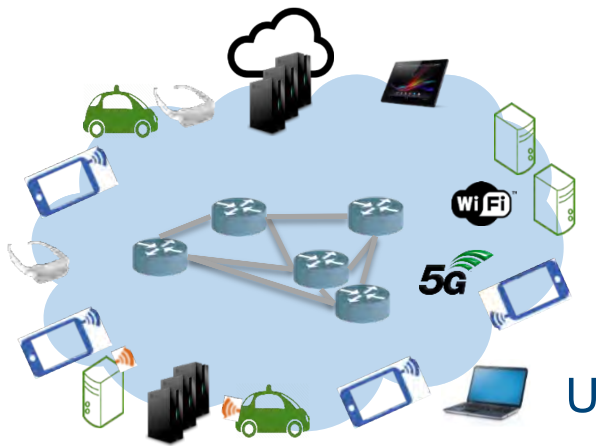
Bandwidth efficient overlays

BUT big performance gap

- Overlay
- Underlay



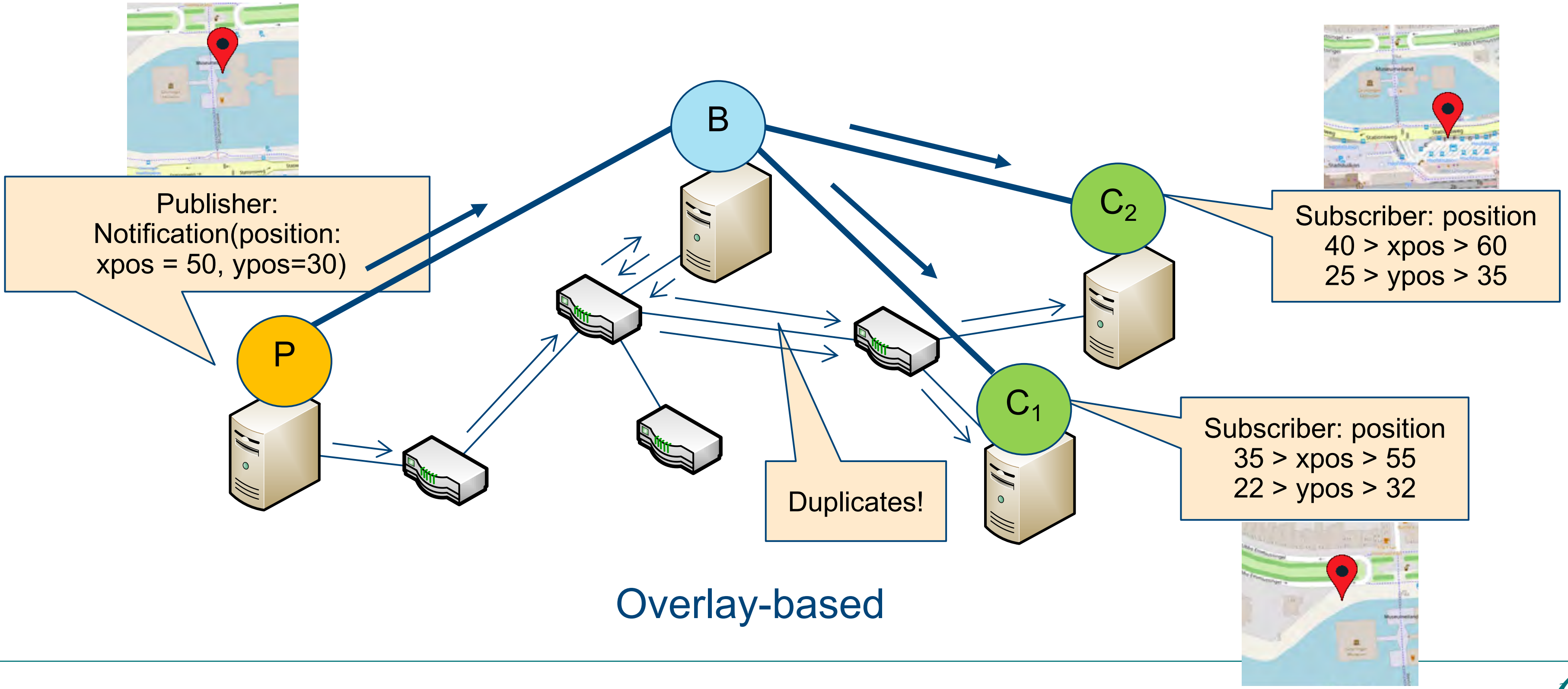
vs



High Performance Publish/Subscribe: Basic Idea

Reduce the overhead:

- Message duplications
- Matching subscriptions at the hardware



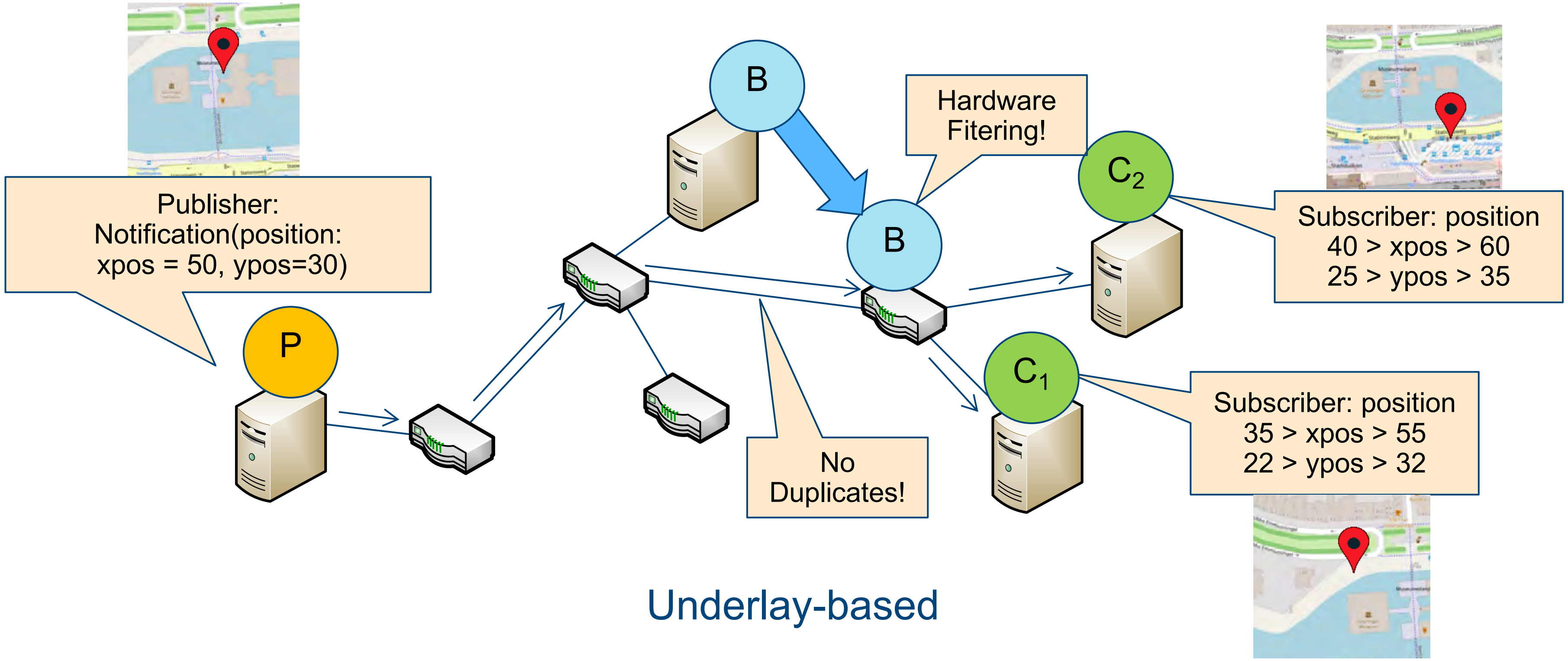
Overlay-based

Bowmik, Tariq, Koldehofe, Kohler, Dürr, Rothermel. **High Performance Publish/Subscribe Middleware in Software-defined Networks.** IEEE Transactions on Networking (ToN), 2016.

High Performance Publish/Subscribe: Basic Idea

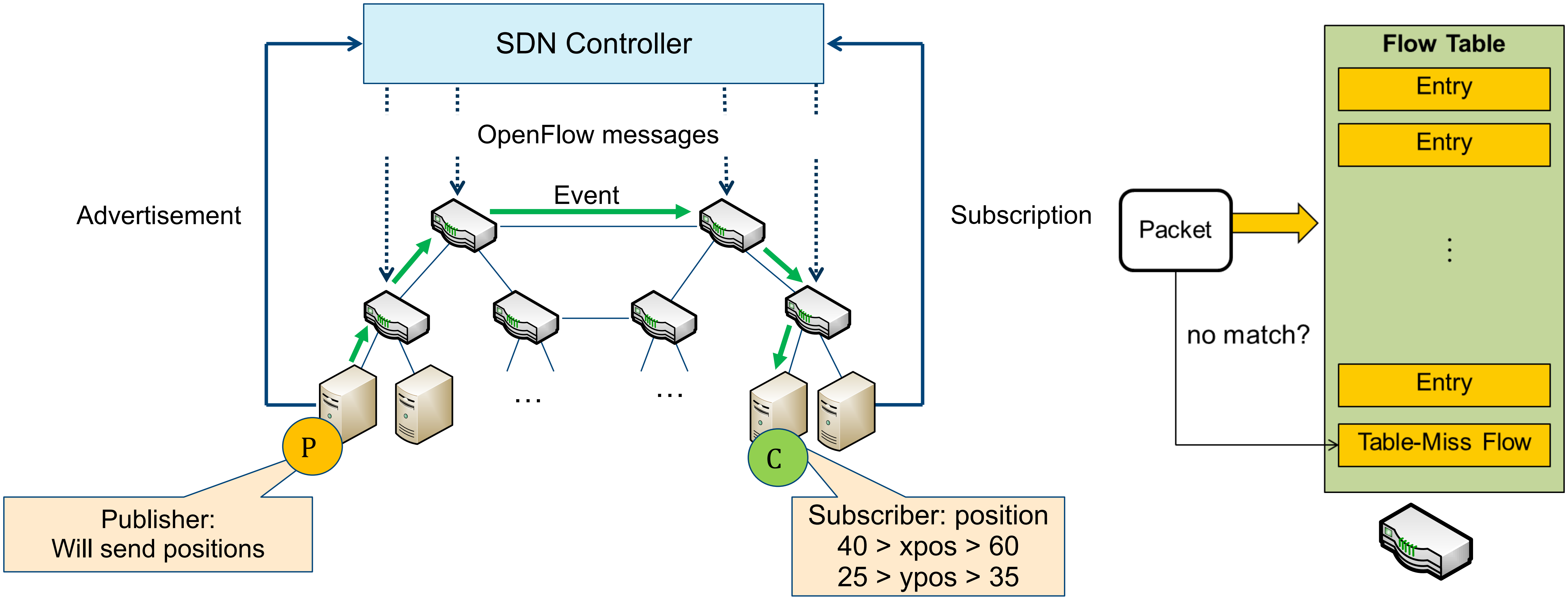
Reduce the overhead:

- Message duplications
- Matching subscriptions at the hardware



Underlay-based

SDN-based Publish/Subscribe Middleware



Bowmik, Tariq, Koldehofe, Kohler, Dürr, Rothermel. **High Performance Publish/Subscribe Middleware in Software-defined Networks.** IEEE Transactions on Networking (ToN), 2016.

Configuration Based on OpenFlow

Requires use of standard packet headers

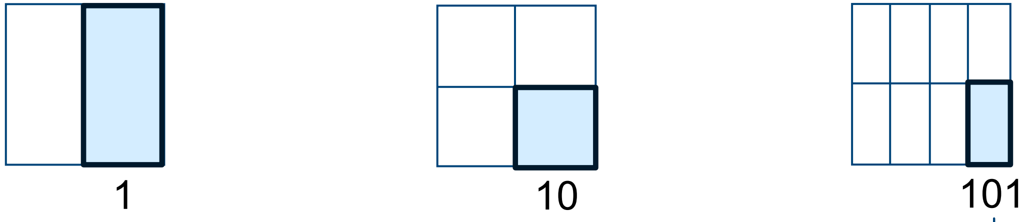
In port	VLAN ID	Ethernet			IP			TCP	
		SA	DA	Type	SA	DA	Prot	Src	Dst

?

Subscriber: position
40 > xpos > 60
25 > ypos > 35

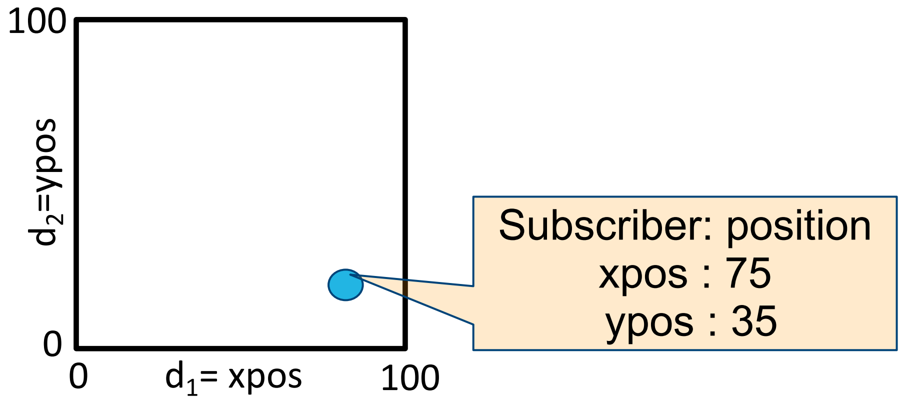
Therefore: Approximation of events in packet headers

1. **Generate binary representation based on spatial indexing**
2. **Map binary representation to IPv6 Multicast address**
   - Coexistence with other services

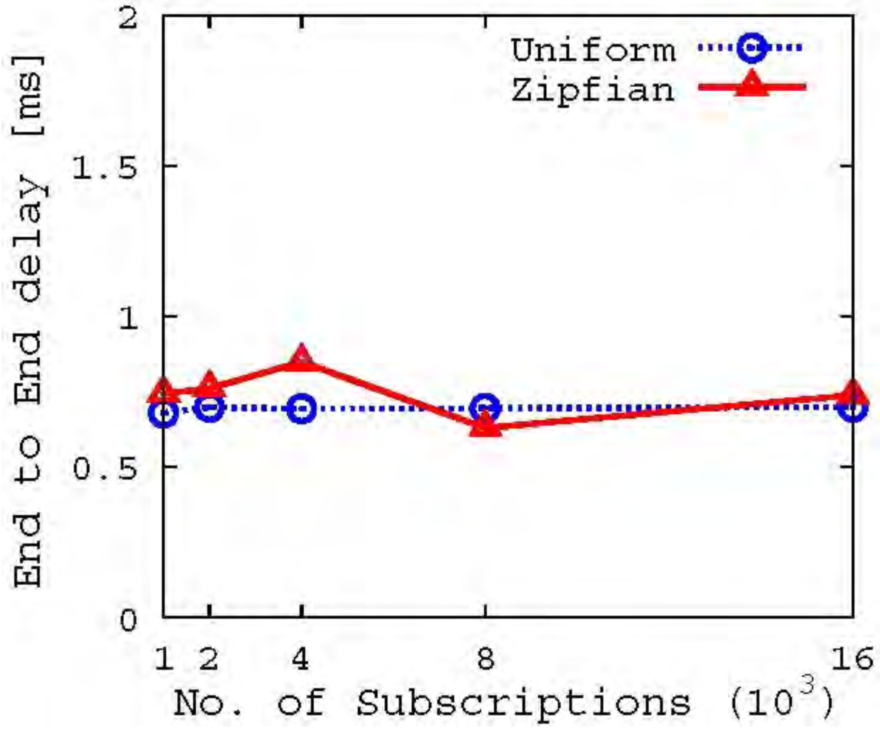


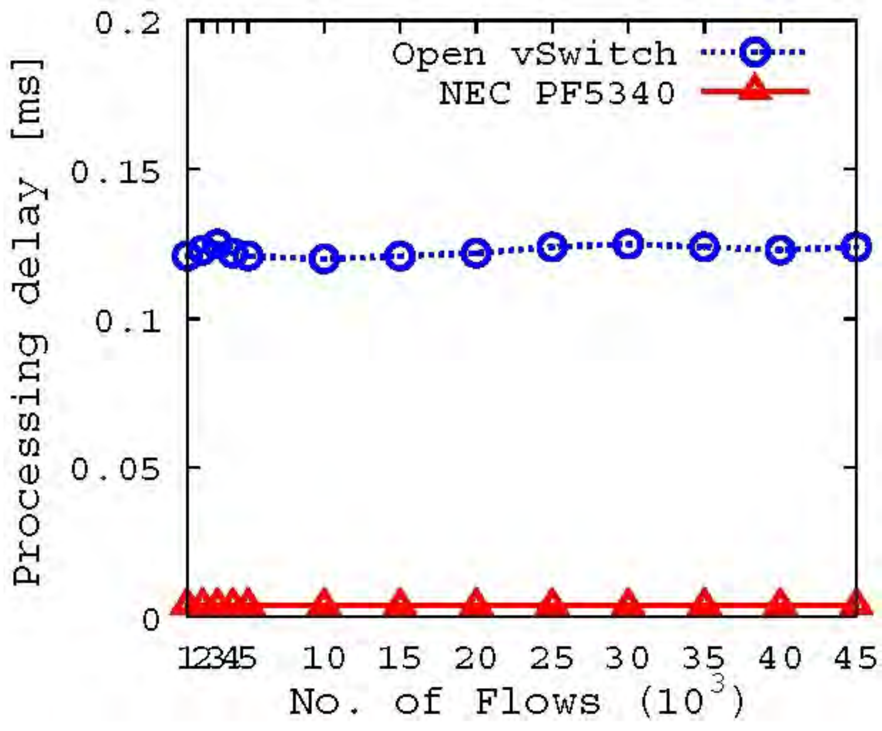
Mapping to IPv6 `ff0e:a000:*`

IP Prefix



Result: Forwarding performance





Hierarchical fat-tree topology

10 Open vSwitches and 8 end-hosts

10,000 events

Bowmik, Tariq, Koldehofe, Kohler, Dürr, Rothermel. **High Performance Publish/Subscribe Middleware in Software-defined Networks.** IEEE Transactions on Networking (ToN), 2016.

Properties

OpenFlow-based Management enables expressive subscription management

But requires from every publisher/ subscriber

- Understand the encoding

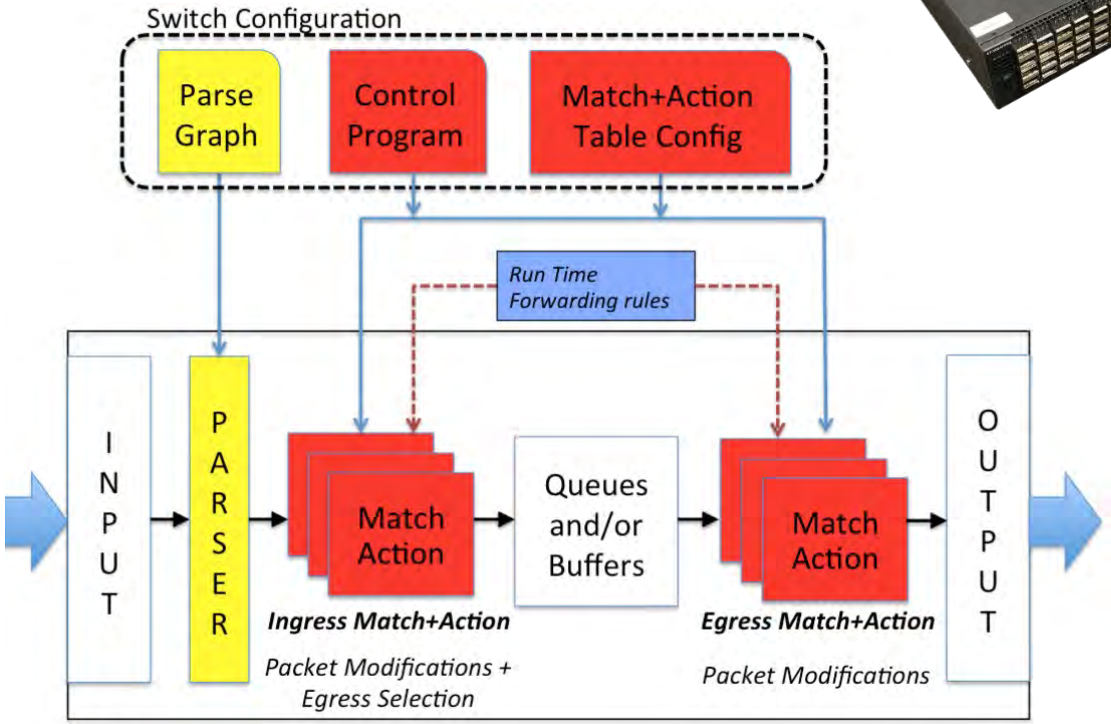
Relied on specific Header Fields!

But would work in general using a big field or mask

More Complex, state-full not considered!

Extending to P4 based INP

- P4 supports Programming Reconfigurable Match Action Pipeline
- Define own Protocol Headers
- Define Matching Operations for specific Header fields



Source p4.org

```
typedef bit<32> timestamp_t;
typedef bit<16> type_t;
typedef bit<8>  attribute_t;
typedef bit<8>  value_t;

header event_h {
    type_t type; /* example: weather. */
    timestamp_t  timestamp; /* event occurance time. */
    attribute_t  attribute; /* example: humidity, temperature. */
    value_t  value; /* example: 45% humidity and 23 degrees celsius temperature. */
}
```

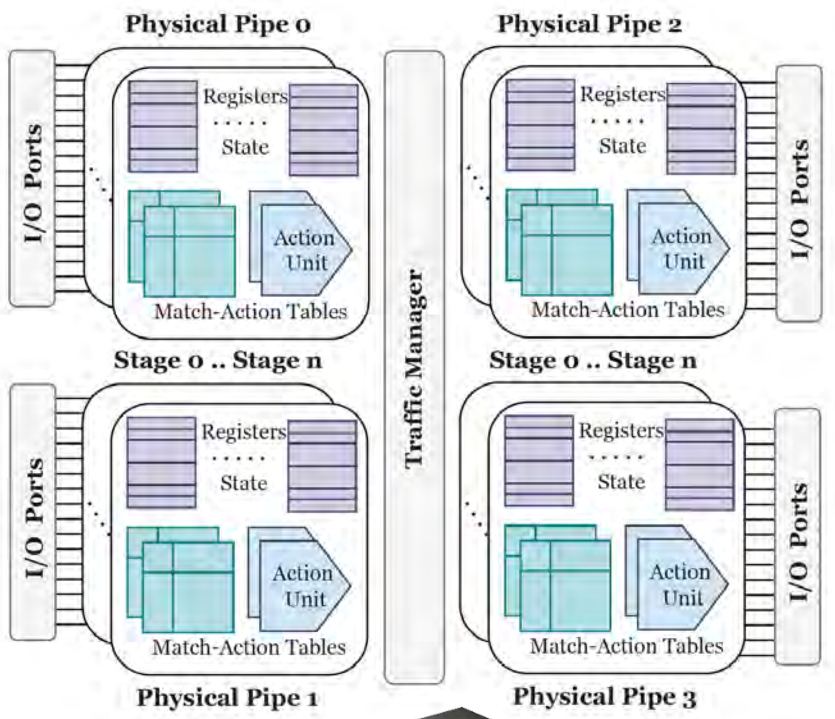
P4: Enhancing Stateful Operations

Limited Support for Stateful Operations

Many pitfalls:

- No sharing of registers between different stages of the pipeline
- Exclusive read or write operations
- Packet cannot iterate over all registers

However, can be used to model for specific platforms stateful Data Analytics!



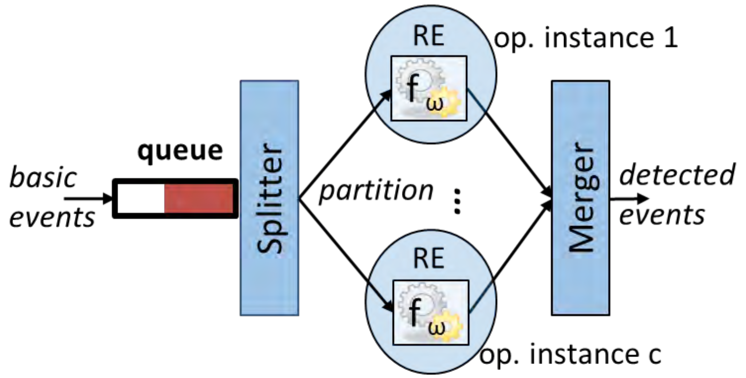
Kohler, Mayer, Dürr, Maaß, Bhowmik, and Rothermel. *P4CEP: Towards In-Network Complex Event Processing.* In Proceedings of the 2018 Morning Workshop on In-Network Computing (NetCompute '18, pp. 33–38. https://doi.org/10.1145/3229591.3229593

Examplary Use Case:
Enhancing parallel operator execution with P4

Operator Parallelization is a common method in Distributed Analytics

Splitter:

- Partition streams in independent processable windows
- Operator instances return results to the merger
- Merger coordinates streams



Processing rate of the splitter is the bottleneck in scaling operators

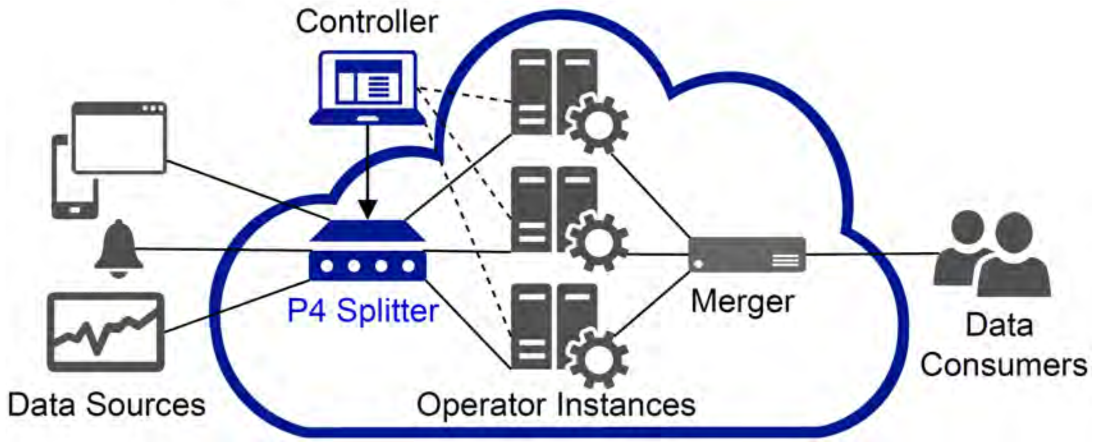
Can be done already on the path between producers and consumers

P4 Splitter: Window Operators

Key Challenges for INP Processing

1. **Huge dynamics and variety in partioning data streams**
   - E.g., expressing multiple distinct window semantics for operators
     - Time-based, Count based, …
   - Each requires a different way of partitioning

2. **Partitioning in line-rate with**
   - Match Action Logic
   - Registers state



How many streams can we support with one device?

Bochra Boughzala, Christoph Gärtner, and Boris Koldehofe. **Window-based Parallel Operator Execution with In-Network Computing.** Proceedings of the 16th ACM International Conference on Distributed and Event-based Systems (DEBS '22), pp. 91–96, ACM press.

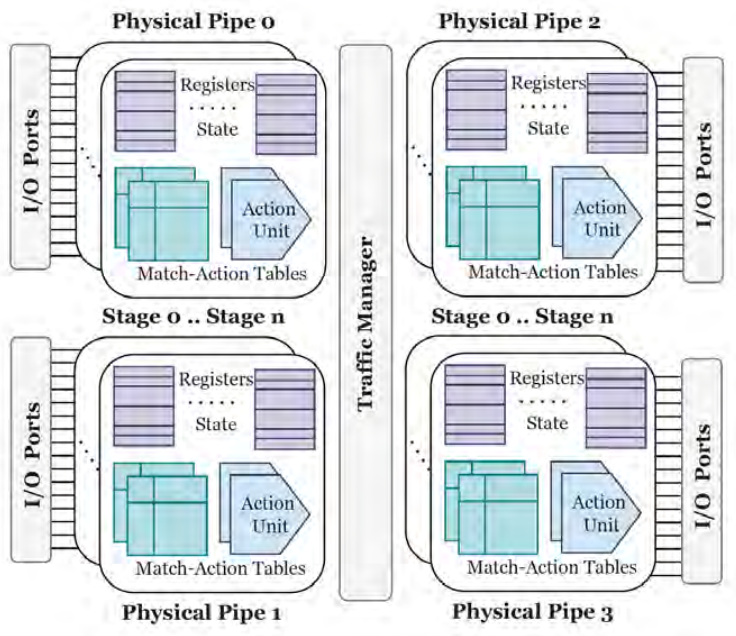
Some findings
Resource Usage

Resource Usage Determines Scalability

Comprises

- Stages, Tables, and Register Arrays
- Tofino1 has max 12 stages

Resource	CBTW	CBSW	TBTW	TBSW
Stages	6	6	7	8
Match Tables	12	12	18	20
Registers Arrays	3	2	6	7



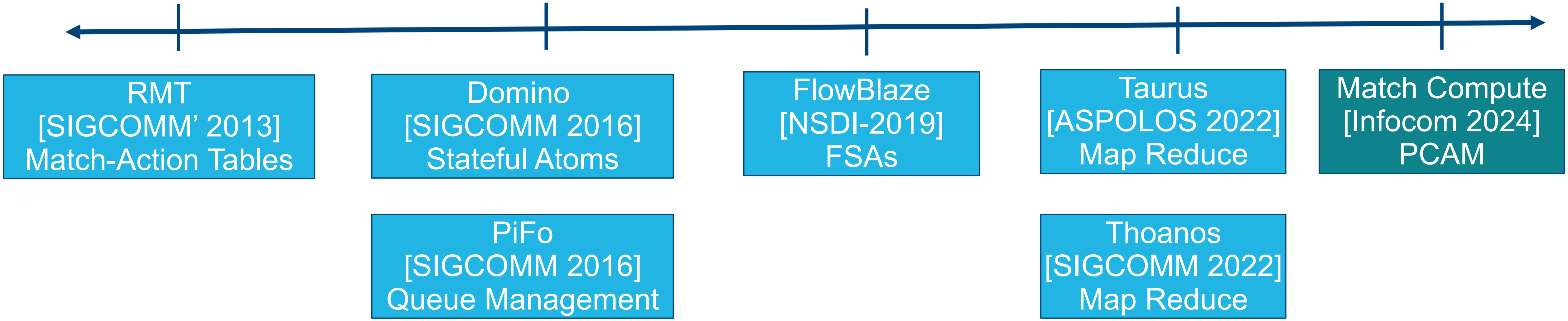
- Could deploy with line rate-performance
  - Count-Based windows : 457k operators, 286k concurrent streams
  - Time-based windows: 362k operators and up to 65k streams
- Bandwidth usage up to : 16 Gbit/s (more possible)
- packet processing delay : 2μs

Bochra Boughzala, Christoph Gärtner, and Boris Koldehofe. **Window-based Parallel Operator Execution with In-Network Computing.** Proceedings of the 16th ACM International Conference on Distributed and Event-based Systems (DEBS '22), pp. 91–96, ACM press.

Interesting Approaches in INP for Data Driven Applications

Networking Community is working on many abstractions for Stateful INP

Challenge: understand practicality and applicability in Middleware services



Adapted from Vishal Shrivastav presentation at SIGCOMM

But also very interesting work in distributed computing!

- E.g. " P4xos: Consensus as a Network Service", IEEE/ACM Transactions on Networking, 2020.

Everything on Performance?

Not really!

Data movements are the cause for high energy efficiency!

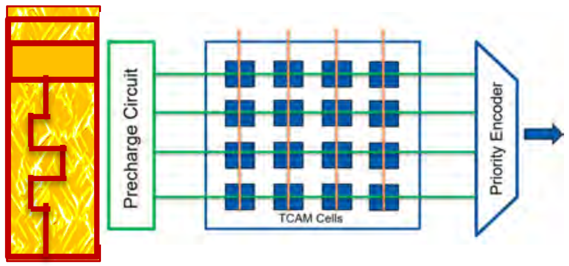
Moving to sustainable computing components!

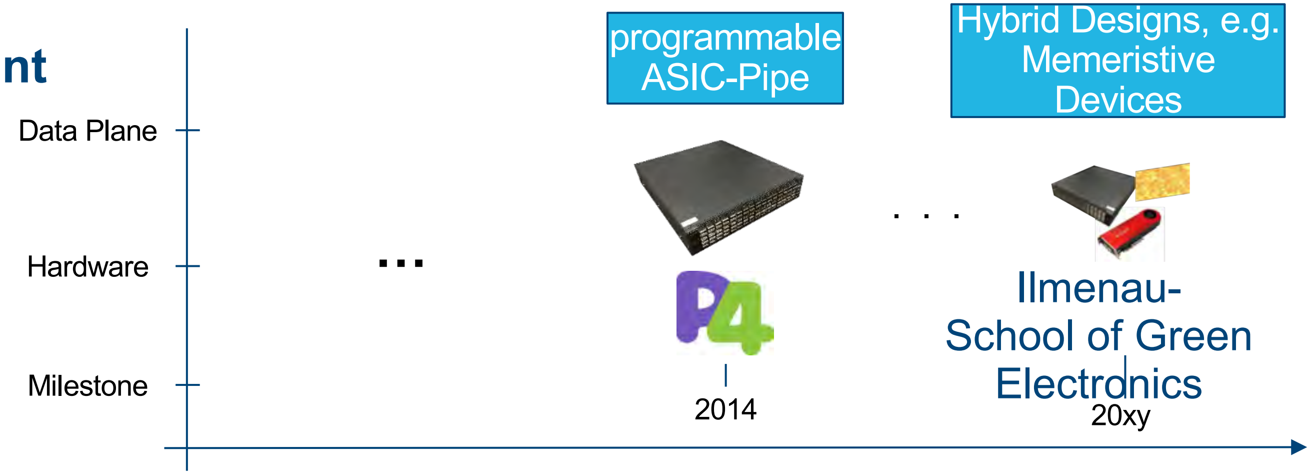
- Programmable ASIC for packet processing
- Very high bandwidth
- Limited flexibility

- Energy Efficient Switching?
- Computational Intelligence inside the network?

Recent example

- **TCA*m*M[CogniGron] and PCA*m*M:Energy Efficient Memristor-Based TCAM for Match-Action Processing**





ASIC = „fixed silicon chip for special purpose, e.g. packet switching"

Saleh, Goossens, Shu, Banerjee and Koldehofe. Analog In-Network Computing through Memristor-based Match-Compute Processing. To appear in Proceedings of the 43rd IEEE International Conference on Computer Communications, (INFOCOM 2024), IEEE, 2024.

Conclusion

Distributed Real-Time Analytics is a fundamental and challenging paradigm in the Internet of Things

Accelerators based on In-Network Computing

- Reduce performance bottlenecks
- Utilize the Distributed Infrastructure more efficient

Distributed systems mechanisms

- Flexible usage of heterogeneous resources
- No single mechanism fits them all

Current and Future Research:

- Better understanding of
  Distributed Computing + In-Network computing
- Energy-efficiency of In-Network Computing
- Performance of heterogeneous In-Network Computing

Many additional challenges

How to understand and train for performance?

How to perform dynamic adaptations at scale?

How to support real-time requirements?

How to support energy-efficient data processing operations?

How to model and ensure privacy for data streams?

SFB 1053

6G Campus Ilmenau

Ilmenau School of GREEN ELECTRONICS

COGNIGRON

Parrot

Questions

1. Ralf Kundel and Fridolin Siegmund and Rhaban Hark and Amr Rizk and Boris Koldehofe. Network Testing Utilizing Programmable Networking Hardware. IEEE Communications Magazine, 7 pages, IEEE 2022.

2. Bowmik, Tariq, Koldehofe, Kohler, Dürr, Rothermel. High Performance Publish/Subscribe Middleware in Software-defined Networks. IEEE Transactions on Networking (ToN), 2016.

3. Bochra Boughzala, Christoph Gärtner, and Boris Koldehofe. Window-based Parallel Operator Execution with In-Network Computing. Proceedings of the 16th ACM International Conference on Distributed and Event-based Systems (DEBS '22), pp. 91–96, ACM press.

4. Christoph Gärtner, Amr Rizk, Boris Koldehofe, René Guillaume, Ralf Kundel, Ralf Steinmetz. Fast incremental reconfiguration of dynamic time-sensitive networks at runtime. Computer Networks, vol. 224, 13 pages, Elsevier, 2023.

5. Pratyush Agnihotri, Boris Koldehofe, Paul Stiegele, Roman Heinrich, Carsten Binnig, Manisha Luthra. ZeroTune: Learned Zero-Shot Cost Model for Parallelism Tuning in Stream Processing. To Appear in Proceedings of the IEEE 40th International Conference on Data Engineering (ICDE 2024).

6. Saad Saleh, Anouk S. Goossens, Sunny Shu, Tamalika Banerjee and Boris Koldehofe. Analog In-Network Computing through Memristor-based Match-Compute Processing. To appear in Proceedings of the 43rd IEEE International Conference on Computer Communications, (INFOCOM 2024), IEEE, 2024.

Ilmenau School of GREEN ELECTRONICS

COGNIGRON

Parrot

SFB 1053

6G Campus Ilmenau

Prof. Dr. Boris Koldehofe

Technische Universität Ilmenau
Department of Computer Science and Automation
Distributed Systems and Operating Systems Group