Universität Stuttgart

Institute of Parallel and
Distributed Systems (IPVS)

Universitätsstraße 38
70569 Stuttgart
Germany

Time-sensitive Software-defined Network (TSSDN) for Real-time Applications

1. KuVS Fachgespräch "Network Softwarization" – From Research to Application

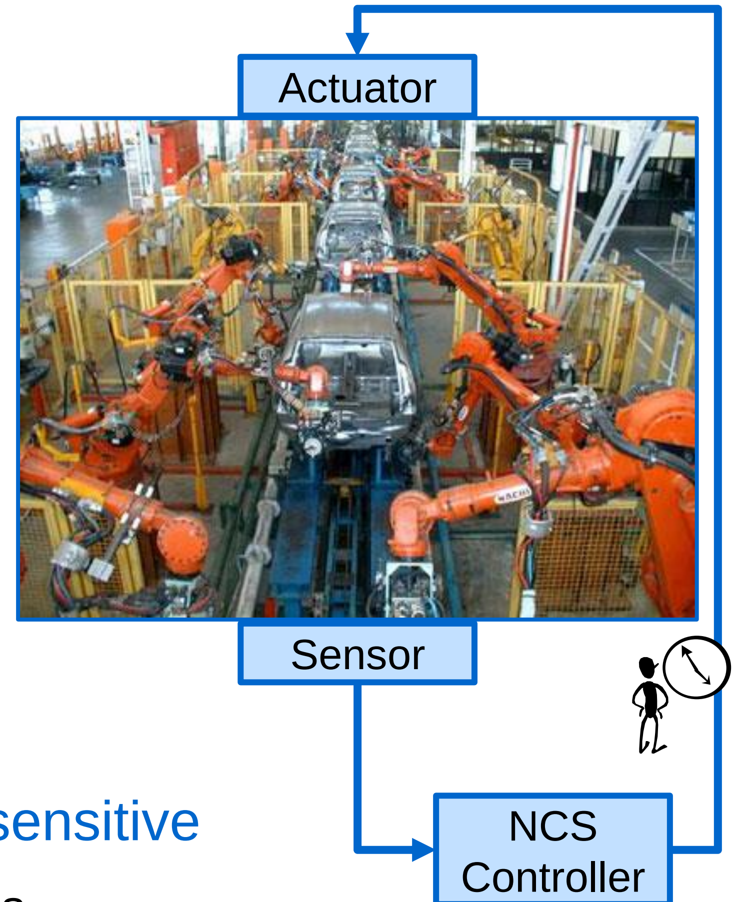
October 12-13, 2016 @ Tübingen, Germany

Naresh Ganesh Nayak, Frank Dürr, Kurt Rothermel

Cyber-physical Systems (CPS) & Networked Control Systems (NCS)

Cyber-physical System (CPS)

- Sensors, actuators, controllers, network
- Controlling physical processes
  ➔ Networked control systems (NCS)



Application domains:

- Manufacturing (Industry 4.0)
- Automotive
- Smart Grid
- Tele-robotics

Networked Control Systems are time-sensitive

- Latency down to tens of µs with jitter < 1 µs

Evolution of TSN Technologies

- Real-time communication in LANs so far based on field-bus technologies
  - Today, mostly Ethernet derivatives
  - EtherCAT, PROFIBUS, SERCOS III, etc.
- Now transition to standard Ethernet with IEEE 802 TSN
  - IEEE 802.1 Time-Sensitive Networking (TSN) Task Group
  - Credit-based shaper: Audio/video
  - Scheduled traffic with timed gates (IEEE 802.1Qbv): Deterministic bounds on delay and jitter

  ➔ One **converged** shared network for time-sensitive and non-time-sensitive traffic

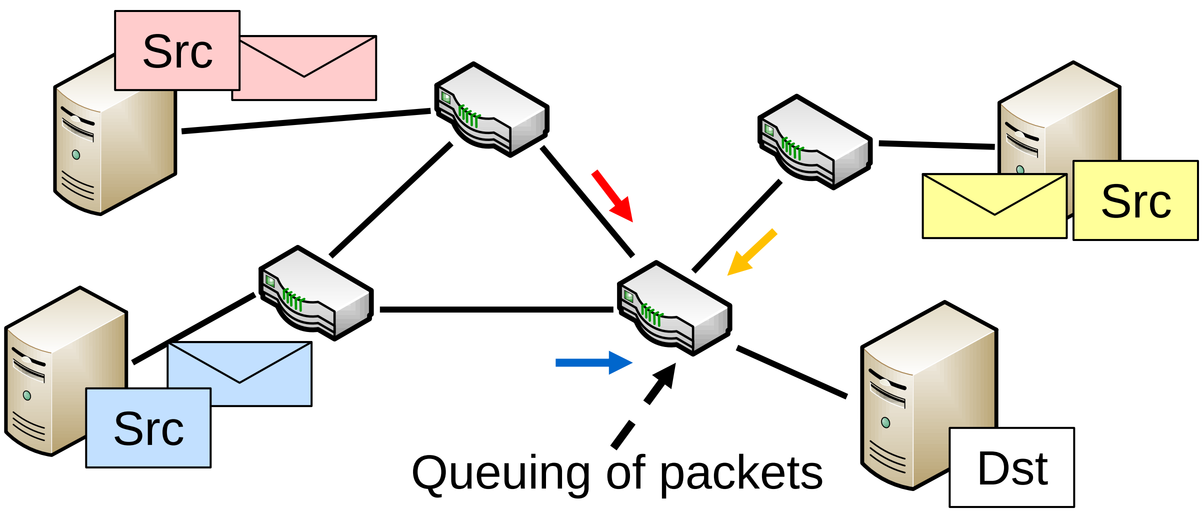
Here, we try to achieve deterministic bounds w/o IEEE 802 TSN!

Need for Transmission Scheduling in Ethernet

Network latency $\rightarrow \underbrace{t_{prop} + t_{proc} + t_{trans}}_{\text{constant \& practically bounded}} + \underbrace{t_{queue}}_{\text{variable \& unbounded}}$

➔ Must deterministically bound in-network queuing delays

One extreme approach taken in this paper: **eliminate** queuing delays



Schedule and **route** real-time traffic to avoid queuing

Contributions

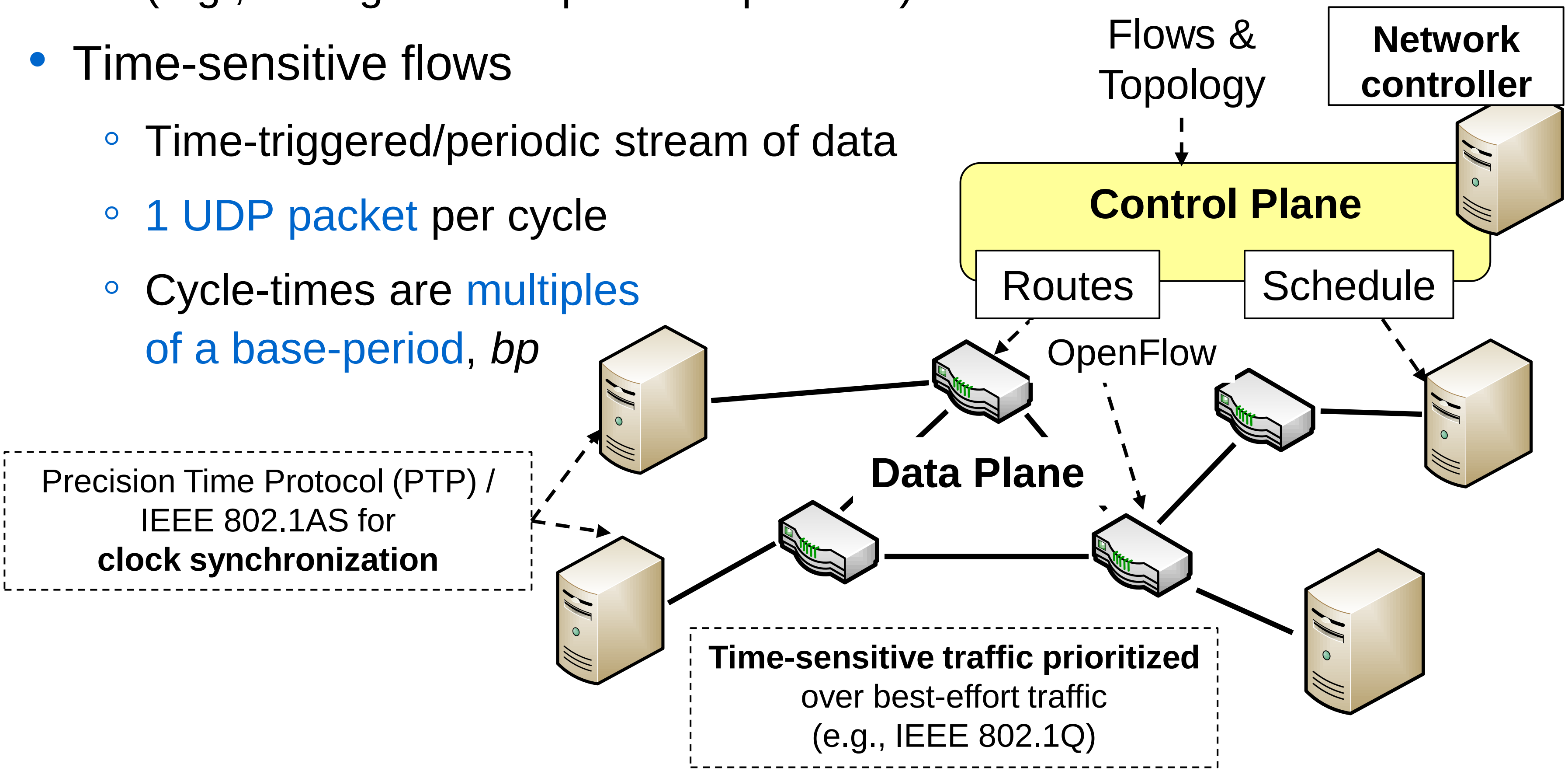
- Time-sensitive Software-defined Networking (TSSDN) Paradigm
  - **Converged network** for real-time traffic and best-effort traffic
  - **Logically centralized configuration** known from **software-defined networking (SDN)** to configure routes and schedules
  - **Deterministic delay/jitter bounds** with **minimal network support**
    - Plain “old” IEEE 802 Ethernet w/o IEEE 802.1 TSN
    - Time-triggered traffic scheduled on the network edge (hosts)
- Combined routing and scheduling configuration algorithms
  - **Integer Linear Programs (ILP)** for **exact** and **heuristic solutions**
- Mechanisms for schedule adherence by end systems (hosts)
  - **High-throughput packet processing framework (DPDK) minimizing** network stack latency on hosts

Agenda

- Introduction
- Time-sensitive Software-defined Network
  - System Model
  - Problem Statement
- Scheduling Algorithms
- Evaluations
- Conclusion & Future Work

System Model – TSSDN

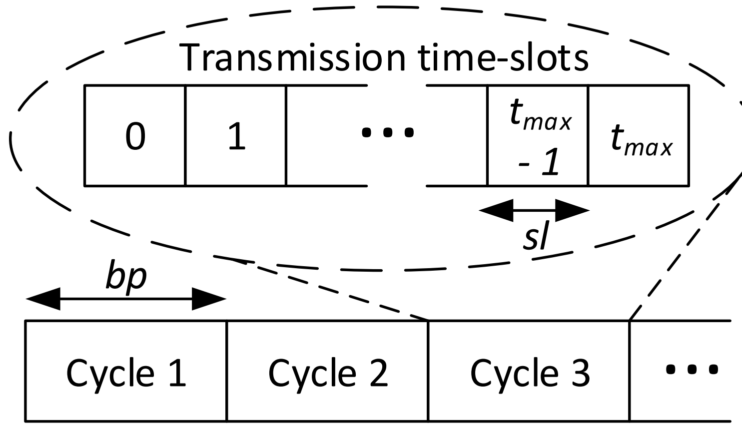
- Logically centralized network-controller (TSSDN controller)
  - Computes transmission schedules and routes
  - Configures schedules on hosts and forwarding tables of switches (e.g., through SDN/OpenFlow protocol)
- Time-sensitive flows
  - Time-triggered/periodic stream of data
  - 1 UDP packet per cycle
  - Cycle-times are multiples of a base-period, *bp*

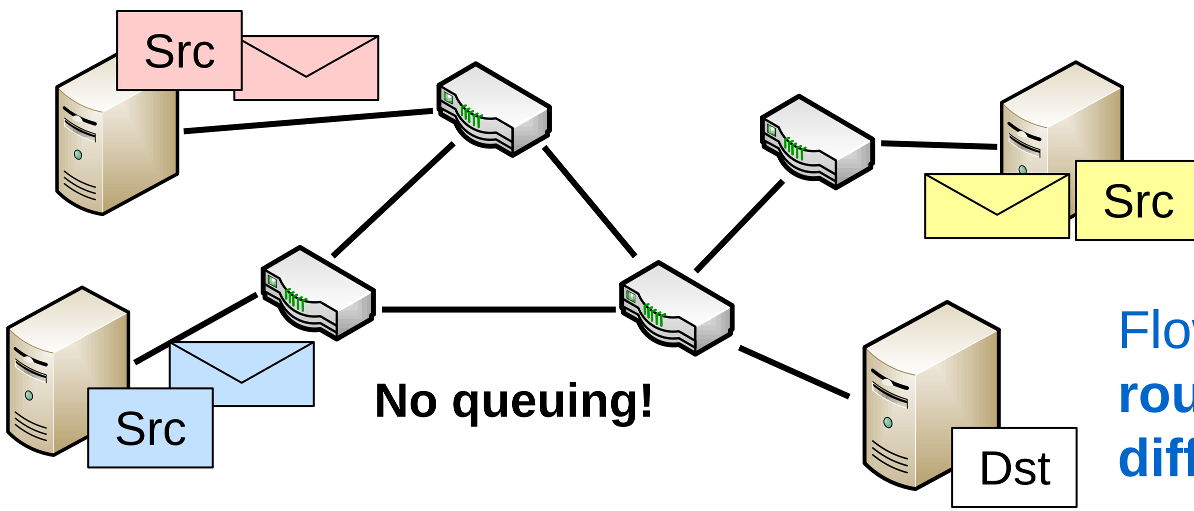


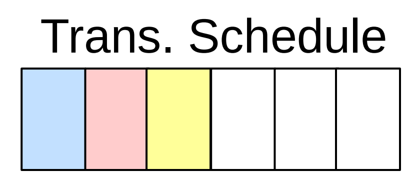
Scheduling Model – TSSDN

Cyclic Schedules

- Length = Base-period, *bp*
- Consists of smaller time-slots
- Length of time-slot ≥ time to send MTU packet over longest n/w path (≤ 7 hops)





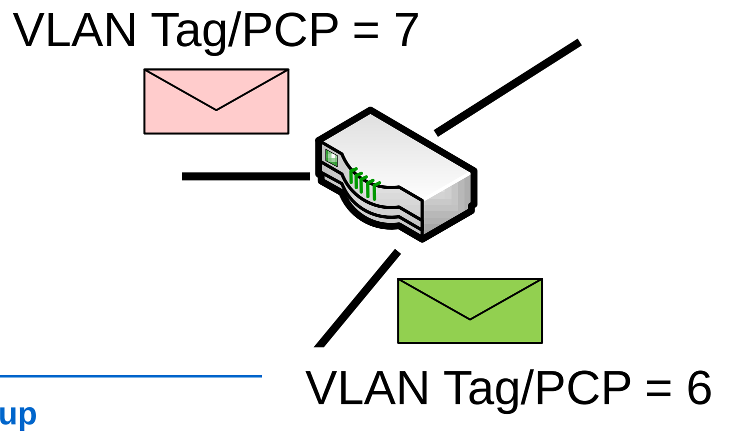


Flows with **overlapping routes** are assigned **different** time-slots

Scheduling Model – TSSDN

TSSDN – One converged network for real-time & best-effort traffic

- Use of IEEE 802.1Q priority classes to isolate real-time traffic from best-effort traffic
  - Real-time traffic uses VLAN tags with PCP = 7
  - Best-effort traffic uses VLAN tags with PCP < 7
  - Frame pre-emption (IEEE 802.1Qbu) to improve performance



Problem Statement

Objective: Maximize the number of time-sensitive flows accommodated (scheduled and routed) in the network

Inputs:

- Flow specifications (src, dst, cycle-time)
- Network topology

Outputs:

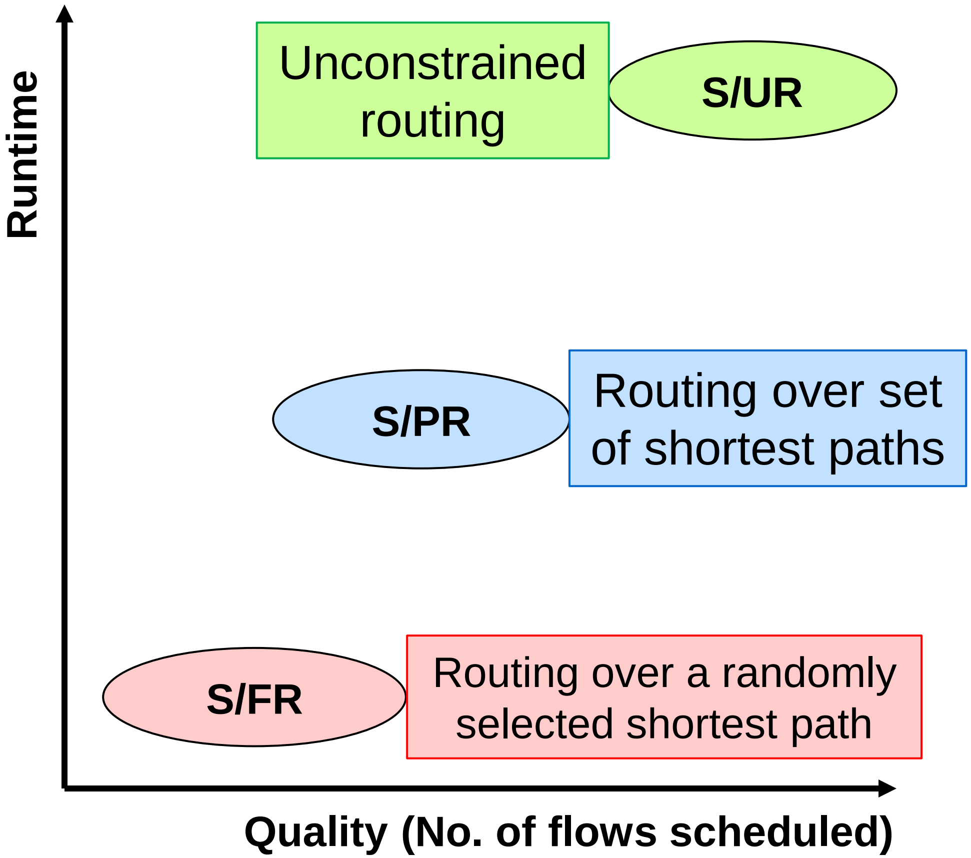
- Routes for flows
- Time-slots for flows

Constraint:

Flows with overlapping routes assigned different time-slots

Complexity – NP-hard

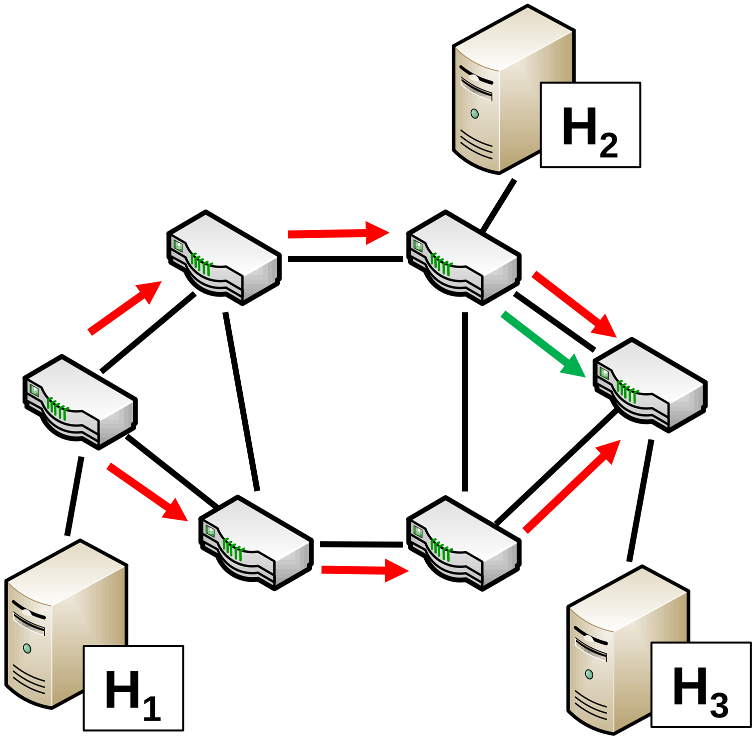
Overview of Solutions



Flows To Schedule

$F_1: H_1 \rightarrow H_3$

$F_2: H_2 \rightarrow H_3$



Scheduling with Unconstrained Routing (S/UR)

- Constraint 1: Assign each flow to at most one time-slot

$$\sum_{t \in T} ST(f,t) \leq 1 \qquad \forall f \in F$$

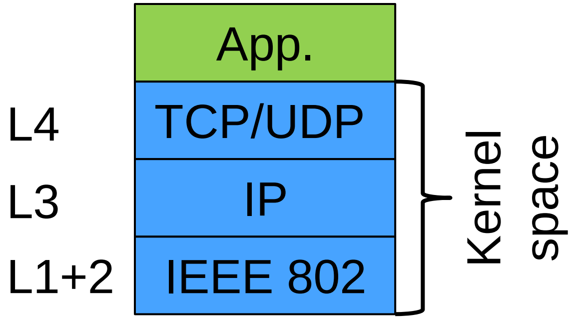
- Constraint 2: During any time-slot no link should be allocated to more than one flow

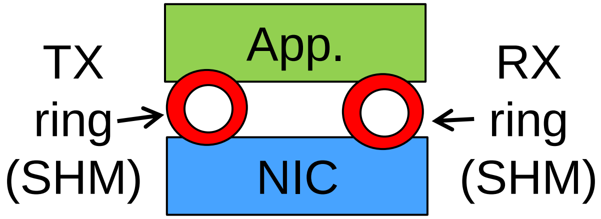
$$\sum_{f \in F} ST(f,t) \cdot SL(f,e) \leq 1 \qquad \forall t \in T; \forall e \in E$$

Non-linear constraint
Can be transformed to linear constraints using auxiliary variables. Refer paper for details

Improving Schedule Adherence

- **Requirement:** hosts must precisely adhere to their time-slots
  - Inject packet at the scheduled time of the assigned slot
- Linux network stack induces significant jitter (see experiments)
- **Approach: Kernel-bypassing**
  - Packet processing frameworks: Intel's Data Plane Development Kit (DPDK), netmap, etc.
  - Shared memory access from user-space to TX/RX rings of NIC (network interface ctrl.)
    - Raw packet processing by application (library)
    - CPU core dedicated to packet processing

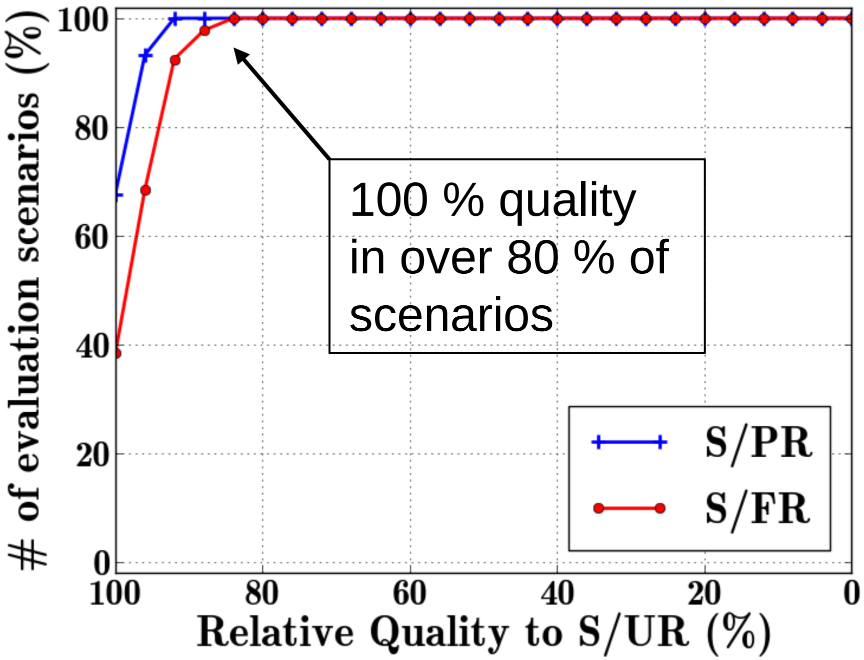




Evaluations – Quality of the ILP Formulations

Setup:

- Random network (24 host, 6 switches)
- Random sources & destinations
- Different topologies (random, scale-free, geographic/Waxman)
- Commodity multi-core server running CPLEX solver
- 3-5 time slots (challenging!)



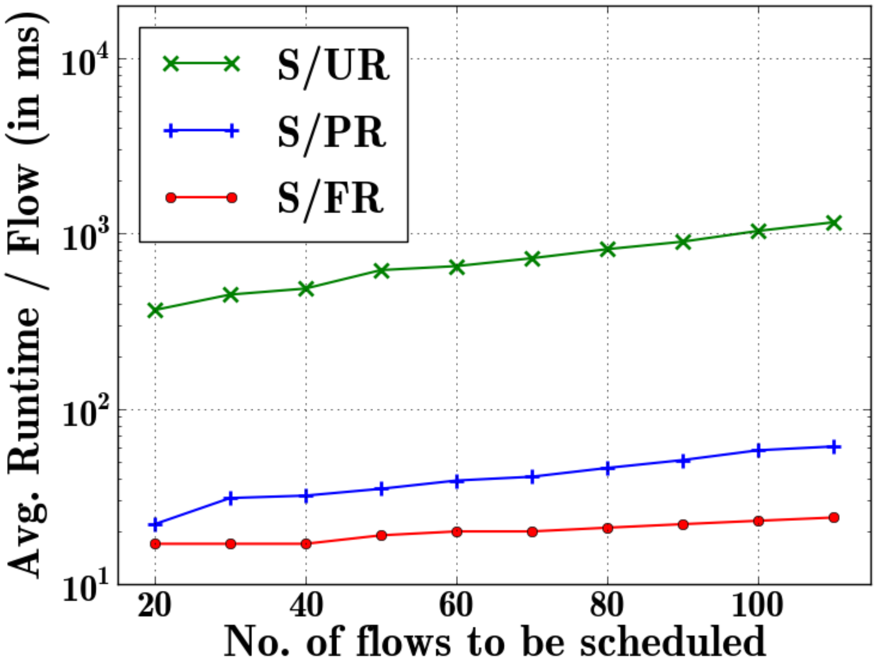
Performance metric:

- Ratio of number of flows scheduled flows by the heuristic vs. S/UR

Evaluations – Runtimes for ILP Formulations

Setup:

- High performance machine
  - 2 × 8 processor cores
  - 128 GB RAM
- Small Erdős–Rényi topology (24 hosts and 6 switches)
- Varying number of (random) flows

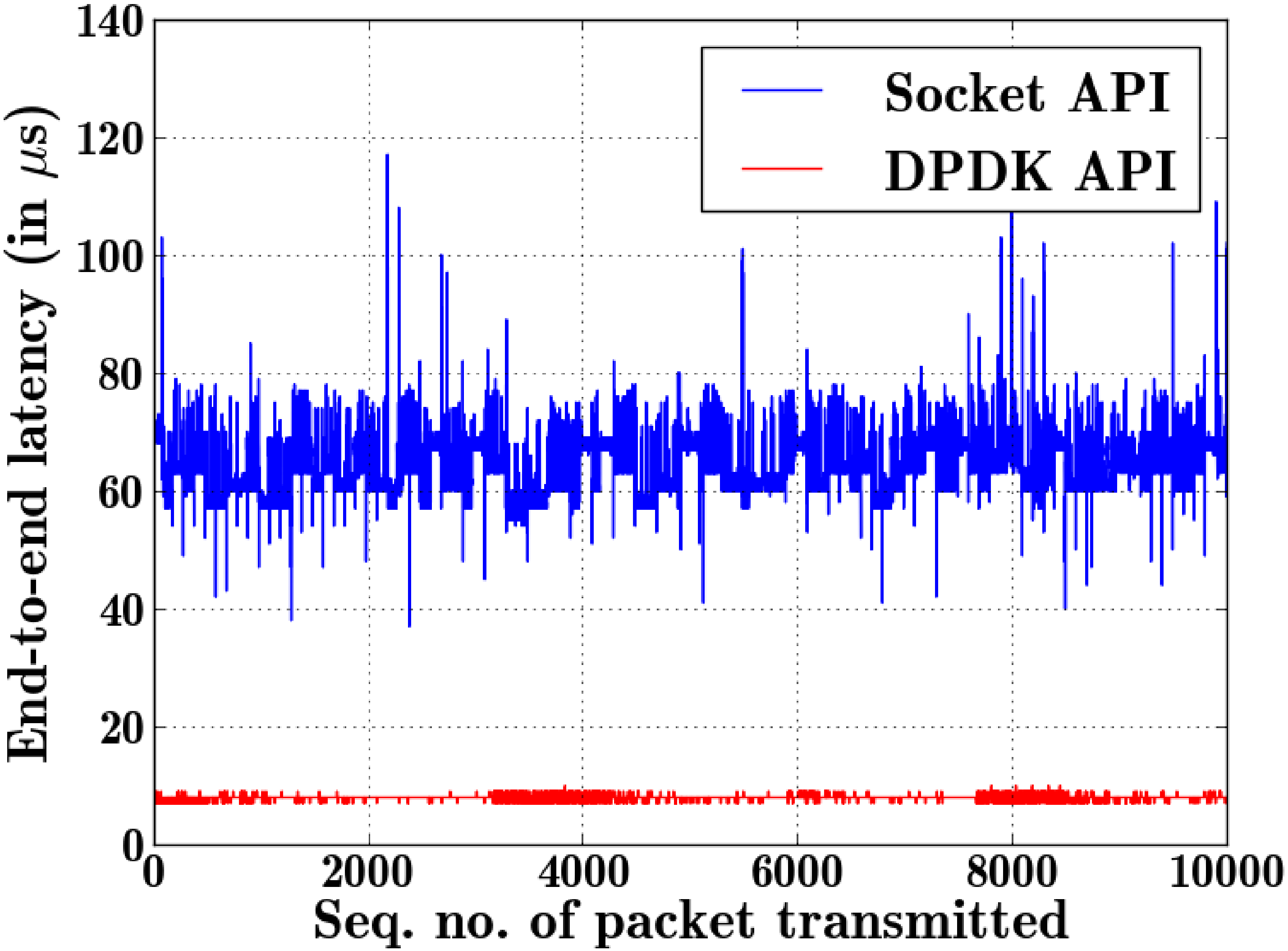


Performance metric:

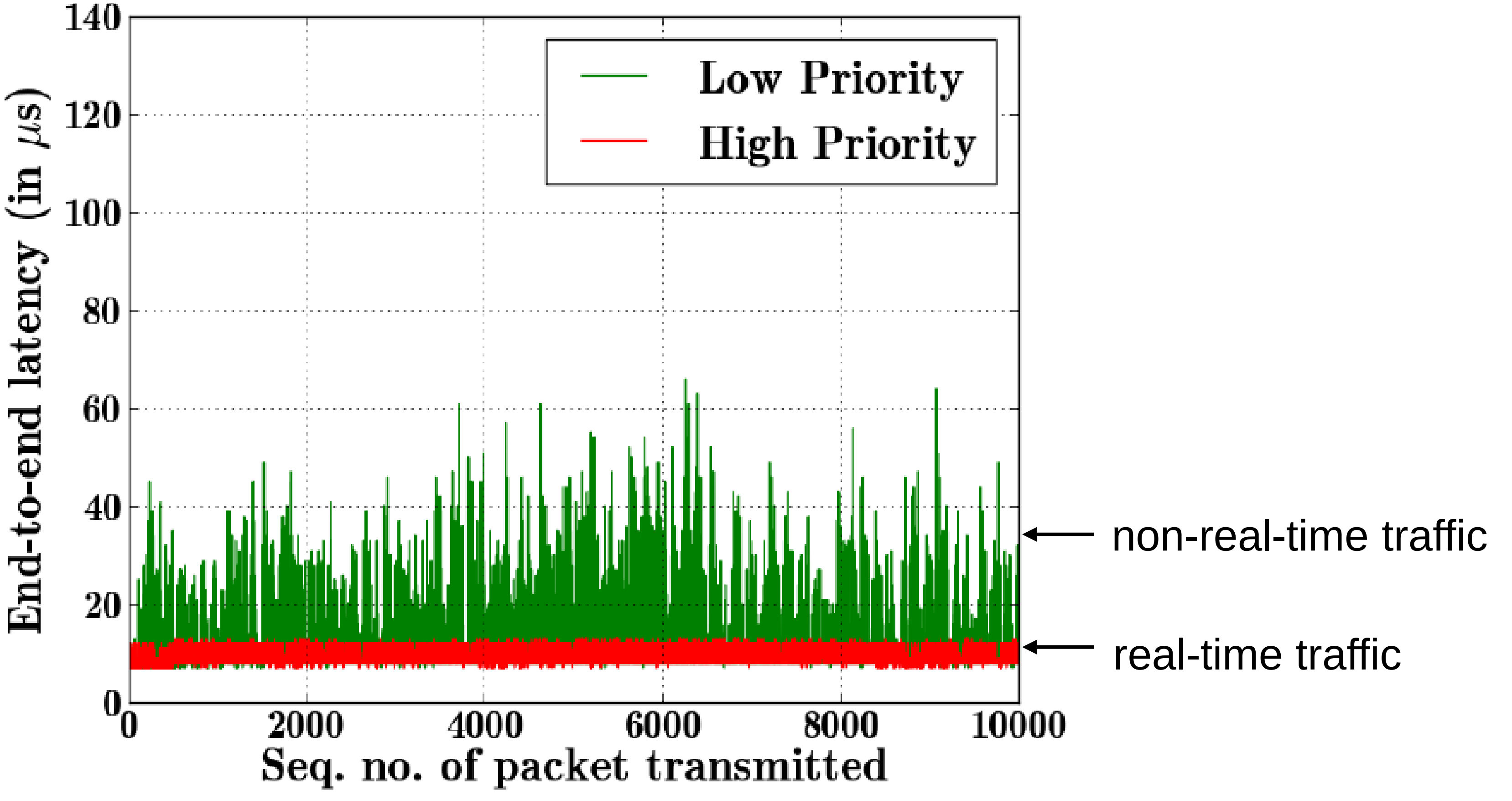
- Average time to schedule a flow

Conclusion: S/UR has runtimes which are an order of magnitude higher than S/PR and S/FR.

Schedule Adherence in TSSDN (2)



Effectiveness of Scheduling



Related Work

- Steiner & Craciunas – RTSS 2010, RTNS 2014
  - SMT-Based schedule synthesis for time-triggered multi-hop networks
  - Fine-grained link schedule synthesis; a priori routing
  - Limited queuing delays allowed
  - Extensions for combined task & network scheduling
- Hanzalek et al. – Transactions on Industrial Informatics 2010
  - Profinet IO IRT Message Scheduling with Temporal Constraints
  - Fine-grained link schedule synthesis; a priori routing
- Dürr et al. – RTNS 2016
  - Mapping to No-wait Job-shop Scheduling Problem
  - Requires more network support (IEEE 802.1Qbv); a priori routing

Summary and Future Work

Summary

- Introduced Time-sensitive Software-defined Network (TSSDN)
  - Logically centralized configuration of routes and schedules
- Solutions for combined routing and scheduling problem
  - ILPs for exact and heuristic solutions
  - Eliminates non-deterministic queueing delay
- Packet processing frameworks to minimize network stack delay

Future Work

- Incremental online scheduling algorithms for TSSDN
- Use RTOS (Linux RTPREEMPT) on end-systems for deterministic task execution

Contact

http://www.d-sdn.de/

Contact

Naresh Nayak

- naresh.nayak@ipvs.uni-stuttgart.de

Frank Dürr

- frank.duerr@ipvs.uni-stuttgart.de

Backup Slides

Scheduling with Unconstrained Routing (S/UR)

- Parameters
  - Hosts, H & Switches, S
  - Nodes, $V \equiv H \cup S$
  - Links, $E \subseteq V \times V$
  - Flows, F
    - $\forall f \in F;\ \ f \equiv (src, dst)$
      src is source of the flow & dst is the destination of the flow
  - Time-slots, $T \equiv \{0, 1, \dots,\ n-1\}$
- Variables
  - Route, SL: $(F \times T) \rightarrow \{0, 1\}$
  - Schedule, ST: $(F \times E) \rightarrow \{0,1\}$

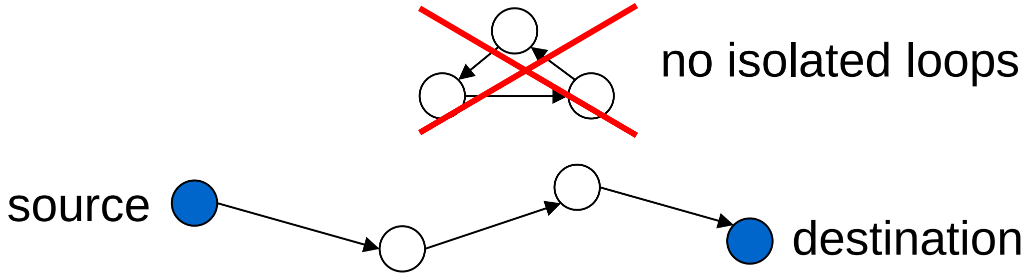
Scheduling with Unconstrained Routing (S/UR)

- ILP Objective

$$max \sum_{(f,t)\in F\times T} ST(f,t) + \frac{1}{|E|\cdot|TS|+1} \times \sum_{(f,e)\in F\times E} SL(f,e)$$

Maximize the number of scheduled flows

If number of flows is equal for two solutions: Prefer solution with lesser assigned links



Scheduling with Unconstrained Routing (S/UR)

- Constraint 1: Assign each flow to at most one time-slot

$$\sum_{t \in T} ST(f,t) \leq 1 \qquad \forall f \in F$$

- Constraint 2: During any time-slot no link should be allocated to more than one flow

$$\sum_{f \in F} ST(f,t) \cdot SL(f,e) \leq 1 \qquad \forall t \in T; \forall e \in E$$

Non-linear constraint
Can be transformed to linear constraints using auxiliary variables. Refer paper for details

Scheduling with Unconstrained Routing (S/UR)

- Constraint 3: Routing constraints
  - For each flow, the source node uses exactly one outgoing link

$$\sum_{e \in out(src(f))} SL(f,e) = 1 \qquad \forall f \in F$$

$$\sum_{e \in in(src(f))} SL(f,e) = 0 \qquad \forall f \in F$$

  - For each flow, the destination node uses exactly one incoming link

$$\sum_{e \in out(dst(f))} SL(f,e) = 0 \qquad \forall f \in F$$

$$\sum_{e \in in(dst(f))} SL(f,e) = 1 \qquad \forall f \in F$$

Scheduling with Unconstrained Routing (S/UR)

- Constraint 3: Routing constraints
  - For all flows, inner nodes have equal number of outgoing and incoming links

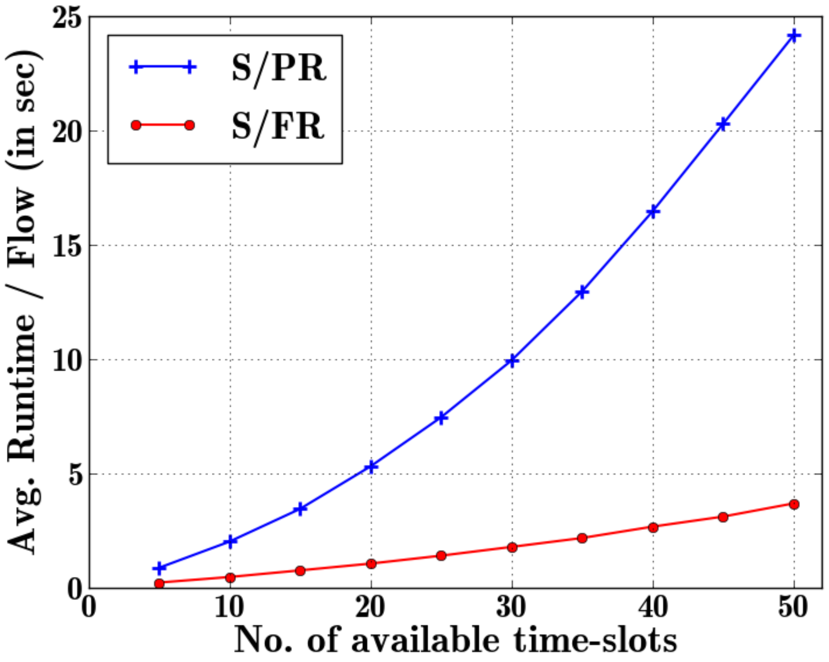
$$\sum_{e \in in(n)} SL(f,e) = \sum_{e \in out(n)} SL(f,e)$$

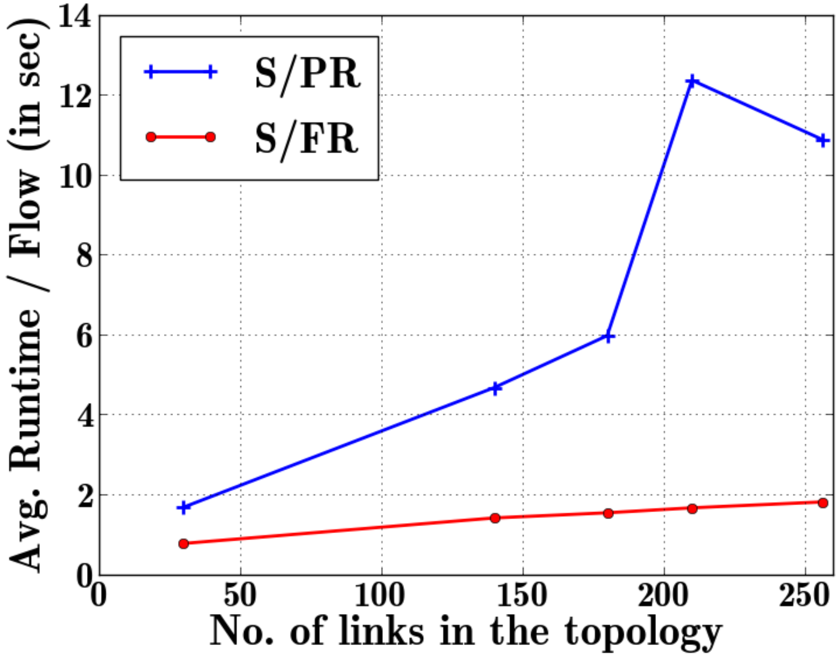
$$\forall f \in F, \forall n \in V \setminus \{src(n), dst(n)\}$$

Detailed ILP formulations of S/PR and S/FR, see paper

Evaluations – Runtimes for ILP Formulations

Results on a commodity machine with 2 processor cores





Setup: Scheduling 300 flows on a Waxman topology (200 hosts and 10 switches)

Setup: Scheduling 100 flows on varying topologies with 50 time-slots

Schedule Adherence in TSSDN (1)

Setup:

- Experiments in real system
- Two 10 Gbps switches (Edge Core AS5712-54X)
- 10 Linux hosts
  - Host clocks synchronized through PTP

- Measures end-to-end latency:
  1. Linux network stack
  2. DPDK (kernel-bypassing)

No cross-traffic