Talker Scheduled Traffic Support for Ultra Low Latency

Johannes Specht, University of Duisburg-Essen

Markus Jochim, General Motors Company

Introduction

- The Time Aware Shaper (TAS) enables low latency and meets e.g. Automotive Requirements
- TAS requires End Stations and Bridges to operate in synchronized TDMA-like schedules
- This slide deck proposes Talker Scheduled Traffic Support (TSTS) as an alternative to TAS in bridges:
  - TSTS can simplify the scheduled traffic concept, …
  - … decrease complexity in bridges and …
  - … still meet ultra low latency Automotive Requirements

Assumptions

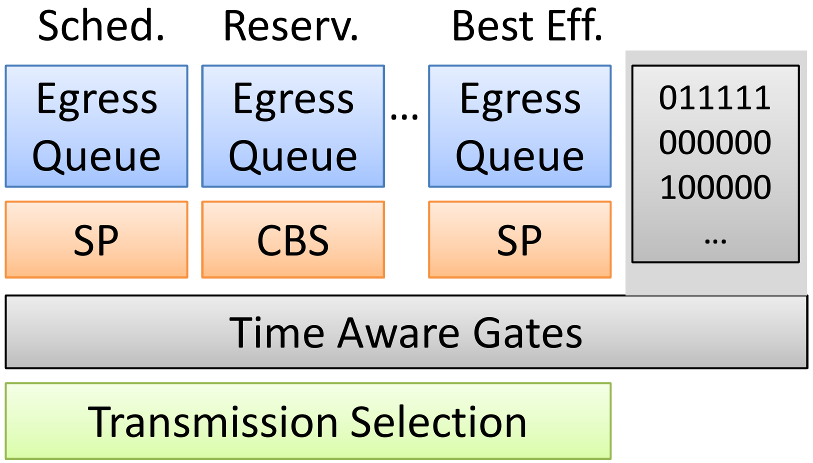
- 802.1 get‘s Preemption!
  - TAS would not be possible without preemption since Guard Windows would be way to large

- Preemption Performance
  - Preemption of a frame takes requires a reasonable short worst case delay from preemption until preempting class can transmit, e.g. 84 byte times ($t_{MaxPreemption}$)

Assumptions

- Talker behavior stay‘s as it is for Scheduled Traffic!
  - Talkers implement the Time Aware Shaper (TAS)
    Bridges implement TSTS instead of TAS in this proposal
  - All talkers sending low latency traffic are synchronized with good precision, e.g. 1 µs
    With TAS in bridges, imprecise or async. talkers couldn‘t reach low latency: The frames of these talkers would be queued until the next TAS window
  - Talker always fulfill their contracts for scheduled traffic, e.g. <u>period</u>, <u>phase</u> and <u>max. frame size</u>
    If talkers would violate contracts, TAS in bridges couldn‘t protect scheduled traffic of other (non contract-violating) talkers

Recap: Time Aware Shaper



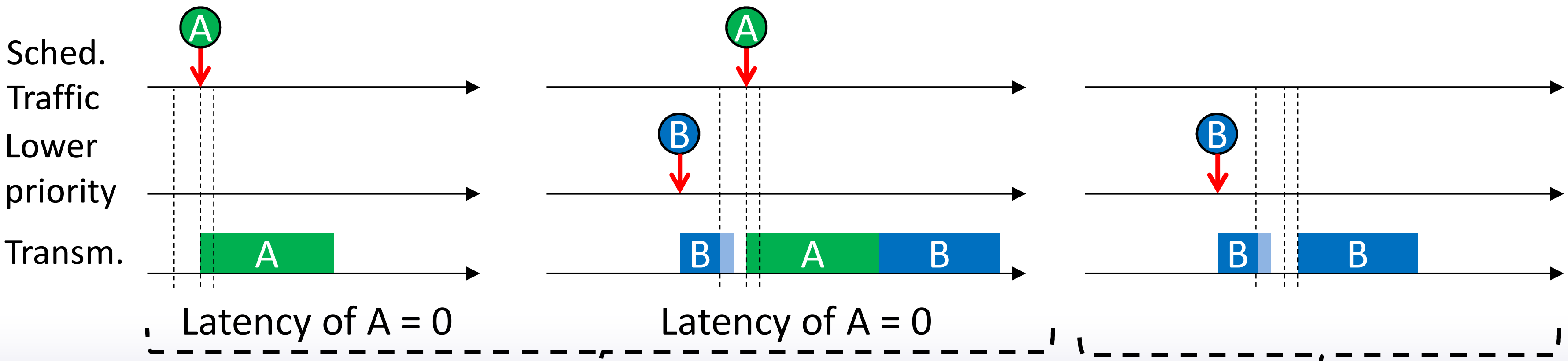
Legend

- (A) Scheduled frame transmission start (after arrival on egress port)
- [A] Frame transmission of A
- (B) Lower priority frame transmission start (may be taken from queue)
- [B] Frame or framelet transmission of B

Case 1: No low priority transmission

Case 2: Long low transmission before scheduled transmission

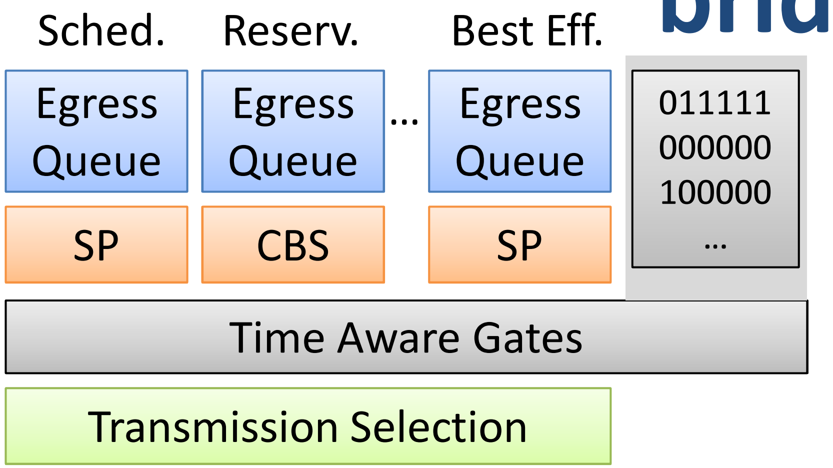
Case 3: No frame for scheduled transmission



→ ***No* additional jitter per Hop!**

→ *Bandwidth lost!*

Removing TAS and Preemption **from bridges …**



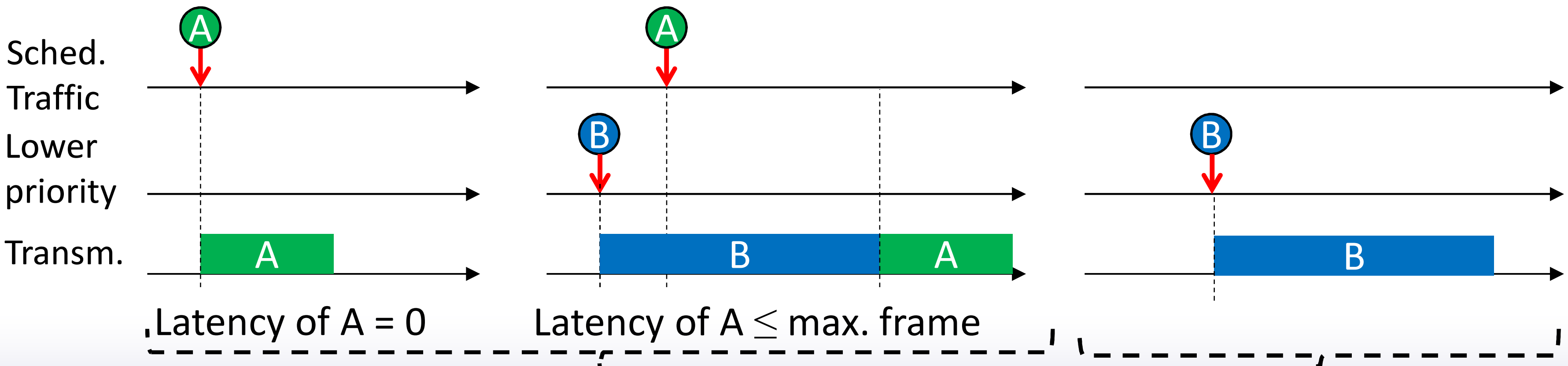
Legend

- (A) Scheduled frame transmission start (after arrival on egress port)
- [A] Frame transmission of A
- (B) Lower priority frame transmission start (may be taken from queue)
- [B] Frame or framelet transmission of B

Case 1: No low priority transmission

Case 2: Long low transmission before scheduled transmission

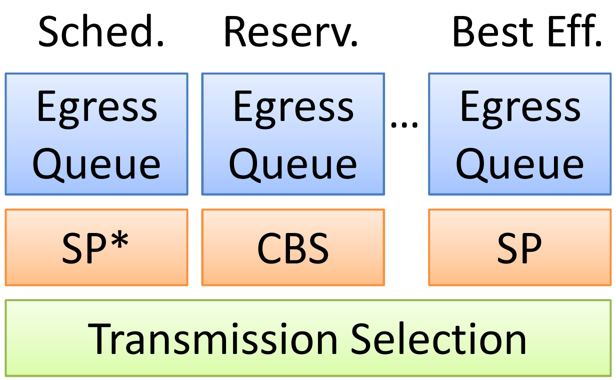
Case 3: No frame for scheduled transmission



→ Large additional Jitter per Hop!

***→ No** Bandwidth lost!*

Basic Idea (first part): Only used mechanism **in bridges: Preemption**



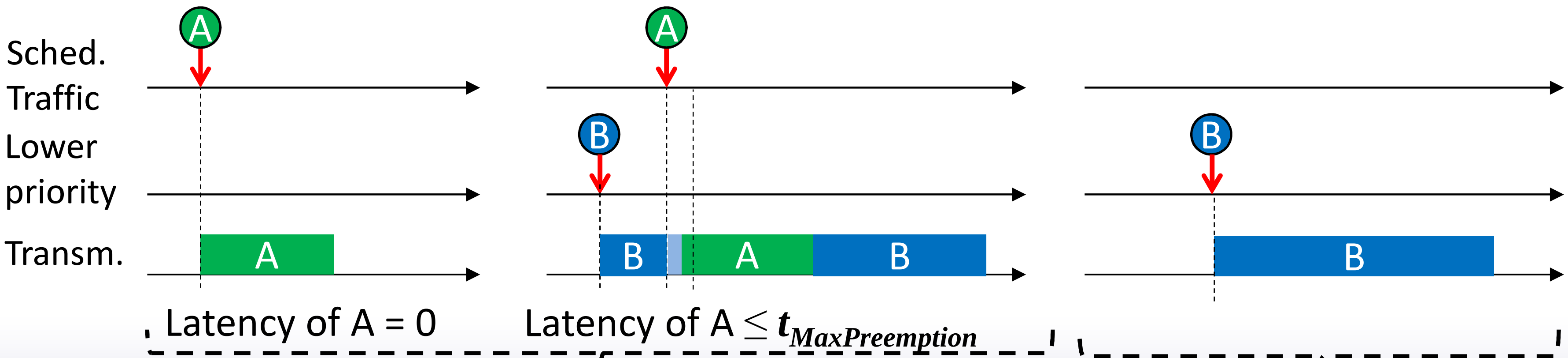
Legend

- (A) Scheduled frame transmission start (after arrival on egress port)
- [A] Frame transmission of A
- (B) Lower priority frame transmission start (may be taken from queue)
- [B] Frame or framelet transmission of B

Case 1: No low priority transmission

Case 2: Long low transmission before scheduled transmission

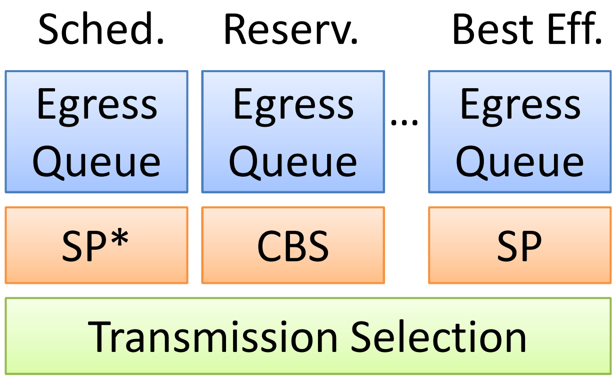
Case 3: No frame for scheduled transmission



→ *Small Jitter per Hop:* $t_{MaxPreemption}$

→ ***No** Bandwidth lost!*

Basic Idea (second & last part): Always wait for $t_{MaxPreemption}$



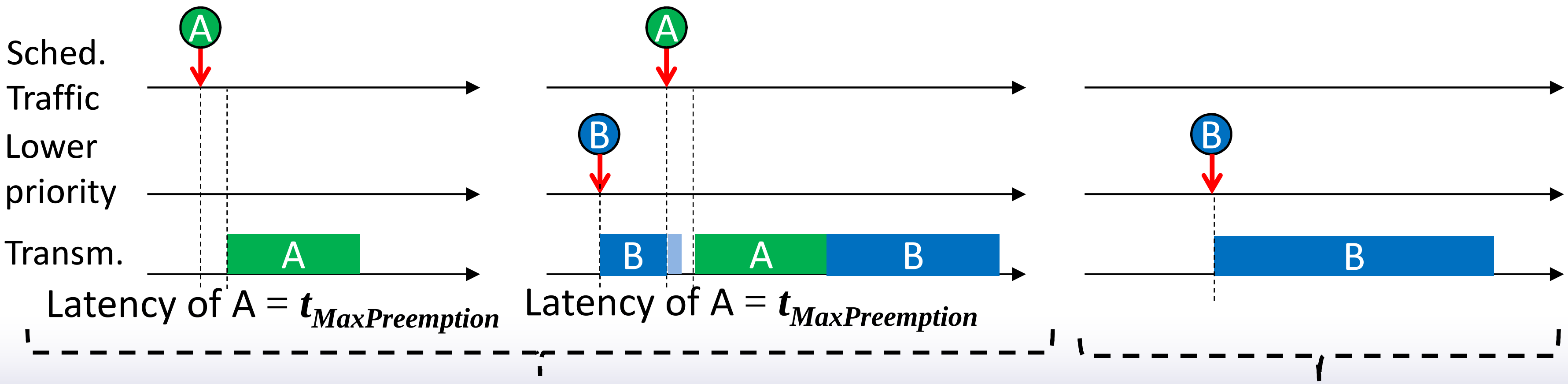
Legend

- (A) Scheduled frame transmission start (after arrival on egress port)
- [A] Frame transmission of A
- (B) Lower priority frame transmission start (may be taken from queue)
- [B] Frame or framelet transmission of B

Case 1: No low priority transmission

Case 2: Long low transmission before scheduled transmission

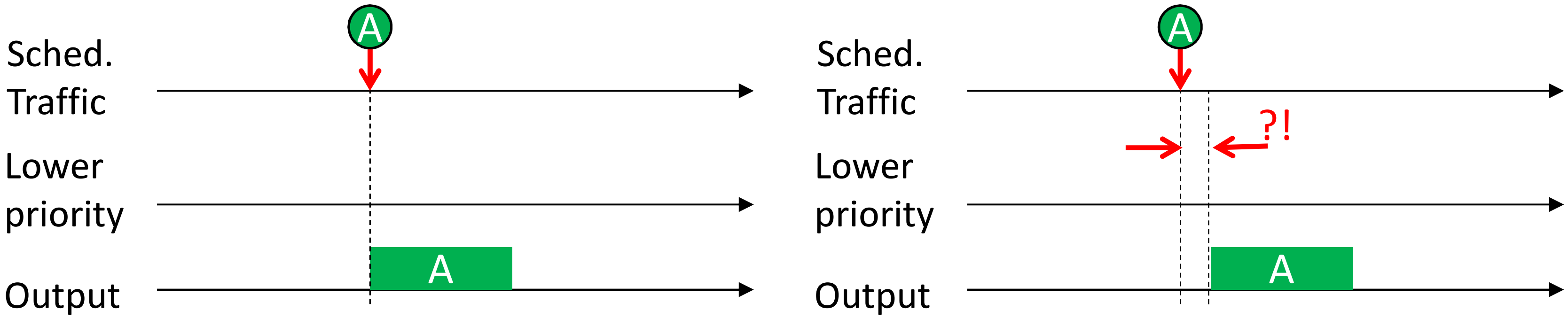
Case 3: No frame for scheduled transmission



→ ***No** additional Jitter per Hop!*

→ ***No** Bandwidth lost!*

But why is waiting needed?



→ **Latency of A:** 0

→ **Latency of A:** $t_{MaxPreemption}$

- If there is no need for preemption, transmission could start ASAP, i.e. at frame arrival on egress port …
- … then transmissions would accumulate a jitter of $t_{MaxPreemption}$ per hop.
- Considering multiple hops, this could lead to race conditions between multiple scheduled traffic streams

Summarized: Egress Operation in Bridges

When a scheduled frame becomes ready for transmission:

1. **Preempt** the current lower priority frame if present and …
2. **… wait** a **constant** time of $t_{MaxPreemption}$ while **holding** transmission permission
3. **Send** the frame

Note:

Waiting is required per queue/port: traffic is scheduled and thus there's no queuing like with e.g. best effort traffic

Comparison: Latency

	Payload Size	Hops						
		2	3	4	5	6	7	8
TAS	64 [Byte]	16,96 [us]	25,44 [us]	33,92 [us]	42,40 [us]	50,88 [us]	59,36 [us]	67,84 [us]
	128 [Byte]	27,20 [us]	40,80 [us]	54,40 [us]	68,00 [us]	81,60 [us]	95,20 [us]	108,80 [us]
	256 [Byte]	47,68 [us]	71,52 [us]	95,36 [us]	119,20 [us]	143,04 [us]	166,88 [us]	190,72 [us]
	512 [Byte]	88,64 [us]	132,96 [us]	177,28 [us]	221,60 [us]	265,92 [us]	310,24 [us]	354,56 [us]
	1024 [Byte]	170,56 [us]	255,84 [us]	341,12 [us]	426,40 [us]	511,68 [us]	596,96 [us]	682,24 [us]

	Payload Size	Hops						
		2	3	4	5	6	7	8
TSTS	64 [Byte]	23,68 [us]	38,88 [us]	54,08 [us]	69,28 [us]	84,48 [us]	99,68 [us]	114,88 [us]
	128 [Byte]	33,92 [us]	54,24 [us]	74,56 [us]	94,88 [us]	115,20 [us]	135,52 [us]	155,84 [us]
	256 [Byte]	54,40 [us]	84,96 [us]	115,52 [us]	146,08 [us]	176,64 [us]	207,20 [us]	237,76 [us]
	512 [Byte]	95,36 [us]	146,40 [us]	197,44 [us]	248,48 [us]	299,52 [us]	350,56 [us]	401,60 [us]
	1024 [Byte]	177,28 [us]	269,28 [us]	361,28 [us]	453,28 [us]	545,28 [us]	637,28 [us]	729,28 [us]

	Payload Size	Hops						
		2	3	4	5	6	7	8
Comparison	64 [Byte]	71,62%	65,43%	62,72%	61,20%	60,23%	59,55%	59,05%
	128 [Byte]	80,19%	75,22%	72,96%	71,67%	70,83%	70,25%	69,82%
	256 [Byte]	87,65%	84,18%	82,55%	81,60%	80,98%	80,54%	80,22%
	512 [Byte]	92,95%	90,82%	89,79%	89,18%	88,78%	88,50%	88,29%
	1024 [Byte]	96,21%	95,01%	94,42%	94,07%	93,84%	93,67%	93,55%

Assumptions:

100 Mbps — no cable delays — $t_{MaxPreemption}$ = 84 bytes

no cut through(reception incl. IPG before forwarding) — no forwarding delays

Comparison: Jitter and Latency

The latency of TSTS is higher then the latency of TAS, however:

- TSTS fulfills the Automotive control data class requirements as presented by an OEM (cmp. [1]/green cells in prev. slide):
  - 100 µs Latency@5 Hops for 128 byte payload
- The ratio between TAS and of TSTS is not that bad, e.g. ~72% in the automotive example
- If the low latency frame (A) is missing, bandwidth utilization by lower priorities is higher (cmp. Case 3 in prev. slides)
- The mechanism is simple!

[1] QoS requirements for Automotive Ethernet backbone systems, 11/2011, http://www.ieee802.org/1/files/public/docs2011/new-avb-nakamura-automotive-backbone-requirements-0907-v02.pdf

Comparison: Complexity

No need to synchronize egress ports …

- … bridges might even be time-unaware*
- The reduced jitter may enable further use-cases beside Automotive ultra low latency control applications, e.g. clock sync. across time-unaware parts of a network

Less gates/transistors on data plane in bridges:

- TAS requires time gates (special logic), Gate driver and gate event memory (in the best case a “large enough“ dedicated SRAM) per Port, mechanisms for gate event configuration, etc.
- The proposed mechanism requires a reasonable small FSM per TSTS queue per port

Comparison: Configuration

No egress data plane configuration:

The only parameter is $t_{MaxPreemption}$ which is a constant – could either be fixed by upcoming standards or at latest during manufacturing of a bridge

- Plug and Play use-cases:
  - No Protocols needed to adjust the data plane/egress ports during runtime, although …
  - … there may be the need for protocols like SRP to guarantee „sufficient remaining bandwidth“ for reserved traffic, BPDUs, etc., but …
  - … this can be handled during resource allocation on the control plane exclusively, e.g. by rejecting conflicting allocation attempts

- Automotive (and other engineered) use-cases:
  - No need to configure bridges during network integration (OEM/Tier-1)
  - No need to identify and specify additional TAS-requirements like the minimum number of Gate Events, etc. (OEM)
  - Freedom to use additional Clock Sync. Protocols beside 802.1AS with high precision

Summary & Conclusions

- TS Traffic Support in Bridges was proposed as an alternative to/simplification of TAS
- TS Traffic Support has the following key aspects:
  - End 2 End Latency is a bit higher than with TAS, but …
  - Bandwidth utilization by lower classes can be a bit lower
  - … Automotive Ultra Low Latency Requeirements can be fulfilled
  - No data plane configuration in bridges
  - Complexity of bridge implementations seems to be low
    → maybe there is some chip area left for ingress policing ;-)

Thank you for your Attention!

Opinions, Questions, Ideas?

Johannes Specht

Dipl.-Inform. (FH)

Dependability of Computing Systems
Institue for Computer Science and
Business Information Systems (ICB)
Faculty of Economics and
Business Administration
University of Duisburg-Essen

Schützenbahn 70
Room SH 502
45127 Essen
GERMANY
T +49 (0)201 183-3914
F +49 (0)201 183-4573

specht@dc.uni-due.de
http://dc.uni-due.de