SIEMENS
Ingenuity for life

Support for µStream Aggregation in RAP (ver 0.3)

Feng Chen, Franz-Josef Goetz, Marcel Kiessling, Jürgen Schmitt

IEEE 802.1 Interim Meeting
Jan. 2019, Hiroshima, Japan

Frei verwendbar

Updates in v0.3

SIEMENS
Ingenuity for life

This slide deck (v0.3) contains the updates to address the questions raised during the presentation of its previous version (v0.2: <dd-chen-flow-aggregation-1218-v02.pdf>, also attached after the update slides) at the TSN call on Dec. 10 2018 and also to reflect some of the offline comments received by emails.

1) Recap: why aggregation of µStreams is needed?

2) Interleaving with no need for network topology information

3) Aggregation for µStreams with very different data frame sizes

4) Indenpendency among different aggregation groups

5) Proposals for RAP for support of µStream aggregation

Recap

SIEMENS
Ingenuity for life

Why aggregation of µStreams is needed?

- Many industrial control devices produce a number of µStreams at a low data exchange rate (e.g. 1kHz =>1 ms interval) but need a relatively high transmission rate in the network to meet the end-to-end latency requirement (e.g. 1ms over 15 hops)

- If transmitting those low-rate µStreams as individual streams and making reservation for each of them, it will cause massive bandwidth overprovisioning (e.g. a factor of 16 as a result of using CQF with a cycle time of 62,5 µs to meet 1ms e2e latency over 15 hops)

- Aggregation of low-rate µStreams into one high rate common stream can help
  - **reduce bandwidth overprovisioning**
  - **reduce worst-case interference**
  - **reduce control plane overhead (**only one reservation for all aggregated µStreams**)**
  - **reduce forwarding plane overhead (**only one Stream-DA for transmission of all aggregated µStreams**)**

Interleaving with no need of topology information

SIEMENS
Ingenuity for life

<u>Question 1:</u> why scheduling for interleaving in this proposal does not require the knowledge of the network topology?

- Interleaving is an effective method for aggregation of a number of low-rate µStreams into a common stream with a higher rate, by scheduling a staggered transmission of µStreams either at the Talker for sending in 1-to-n communication or at the Listener for receiving in n-to-1 communication.

- An Interleaving Schedule (IS) can be computed locally within the Talker or the Listener using (see Page 14):
  - Traffic specifications of all the µStreams to be aggregated to the same common stream
  - The class measurement interval of the SRclass used for stream transmission in the network

- The used algorithm should compute the IS in such a way that the resulted bandwidth (i.e. the maximum of all column totals, see Page 15) in the Tspec used for reservation of the common stream is minimized. **No topology related information is used for calculating an IS**.

- In 1-to-n cases, the IS is only used internally at the Talker and stays invisible to network / all Listeners.
- In n-to-1 cases, the IS computed by the Listener needs to be adjusted for use at each Talker. This can be done trivially within the network by the reservation protocol at runtime, because of the use of CQF.

Aggregation for µStreams with different frame sizes

SIEMENS
Ingenuity for life

<u>Question 2:</u> how to apply aggregation to µStreams with very different frame sizes without causing excessive bandwidth overprovisioning?

- The current proposal extracts the maximum value from the Max. frame sizes of all µStreams and uses it as the Max. frame size in the Tspec of the common stream for resource reservation in the network.
  - with the goal of making aggregation **transparent** to the network and reducing control-plane overhead
  - by carrying as less µStream-level information as possible on the network control plane and **avoiding using µStream-level information** for reservation at bridges
- This may cause over-reservation problems when aggregating µStreams with very different frame sizes, in particular at the branch links which are used only by one or a few µStreams with a small frame size when transmitted without aggregation.
- To avoid this, it is important for the user application to make a wise decision on which of the µStreams should be aggregated into to one common stream, e.g. only those with similar frame size. If possible, the application should pack their data into frames of a similar size for all its µStreams.
- Another possible solution is to carry additional µStream information, e.g. per-µStream Tspec in the reservation protocol, which allows making reservation with finer granularity. However, this would break the transparency of aggregation and add much complexity and overhead to the control plane.

Independency of Stream Aggregation

SIEMENS
Ingenuity for life

Question 3: is there mutual dependency among multiple aggregation groups and also other unaggregated streams?

- As previously described on slide 4 for Question 1, calculating an IS for an aggregation group uses the information of the µStreams within that group and the parameters of the used SRclass, e.g. for CQF the cycle interval and cycle start time. **There is no timing dependency among different aggregation groups with regard to interleaving.**

- The proposed aggregation scheme assumes use of a certain **traffic shaping scheme** for stream transmission, in particular using CQF for aggregation in n-to-1 communication, and use of **a MSRP-like distributed control plane**, i.e. RAP, for resource reservation.

- RAP for reserving a common stream with aggregated µStreams follows the same principle as that for reserving a normal unaggregated stream.
  - Reservation is made for each common stream based on its Tspec, independent of the timing for interleaved transmission of the µStreams constituting that common stream.
  - Adding/removing a stream (either a common or a normal stream) won‘t change the latency bounds of the existing/remaining streams, because the latency bounds are already calculated by taking into worst-case situation into account.

Proposals for P802.1Qdd RAP

SIEMENS
Ingenuity for life

- To support µStream aggregation with **Talker Interleaving for 1-to-n communication**, **RAP can use the same reservation mechanisms as used for reservation of normal streams**
  - Interleaving is only applied within the Talker and not visible to the network or the Listeners.
  - The resulted common stream is nothing other than a normal multicast stream for the network on both data plane and control plane.

- To support µStream aggregation with **Listener Interleaving for n-to-1 communication**, the following additional features are needed for RAP:
  - specify a new reservation flow called „Listener Advertise and Talker Join“ (see Page 16)
  - define two new attributes for Listener-Advertise and Talker-Join
    - **Listener-Advertise** has similar data and functions as the legacy Talker-Advertise, but carrying an extra „Schedule Table“, which gets „rotated“ at each hop (see Page 17)
    - **Talker-Join** has similar data and functions as the legacy Listener-Join.

Aggregation of Micro-Streams into one Common Stream (v0.2)

Feng Chen, Franz-Josef Goetz, Marcel Kiessling, Jürgen Schmitt

IEEE 802.1 TSN Call
Dec. 12, 2018

Frei verwendbar

Control Loops within Industrial Machines

A typical industrial machine is build up of a huge amount of different physical actuators and sensors. They are connected to so called IO-Devices by different technologies, from electric wire to small busses (e.g. IO Link, PROFIBUS, CAN…). The IO-Devices in turn send/receive data in **a wide range of different rates** (e.g. typically between 1 kHz and 60 kHz) and **in a wide range of amount of real time data** according to the requirements of the control application and the type of sensors and actuators connected.

Industrial use cases:
Up to several hundred IO-Devices that are connected to thousands of actuators and sensors periodically exchange their real time data with one or several PLCs with

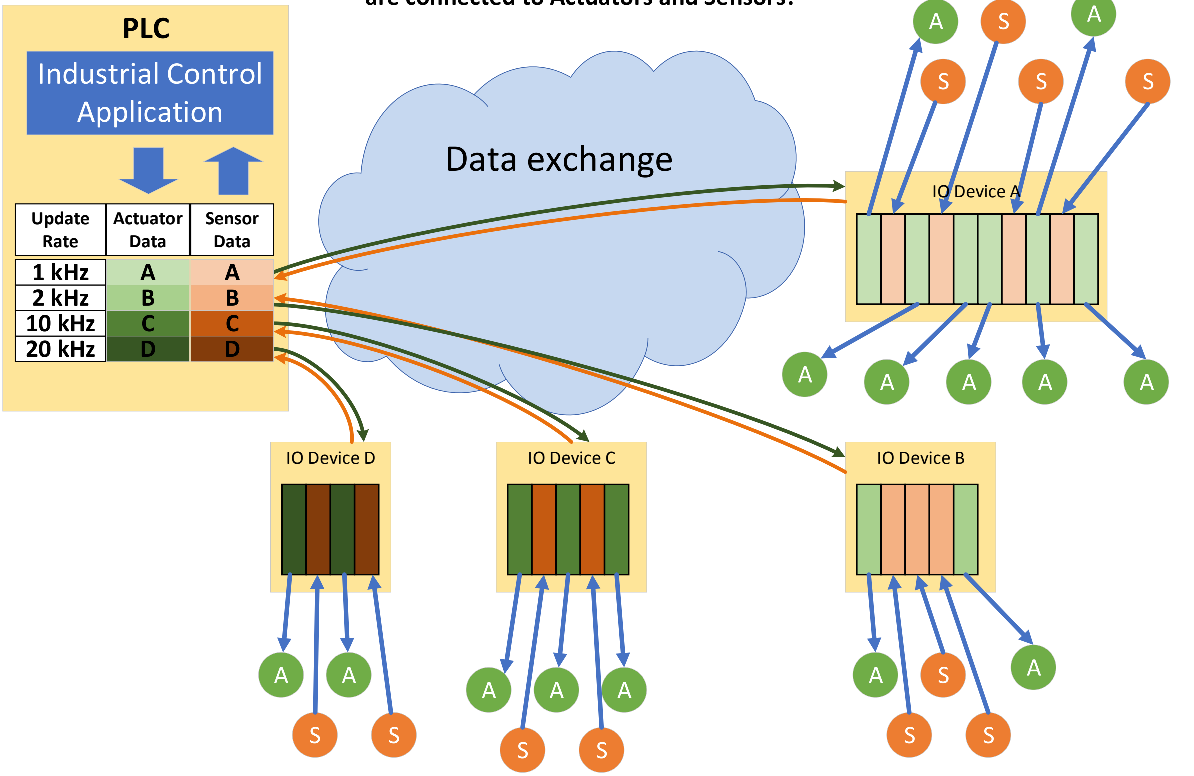
- **low data rates and small amount of real time data,**
- **low data rates and huge amount of real time data,**
- **high data rates and small amount of real time data,**
- **high data rates and huge amount of real time data,**

or with a mixture of all of them.

In contrast, Audio / Video applications typically have high data rates with huge amount of data.

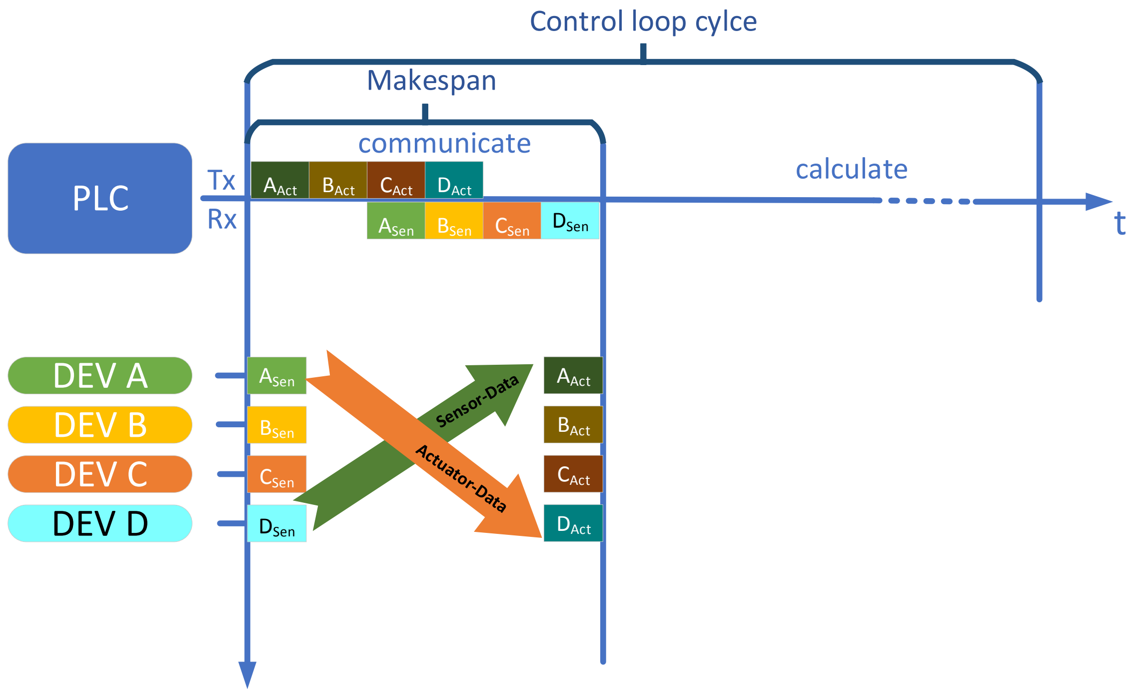
A typical Industrial Automation Use Case for Connectivity

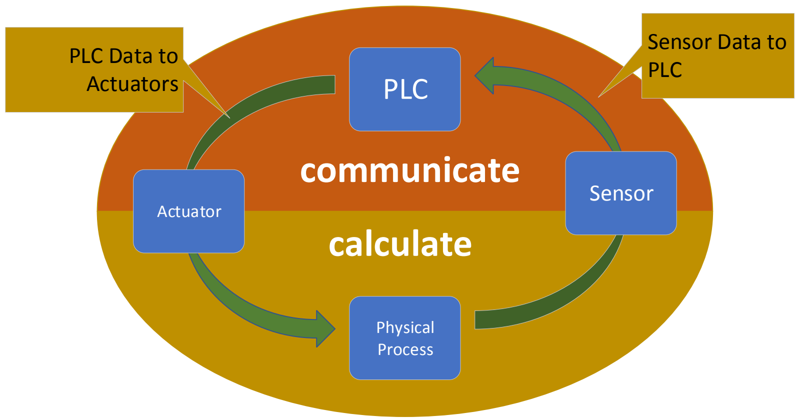
Programmable Logic Control (PLC) exchange periodically real time data with Input/Output(IO)-Devices which are connected to Actuators and Sensors!



Control Loops within Industrial Machines Requirements

SIEMENS
Ingenuity for life





Goals of the application:

- Minimize Makespan (time to exchange **all** Data between Actuators/Sensors ←→ PLC within a Closed Loop Application)
- Transmission time from PLC to Actuators should be close to transmission time from Sensors to PLC.
  If they are not equal the maximum of both counts as “Makespan”

Goals for the network:

- Minimize the required resources (e.g. queues, addresses, …)

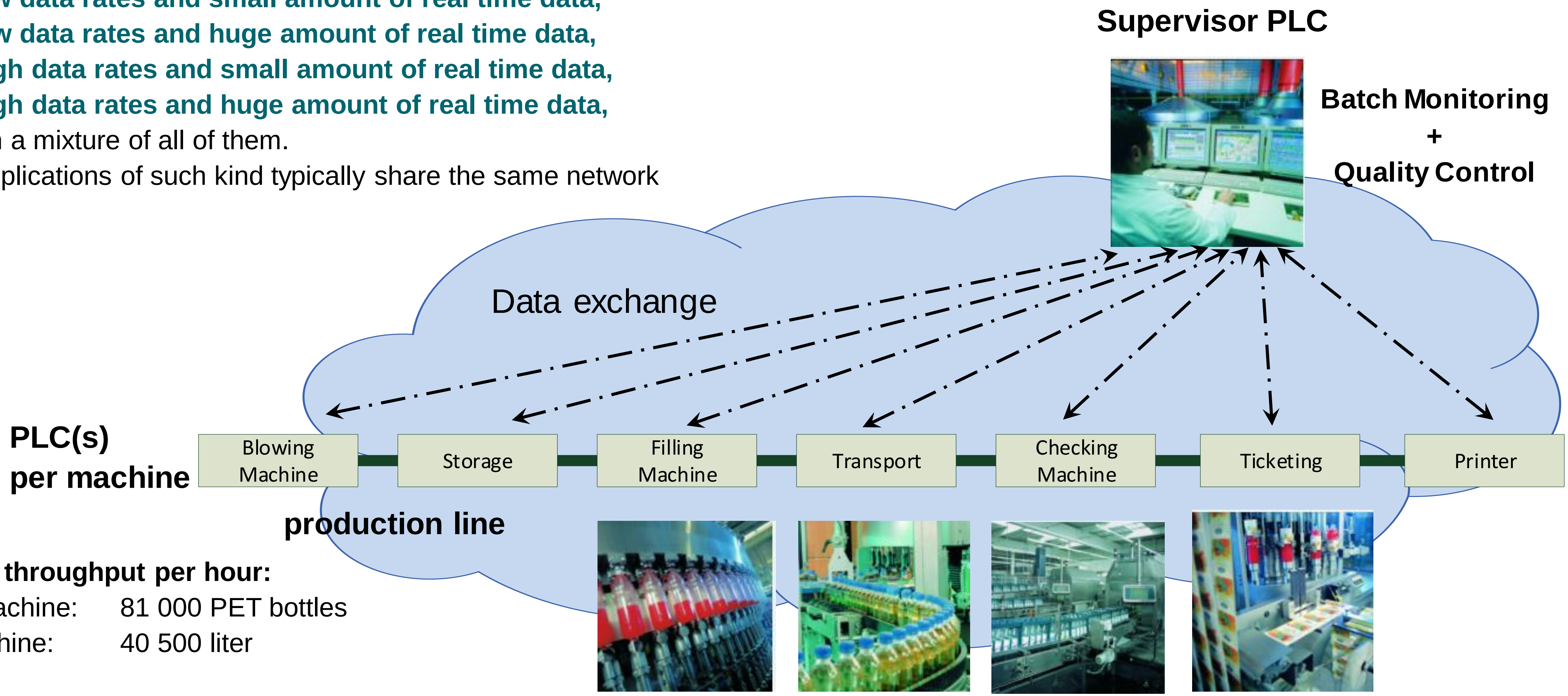
Supervisor PLC <-> PLCs Use Case:
Quality Control at Real Time in a Bottling Plant

SIEMENS
Ingenuity for life

Industrial use cases:

- ❑ Up to tens of PLCs periodically exchange real-time data with one supervisor PLC with
  - **low data rates and small amount of real time data,**
  - **low data rates and huge amount of real time data,**
  - **high data rates and small amount of real time data,**
  - **high data rates and huge amount of real time data,**

  or with a mixture of all of them.
- ❑ Many applications of such kind typically share the same network



Example for throughput per hour:

- Blowing Machine: 81 000 PET bottles
- Filling Machine: 40 500 liter

Example for Stream Reservation
based on MSRP without Micro-Stream (µStream) aggregation

SIEMENS
Ingenuity for life

Example:

- 1 PLC ←→ 50 IO-Devices (bidirectional, ~ 50 µStreams per direction)
  - IO-Device with real time data rate of 1 kHz (µStream transmission rate)
  - Max e2e latency: 1ms
  - Max hop count: 16
  - ➔ Max per hop latency: 62,5µs

- Stream reservation based on MSRP
  - a SR-Class with class measurement interval 62,5µs ~ 16 kHz to fulfill the max e2e latency requirement

➔ If making each µStream an individual Stream using the existing MSRP mechanisms without µStream-Aggregation, it will result in an overprovisioning factor of **16** for reservation of such 50 µStreams per direction.

Proposal: µStream Aggregation using Interleaving

SIEMENS
Ingenuity for life

Definition

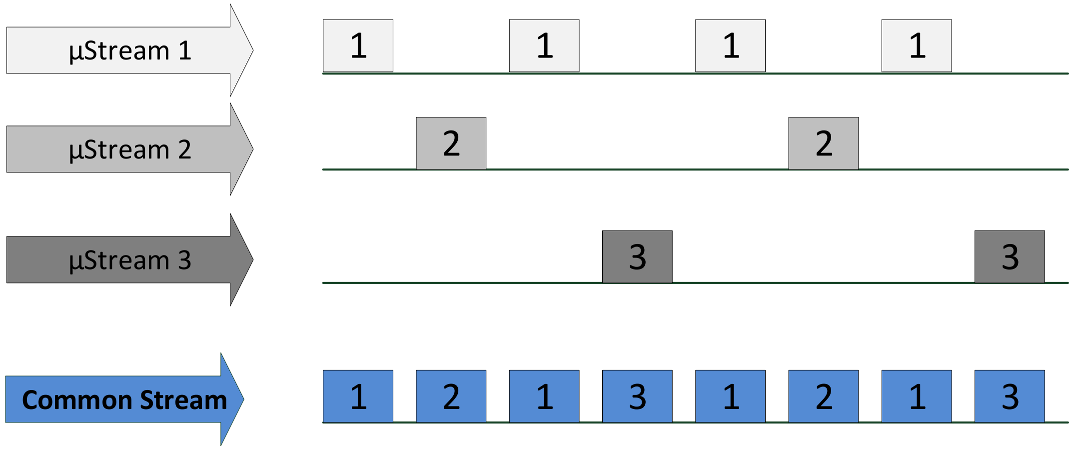
- **µStream Aggregation** combines multiple µStreams into one common Stream
  - forwarded in the network using one multicast destination MAC address
  - reserved in one stream reservation process by using a common Traffic Specification (TSpec)

- **µStream Interleaving is a method to combine multiple µStreams into one common Stream using multiplexing.**

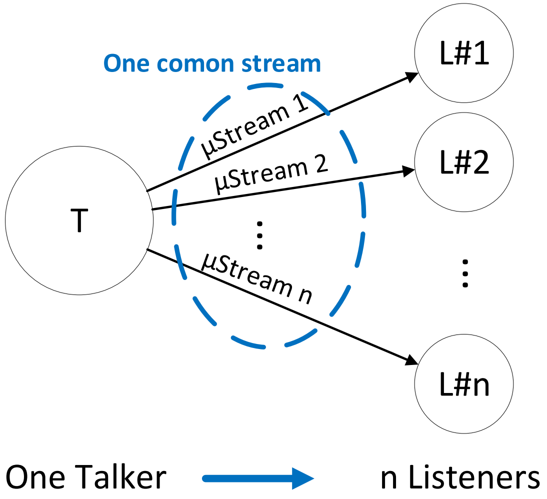
  - **Talker µStream-Interleaving** (for 1-to-n communication)
    A single Talker is responsible for organizing its µStreams for multiple Listeners into a common Stream.

  - **Listener µStream-Interleaving** (for n-to-1 communication)
    A single Listener is responsible for organizing the µStreams that are transmitted by multiple talkers to this Listener into a common Stream.

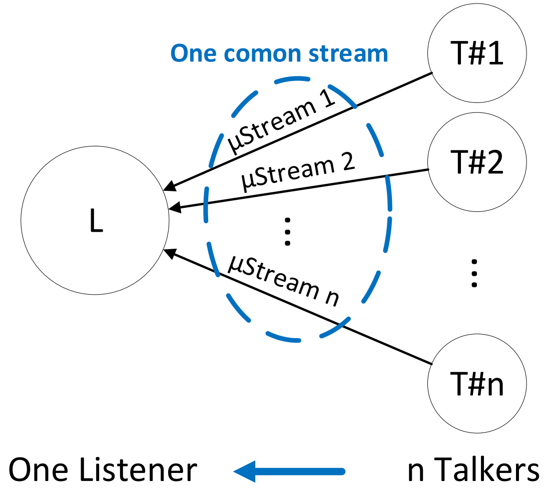
µStream Aggregation using Interleaving



Talker µStream-Interleaving



Listener µStream-Interleaving



Local Computation of Interleaving Schedule for µStream Aggregation

An **Interleaving Schedule (IS)** for a common stream

- is locally computed either at the Talker or at the Listener, who knows the traffic specifications of all aggregated µStreams, but not required to have the knowledge of the network topology.
- specifies a repeating time schdule that allocates the time slots to transmission or reception of all aggregated µStreams in a certain way, e.g. distribute total bandwidth of all µStreams among slots as evenly as possible.

❑ For **Talker µStream-Interleaving** in 1-to-n communication, an **IS**

- is computed locally at the **Talker**, who intends to transmit n µStreams to multiple Listeners.
- **schedules interleaved µStream transmission at the Talker**
- is used by the Talk, not necessarily known to the Listeners

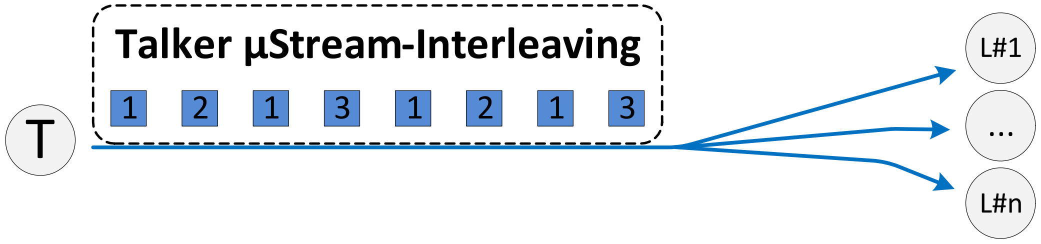
❑ For **Listener µStream-Interleaving** in n-to-1 communication, an **IS**

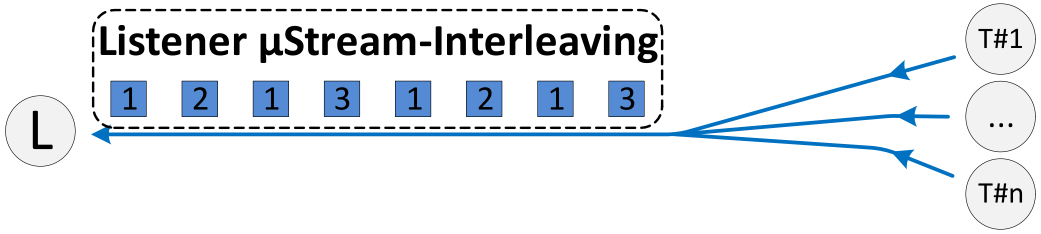
- is computed locally at the **Listener**, who intends to receive n µStreams from multiple Talkers.
- **schedules interleaved µStream reception at the Listener**
- **must be propagated to and converted for use by each Talker**

Example of a locally computed Interleaving Schedule

Traffic specification of µStream i given by application: $\mathbf{TS_i(M_i, N_i, L_i)}$
$\mathbf{M_i}$: max frame size; $\mathbf{N_i}$: number of frames; $\mathbf{L_i}$: trans. interval

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	…	Slot m
µStream 1	M_1xN_1	-	M_1xN_1	-	M_1xN_1	-	-	M_1xN_1
µStream 2	-	M_2xN_2	-	-	-	M_2N_2	-	-
…	…	…	…	…	…	…	…	…
µStream n	-	-	-	M_nxN_n	-		-	M_nxN_n





Stream Reservation for Talker µStream-Interleaving

SIEMENS
Ingenuity for life

Target application of Talker µStream-Interleaving:

- one talker to n listeners communcaiton, e.g. PLC to I/O devices, supervisor PLC to PLCs

Assumption:

- The talker has the information of all the µStreams to be aggregated, such as µStream TSpecs, application-level µStream identification, etc.

Workflow:

- The Talker computes the IS (*see previous slide*) and derives the TSpec for use in the reseravtion of the common stream (*see figure right above*)

- The Talker initiates the reservation process using TSpec of the common stream, which follows the conventional reservation procedures with **„Talker-Advertise / Listener-Join“**

  => Talker µStream-Interleaving is transparent to the network and can be applied with the exsiting reservation method.

- Upon successful reservation, the Talker starts stream transmission according to the locally computed IS and using the same stream DA for all aggregated µStreams.

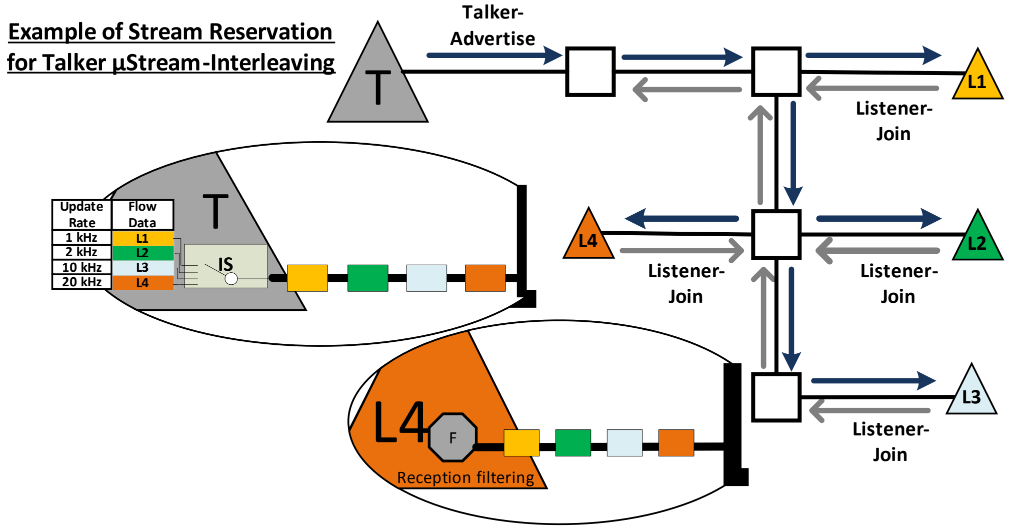
- Each Listener performs local filtering of received µStreams.

$N_{s1} = \sum_{i=1}^{n} N_i$ in slot 1

	Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Slot 6	…	Slot m
µStream 1	M_1xN_1	-	M_1xN_1	-	M_1xN_1	-	-	M_1xN_1
µStream 2	-	M_2xN_2	-	-	-	M_2N_2	-	-
…	…	…	…	…	…	…	…	…
µStream n	-	-	-	M_nxN_n	-		-	M_nxN_n

Derive TSpec of the common stream: $\boldsymbol{TS(M,N,L)}$ from the TSepcs of n aggregated µStreams $TS_i(M_i, N_i, L_i)$ and the IS

$$\boldsymbol{M = \max\{M_1, M_2, M_3, \dots, M_n\} \rightarrow max.frame\ size}$$
$$\boldsymbol{N = \max\{N_{s1}, N_{s2}, N_{s3}, \dots, N_{sm}\} \rightarrow max.number\ of\ frames}$$
$$\boldsymbol{L = slot\ length}$$



Stream Reservation for Listener µStream-Interleaving

SIEMENS
Ingenuity for life

Target application of Listener µStream-Interleaving:

- n talkers to one listener communcaiton, e.g. I/O devices to PLC, PLCs to supervisor PLC

Assumption:

- The Listener has the information of all the µStreams to be aggregated, such as µStream TSpecs, application-level µStream identification.
- The **Cyclic Queuing and Forwarding** (CQF) is applied in the network for transmission of the aggregated µStreams, while the CQF cycle time is made equal to the time slot length of the IS

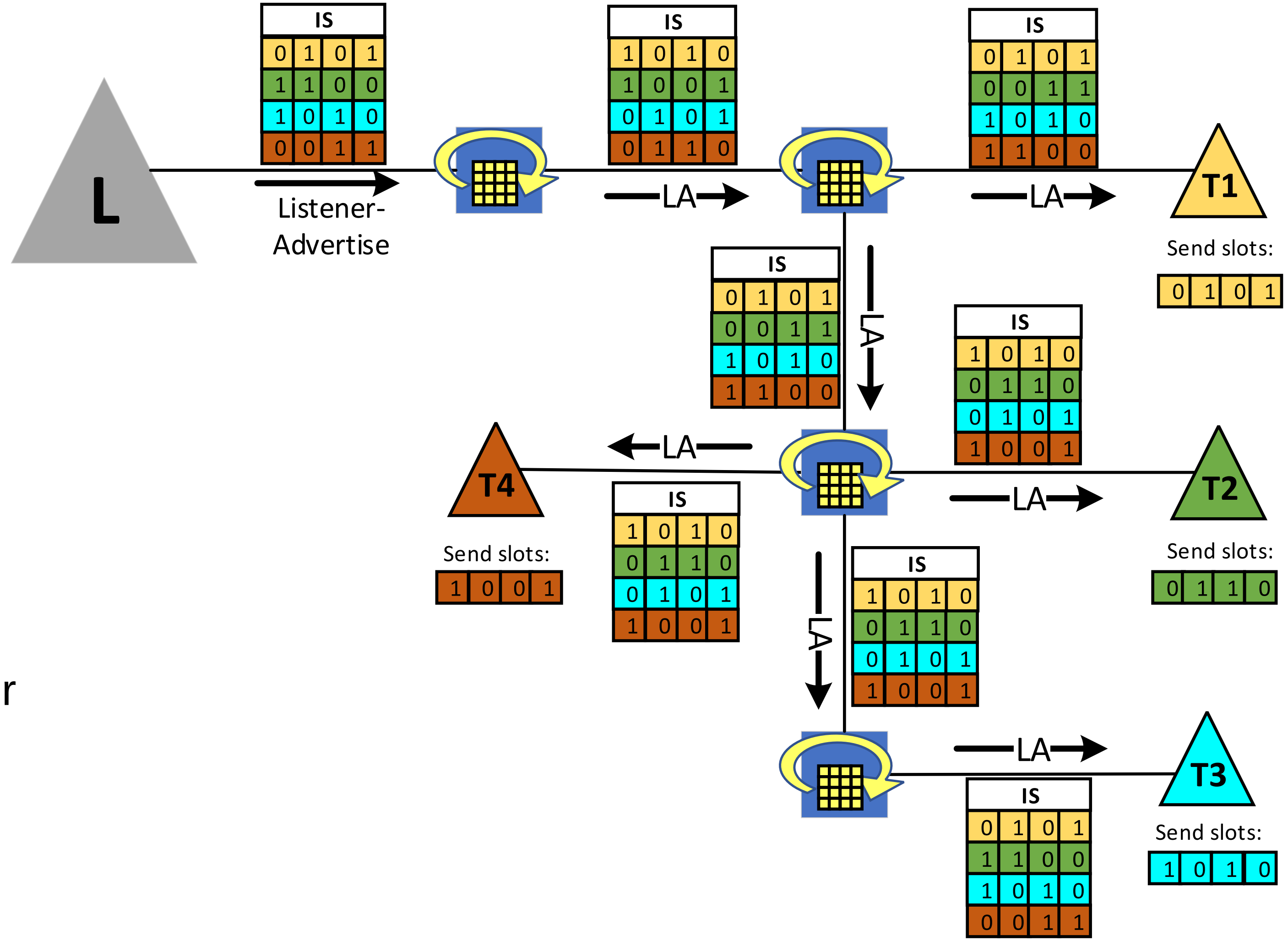
Workflow:

- The Listener computes the IS and derives the TSpec for use in the reseravtion of the common stream (*see previous slide*)
- The Listener initiates the reservation process, which follows a reverse reservation procedure (in contrast to the conventional one), called **„Listener-Advertise (LA) / Talker-Join (TJ)“**
    - The LA carries the TSpec of the common stream and other stream information as in the legacy Talker-Advertise, e.g. stream-DA
    - In addition, the LA carries the IS computed by the Listener together with the timing information necessary for the Talkers to derive the beginning of each scheduling cycle.
    - The IS is adjusted at each hop during its porpagation along each path from the Listener to each Talker, which rotates the IS once in that each row in the IS is shifted back (to earlier time) by exactly one time slot. (This is feasible due to the use of CQF)
- Each Talker transmits its µStream according to the corresponding row in the received IS.

Example of Interleaving Schedule Rotation in Reservation for Listener µStream-Interleaving

SIEMENS
Ingenuity for life

- An essential step in Listener µStream-interleaving is to convert a receiving schedule locally computed by the Listener to a sending schedule for use by each Talker within a distributed stream reservation process.
- Using CQF on the data plane and adding support for a new reservation µStream with “Listener-Advertise / Talker-Join” to the reservation protocol are two keys to applying µStream aggregation with listener µStream-interleaving for n-to-1 communication.



Summary and Outlook

- ❑ Advantages of µStream-Aggregation to a common Stream (focus of this presentation)
  - Reduce bandwidth overprovisioning
  - Reduce the control plane overhead, e.g. the number of reservation data and Stream Das
  - Talker / Listener µStream-Interleaving is independent from topology
  - Locally computable by Talker / Listener, no need for a central controller
  - Streams / common Streams remain independent from each other
  - Dynamic given by a reservation protocol is still available to Streams / common Streams

- ❑ Outlook for aggregated Streams (upcoming)
  - Support for a large number of streams is required by industrial backbone networks.

Thank You!

SIEMENS
Ingenuity for life

Discussion