Flow Aggregation on Common Stream for Industrial Applications

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

IEEE 802.1 Plenary Meeting
Nov. 2018, Bangkok, Thailand

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 of IO-Devices which are connected to some thousand of actuators and sensors periodically exchange their real time data with one or several PLC's with

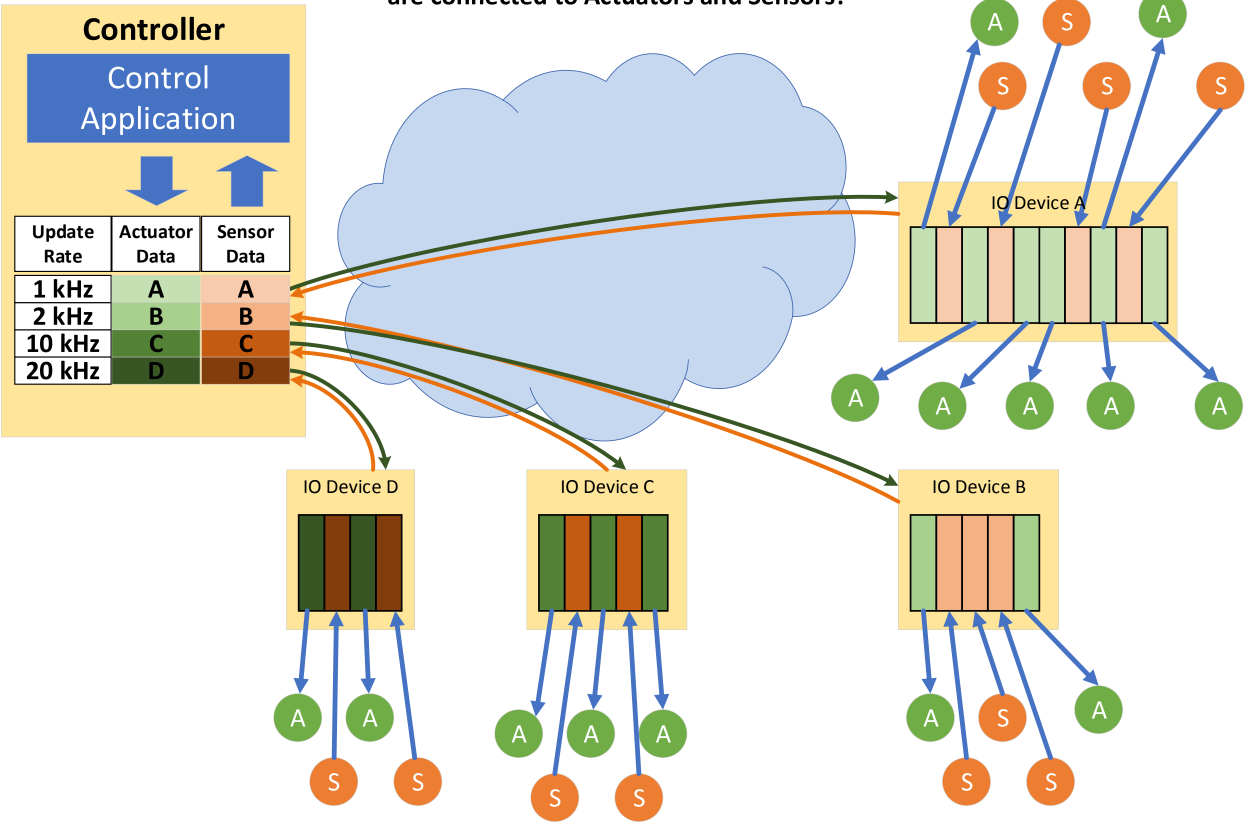
- **low data rates and small amount of real time data,**
- **low data rates and huge amount of real time data,**
- **high data rates and small amount of real time data,**
- **high data rates and huge amount of real time data,**

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

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



Update Rate	Actuator Data	Sensor Data
1 kHz	A	A
2 kHz	B	B
10 kHz	C	C
20 kHz	D	D

Example for Stream reservation based on MSRP without flow aggregation

Assumption:

- 1 PLC ←→ 50 IO-Devices (typical bidirectional, ~ 50 flows per direction)
    - IO-Device with real time data rate of 1 kHz (flow transmission rate)
    - Max e2e latency: 1ms
    - Max hop count: 16

    ➔ Max per hop latency: 62,5µs

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

➔ Overprovisioning of factor 16 by reservation of 50 Streams (based on MSRP) without flow aggregation

Proposal:
Flow Aggregation on Common Stream using Interleaving

Definitions:

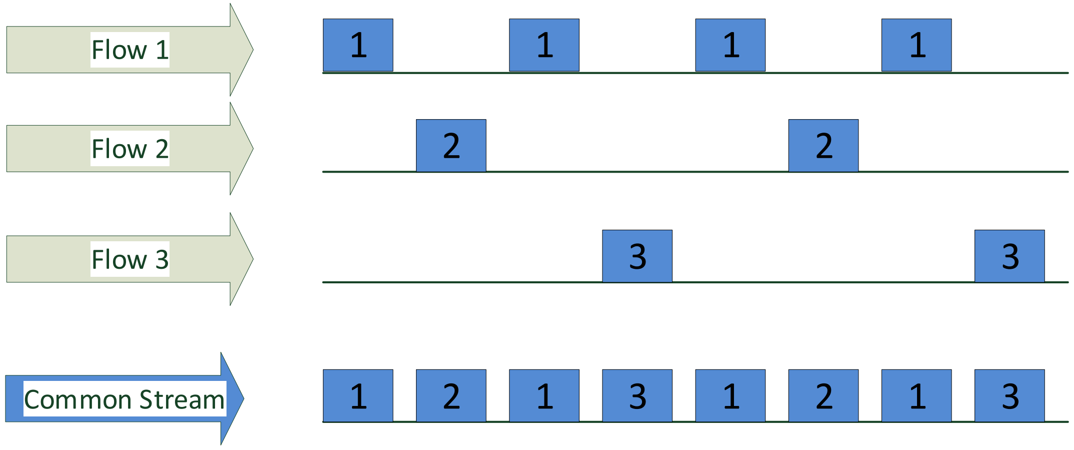
- **Flow Aggregation**
  Combining multiple Flows into one common Stream.

- **Interleaving**
  A method to combine multiple Flows into one common Stream by multiplexing.

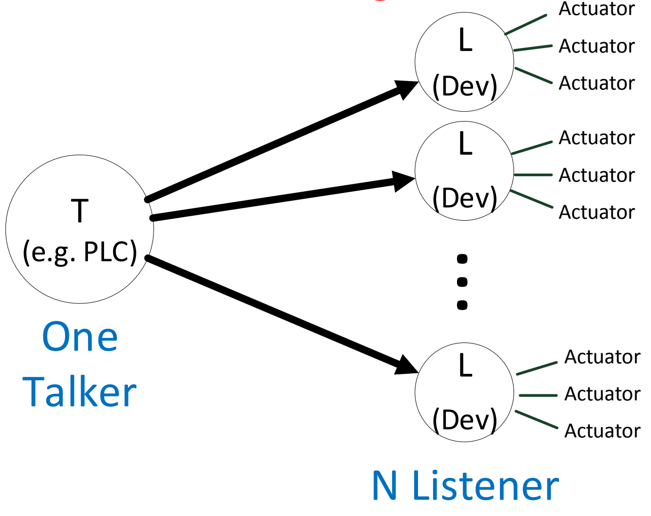
  - **Talker-Interleaving**
    A Talker is responsible for organizing its Flows for multiple Listeners into one common Stream.

  - **Listener-Interleaving**
    A Listener is responsible for organizing its common Stream which consists of multiple Talker Flows.

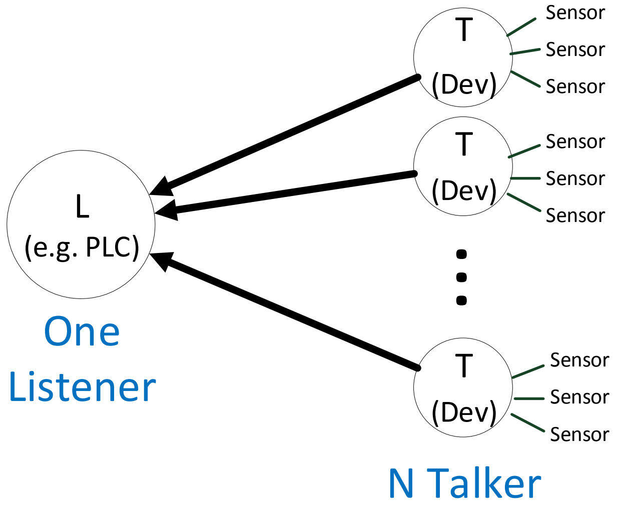
Flow Aggregation using Interleaving



Talker- Interleaving



Listener-Interleaving



Talker-Interleaving

SIEMENS
Ingenuity for life

Stream reservation

- Talker announces a common Stream by sending Talker-Advertise
- Listeners are joining the common Stream by sending Listener-Join

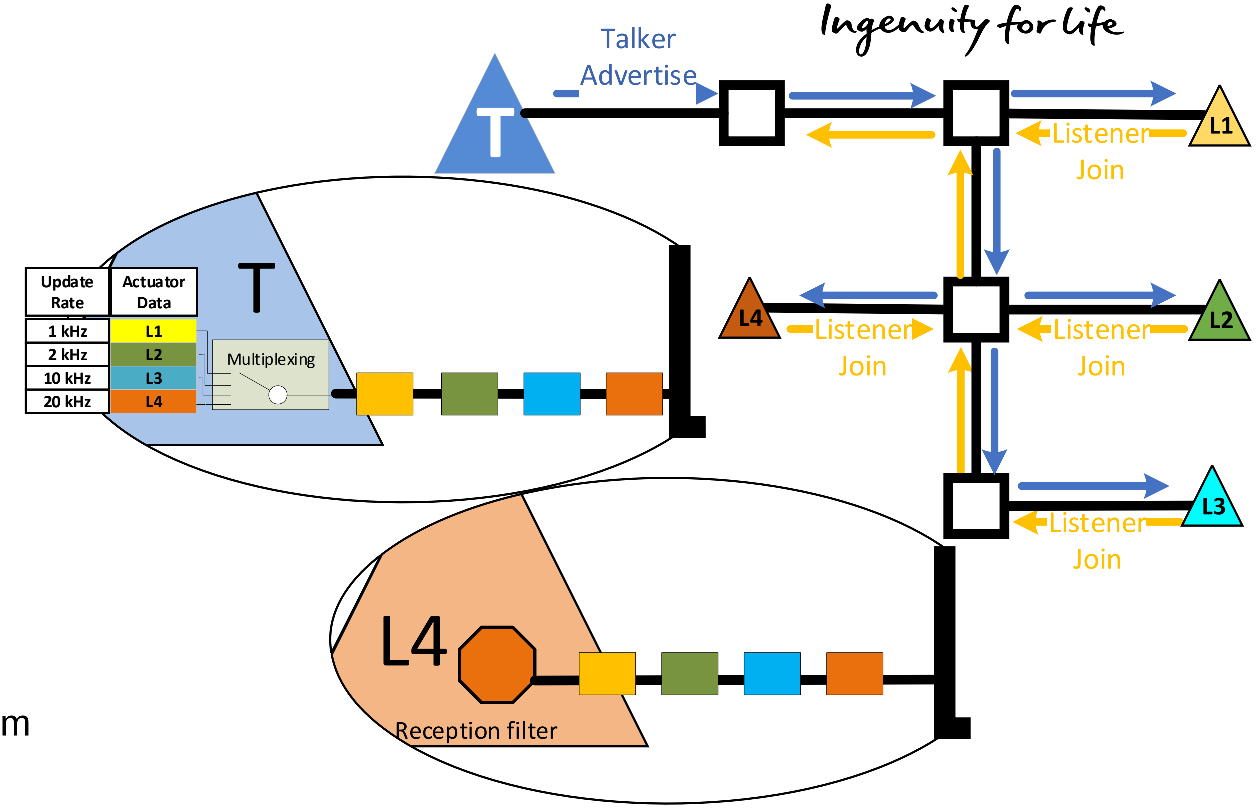
➔ Talker-Interleaving is transparent for Stream reservation

Talker scheduling per common Stream

- Talker collects from Listeners all listener flow specifications by e.g. upper layer protocol (ULP)
- Talker schedules all Listener flow specifications for a common Stream (e.g. using Least-Common-Multiple (LCM) principle)
- Talker (T) transmits the Listener Flows according to their update rats for a common Stream according to the local Talker schedule

➔ There are no timing dependencies among multiple common Streams
➔ Talker-Interleaving is transparent for network data plane
➔ Listener (L4) filters its flow (flow reception filter)

Stream Reservation



Update Rate	Actuator Data
1 kHz	L1
2 kHz	L2
10 kHz	L3
20 kHz	L4

Schedule for common Stream

#Columns = Typically LCM of all listener flow specifications

#Lines = #Listener

Listener_1	0	1	0	…	1
Listener_2	1	1	0	…	0
…	…	…	…	…	…
Listener_N	0	0	1	…	1

Listener-Interleaving

Stream reservation

- Listener announces a Stream by sending Listener-Advertise
- Talkers are joining a Stream by sending Talker-Join

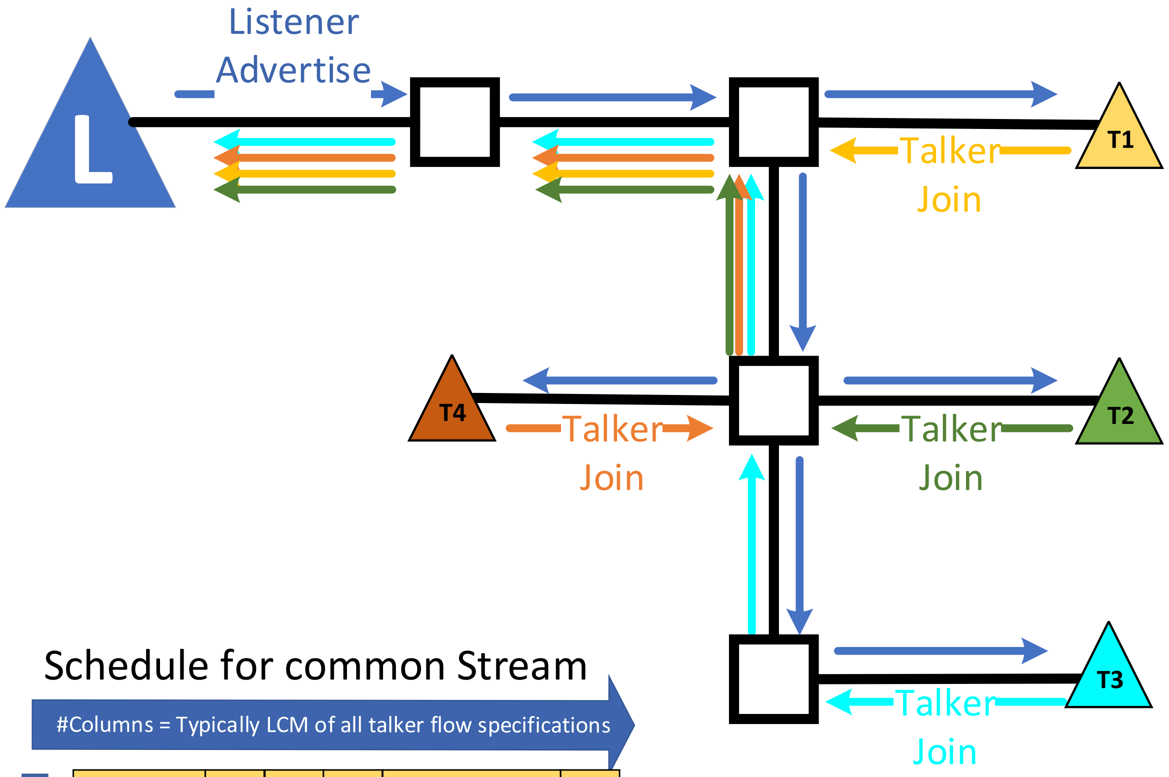
➔ For Listener-Interleaving these two new attributes must be introduced

Listener scheduling per common Stream

- Listener collects from Talkers all talker flow specifications by e.g. upper layer protocol (ULP)
- Listener schedules all talker flow specifications for a common Stream for his reception (e.g. using Least-Common-Multiple (LCM) principle)
- The Listener advertise attribute can be used to distribute the schedule for the common Stream (see example on next slide)
- Each Talker transmits it's talker flow according to the Listener's schedule of the common Stream

➔ Listener-Interleaving is transparent for the network data plane but shaper dependent, supported e.g. by CQF

Stream Reservation



Schedule for common Stream

#Columns = Typically LCM of all talker flow specifications

#Lines = #Talker

TalkerMAC$_1$	0	1	0	…	1
TalkerMAC$_2$	1	1	0	…	0
…	…	…	…	…	…
TalkerMAC$_N$	0	0	1	…	1

Example for Listener-Interleaving

Assumption for data plane:

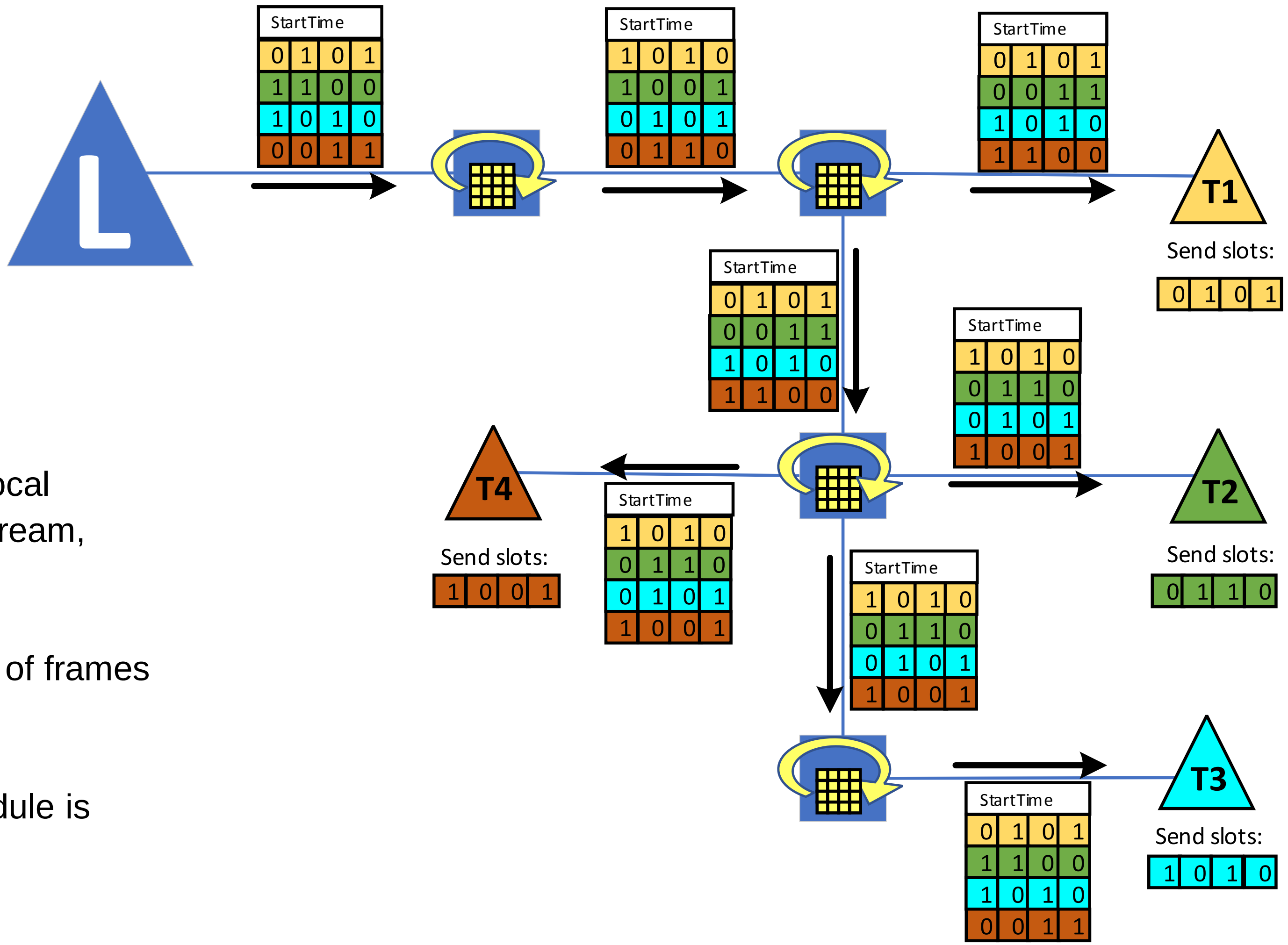
- Traffic shaper: Cyclic Queuing and Forwarding (CQF)
  IEEE Std. 802.1Qch-2017
  - All slots are aligned to each other
  - Per hop cycle time is identical at each bridge and equal to ClassMeasurementInterval
- End-Stations and bridges are synchronized

Proposal:

- Listener-Advertise attribute can be used to distribute the local Listener schedule of multiple talker flows for a common Stream, encoded e.g. as a matrix.

- Every line in the matrix represents one Talker, the number of frames transmitted by the talker is encode in every column.

- For a Stream-Class using CQF as traffic shaper, the schedule is adjusted by every bridge for the next hop.
  - Each column of the matrix is rotated by one.



StartTime			
0	1	0	1
1	1	0	0
1	0	1	0
0	0	1	1

StartTime			
1	0	1	0
1	0	0	1
0	1	0	1
0	1	1	0

StartTime			
0	1	0	1
0	0	1	1
1	0	1	0
1	1	0	0

Send slots:

0	1	0	1

StartTime			
0	1	0	1
0	0	1	1
1	0	1	0
1	1	0	0

StartTime			
1	0	1	0
0	1	1	0
0	1	0	1
1	0	0	1

Send slots:

1	0	0	1

StartTime			
1	0	1	0
0	1	1	0
0	1	0	1
1	0	0	1

StartTime			
1	0	1	0
0	1	1	0
0	1	0	1
1	0	0	1

Send slots:

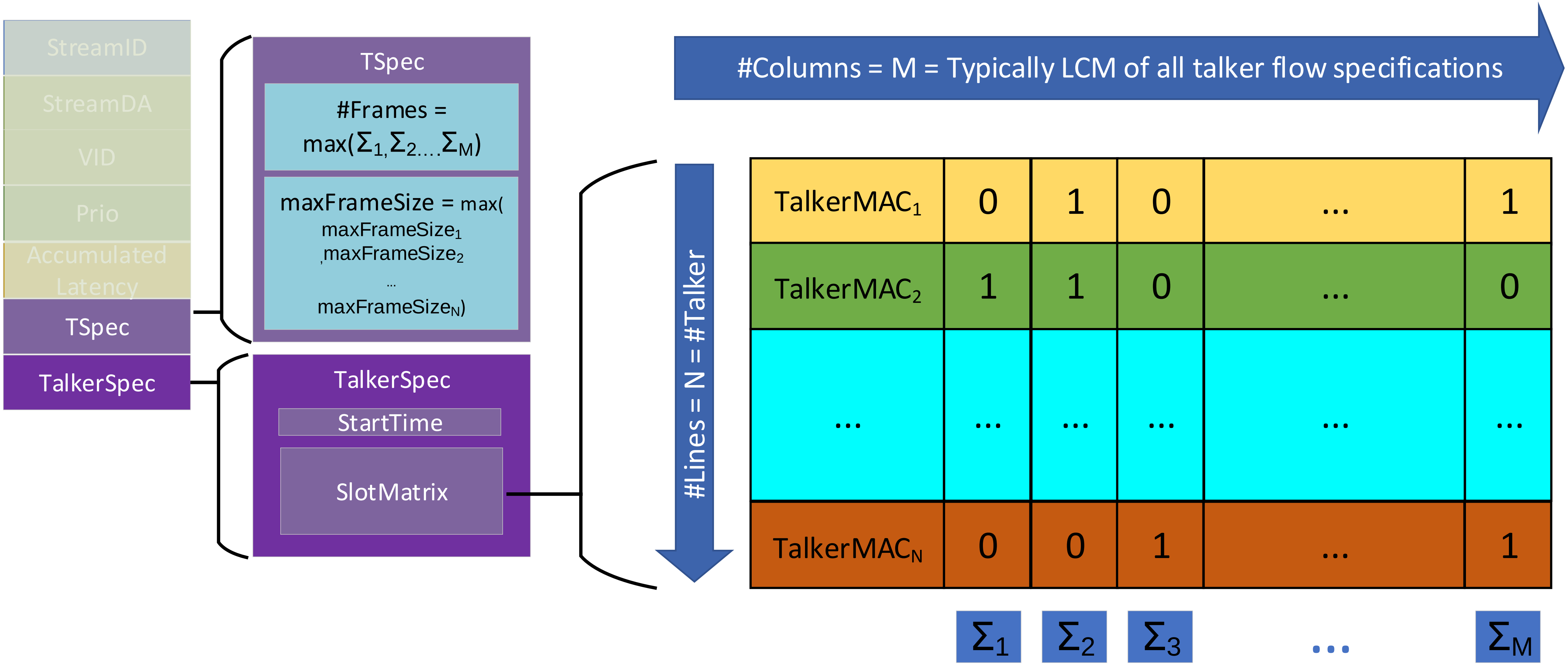
0	1	1	0

StartTime			
0	1	0	1
1	1	0	0
1	0	1	0
0	0	1	1

Send slots:

1	0	1	0

Proposal: Content of Listener-Advertise for Listener-Interleaving


StreamID
StreamDA
VID
Prio
Accumulated Latency
TSpec
TalkerSpec
TSpec
#Frames = $\max(\Sigma_{1,}\Sigma_{2....}\Sigma_M)$
maxFrameSize = max(maxFrameSize_1 ,maxFrameSize_2 … maxFrameSize_N)
TalkerSpec
StartTime
SlotMatrix
#Columns = M = Typically LCM of all talker flow specifications
#Lines = N = #Talker

TalkerMAC_1	0	1	0	…	1
TalkerMAC_2	1	1	0	…	0
…	…	…	…	…	…
TalkerMAC_N	0	0	1	…	1
	Σ_1	Σ_2	Σ_3	…	Σ_M

Thank You!

SIEMENS

Ingenuity for life

Discussion