



HAL
open science

Formal specification of satellite on-board networks requirements

Pierre-Julien Chaine, Marc Boyer, Claire Pagetti, Franck Wartel

► **To cite this version:**

Pierre-Julien Chaine, Marc Boyer, Claire Pagetti, Franck Wartel. Formal specification of satellite on-board networks requirements. 2020. hal-02926971

HAL Id: hal-02926971

<https://hal.science/hal-02926971v1>

Preprint submitted on 1 Sep 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Formal specification of satellite on-board networks requirements

Pierre-Julien CHAINE¹, Marc BOYER², Claire PAGETTI², and Franck WARTEL¹

1 Airbus Defence and Space

Toulouse, France

pierre-julien.chaine@airbus.com, franck.wartel@airbus.com

2 ONERA

Toulouse, France

marc.boyer@onera.fr, claire.pagetti@onera.fr

Abstract

This report proposes a formal introduction to the satellite on-board network requirements. These requirements have been sorted into several categories: *Input Contracts* which describe constraints on the injection of the messages of the flows in the system (network), *Generic Constraints* which describe generic constraints as well as medium specific constraints, *Quality of Service*

Constraints which describe constraints on the reception of messages of the flows in the system and *Precedence Constraints* which describe temporal relationships between flows. One category of constraints have been left behind : *Fault Tolerance Constraints*, they will not be described in this document.

Keywords and phrases Time Sensitive Networking, Ethernet, Embedded Networks, Requirements, Satellite

1 System Definition

This section will define the system we are considering. These definitions are separated into three parts: *Devices and applications*, *Flows* and *Time Sensitive Networking Network*

1.1 Devices and applications

The system we consider, a satellite, is composed of two parts: *Payload* and *Platform*. The *Payload* is the purpose of the satellite, it is the part of the satellite that generates added value for our clients, it is specific to each and every mission. The *platform* is what allows the satellite to achieve its mission. If this part of the satellite does not work properly, the satellite is useless or might even be lost. The platform is composed of all the systems and subsystems that ensure a nominal behaviour of the satellite. This separation *Payload/Platform* is also applied to the satellite network. The network constraints applied to *Payload* flows are soft real time whereas constraints applied to *Platform* flows are hard real time constraints. Let us now describe the devices and applications of our system.

1.1.1 Devices

The first elements of our system are the devices. We start with a known set of devices. Let us name that set \mathcal{D} . The devices ($Dev \in \mathcal{D}$) can be part of four different families:

- *Platform Computing*: The computing devices are where, you name it, the processing of data from sensing devices is made and where commands directed to actuating devices are generated.
- *Sensing*: The sensing devices collect data through analog or digital sensors.
- *Actuating*: The actuating devices execute commands received from a computing device so as to control the attitude and the orbit of the satellite.

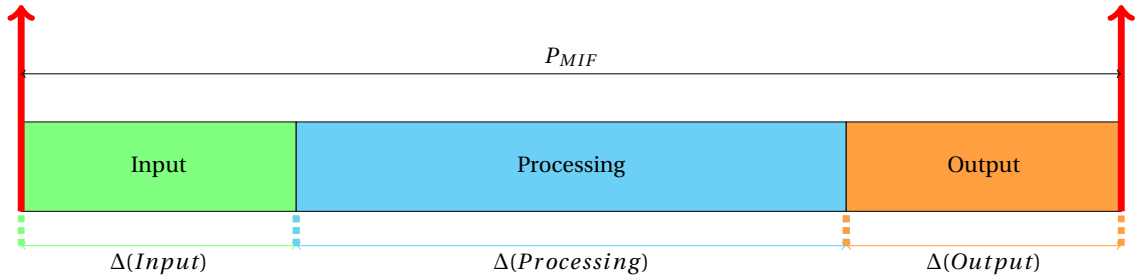
2 Formal specification of satellite on-board networks requirements

21 ■ *Payload (Data and Computing)*: the payload devices collect useful data for the mission, for ex-
 22 ample: antennas and transponders for telecommunication satellites, telescopes and cameras for
 23 Earth/Space observation satellites or basically any type of scientific instruments for scientific mis-
 24 sions.

25 In their behaviour, both *Actuating* and *Sensing* devices act as slaves of the *Computing* devices.

26 Some of the sensing or actuating devices are analog devices and hence are connected to a *RIU* for
 27 *Remote Interface Unit* that acts as controller for these devices. This RIU will also be considered as an
 28 *Actuating* or a *Sensing* node.

29 1.1.2 Applications



■ **Figure 1** Typical application pattern

30 Several time constants are used to define the behaviour of the applications (*Command & Control*,
 31 *Vision Based Navigation*, *Payload Processing*, etc.) running on the computing devices. In fact the
 32 application follows a pattern, named a *cycle*, shown in Fig.1, where a reserved time quantum (δt_1) of its
 33 time is dedicated to gathering input data from sensing devices, then another quantum (δt_2) of its time
 34 is dedicated to processing this data and one last quantum (δt_3) is dedicated to sending output data to
 35 actuating devices according to the output of the processing. Let us name $\Delta(Input)$, the duration of the
 36 *input* part, $\Delta(Processing)$, the duration of the *processing* part, $\Delta(Output)$, the duration of the *output*
 37 part. We also denote $Start(Part)$ and $End(Part)$, the time instant where the part starts (resp. ends).
 38 We can now define the *MIF Period* P_{MIF} such as :

$$\begin{cases}
 \Delta(Input) + \Delta(Processing) + \Delta(Output) = P_{MIF} \\
 \Delta(Input) = \delta t_1 * P_{MIF} \\
 \Delta(Processing) = \delta t_2 * P_{MIF} \\
 \Delta(Output) = \delta t_3 * P_{MIF} \\
 End(Input) = Start(Processing) \\
 End(Processing) = Start(Output)
 \end{cases} \quad (1)$$

40 ► **Rule 1.** Starting now, all values for periods in this document will be expressed in milliseconds.

41 Usual values of P_{MIF} are $125ms$, $62.5ms$ or even $31.25ms$.

42 The scheme of duration P_{MIF} is repeated k times $\in \mathbb{N}$ leading to P_{MAF} so that:

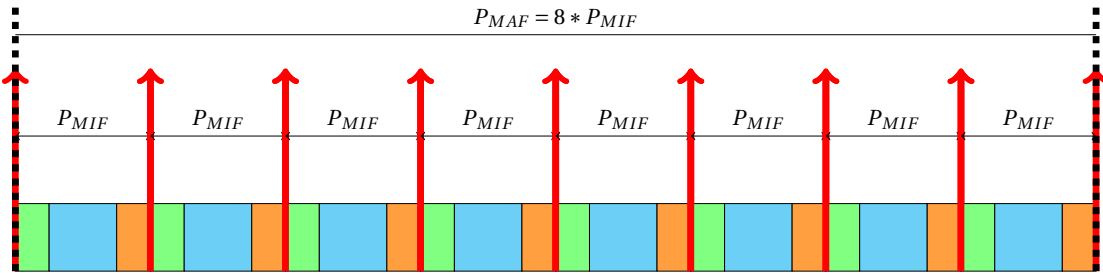
$$43 \quad P_{MAF} = k * P_{MIF} \quad (2)$$

44 ► **Hypothesis 1.** In our system, $P_{MAF} = 1000$ hence $k = \frac{1000}{P_{MIF}}$.

45
 46 This hypothesis is illustrated in Fig. 2. Tab. 1 gives the value of k for the usual values of P_{MIF} .

k	P_{MIF}
8	125ms
16	62.5ms
32	31.25ms

■ **Table 1** Usual values of k and P_{MIF}



■ **Figure 2** Link $P_{MIF} - P_{MAF}$ with $k = 8$

1.1.3 Motivating Example

All along this report, we will use a motivating example to illustrate the given definitions, constraints, etc. Every now and then, we will use this example or add to it so as to ease the understanding of the concepts of this document. This example, for now, is composed of three devices:

- One Computing Device that we will name *OBC*.
- One Sensing Device that we will call *High Performance Sensor*
- One Actuating Device that we will call *High Performance Actuator*, that is, in reality, connected through a *RIU - Remote Interface Unit*

The connexions between devices are illustrated in Fig.3. Two communication links are illustrated in the figure. Communication links will be introduced later in this technical note.

1.2 Flows

The applications running on the computing devices communicate with the sensing and actuating devices with messages. These messages are gathered under the concept of flows.

► Definition 1. Flow

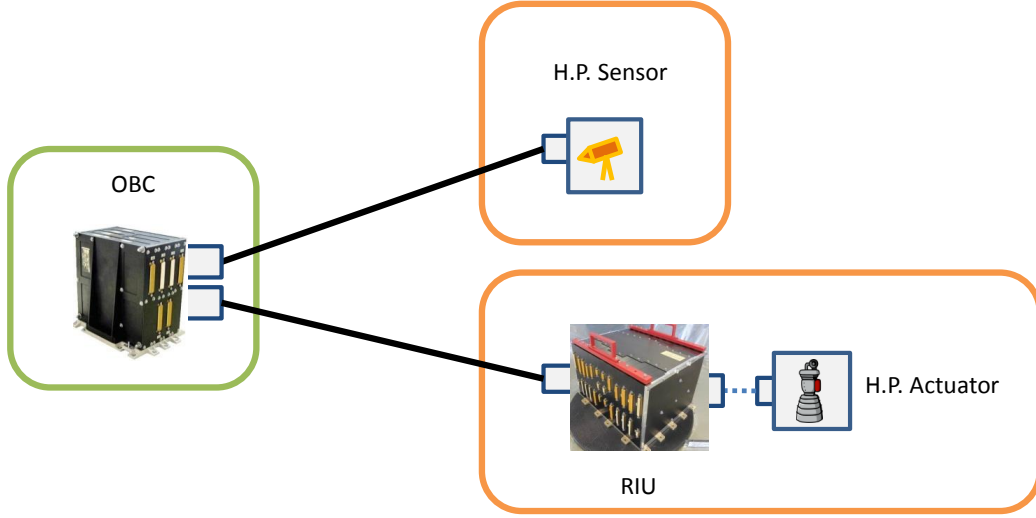
A flow is a unidirectional sequence of messages from one sender to one or several receivers. Let us denote f a flow, and \mathbb{F} the set of flows of our system. A flow is characterized with the following tuple:

$$\forall f \in \mathbb{F} \langle f.Src_f, f.Dest_f, f.MaxDataSize, f.r_f \rangle \quad (3)$$

Where :

- $f.Src_f$ is the device from which the messages are generated and emitted
- $f.Dest_f$ are the device(s) to whom the messages are sent.
- $f.MaxDataSize$ is the maximum data size the data sent by the flow in one applicative message. We express it in bytes.

4 Formal specification of satellite on-board networks requirements



■ **Figure 3** Motivating Example with three devices

71 ■ $f.r_f$ is our way to express the periodicity of the flow (see Eqn. 5), $r_f \in \mathbb{Z}$.

72 ► **Hypothesis 2.** In our system, we will use an Ethernet medium and we will consider that :

$$73 \quad \forall f \in \mathbb{F}, f.MaxDataSize \leq MTU_{Ethernet} \quad (4)$$

74 This means, that one applicative message of a flow will lead to one message in the network. The defini-
75 tions, properties and constraints in the rest of the document apply this hypothesis. If this hypothesis
76 is later relaxed, the definitions, properties and constraints of the document will have to be slightly
77 redefined.

78 The period P_f of flow f , is linked with its ratio r_f with the following equation:

$$79 \quad \forall f \in \mathbb{F}, \begin{cases} r_f < -1 & \Rightarrow P_f = |r_f| * P_{MIF} \\ r_f > 0 & \Rightarrow P_f \iff r_f \text{ messages per } P_{MIF} \\ r_f = 0 & \Rightarrow P_f = NA (f \text{ is non-periodic}) \end{cases} \quad (5)$$

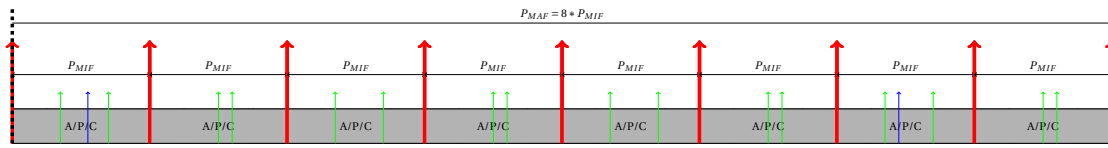
80 Figuratively, if $r_f > 0$, flow f sends r_f messages during one P_{MIF} , hence $k * r_f$ messages during
81 P_{MAF} . If $r_f < -1$, flow f sends one message during $|r_f| * P_{MIF}$, hence $\frac{k}{|r_f|}$ messages during P_{MAF} .
82 The concept of ratio is illustrated in Fig. 4 with, in blue a flow with $r_f = -4$ and in green a flow with $r_f = 2$.

84 ► **Property 1.** Link $r_f - k$

$$85 \quad \begin{aligned} r_f < -1 & \Rightarrow r_f \mid k \\ \text{hence } r_f < -1 & \Rightarrow lcm(|r_f|, k) = k \end{aligned} \quad (6)$$

86 ► **Hypothesis 3.** In our system we only consider periodic flows and a flow shall at least send one
87 message every P_{MAF} i.e. :

$$88 \quad r_f \in [-k, -1] \cup [1, +\infty[\quad (7)$$



■ **Figure 4** Ratio concept with two flows ($r_{f1} = 2$, $r_{f2} = -4$) and $k = 8$

89 1.2.1 Motivating Example: Adding Flows

90 In our motivating example (cf. 1.1.3), the applications running on the devices communicate with three
91 flows. Starting now, the convention used for naming flows is the following:

92 ► **Rule 2.** Naming Flows

93 A flow originating from Src_f and going to $Dest_f$ will be named:

$$94 \quad f_Src_f_Dest_f_ID \quad (8)$$

95 Where ID is a user defined identifier used to distinguish several flows with the same senders and
96 receivers.

97 According to R. 2, the flows, represented in Fig. 5, are the following:

- 98 ■ $f_OBC_HPActuator_1$
- 99 ■ $f_OBC_HPSensor_1$
- 100 ■ $f_HPSensor_OBC_1$

101 For each flow, we give below the tuple that characterizes them:

- 102 ■ $f_OBC_HPActuator_1 : \langle OBC, HPActuator, 1500, 1 \rangle$,
- 103 ■ $f_OBC_HPSensor_1 : \langle OBC, HPSensor, 1500, 1 \rangle$,
- 104 ■ $f_HPSensor_OBC_1 : \langle HPSensor, OBC, 1500, 1 \rangle$,

105 The following subsections will introduce the different contracts identified for the flows, grouped in
106 four categories: *Input Contracts*, *Generic Constraints*, *QoS Constraints*, *Precedence Constraints*.. This
107 constraints will often be called *flow constraints*. For every constraint, an example will illustrate the
108 constraint being satisfied. In these figures, the colors representing the Input/Processing/Output parts
109 have been removed for the sake of readability of the flow constraints.

110 1.2.2 Input Contract

111 In this paragraph, we will introduce the *Input contracts* identified for our system. Three *Input contracts*
112 are identified for our system: the *Periodic Message Production* constraint, the *Injection Zone* constraint
113 and the *Controlled Input Traffic Shape* constraint.

114
115 ► **Rule 3.** Emission and Reception dates

116 In this document, all the Emission and Reception dates are at applicative level.

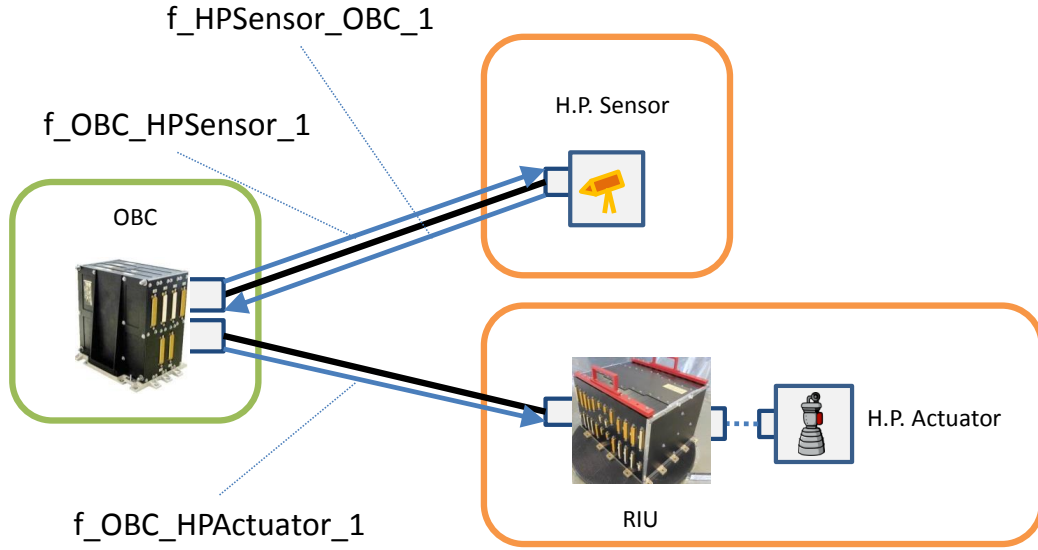
117 ► **Rule 4.** Emission and Reception dates in figures

118 Starting now, in this document, we will materialize:

- 119 ■ The emission date of a message by an upward facing arrow
- 120 ■ The reception date of a message by a downward facing arrow

121 This rule will help readability of figures in particular if several messages are present in a single cycle.

6 Formal specification of satellite on-board networks requirements



■ **Figure 5** Adding three flows to the motivating example

122 ► **Constraint Template 1.** Periodic message production

123 Let $f \in \mathbb{F}$ be a flow such that $r_f \neq 0$. Let f_l be the l th message of flow f (from device Src_f). Let us denote

124 $\forall l \in \mathbb{N}, T_p(f_l)$ the emission date of f_l .

$$\begin{aligned}
 & f \models \text{PeriodicMessageProduction} \\
 & \stackrel{\text{def}}{=} \\
 & \forall l \in \mathbb{N}, \begin{cases} -k \leq r_f < -1 & \implies l * P_f \leq T_p(f_l) < (l+1) * P_f \\ r_f > 0 & \implies \lfloor \frac{l}{r_f} \rfloor * P_{MIF} \leq T_p(f_l) < (1 + \lfloor \frac{l}{r_f} \rfloor) * P_{MIF} \end{cases} \quad (9)
 \end{aligned}$$

127 This first constraint defines a window for every message of a periodic flow (i.e. $r_f \neq 0$ in which the
 128 message of the flow shall be emitted. It is illustrated in Fig. 6. In blue, we have represented a flow with
 129 $r = -4$, and in green, a flow with $r = 2$.

130 We define $Refcycle(f_l)$, a reference date for f_l such as:

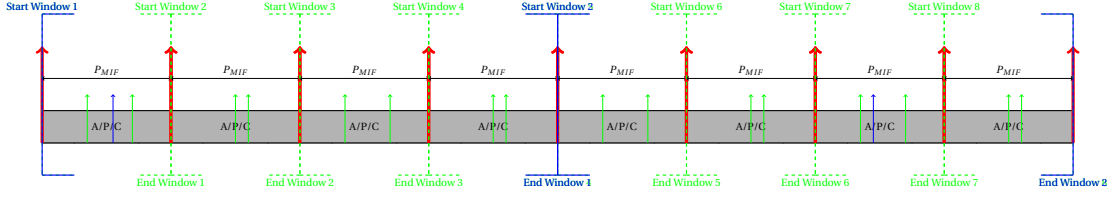
131
 132 ► **Definition 2.** Reference Date

$$\forall f \in \mathbb{F} \text{ with } r_f \neq 0, \forall l \in \mathbb{N}, \begin{cases} -k \leq r_f < -1 & \implies Refcycle(f_l) = l * P_f \\ r_f > 0 & \implies Refcycle(f_l) = \lceil \frac{l}{r_f} \rceil * P_{MIF} \end{cases} \quad (10)$$

134 The concept of *reference date* is also illustrated in Fig. 6 by the blue and green brackets.

135
 136 In our system, one may want to reduce even more the emission window of the flows. That is why a
 137 second constraint is introduced: the *Injection Zone* constraint.

138



■ **Figure 6** Periodic Message Production Constraints, with 2 flows ($r_{f_1} = 2, r_{f_2} = -4$) and $k = 8$

139 ► **Constraint Template 2.** $\text{InjectionZone}(\text{EarliestProdOffset}, \text{LatestProdOffset})$
 140 Let $f \in \mathbb{F}$ be a flow with $r_f \neq 0$. Let f_l be the l th message of flow f (from device Src_f). Let us denote
 141 $\forall l \in \mathbb{N}, T_p(f_l)$ the emission date of f_l . Flow f satisfies Injection Zone constraints if all messages of the
 142 flows are emitted between their earliest and latest possible emission dates i.e.:

$$f \models \text{InjectionZone}(\text{EarliestProdOffset}, \text{LatestProdOffset})$$

$$\stackrel{\text{def}}{=} \forall l \in \mathbb{N}, \mathbf{Prod}_{\text{early}}(f_l) \leq T_p(f_l) \leq \mathbf{Prod}_{\text{late}}(f_l)$$

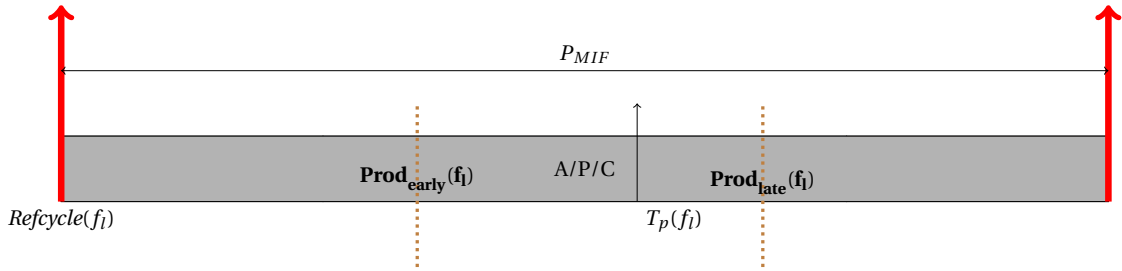
Where:

$$\left\{ \begin{array}{l} \text{EarliestProdOffset} \geq 0 \\ \text{EarliestProdOffset} \leq \text{LatestProdOffset} < \max(P_{MIF}, \text{Refcycle}(f_{l+1}) - \text{Refcycle}(f_l)) \\ \mathbf{Prod}_{\text{early}}(f_l) = \text{Refcycle}(f_l) + \text{EarliestProdOffset} \\ \mathbf{Prod}_{\text{late}}(f_l) = \text{Refcycle}(f_l) + \text{LatestProdOffset} \end{array} \right. \quad (11)$$

144 ► **Property 2.** In all cases:

$$\text{Refcycle}(f_l) \leq \mathbf{Prod}_{\text{early}}(f_l) \leq T_p(f_l) \leq \mathbf{Prod}_{\text{late}}(f_l) < \max(\text{Refcycle}(f_l) + 1, \text{Refcycle}(f_{l+1})) \quad (12)$$

146 This constraint is illustrated in Fig. 7.



■ **Figure 7** Injection Zone constraint with $\text{EarliestProdOffset} = \frac{P_{MIF}}{3}$ and $\text{LatestProdOffset} = \frac{2 * P_{MIF}}{3}$

147 We raise the readers' awareness on the fact that in Fig. 7, $\text{EarliestProdOffset}$ and
 148 LatestProdOffset have been represented in the same P_{MIF} cycle. However, as Constraint 2 states,
 149 $\text{EarliestProdOffset}$ and LatestProdOffset can be in separate cycles (if $\text{LatestProdOffset} >=$
 150 P_{MIF}).

151 One constraint shall also be mentioned: in our system we shall ensure an ordered emission of the
 152 messages within a flow. This translates in the *Ordered Emission* constraint.

153

154 ► **Constraint Template 3.** Ordered Emission
 155 Let $f \in \mathbb{F}$ be a flow. Let f_l, f_m two messages of f .

$$\begin{aligned}
 & f \models \text{OrderedEmission} \\
 & \stackrel{\text{def}}{\equiv} \\
 & \forall l, m \in \mathbb{N}, l < m, T_p(f_l) < T_p(f_m)
 \end{aligned} \tag{13}$$

157 The last constraint that we have identified is motivated by applications running on payload nodes.

158 ► **Constraint Template 4.** Controlled Input Traffic Shape

159 Let $f \in \mathbb{F}$ be a flow. Let R be a rate, in Mbits/s. Let B be a burst, in bytes. Flow f satisfies the Controlled
 160 Input Traffic Shape if and only if f input traffic is within a token bucket of parameters (R, B) i.e.:

$$\begin{aligned}
 & f \models \text{ControlledInputTrafficShape}(R, B) \\
 & \stackrel{\text{def}}{\equiv} \\
 & \forall l, m \in \mathbb{N}, \sum_{i=l}^m \text{size}(f_i) \leq R * (T_p(f_m) - T_p(f_l)) + B
 \end{aligned} \tag{14}$$

162 1.2.3 Generic Constraints

163 In this paragraph, we will introduce the *Generic Constraints* for our system. Two constraints are
 164 identified, for now, for our system: the *Reception after Emission* constraint and the *Ethernet Frame Size*
 165 constraint.

166

167 ► **Constraint Template 5.** Reception after Emission

168 Let $f \in \mathbb{F}$ be a flow. Let $T_p(f_l), T_r(f_l)$ be the production (resp. reception) date of the l th message of
 169 f (i.e. f_l). f respects the *Reception after Emission* constraints, if and only if, for all its messages, the
 170 reception date is greater than the emission date i.e.:

$$\begin{aligned}
 & \forall f \in \mathbb{F}, \\
 & f \models \text{ReceptionAfterEmission} \\
 & \stackrel{\text{def}}{\equiv} \\
 & \forall l \in \mathbb{N}, T_p(f_l) < T_r(f_l);
 \end{aligned} \tag{15}$$

172 This constraint is naturally extended to multicast flows as follows:

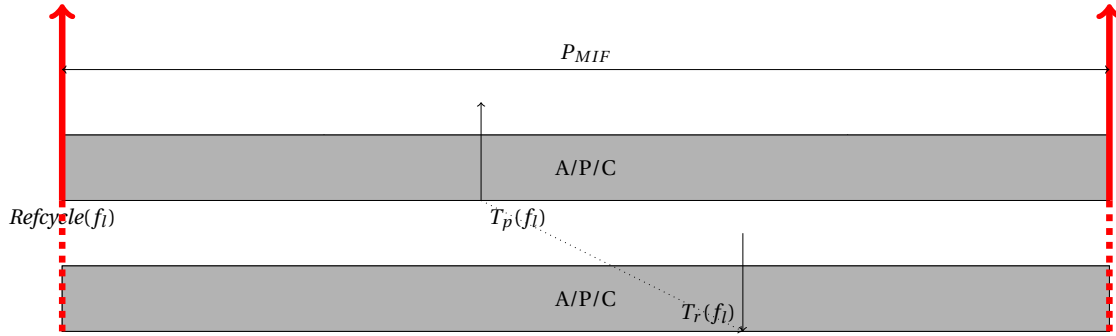
$$\begin{aligned}
 & \forall f \in \mathbb{F}, \\
 & f \models \text{ReceptionAfterEmission} \\
 & \stackrel{\text{def}}{\equiv} \\
 & \forall d \in \text{Dest}_f, \forall l \in \mathbb{N}, T_p(f_l) < T_r(f_l, d)
 \end{aligned} \tag{16}$$

174 In Fig. 8, we have represented $T_p(f_l)$ and $T_r(f_l)$ respecting the previous constraint.

175 ► **Constraint Template 6.** Ethernet Frame Size

176 Let $f \in \mathbb{F}$ be a flow. Let f_l be the l th message of f . Let $\text{Frame}(f_l)$ be the Ethernet frame corresponding
 177 to the l th message of f . f respects the *Ethernet Frame Size* constraint, if and only if, for all its messages,
 178 the size of their associated frames is lower or equal to the Ethernet MTU i.e.:

$$\begin{aligned}
 & \forall f \in \mathbb{F}, \\
 & f \models \text{EthernetFrameSize} \\
 & \stackrel{\text{def}}{\equiv} \\
 & \forall f_l, \text{size}(\text{Frame}(f_l)) < \text{Ethernet}_{\text{MTU}}
 \end{aligned} \tag{17}$$



■ **Figure 8** Reception After Emission constraints on f_l

1.2.4 Quality of Service Constraints

Several *Quality of Service constraints* are identified for our system. One flow can be subject to any combination of this *Quality of Service constraints*.

As well as the order of emission of the messages of the flows was constrained, the reception of said messages is also constrained. This translates into *Ordered Delivery constraint*.

► **Constraint Template 7.** Ordered Delivery

Let $f \in \mathbb{F}$ be a flow. Let f_l, f_m two messages of f .

$$\begin{aligned}
 f \models \text{OrderedDelivery} \\
 \stackrel{\text{def}}{\equiv} \\
 \forall l, m \in \mathbb{N}, l < m, T_r(f_l) < T_r(f_m)
 \end{aligned} \tag{18}$$

► **Constraint Template 8.** Implicit Deadlines

Let $f \in \mathbb{F}$ be a flow such that $r_f \neq 0$.

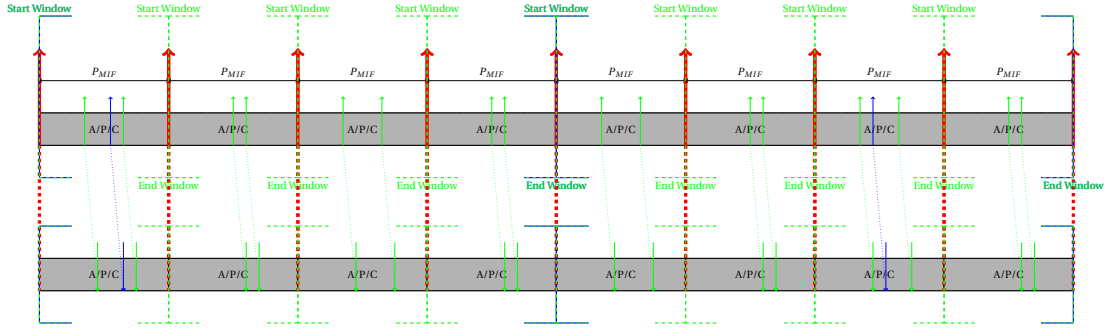
$$\begin{aligned}
 f \models \text{ImplicitDeadlines} \\
 \stackrel{\text{def}}{\equiv} \\
 \forall l \in \mathbb{N}, \begin{cases} -k \leq r_f < -1 & \implies l * P_f \leq T_r(f_l) < (l + 1) * P_f \\ r_f > 0 & \implies \lfloor \frac{l}{r_f} \rfloor * P_{MIF} \leq T_r(f_l) < (1 + \lfloor \frac{l}{r_f} \rfloor) * P_{MIF} \end{cases}
 \end{aligned} \tag{19}$$

This constraints implies that the latest reception date of f_l of flow f must be received before the beginning of the emission window of f_{l+1} . It is illustrated in Fig. 9.

Again, in our system, one may want to reduce even more the reception window for the messages of a flow. This translates into *TimeZone constraints*.

► **Constraint Template 9.** $\text{TimeZone}(\text{EarliestRecOffset}, \text{LatestRecOffset})$

Let $f \in \mathbb{F}$ be a flow such that $r_f \neq 0$. Let f_l be the l th message of f . Let $T_r(f_l, d)$ be the reception date of



■ **Figure 9** ImplicitDeadlines constraint, with 2 flows ($r_{f1} = 2, r_{f2} = -4$) and $k = 8$

201 f_l at destination d .

$$f \models \text{TimeZone}(\text{EarliestRecOffset}, \text{LatestRecOffset})$$

$$\stackrel{\text{def}}{=} \equiv$$

$$\forall d \in \text{Dest}_f, \forall l \in \mathbf{N}, \text{Rec}_{\text{early}}(f_l) \leq T_r(f_l, d) \leq \text{Rec}_{\text{late}}(f_l)$$

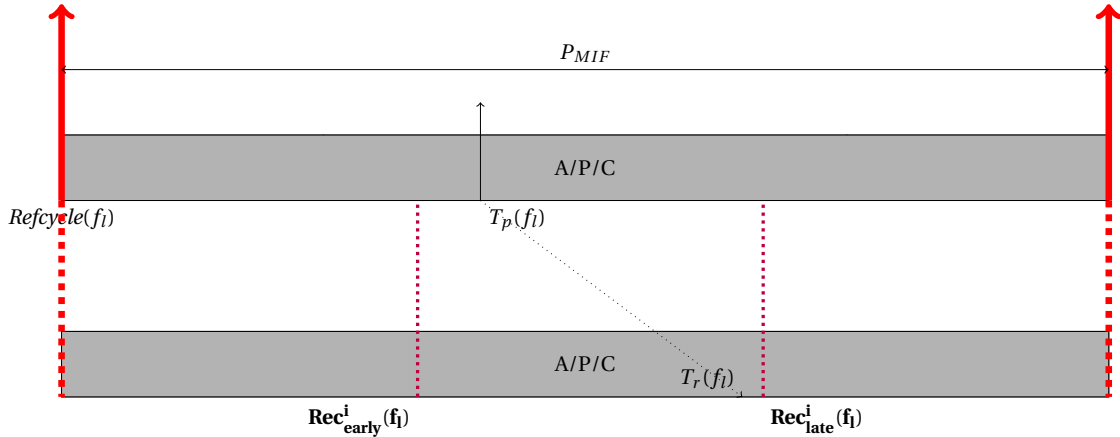
Where :

$$\left\{ \begin{array}{l} \text{EarliestRecOffset} \geq 0 \\ \text{EarliestRecOffset} \leq \text{LatestRecOffset} < \max(P_{MIF}, \text{Refcycle}(f_{l+1}) - \text{Refcycle}(f_l)) \\ \text{Rec}_{\text{early}}(f_l) = \text{Refcycle}(f_l) + \text{EarliestRecOffset} \\ \text{Rec}_{\text{late}}(f_l) = \text{Refcycle}(f_l) + \text{LatestRecOffset} \end{array} \right. \quad (20)$$

203 **Remark:** In all cases:

$$204 \quad \text{Refcycle}(f_l) \leq \text{Rec}_{\text{early}}(f_l) \leq T_r(f_l, d) \leq \text{Rec}_{\text{late}}(f_l) < \max(\text{Refcycle}(f_l) + 1, \text{Refcycle}(f_{l+1})) \quad (21)$$

205 This constraint is illustrated in Fig. 10.



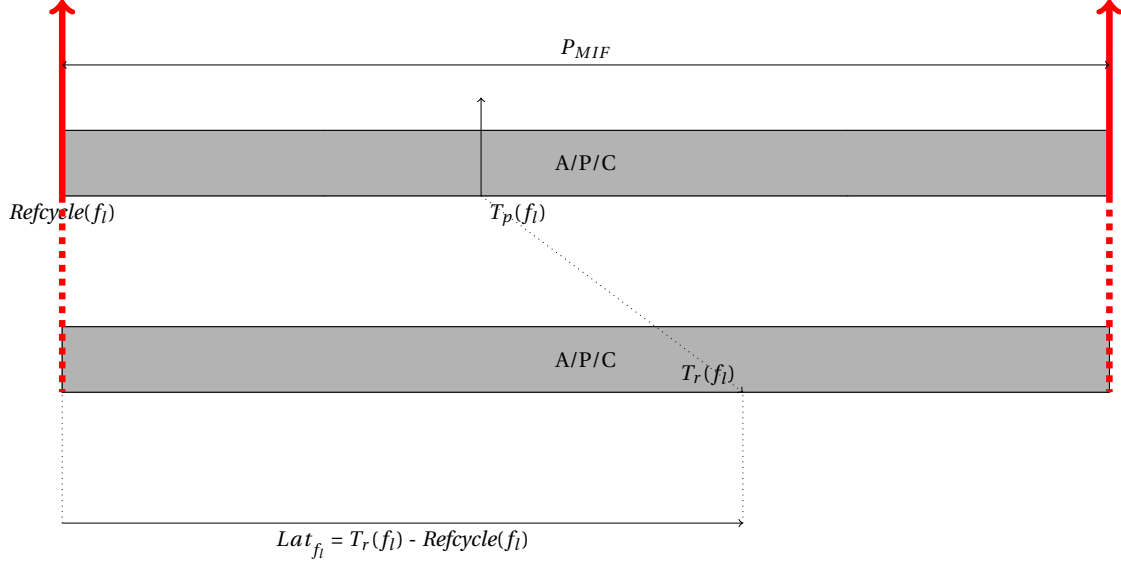
■ **Figure 10** Timezone constraint with $\text{EarliestRecOffset} = \frac{P_{MIF}}{3}$ and $\text{LatestRecOffset} = \frac{2 * P_{MIF}}{3}$

206 Now that the emission and reception date of the l th message of flow f have been defined
 207 (respectively $T_p(f_l)$ and $T_r(f_l, d)$), the concept of latency can be defined. Usually, the network
 208 end-to-end latency of the l th message of flow f would be defined as the difference between the
 209 reception date and the emission date of f_l . However, we define the latency of f_l as follows:

210

211 ► **Definition 3.** Latency

$$212 \quad \forall d \in Dest_f, \forall l \in \mathbb{N}, Lat_{f_l, d} = |T_r(f_l, d) - Refcycle(f_l)| \quad (22)$$



■ **Figure 11** Latency concept in our system

213 This definition is illustrated in Fig. 11. With this new definition, instead of considering the time spend
 214 in the network by f_l , we consider the age of f_l compared to its reference date $Refcycle(f_l)$. The *Jitter*
 215 *constraint* can now be defined.

216
 217 ► **Constraint Template 10.** $Jitter(\alpha)$

218 Let $f \in \mathbb{F}$ be a flow, let f_l, f_m be two messages of f . Let α be a positive constant.

$$219 \quad f \models Jitter(\alpha) \stackrel{def}{=} \begin{cases} -k \leq r_f < -1 & \implies \forall l, m \in \mathbb{N}, |Lat_{f_l} - Lat_{f_m}| \leq \alpha \\ r_f > 0 & \implies \forall l, m \in \mathbb{N}, l \equiv m \pmod{r_f}, |Lat_{f_l} - Lat_{f_m}| \leq \alpha \end{cases} \quad (23)$$

220 Hence, in our system, jitter is understood (like in classic networking system) as the variability of the
 221 latency of the frames of the flow. However, the definition of latency is different.

222 This constraint is illustrated in Fig. 12.

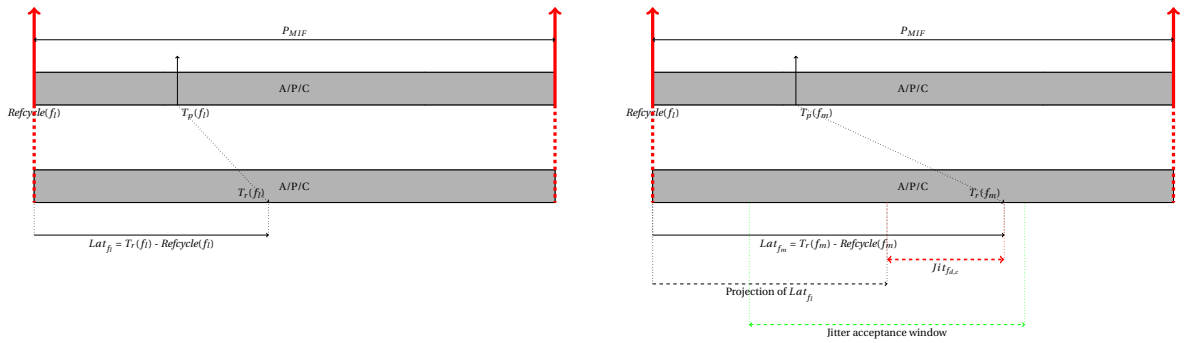
223 As flows may have more than one destination (gathered in $Dest_f$), we added one constraint to add
 224 a relation between the reception dates on the different devices in $Dest_f$. This constraint is the *Sync*
 225 *constraint*.

226 ► **Constraint Template 11.** $Sync(\epsilon)$

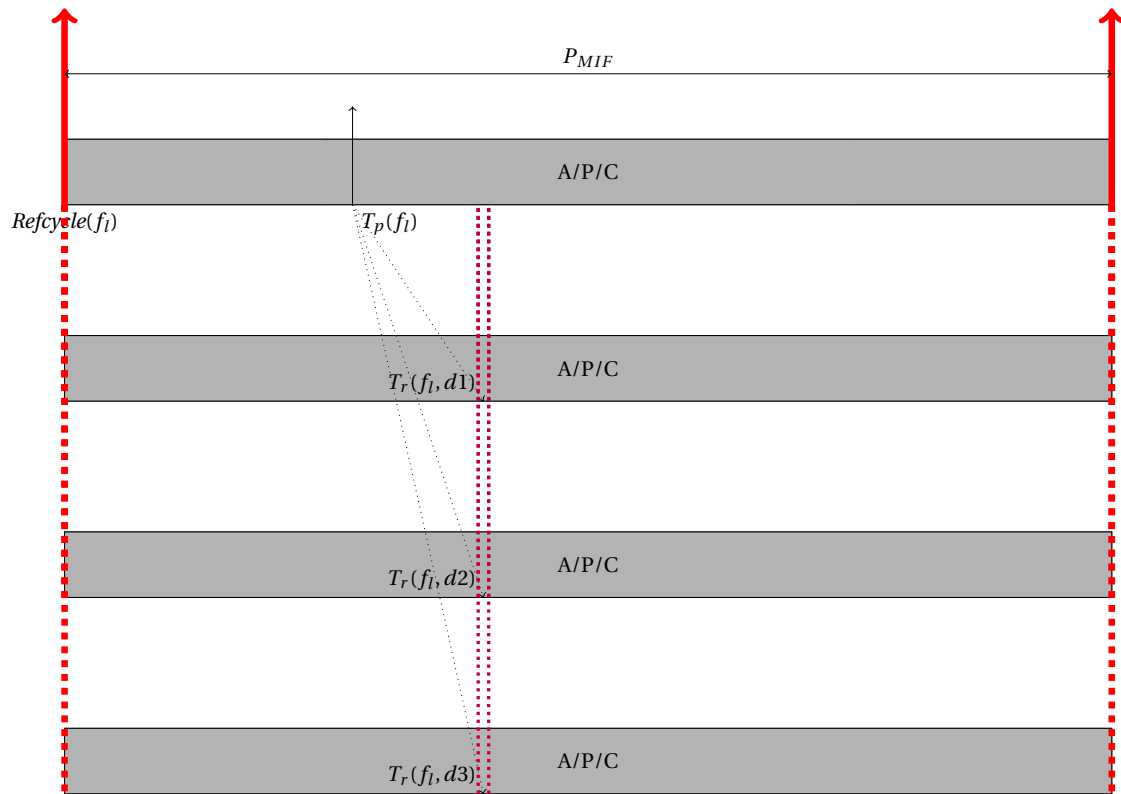
227 Let $f \in \mathbb{F}$ be a flow, let f_l be the l th message of f , let $Dest_f$ be the list of recipients of f .

$$228 \quad f \models Sync(\epsilon) \stackrel{def}{=} \forall l \in \mathbb{N}, \forall x, y \in Dest_f, |T_r(f_l, x) - T_r(f_l, y)| < \epsilon \quad (24)$$

229 This constraint is illustrated in Fig. 13. The two dotted red bars represented the acceptance window ϵ .



■ **Figure 12** Jitter visualisation with two messages f_l and f_m



■ **Figure 13** Sync constraint with 3 recipients for flow f

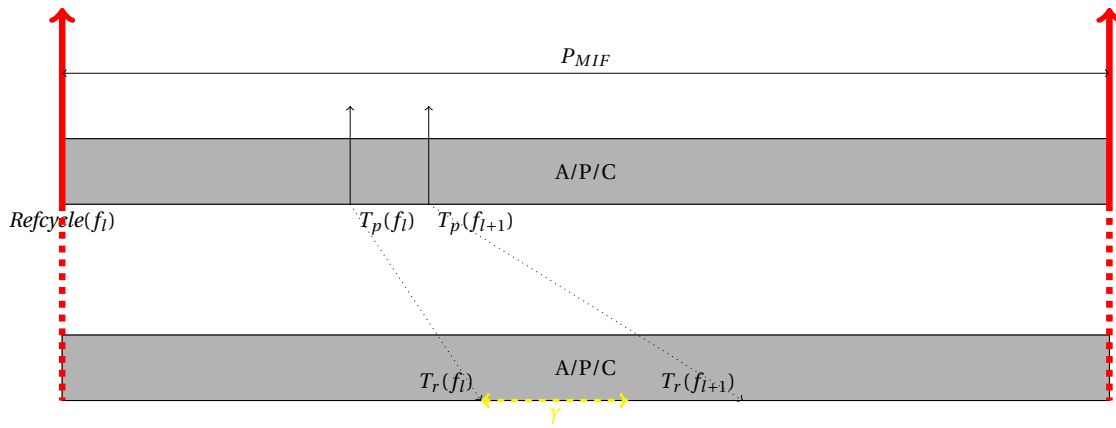
The *Quality of Service constraints* identified up to now in this document describe constraints that would be qualified of *real-time* constraints. The last constraint of this paragraph introduces a relation between two consecutive messages of a flow.

► **Constraint Template 12.** Minimum Space(γ)

Let $f \in \mathbb{F}$, be a flow, let f_i, f_{i+1} be two consecutive messages of f . Let γ be a positive constant.

$$\begin{aligned}
 f \models \text{MinimumSpace}(\gamma) \\
 \stackrel{\text{def}}{\equiv} \\
 \forall l \in \mathbb{N}, T_r(f_{l+1}) - T_r(f_l) \geq \gamma
 \end{aligned}
 \tag{25}$$

This constraint is illustrated in Fig. 14.



■ **Figure 14** Constraint Minimum Space(γ) for flow f

This constraint allows to define a minimum space between the reception of two consecutive messages of a flow regardless of their emission date.

► **Constraint Template 13.** [Option 1] Bounded Arrival For Processing (α_d)

Let $f \in \mathbb{F}$ be a flow. Let α_d be the expected number of messages of flow f to be received during duration d . Let us assume that when the l th message of f , f_l , is received, it is immediately processed during duration $\delta_{process}(f_l)$. Flow f satisfies the Bounded Arrival For Processing constraint if and only if the last message of the set of received messages is received early enough to be processed before the end of the cycle i.e.:

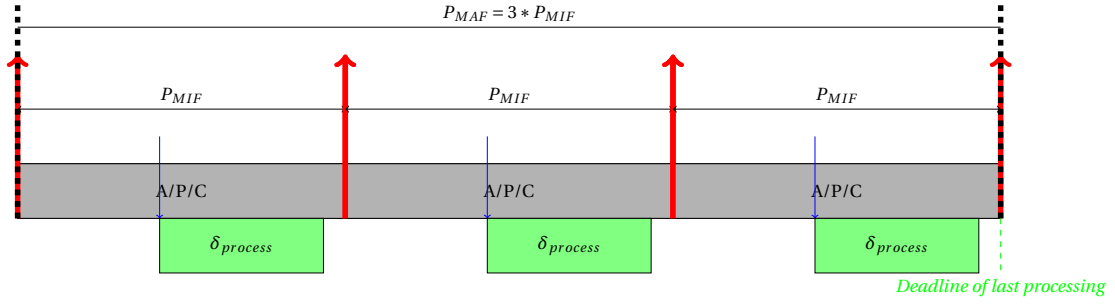
$$\begin{aligned}
 f \models \text{BoundedArrivalForProcessing}(\alpha_{P_{MAF}}) \\
 \forall l \in \mathbb{N}, \\
 T_r(f_{(l+1)*\alpha_{P_{MAF}}-1}) \leq \text{Recycle}(f_{l*\alpha_{P_{MAF}}}) + P_{MAF} - \delta_{process}(l * \alpha_{P_{MAF}})
 \end{aligned}
 \tag{26}$$

Remark: In order for this constraint to hold, the network designer shall ensure that the reception of the $l+1$ th message is after the reception of the l th message plus its processing time i.e.:

$$\forall l, T_r(f_{l+1}) \leq T_r(f_l) + \delta_{process}(f_l)
 \tag{27}$$

► **Constraint Template 14.** [Option 2] Bounded Arrival For Processing (α_d)

Let $f \in \mathbb{F}$ be a flow. Let α_d be the expected number of messages of flow f to be received during duration d . Let us assume that when all the α_d messages are received, their processing starts. We denote $\delta_{process}(f_l)$,



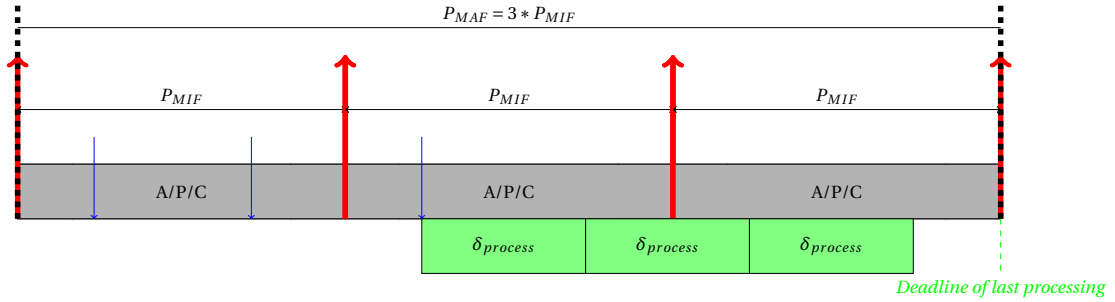
■ **Figure 15** Bounded Arrival For Processing with $\alpha_{P_{MAF}} = 3$ - Option 1

253 the required processing time for the l th message of f . Flow f satisfies the Bounded Arrival For Processing
 254 constraint if and only if the last message of the set of received messages is received early enough so that
 255 all the messages are processed before the end of the cycle i.e.:

$$f \models \text{BoundedArrivalForProcessing}(\alpha_{P_{MAF}}) \quad \forall l \in \mathbb{N}, \quad (28)$$

$$T_r(f_{(l+1)*\alpha_{P_{MAF}}-1}) \leq \text{Recycle}(f_{l*\alpha_{P_{MAF}}}) + P_{MAF} - \sum_{i=0}^{\alpha_{P_{MAF}}-1} \delta_{process}(f_{l+i})$$

257 **Remark:** In order for this constraint to hold, the network designer shall ensure that the necessary time
 258 to receive and process all α_d frames is lower than duration d .



■ **Figure 16** Bounded Arrival For Processing with $d = P_{MAF}$ and $N_{Frames} = 3$ - Option 2

259 1.2.5 Precedence Constraints

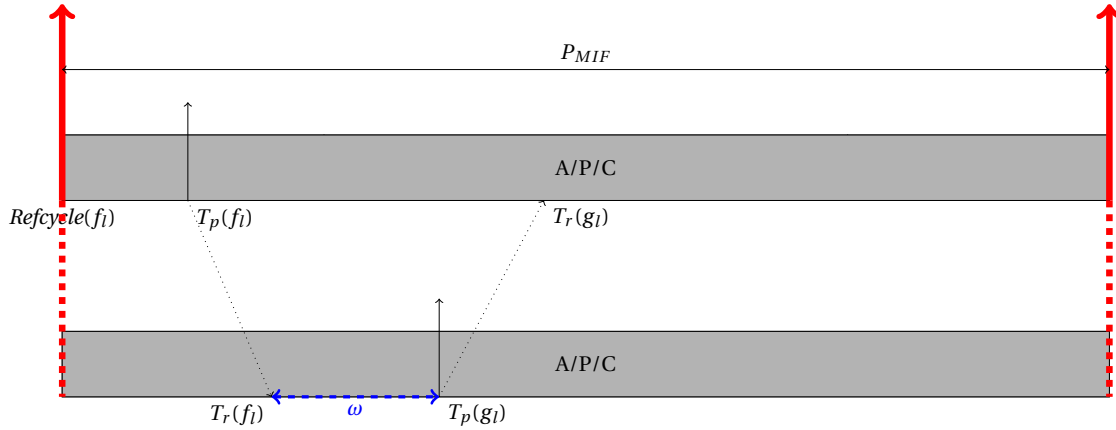
260 This last paragraph will introduce constraints that link flows together. We have identified one constraint
 261 in this category: the *Reaction constraint*.

262
 263 ► **Constraint Template 15.** $\text{Reaction}(\omega, f)$

264 Let $f, g \in \mathbb{F}$ be two flows, let f_l (resp. g_l) the l th message of f (resp. g). Let ω a positive constant.

$$f \models \text{Reaction}(\omega, f) \quad \stackrel{\text{def}}{\equiv} \quad \forall l \in \mathbb{N}, (0 \leq) \omega \leq (T_p(g_l) - T_r(f_l)) \quad (29)$$

266 This constraint specifies that the l th message of g shall be emitted after the emission of the l th message
 267 of f plus a duration ω . It is illustrated in Fig. 17.



■ **Figure 17** Reaction constraints with two flows f and g

1.2.6 Motivating Example: Adding Flow Constraints

Three flows have been identified in the motivating example (cf. 1.2.1). Let us now assign them flow constraints according to the previously defined constraints. We denote $Y(f)$ the function that, given a flow, returns its flow constraints. In the motivating example:

$$Y(f_OBC_HPActuator_1) = \begin{cases} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(\frac{3}{4} * P_{MIF}, P_{MIF}) \\ \text{TimeZone}(\frac{3}{4} * P_{MIF}, P_{MIF}) \end{cases} \quad (30)$$

$$Y(f_OBC_HPSensor_1) = \begin{cases} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, \frac{1}{4} * P_{MIF}) \\ \text{TimeZone}(0, \frac{1}{4} * P_{MIF}) \\ \text{Jitter}(500\mu s) \end{cases} \quad (31)$$

$$Y(f_HPSensor_OBC_1) = \begin{cases} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, \frac{1}{4} * P_{MIF}) \\ \text{TimeZone}(0, \frac{1}{4} * P_{MIF}) \\ \text{Reaction}(\omega, f_OBC_HPSensor_1) \end{cases} \quad (32)$$

1.3 Relaxing Hypothesis 2

In a previous section of this document, we have chosen through *Hypothesis 2* that the maximum size of any message of any flow could not be greater than the Ethernet MTU. In fact, this allows us to state that one applicative message correspond to one Ethernet Frame. Relaxing Hyp. 2 will lead to having one applicative message being split (or fragmented) into several Ethernet frames. In that case, the concepts of *Emission Date* and *Reception Date* of a message shall be redefined.

283 ► **Definition 4.** Emission Date

284 Let f be a flow ($\in \mathbb{F}$), where f does not satisfies Hyp. 2. Let f_l be the l th message of f . Let us denote
 285 $Frames(f_l)$ the set of Ethernet frames corresponding to f_l . Let $First(Frames(f_l))$ (resp. $Last(Frames(f_l))$)
 286 the first (resp. last) frame corresponding to f_l . The emission date of f_l , i.e. $T_p(f_l)$, correspond to the
 287 emission date (at applicative level) of the first bit of $First(Frames(f_l))$.

288 ► **Definition 5.** Reception Date

289 Let f be a flow ($\in \mathbb{F}$), where f does not satisfies Hyp.2. Let f_l be the l th message of f . Let us denote
 290 $Frames(f_l)$ the set of Ethernet frames corresponding to f_l . Let $First(Frames(f_l))$ (resp. $Last(Frames(f_l))$)
 291 the first (resp. last) frame corresponding to f_l . The reception date of f_l , i.e. $T_r(f_l)$, correspond to the
 292 reception date (at applicative level) of the last bit of $Last(Frames(f_l))$.

293 With these newly adapted definitions, Hyp. 2 can be relaxed. The constraints previously introduced
 294 in this technical note are based on the concept of production date and reception date. Since they have
 295 just been redefined, the size of applicative message is no longer limited to the Ethernet MTU.

296 1.4 TSN Network

297 In our system, up to now, we have identified the devices and the flows. These flows travel through their
 298 messages between devices thanks to a TSN network. Let us define what this network is.

299 1.4.1 Definitions

300 A Time Sensitive Network is composed of End-Systems, switches and links. The End-systems are the
 301 devices defined in a previous section. The switches connect a set of links to another set of links. In a
 302 network topology, the end-systems can be connected with each other with a given number of links and
 303 switches.

304

305 ► **Hypothesis 4.** In our system, we chose a topology with 2 switches, as represented in Fig.18. In this
 306 document, we denote this network by Net .

307 The elements that are left to be defined are the network links.

308

309 ► **Definition 6.** Links

310 A link is a unidirectional path between a source node (i.e. end-system or switch) and a destination
 311 network node. We denote \mathcal{L} the set of links in Net (the network we consider, defined in Hyp. 4). A link is
 312 characterized by the following tuple:

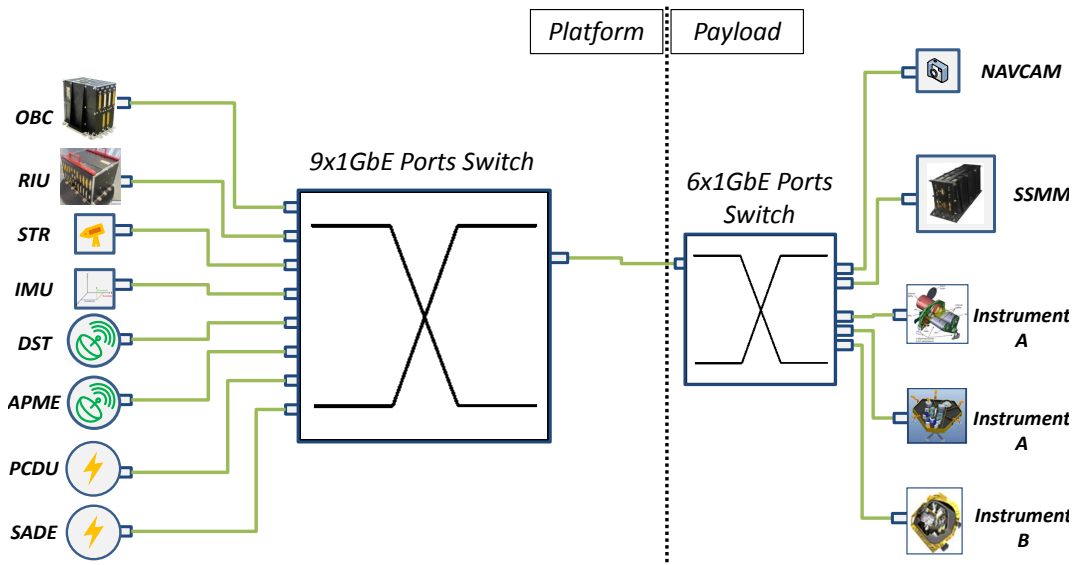
$$313 \quad \forall v \in \mathcal{L}, \langle v.Start, v.End, v.Speed, v.PropagationDelay, v.Macrotick \rangle \quad (33)$$

314 Where :

- 315 ■ $v.Start$ is a node (switch or end-system) at the start of the link,
- 316 ■ $v.End$ is a node (switch or end-system) at the end of the link,
- 317 ■ $v.Speed$ is the maximum speed on the link. We express it in Megabits per seconds,
- 318 ■ $v.PropagationDelay$ is a medium specific delay. It varies between physical mediums and links
 319 length. We express it in nanoseconds.
- 320 ■ $v.Macrotick$ is the length of a discrete time unit defining the granularity of time events for the
 321 given link. We express it in nanoseconds.

322 ► **Hypothesis 5.** In our system, we only consider 1Gbit/s links i.e.:

$$323 \quad \forall v \in \mathcal{L}, v.Speed = 1000 \quad (34)$$



■ **Figure 18** Network topology considered in our hypothesis

324 ► **Hypothesis 6.** In our system, we consider that all links have the same *Macrotick* i.e.:

$$325 \quad \forall va, vb \in \mathcal{L}, va.Macrotick = vb.Macrotick \quad (35)$$

326 1.4.2 Fault Tolerance Requirements

327 In this report, we have only formalized the network performance requirements. However, these are not
 328 the only requirements of the satellites on-board networks. Another category of requirements exists : the
 329 fault tolerance requirements. We will not detail these requirements in this report.

330 2 TSN Configuration Concept

331 In the prospect of using Time Sensitive Networking as the on-board network solution able to satisfy
 332 the previously introduced requirements, this section introduces the concept of *Configuration* of a TSN
 333 Network as well as some TSN vocabulary.

334 2.1 Vocabulary

335 In this section, we introduce some TSN vocabulary needed to understand the TSN configuration
 336 concept. These definitions are not introduced in IEEE TSN standards but reflect our understanding of
 337 Time Sensitive Networking.

339 ► **Definition 1** (Feature). A *feature* refers to a TSN standard, amendment or on-going project. For
 340 instance, 802.1Qbv is a feature.

341 ► **Definition 2** (Mechanism). A *mechanism* is a part of a feature. Thus a feature is a union of mechan-
 342 isms. For instance, Time Aware Shaper is a mechanism of 802.1Qbv feature.

343 ► **Definition 3** (Parameter). A *parameter* is a variable in a specific mechanism of a specific feature
 344 that can have several values. Thus a mechanism is a union of parameters. A mechanism has at least one
 345 parameter. For instance, the value and length of a Gate Control List (GCL) is a parameter of Time Aware
 346 Shaper mechanism.

347 2.2 Configuration

348 In this section, we introduce the concept of *Configuration* for a TSN Network.

349 ► **Definition 7.** Configuration of a Network

350 Let Net be a network, let \mathcal{N} be the set of nodes, of size n , in Net . A configuration of a network corresponds
 351 to a configuration for each node in the network. Let us denote $\Theta(Net)$ (resp. $\Theta(n)$) a configuration of the
 352 network Net (resp. of node d). Hence:

$$353 \quad \forall n_1, \dots, n_n \in \mathcal{N}, \Theta(Net) = \langle \Theta(n_1), \dots, \Theta(n_n) \rangle \quad (36)$$

354 ► **Definition 8.** Configuration of a device

355 Let $d \in \mathcal{D}$ a node. Let $Features$ be the set of all Time Sensitive Networking features. Let $Feat_1, \dots, Feat_n$
 356 be a subset of size n ($\in \mathbb{N}$) of $Features$. We denote $\Theta(Feat)$ a configuration of TSN feature $Feat$. A
 357 configuration of device d is characterized by the following tuple:

$$358 \quad \forall Feat_1, \dots, Feat_n \in Features, \Theta(d) = \langle \Theta(Feat_1), \dots, \Theta(Feat_n) \rangle \quad (37)$$

359 According to this definition, if a feature does not appear in the configuration of a device, this means that
 360 this feature will not be used in this device. Accordingly, if a feature appears in the configuration of the
 361 device, this means that it will have to be configured (see the following definition).

362 ► **Definition 9.** Configuration of a feature

363 Let $Feat \in Features$ be a feature in the set of all TSN features. Let $Mechanisms(Feat)$ be the set of all
 364 mechanisms of $Feat$. Let $Mechanism_1, \dots, Mechanism_n$ be a subset of size n of $Mechanisms(Feat)$. We
 365 denote $\Theta(Mechanism)$ a configuration of a mechanism of $Feat$. A configuration of TSN feature $Feat$ is
 366 characterized by the following tuple:

$$367 \quad \forall Mechanism_1, \dots, Mechanism_n \in Mechanisms(Feat), \quad (38)$$

$$\Theta(Feat) = \langle \Theta(Mechanism_1), \dots, \Theta(Mechanism_n) \rangle$$

368 According to this definition, if a mechanism does not appear in the configuration of a feature, this
 369 means that this mechanism will not be used in this feature. Accordingly, if a mechanism appears in the
 370 configuration of the feature, this means that it will have to be configured (see the following definition).

371 ► **Definition 10.** Configuration of a mechanism

372 Let $Feat$ be a TSN feature, let $Mechanism$ be a mechanism of $Feat$. Let $Parameters(Mechanism)$
 373 be the set of parameters of mechanism. Let $param_1, \dots, param_n$ be a subset of size n of $Paramet-$
 374 $ers(Mechanism)$. We denote $\Theta(param)$ the value of parameter $param$. A configuration of TSN mechan-
 375 $ism Mechanism$ of feature $Feat$ is characterized by the following tuple:

$$376 \quad \forall param_1, \dots, param_n \in Parameters, \Theta(Mechanism) = \langle \Theta(param_1), \dots, \Theta(param_n) \rangle \quad (39)$$

377 According to this definition, if a parameter does not appear in the configuration of a mechanism, this
 378 means that this parameter will not be used in this mechanism. Accordingly, if a parameter appears in
 379 the configuration of the mechanism, this means that it will have to be given a value.

380 ► **Definition 11.** Flow to Traffic Class mapping

381 Let f be a flow in the set of flows \mathbb{F} of Net , let \mathcal{N} be the set of nodes of Net . Let n_1, \dots, n_n be a subset of

382 \mathcal{N} corresponding to the nodes in the path of f , let us denote it $Path(f)$. We denote $TC(f, d)$ the traffic
 383 class of flow f in node N . The Flow to Traffic Class mapping of Net is characterized by the following
 384 tuple:

$$385 \quad \forall f \in \mathbb{F}, \forall d \in Path(f), Mapping(Net) = \langle TC(f, d) \rangle \quad (40)$$

386 With this definition, we introduce the possibility for a flow to change of traffic classes between nodes. In
 387 all cases (802.Q-2018 property, see chapter 3):

$$388 \quad \forall f \in \mathbb{F}, \forall d \in Path(f), 1 \geq TC(f, d) \leq 8 \quad (41)$$

389 ► **Definition 12.** Configuration of a system

390 Let S be the system we consider. We denote $\Theta(S)$ a configuration of S . Let Net be the network of your
 391 system. A configuration of S is characterized by the following tuple:

$$392 \quad \Theta(S) = \langle \Theta(Net), Mapping(Net) \rangle \quad (42)$$

393 This means that a configuration of the system is composed of a network configuration and a flows to
 394 traffic class mapping.

395 **3 Satellite On-board Network Use Case**

396 In this section, we instantiate the model and constraints introduced above to a Satellite On-Board
 397 Network use case.

398 **3.1 Network topology**

399 The system we consider is a network composed of:

- 400 ■ One computing device, named *On-Board Computer* (OBC)
- 401 ■ Two sensing devices, named *Star Tracker* (STR) and *Navigation Camera* (NAVCAM)
- 402 ■ One actuating device connected in the network through a *Remote Interface Unit* (RIU)
- 403 ■ Three payload devices, named *Mass Memory* (SSMM), *Instrument 1* (Instr. 1) and *Instrument 2* (Instr.
 404 2).

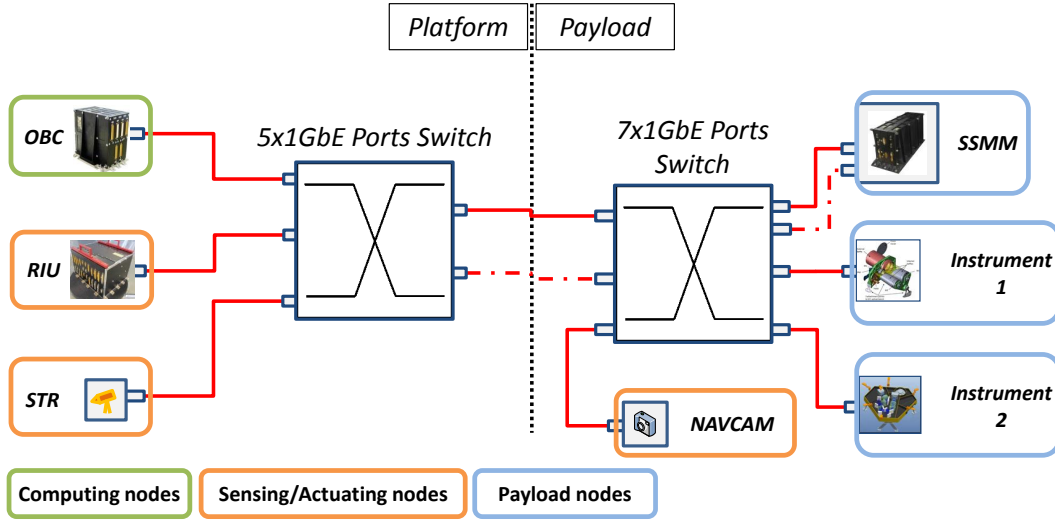
405 These devices are connected together through a set of links and two switches, according to the topology
 406 of Fig. 19.

407 **3.2 Applications**

408 In this satellite use case, we consider three applications:

- 409 ■ One *Command & Control* application
- 410 ■ One *Vision Based Navigation* (VBN) application
- 411 ■ One *Payload* application

412 This section will describe the parameters of these applications. Note that we now denote the *input* part
 413 as the *Acquisition* part and the *output* part as the *Command* part.



■ **Figure 19** Network topology and devices considered in the Satellite use case

3.2.1 Command & Control application use case

414

415 This C & C application is running on the OBC (Platform Computing Device). Flows runs between the
416 OBC, the STR and the RIU. For this application, we choose:

$$417 \quad k = 8 \text{ hence } P_{MIF} = 125ms \quad (43)$$

418 Within one cycle (of length P_{MIF}), we specify the duration of the Acquisition, Processing and Command
419 part. We choose:

$$420 \quad \delta t_1 = \frac{1}{4}, \delta t_2 = \frac{1}{2}, \delta t_3 = \frac{1}{4} \quad (44)$$

3.2.2 VBN application use case

421

422 This VBN application is running on the OBC (Platform Computing Device). Flows runs between the
423 OBC and the NAVCAM. For this application, we choose:

$$424 \quad k = 30 \text{ hence } P_{MIF} = 33,33ms \quad (45)$$

425 Within one cycle (of length P_{MIF}), we specify the duration of the Acquisition, Processing and Command
426 part. We choose:

$$427 \quad \delta t_1 = \frac{1}{3}, \delta t_2 = \frac{1}{3}, \delta t_3 = \frac{1}{3} \quad (46)$$

3.2.3 Payload application use case

428

429 This C & C application is running on the OBC (Platform Computing Device) and on the Payload Devices.
430 Flows runs between the OBC, the INSTR1, the INSTR2 and the SSMM. For this application, we choose:

$$431 \quad k = 1000 \text{ hence } P_{MIF} = 1ms \quad (47)$$

432 Within one cycle (of length P_{MIF}), we specify the duration of the Acquisition, Processing and Command
433 part. We choose:

$$434 \quad \delta t_1 = \frac{P_{MIF}}{2}, \delta t_2 = \frac{P_{MIF}}{2}, \delta t_3 = 0 \quad (48)$$

435 **3.3 Flows**

436 In this satellite use case, we consider that the applications running on the different devices send their
437 messages through several flows. Let us now define these flows and their constraints with the formalism
438 of Section 1. For easier readability, the flow's definition are gathered in Tables 2 to 8.

439 3.3.1 C & C application use case : adding flows

Name	Source	Dest	Max. Data Size	r_f
f_RIU_OBC_HK_1	RIU	OBC	64	-8
f_RIU_OBC_HK_2	RIU	OBC	64	-8
f_RIU_OBC_HK_3	RIU	OBC	64	-8
f_RIU_OBC_HK_4	RIU	OBC	64	-8
f_RIU_OBC_HK_5	RIU	OBC	64	-8
f_RIU_OBC_HK_6	RIU	OBC	64	-8
f_RIU_OBC_HK_7	RIU	OBC	64	-8
f_RIU_OBC_HK_8	RIU	OBC	64	-8
f_RIU_OBC_HK_9	RIU	OBC	64	-8
f_RIU_OBC_HK_10	RIU	OBC	64	-8
f_RIU_OBC_HK_11	RIU	OBC	64	-1
f_RIU_OBC_HK_12	RIU	OBC	64	-1
f_RIU_OBC_FDIR_1	RIU	OBC	64	-8
f_RIU_OBC_FDIR_2	RIU	OBC	64	-8
f_RIU_OBC_FDIR_3	RIU	OBC	64	-8
f_RIU_OBC_FDIR_4	RIU	OBC	64	-8
f_RIU_OBC_FDIR_5	RIU	OBC	64	-8
f_RIU_OBC_FDIR_6	RIU	OBC	64	-8
f_RIU_OBC_FDIR_7	RIU	OBC	64	-1
f_RIU_OBC_FDIR_8	RIU	OBC	64	-1
f_RIU_OBC_FDIR_9	RIU	OBC	64	-8
f_RIU_OBC_FDIR_10	RIU	OBC	64	-8
f_RIU_OBC_FDIR_11	RIU	OBC	64	-8
f_RIU_OBC_FDIR_12	RIU	OBC	64	-8
f_RIU_OBC_MEO_1	RIU	OBC	64	-1
f_RIU_OBC_MEO_2	RIU	OBC	64	-1
f_RIU_OBC_MEO_3	RIU	OBC	64	-1
f_RIU_OBC_MEO_4	RIU	OBC	64	-1
f_RIU_OBC_MEO_5	RIU	OBC	64	-1
f_RIU_OBC_MEO_6	RIU	OBC	64	-1
f_RIU_OBC_MEO_7	RIU	OBC	64	-1
f_RIU_OBC_MEO_8	RIU	OBC	64	-1
f_RIU_OBC_ACK_1	RIU	OBC	64	-1
f_RIU_OBC_ACK_2	RIU	OBC	64	-1
f_RIU_OBC_ACK_3	RIU	OBC	64	-1
f_RIU_OBC_ACK_4	RIU	OBC	64	-1
f_RIU_OBC_ACK_5	RIU	OBC	64	-1
f_RIU_OBC_ACK_6	RIU	OBC	64	-1
f_RIU_OBC_ACK_7	RIU	OBC	64	-1
f_RIU_OBC_ACK_8	RIU	OBC	64	-1
f_RIU_OBC_ACK_9	RIU	OBC	64	-1
f_RIU_OBC_ACK_10	RIU	OBC	64	-1
f_RIU_OBC_DATA_1	RIU	OBC	64	-1
f_RIU_OBC_DATA_2	RIU	OBC	64	-1

440

441 ■ **Table 2** Flow definition for the satellite C&C use case, device RIU

Name	Source	Dest	Max. Data Size	r_f
f_OBC_RIU_HK_1	OBC	RIU	64	-8
f_OBC_RIU_HK_2	OBC	RIU	64	-8
f_OBC_RIU_HK_3	OBC	RIU	64	-8
f_OBC_RIU_HK_4	OBC	RIU	64	-8
f_OBC_RIU_HK_5	OBC	RIU	64	-8
f_OBC_RIU_HK_6	OBC	RIU	64	-8
f_OBC_RIU_HK_7	OBC	RIU	64	-8
f_OBC_RIU_HK_8	OBC	RIU	64	-8
f_OBC_RIU_HK_9	OBC	RIU	64	-8
f_OBC_RIU_HK_10	OBC	RIU	64	-8
f_OBC_RIU_HK_11	OBC	RIU	64	-1
f_OBC_RIU_HK_12	OBC	RIU	64	-1
f_OBC_RIU_FDIR_1	OBC	RIU	64	-8
f_OBC_RIU_FDIR_2	OBC	RIU	64	-8
f_OBC_RIU_FDIR_3	OBC	RIU	64	-8
f_OBC_RIU_FDIR_4	OBC	RIU	64	-8
f_OBC_RIU_FDIR_5	OBC	RIU	64	-8
f_OBC_RIU_FDIR_6	OBC	RIU	64	-8
f_OBC_RIU_FDIR_7	OBC	RIU	64	-1
f_OBC_RIU_FDIR_8	OBC	RIU	64	-1
f_OBC_RIU_FDIR_9	OBC	RIU	64	-8
f_OBC_RIU_FDIR_10	OBC	RIU	64	-8
f_OBC_RIU_FDIR_11	OBC	RIU	64	-8
f_OBC_RIU_FDIR_12	OBC	RIU	64	-8
f_OBC_RIU_MEO_1	OBC	RIU	64	-1
f_OBC_RIU_MEO_2	OBC	RIU	64	-1
f_OBC_RIU_MEO_3	OBC	RIU	64	-1
f_OBC_RIU_MEO_4	OBC	RIU	64	-1
f_OBC_RIU_MEO_5	OBC	RIU	64	-1
f_OBC_RIU_MEO_6	OBC	RIU	64	-1
f_OBC_RIU_MEO_7	OBC	RIU	64	-1
f_OBC_RIU_MEO_8	OBC	RIU	64	-1
f_OBC_RIU_CMD_1	OBC	RIU	64	-1
f_OBC_RIU_CMD_2	OBC	RIU	64	-1
f_OBC_RIU_CMD_3	OBC	RIU	64	-1
f_OBC_RIU_CMD_4	OBC	RIU	64	-1
f_OBC_RIU_CMD_5	OBC	RIU	64	-1
f_OBC_RIU_CMD_6	OBC	RIU	64	-1
f_OBC_RIU_CMD_7	OBC	RIU	64	-1
f_OBC_RIU_CMD_8	OBC	RIU	64	-1
f_OBC_RIU_CMD_9	OBC	RIU	64	-1
f_OBC_RIU_CMD_10	OBC	RIU	64	-1
f_OBC_RIU_ACQUI_1	OBC	RIU	64	-1
f_OBC_RIU_ACQUI_2	OBC	RIU	64	-1

442

443 ■ **Table 3** Flow definition for the satellite C&C use case, device OBC-Part 1

444

Name	Source	Dest	Max. Data Size	r_f
f_STR_OBC_HK_1	STR	OBC	64	-1
f_STR_OBC_HK_2	STR	OBC	64	-1
f_STR_OBC_FDIR	STR	OBC	64	-1
f_STR_OBC_MEO_1	STR	OBC	64	-1
f_STR_OBC_MEO_2	STR	OBC	64	-1
f_STR_OBC_ACK_1	STR	OBC	64	-1
f_STR_OBC_ACK_2	STR	OBC	64	-1
f_STR_OBC_ACK_3	STR	OBC	64	-1
f_STR_OBC_ACK_4	STR	OBC	64	-1
f_STR_OBC_ACK_5	STR	OBC	64	-1
f_STR_OBC_DATA	STR	OBC	1088	-1

445 ■ **Table 4** Flow definition for the satellite C&C use case, device STR

446

Name	Source	Dest	Max. Data Size	r_f
f_OBC_STR_HK_1	OBC	STR	64	-1
f_OBC_STR_HK_2	OBC	STR	64	-1
f_OBC_STR_FDIR	OBC	STR	64	-1
f_OBC_STR_MEO_1	OBC	STR	64	-1
f_OBC_STR_MEO_2	OBC	STR	64	-1
f_OBC_STR_CMD_1	OBC	STR	64	-1
f_OBC_STR_CMD_2	OBC	STR	64	-1
f_OBC_STR_CMD_3	OBC	STR	64	-1
f_OBC_STR_CMD_4	OBC	STR	64	-1
f_OBC_STR_CMD_5	OBC	STR	64	-1
f_OBC_STR_ACQUI	OBC	STR	64	-1
f_OBC_INSTR1_HK	OBC	INSTR1	128	-4
f_OBC_INSTR2_HK	OBC	INSTR1	128	-4
f_OBC_NAVCAM_HK	OBC	NAVCAM	128	-4

447 ■ **Table 5** Flow definition for the satellite C&C use case, device OBC-Part 2

448

Name	Source	Dest	Max. Data Size	r_f
f_INSTR1_OBC_HK	OBC	INSTR1	128	-4
f_INSTR2_OBC_HK	OBC	INSTR1	128	-4
f_NAVCAM_OBC_HK	OBC	NAVCAM	128	-4

449 ■ **Table 6** Flow definition for the satellite C&C use case, device INSTR1-2 and NAVCAM

450 **3.3.2 VBN application use case : adding flows**

451

Name	Source	Dest	Max. Data Size	r_f
f_NAVCAM_OBC_SCI	NAVCAM	OBC	1048576	-1

452 ■ **Table 7** Flow definition for the satellite VBN use case, device INSTR1-2

3.3.3 Payload application use case : adding flows

Name	Source	Dest	Max. Data Size	r_f
f_INSTR1_SSMM_SCI	INSTR1	SSMM	4096	-3
f_INSTR2_SSMM_SCI	INSTR2	SSMM	4096	2

Table 8 Flow definition for the satellite payload use case, device INSTR1-2

3.4 Adding Flow Constraints

Now that the flows of our system have been identified, let us assign them some flow constraints. Again, for easier readability, the flow constraints are gathered in Tables 9 to 17.

Name (Suffix)	$\Upsilon(f)$
f_* (all flows)	ReceptionAfterEmission()
f_* (all flows)	Ordered Emission()
f_* (all flows)	Ordered Delivery()
f_* (all flows)	Ethernet Frame Size()

Table 9 Flow constraints definition for the satellite use case

Please note that the parameters of the Reaction and MinimumSpace Constraints have not be fully computed at this stage.

3.4.1 C & C application use case : adding flows constraints

Name (Suffix)	$\Upsilon(f)$
f_STR_OBC_HK_*	$\left\{ \begin{array}{l} \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \\ \text{Reaction}(\omega_1, f_OBC_STR_HK_*) \end{array} \right.$
f_STR_OBC_FDIR_*	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \\ \text{Reaction}(\omega_2, f_OBC_STR_FDIR_*) \end{array} \right.$
f_STR_OBC_MEO_*	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, P_{MIF}) \\ \text{TimeZone}(0, P_{MIF}) \\ \text{Reaction}(\omega_3, f_OBC_STR_MEO_*) \end{array} \right.$
f_STR_OBC_ACK_*	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, P_{MIF}) \\ \text{TimeZone}(0, P_{MIF}) \\ \text{Reaction}(\omega_4, f_OBC_STR_CMD_*) \end{array} \right.$
f_STR_OBC_DATA_*	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, P_{MIF}) \\ \text{TimeZone}(0, P_{MIF}) \\ \text{Reaction}(\omega_5, f_OBC_STR_ACQUI_*) \end{array} \right.$

26 Formal specification of satellite on-board networks requirements

465 ■ **Table 10** Flow constraints definition for the satellite C&C use case, device STR

f_RIU_OBC_HK_*	$\left\{ \begin{array}{l} \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \\ \text{Reaction}(\omega_6, f_OBC_RIU_HK_*) \end{array} \right.$
f_RIU_OBC_FDIR_*	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \\ \text{Reaction}(\omega_7, f_OBC_RIU_FDIR_*) \end{array} \right.$
f_RIU_OBC_MEO_*	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, P_{MIF}) \\ \text{TimeZone}(0, P_{MIF}) \\ \text{Reaction}(\omega_8, f_OBC_RIU_MEO_*) \end{array} \right.$
f_RIU_OBC_ACK_*	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, P_{MIF}) \\ \text{TimeZone}(0, P_{MIF}) \\ \text{Reaction}(\omega_9, f_OBC_RIU_CMD_*) \end{array} \right.$
f_RIU_OBC_DATA_*	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, P_{MIF}) \\ \text{TimeZone}(0, P_{MIF}) \\ \text{Reaction}(\omega_{10}, f_OBC_RIU_ACQUI_*) \end{array} \right.$

467 ■ **Table 11** Flow constraints definition for the satellite C&C use case, device RIU

f_OBC_STR_HK_*	$\left\{ \begin{array}{l} \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \end{array} \right.$
f_OBC_STR_FIDR_*	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \end{array} \right.$
f_OBC_STR_MEO_*	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \\ \text{MinimumSpace}(\gamma_1) \end{array} \right.$
f_OBC_STR_CMD_*	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}((\Delta(\text{Input}) + \Delta(\text{Processing})) * P_{MIF}, P_{MIF}) \\ \text{TimeZone}((\Delta(\text{Input}) + \Delta(\text{Processing})) * P_{MIF}, P_{MIF}) \\ \text{Jitter}(1\mu\text{s}) \end{array} \right.$
f_OBC_STR_ACQUI_*	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, \Delta(\text{Input}) * P_{MIF}) \\ \text{TimeZone}(0, \Delta(\text{Input}) * P_{MIF}) \\ \text{Jitter}(500\mu\text{s}) \end{array} \right.$
f_OBC_RIU_HK_*	$\left\{ \begin{array}{l} \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \end{array} \right.$
f_OBC_RIU_FDIR_*	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \end{array} \right.$

468

469 ■ **Table 12** Flow constraints definition for the satellite C&C use case, device OBC part. 1

f_OBC_RIU_MEO	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \\ \text{MinimumSpace}(\gamma_2) \end{array} \right.$
f_OBC_RIU_CMD	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}((\Delta(\text{Input}) + \Delta(\text{Processing})) * P_{MIF}, P_{MIF}) \\ \text{TimeZone}((\Delta(\text{Input}) + \Delta(\text{Processing})) * P_{MIF}, P_{MIF}) \\ \text{Jitter}(1\mu\text{s}) \end{array} \right.$
f_OBC_RIU_ACQUI	$\left\{ \begin{array}{l} \text{PeriodicMessageProduction}() \\ \text{ImplicitDeadlines}() \\ \text{InjectionZone}(0, \Delta(\text{Input}) * P_{MIF}) \\ \text{TimeZone}(0, \Delta(\text{Input}) * P_{MIF}) \\ \text{Jitter}(500\mu\text{s}) \end{array} \right.$
f_OBC_INSTR1_HK	$\left\{ \begin{array}{l} \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \end{array} \right.$
f_OBC_INSTR2_HK	$\left\{ \begin{array}{l} \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \end{array} \right.$
f_OBC_NAVCAM_HK	$\left\{ \begin{array}{l} \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \end{array} \right.$

471 ■ **Table 13** Flow constraints definition for the satellite C&C use case, device OBC part. 2

f_INSTR1_OBC_HK	$\left\{ \begin{array}{l} \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \\ \text{Reaction}(\omega_{11}, f_OBC_INSTR1_HK_*) \end{array} \right.$
f_INSTR2_OBC_HK	$\left\{ \begin{array}{l} \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \\ \text{Reaction}(\omega_{12}, f_OBC_INSTR2_HK_*) \end{array} \right.$

472
473 ■ **Table 14** Flow constraints definition for the satellite C&C use case, device INSTR1, INSTR2

f_NAVCAM_OBC_HK	$\left\{ \begin{array}{l} \text{InjectionZone}(0, r_f * P_{MIF}) \\ \text{TimeZone}(0, r_f * P_{MIF}) \\ \text{Reaction}(\omega_{13}, f_OBC_NAVCAM_HK_*) \end{array} \right.$
-----------------	---

474
475 ■ **Table 15** Flow constraints definition for the satellite C&C use case, device NAVCAM

476 3.4.2 VBN application use case : adding flows constraints

f_NAVCAM_OBC_SCI	$\left\{ \begin{array}{l} \text{ControlledInputTrafficShape}(R, B) \\ \text{ControlledOutputTrafficShape}(d, N_{Frames}) \end{array} \right.$
------------------	---

477
478 ■ **Table 16** Flow constraints definition for the satellite VBN use case, device NAVCAM

479 ► Remark. The parameters of the ControlledInputTrafficShape and ControlledOutputTrafficShape
480 constraints for NAVCAM have not been computed yet.

481 3.4.3 Payload application use case : adding flows constraints

482	f_INSTR1_SSMM_SCI	$\left\{ \begin{array}{l} \text{ControlledInputTrafficShape}(R, B) \\ \text{ControlledOutputTrafficShape}(d, N_{Frames}) \end{array} \right.$
	f_INSTR2_SSMM_SCI	$\left\{ \begin{array}{l} \text{ControlledInputTrafficShape}(R, B) \\ \text{ControlledOutputTrafficShape}(d, N_{Frames}) \end{array} \right.$

483 ■ **Table 17** Flow constraints definition for the satellite Payload use case, device INSTR1-2

484 ► Remark. The parameters of the ControlledInputTrafficShape and ControlledOutputTrafficShape
 485 constraints for INSTR1 and INSTR2 have not been computed yet.