Aalborg Universitet

AALBORG UNIVERSITY

Worst-Case Traversal Time Modelling of Ethernet based In-Car Networks using Real Time Calculus

Revsbech, Kasper; Schiøler, Henrik; Madsen, Tatiana Kozlova; Nielsen, Jimmy Jessen

Worst-Case Traversal Time Modelling of Ethernet based In-Car Networks using Real Time Calculus

Kasper Revsbech[1], Henrik Schiøler[1], Tatiana K. Madsen[1], Jimmy J. Nielsen[1]

Dept. of Electronic Systems, Aalborg University, Denmark, {kar | henrik | tatiana | jjn}@es.aau.dk

Abstract. The paper addresses performance modelling of safety critical Ethernet networks with special attention to in-car networks. A specific Ethernet/IP based in-car network is considered as use-case. The modelling is based on the analytical method Real-Time Calculus (RTC), providing hard bounds on delay and buffer utilization. We show how RTC can be applied on the use-case network. Furthermore, we develop a network simulation, used to evaluate the overestimation. It is found that the delays from RTC is significantly overestimated. The bounds from RTC, however, are guaranteed bounds, which is not the case for the simulated.

Real-Time Calculus, Vehicular Networks, Performance Modelling, Ethernet Modelling, Delay Bounds.

1   Introduction

Recently the car industry has started investigating the feasibility of merging all, or subsets of, in-car networks to Ethernet/IP. As an example [1] proposes Ethernet/IP as a bridging network, facilitating the variety of bus technologies and protocols currently used. Ethernet is currently widely used in a broad range of applications, however in safety-critical networks its applicability is still under question. As in-car networks are used for communication of critical date, e.g. wheel sensor information, there is a need of verifying its performance and reliability, as accidents and ultimately loss of human life can occur upon malfunction.

In this work we propose the analytical method of Real-Time Calculus (RTC) [2], to model Ethernet/IP based in-car networks. As RTC is an analytical method, it is appropriate, also in the early phases of the network design. Furthermore, RTC models will provide hard bounds on delay and buffer utilization, which are suitable dealing with safety critical systems, where deadline violations can have a crucial impact.

In this work we show how to model an Ethernet/IP based in-car network, to illustrate the applicability of RTC to model in-car networks. Furthermore, we contribute models that are generally applicable when modelling Ethernet networks, and in particular safety critical networks based on Ethernet.

The theory of RTC has been designed specifically for modelling/analyzing real-time systems. RTC is a derivative/flavor of Network Calculus (NC), which
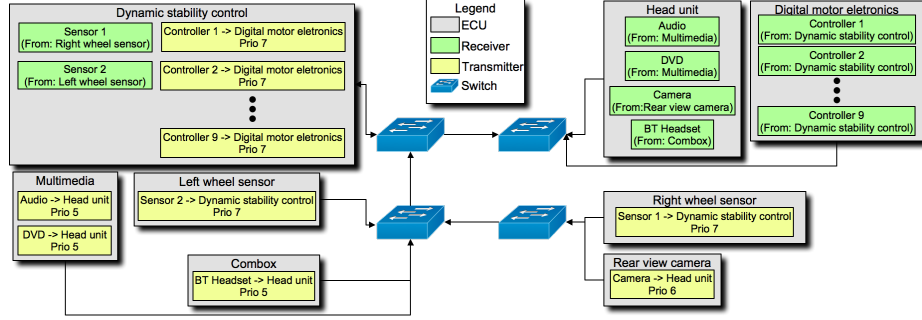
Fig. 1. The topology of the use-case network

was first proposed by Cruz in [3] and [4], and later formalized by Boudec & Thiran in [5], and by Chang in [6]. Fidler recently provided an extensive survey [7] as well. In [8] Wandeler Et al. shows RTC applied in a case study, studying a distributed in-car radio navigation system.

The remainder of this paper is structured as follows: First we present an in-car network example, which serves as use-case network. In Sec 3 we present fundamentals of RTC, which is followed by a description of how RTC is applied on the use-case network. In Sec 5 we present a simulation model of the use-case network, in order to compare the RTC based model with a simulation model. The results of the RTC model, the simulation, and the comparison are given in Sec 6, followed by the conclusion and outlooks.

2 An Ethernet based In-Car Network

To investigate the feasibility of applying the theory of RTC to Ethernet based in-car networks, an use-case network has been constructed. The constructed use-case network is a realistic sub-set of an in-car network, facilitating a sub-set of flows, present on a full scale in-car network. The link layer functionality is based on Ethernet (IEEE 802.3), where prioritization is achieved by use of VLAN headers (IEEE 802.1Q), and the prioritization specified in IEEE 802.1P. The Network layer is based on IPv4 and the Transport layer on UDP. The topology of the use-case network is depicted in Fig. 1. As seen from the figure, the topology is made such that two switches are placed in the back of the car, and two in the front. As seen, the communication lines are directed hence the model only consider one way traffic. The traffic flows in the use-case network are listed in Table 1. The two flows from the *Right wheel sensor*, and the *Left wheel sensor* to the *Dynamic stability control*, and the flows from the *Dynamic stability control* to the *Digital motor electronics*, are hard real-time sensor flows, and are assigned the highest priority (7) as they are safety-critical. The flow from the *Rear view camera* to the *Head unit* is a soft real-time video flow, and is assigned priority 6. The flow from the *Combox* to the *Head unit* is a Bluetooth audio flow. There are

Flow	Min CT	PS	P	Flow	Min CT	PS	P
Right wheel sensor → Dynamic stability control							
Sensor 1	2.5	15	7				
Left wheel sensor → Dynamic stability control							
Sensor 2	2.5	15	7				
Dynamic stability control → Digital motor eletronics							
Controller 1	10	8	7	**Controller 2**	20	13	7
Controller 3	20	12	7	**Controller 4**	5	12	7
Controller 5	100	3	7	**Controller 6**	1000	3	7
Controller 7	100	7	7	**Controller 8**	200	7	7
Controller 9	10	8	7				
Rear view camera → Head unit							
Camera	1.59	1472	6				
Combox → Head unit							
BT Headset	1.25	10	5				
Multimedia → Head unit							
Audio	8.4	1472	5	**DVD**	2.5	1472	5

Table 1. Traffic flow specification, where **Min CT** is the minimum cycle time is in ms, **PS** the payload size in Bytes and **P** the priority.

two flows from *Multimedia* to the *Head unit*, an audio and a DVD flow. These three flows are assigned the lowest priority (5), as they are not safety critical.

The definition of the use-case network also comprises a number of assumptions and delimitations, which are elaborated in the following:

– **No Address Resolution Protocol (ARP) requests.** ARP requests are not present in the network. It is fair to assume that the relation between the IP and the MAC addresses will be hardcoded, as the network does not have nodes entering and leaving.
– **Full duplex, switched network.** By use of a switched network cable collisions are avoided.
– **0.003 ms processing delay in the switch relay unit.** Device specific delay, in this case selected to 0.003 ms as we find this realistic.
– **Unlimited buffers.** It has been chosen not to model the effect of packet drops due to filled buffers, hence the buffers in the network are assumed to be unlimited [1].
– **No processing stack processing delays, except for a 0.01 ms delay in the IP stack.** Also a device specific delay, chosen to a realistic value.
– **No propagation delay.**
– **100 Mbit/sec Ethernet.**

---

[1] The theory of RTC provide methods to calculate bounds on needed buffer size
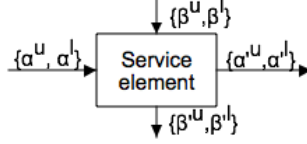
Fig. 2. The input and output bounds of a service element.

3   RTC Theory

RTC is an analytical method allowing the network designer to calculate hard bounds on delay, and queue utilization of a system. In RTC every service element is modelled individually, where a service element can be in an abstract form, e.g. a whole switch, or more specific, e.g. a specific input port. The inputs and outputs of service elements, are then connected to model the whole network. Fig. 2 shows a service element with inputs and outputs. As shown in Fig. 2 the inputs of a service element is an upper and lower arrival bound $\{\alpha^u, \alpha^l\}$, and an upper and lower service bound $\{\beta^u, \beta^l\}$. The upper and lower arrival bounds, are use to model the input traffic to the service element. If we let $R(t)$ be a cumulative function, defined as the number of events from a particular traffic flow in the time interval $[0, t)$. Then we provide an upper and a lower arrival function $\alpha(\Delta) = [\alpha^u(\Delta), \alpha^l(\Delta)]$, for which the upper arrival curve $\alpha^u(\Delta)$ is an upper bound of the number of events that can occur in the traffic flow, in any time interval of length Δ. Similarly the lower arrival function $\alpha^l(\Delta)$ is the lower bound of the events that will occur in a time interval Δ. This leads to the following definition:

Definition 1 (Arrival curves [8]). *Let $R(t)$ denote the number of events that arrive on an event stream in the time interval $[0, t)$. Then, R, α^u and α^l are related to each other by the following inequality:*

$$\alpha^l(t - s) \leq R(t) - R(s) \leq \alpha^u(t - s), \qquad \forall s < t \qquad (1)$$

with $\alpha^u(0) = \alpha^l(0) = 0$

As described, a service element (Fig. 2) also has an upper and a lower service bound as inputs, bounding the service provided by the element. In analogue to the cumulative arrival function $R(t)$, we define a cumulative function $C(t)$ denoting the available processing resources of the element in the time interval $[0, t)$. Similarly to the arrival curves, we define an upper and lower service function $\beta(\Delta) = [\beta^u(\Delta), \beta^l(\Delta)]$. The upper service function $\beta^u(\Delta)$ denotes the maximum service available by the element in any time interval of length Δ. The lower service function $\beta^l(\Delta)$ denotes the guaranteed minimum service, provided by the service element in any time interval of length Δ. The formal definition of the upper and lower service curves is as follows:

Definition 2 (Service curves [8]). *Let $C(t)$ denote the number of processing or communication cycles available, from a resource over the time interval $[0, t)$, then C, β^u and β^l are related by the following inequality:*

$$\beta^l(t-s) \leq C(t) - C(s) \leq \beta^u(t-s), \qquad \forall s < t \qquad (2)$$

with $\beta^l(0) = \beta^u(0) = 0$.

In this work we have chosen to confine ourselves to the following functions, to model arrival and service curves: The *affine* function, the *peak rate* function and the *rate latency* function. The affine function is defined in Eq. (3) [5], where r defines the rate over time, and b the burst that can arrive at any instance of time. Fig 3(a) shows an example of the affine function.

$$\gamma_{r,b}(t) = \begin{cases} rt + b & \text{if } t > 0 \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

The peak rate as defined in Eq. (4) [5] only takes on parameter: R, which is the rate over time. An example of the peak rate function is shown in Fig. 3(b).

$$\lambda_R(t) = \begin{cases} Rt & \text{if } t > T \\ 0 & \text{otherwise} \end{cases} \qquad (4)$$

The rate latency function is defined in Eq. (5) [5], where R denotes the rate over time, and T the latency. Fig. 3(c) depicts and example of the rate latency function.

$$\beta_{R,T}(t) = R[t-T]^+ = \begin{cases} R(t-T) & \text{if } t > T \\ 0 & \text{otherwise} \end{cases} \qquad (5)$$
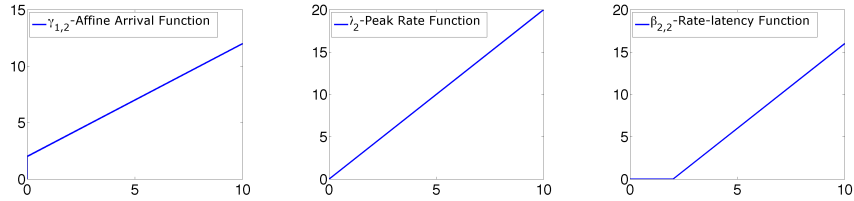
(a) Affine function with: $r = 1$ and $b = 2$ ($\gamma_{1,2}(t)$) (b) Peak rate function with: $R = 2$ ($\lambda_2(t)$) (c) Rate latency function with: $R = 2$ and $T = 2$ ($\beta_{2,2}$)

As seen from Fig 2, the outputs from a service element are: An upper and lower bound of the traffic flow, leaving the service element $\{\alpha'^u, \alpha'^l\}$ and the

upper and lower service bounds $\left\{\beta'^u, \beta'^l\right\}$ of the service remaining, after serving the traffic flow traversing the element. For a scenario as depicted in Fig 2 where only one flow traverses the service element, the output bounds: $\left\{\alpha'^u, \alpha'^l\right\}$ and $\left\{\beta'^u, \beta'^l\right\}$ are found according to Eq. (6)-(9) [8], where the definitions of $\otimes, \oslash, \overline{\otimes}$ and $\overline{\oslash}$ follows the same definitions in as regular NC and can be found in [5].

$$\alpha'^u = \min\left\{(\alpha^u \otimes \beta^u) \oslash \beta^l, \beta^u\right\} \tag{6}$$

$$\alpha'^l = \min\left\{\left(\alpha^l \oslash \beta^u\right) \otimes \beta^l, \beta^l\right\} \tag{7}$$

$$\beta'^u = \left(\beta^u - \alpha^l\right) \overline{\oslash} 0 \tag{8}$$

$$\beta'^l = \left(\beta^l - \alpha^u\right) \overline{\otimes} 0 \tag{9}$$

In Fig. 2 only one traffic flow traverses the scheduling element. However it is often the case that the service element serves a number of flows. This is done according to some predefined scheduling discipline. In this work we confine ourselves to Fixed Priority (FP) scheduling, and First In First Out (FIFO) scheduling. For FP scheduling in a preemptive scheduling setting the service curves $\left\{\beta^u_{FP,\alpha^x}, \beta^l_{FP,\alpha^x}\right\}$, available for a particular flow α^x can be found according to Eq. (10) and (11)[8].

$$\beta^u_{FP,\alpha^x} = \left(\beta^u - \alpha^l_{AGG}\right) \overline{\oslash} 0 \tag{10}$$

$$\beta^l_{FP,\alpha^x} = \left(\beta^l - \alpha^u_{AGG}\right) \overline{\otimes} 0 \tag{11}$$

α^u_{AGG} and α^l_{AGG} are the aggregate i.e. the sum of the upper and lower arrival bounds in set $\mathcal{P}$. In this case set $\mathcal{P}$ contains the flows which have higher priority than α^x. α^u_{AGG} and α^l_{AGG} are found according to Eq. (12) and (13).

$$\alpha^u_{AGG} = \sum_{i \in \mathcal{P}} \alpha^u_i \quad (12) \qquad \alpha^l_{AGG} = \sum_{i \in \mathcal{P}} \alpha^l_i \quad (13)$$

Similarly the service available for flow α^x in the case of FIFO scheduling can be found according to Eq. (10) and (11). The aggregates of the cross flow are found according to Eq. (12) and (13), however here the set $\mathcal{P}$ contains all cross flows. Note that this is using the principle of Blind Scheduling [5].

The delay bound for traversing a single service element can be found according to Eq. (14) [8]. Similarly the bound of needed buffer capacity for a single service element can be found according to Eq. (15) [8].

$$d_{max} \leq \sup_{\phi \geq 0}\left\{\inf\left\{\tau \geq 0 : \alpha^u(\phi) \leq \beta^l(\phi + \tau)\right\}\right\} \overset{\text{Def}}{=} \text{Del}(\alpha^u, \beta^l) \tag{14}$$

$$b_{max} \leq \sup_{\phi \geq 0}\left\{\alpha^u(\phi) - \beta^l(\phi)\right\} \overset{\text{Def}}{=} \text{Buf}(\alpha^u, \beta^l) \tag{15}$$

In the case where the delay and the utilized buffer bounds should be for the whole system or a specific part of the system, the bounds are found according to Eq. (16) and (17). Here the lower service bounds $(\beta^l_1, \beta^l_2, \ldots \beta^l_n)$ of the of the service provided to the flow in question are concatenated by use of the min-plus convolution ($\otimes$). As shown in [5] the concatenation gives more tight delay

bounds that simply summing the delays experienced in every traversed service element. It can be shown that the buffer bound is similar, whether or not the concatenation is applied.

$$d_{max} \leq Del\left(\alpha^u, \beta_1^l \otimes \beta_2^l \otimes \cdots \otimes \beta_n^l\right) \tag{16}$$

$$b_{max} \leq Buf\left(\alpha^u, \beta_1^l \otimes \beta_2^l \otimes \cdots \otimes \beta_n^l\right) \tag{17}$$

4 Ethernet/IP Based In-Car Network RTC Model

Having outlined the fundamentals of RTC, we now describe how RTC can be used to model the network described in Sec. 2. In the following we will describe how to model: A node, the traffic in the network, and a switch. Note that the time units of the model has been chosen as ms, and the data units as Bytes.

4.1 Node Modelling

The end nodes in the network are all 100 Mbit/s full-duplex Ethernet nodes. As described in Sec. 2, we have chosen to introduce a 0.01 ms processing delay at the IP layer. The link-speed constraint and the processing delay constraint, are used as input to a single service element, modelling a node. The upper service curve (β^u) is modelled as a peak-rate service curve ($\lambda_R(t)$), where the rate (R) is: 100 Mbit/s = 12500 Byte/ms. The lower service curve is modelled as a rate-latency service curve $\beta_{R,T}(t)$, with a rate (R) similar to the upper service bound, as the Ethernet link guarantees 100 Mbit/sec. The delay part (T) of the service curve is set to 0.01, as we have chosen a 0.01 ms processing delay. Hence, the upper and lower service bounds of a node in the network become: $\left\{\beta^u = \lambda_{12500}, \beta^l = \beta_{12500,\ 0.01}\right\}$

4.2 Traffic Modelling

In this work we have chosen to confine ourselves to affine arrival bounds, bounding the output of the physical layer from the nodes. As the bound is given for the physical layer, the flow data in Table 1 has been converted to affine arrival bounds, by adding the sum of the headers appended along with the Ethernet inter-frame gap, which is also considered as a part of a packet. The affine arrival bounds are presented in Table 2. Note that the minimum physical size of an Ethernet frame, including all headers and the inter-frame gab, is 88 Bytes. Packets are zero padded, if they are smaller. Also note that the "Controller aggregate" flow in Table 2 is an aggregate of the Controller flows, which has been created to simplify the model.

4.3 Switch Modelling

As described in Sec. 2, the network is prioritized by use of the methods defined in IEEE 801 P and Q. By use of the prioritization field (3 bit field), the switch

Flow	Rate	Burst	Flow Bound	Flow	Rate	Burst	Flow Bound
Right wheel sensor → Dynamic stability control							
Sensor 1	35.2	88	$\gamma_{35.2,88}$				
Left wheel sensor → Dynamic stability control							
Sensor 2	35.2	88	$\gamma_{35.2,88}$				
Dynamic stability control → Digital motor eletronics							
Controller 1	8.8	88	$\gamma_{8.8,88}$	**Controller 2**	4.4	88	$\gamma_{4.4,88}$
Controller 3	4.4	88	$\gamma_{4.4,88}$	**Controller 4**	17.6	88	$\gamma_{17.6,88}$
Controller 5	0.88	88	$\gamma_{0.88,88}$	**Controller 6**	0.088	88	$\gamma_{0.088,88}$
Controller 7	0.88	88	$\gamma_{0.88,88}$	**Controller 8**	0.44	88	$\gamma_{0.44,88}$
Controller 9	8.8	88	$\gamma_{8.8,88}$				
Rear view camera → Head unit							
Camera	969.812	1542	$\gamma_{969.812,1542}$				
Combox → Head unit							
BT Headset	70.4	88	$\gamma_{70.4,88}$				
Multimedia → Head unit							
Audio	183.572	1542	$\gamma_{183.572,1542}$	**DVD**	616.8	1542	$\gamma_{616.8,1542}$

Table 2. Affine arrival bounds of the flows, where the rate is in Bytes/ms and the Burst in Bytes

can prioritize the Ethernet frames. As seen from Table 1, the modelled network only utilize three of these values: 7,6, and 5. Note that 7 is the highest priority. As described in [9] a switch can be modelled in multiple ways. In this work we have chosen to abstract the switch model into a model of the output ports only. This means that we neglect the input buffers, and the backplane in the model. We assume that the processing delay in the input ports is insignificant, since we neglect them. Note that we assume that the switch does not apply cut-through, hence we have to account for the Store-and-Forward (STF) delay. Our modelling of the delay due to STF, is explained in the end of this section. The output ports of a switch are modelled as depicted in Fig. 3, where a chain of service elements, each representing a priority, is used. Each service element uses Blind scheduling (As described in Sec. 3) to model FIFO scheduling among the flows of same priority. As seen from Fig. 3 the remaining upper and lower service bounds $\{\beta'^u, \beta'^l\}$ of the priority 7 service element, are used as input service bounds for the priority 6 service element. Thus the flows of priority 6 get the "service available", after serving the flows of priority 7. Equally the input service bounds of the priority 5 service element, is the remaining service of the priority 6 element.

As the link-speed is 100 Mbit/sec, and we have chosen to introduce 0.003 ms delay in the switch (See Sec. 2), the upper and lower service curves provided to the highest priority service element (Priority 7) become: $\beta^u = \lambda_{12500}$ and $\beta^l = \beta_{12500,0.003}$.

As Ethernet switches are non-preemptive ([10]) all flows, except those served in the lowest priority service element, must expect Head-of-line Blocking (HOL)
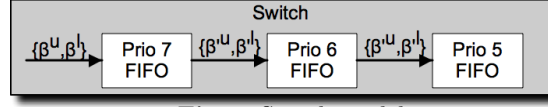
Fig. 3. Switch model

from a lower priority flow. This is modelled conservatively by adding a delay corresponding to the worst case waiting time of a lower priority flow $\frac{1542 \text{ Bytes}}{12500 \text{Bytes/ms}} = 0.1234$ ms, to flows exposed to HOL per scheduler.

As mentioned above, the service curves in the model do not incorporate the delays due to STF. As these only depend on the frame length, and not on delays due to cross flows[2], the delay per receiver where STF occur can be calculated, and then added manually, according to Eq. (18), where FS is the size of the transmitted frame (denoted as the burst in Table 2).

$$t_{stf} = \frac{FS \text{ Bytes}}{12500 \text{ Bytes/ms}} \quad (18)$$

5 Ethernet/IP Based In-Car Network Simulation Model

To compare the results of previously described RTC model with commonly used methods to evaluate network performance, the calculated bounds are compared with network simulation results of the use-case network. The simulation model is created in OMNET++ (V. 3), using the INET Framework.

The nodes in the use-case network are modelled in the following way: On the application layer, the *UDPApp* from the INET framework is used to represent a flow; meaning that if a node has multiple flows, it has multiple entities of the *UDPApp*. Each application is configured with a packet size according to Table 1. The cycle time is uniformly distributed between t_{min} and t_{max}, where t_{min} is the minimum cycle time from Table 1, and $t_{max} = 1.5 \cdot t_{min}$. This jitter is introduced to ensure simulation of non-synchronized flows. In reality the jitter might be smaller, however as we want to investigate the worst case, we are interested in this jitter to "seek" the worst/most congested state of the network.

The link layer is configured to be 100 Mbit/s Ethernet with VLAN based prioritization. The ability to append VLAN headers is not at part of the standard INET framework, but has been made possible due to the work in [11]. From this work it has also been made possible to simulate VLAN based switches. The switches are modelled as 100 Mbit/s switches, using the VLAN prioritization from [11], and the switch components from the INET framework. On top of Ethernet, the simulated network uses IPv4 and UDP according to the specifications in Sec. 2.

As we seek to investigate the maximum delay of every flow, we have to ensure it is observed in the simulation. In this work we assume that a simulation time of 5 hours is sufficient to observe the a realistic maximum delay. Hence we have

---

[2] As the flows already are interleaved when they are transmitted.

configured the simulation with a simulation time of 5 hours. As we introduced the jitter of every flow, and each flow is appended to its own random generator, a long simulation time is as sufficient as many repetitions with different seeds.

6   Results

In Table 3 we present the delays obtained from the RTC model, along with the maximum observed delays obtained by network simulation. As seen from the

Flow	RTC	Maximum observation from simulation
Right wheel sensor → Dynamic stability control		
Sensor 1	0.512	0.303
Left wheel sensor → Dynamic stability control		
Sensor 2	0.38	0.181
Dynamic stability control → Digital motor electronics		
Controller aggregate	0.418	0.177
Rear view camera → Head unit		
Camera	1.601	0.78
Combox → Head unit		
BT Headset	2.575	0.547
Multimedia → Head unit		
Audio	2.689	0.776
DVD	2.448	0.778

Table 3. End to end delay bounds of the flows in ms. The delays presented are those obtained from the RTC model, and the maximum observed in the simulation

table, the flows of high priority (*Sensor 1, Sensor 2* and *Controller aggregate*) have the smallest delays. This is as expected since they are not interfered by any cross flows. Furthermore as expected, the *Audio* and the *DVD* flows have the highest delays as they are of the lowest priority.

To compare the results from the RTC model and the simulation, the overestimation from the RTC model against the simulation is shown in Fig. 4. As seen from the figure, the overestimation by the RTC model seems significant, the mean overestimation is 179%. However the results from the RTC model are hard-bounds derived from an analytical model, thus one can be confident that no delays will arise greater than the calculated bound. This is a fundamental property of the RTC method. Regarding simulation one has to realize that it is impossible to know when the absolute worst case delay has been seen. As explained in Sec. 5, we assume that a simulation time of 5 hours is enough to reveal the true worst case of the system, however we cannot guarantee this.

The significant difference in the overestimation between the last three flows and the remaining four, is due to the fact that the three flows (*BT Headset, Audio* and *DVD*) have high bursts, and traverses the same service elements multiple

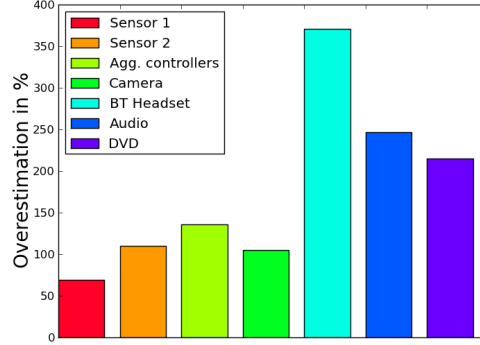**Fig. 4.** The overestimation, per flow, for the RTC model compared to the simulation

times, thus they are multiplexed several times. In reality when flows are multiplexed once and traverses the same path, they do not interfere each other any more. In the domain of NC this problem is referred to as Pay Multiplexing Only Once (PMOO), and is addressed in e.g. [12,13,14,7]. However to the knowledge of the authors, the methods derived have not (yet) been proved within the domain of RTC. Hence, methods can be applied to reduce the overestimation.

7   Conclusion and Outlook

An RTC model of an Ethernet based use-case in-car network has been proposed. The delays obtained by the calculations in RTC, have been compared with delays obtained by performing simulation of the same use-case network. It can be concluded from the comparison that the overestimation in RTC for this scenario is significant, the level depends on the nature of the flow, and the path it traverses. As explained in Sec. 6, there are known optimizations that, (when proven valid in the domain of RTC), can be applied. Even though the overestimation is significant it is important to emphasize that, the maximum observed delay from simulation, is not a bound and delays greater than the observed can occur. This means that as we know the delays obtained from RTC are overestimated, and those from simulation might be "underestimated", the "true delays" are found in between.

Due to the overestimation in RTC the price of over-dimensioning a network can be high. However as in-car networks are safety critical, hard bounds are needed, and bounds need to be found by methods like RTC. Therefore, it is of a high importance to find methods to reduce the overestimation in RTC, by improving the bounding methods. As mentioned in Sec. 6, one way would be to investigate how the principles of PMOO can be applied within the domain of RTC.

8 Acknowledgment

References

1. R. Steffen, R. Bogenberger, J. Hillebrand, W. Hintermaier, A. Winckler, and M. Rahmani, "Design and realization of an ip-based in-car network architecture," in *The First Annual International Symposium on Vehicular Computing System*, (Dublin, Ireland), July 22-24, 2008.
2. L. Thiele, S. Chakraborty, and M. Naedele, "Real-time calculus for scheduling hard real-time systems," *Circuits and Systems, 2000. Proceedings. ISCAS 2000 Geneva. The 2000 IEEE International Symposium on*, vol. 4, pp. 101 – 104 vol.4, 2000.
3. R. Cruz, "A calculus for network delay. i. network elements in isolation," in *Information Theory, IEEE Transactions on*, vol. 37, pp. 114–131, Jan 1991.
4. R. Cruz, "A calculus for network delay. ii. network analysis," *Information Theory, IEEE Transactions on*, vol. 37, pp. 132 – 141, Jan 1991.
5. J.-Y. Le Boudec and P. Thiran, *Network calculus: a theory of deterministic queuing systems for the internet.* Springer-Verlag New York, Inc., May 10, 2004.
6. C.-S. Chang, *Performance Guarantees in Communication Networks.* London, UK: Springer-Verlag, 2000.
7. M. Fidler, "A survey of deterministic and stochastic service curve models in the network calculus," *IEEE Communications Surveys and Tutorials*, vol. 12, pp. 59–86, Jan 2010.
8. E. Wandeler, L. Thiele, M. Verhoef, and P. Lieverse, "System architecture evaluation using modular performance analysis: a case study," *International Journal on Software Tools for Technology Transfer (STTT)*, vol. 8, Oct 2006.
9. J. Georges, T. Divoux, and E. Rondeau, "Comparison of switched ethernet architectures models," *Emerging Technologies and Factory Automation, 2003. Proceedings. ETFA '03. IEEE Conference*, vol. 1, pp. 375 – 382 vol.1, 2003.
10. R. Seifert, *The Switch Book: The Complete Guide to LAN Switching Technology.* New York, NY, USA: John Wiley & Sons, Inc., 2000.
11. M. Rahmani, K. Tappayuthpijarn, B. Krebs, E. Steinbach, and R. Bogenberger, "Traffic shaping for resource-efficient in-vehicle communication," *Industrial Informatics, IEEE Transactions on*, vol. 5, no. 4, pp. 414 – 428, 2009.
12. J. Schmitt and F. Zdarsky, "The disco network calculator: a toolbox for worst case analysis," *Proceedings of the 1st international conference on Performance evaluation methodolgies and tools*, Jan 2006.
13. J. Schmitt, F. Zdarsky, and I. Martinovic, "[technical report] performance bounds in feed-forward networks under blind multiplexing." URL: http://disco.informatik.uni-kl.de/publications/SZM06-1.pdf, Jan 2006.
14. A. Bouillard, B. Gaujal, S. Lagrange, and E. Thierry, "Optimal routing for end-to-end guarantees: the price of multiplexing," *Proceedings of the 2nd international conference on . . .*, Jan 2007.