TECHNICAL REPORT IDE0892

# Improving the scheduling analysis of hard real-time traffic by analyzing traffic dependencies

KRISTINA KUNERT


School of Information Science, Computer and Electrical Engineering
Halmstad University

Halmstad, Sweden, November 2008

# Table of Contents

# Abstract

*Support for hard real-time traffic requires throughput guarantees for packets with timing constraints. The deterministic analysis available for real-time communication has its origin in the area of task scheduling in real-time systems and has been mapped onto the communication context. Mapping uniprocessor scheduling techniques directly on a multichannel network with the possibility of concurrent transmissions, however, introduces pessimism to the analysis. This paper presents an approach of successfully increasing the amount of possible guarantees by analyzing traffic interdependencies. By taking into account those traffic interdependencies and integrating concurrent transmissions into the analysis, the amount of throughput guarantees can be increased considerably as shown in our simulations.*

.

# 1 Introduction

Supporting hard real-time traffic means being able to guarantee the throughput of a certain number of packets with timing constraints. If a communication protocol has real-time properties it means that it is designed such that it tries to meet the deadlines of as many packets as possible or alternatively the most important packets. This is when scheduling comes into play. When scheduling packets, they are ordered to maximize some type of performance parameter, e.g., throughput, priority, or jitter. Without scheduling, packets are typically sent in the order they arrive ready for transmission, i.e., according to the First Come First Served (FCFS) service principle.

To be able to state a deadline guarantee, i.e., enabling packets to be scheduled such that they meet their deadlines, the worst-case delay must be analyzed for all existing traffic flows plus the new one and must be upper-bounded. This implies that when scheduling packets with real-time constraints over a point-to-point link, a deterministic queuing principle (or service discipline) with a known maximum delay must be used [10]. Even though FCFS is such a deterministic queuing principle [3], there are ordering principles specially developed for real-time communication. Two examples of such are delay-EDD (Earliest Due Date) [4] and jitter-EDD [9]. Both delay-EDD and jitter-EDD are based on Earliest Deadline First (EDF) scheduling [6], where the packets are sorted according to their deadlines.

The choice of EDF in this work is based mainly upon two reasons. Firstly, EDF is an optimal scheduling algorithm in the meaning that if a feasible schedule for a set of tasks exists, then EDF will be able to find it. Secondly, a complete analytical framework is available for EDF under the assumption of error-free communication. Therefore, in order to determine the performance characteristics for hard real-time traffic, a deterministic analysis of the delay of a network architecture under the assumption of EDF scheduling in the communicating end nodes and over the medium is given. The required input parameters for the analysis are the traffic characteristics of a given application, provided in the form of real-time traffic flows.

EDF is traditionally used in uni-processor systems for scheduling periodic real-time tasks. Here a so-called schedulability analysis method has been developed [6] that uses a well-proven algorithm to check the feasibility of a given real-time task set before it is executed. If the set is feasible, it means that all tasks in the set will meet their deadlines at each instance of their period, given that they are scheduled according to EDF. To be able to state a deadline guarantee for a new task that enters into the system, the worst-case delay must be analyzed for all existing tasks in the set plus the new one. Only if the delay bounds for this new set of tasks can be met, without violating already stated delay bound guarantees, the new task is accepted. The real-time schedulability analysis is thus used as an admission control mechanism. To determine if a task set is feasible, the analysis uses a two-step method: a utilization check and a workload check. The utilization condition is basic: the utilization must not be higher than 100%. However, this condition is not sufficient to guarantee that no deadlines will be missed, unless for the case that all task

period times are equal to their deadlines. Therefore the workload check is made, stating that for all values of t, the workload must not be greater than t, which ensures that the momentary utilization is not greater than 100%. A positive response from both checks provides the deadline guarantees necessary for real-time applications.

It should be noted that, unlike in many cases of processor scheduling, scheduling in a communication context is defined to be non-preemptive, meaning that once the transmission of a packet has started it cannot be stopped for the transmission of another packet. However, the EDF scheduling theory has been mapped to a networking context in [5] using traffic flows rather than task sets. The utilization and workload checks only have to be executed whenever a new traffic flow is requesting network allocation; otherwise, as soon as the tests have returned positive answers, all deadlines can be guaranteed as long as system or network parameters are maintained.

# 2 Traffic flow abstraction

The concept of logical real-time channels (RT channels or RTCs), also referred to as RTVCs (Real-Time Virtual Channels), was introduced in [4]. A real-time channel is an abstraction of a traffic flow over a link or a network, where resources have been allocated to guarantee a certain minimum throughput and a bounded end-to-end delay.

Each traffic flow $F_q$ has the following attributes

- source $S_q$ (sending node)
- destination $R_q$ (receiving node)
- period $P_q$ (minimum message interarrival time)
- deadline $D_q$ (end-to-end delay bound)
- capacity $C_q$ (maximum message transmission time each period)

The source nodes are bound to behave according to the traffic specifications and must not violate them by e.g. sending more frequently. The total number of flows in the network is denoted by $Q$, and $1 \leq q \leq Q$.

# 3 Real-time analysis in a communication context

The task of analysing traffic flows, which resources they use and if they constitute a feasible system, i.e., if a schedulable traffic allocation is possible, can be mapped onto the problem of uniprocessor task scheduling [5]. This means that during the course of this analysis, those traffic flows are looked upon as synchronous, periodic tasks which have to be scheduled on the network. The capacity $C_q$ of a traffic flow corresponds to the worst-case execution time (WCET) of the task to be scheduled.

Assuming a worst-case scenario, i.e., when all of the nodes in the suggested network want to send data traffic to the same destination in any given time slot, there is always at least one packet, namely the one with the earliest deadline, that can be guaranteed access to the medium due to the assumption of centralized EDF scheduling (and locally in the end node queues). This means that a capacity of 1 always can be guaranteed for hard real-time traffic. Assuming periodic traffic, basic EDF theory [6] can be used to analyse the system. According to EDF scheduling theory, the utilisation of a hard real-time system is defined as:

$$U = \sum_{q=1}^{Q} \frac{C_q}{P_q} \leq U_{\max} \tag{1}$$

Translated into a communication context, this means that $U$ defines the utilization of the periodic traffic in the network, where $C_q$ is the maximum transmission time for data per period, $P_q$ is the period of the data traffic, and $U_{max}$ denotes the maximum utilisation of the network by hard real-time traffic that must not be exceeded.

For further description of the feasibility test in detail, the following concepts from the area of real-time task scheduling have to be introduced into the discussion.

### *Hyperperiod*

Given a task set consisting of periodic tasks, the hyperperiod is the least common multiple of all periods of those tasks, i.e., the length of time from when all tasks' periods start at the same time until they start at the same time again. In the context of communication, a task corresponds to a communication channel.

### *Busyperiod*

A busyperiod is any interval of time in which the resource is not idle, i.e., the busyperiod of a communication link is generally any time interval during which the link is not idle.

*Workload function*

The traffic demand on the analyzed link corresponds to the processor demand in a real-time system and can be defined by a workload function, *h(t)*. It is the sum of all the capacities of the instances of tasks *q* with an absolute deadline less than or equal to a point in time *t*, where *t* is the time elapsed from the start of the hyperperiod. Mapping this on the communication scenario, the function is summing up the maximum packet transmission times per period, $C_q$, of all instances of RT channels *q* that have an absolute deadline less than or equal to *t*. The workload function is calculated as follows [1, 2, 7].

$$h(t) = \sum_{q=1, D_q \le t} \left\lfloor \frac{t + P_q - D_q}{P_q} \right\rfloor \cdot C_q \tag{2}$$

Feasibility tests investigate if a system is in a feasible system state, i.e., if all the tasks in the system are feasible. Following the discussion from above, feasibility testing in a communication context spells checking if the admission of an additional logical RT channel still results in a set of feasible RT channels. Feasibility testing is performed in two steps, each being a test of its own, with the following two constraints having to be met.

*Constraint 1*

The utilization of the link has to be less than or equal to one (100 %), following the previous discussion about guaranteed access to the medium.

□

In order for the task set to be schedulable, or, in this case, for the flows to be allocatable over the network link, the utilization parameter has to be less than or equal to 1 (100 %). This means that $U_{max}$ is equal to 1 and the constraint therefore defined as

$$U = \sum_{q=1}^{Q} \frac{C_q}{P_q} \le 1 \tag{3}$$

This condition is necessary, but not sufficient to be able to ensure a 100 % success rate for the transmission of real-time traffic with guaranteed deadlines. This means, as it is hard real-time traffic that is analyzed, solely the fulfilment of this utilization constraint will not be sufficient to guarantee that no deadlines will be missed. However, for the case of all deadlines being equal to or longer than their corresponding periods, this test is both necessary and sufficient. For schedulability reasons, it is assumed in the analysis that the utilization of each single task, i.e., flow or channel, is not higher than 100 %.

*Constraint 2*

For all values of t, the workload function h(t) has to be less than or equal to t.

□

$$h(t) = \sum_{q=1, D_q \le t} \left\lfloor \frac{t + P_q - D_q}{P_q} \right\rfloor \cdot C_q \le t \quad \forall t \tag{4}$$

This second condition, introduced in [1, 2] and generalized in [7], was added in order to insure the continued feasibility of the system when adding a new task, or in this case, a new traffic flow.

As mentioned earlier, this feasibility analysis is developed for real-time systems and therefore assumes fully preemptive tasks. In network communication, packets normally cannot be preempted and therefore the possibility of further delay has to be taken into account. A constant $T_{blocking}$ is defined which denotes the maximum blocking time that one packet can introduce into the system, i.e., $T_{blocking}$ equals the transmission time of a packet with the maximum packet size. Furthermore, the sending and receiving of control information will introduce an additional delay, represented here by the constant $T_{control}$. These compensations result into a shortening of the delay bound.

$$D_q' = D_q - T_{blocking} - T_{control} \tag{5}$$

This means that the workload function is remodelled as follows.

$$h(t) = \sum_{q=1, D_q' \le t} \left\lfloor \frac{t + P_q - D_q'}{P_q} \right\rfloor \cdot C_q \tag{6}$$

Unfortunately, the second constraint, in the form given above, does not lend itself to calculation particularly well due to the high computational complexity it introduces into the feasibility test because of the continuous property of time. It is shown in [8] how it is possible to reduce the time and memory complexity of the second constraint check by reducing the number of instances of calculation to include merely a reduced number of integer time values during an interval upperbounded by $P_{busy1}$, the first busyperiod during the first hyperperiod of the schedule where all periods start at time zero.

The value of $P_{busy1}$ can be calculated by the following recursive algorithm [7, 8]

$$\begin{cases} P_{busy}^{(0)} = \sum_{i=1}^{n} C_i \\ P_{busy}^{(k)} = W\left(P_{busy}^{(k-1)}\right) \end{cases} \tag{7}$$

where $W$ denotes the cumulative workload function and $W(t)$ is the cumulative workload at the point in time $t$. The recursive algorithm is calculated until

$$P_{busy}^{(k)} = P_{busy}^{(k-1)} \tag{8}$$

Then we set

$$P_{busy1} = P_{busy}^{(k-1)} \tag{9}$$

$W(t)$ is calculated as the sum of maximum message transmission times of the messages released before $t$, i.e.

$$W(t) = \sum_{i=1}^{n} \left\lceil \frac{t}{P_i} \right\rceil \cdot C_i \tag{10}$$

If $h(t) \leq t$ in the first busyperiod of the hyperperiod in the supposed schedule to come, then $h(t) \leq t$ for all $t$. The following upper bound of the interval to be checked is therefore an improvement of the algorithm above.

$$t : 1 \leq t \leq P_{busy1} \quad \forall t \in \mathbb{N} \tag{11}$$

Furthermore, one does not need to check each integer from the first time slot, but only the integers $t$ where

$$t \in \bigcup_q \left\{ m \cdot P_q + D_q : m = 0,1,2... \right\} \tag{12}$$

and where

$$t \in \left[ 1; P_{busy1} \right] \tag{13}$$

In the case of non-preemptive communication, the local delay bound again has to be further adjusted due to the possibility of blockage and control traffic, which finally results in the following set.

$$t \in \bigcup_q \left\{ m \cdot P_q + D_q' : m = 0,1,2... \right\} \tag{14}$$

where

$$t \in \left[ 1; P_{busy1} \right] \tag{15}$$

The upper and lower boundaries of the interval for $t$ stay unmodified. Only when both the utilization constraint and the workload constraint are fulfilled can a feasible traffic allocation be guaranteed.

# 4 Improved hard real-time traffic support

The real-time schedulability analysis described previously uses a worst-case situation as its base assumption. In this report we suggest an improvement for the real-time analysis when used in networks able to transmit concurrent traffic. Taking into consideration traffic interdependencies, the analysis decreases the amount of pessimism included in the original real-time analysis.

## 4.1 Traffic analysis algorithm

When using the feasibility analysis on a set of real-time flows over a network, it will return a simple 'yes/no' answer, providing the result to the question if all flows could be allocated over the network as if the network was a single resource. This is due to the original application of this analysis to uniprocessor task scheduling. In reality, a network is a set of overlapping resources, depending upon the physical and logical network architecture and the medium access control method used. In the presented context, a resource is constituted by a sender, a receiver and the unidirectional light path connecting them. If any of those components is busy, the whole resource is seen as reserved. This leads to the following definitions.

*Definition 1*

A **resource** consists of a sender, a receiver and the unidirectional light path connecting them.

□

*Definition 2*

In this analysis, a resource is defined to be busy if either the sender at the source node is transmitting to any other node connected to it or the receiver at the destination is receiving from any other node connected to it, or both are participating in the communication (sending and receiving respectively) over the unidirectional link between them, i.e., they actually use the medium for which the current traffic flow competes.

□

The reason for this definition of the 'busy' principle is the fact that in the suggested network architecture, the sender at each source node only can send to one destination at a time, and the receiver at each destination node only can receive from one source at a time (not taking into account the communication with the protocol processor which is allocated on a separate control channel). This automatically leads to the consequence that the resource, and therefore the unidirectional link between those two nodes, cannot be used for other communication as soon as one of them is busy.

The usage of the original feasibility analysis contains a rather large amount of pessimism as it completely dispenses with the possibility of simultaneous transmissions over resources independent of each other. The approach presented here concentrates upon the construction of virtual overlapping subnets in order to analyze them individually according to the already described feasibility test.

*Definition 3*

One subnet consists of one main traffic flow and all other traffic flows it shares at least one part of the resource with, i.e., the flows which have the same source or destination node (or both) as the main studied flow.

□

*Definition 4*

Two subnets are overlapping if they share at least the sender at the source node or the receiver at the destination node. As long as two subnets are not overlapping, simultaneous transmissions in them can occur at any time.

□

A subnet, which is a set of real-time channels, has to be found for each individual traffic flow over the network, and both the utilization and the workload tests have to be applied on all these subnets. Each subnet contains solely those traffic flows that are directly competing with the main studied flow, not taking into account flows that in their turn are competing with those. The details are described in the following paragraph.

In order to analyze the interdependencies of the traffic flows contained in one subnet, it has to be investigated for each single flow which parts of the resource it shares with any other flow. Flow $F_i$, which is the studied RT channel, is assumed to be characterized by its source $S_i$, its destination $R_i$, its period $P_i$, its deadline $D_i$, and its capacity $C_i$. The link between $S_i$ and $R_i$ is denoted $K_i$. The resource including $S_i$, $R_i$ and $K_i$ is denoted $O_i$. $F_i$ might have to share either its source or its destination node (or in fact both) and therefore the first step has to be to create a set $M_i$, i.e., a subnet, with all traffic flows $F_j$ that either have $S_i$ as its source node $S_j$ or $R_i$ as its destination node $R_j$. These flows are the following:

$$M_i = \bigcup_{j=1}^{Q} \left\{ F_j \,\middle|\, S_j = S_i \text{ or } R_j = R_i \right\} \tag{16}$$

This set also includes the studied flow $F_i$ itself and those flows parallel to it, i.e., which share all parts of the resource with the currently studied flow (i.e., it is an inclusive OR not exclusive OR in the set definition). However, apart from flows competing with $F_i$, no other flows competing merely with any flow $F_j$ are included in the set.

## 4.2 Improved, less pessimistic feasibility analysis

In order to be able to decide if all tasks or flows of the ones competing for the same resource can be allocated in a way so that no deadline will be missed, a feasibility analysis has to be conducted for each set of competing flows, i.e., subnet. The number of subnets is equal to the number of traffic flows, i.e., $Q$. The traffic interdependency analysis introduced in the previous chapter opens up the possibility to now calculate the utilization $U_i$ of $O_i$ by all flows $F_j$ in $M_i$ as

$$U_i = \sum_{j=1}^{Q} e_j \quad where \quad e_k = \begin{cases} \frac{C_k}{P_k} & if \quad F_k \in M_i \\ 0 & otherwise \end{cases} \tag{17}$$

It is still a necessary condition for this utilization to be ≤ 1. In order to satisfy the second constraint connected to the workload of the link, the workload function is applied as before, but upon the smaller subnet $M_i$. The result answers the question whether hard real-time traffic can be guaranteed for the studied flow $F_i$.

$$h_i(t) = \sum_{j=1}^{Q} g_j(t) \quad where \quad g_k(t) = \begin{cases} \left\lfloor \frac{t + P_k - D_k'}{P_k} \right\rfloor \cdot C_k & if \quad F_k \in M_i \quad and \quad D_k \le t \\ 0 & otherwise \end{cases} \tag{18}$$

Both the utilization and the workload check are carried out for each set of traffic flows, i.e., each single traffic flow will be studied as the main flow of a set. The feasibility checks will generate the necessary information to decide if the studied flow can be guaranteed to fulfil its timing constraint. A negative answer does in itself not mean that the flow automatically will miss its deadline, but simply that no guarantee for it can be given. $X$ denotes the set containing all traffic flows for which hard real-time can be guaranteed, while $Y$ is the size of this set, i.e., the actual number of flows, for which timely treatment can be guaranteed:

$$X = \bigcup_{j=1}^{Q} \left\{ M_i \,\middle|\, U_i \le 1 \;\; and \;\; (\forall t)\left(h_i(t) \le t\right) \right\} \tag{19}$$

$$Y = |X| \tag{20}$$

When analyzing a hard real-time system, $Y$ must be equal to the number of traffic flows in the system, i.e., $Q$, for the system to be feasible. However, for the analysis of soft real-time systems, $Y$ can be used as a parameter that indicates the degree to which real-time traffic can be guaranteed.

There can still be pessimism contained in this analysis. Flows might be included in several subnets, but due to the possibility of intricate traffic interdependencies, the complexity of an analysis studying dependencies more than one step from the main studied traffic flow might grow immensely fast, and is, for now, outside the scope of this

work. However, the presented approach can increase the capacity which can be guaranteed for hard real-time traffic from a throughput of one packet per time slot to the throughput reached by $Y$ traffic flows instead, with $1 \leq Y \leq Q$, due to the reduction of the number of flows participating in the feasibility analysis to those flows that actually compete in a worst-case situation. In other words, this adapted analysis makes it possible, depending on the actual traffic pattern, to guarantee a total throughput of hard real-time traffic higher than 1.

How can the possibility be excluded that this analysis results are too optimistic? Being too optimistic would infer that there are cases were this analysis method gives a positive answer, while in reality the traffic flow studied will miss its deadline. That in its turn could only happen if the admission control had not taken into account all flows demanding capacity of the resource competed for. As the subnet is defined as the set of all flows that share this resource, no relevant competitors are left out, and therefore the admission control and the feasibility analysis have all necessary information to produce a reliable, nonoptimistic prediction as they calculate on the worst-case situation.

# 5 Simulation analysis

In order to demonstrate the improvement made possible by the usage of the improved feasibility analysis, a simulation program was implemented in Java. The assumptions are a 16 × 16 AWG, which means that there will be 15 end nodes and the protocol processor. The traffic pattern was assumed to be the following. The period as well as the deadline of all real-time traffic flows is 100 time slots and their maximum message transmission time each period corresponds to the length of one time slot. The source of each traffic flow is randomized with an even distribution, while the choice of destination is limited to include nodes contained in destination groups of a certain (variable) size. The maximum number of channels requested in the system is set to 2000. Each data point in the evaluation curves is the result of 100 iterations in order to increase the statistical reliability of the result.

In the first figure, Figure 1, the number of requested real-time channels is increased (in steps of one) from 1 to 2000, after which the network seems to be saturated by the traffic load for all curves, i.e., for all destination group sizes. The parameter plotted in the figure is the throughput in packets per time slot experienced by the guaranteed real-time channels depending on the number of requested real-time channels. (The calculation of the actual number of accepted real-time channels is basic as each channel $F_q$ has a bandwidth utilization of 1% ($C_q$=1, $P_q$=100)). Different curves are plotted reflecting different sizes of destination groups. The cases illustrated in the figure are when each source can send to one possible other destination, or 4, 7 or 14 other destinations, which is the maximum possible number. (All possible sizes of destination groups were simulated, but some are excluded from the figure for readability reasons.) Which destinations are included in each destination group is randomized with an even distribution.

Looking at Figure 1, the maximum throughput by accepted real-time channels is reached at the network saturation point of about 800 RTC requests. Most prominent, however, is the curve for the case of a destination group size of one, where the maximum throughput is 9.53, instead of the remaining values of around 7. The reason for this deviating behaviour is the relatively low probability of overlapping subnets when each traffic flow from a particular source has to have the same destination. The interference between all traffic flows is, over a statistically relevant time seen, smallest in this case when all traffic flows are randomized with an even distribution of source-destination pairs.

The theoretical throughput maximum, denoted $G_{max}$, reachable in a network with this configuration, and under the assumption of equally many traffic flows between any source-destination pair, can be calculated by

$$G_{max} = \frac{N \cdot N_{dest}}{2 \cdot N_{dest} - 1} \tag{21}$$

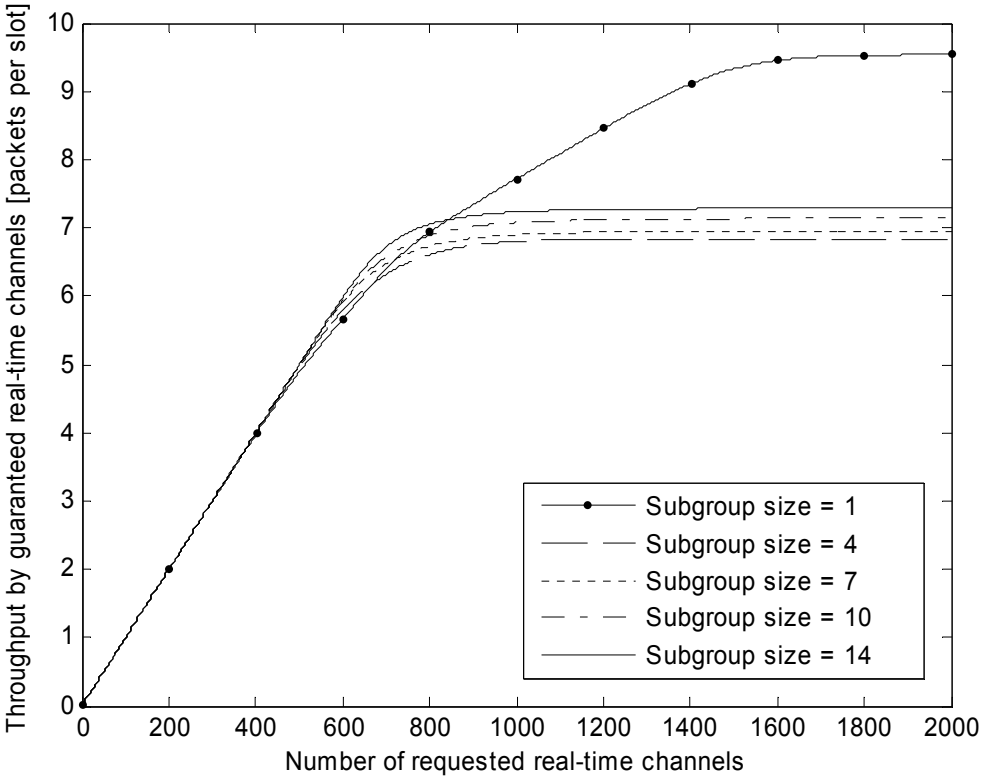


**Figure 1. Throughput by guaranteed traffic flows**

where $N$ is the total number of nodes in the network, and $N_{dest}$ is the number of destinations per destination group. This theoretical average throughput should however only be used for an approximate comparison since it is based on the assumption of a nonrandom even distribution. The theoretically calculated values are compared to the simulation results in Figure 2 and Figure 3.

For small subgroup sizes each source only has a few numbers of possible destinations. This means also that RT channels from different sources are less likely to have the same destinations as the probability of the subgroups overlapping is not so high. The larger the size of the subgroups, the higher is the probability of a destination overlap. In other words, the degree of traffic interdependency is higher for larger subgroups, and therefore the amount of guaranteed throughput is decreasing for increasing subgroup size. As can be seen in Figure 2 and Figure 3, the difference between the simulated and calculated throughput decreases continuously as the number of possible destinations per source increases. For a small number of possible destinations, the lower simulated throughput is due to higher effects of randomness, while the randomly generated groups are more similar to groups with even distribution for large group sizes. The turning point where the simulated throughput changes from a sinking to an increasing trend lies around a destination group size of four. In other words, this is the point where the combined effect of randomness and probability of overlapping subnets is worst. The difference still experienced for a destination subgroup size of 14, i.e. when the destination is randomized between all available destinations in the network, can be found in the fact that the

calculated value assumes an even distribution of destinations, while the simulated value is the result of a random distribution of destinations, but with all destinations having the same probability.

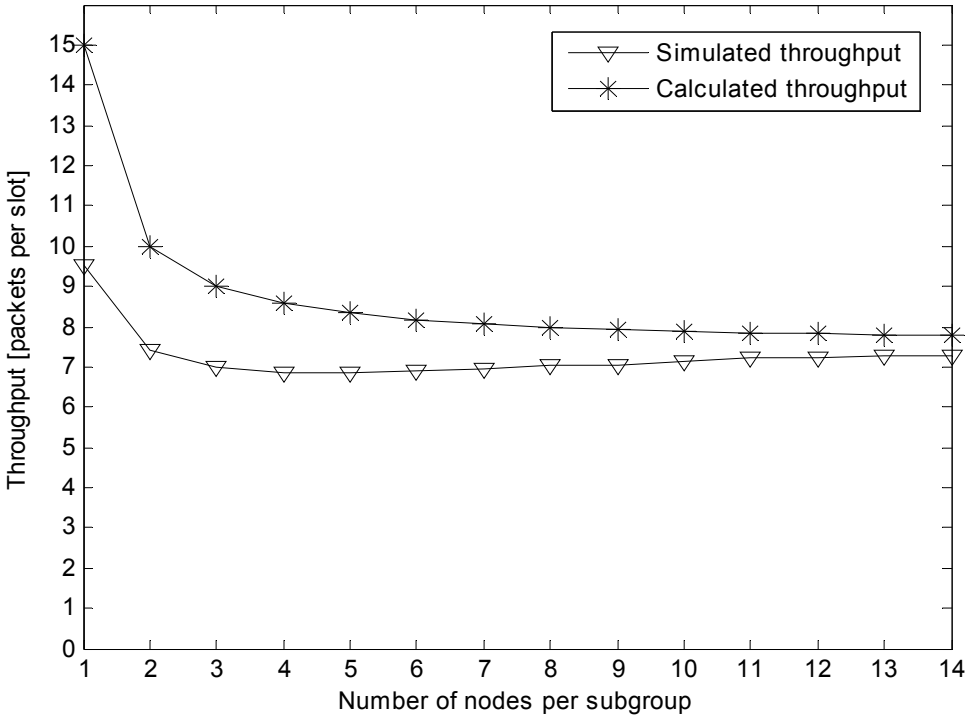


Figure 2. Calculated versus simulated throughput

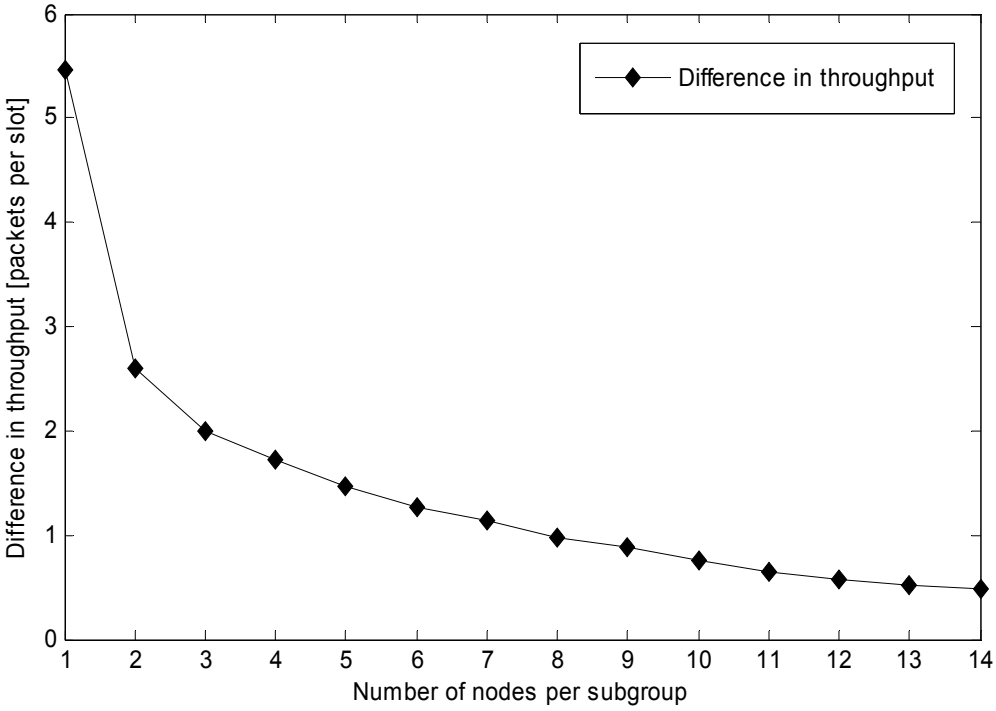


Figure 3. Difference between calculated and simulated throughput

In order to further investigate this behaviour, the throughput of the guaranteed real-time channels depending on the number of destinations per source, i.e., destination group size is studied in Figure 4. The curves illustrate the results for different traffic loads in the system. While 500 RTCs can be accommodated easily by the network, the higher traffic loads can be seen experiencing the same behaviour as described earlier regarding Figure 2. However, this curve shows also that the influence by the low probability of overlapping subnets (which was illustrated by the 'Destination group size = 1' curve in Figure 1 earlier) is observable up to a destination group size of four different destinations per source. In that point the three upper curves have their minimum after which they start increasing.

In summation, the simulations show that the number of real-time channels that can be guaranteed to meet their deadlines is considerably higher when using the suggested traffic analysis in combination with a real-time analysis compared to simply using the real-time analysis. The guaranteed throughput was increased from being one packet per slot to being around seven, i.e., a utilization of about 700%. For some traffic patterns, even higher guaranteed throughputs can be reached as shown by the simulations.

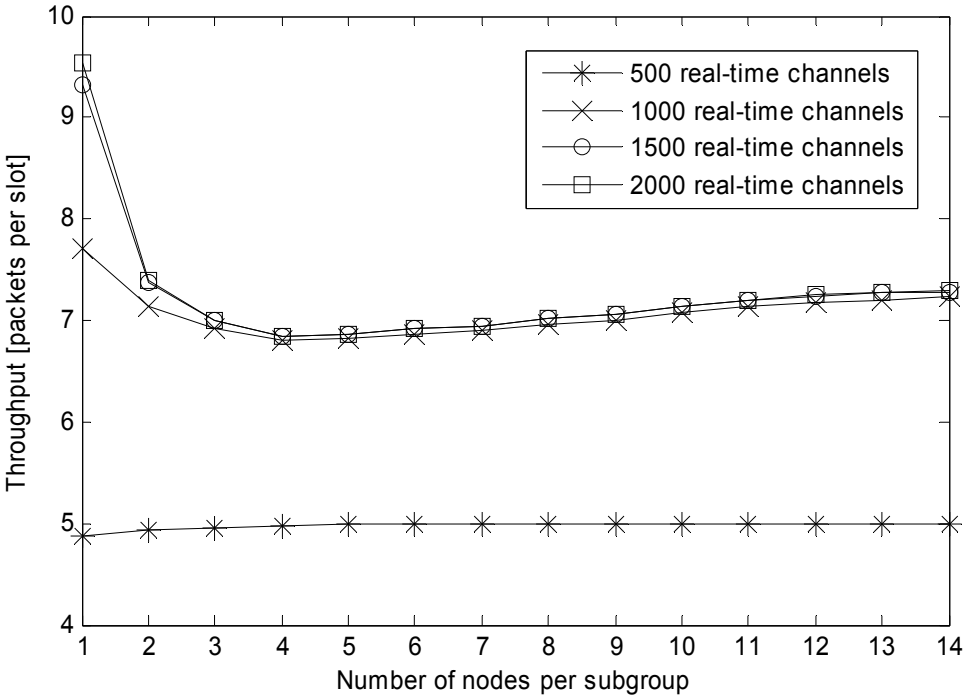


**Figure 4. Throughput depending on subgroup size.**

# 6 Conclusions

Targeting data communication in hard real-time systems, the need of guaranteed throughput and a limited delay bound is important to meet. In order to do that, a real-time analysis was included in the admission control process which used a busy-period analysis method originating from the area of real-time systems, but adapted for a communication context, in its decision process. Being aware of the pessimism incorporated in the analysis, a traffic dependency analysis was described to improve the performance of the analysis when used for multichannel networks where several traffic flows can be scheduled concurrently. Simulating the analysis for different traffic patterns indicated that the throughput guarantee can be improved substantially.

# 7 References


[1] S. K. Baruah, A. K. Mok, and L. E. Rosier, "Preemptively scheduling hard-real-time sporadic tasks on one processor," in Proc. *Real-Time Systems Symposium, 1990. Proceedings., 11th*, 5-7 Dec 1990 1990, pp. 182-190.

[2] S. K. Baruah, L. E. Rosier, and R. R. Howell, "Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor," *Real-Time Systems,* vol. 2, pp. 301-324, November 1990.

[3] X. Fan and M. Jonsson, "Guaranteed real-time services over standard switched Ethernet," in Proc. *Local Computer Networks, 2005. 30th Anniversary. The IEEE Conference on*, 17-17 Nov. 2005 2005, pp. 3 pp.-492.

[4] D. Ferrari and D. C. Verma, "A scheme for real-time channel establishment in wide-area networks," *Selected Areas in Communications, IEEE Journal on,* vol. 8, pp. 368-379, 1990.

[5] H. Hoang and M. Jonsson, "Switched real-time Ethernet in industrial applications - deadline partitioning," in Proc. *Communications, 2003. APCC 2003. The 9th Asia-Pacific Conference on*, 21-24 Sept. 2003 2003, pp. 76-81 Vol.1.

[6] C. L. Liu and J. W. Layland, "Scheduling Algorithms for Multiprogramming in a Hard-Real-Time Environment," *J. ACM,* vol. 20, pp. 46-61, 1973.

[7] M. Spuri, "Analysis of deadline scheduled real-time systems," Institut National de Recherche en Informatique et en Automatique (INRIA), France, Technical Report RR-2772, January 1996.

[8] J. A. Stankovic, M. Spuri, K. Ramamritham, and G. C. Buttazzo, *Deadline Scheduling for Real-Time Systems - EDF and Related Algorithms*: Kluwer Academic Publishers, 1998.

[9] D. C. Verma, H. Zhang, and D. Ferrari, "Guaranteeing delay jitter bounds in packet switching networks," presented at the TriComm '91, Chapel Hill, NC, USA, 1991.

[10] H. Zhang, "Service disciplines for guaranteed performance service in packet-switching networks," *Proceedings of the IEEE,* vol. 83, pp. 1374-1396, 1995.