Extending PMOO Principle to (some) Non-FeedForward Networks

Ahlem Mifdaoui

University of Toulouse-ISAE

Ahlem.mifdaoui@isae.fr

6th April 2016

Introduction

- For **safety-critical systems**, worst-case guarantees are key issues to fulfill **certification** requirements.
- Many challenges arise from conducting such analyses:
  - **Shared resources**, e.g., CPU, cache memories and communication networks
  - Handling **cyclic dependencies** at different system levels, e.g., the software code, the task graph and the network graph.
- Large body of work for WCET and WCRT analyses [1,2] (Wilhelm08, Burns00)
- Only few analytical approaches for WCTT computation.
  $\Rightarrow$ **Need an appropriate approach for WCTT analysis of Networks with cyclic dependencies**

Table of contents

Related Work

Breaking the potential cycles through prohibiting the use of some links or sub-paths to ensure the feed-forward property [3,4] (Schroeder91, Starobinski03)

- **+ + simplify the timing analysis of non-feedforward networks**
- **- - Reliability level deterioration, since the use of some links is forbidden**

Related Work

Computation methods to support cycles using an **iterative approach**:

- Holistic approach [0] (Tindell 94) with various extensions [5,6] (Palencia03, Pellizzoni05)
- Network Calculus focusing on each crossed node delay bound [7,8,9] (Cruz 91, Leboudec00, Thiele08)
- Network Calculus focusing on each crossed node backlog bound [10,11] (Tassiulas 96, Le Boudec01)

$\Rightarrow$ **Pessimistic delay bounds, limiting the network performance in terms of resource-efficiency and system scalability**

Related Work

Computation methods based on **global analysis** along the flow path

- Scheduling theory [12] (Abdelzaher09)
  + + Less pessimistic than holistic approach
  - - Applicable to medium scale networks with 25 nodes
- Network Calculus using the Pay Multiplex Only Once (PMOO) [13] (Amari 16)
  + + Handle large scale of non-feedforward networks
  + + Outperform the conventional approaches
  - - Need to be generalized for any non-feedforward network

Time Stopping Method

This approach has been proposed in [7] and consists of two steps:

- First, a **finite burstiness** bound for transmitted flows is assumed to obtain a set of equations to compute the delay bounds.
- Then, the **feasibility condition** to solve these equations is defined.

Backlog-based Method

- This approach initially proposed in [10] and more recently generalized in [11].
- Maximum backlog bound when considering **non work-conserving** nodes
- Maximum bound on the **total amount of data present in the network at any time**

Limitations: Impact of Congestion

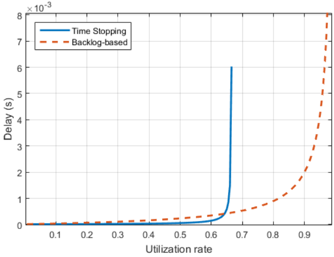

Figure : Upper bound delay for M=4 vs utilization rate

- a network of 4 nodes
- each node generates a flow $(864bit, \rho)$-constrained,
- varying the rate ρ to obtain an utilization rate between 1% and 100%.

Limitations: Impact of Congestion

- Congestion induces increased network delay bounds for both methods;
- Time stopping approach: **no bounded delays** when the feasibility condition is not verified i.e., utilization rate $> 66\%$;
- Backlog-based approach: bounded delays under a full utilization rate, but **too pessimistic**;

Limitations: Scalability to large Networks

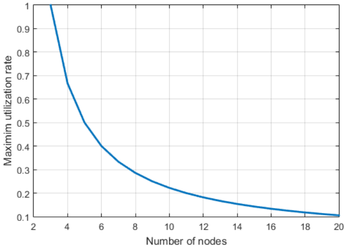

Figure : The maximum utilization rate with Time Stopping Method

⇒ Utilization rate decreases dramatically when the network size increases, e.g. $U < \frac{2}{M-1}$ **for general case**

Limitations: Scalability to large Networks

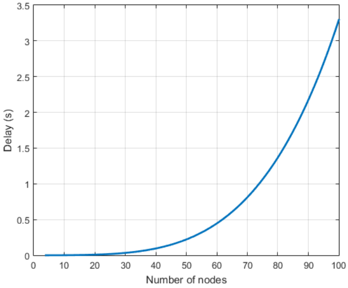

Figure : Upper bound delay with Backlog-Based Method

⇒ Delay bounds increase dramatically when the network size increases,e.g., greater than 100ms for a network of 30 nodes

PMOO & Non-FeedForward Networks

- PMOO [14] (Fidler 03) allows computing the end-to-end service curve of a flow of interest, accounting flow serialization phenomena.
  $\Rightarrow$ tighter upper bounds on end-to-end delays.
- PMOO has been applied for feedforward networks
  $\Rightarrow$ **Need to be extended to non-feedforward network**

PMOO & Non-FeedForward Networks: Assumptions

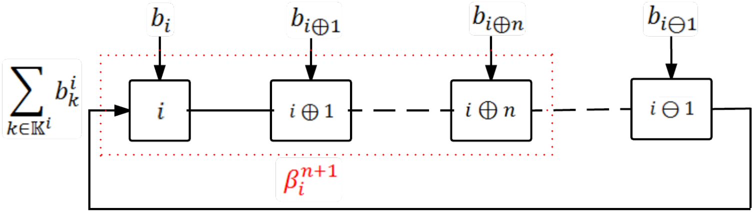

Figure : Direct and Indirect Interferences

Non-feedforward networks with disjoint cycle dependencies, i.e., any flow route form at most one cycle

PMOO & Non-FeedForward Networks: Service Curve

Let $\beta_{i,p}^{n}$ be the end-to-end service curve offered to a flow i with priority p on a sub-path of length n starting in node i:

$$\beta_{i,p}^{n}(t) = \min_{j \in \mathbb{J}_{i,p}^{n}} \left[R_{j,p} - \sum_{k \in \mathbb{K}_{p}^{j}} r_{k,p} \right] \times$$

$$\left(t - \sum_{j \in \mathbb{J}_{i,p}^{n}} T_{j,p} - \overbrace{\sum_{k \in \mathbb{J}_{i,p}^{n}, k \neq i} \frac{b_{k,p}}{\min_{j \in \mathbb{J}_{i \cap k,p}^{n}} [R_{j,p}]}}^{\text{Direct interference}} - \underbrace{\sum_{k \in \mathbb{K}_{p}^{i}} \frac{b_{k,p}^{i}}{\min_{j \in \mathbb{J}_{i \cap k,p}^{n}} [R_{j,p}]}}_{\text{Indirect interference}}\right)$$

where $j_{first} = first\{j \in \mathbb{J}_{i \cap k,p}^{n}\}$ is the first shared link in the considered sub-path by the flows i and k.

Enhanced Analyses of Non-FeedForward Networks

- The indirect interfering flows are unknown due to the cycle issue.
  - **Basic Analysis**: breaking dependencies when considering an **upper bound** on the indirect interfering flows.
  - **Tight Analysis**: Computing a **tight bound** on the indirect interfering flows to compute delay bounds.

Basic Analysis

- The maximum backlog bound $Backlog_p$ is a maximum bound on the total amount of data present in the network at any time.
  $\Rightarrow$ **Maximum bound on the upstream interfering flows**
- **++ This method has a linear complexity ($\mathcal{O}(1)$)**
- **- - Pessimistic end-to-end delay bounds**

Tight Analysis: Latency Formula

When considering the direct and indirect interference effects:

$$
\begin{aligned}
T_{i,p}^{n} &= \underbrace{\sum_{j=1}^{n} T_{i\oplus(j-1),p} + \sum_{k=1}^{n-1} \frac{b_{(i\oplus k),p}}{\min_{j\in \mathbb{J}_{i\cap i\oplus k,p}}[R_{j,p}]}}_{\text{Constant}} \\
&+ \underbrace{\sum_{k=1}^{M-1} \frac{b^{i}_{(i\oplus k),p}}{\min_{j\in \mathbb{J}_{i\cap i\oplus k,p}}[R_{j,p}]} \cdot \mathbb{1}_{\{(i\oplus k)\in \mathbb{K}_p^i\}}}_{\text{Indirect interference}} \\
&= cst1_{i,p}^{n} + \sum_{k=1}^{M-1} \frac{b^{i}_{(i\oplus k),p}}{\min_{j\in \mathbb{J}^{n}_{i\cap i\oplus k,p}}[R_{j,p}]} \cdot \mathbb{1}_{\{(i\oplus k)\in \mathbb{K}_p^i\}}
\end{aligned}
$$

Tight Analysis: Arrival Curves

The arrival curve of the traffic class p sent by the node j, received at node i is:

$$\begin{aligned} \alpha^i_{j,p}(t) &= \alpha_{j,p} \oslash \beta^{i\ominus j}_{j,p}(t) \\ \Longrightarrow \quad b^i_{j,p} &= b_{j,p} + r_{j,p} \times T^{i\ominus j}_{j,p} \\ &= cst2_{j,p} + r_{j,p} \times T^{i\ominus j}_{j,p} \end{aligned}$$

Tight Analysis: Matrix System

- The interdependency between the latency and the burst is due to the cycle issue.
  $\Rightarrow$ Defining a matrix system:

$$\begin{cases} T_p = C_1 + A_1 \times b_p \\ b_p = C_2 + A_2 \times T_p \end{cases}$$

  where,
  - A_1 holds all the coefficients of the unknown bursts and C_1 the constants of latency formula;
  - A_2 holds all the coefficients of the unknown latencies and C_2 the constants of bursts formula.

Tight Analysis: Feasibility Condition

$$T_p = (Id - A_1 \times A_2)^{-1} \times C_3$$

where $C_3 = C_1 + A1 \times C_2$

- The system admits a solution if the matrix $(Id - A_1 \times A_2)$ is invertible.
- If this condition is verified, then we can compute the vectors T_p and b_p.

$\Rightarrow$ To find the residual service for a priority p, all the vectors b_{pp}, for $pp < p$, need to be computed

$\Rightarrow$ **An iterative computation algorithm [13]**

Tight Analysis: Feasibility Condition

The determinant of the matrix $(Id - A_1 \times A_2)$ is a polynomial function of the variable x with a degree M:

$$(1 - M) \times (x + 1)^{(M-1)} \times (x - \frac{1}{M - 1})$$

x: utilization rate per node

- This matrix system resolution is feasible for $x \leq \frac{1}{M-1}$
  $\Rightarrow$ Feasibility condition under a full network utilization, i.e., $U \leq \frac{M}{M-1}$
  $\Rightarrow$ **Enhancing the resource-efficiency of the system, compared to conventional analytical approaches .**

Case of Study: Assumptions

- The links speed is $C = 1Gbit/s$;
- All equipments are similar and send the same traffic in broadcast mode;
- Technological latency within each node is $600ns$;
- Each equipment generates 3 types of traffic classes (TC)

Case of Study: Traffic Characteristics

	TC	Payload (byte)	rate (Kbps)
I/O data	HRT	64	80
Audio streaming	SRT	128	128
File transfer	NRT	1024	1000

Table : Traffic Characteristics

Case of Study: Scenarios

To conduct the performance analysis:

- **Scenario 1**: variation of the node number: from 10 to 100 nodes by a step of 10 nodes.
- **Scenario 2**: variation of the network utilization: the number of nodes is fixed, $M = 10$, and the network load is increasing by a step of 10% until reaching 100%.
- **Scenario 3**: variation of the burst size of the NRT traffic: from 64 bytes until 1500 bytes for a network of 35 nodes.

Sensitivity Analysis: Impact of network size (Sc.1)

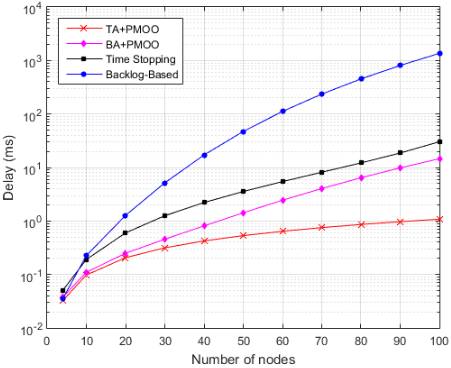

Figure : Upper bounds on the end-to-end delays of HRT

Sensitivity Analysis: Impact of network utilization (Sc.2)

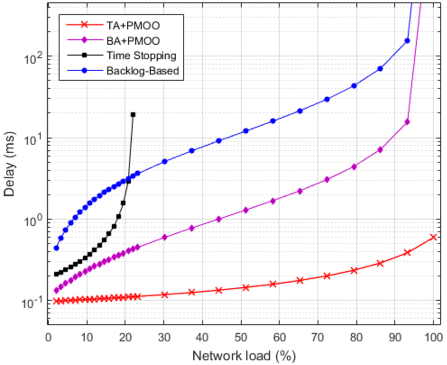

Figure : Upper bounds on the end-to-end delay bounds of HRT

Sensitivity Analysis: Impact of burst size (Sc.3)

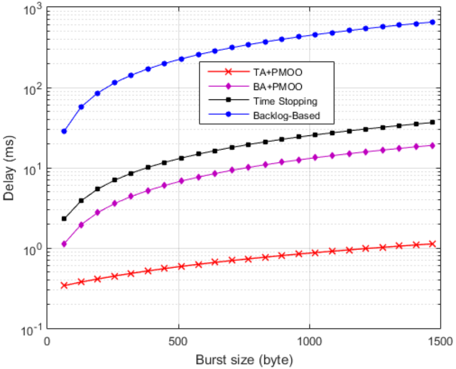

Figure : Upper bounds on the end-to-end latencies of HRT

Key Findings

- **Enhanced end-to-end delay bounds tightness with TA+PMOO**, compared to the basic approach (BA+PMOO) and the conventional ones.
- **Impact of Congestion**: Under TA+PMOO, tight delay bounds under high congestion, i.e. full utilization rate
- **Scalability to Large Networks**: Under TA+PMOO, improved system scalability, i.e., tight delay bounds are guaranteed for large-scale networks

Conclusions and Perspectives

- Enhanced analysis, based on **PMOO principle**, for (some) Non-feedforward Networks
- Improving system performance in terms of **delay bounds tightness** and the **system scalability**, in contrast to existing solutions.
- **Next step**: Extending the proposed approach to any non-feedforward network topology.

References

[0] K. Tindell and J. Clark. Holistic schedulability analysis for distributed hard real-time systems. Microprocessing and microprogramming, 40(2-3):117134, 1994.

[1] R. Wilhelm, et al. The worst-case execution-time problem: overview of methods and survey of tools. ACM Transactions on Embedded Computing Systems (TECS), 7(3):36, 2008.

[2] P. Puschner and A. Burns. Guest editorial: A review of worst-case execution-time analysis. Real-Time Systems, 18(2):115128, 2000.

[3] M. D. Schroeder, A. D. Birrell, and et al. Autonet: A high-speed, self-configuring local area network using point-to-point links. IEEE J. Sel. Areas Commun., 1991.

[4] D. Starobinski, M. Karpovsky, and L. Zakrevski. Application of network calculus to general topologies using turn-prohibition. IEEE/ACM Trans. Netw., 2003.

[5] J. C. Palencia and M. G. Harbour. Offset-based response time analysis of distributed systems scheduled under edf. In ECRTS, 2003.

References

[6] R. Pellizzoni and G. Lipari. Improved schedulability analysis of real-time transactions with earliest deadline scheduling. In RTAS, 2005.
[7] R. L. Cruz. A calculus of delay Part II: Network analysis. IEEE Trans. Inform. Theory, 1991.
[8] A. Charny and J.-Y. Le Boudec. Delay bounds in a network with aggregate scheduling. In Quality of Future Internet Services. Springer, 2000.
[9] B. Jonsson, S. Perathoner, L. Thiele, and W. Yi. Cyclic dependencies in modular performance analysis. In Proceedings of the 8th ACM international conference on Embedded software, 2008.
[10] L. Tassiulas and L. Georgiadis. Any work-conserving policy stabilizes the ring with spatial re-use. IEEE/ACM Trans. Netw., 1996.
[11] J.-Y. Le Boudec and P. Thiran. Network calculus: a theory of deterministic queuing systems for the internet. Springer Science Business Media, 2001.
[12] P. Jayachandran and T. Abdelzaher. End-to-end delay analysis of distributed systems with cycles in the task graph. In ECRTS, 2009.

References

[13] A. Amari and A. Mifdaoui et al. Enhanced WCTT Analysis of Distributed Networked Systems with Cyclic Dependencies. Technical Report 2016 (under review ECRTS 2016)

[14] M. Fidler. Extending the network calculus pay bursts only once principle to aggregate scheduling. In Quality of Service in Multiservice IP Networks. Springer, 2003.

Thank you for you attention

Questions?