DelayLyzer: A Tool for Analyzing Delay Bounds in Industrial Ethernet Networks

Mark Schmidt[1], Sebastian Veith[1], Michael Menth[1], and Stephan Kehrer[2]

[1] University of Tuebingen, Chair of Communication Networks, Tuebingen, Germany
[2] Hirschmann Automation and Control GmbH, Neckartenzlingen, Germany

Abstract. In this paper, we present the DelayLyzer which is a tool for the calculation of delay bounds using network calculus. It respects the specifics of Industrial Ethernet by implementing not only STP and RSTP but also MRP as forwarding mechanism. The tool allows to specify failure scenarios and alternate forwarding protocols for which delay bounds can also be computed.

1 Introduction

Industrial Ethernet is used for factory automation and other mission critical applications. It excels through the robustness and durability of its devices, as well as the implementation of special protocols, e.g. for failure protection. While conventional Ethernet implements the Spanning Tree Protocol (STP) or Rapid STP (RSTP) as forwarding and rerouting mechanism [6], Industrial Ethernet frequently relies on ring structures and the Media Redundancy Protocol (MRP) to protect failures [5]. Since Industrial Ethernet is often deployed for real-time applications, it is important to show a priori that maximum delay bounds will not be exceeded when the network is fed with a certain traffic pattern.

This paper describes the DelayLyzer, which is a tool for the calculation of delay bounds in Industrial Ethernet networks based on network calculus. It supports forwarding mechanisms that are specific to Industrial Ethernet like MRP. In addition, the tool allows to specify failure scenarios, calculates backup paths according to the implemented forwarding and protection protocols, and recomputes the delay bounds for these failure cases. Likewise, alternative forwarding protocols can be easily investigated. Thus, the DelayLyzer supports planning of Industrial Ethernet networks for challenging conditions.

Section 2 gives a brief overview of network calculus which is the theoretical base of the tool. Section 3 describes the graphical user interface of the tool, the internal data and its visualization, how use cases may be analyzed, and how results are displayed.

This work was funded by Hirschmann Automation and Control GmbH. The authors alone are responsible for the content of the paper.

©Springer, 17th GI/ITG MMB Conference with Dependability and Fault Tolerance, Bamberg, Germany, March 2014

2 Network Calculus

We first explain basics about how network calculus computes performance metrics for a single link, and then we clarify how this method can be applied to an entire network.

Network Calculus Applied to a Single Link Network calculus is a mathematical framework developed by Cruz [3,4], Chang [2] and Le Boudec [7] to determine delay bounds in networks. Network calculus models the maximum amount of data delivered by a flow to a link or a node by a rate-burst curve. This is essentially a token-bucket limited description of a flow's traffic. The general case of such an arrival curve is a piece-wise linear function of time. In a similar way, the delay added by the processing in links and nodes can be expressed by a rate-latency curve, the so-called service curve. Network calculus provides operations on curves that eventually facilitate the computation of maximum delay and backlog on a link. In addition, an output bound can be calculated to characterize the timely behavior of the flow after having passed the considered network element. The approach is scientifically backed by an algebraic theory.

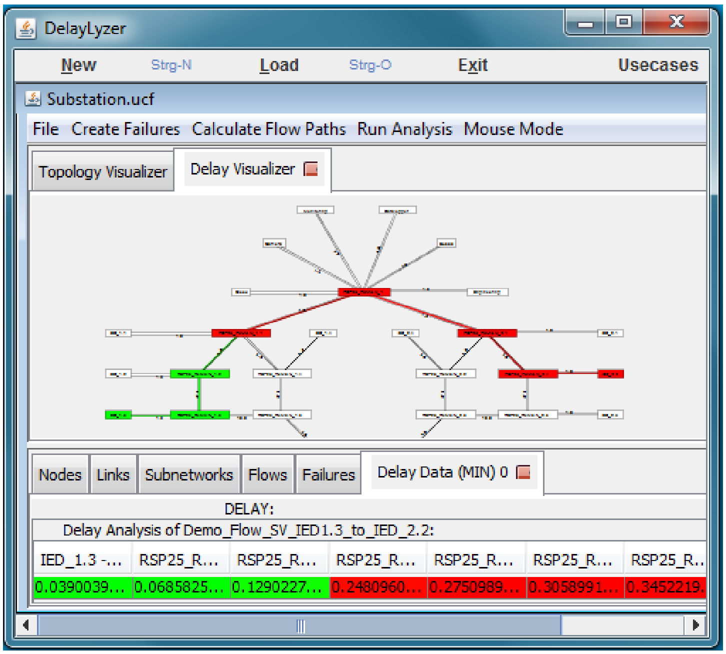
Network Calculus Applied to a Network To extend network calculus from a single link to a network, essentially the output bound of a flow from a predecessor link is taken as arrival curve for the next link. Schmitt and Zdarsky developed algorithms to calculate the maximum delay for a flow when traversing a series of nodes and links in a feed-forward network [10,12]. They proposed two simple variants: Total Flow Analysis (TFA) and Separated Flow Analysis (SFA). In addition, other authors proposed more complex algorithms leading to tighter delay bounds [1,8,9,11].

Forwarding and protection mechanisms in Ethernet networks are simple and lead to feed-forward networks so that network calculus can be used for the computation of delay bounds in that context. We developed variants of TFA and SFA as well as a combination thereof (mTFA) and implemented them in our tool.

3 Tool Description

The DelayLyzer calculates upper bounds for flow delays in specified use cases. To that end, a description of the network, of the flows, and of the path layout of the flows are required. The path layout is determined by the network, the set of failed network elements, and the applied forwarding protocols. In the following we explain the graphical user interface of the DelayLyzer, the components of a use case, how failures may be specified and how the path layout of flows are computed, how a delay analysis can be performed, and how results are provided.

Graphical User Interface Figure 1 illustrates a screenshot of the DelayLyzer for the use case "Substation". The menu bar of the main window allows to load existing use cases from file (Load), to create new use cases from scratch (New), and to exit the application (Exit). Several use cases may be open simultaneously which can be chosen from the drop-down menu "Use Cases". Each use case has its own internal window. The drop-down menu "File" in the menu bar of the use case window allows storing and closing a use case.



DELAY:

Delay Analysis of Demo_Flow_SV_IED1.3_to_IED_2.2:

IED_1.3 -...	RSP25_R...	RSP25_R...	RSP25_R...	RSP25_R...	RSP25_R...	RSP25_R..
0.0390039...	0.0685825...	0.1290227...	0.2480960...	0.2750989...	0.3058991...	0.3452219.

Fig. 1: Screenshot of the DelayLyzer.

Specification of Use Cases A use case comprises a network consisting of nodes, links, subnetworks, and flows. Details about them are available in the tabs in the lower part of a use case window. They indicate, e.g., performance and configuration data of links and nodes, the set of nodes and links belonging to subnetworks together with the applied forwarding protocols, and the source and destination node for each flow as well as its maximum burst size and rate.

The tool provides features to specify use cases and to store them in the "use case format" (ucf). It is also able to read the data from ucf-files and from the format of Hirschmann's HiVision tool. Although HiVision data contain only a subset of the ucf information, they are useful to import basic data of existing use cases.

The upper part of the use case window holds the "Topology Visualizer" tab which displays the network topology based on the use case information. When nodes, links, and flows are selected in the respective tabs, they are highlighted in the "Topology Visualizer". The path of a flow can be shown only after the calculation of its layout.

Specification of Failures and Computation of Path Layout of Flows The tab "Failures" provides a choice of failure scenarios and allows selection. The choice contains by default only the failure-free scenario but may be enlarged by the drop-down menu "Create Failures". The "Topology Visualizer" tab shows the topology for the selected failure scenario and also the delay analysis will performed only for the selected failure scenario.

The delay analysis requires the path layout of all flows. The drop-down menu "Compute Path Layout" allows to compute this path layout, which is done based on the use case information. Other entries in the drop-down menu "Compute Path layout" support changing the forwarding protocols in subnetworks. After a successful change, the path layout needs to be recomputed.

Delay Analysis and Results The drop-down menu "Run Analysis" provides a choice of different algorithms (TFA, SFA, mTFA) for delay analysis which may be performed upon selection. As soon as an analysis is completed, a "Delay Visualizer" tab and a "Delay Data" tab appear in the upper and lower part of the use case window. The "Delay Data" tab indicates the maximum delay for each flow after each hop as well as its maximum overall delay. A flow in the "Delay Data" tab may be selected; then the "Delay Visualizer" highlights the path of that flow within the topology. Each flow is associated with an allowed delay budget. Appropriate colors from green to red indicate in the "Delay Data" and the "Delay Visualizer" how much of that budget is already spent from the source of a flow up to each intermediate hop and its destination. This feature facilitates the interpretation of the numerical results.

Investigations have shown that the upper bound for the delay of a flow, that is not affected by a failure, may be longer or shorter in failure scenarios compared to the failure-free scenario. It is longer if additional flows share the paths of that flow, but it may be shorter if fewer flows share its path since some network elements have failed. Likewise, flows redirected to other paths may have shorter or longer delay bounds.

References

1. Bouillard, A., Jouhet, L., Thierry, E.: Tight Performance Bounds in the Worst-Case Analysis of Feed-Forward Networks. In: IEEE Infocom (Mar 2010)
2. Chang, C.S.: Performance Guarantees in Communications Networks. Springer
3. Cruz, R.L.: A Calculus for Network Delay, Part I: Network Elements in Isolation. IEEE Transactions on Information Theory 37(1), 114–131 (Jan 1991)
4. Cruz, R.L.: A Calculus for Network Delay, Part II: Network Analysis. IEEE Transactions on Information Theory 37(1), 132–141 (Jan 1991)
5. IEC: IEC 62439-2: Industrial communication networks - High availability automation networks - Part 2: Media Redundancy Protocol (MRP) (2010)
6. IEEE: IEEE 802.1D: IEEE Standard for Local and metropolitan area networks: Media Access Control (MAC) Bridges (2004)
7. Le Boudec, J.Y., Thiran, P.: Network Calculus. Springer (2004)
8. Lenzini, L., Martorini, L., Stea, G.: Tight End-to-End per-Flow Delay Bounds in FIFO Multiplexing Sink-Tree Networks. Performance Evaluation 63(9) (Oct 2006)
9. Lenzini, L., Mingozzi, E., Stea, G.: A Methodology for Computing End-to-End Delay Bounds in FIFO-Multiplexing Tandems. Performance Evaluation 65(11-12)
10. Schmitt, J.B., Zdarsky, F.A.: The DISCO Network Calculator – A Toolbox for Worst Case Analysis. In: VALUETOOLS (2006)
11. Schmitt, J.B., Zdarsky, F.A., Fidler, M.: Delay Bounds under Arbitrary Multiplexing: When Network Calculus Leaves You in the Lurch... In: IEEE Infocom
12. Schmitt, J.B., Zdarsky, F.A., Martinovic, I.: Performance Bounds in Feed-Forward Networks under Blind Multiplexing. Tech. rep., University of Kaiserslautern