Routing Algorithms for IEEE802.1Qbv Networks

Naresh Ganesh Nayak, Frank Dürr, Kurt Rothermel
Institute for Distributed and Parallel Systems
Stuttgart, Germany
{nayaknh,duerrfk,rothermel}@ipvs.uni-stuttgart.de

ABSTRACT

The recently published *IEEE 802.1Qbv* standard specifies enhancements for providing real-time communication guarantees for time-triggered flows while also handling best-effort traffic in a converged Ethernet network. The enhancements include a programmable time-based gating mechanism for stipulating which of the queues of an egress port are available for transmission at any given point of time. By appropriately programming (opening and closing) these gates, the traversal of packets through the network can be controlled to precisely follow a precomputed schedule that satisfies the timing constraints of the time-triggered flows. Computing such transmission schedules requires routing of the flows in the first step, followed by the computation of gate schedules for the flows along their respective routes. So far off-the-shelf algorithms like shortest path routing, which optimize the number of hops over which flows are routed, have been used for computing routes for the time-triggered traffic. In this paper, we explore how the routing of time-triggered flows affects their schedulability. Moreover, we identify additional parameters that must be considered while routing time-triggered traffic and propose ILP-based algorithms for the purpose. Our evaluations show that the proposed routing algorithms could improve the slack in the computed schedules by upto 60 % and 30 % compared to shortest path routing and equal cost multi-pathing (ECMP), respectively, and, thus, increase the capacity of the network to accommodate more time-triggered traffic.

CCS CONCEPTS

•**Networks** → *Traffic engineering algorithms; Network design and planning algorithms;*

KEYWORDS

Routing, IEEE 802.1Qbv, Time-sensitive Networks, Integer Linear Programming

ACM Reference format:
Naresh Ganesh Nayak, Frank Dürr, Kurt Rothermel. 2017. Routing Algorithms for IEEE802.1Qbv Networks. In *Proceedings of Real-time Networks, Dubrovnik, Croatia, 27th June 2017 (RTN '17),* 6 pages.

1 INTRODUCTION

The IEEE Time-sensitive Networking (TSN) Task Group has recently standardized enhancements (referred to as *IEEE 802.1Qbv*) for handling scheduled traffic in Ethernet networks as a part of Time-sensitive Networking [4]. This standard targets applications requiring deterministic network latency and jitter for their time-triggered (periodic) communication flows, for instance, to support the networked control systems used in industrial automation. Instead of using dedicated field-bus networks for providing real-time guarantees for such time-sensitive traffic, the incorporated enhancements facilitate a converged Ethernet network which transports time-triggered traffic along with best-effort traffic. The enhancements primarily specify a programmable time-based gating mechanism for the Ethernet switches that stipulates which of the queues of the egress ports are available for transmission. By using this mechanism in combination with clock synchronization protocols, like the Precision Time Protocol (PTP) or the *IEEE 802.1AS*, transmission of packets through each node in the network can be controlled to precisely follow a precomputed schedule which satisfies the timing constraints of the time-triggered flows.

While the standards specify the behaviour of these enhancements, the resulting scheduling problem in these networks is still open for research. Few initial solutions have been proposed for computing the transmission schedules for *IEEE 802.1Qbv* networks [7][8], while a few can be adapted from similar scheduling problems in other field-bus networks like ProfiNET [9]. All these approaches separate the scheduling of time-triggered flows from its routing owing to the high time-complexity of the scheduling problem, usually NP-hard. Thus, these scheduling approaches require as input the specifications of the time-triggered flows (source, destination, transmission period, deadlines, etc.) along with their routes. Usually, time-triggered flows are routed using the same off-the-shelf algorithms that are also used for routing best-effort traffic, like shortest path routing, which minimizes the number of hops over which the traffic is routed. The transmission schedules are then computed by modelling the scheduling problem using popular techniques like Satisfiability Modulo Theories (SMT) or by mapping the scheduling problem to other known problems like the Resource Constrained Project Scheduling (RCPS), No-wait Job-shop Scheduling Problem (NW-JSP) etc. The existing body of work has, however, not explored the impact of the used routing algorithm on the schedulability of the set of time-triggered flows. This has now gained greater importance with the possibility to achieve explicit control over routing by means of software-defined networking protocols like OpenFlow or the amendments mentioned in the *IEEE 802.1Qca* standards [5][12].

In our previous work, we developed an approach to compute transmission schedules for networks compliant with the *IEEE 802.1Qbv* standard by mapping the scheduling problem to No-wait Job-shop Scheduling Problem (NW-JSP), a well-known problem from operations management [8]. In this paper, we evaluate the impact of different routing algorithms on the schedulability of the flows expressed using the slack in the computed schedules. We also present potential heuristics that may be specifically used for routing time-triggered flows in an *IEEE 802.1Qbv* network for yielding

Copyright retained by the owner/author(s).

schedules with an increased slack, and, thus, increase the capacity of the network to accommodate more time-triggered traffic.

In particular, our contributions in this paper are as follows:

(1) We show that the routing algorithm for time-triggered flows impacts the schedulability of the computed transmission schedules in *IEEE 802.1Qbv* networks.
(2) We identify the parameters of a routing algorithm that have an impact on the slack of the subsequently computed schedules. Moreover, we use these parameters to propose ILP-based algorithms for routing time-triggered flows, and, thus, improve their schedulability.
(3) Finally, we show with empirical evaluations that routing using the proposed algorithms improve the slack in the schedules compared to the benchmark routing algorithms, shortest path routing and equal cost multipathing (ECMP), by up to 60 % and 30 %, respectively.

The remaining paper is structured as follows. We provide a brief overview of the computation of transmission schedules using No-wait Job-shop Scheduling Problem and show the impact of the used algorithm for routing time-triggered flows on its schedulability in Section 2. In Section 3, we identify the parameters of a routing algorithm that affect the slack in the computed schedules, based on which we propose specialized ILP-based algorithms for routing time-triggered flows. We present the evaluations of our algorithms and the related work in Section 4 and 5, respectively. Finally, we conclude in Section 6.

2 SCHEDULING & ROUTING IN TIME-SENSITIVE NETWORKS

In this section, we present a brief overview of our scheduling approach using No-wait Job-shop Scheduling Problem (NW-JSP) and introduce the problem of routing time-triggered flows in *IEEE 802.1Qbv* networks.

2.1 Scheduling in IEEE 802.1Qbv Networks

2.1.1 System Model. Our system model consists of a converged Ethernet network comprising switches compliant with the *IEEE 802.1Qbv* standard, hosts (end-systems), and a centralized network controller. The hosts are basically the sources and sinks of the traffic, while the network controller is responsible for computing the routes and schedules for the traffic and programming the underlying switches accordingly. We assume that the switches provide programmatic interfaces (e.g. OpenFlow [12] or mechanisms from IEEE 802.1Qca [5]) to set arbitrary routes for the traffic in addition to the interfaces for programming the gating mechanism.

Furthermore, we assume that all the nodes and the switches in the network are precisely synchronized and that the hosts can adhere with the computed schedules reasonably.

2.1.2 IEEE 802.1Qbv Specification. The *IEEE 802.1Qbv* standard introduces a gating mechanism (cf. Figure 1) that controls which of the queues of the egress port are considered for transmission selection. This gating mechanism is to be programmed with a sequence of gate events, each consisting of a relative time-stamp (represented as T_i) to the previous event in the sequence and a bit-mask indicating the queues which are to be considered for transmission selection

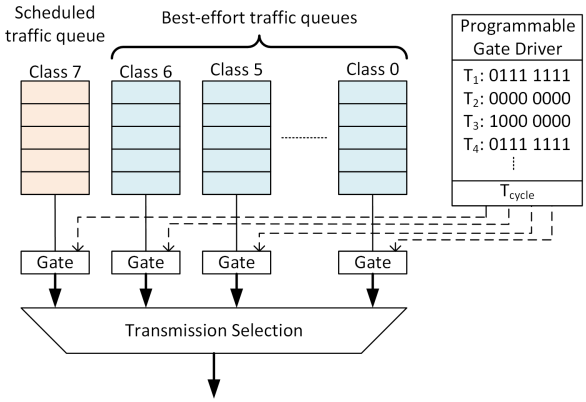


Figure 1: Egress port with IEEE 802.1Qbv enhancements [10].

till the next event. By continually repeating this sequence after a pre-programmed duration, T_{cycle}, the mechanism achieves a cyclic schedule of length T_{cycle} for the gate events. Furthermore, using clock synchronization protocols to synchronize the clocks of all switches in the networks, the cyclic patterns of the gate events of the ports for all switches can be precisely aligned.

In an *IEEE 802.1Qbv* network, one or more queues per egress port are exclusively reserved for time-sensitive traffic. Packets belonging to time-triggered flows traverse through these queues following a pre-computed global transmission schedule, thus, resulting in deterministic and bounded network latency and jitter. The gating mechanism ensures, by appropriately opening and closing the gates on different egress ports, that all time-sensitive packets in the network precisely adhere with the global transmission schedule. While the standard itself does not specify any algorithms for computation of such transmission schedules, developing a customized schedule to satisfy the real-time constraints of the constituting time-triggered flows is the most critical aspect of an *IEEE 802.1Qbv* network. It is based on this computed transmission schedule that the sequence of gate entries for all the egress ports are derived.

2.1.3 Mapping to the No-wait Job-shop Scheduling Problem. The computation of transmission schedules in time-triggered networks is equivalent to the bin-packing problem [15]. Most approaches model the problem as a constrained optimization problem for optimizing network utilization [7][9]. Owing to the high time-complexity (NP-hard) of the scheduling problem, it is hard to jointly optimize the routes and the schedules for the flows in these networks. Hence, scheduling approaches first determine the routes of the time-triggered flows using off-the-shelf routing algorithms, and then, based on these routes compute the transmission schedules.

In [8], we mapped the transmission scheduling problem in the *IEEE 802.1Qbv* networks as No-wait Job-shop Scheduling Problem (NW-JSP), a well-researched problem from operations management. The Job-shop Scheduling Problem deals with computation of a shop-floor schedule of shortest possible duration (makespan) for manufacturing jobs which are to be executed on the machines in the shop-floor. Here, a manufacturing job is defined as a sequence of operations, each of which must be executed on a particular

machine for a pre-determined time duration, while ensuring that no machine processes more than one job at any given time. To compute transmission schedules, we model the network interface controllers (NIC) of the hosts and the ports of the switches as machines in the shop-floor, the time-triggered flows as manufacturing jobs, and the route of the flows as the sequence of the operations for the corresponding jobs. The duration of a given operation equals the time required for the packets belonging to the corresponding flow to be forwarded by the corresponding switch port. Moreover, we include a no-wait constraint, i.e., once a job is started it must be processed to its completion. In terms of the transmission schedules, this implies that the packets belonging to time-triggered flows traverse through the network uninterruptedly without queuing till it reaches its destination(s). This guarantees least possible network delay for time-sensitive traffic, and satisfies any feasible deadline that the flows may have. We term the resulting problem from this modelling as the No-wait Packet Scheduling Problem (NW-PSP).

Similar to the NW-JSP, NW-PSP also deals with the computation of a transmission schedule with minimal flowspan (T_{fs}), where the flowspan of a schedule is the total time required in a scheduling cycle for handling all the time-triggered flows in the network. The flowspan of a transmission schedule is different from its length, T_{cycle}. Figure 2 shows the schedule of three time-triggered flows traversing over the same set of links and highlights the difference between the flowspan and the schedule length. In NW-PSP, T_{cycle} is fixed based on the smallest transmission period that must be supported for a time-triggered flow in the network, as the approach requires that the transmission periods of the flows in the network are an integral multiple of the schedule length, T_{cycle}. The flowspan minimization has the effect of bunching up time-triggered flows towards the start of the scheduling cycle leaving bandwidth for the best effort traffic at the end of the scheduling cycle. The flowspan of the schedule for a given set of time-triggered flows in a network is critical with respect to their schedulability. Here, schedulability implies that the computed schedule satisfies the timing constraints of all the time-triggered flows in the network. For this, it is necessary that the flowspan is less than or equal to the length of the schedule, i.e., $T_{fs} \leq T_{cycle}$. If the flowspan exceeds the length of the schedule, then the traffic from a given scheduling cycle may interfere with the traffic from the subsequent cycle leading to the violation of the timing guarantees. This, also, justifies the optimization goal of the NW-PSP. Moreover, the flowspan of the schedule reflects the capacity of the network to accommodate further flows, i.e., the slack in the schedule. The lower the flowspan, higher is the slack, and higher is the quantum of time-triggered traffic that can be further accommodated in the network. For further details on NW-PSP, we direct our readers to [8].

Other approaches, for instance, the ones using Satisfiability Modulo Theories (SMT), compute schedules based on the concept of "hypercycles". Here, the length of the schedule, T_{cycle}, is equal to the least common multiple of the transmission periods of all the time-triggered flows to be scheduled. These approaches do not constrain the transmission period of the flows in any way, and distribute the scheduled traffic throughout the length of the schedule in compliance with their timing constraints. Hence, the concept of flowspan is not meaningful for such scheduling approaches as they do not strive to tightly bunch the time-sensitive traffic at the beginning of a scheduling cycle. We, nonetheless, choose flowspan as a metric for comparing schedules for this paper, and leave the task of identifying a universal metric for schedule comparison as future work.

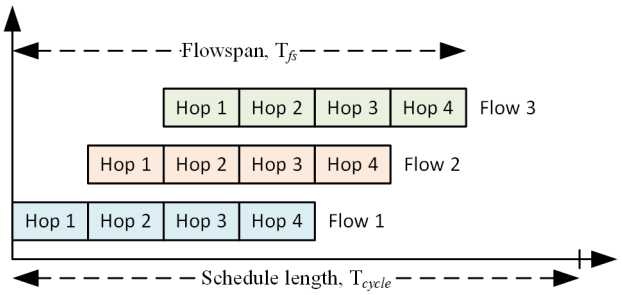


Figure 2: Flowspan of schedules computed using NW-PSP

2.2 Problem Statement - Routing time-triggered flows

As production processes dictate the sequence of operations composing the manufacturing jobs in NW-JSP, reordering or modifying the sequence is usually avoided even if it may lead to better schedules. This is, however, not the case in NW-PSP, where the constituent forwarding operations of the flows can be modified to a certain extent, especially, if it results in improved schedules. In other words, the routes of time-triggered flows may be modified to yield transmission schedules with a lower flowspan. So, we analyze the impact of routes serving as input parameter to the NW-PSP on the flowspan of the calculated schedules.

The standard algorithms (like shortest path first, equal cost multipathing etc.) optimize the number of hops over which the traffic is routed. In terms of the resulting NW-PSP, the flows to be scheduled consist of fewest forwarding operations possible. However, flows with fewer forwarding operations alone do not guarantee an optimal flowspan. In NW-PSP, the flowspan is also dependent on the number of flows with conflicting forwarding operations, i.e., the forwarding operations belonging to different flows that must be processed on the same egress ports of the switches, and the maximal duration that any of the switch ports is kept occupied. Hence, time-triggered flows must be routed to have fewer flows with overlapping paths, while also minimizing the aggregated amount of time-triggered traffic that any switch port should transmit. For this, we introduce a metric, Maximum Scheduled Traffic Load (MSTL), which can be used for routing time-triggered traffic. We define Maximum Scheduled Traffic Load (MSTL) as the maximum amount of scheduled traffic that is transmitted by any of the switch port in a network per cycle of the transmission schedule. *MSTL* is directly influenced by the routing algorithm, for instance, routing all time-triggered traffic over any single bottleneck link leads to a higher value of *MSTL*, compared to the case where this traffic is distributed throughout the network.

In our evaluations, we observed a direct relationship between the *MSTL* and the resulting flowspan of the schedule. Figure 3a shows the variance of *MSTL* (using shortest path routing (SP) and equal-cost multi-pathing (ECMP)), and the resulting schedule flowspan against a varying number of time-triggered flows (200–1000), each

sending packets of varying sizes (300–1500 bytes) per cycle, in an Erdős-Rényi (ER) topology with 10 switches and 50 hosts. The figure clearly shows similar behaviour for the schedule flowspan and the corresponding *MSTL*. Thus, we argue that to have schedules with lower flowspan, it is necessary that the preceding routing stage routes time-triggered flows accounting for the corresponding *MSTL*.

In the following, we present two ILP-based algorithms for routing time-triggered flows which explicitly minimize the resulting MSTL, in order to yield schedules with lower flowspan.

3 ILP BASED ROUTING ALGORITHMS

In this section, we present our system model and the ILP-based algorithms for determining routes for time-triggered flows.

3.1 Terminologies

We denote the network as a directed graph, $G \equiv (V, E)$, where V is the set of all nodes (hosts and switches) in the network, while $E \subseteq V \times V$ is the set of edges connecting a pair of nodes. Time-triggered flow is denoted as a tuple, $f \equiv (src_f, dst_f, size_f, period_f)$, which implies that the source host, src_f, sends time-sensitive packets with a total aggregated size of $size_f$ bytes to the host(s), dst_f, every $period_f$ time units.

The length of the schedule is denoted as T_{cycle}, and is based on the scheduling approach to be used.

3.2 Routing Heuristics

We mainly present two ILP-based routing approaches, the first optimizes the routes of the time-triggered traffic based on the resulting MSTL only, while the second one also considers the number of hops over which the flow is routed.

3.2.1 MSTL Based Routing. In this heuristic, we solely base the routing decision on the resulting *MSTL*.

The inputs for this ILP are:

(a) Network topology, G,
(b) Set of flows to be scheduled, $F \equiv \{f_1, f_2, \ldots f_n\}$.

The variables used for this ILP are:

(a) Route allocation, $Routes \equiv \{r_{i,j}\}\ \forall i \in F, j \in E$.
Here, $r_{i,j} = 1$, if flow i is routed over link j, else 0. The values of these variables, basically, determine the routes for the flows,
(b) Maximum scheduled traffic load, *MSTL*. This variable is used in the objective function. It must be noted that, we do not set a value for the *MSTL* upfront, rather allow the solver to route the flows while minimizing it,
(c) Destination counters, $DC \equiv \{d_{i,j}\},\ \forall i \in F, j \in E$.
Here, $dc_{i,j}$ = no. of destinations of flow i reachable over link j, if flow i is routed over link j, else 0. These are auxiliary variables for handling multicast time-triggered flows.

The objective of this ILP is, thus, to minimize *MSTL*, subject to:

(a) The route for each flow starts at its source and ends at its destination(s). i.e., the number of destinations reachable over the outgoing links of the source host is equal to the number of destinations of the flow, while the number of destinations reachable over the incoming links of the destination hosts is 1. For all the other nodes in graph G, the sum of destinations reachable over incoming links is equal to the sum of destinations reachable over outgoing links. The below constraints are applicable for all flows, i.e., $\forall i \in F$.

$$\sum_{j \in \text{in}(src_i)} dc_{i,j} = 0 \qquad \sum_{j \in \text{out}(src_i)} dc_{i,j} = |dst_i| \tag{1}$$

$$\sum_{j \in \text{in}(n)} dc_{i,j} = 1 \qquad \sum_{j \in \text{out}(n)} dc_{i,j} = 0 \qquad \forall n \in dst_i \tag{2}$$

$$\sum_{j \in \text{in}(n)} dc_{i,j} = \sum_{j \in \text{out}(n)} dc_{i,j} \qquad \forall n \in V \setminus (\{src_i\} \cup dst_i) \tag{3}$$

Here, the *in*() and *out*() functions return the incoming edges and outgoing edges of the node passed as parameter, respectively.

(b) The flows must be routed such that no switch port is transmitting more scheduled traffic than stipulated by *MSTL* (which is being minimized in the objective).

$$\sum_{\forall i \in F} r_{i,j} \cdot size_i \cdot \frac{T_{cycle}}{period_i} \leq MSTL \qquad \forall\, j \in E \tag{4}$$

Though NW-PSP restricts the periods of the flows to be integral multiples of T_{cycle}, in practice, the scheduling algorithm schedules all the flows assuming that their period is equal to T_{cycle}. Thus, for NW-PSP, this constraint simplifies to consider the size of the flow only. However, the constraint enables the usage of these approaches for the SMT-based approaches also.

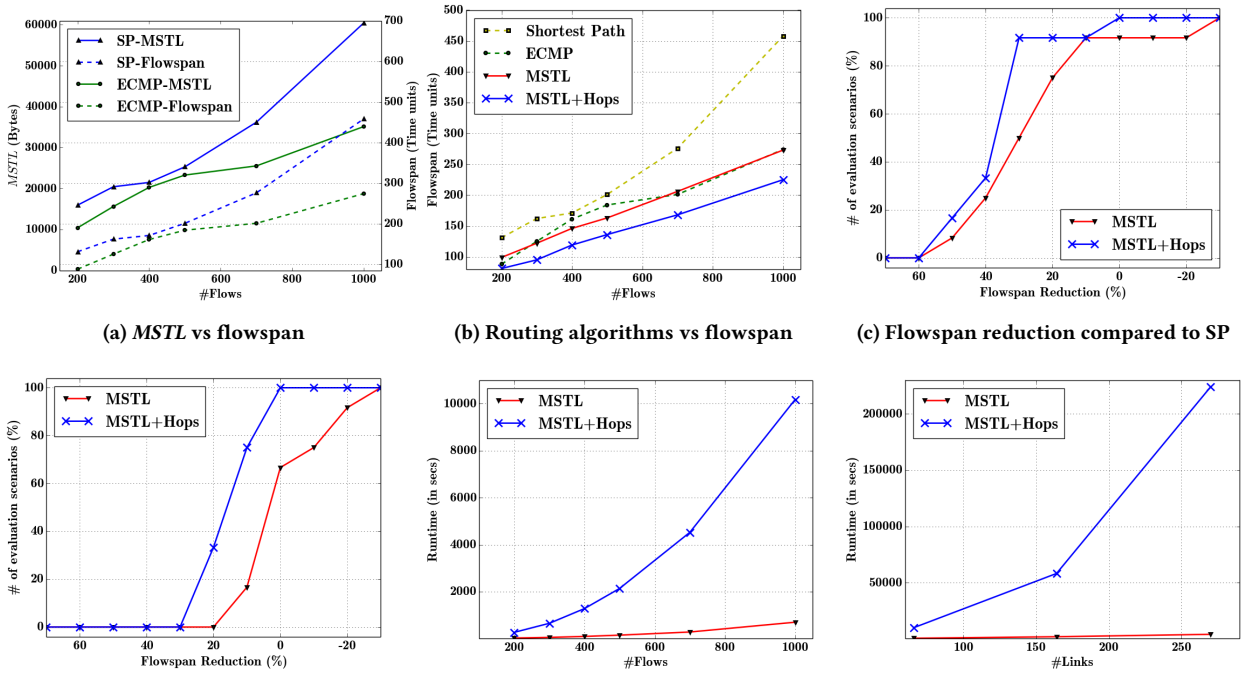
(c) As an auxiliary constraint, it is required that the routing variables are inline with the destination counter. These variables must be related as follows.

$$r_{i,j} \cdot |dst_i| \geq dc_{i,j} \qquad \forall\, i \in F, \forall\, j \in E \tag{5}$$

After solving this ILP, the routes for the time-triggered flows can be derived from the values of the ILP variable, *Routes*. It must be noted that the computed routes for the flows may have loops resulting from links that handle scheduled traffic much lower than the *MSTL*. These loops can be removed by means of post processing the routes or adding constraints to the objective function to constrain the solver from routing flows over paths with loops. However, our evaluations show that such modifications to the ILP lead to a significant increase in the execution runtimes. Hence, we chose to post process the ILP solution to remove any loops in the final routes. The post processing of the solution in no way alters the resulting *MSTL*, as it is already minimized by the solver.

3.2.2 MSTL+Hops Based Routing. Routing of time-triggered traffic minimizing the *MSTL* only, may result in some flows being routed over longer paths. In some cases, this may lead to an increase in the flowspan, as a few flows in the NW-PSP instance would now have an increased number of forwarding operations due to the longer routing of the flows. Hence, we now extend the aforementioned ILP to compute routes for time-triggered traffic, while optimizing the number of hops along with the resulting *MSTL*.

For this, we modify the objective of the aforementioned ILP as follows.



(a) *MSTL* vs flowspan

(b) Routing algorithms vs flowspan

(c) Flowspan reduction compared to SP

(d) Flowspan reduction compared to ECMP

(e) Runtime vs No. of Flows

(f) Runtime vs No. of Links

Figure 3: Evaluations Results for the ILP formulations presented in Section 3

Minimize:

$$\frac{MSTL}{1 + \sum\limits_{\forall i \in F} size_i} + \frac{\sum\limits_{\forall i \in F} \sum\limits_{\forall j \in E} r_{i,j}}{1 + (|F| \cdot |E|)}$$

The first term of the objective is, basically, minimizing the *MSTL*, while the second term minimizes the cumulative number of hops over which all the flows are routed. Both these terms are normalized, so that the ILP solver does not prioritize one over the other.

Our evaluations show that the presented algorithms do, indeed, reduce the flowspan for the schedules computed using NW-PSP. We leave the task of evaluating the impact of these algorithms on other scheduling approaches based on the concept of hyper-cycles (e.g. SMT based approaches) as future work.

4 EVALUATIONS

In this section, we present the results of our evaluations of the ILP-based routing algorithms with respect to their impact in improving the transmission schedules and their scalability.

4.1 Impact on Scheduling

To evaluate the impact of the routing algorithms on the subsequent transmission scheduling, we routed a varying number (200–1000) of time-triggered flows in a random network topology consisting of 50 hosts and 10 switches, generated using the Erdős-Rényi model, using four different routing schemes—Shortest path routing, Equal Cost Multi-Pathing (ECMP), *MSTL* based routing, and *MSTL*+Hops based routing. In the next step, we computed the transmission schedules for the flows using NW-PSP and the routes computed in the previous step with the different algorithms. The results of this evaluation are summarized in Figure 3b.

The results show that with an increase in the number of flows the schedule flowspan increases rapidly in the case of shortest path routing. This is because the shortest path routing is agnostic to the load of scheduled traffic while computing routes. ECMP fares much better with the increase in flowspan being gradual, as it tries to randomly distribute the load of scheduled traffic throughout the network. Further, *MSTL*-based routing typically yields better schedules compared to routing using ECMP, but as it may route flows over longer paths, occasionally the flowspan may be higher than that with ECMP. In all the cases, *MSTL*+Hops based routing outperforms all the other routing schemes, and yield schedules that have on an average 38 % and 20 % lower flowspan compared to the shortest path routing and ECMP, respectively.

We also executed the presented ILP-based routing algorithms for computing routes for time-triggered flows in 24 different scenarios (varying number of flows on three topologies of differing sizes). The subsequently computed schedules were then compared with the ones resulting from the shortest path routing and ECMP. Figure 3c and Figure 3d show the cumulative distributions of flowspan reduction compared to the shortest path routing and ECMP, respectively. The results show that while the *MSTL*-based approach, in general, improves quality of schedules, in a few cases, it ends up increasing the flowspan compared to the off-the-shelf algorithms. Overall, such cases were limited to less than 10 % and 30 % with

shortest path routing and ECMP as references, respectively. However, the *MSTL*+Hops based approach always yields an improvement in flowspans, with schedule flowspans reduced by up to 60 % and 30 % compared to shortest path routing and ECMP, respectively.

4.2 Scalability Evaluations

The runtimes for the ILP-based routing algorithms that we presented in this paper depend on the number of flows which are to be routed and the size of the topology (in terms of the number of links). To determine the scalability of the algorithms with respect to the number of flows, we executed the algorithms for computing routes for varying number of flows (200–1000) in an Erdős-Rényi network with 50 hosts and 10 switches. As shown in Figure 3e, the runtimes for the *MSTL*+Hops based algorithm increases steeply with the number of flows being scheduled, with approximately 3 hours of runtime for routing 1000 flows. In contrast, the runtimes for the MSTL-based approach increase gradually with the number of flows. The runtime for routing 1000 time-triggered flows using this approach is approximately 11 minutes.

We also evaluated the scalability of the algorithms with respect to the size of the topology. We routed 1000 flows on networks of different sizes. Figure 3f summarizes the results of this evaluation. Similar to the earlier evaluations, the runtimes for *MSTL*+Hops based routing algorithm increase rapidly with the number of links. The runtimes increase from about 3 hours to 63 hours when the size of topology is increased from 66 links to 270 links. The *MSTL*-based routing can, however, route 1000 flows in a network topology with 270 links in approximately 1 hour.

5 RELATED WORK

The IEEE Time-sensitive Networking Task Group mainly divides the network traffic in three categories, viz., the scheduled or the time-triggered traffic, the shaped traffic, and best-effort traffic. The best effort traffic is routed usually based on the spanning tree of the network created using different variants of the spanning tree protocol in IEEE 802.1Q and IEEE 802.1D standards [1][2]. Further, the recently published IEEE 802.1Qca provides explicit control over routing and mechanisms to incorporate redundant multipaths for network resilience [5]. The shaped traffic has soft real-time constraints and is to be used for audio/video streams to be transported over the network. The routing of AVB streams to meet its constraints in the presence of intervening scheduled traffic is addressed in [11].

In our earlier work, we solved the scheduling and routing problem of time-triggered traffic jointly by means of Time-sensitive Software-defined Networks (TSSDN) [13]. However, TSSDN schedules the transmissions on the hosts (network edge) alone, as it relies on frame pre-emption [3] and priority queuing alone to isolate time-triggered traffic from the others. Additionally, there are other approaches to compute routes for the network traffic in datacenters to achieve low latencies [6][14].

However, these routing approaches cannot be directly applied on the time-triggered traffic, as they are agnostic to the schedulability of the set of flows. In contrast, we presented ILP-based algorithms which improve the chances of the set of the time-triggered flows to be successfully scheduled.

6 CONCLUSION

Computing appropriate transmission schedules is an important aspect for providing real-time guarantees for the scheduled traffic in the *IEEE 802.1Qbv* networks. As a consequence of the high time-complexity, the scheduling problem is solved disjointly from the corresponding routing problem. In this paper, we discussed the impact that routing of time-triggered traffic has on the quality of the schedules computed, and the need for specialized algorithms for routing such traffic. We identified parameters, in addition to the number of hops, which must be considered by routing algorithms while computing routes for time-triggered flows. We also proposed two ILP-based routing algorithms based on our findings for this purpose. Our evaluations show that specialized routing algorithms can improve the quality of schedules substantially. In our future work, we would develop heuristics in order to reduce the runtimes of our algorithms, and improve its scalability.

REFERENCES

[1] 2004. IEEE Standard for Local and metropolitan area networks: Media Access Control (MAC) Bridges. *IEEE Std 802.1D-2004 (Revision of IEEE Std 802.1D-1998)* (2004), 1–277.

[2] 2014. IEEE Standard for Local and metropolitan area networks–Bridges and Bridged Networks. *IEEE Std 802.1Q-2014 (Revision of IEEE Std 802.1Q-2011)* (2014), 1–1832.

[3] 2016. IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks – Amendment 26: Frame Preemption. *IEEE Std 802.1Qbu-2016 (Amendment to IEEE Std 802.1Q-2014)* (2016), 1–52.

[4] 2016. IEEE Standard for Local and metropolitan area networks – Bridges and Bridged Networks - Amendment 25: Enhancements for Scheduled Traffic. *IEEE Std 802.1Qbv-2015* (2016), 1–57.

[5] 2016. IEEE Standard for Local and metropolitan area networks– Bridges and Bridged Networks - Amendment 24: Path Control and Reservation. *IEEE Std 802.1Qca-2015 (Amendment to IEEE Std 802.1Q— as amended by IEEE Std 802.1Qcd-2015 and IEEE Std 802.1Q—/Cor 1-2015)* (2016), 1–120.

[6] Mohammad Al-Fares, Sivasankar Radhakrishnan, Barath Raghavan, Nelson Huang, and Amin Vahdat. 2010. Hedera: Dynamic Flow Scheduling for Data Center Networks. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation (NSDI'10)*. 19–19.

[7] Silviu S. Craciunas, Ramon Serna Oliver, Martin Chmelík, and Wilfried Steiner. 2016. Scheduling Real-Time Communication in IEEE 802.1Qbv Time Sensitive Networks. In *Proceedings of the 24th International Conference on Real-Time Networks and Systems (RTNS '16)*. 183–192.

[8] Frank Dürr and Naresh Ganesh Nayak. 2016. No-wait Packet Scheduling for IEEE Time-sensitive Networks (TSN). In *Proceedings of the 24th International Conference on Real-Time Networks and Systems (RTNS '16)*. 203–212.

[9] Z. Hanzalek, P. Burget, and P. Sucha. 2010. Profinet IO IRT Message Scheduling With Temporal Constraints. *IEEE Transactions on Industrial Informatics* 6, 3 (2010), 369–380.

[10] M.D. Johas Teener, A.N. Fredette, C. Boiger, P. Klein, C. Gunther, D. Olsen, and K. Stanton. 2013. Heterogeneous Networks for Audio and Video: Using IEEE 802.1 Audio Video Bridging. *Proc. of the IEEE* 101, 11 (Nov 2013), 2339–2354.

[11] Sune Mølgaard Laursen, Paul Pop, and Wilfried Steiner. 2016. Routing Optimization of AVB Streams in TSN Networks. *SIGBED Rev.* 13, 4 (Nov. 2016), 43–48.

[12] Nick McKeown, Tom Anderson, Hari Balakrishnan, Guru Parulkar, Larry Peterson, Jennifer Rexford, Scott Shenker, and Jonathan Turner. 2008. OpenFlow: Enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review* 38, 2 (2008), 69–74.

[13] Naresh Ganesh Nayak, Frank Dürr, and Kurt Rothermel. 2016. Time-sensitive Software-defined Network (TSSDN) for Real-time Applications. In *Proceedings of the 24th International Conference on Real-Time Networks and Systems (RTNS '16)*. 193–202.

[14] Jonathan Perry, Amy Ousterhout, Hari Balakrishnan, Devavrat Shah, and Hans Fugal. 2014. Fastpass: A centralized zero-queue datacenter network. In *ACM SIGCOMM Computer Communication Review*, Vol. 44. ACM, 307–318.

[15] Wilfried Steiner. 2010. An evaluation of SMT-based schedule synthesis for time-triggered multi-hop networks. In *IEEE 31st Real-Time Systems Symposium (RTSS), 2010*. IEEE, 375–384.