

An Extension of the OMNeT++ INET Framework for Simulating Real-time Ethernet with High Accuracy

Till Steinbach, Hermand Dieumo Kenfack, Franz Korf, Thomas C. Schmidt
HAW-Hamburg, Department Informatik
Berliner Tor 7, D-20099 Hamburg, Germany
{till.steinbach, hermand.dieumo, korf, schmidt}@informatik.haw-hamburg.de

ABSTRACT

Real-time extensions to standard switched Ethernet widen the realm of computer networking into the time-critical domain. These technologies have started to establish in process automation, while Ethernet-based communication infrastructures in vehicles are novel and challenged by particularly hard real-time constraints. Simulation tools are of vital importance to explore the technical feasibility and facilitate the distributed process of vehicle infrastructure design.

This paper introduces an extension of the OMNeT++ INET framework for simulating real-time Ethernet with high temporal accuracy. Our module implements the TTEthernet protocol, a real-time extension to standard Ethernet that is proposed for standardisation. We present the major implementation aspects of the simulation model and apply our tool to an abstract in-vehicle backbone. A careful evaluation that compares our results with calculations obtained from a mathematical framework, as well as with real-world measurements using TTEthernet hardware shows simulation and reality in good agreement.

Categories and Subject Descriptors

C.2.2 [Computer-Communication Networks]: Network Protocols—TTEthernet; I.6.3 [Simulation and Modeling]: Applications—In-vehicle network simulation; I.6.5 [Simulation and Modeling]: Model Development—Modeling methodologies

General Terms

Design, Measurement, Verification

Keywords

OMNeT++, Network traffic simulation, In-vehicle network, Time-triggered, TTEthernet

1. INTRODUCTION

Various fields of application require communication in fixed temporal bounds. Examples are process control or automotive applications. The increasing demand of bandwidth requires new concepts for real-time communication in

those fields; real-time Ethernet is a promising candidate for the upcoming tasks.

Ethernet has proven to be a flexible, widely deployed, and highly scalable protocol. Current Ethernet is a technology based on switching that also allows to increase the amount of traffic simultaneously transferred, by using segregated communication in groups. However, due to its randomised media access and best effort approach, it does not provide reliable temporal performance bounds. Real-time extensions to Ethernet promise to overcome those obstacles.

In process automation, several Ethernet-based products already provide real-time functionality for tasks with strict temporal constraints. The use of real-time Ethernet in an automotive communication backbone is novel.

To evaluate the feasibility of an Ethernet-based unified communication infrastructure for future in-vehicle applications, it is important to identify characteristic and comparable network metrics. A future option of technological and economic success can be only made on the ground of these valid performance estimates. Further, a detailed quantitative analysis allows for optimisations, including a critical view on proposed solutions, and can be used to rate improvements of the components and protocols. Event-based simulation is suitable for these analyses and provides the necessary performance estimates. Since automotive applications have very strict requirements – usually the temporal precision must be within a few microseconds – it is necessary to simulate the temporal behaviour with high accuracy.

During the development process of a car, several service providers and suppliers are involved. Different components are designed by various partners. The Original Equipment Manufacturer (OEM) is responsible for the design and configuration of the in-vehicle communication infrastructure. Temporal attributes and information about the load of an in-vehicle backbone are necessary at an early stage of the development, when the components are not yet available. Further extensions and build-to-order variants must be regarded and result in a large configuration space. Thus, a simulation-based evaluation strategy is indispensable for the design of an in-vehicle backbone. It allows the evaluation of different configuration parameters before construction and in a more comprehensive manner. Requirements for suppliers and service providers are assigned based on the simulation results.

This paper presents a simulation model [4] for TTEthernet [16], a real-time Ethernet extension by TTTech, for experimenting with future in-vehicle communication networks. TTEthernet is currently proposed for standardisation [13].

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

OMNeT++ 2011, March 21, Barcelona, Spain

Copyright © 2011 ICST 978-1-936968-00-8

DOI 10.4108/icst.simutools.2011.245510

It integrates time-triggered and event-triggered traffic on the same layer 2 infrastructure. For time-triggered communication, a coordinated policy based on time-division-multiple-access (TDMA) is used. This requires a model with local clocks on each device and a distributed clock synchronisation protocol to provide a global time. To achieve simulation results with the necessary precision, it is unavoidable to model the local clocks and the synchronisation accurately.

OMNeT++ and the INET framework have been chosen as basis for this development. One major reason for selecting OMNeT++ was given by its object oriented design. Since TTEthernet is based on standard Ethernet, many parts in the physical and MAC layer could be inherited or extended from the INET model. The presented implementation consists of the parts TTEthernet host-, switch-, and clock-model. The model of the host contains an implementation of the TTEthernet-API and allows a direct integration of application code into the simulation.

To validate the model and its simulation performance, an abstract in-vehicle backbone was simulated; the results were contrasted with calculations of a mathematical framework of time-triggered Ethernet, as well as results obtained from real-world measurements of TTEthernet hardware.

The remainder of this paper is organised as follows. In section 2, related work and the base techniques that are essential for the TTEthernet protocol are introduced. Section 3 explains the concepts behind the TTEthernet OMNeT++ model and the extensions that were done. Section 4 shows selected details of the implementation. Simulation results and an evaluation of the implementation of the TTEthernet model are discussed in section 5. Finally, section 6 concludes and gives an outlook.

2. BACKGROUND & RELATED WORK

2.1 Fundamentals

Real-time Ethernet extensions can be classified in three categories: *Token-based*, *bandwidth-limiting*, and *time-triggered* systems. Time-triggered systems are common in the automotive industry and therefore in focus of our work. Besides other protocols, that are mainly deployed in process automation like PROFINET [12], TTEthernet is a time-triggered real-time Ethernet extension that satisfies the special needs of in-vehicle applications. It allows standard best-effort communication and hard real-time network traffic to share the same layer 2 infrastructure. The TTEthernet specification [16] was developed by TTTech and is currently proposed for standardisation by the Society of Automotive Engineers [13].

TTEthernet relies on switched Ethernet. Any topology is formed of switches that relay the messages. Redundancy is achieved by multiple redundant channels.

Time-triggered Ethernet is centered around periodic cycles. For real-time communication, each node is assigned to offline configured timeslots. This coordinated TDMA based access policy ensures predictable transmission delays without queuing, and therefore low latency and jitter. To allow each node to access its dedicated transmission slot, all components have their own local clock and transmission schedule. Since a global synchronised time across all participants is needed, the TTEthernet specification defines a fail-safe synchronisation protocol.

Besides the *time-triggered* traffic, TTEthernet defines two

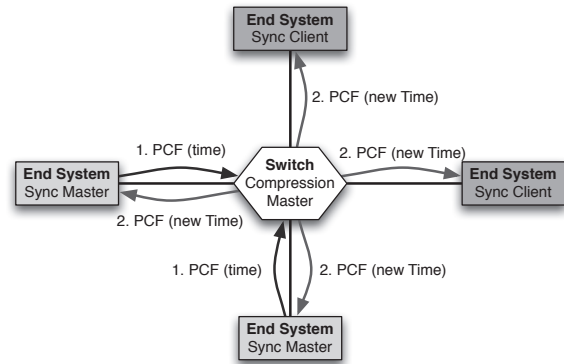


Figure 1: TTEthernet two step synchronisation

other traffic classes. *Rate-constrained* traffic is based on the AFDX-Protocol [1] and intended for communication with less rigid temporal requirements. *Best-effort* traffic has the lowest priority and is based on standard Ethernet. TTEthernet Networks are capable of working with hosts that are unaware of the time-triggered protocol and thus remain unsynchronised. Those hosts only communicate by best-effort traffic.

For the simulation of TTEthernet networks, the components of the INET framework must be extended. The coordinated TDMA based access policy requires a scheduler on each switch and host, that controls the transmission of real-time traffic. To support scheduled transmission, the model contains a clock for each node. The clocks are synchronised with a fail-safe two step synchronisation protocol.

2.2 Synchronisation

The TTEthernet synchronisation protocol defines three roles: Synchronisation masters start the synchronisation, compression masters calculate the global time, synchronisation clients receive the global time.

Synchronisation in TTEthernet is based on Ethernet messages called protocol control frames (PCF). Each synchronisation master sends its PCF at a dedicated time in the cycle. The compression master collects the PCF of each synchronisation master and calculates a global time of all messages. Then, the new time is broadcasted in a new PCF to all participants. Each node in the network can have one or more roles in the synchronisation. Figure 1 shows an example with two synchronisation masters and two synchronisation clients on end systems and the switch as compression master.

2.3 Traffic Classes in TTEthernet

TTEthernet defines three traffic classes: Time-triggered, rate-constrained and best-effort traffic.

2.3.1 Time-triggered Traffic

A global out of band schedule defines a plan for transmission or relaying in the time-triggered domain. The schedule is shared among all TTEthernet participants and switches. It contains events for receiving and sending of time-triggered messages. Routing of time-triggered traffic is static. This allows a completely deterministic operation, but requires an offline configured system.

Time-triggered messages use a content oriented addressing format, similar to Ethernet multicast. Instead of addressing a node in the network, the content of a frame is determined by the destination address. The 48 bit of the destination

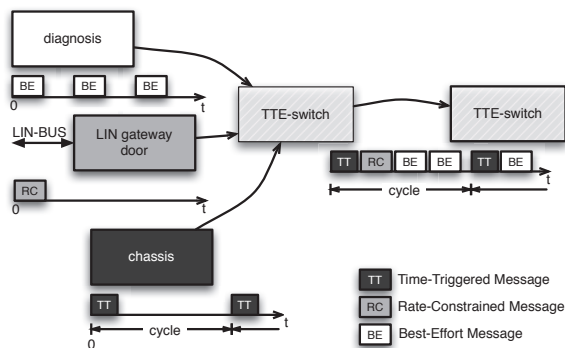


Figure 2: TTEthernet in-vehicle sample application

address are divided into two parts. The first part is the critical traffic marker. It is used to detect real-time traffic. The critical traffic ID (CT-ID) is located in the second part. The CT-ID is unique for each message in the cycle and used to address the message in the TTEthernet schedules. Routing decisions are based on the CT-ID.

2.3.2 Rate-constrained Traffic

Rate-constrained Traffic is based on the ARINC 664 standard [1]; known as the Avionics Full Duplex Switched Ethernet (AFDX). For rate-constrained traffic, it is ensured that sufficient bandwidth is available for each message (identified by its CT-ID). To achieve this, *Bandwidth Allocation Gap accounts* (BAG-accounts) are defined. BAG-accounts determine the minimum time between two subsequent messages with the same CT-ID and their maximum length. The application sending the RC message has to respect the constraints of the configured BAG-account, otherwise the message will be considered as invalid and is dropped by the switch.

Since forwarding is asynchronous and with lower priority than the forwarding of time-triggered traffic, the exact sending time of rate-constrained messages is unpredictable. Nevertheless, rate-constrained messages stay below an upper bound of delay and jitter [3], that is provable based on the configuration of the network.

2.3.3 Best-effort Traffic

The best-effort traffic in TTEthernet equals standard Ethernet traffic. It has the lowest priority of all three traffic classes and is relayed in idle times within the cycle. TTEthernet switches allow fixed routing of best-effort messages or auto-learning of MAC tables, which is the usual procedure in standard switched Ethernet. Best-effort traffic provides no guarantee for the upper bound of latency or even the delivery of packages.

Figure 2 visualises a sample application in the automotive context, where different traffic classes coexist on the same link. While time-triggered traffic, e.g. for chassis information, is transferred in periodic cycles with the highest priority, rate-constrained messages are transferred in free slots between the time-triggered messages. Afterwards free bandwidth is filled with best-effort traffic.

2.4 Related Work

Simulation of real-time Ethernet backbones that suits the requirements of the automotive industry requires a high

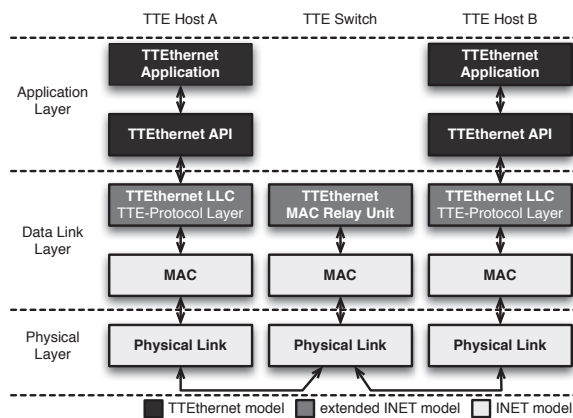


Figure 3: TTEthernet integration in INET

temporal precision. For OMNeT++, there have been already efforts [10] to implement real-time Ethernet protocols for experiments and predication of the protocol behaviour that demonstrated different strategies for real-time traffic transmission. It was shown that real-time networks have special traffic characteristics and requirements. For the RTnet protocol, an OMNeT++ model with less rigid temporal constraints was implemented [7]. In contrast, the model presented in this work complies to the firm temporal requirements of automotive applications.

TTEthernet is a real-time Ethernet extension that combines different types of traffic on the same layer 2 infrastructure. Experiences with the simulation of coexistent real-time Ethernet protocols [5] were used in the proposed model. Since TTEthernet uses local clocks, basic concepts may be taken from simulation models of the IEEE 1588 precision clock synchronisation protocol [11].

To the best of our knowledge, there are currently no complete OMNeT++ implementations of real-time Ethernet extensions publicly available. The University of Padova published models for industrial real-time Ethernet communication systems for the OPNET modeler [6,14]. In the automotive domain, there were simulation approaches based on a coupled framework of SystemC and OMNeT++ to verify the core requirements of switched in-vehicle communication [9].

3. CONCEPTS & MODEL

The OMNeT++ platform with the INET framework is the basis of our TTEthernet model. The fundamental network functionality is inherited from the INET framework. Similar to INET, the extension is provided as a shared library. Figure 3 shows the major components of the model.

3.1 Clock Model

The most significant difference between standard Ethernet and TTEthernet is the globally synchronised time and the schedule for each instance in the network. Each node in a TTEthernet network needs its own clock that is synchronised with other clocks through the reliable synchronisation protocol. The clock in TTEthernet is based on ticks. The time between two ticks is configurable. All clocks have a certain inaccuracy called clock drift [17]. Clock drifts may have significant influence on the overall protocol behaviour and must therefore be carefully included in the model. Ob-

viously, it is impossible to model the drift by simulating each tick of the clock as a separate event, as this would slow down the simulation extraordinarily.

In our clock model, several ticks are simulated in one step. To take account for the clock drift, a drift factor is introduced and assumed constant for a configurable interval. Equation 1 shows the formula for the clock calculations:

$$t' = t + \delta * (\Delta t_{Tick} + \Delta t_{Drift}) \quad (1)$$

Where t' is the time of the next event, t is the current simulation time, δ is the amount of ticks in the schedule for the appropriate event, Δt_{Tick} is the time for one tick, and Δt_{Drift} is the average drift. This is a valid and sufficiently accurate simplification of the clock model, since the variance of the average drift in real-world clocks is very low during one cycle [17].

The clock drift causes the clock to “loose” accuracy. The cyclic synchronisation messages (protocol control frames) reset the clock to a common point in the cycle.

3.2 Switch Model

TTEthernet requires a switch extension that is aware of the time-triggered protocol. A TTEthernet switch subsumes a standard Ethernet device for best-effort traffic and a module that contains the schedule and the local clock and implements the synchronisation protocol and relaying of critical traffic (time-triggered and rate-constrained).

For the implementation of the TTEthernet switch module, the INET MACRelayUnit interface was used. The relaying logic for critical traffic uses the clock described above. Further, the critical traffic module contains the forwarding logic. Arriving packets are classified by the destination address, as defined by the TTEthernet specification. Based on the different message types the forwarding decision is made:

- Sync messages are evaluated in the synchronisation module of the device
- Time-triggered messages are stored in a buffer until their dispatch time is scheduled
- Rate-constrained messages are forwarded as soon as possible (but with lower priority than time-triggered traffic)
- Best-effort messages are relayed in idle times

Immediately upon reception of a frame, the TTEthernet switch checks whether the message conforms to the pre-configured constraints. For critical traffic, special conformance checks are configured in a configuration table of the switch (CTC-Table). The switch verifies whether the message arrived on the correct port and with the right timing. For rate-constrained messages the compliance with the bandwidth allocation gap accounts is checked. For time-triggered messages the switch checks whether the message arrived at the scheduled time. These checks are important to prevent the system from being corrupted by defective senders that do not follow temporal constraints. Even if a device pretends to be a time-triggered sender, the messages are dropped while arriving at a wrong port. Best-effort frames are allowed to arrive at any port and any time in the schedule. This allows the user to add devices to the network that are unaware of the time-triggered protocol.

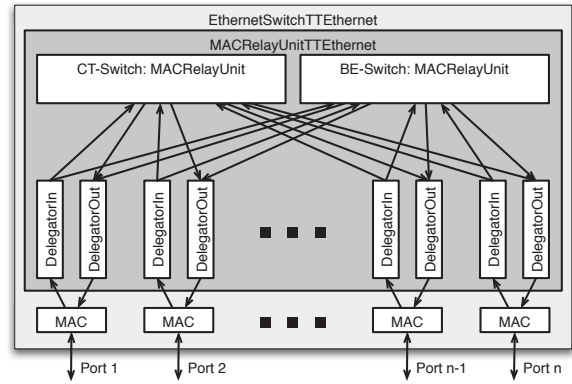


Figure 4: Switch design with two relay units and two delegators per port

3.3 Host Model

To realise TTEthernet on an end system, the standard Ethernet protocol stack has to be extended with the TT-Ethernet service. The TTEthernet service on the end system consists of a message schedule, a scheduler, a classification module, and a synchronization module. The message schedule is configured offline and determines when a time-triggered message is to be sent or expected. The scheduler contains a task to trigger the sending and receiving of messages, based on the message schedule. The classification module classifies messages at arrival and processes them in conformance with the schedule.

4. IMPLEMENTATION DETAILS

During the implementation, the concepts of the INET-Framework were followed as closely as possible. Inheritance was used to extend existing structures where adjustments to the INET modules were necessary. New modules were only designed when extensions could not satisfy the special needs of the time-triggered Ethernet.

4.1 TTEthernet Switch

The functionality of a TTEthernet switch can be divided in two logical relay units. One relay unit is responsible for the best-effort traffic and one forwards critical-traffic (rate-constrained and time-triggered). Both units share the same physical ports. Special modules between the relay units and the MAC layer observe the adherence of priorities and the media access policy.

For our simulation model, we adapted this view of a TT-Ethernet component. This grants use of the standard INET implementations for the best-effort part of the TTEthernet switch and thereby reduces the implementation effort. Like all Ethernet switches in INET, the TTEthernet switch is based on the MACRelayUnit interface. The TTEthernetRelayUnit itself contains two MACRelayUnits, one for best-effort and one for critical traffic (figure 4).

Each port of the switch has two delegator modules to grant access to the lower layer in- and outputs for each relay unit. The input delegator is responsible for relaying incoming frames from the EtherMAC module to the appropriate switch. It checks whether the incoming frame is critical traffic using the destination MAC address of the frame. Then the frame is forwarded to the determined relay unit. For

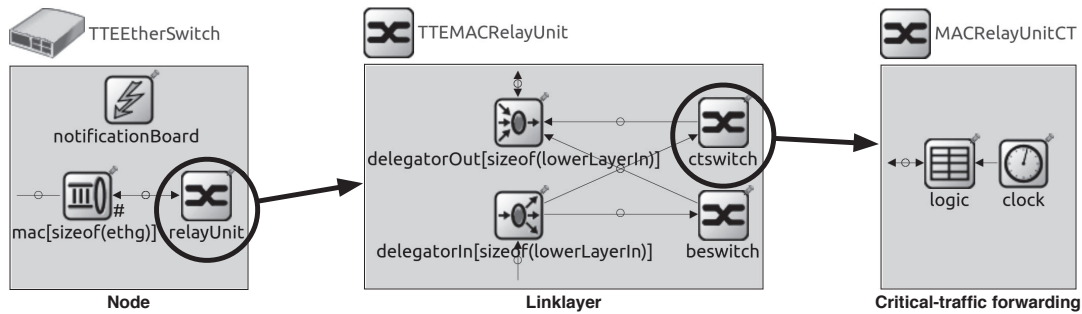


Figure 5: Overview over the layers of the TTEthernet switch model

outgoing frames, the output delegator receives frames from the output ports of both relay units and forwards them to the EtherMAC. Figure 4 shows the usage of delegators.

In time-triggered Ethernet, a time-triggered frame should never be delayed due to other frames blocking the line card. Therefore, the outgoing delegator ensures that frames are forwarded with the appropriate priority. Before sending a rate-constrained or best-effort message, the delegator checks whether the message can be transmitted within the time until the next time-triggered message is released. This way, the EtherMAC is idle when a time-triggered message is scheduled. During this reserved time, all best-effort and rate-constrained messages are stored in a separate buffer at the delegator. When the reserved mode is released again, the messages in the buffer are forwarded.

A problem in forwarding of time-triggered messages is the buffer in the EtherMAC module. Even when the delegator is in reserved mode, there may be frames in the buffer of the MAC layer that delay the transmission of the time-triggered message. We have overcome this deficiency by preventing the delegator from using the buffer. The delegator only sends the next frame to the MAC layer when the MAC buffer is empty. This behaviour depends on a message from the EtherMAC module to the delegator, indicating that a frame was sent and the interface is idle again. Since the standard implementation of EtherMAC does not allow such message exchange directly between the modules, we took advantage of the INET notification board.

The notification board is a publisher subscriber system. Each module is allowed to place notifications to the board or register for events that are delivered through a callback method. The EtherMAC modules put several events containing their current state to the notification board. One message is the `NF_PP_TX_END` event that indicates the end of delivery of a frame. The delegator now receives the event, checks whether the buffer of the MAC layer is empty and then sends its next frame. This technique allows us to use the standard EtherMAC modules. Further investigations should verify whether there is a significant performance profit with a tailored EtherMAC module that has no buffer and sends its idle message directly to the delegator.

Figure 5 displays the TTEthernet switch implementation in OMNeT++. The implementation is divided in three layers. The basic layer contains the EtherMAC modules, the notification board, and the relay unit. The relay unit consists of the delegators for incoming and outgoing traffic and the relay units for critical and best-effort traffic. The relay unit for critical traffic contains the forwarding logic for critical traffic and the local clock.

4.2 TTEthernet Host

To implement the TTEthernet protocol stack in the simulation environment, two major extensions to the INET Ethernet protocol stack (figure 3) were made: An extension of the INET Logical-Link-Control (LLC) and the implementation of the TTEthernet-API [18] for the simulation.

4.2.1 TTEthernet LLC sublayer

The TTEthernet LLC sublayer extends the LLC sublayer of the INET-Framework with the TTEthernet services (figure 6). It is connected to a scheduler and responds to its commands. For each time-triggered message, a transmission and a receive buffer is reserved. The buffers are identified by the CT-ID of the message. The scheduler is message-based and determines which messages the TTEthernet LLC sublayer should read from the buffers and send. For example, if the sending timeslot of a time-triggered message has arrived, the scheduler sends a `TTOutgoing` traffic event command with the corresponding CT-ID to the TTEthernet LLC sublayer. The layer then reads the message from the buffer, packages it in an INET Ethernet frame and finally forwards it to the MAC sublayer. In addition, tasks connect time-triggered applications with the schedule. A task is activated by a special event sent by the scheduler. This concept allows applications to provide the data immediately prior to transmission.

4.2.2 TTEthernet-API

The TTEthernet-API [18] was defined by TTTech and serves for sending and receiving messages at the TTEthernet application level. It connects the application layer and the TTEthernet LLC sublayer (figure 3). Sending and receiving messages with the TTEthernet-API is based on buffers. Buffer-based message access is an abstract method of sending and receiving messages, that optimises real-time performance. It allows the application to access the message payload byte-wise. Each application has one or more handlers that control access to the buffers. The handlers are usually determined at the initialisation phase of the application and include a controller ID to identify the devices, the traffic type for the classification of messages, and an indication to detect the buffer direction (transmit or receive). When the application tries to send or receive a message, it has to specify the frame (header information and buffer for the payload), as well as the buffer handler. While time-triggered messages are stored in transmission and receive buffers, best-effort frames are stored in queues.

The implementation of the TTEthernet-API allows the

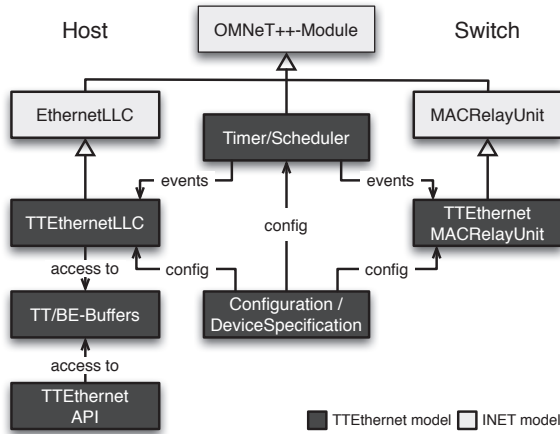


Figure 6: TTEthernet protocol layer object model

integration of real applications during development or testing. There are several techniques (sockets, shared libraries and code integration) to enable the integration of applications [8]. We are currently using code integration, since this is the common, suitable method for an early stage of development.

5. PERFORMANCE & EVALUATION

For proof-of-concept, several simulations and an evaluation of the presented simulation model are made. A validation based on an analytical framework and real-world measurement guarantees the accuracy of the model.

5.1 Simulation Set-up and Methodology

We test and evaluate our TTEthernet implementation by simulating the behaviour of a synthetically generated topology of a simplified in-vehicle backbone network.

To check whether the model scales to our needs, generated topologies within our expected maximum backbone size are used for the evaluation of scalability and simulation model overhead. Based on current in-vehicle communication systems, we expect not more than 70 hosts and 7 switches in the simulation.

To show the metrics and protocol conformance of the TTEthernet model, we compare the latency results of a sample network with TTEthernet and standard INET switches. The simulated topology contains two interconnected switches, two TTEthernet hosts that are sending and receiving time-triggered traffic, and ten standard INET Ethernet hosts for best-effort traffic generation (figure 7). All links are 100Mbit/s. The clock drift of the components is configured to 200ppm. Wire delay is 100ns per link.

To measure the influence of best-effort traffic on the transmission the link between the two switches is stressed with varying load between 0% and 100%. During the simulation minimum and maximum end-to-end latency was recorded over 10,000 frames. The boundaries of end-to-end latency is the most important metric for in-vehicle communication.

5.2 Overhead of the Simulation Model

We determine the overhead introduced by the TTEthernet extensions by comparing CPU time and memory consumption of simulations of identical topologies in the INET

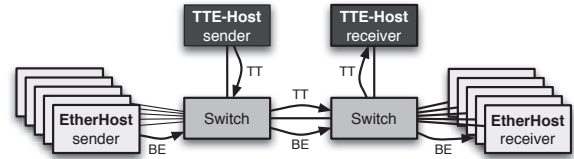


Figure 7: Simulated topology

Ethernet model and our TTEthernet model. The TTEthernet protocol performs several conformance checks for incoming frames that are not present in standard Ethernet. Further, approximately 3% more events are needed due to the clock model in TTEthernet. The result is an approx. 10% higher CPU time for the TTEthernet simulation than in the INET Ethernet model. Memory consumption for the TTEthernet model is approximately 60% higher than in INET. The latter is caused by several additional buffers that hold time-triggered and rate-constrained messages.

Our simulation runs on PC hardware with two cores of 2.4 GHz and 2GB memory. Experiments show that CPU time and memory consumption has a linear dependency on the amount of time-triggered messages in the cycle and nodes in the network. With our expected maximum of 70 hosts and 7 switches in a future in-vehicle backbone design, the simulation is sufficiently fast for our experiments.

5.3 Results

We measure the peak values of end-to-end latency between the time-triggered hosts, both for the standard INET switch and the TTEthernet model. The results in figure 8 show the maximum and minimum end-to-end-latency for the topology shown in figure 7, taken from 10,000 packets, while the network is utilised by a varying amount of best-effort traffic.

TTEthernet promises reliable communication with fixed temporal bounds. Figure 8 shows that due to the synchronised protocol, the priorities and the link reservation mechanism (see section 2.1), a uniform latency is achieved independently of the utilisation. The slight difference between minimum and maximum latency is caused by the clock drifts of the timers (section 3.1) and can be analytically verified [15].

For the standard INET switch, the difference between minimum and maximum latency increases with increasing link utilisation due to queueing on the linecards. The communication becomes unstable on a fully loaded link – the transition capacities are unbalanced and frames are dropped.

Figure 9 shows the latency distribution for the three traffic classes in the same fully utilised network with TTEthernet switches. The time-triggered messages have an almost constant latency. Rate-constrained messages are forwarded at a variable but limited forwarding delay below 500µs. Because of queueing and bandwidth reservation, the best-effort traffic has high jitter.

Latency distribution is an important metric, that characterises the topology and configuration of a network and facilitates the decision, whether the forwarding of event-triggered traffic matches the requirements of the applications that use rate-constrained or best-effort communication.

5.4 Verification

To verify the core features of our implementation, we compare the results obtained from simulation with an analytical model and with real-world measurements taken from

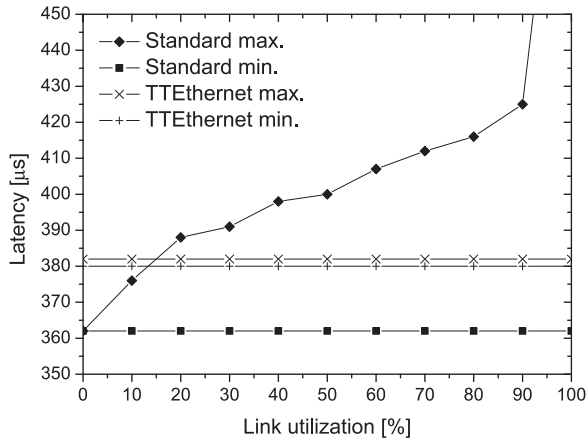


Figure 8: End-to-end latency comparison – Standard INET switch and TTEthernet switch

TTEthernet hardware. Since only one TTEthernet switch was available, a simpler topology is used for the verification in this section. A more complex comparison based on the methods presented here can be done in a straight-forward manner.

The evaluation is applied to a reduced network, containing one switch and several hosts. Each communication among end nodes contains one switching hop. We use different schedules in the switch, with the result of different forwarding delays.

The simulation is run over 10,000 messages for a switch schedule of 350μs between reception and forwarding of the frame and a schedule of 9μs, which is currently the minimum at reliable operation. The schedule of 9μs leads to a simulated end-to-end latency of 19.5μs for a frame with minimum payload, to 252.0μs for a full size frame. For the schedule of 350μs delay, the end-to-end latency is – depending on frame size – between 360.5μs and 593.0μs.

5.4.1 Mathematical Model

Our mathematical model of the TTEthernet components was created for a competitive comparison of theoretical performance attributes of TTEthernet and FlexRay [15].

Following equations of this framework calculate the time behaviour of TTEthernet components. The Latency t_L , calculated for minimum and maximum payload, is composed of signal runtime $t_{WD} * l_W$, frame transmission $2 * l_F * t_b$, and the delays of the switch schedule t_{SD} :

$$t_L(t_{SD}) = t_{WD} * l_W + 2 * l_F * t_b + t_{SD} \quad (2)$$

$$t_{L_{min}}(t_{SD}) = 10 \frac{ns}{m} * 0.5m + 2 * 5.12\mu s + t_{SD} \quad (3)$$

$$t_{L_{max}}(t_{SD}) = 10 \frac{ns}{m} * 0.5m + 2 * 121.44\mu s + t_{SD} \quad (4)$$

As expected we see a linear dependency on the frame length l_F . For a schedule with $t_{SD} = 350\mu s$ delay we get a latency between 360.245μs and 592.885μs, depending on the payload size. The same calculation with the minimal switch delay of $t_{SD} = 9\mu s$ results in an latency between 19.245μs and 251.885μs.

The results of the mathematical framework are slightly (<300ns) lower than the simulation results. This is because

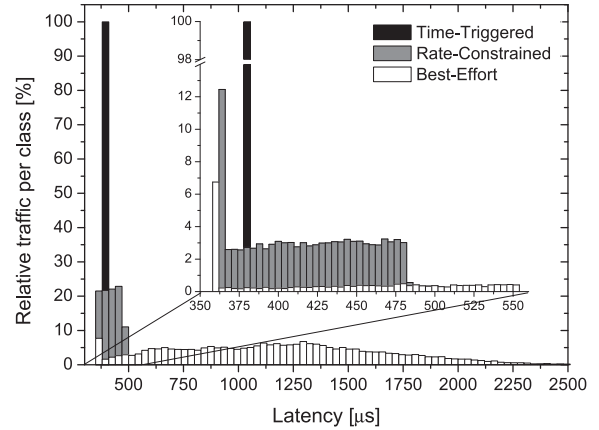


Figure 9: Latency distribution of the three traffic classes with completely utilised link

of the simplification of hardware delays that were made within the mathematical model.

5.4.2 Measurement using TTEthernet Hardware

To compare the simulation results with the behaviour of real time-triggered Ethernet hardware, we developed a performance measurement method for time-triggered Ethernet networks that is based on off-the-shelf-components [2]. The setup is based on an embedded system running a real-time Linux kernel with two Ethernet ports. The network under test is connected by the two ports. For the measurement, timestamped time-triggered messages are sent from the first port, traverse the network under test, and arrive at the second port. During arrival, the message is timestamped again. This method allows to measure the latency, as well as the jitter of the network under test, at a precision of microseconds.

For the set-up described above, and a schedule of 350μs, we measure a latency of 360μs for a minimal and 592μs for a maximal framesize.

Besides the lower precision of the measurement approach, the results closely agree with our simulation values and the results of the mathematical model. Since the measurement is based on a software implementation of the TTEthernet stack that provides less precise synchronisation, we were unable to measure the 9μs schedule without packet loss.

5.5 Discussion

The simulation model achieves the expected behaviour of a TTEthernet system with high accuracy. The evaluation results demonstrate that the simulation model does conform to the TTEthernet specification and matches with the mathematical framework, as well as with measurements obtained from real-world experiments. Latency of time-triggered messages is constant, regardless of the amount of background traffic that simultaneously traverses the switch. Table 1 compares the results of the different approaches for schedules with 9μs and 350μs delay.

The evaluation of the simulation overhead shows that the model consumes more resources than the INET standard Ethernet implementation. The CPU time is approximately 10% higher and 60% more memory is used by the TTEther-

Table 1: Comparison of latency bounds with maximum and minimum payload

Payload	Simulation	Theory	Measurement
350μs Schedule			
minimum	360.5 μ s	360.245 μ s	360 μ s
maximum	593.0 μ s	592.885 μ s	592 μ s
9μs Schedule			
minimum	19.5 μ s	19.245 μ s	-
maximum	252.0 μ s	251.885 μ s	-

net model. CPU time and memory consumption have a linear dependency on the number of nodes in the network and the amount of messages in the cycle. Simulation of topologies within the expected network sizes run sufficiently fast on standard PC hardware to support experiments of future in-vehicle communication.

6. CONCLUSION & OUTLOOK

Real-time Ethernet is a realistic candidate for next generation in-vehicle backbones. Therefore, simulation results of Ethernet based in-vehicle communication gains importance for OEMs and suppliers. This paper presented an accurate, well scaling simulation model for real-time Ethernet-based in-vehicle backbone designs.

Simulation results have been carefully evaluated by comparing the results of the simulation with a mathematical framework and measurements of real TTEthernet hardware. The presented TTEthernet module [4] tightly conforms to the temporal specifications of time-triggered traffic and allows for a realistic modeling and analysis of in-vehicle networks. The implementation of the TTEthernet-API further allows to simulate application behaviour. Since configurations of the proposed TTEthernet simulation model rely on the same XML based configuration files that are used for real-hardware, it is unchallenging to transfer real scenarios to the simulation and vice versa. This greatly simplifies the exchange between simulation and experiment and thereby accelerates the simulation-driven design.

In future work we plan a clock module extension where the simulated drift of the clocks of all devices is artificially synchronised with each other. This will improve simulation coverage. It allows to generate dedicated worst case scenarios where the timers have the largest difference and thus the communication has the highest jitter.

Additional work will focus on the simulation of future in-vehicle communication networks, based on concrete automobile topologies and application data. Our goal is to build a development tool chain for in-vehicle networks and applications around the simulation in OMNeT++. Using such an environment, we will analyse how the simulation of TTEthernet can guarantee a smooth integration of time-triggered Ethernet systems into current in-vehicle communication infrastructures.

7. REFERENCES

[1] Aeronautical Radio Incorporated. Aircraft Data Network. Standard 664, ARINC, Annapolis, 2002.

[2] F. Bartols, T. Steinbach, F. Korf, and T. C. Schmidt. Performance Analysis of Time-Triggered Ether-Networks Using Off-The-Shelf-Components. In *14th IEEE International Symposium on*

Object/Component/Service-oriented Real-time Distributed Computing, 2011. to appear.

[3] H. Charara, J.-L. Scharbag, J. Ermont, and C. Fraboul. Methods for bounding end-to-end delays on an AFDX network. In *18th Euromicro Conference on Real-Time Systems*, 2006.

[4] CoRE RG. TTE for INET. <http://www.informatik.haw-hamburg.de/tte4inet.html>

[5] P. Ferrari, A. Flammini, S. Rinaldi, and G. Gaderer. Evaluation of clock synchronization accuracy of coexistent Real-Time Ethernet protocols. In *IEEE International Symposium on Precision Clock Synchronization for Measurement, Control and Communication*, pages 87–91, Sept. 2008.

[6] P. Ferrari, A. Flammini, and S. Vitturi. Performance analysis of PROFINET networks. *Computer Standards & Interfaces*, 28:369–385, Apr. 2006.

[7] C. Ling and Z. Wang. Performance Analysis of A Real-Time Ethernet Using OMNeT++. Masterthesis, TU München, München, Sept. 2008.

[8] C. P. Mayer and T. Gamer. Integrating real world applications into OMNeT++. Technical report, Universität Karlsruhe (TH), Institut für Telematik, Karlsruhe, 2008.

[9] B. Müller-Rathgeber, M. Eichhorn, and H.-U. Michel. A unified Car-IT Communication-Architecture: Network switch design guidelines. In *IEEE International Conference on Vehicular Electronics and Safety*, pages 16–21, Sept. 2008.

[10] T. Pensawat. Real-Time Ethernet Networks Simulation Model. Masterthesis, Halmstad University, Halmstad, Sweden, Dec. 2006.

[11] F. Praus, W. Granzer, G. Gaderer, and T. Sauter. A simulation framework for fault-tolerant clock synchronization in industrial automation networks. In *IEEE Conference on Emerging Technologies and Factory Automation 2007*, pages 1465–1472, 2007.

[12] PROFIBUS & PROFINET International. Profinet.

[13] SAE - AS-2D Time Triggered Systems and Architecture Committee. Time-Triggered Ethernet (AS 6802), 2009.

[14] L. Seno and C. Zunino. A simulation approach to a Real-Time Ethernet protocol: EtherCAT. In *IEEE Int. Conference on Emerging Technologies and Factory Automation 2008*, pages 440–443, Sept. 2008.

[15] T. Steinbach, F. Korf, and T. C. Schmidt. Comparing Time-Triggered Ethernet with FlexRay: An Evaluation of Competing Approaches to Real-time for In-Vehicle Networks. In *8th IEEE Intern. Workshop on Factory Communication Systems*, pages 199–202, Piscataway, New Jersey, May 2010. IEEE Press.

[16] W. Steiner. TTEthernet Specification. TTTech Computertechnik AG, Nov. 2008.

[17] D. B. Sullivan, D. W. Allan, D. A. Howe, and F. L. Walls, editors. *Characterization of Clocks and Oscillators*. National Institute of Standards and Technology, Boulder, Colorado, 1990. technical note 1337.

[18] TTTech Computertechnik AG. TTEthernet Application Programming Interface. TTTech Computertechnik AG, Dec. 2008.