A Component-based Simulation Model and its Implementation of a Switched Ethernet Network

Symposium on Automotive/Avionics Systems Engineering

Jörg Sommer and Walter Franz (Daimler AG)
joerg.sommer@ikr.uni-stuttgart.de
2009, October 16th (San Diego, CA, USA)

Universität Stuttgart
Institute of Communication Networks
and Computer Engineering (IKR)
Prof. Dr.-Ing. Andreas Kirstädter
Outline

Motivation and Definition
Modeling and Settings
Architecture and Conceptual Structure
Simulation Basics
Examples
Conclusion and Outlook
Why Ethernet

Ethernet’s Story of Success

- Ethernet evolved from a bus topology to a micro-segmented network (with switches and point-to-point connections)
- Today, it is the predominant LAN technology
- It is also used in different embedded application fields, e.g., avionics (AFDX standard)
- It will find its way into further (embedded) application fields, e.g., in cars
Why Ethernet

• Fast bandwidth evolution in the mid/late 1990-ies, while maintaining backward-compatibility
• Scalability with respect to network size, planning reliability, easy management, and plug-and-play capabilities
• Besides wired interconnection, it also supports the interconnection of wireless technologies, e.g. IEEE 802.11
• A high degree of freedom regarding the network topology and design
• Easy integration with other network technologies
  → attractive for industrial domains to interconnect, e.g., plant level and control level
• Long lifetime
• Low cost thanks to
  – the use of commercial-off the shell (mass-market) components
  – large number of vendors and wide range of products
• ...
### Definition

#### Term Ethernet

- **Ethernet** today: Carrier Grade Ethernet, EPON, Industrial Ethernet, Avionics Full Duplex Ethernet (AFDX), Audio/Video Bridging (AVB) Ethernet, ...

- Definition of the term Ethernet
  - system based on **IEEE 802.3 standards**
  - supplemented by layer 2 concepts and management functions
  - point-to-point connections (**full duplex** operation mode; micro-segmentation)
  - frame Format (extended format, e.g. using traffic classes)

→ No domain-specific enhancements

Motivation

Ethernet’s Evolution and Penetration

Ethernet’s range of application areas is still growing, e.g.,

- Currently, the automotive industry is investigating Ethernet
- IEEE is specifying protocols and mechanisms supporting services for streaming multimedia applications with real-time requirements (AVB)

Why not adapting Ethernet one-to-one from one application area to another?

• In each area, unique requirements
  (in terms of real-time behavior, quality of services, data rate, price per unit, ...)
• New Ethernet-(based) standards and enhancements
  (examples are traffic shapers and policers to fulfill real-time requirements)

→ Strong need to evaluate new standards and enhancements
  (with less costs and in a short time)

→ Simulation as a perfect way

→ Strong need for a flexible simulation program with
  - a modular design enabling separate handling and treatment of components
  - a high reusability of components

→ Component-based Simulation Model
A Switched Ethernet network consists of:
- full duplex links,
- stations (terminals, ECU),
- and switches

Component-based simulation model:
- Decomposition of the network
- A model for each network entity and decomposition of the network entities into (sub)components
Full Duplex Link

- Switched Ethernet network
  → No need for CSMA/CD
- Full duplex link
  → Transmission and receiving link
- Each link: Two servers
  1. Delay of a frame transmission
     \[ T_{Tx} = \frac{\text{frame length}}{\text{bit rate}} + \text{propagation delay} \]
  2. Inter-frame gap (IFG)
     (96-bit times)
     \[ T_{IFG} = \frac{96 \text{ bit}}{\text{bit rate}} \]
     (max. link utilization < 100%)
## Settings

### Full Duplex Link

<table>
<thead>
<tr>
<th>(Sub-)Component</th>
<th>Parameter</th>
<th>Default value</th>
<th>Valid values / range</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>bit rate [Mbps]</td>
<td>100</td>
<td>1 ... ∞</td>
</tr>
<tr>
<td></td>
<td>IFG [bit times]</td>
<td>96</td>
<td>1 ... ∞</td>
</tr>
<tr>
<td></td>
<td>propagation delay [µs]</td>
<td>0</td>
<td>1 ... ∞</td>
</tr>
</tbody>
</table>

![Diagram](image-url)
Modeling

Station

Subdivided model (decomposition) with

- **Receiver part**
  - generators model higher layer applications and are individually configurable
  - multiplexer LLC models the Logical Link Control layer
  - single queue stores outgoing frames (Tx buffer)

- **Sender part**
  - single queue stores incoming frames (Rx buffer)
  - server $T_{NIC}$ delays frames in the Network Interface Card (NIC)
  - demultiplexer LLC delivers the frames to higher-layer communication ports
### Settings

#### Station

<table>
<thead>
<tr>
<th>(Sub-)Component</th>
<th>Parameter</th>
<th>Default value</th>
<th>Valid values / range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Generators</td>
<td>number of generators</td>
<td>0</td>
<td>0 .... ∞</td>
</tr>
<tr>
<td>Generator</td>
<td>arrival process</td>
<td>non</td>
<td>set of distributions and source models</td>
</tr>
<tr>
<td>Tx buffer</td>
<td>size [byte]</td>
<td>∞</td>
<td>1 ... ∞</td>
</tr>
<tr>
<td>Tx buffer</td>
<td>queuing discipline</td>
<td>FIFO</td>
<td>FIFO, strict priority</td>
</tr>
<tr>
<td>Rx buffer</td>
<td>size [byte]</td>
<td>∞</td>
<td>1 ... ∞</td>
</tr>
<tr>
<td>Rx buffer</td>
<td>queuing discipline</td>
<td>FIFO</td>
<td>FIFO, strict priority</td>
</tr>
<tr>
<td>NIC (receiver part)</td>
<td>T\textsubscript{NIC} [μs]</td>
<td>non</td>
<td>0 ... ∞</td>
</tr>
<tr>
<td>Sinks</td>
<td>number of sinks</td>
<td>1</td>
<td>1 .... ∞</td>
</tr>
</tbody>
</table>
Modeling

Switch

Main task: **Frame Forwarding**
using a forwarding database (FDB)

Subdivided model (decomposition) with

- **Input Stage**
  buffers incoming frames

- **Forwarding stage**
  with switching latency (fixed value)
  - address lookup
  - FDB update

- **Receiving stage**
  buffers outgoing frames

- **Memory management unit** (MMU)
  - handles memory requests to buffer frames
  - principle architectures: Shared memory and multiple queues
## Settings

### Switch

![Switch Diagram]

<table>
<thead>
<tr>
<th>(Sub-)Component</th>
<th>Parameter</th>
<th>Default value</th>
<th>Valid values / range</th>
</tr>
</thead>
<tbody>
<tr>
<td>MMU</td>
<td>architecture</td>
<td>non</td>
<td>shared memory, multiple queues</td>
</tr>
<tr>
<td>Tx buffer</td>
<td>size [byte]</td>
<td>∞</td>
<td>1 ... ∞</td>
</tr>
<tr>
<td>Tx buffer</td>
<td>queuing discipline</td>
<td>FIFO</td>
<td>FIFO, strict priority, token bucket</td>
</tr>
<tr>
<td>Input and output logic unit</td>
<td>forwarding algorithm</td>
<td>standard</td>
<td>standard, ring</td>
</tr>
<tr>
<td>Forwarding unit</td>
<td>$T_{SL}$ [μs]</td>
<td>non</td>
<td>0 ... ∞</td>
</tr>
</tbody>
</table>
Intermediate Summary

- Motivation Why Ethernet
- Motivation Ethernet’s evolution and penetration
  → Strong need for a flexible simulation program with a modular design and high reusability

- Component-based simulation model
  → decomposition of the network
  → a model for each network entity and decomposition of the network entities into (sub)components

- Network entities
  - full duplex links
  - stations
  - switches
• Simulation program bases upon the IKR Simulation Library (IKR SimLib)
• The IKR SimLib includes components for event-driven simulation, e.g.
  – global calendar and events
  – modular I/O concept
  – statistical evaluation
  – random number generators (various distributions and source models)
  – basic entities and components (FIFO queues, etc.)
  – ....
• IKR SimLib (Java Edition) bases upon the Java’s Base Libraries
• Emulation Library supports an interface between a simulation model and the real-world
Architecture

Overview of the IKR Simulation Library

• History
  – origin: Pascal simulation library (1980ies)
  – since then enhanced and improved continuously

• Basic structure
Conceptual Structure

Port Concept

• Connecting components in a **plug-n-play** manner
• Message exchange between model components via ports (communication via handshake protocol)
• Distinction between input and output ports
→ **Uniform interface** for all components
Conceptual Structure

Measurements: Message Filters and Meters

Filters

- Attached to port in a plug-n-play manner
- Motivation
  - evaluation or modification of message contents
  - message tracing

Meters

- Attached to ports via message filters
- Application
  - message counting/rate measurement
  - transfer time measurement
- Internal evaluation of measured values using statistic classes
Simulation Basics

Event-Driven Simulation

- System state changes at discrete points in time (e.g. arrival of a message, end of service, timer expiration)
- State change represented by an event
- Registration of events in a calendar (together with a timestamp representing planned time of occurrence)
- Sequential processing of events in the calendar by order of timestamps
  - update simulated system time
  - execute actions to realize state changes
  - plan successive events
Simulation Basics

Example: $M/M/1$ Single Server System

- Next event: arrival of a request
- New system time: 17.12
- State change
  - occupy server (if idle) or enter queue
  - update of statistics
- Successive events
  - next arrival event + (possible) end of service
  - timestamp: current system time + random duration according to corresponding distribution
  - register in the calendar

neg.-exp. distributed interarrival time with rate $\lambda$

neg.-exp. distributed service time with mean value $h$

17.12
arrival

calendar

25.31
end of service

21.84
arrival
Simulation Basics

Statistical Evaluation

- Examples for measured data
  - transfer time
  - queue length
  - loss

- Evaluation according to
  - moments (mean value, variance, coefficient of variation)
  - relative occurrence (distribution and density)
  - quantile (e.g. 99% quantile \( x_{0.99} \): \( F(x_{0.99}) = P\{X \leq x_{0.99}\} = 0.99 \))
  - coefficient of (auto-)correlation/(auto-)covariance

- Realization according to estimation functions or algorithms
Simulation Basics

Statistical Significance

• Questions
  – how exact are the simulation results compared to the real values?
  – how long does a simulation have to last in order to reach sufficient exactness?
• Commonly used: batch means method
  – partitioning of simulation into $n$ batches (e.g. $n = 10$)
  – assumption of statistical independence between batches
  – interpretation of batch results as samples
  – averaging and computation of the confidence interval

\[
\bar{X} = \frac{1}{n} \cdot \sum_{i=1}^{n} \bar{X}_i
\]
Example

Performing Simulation Studies

The IKR SimLib supports
• parsing input parameters
• printing results in XML format
Example

Practical Usage

Issues

- System dimensioning (e.g. buffer sizes in cost-sensitive fields like automotive)
- Latency and jitter (temporal requirements, e.g. for safety-critical applications)
- Evaluation of new standards and enhancements (e.g. AVB Ethernet)
Example

In-car Communication Network

- Applied in a research project with a car manufacturer
- Goal of this example
  - demonstration of the nature of the simulation results
  - benefits of the simulator
- Evaluation and comparison of in-car Ethernet topologies: Star and Daisy-Chain

![Diagram of in-car communication network with Ethernet topologies]
Example

Visualization End-to-End Latency

[Graph showing End-to-end Latency with different network topologies (Star, Daisy-Chain, and Fifo) and priority schemes (Strict priority, CoV = 1.0, CoV = 5.0) for various loss probabilities (ρ_{Rx,RR}).]
Example

Visualization Loss Probability

![Graph showing loss probability vs. queue size for different mean values (μ) and coefficients of variation (CoV). The graph includes lines for CoV = 5.0 and CoV = 1.0, with μ values of 0.550 ms, 0.575 ms, 0.600 ms, and 0.675 ms.](image)
Conclusion and Outlook

- Ethernet’s range of application areas is still growing
- Permanent evolution of Ethernet standards
  → Simulation as an excellent way (to reduce the development time and costs)
  → Strong need for a flexible simulation program with a modular design and high reusability
- Decomposition of the network and network entities
- Component-based simulation model with a model for each network entity
- Architecture and conceptual structures for a modular design
- Modular design enables a simple evaluation of new standards and enhancements (like novel traffic scheduler, etc.)

Outlook

- Use in further research projects (in different domains)
- Developing and evaluation of a novel dynamic link-state forwarding algorithm (monitors the link loads and reconfigures the logical topology)
  → Improves the QoS in terms of end-to-end delay