

Simulation von Time-Triggered Ethernet im Automobilkontext mit OMNeT++ : Einführung



Herman Dieumo Kenfack

Hochschule für Angewandte Wissenschaften Hamburg

Anwendungen 1 WS 09/10

Betreuer: Prof. Dr. Thomas C. Schmidt; Prof. Dr. Franz Korf

Agenda

- ▶ Einführung
- ▶ Aktuelle Situation in der Automobilindustrie
- ▶ Ethernet
- ▶ TTEthernet
- ▶ Gegenwertige Arbeiten an der HAW
- ▶ OMNeT++
- ▶ Modellierung des Netzwerks
- ▶ Fazit und Ausblick
- ▶ Literatur

Einführung

- ▶ Ziele
 - ▶ Erkundung des möglichen Einsatzes von TTEthernet für die Vernetzung im Kraftfahrzeug
 - ▶ Entwurf eines Modells für typische Kraftfahrzeug-Netzwerke
 - ▶ Simulation des Modells im OMNeT++

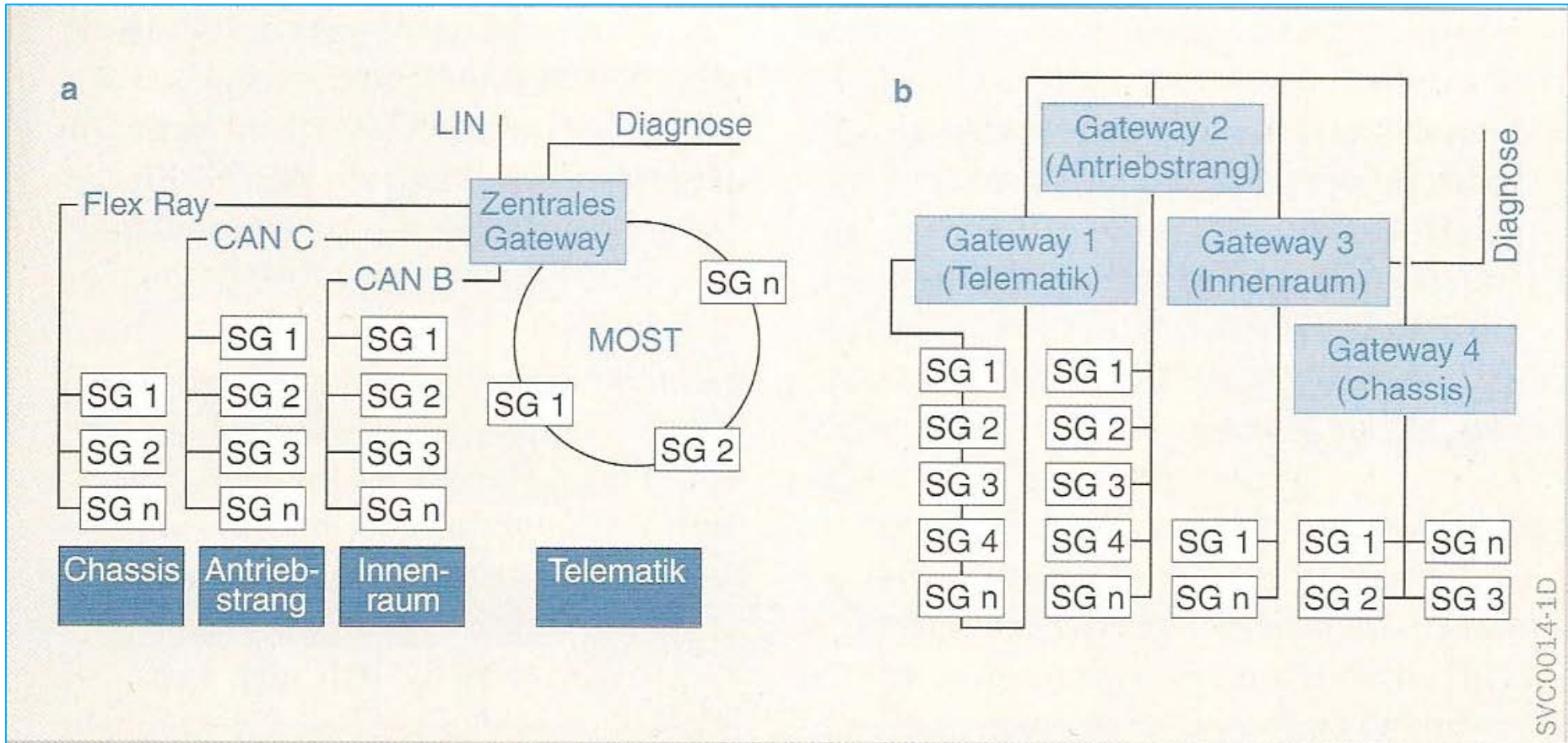
Aktuelle Situation in der Automobilindustrie

Klasse A	
Übertragungsraten	Geringe Datenraten (bis 10 kBit/s)
Anwendung	Vernetzung von Aktoren und Sensoren
Vertreter	LIN
Klasse B	
Übertragungsraten	Mittlere Datenraten (bis 125 kBit/s)
Anwendung	Komplexe Mechanismen zur Fehlerbehandlung, Vernetzung von Steuergeräten im Komfortbereich
Vertreter	Lowspeed-CAN
Klasse C	
Übertragungsraten	Hohe Datenraten (bis 1 MBit/s)
Anwendung	Echtzeitanforderungen, Vernetzung von Steuergeräten im Antriebs- und Fahrwerks- bereich
Vertreter	Higspeed-CAN

Klasse C+	
Übertragungsraten	Sehr hohe Datenraten (bis 10 MBit/s)
Anwendung	Echtzeitanforderungen, Vernetzung von Steuergeräten im Antriebs- und Fahrwerks- bereich
Vertreter	FlexRay
Klasse D	
Übertragungsraten	Sehr hohe Datenraten (> 10 MBit/s)
Anwendung	Vernetzung von Steuergeräten im Telematik- und Multimedia- bereich
Vertreter	MOST

[1, S. 19]

Aktuelle Situation in der Automobilindustrie (2)



[I, S. 21]

Ethernet - Motivation

- ▶ Bewährte Technologie
 - ▶ Computernetze, Automatisierungsindustrie, etc.
- ▶ Standardisiert
- ▶ Hat Vermittlungsintelligenz
 - ▶ Segmentierung 802.1Q
 - ▶ Priorisierung 802.1P
- ▶ Hat eine Dynamische Strukturierung
 - ▶ Unicast, Multicast und Broadcast
- ▶ Preiswert
- ▶ Datenrate: 10/100/1000/10000/100000 Mbit/s

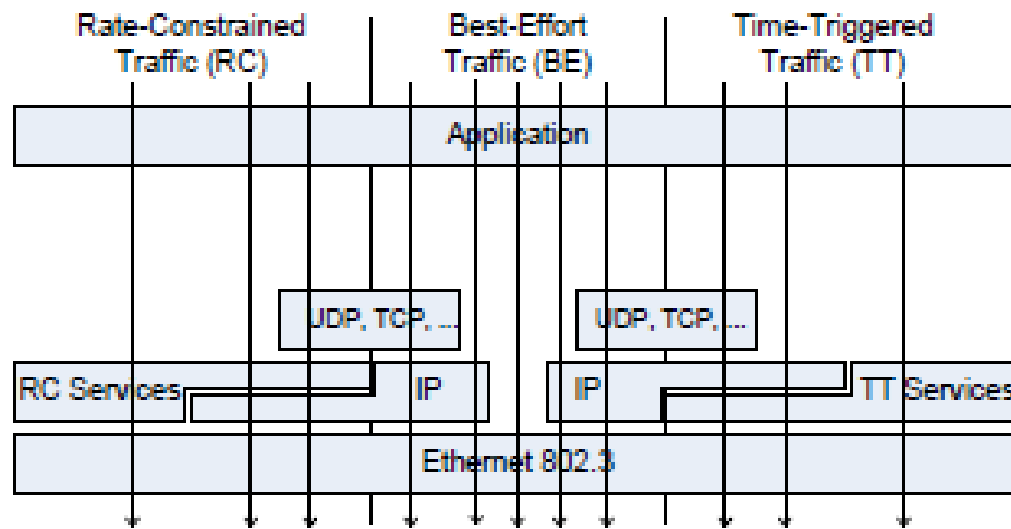
Ethernet - Problem

- ▶ Nicht echtzeitfähig
 - ▶ „Random Access“ auf Switches
 - ▶ In Switches wird das „first-come first-served“-Verfahren eingesetzt
 - ▶ Latenz und Verzögerungen in Switches

TEthernet

- ▶ TDMA
- ▶ Ethernet für Echtzeitanwendungen
- ▶ Standardkonforme Ethernet-Erweiterung
- ▶ Ausfallgesichertes Synchronisationsprotokoll
 - ▶ Zur Synchronisation der Teilnehmer mit einer Synchronisationszeit
- ▶ Zeitfenster pro Teilnehmer
- ▶ Deterministisches Zugangsverfahren

TEthernet – Traffic-Klassen



TTEthernet - Vermittlung im Automobilkontext

- ▶ Keine Realisierung, aber erste Untersuchungen
- ▶ Kaum Leistungsbewertung und Messung
- ▶ Keine Werkzeuge zur Modellierung, Simulation und Analyse

TTEthernet - Stand der Technik

- ▶ TTEthernet – TU Wien
- ▶ TTEthernet – TTTech
 - ▶ TTEthernet Spezifikation
 - ▶ Evaluation Systeme



Gegenwertige Arbeiten an der HAW

▶ Till Steinbach

▶ AW1

- ▶ Einführung TTEthernet im Automobil
- ▶ Anforderungen für Echtzeit-Bussysteme im Automobil
- ▶ Stand der Technik von Echtzeit-Ethernet

▶ AW2

- ▶ Vorstellung relevanter und verwandter Projekte
- ▶ Vorstellung relevanter Arbeitsgruppen und Konferenzen

▶ Masterarbeit

- ▶ TTEthernet-Metriken und deren simulationsbasierte Evaluierung

Gegenwertige Arbeiten an der HAW (2)

- ▶ Florian Bartols
 - ▶ Bachelorarbeit: Evaluierung eines TTEthernet-Switch (von TTTech)
 - ▶ Inbetriebnahme, Konfiguration, Werkzeuge zur Zeitmessung, Experimentierung, etc.

Simulation – Motivation

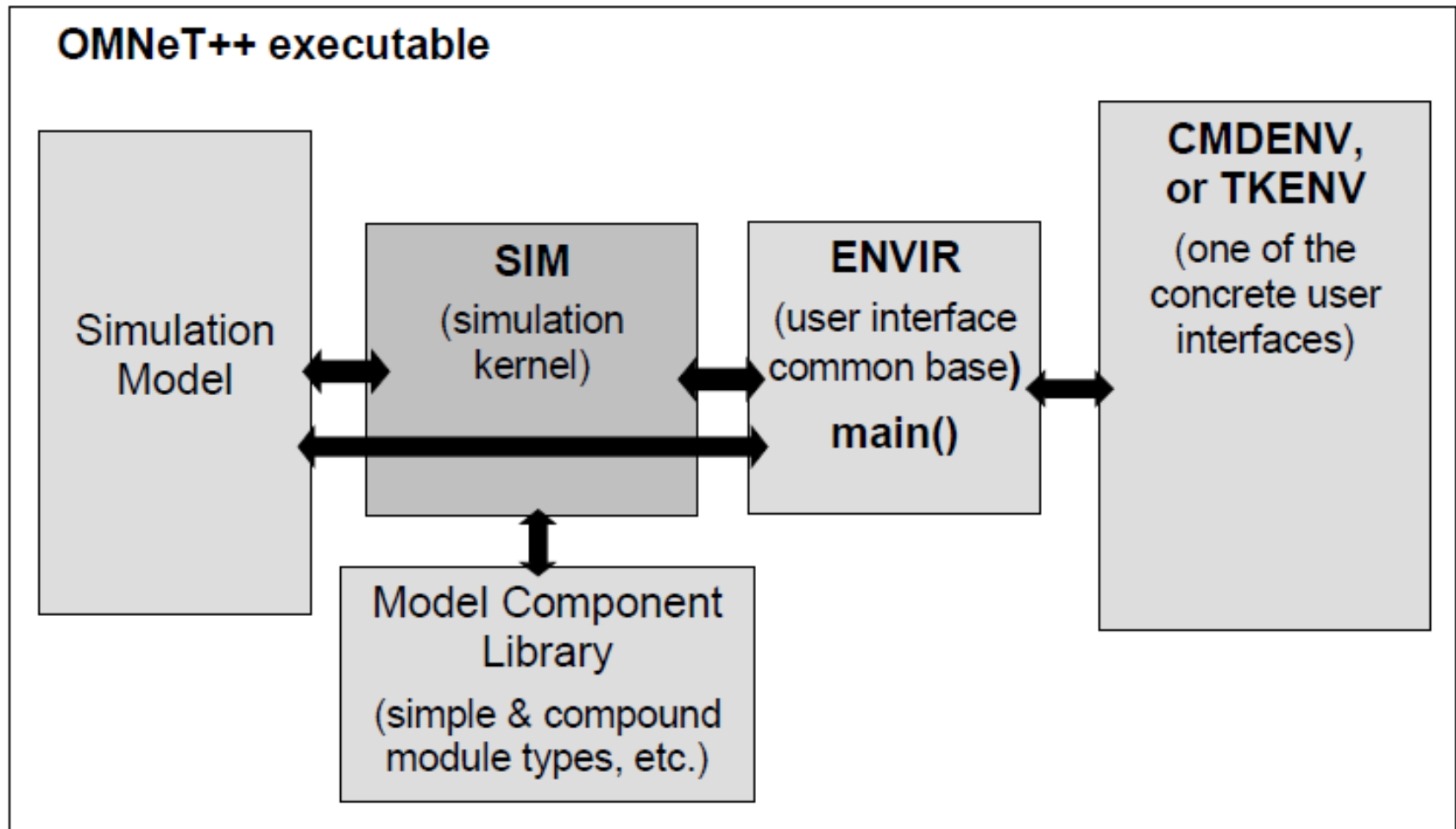
- ▶ Analyse des Netzwerks
- ▶ Experimentieren mit dem Netzwerk
- ▶ Erkenntnisgewinn über das Verhalten des Netzwerks
- ▶ Protokoll-, Topologie-, Konfigurations-Änderungen mit wenig Aufwand und Risiken
- ▶ Preiswert
- ▶ ...

OMNeT++

- ▶ **Objective Modular Network Testbed in C++**
- ▶ C++-basierter Diskret Ereignis-Simulator
- ▶ Modular und Objektorientiert
- ▶ Open-Source
- ▶ Verfügt über umfangreiche GUI-Unterstützung
- ▶ Wird eingesetzt zur
 - ▶ Modellierung von Kommunikationsnetzen
 - ▶ Modellierung von Multiprozessorsystemen
 - ▶ Modellierung von verteilten und parallelen Systemen
 - ▶ ...

[5]

OMNeT++ - Architektur



[5]

OMNeT++ - Simulationsmodell

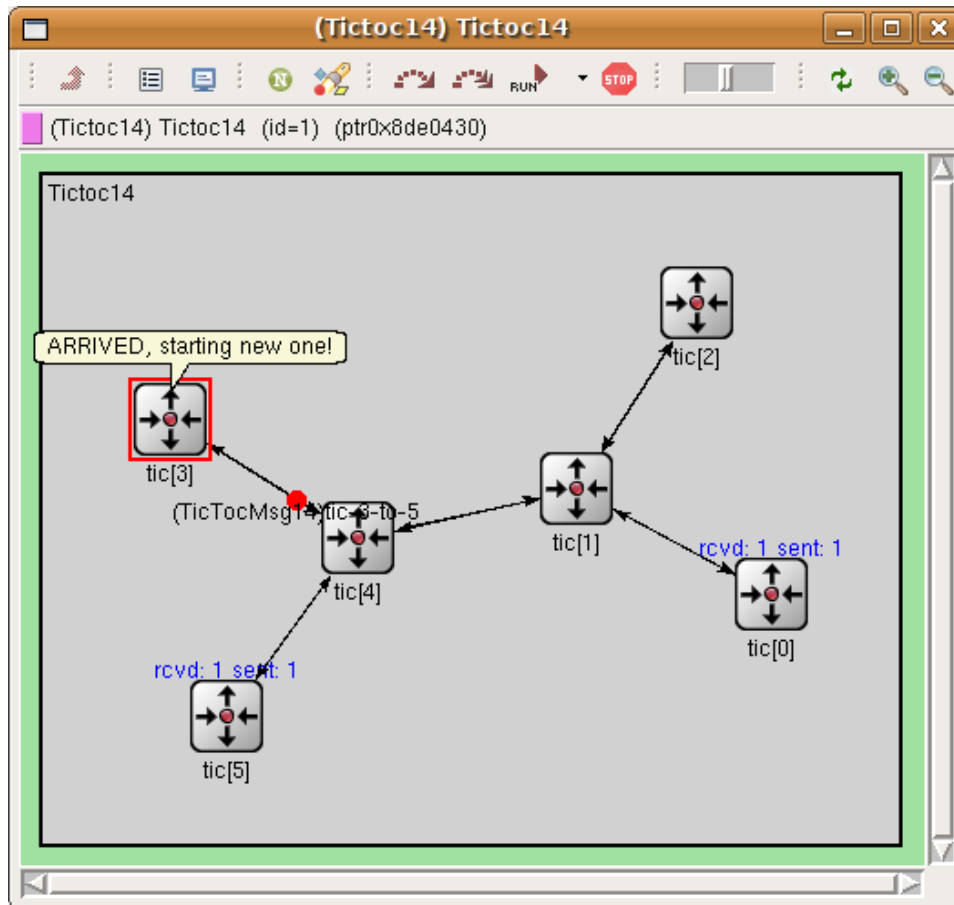
- ▶ Besteht aus hierarchisch geschachtelten Modulen
 - ▶ Einfache Module
 - ▶ Implementieren das Verhalten des Modells (C++ Funktionen)
 - ▶ Benutzen dabei Simulations-Objekte (C++ Klassen), -Methoden, etc.
 - parameters
 - gates, chanel, etc.
 - Messages, events, etc.
 - Random numbers
 - Timers
 - container classes
 - data collection classes
 - ▶ Zusammengesetzte Module

OMNeT++ - Simulationsmodelle

- ▶ INET Framework
 - ▶ UDP, TCP, SCTP, IP, IPv6, Ethernet, PPP, IEEE 802.11, MPLS, OSPF, xMIPv6 ,etc.
- ▶ INETMANET
- ▶ OverSim
- ▶ Web server simulation
- ▶ ...

OMNeT++ - Simulationsbeispiel

1. Netzwerktopologie mit NED definieren



```

simple Txc15
{
    parameters:
        @display("i=block/routing");
    gates:
        inout gate[];
}

//
// Same as Tictoc12
//
network Tictoc15
{
    types:
        channel Channel extends ned.DelayChannel {
            delay = 100ms;
        }
    submodules:
        tic[6]: Txc15;
    connections:
        tic[0].gate++ <--> Channel <--> tic[1].gate++;
        tic[1].gate++ <--> Channel <--> tic[2].gate++;
        tic[1].gate++ <--> Channel <--> tic[4].gate++;
        tic[3].gate++ <--> Channel <--> tic[4].gate++;
        tic[4].gate++ <--> Channel <--> tic[5].gate++;
}
    
```

.ned Datei

OMNeT++ - Simulationsbeispiel

2. Funktionalitäten der einfachen Modulen implementieren

```

#include <stdio.h>
#include <string.h>
#include <omnetpp.h>
#include "tictoc15_m.h"

class Txc15 : public cSimpleModule
{
private:
    long numSent;
    long numReceived;
    cLongHistogram hopCountStats;
    cOutVector hopCountVector;

protected:
    virtual TicTocMsg15 *generateMessage();
    virtual void forwardMessage(TicTocMsg15 *msg);
    virtual void initialize();
    virtual void handleMessage(cMessage *msg);

    // The finish() function is called by OMNeT++
    virtual void finish();
};

Define_Module(Txc15);

void Txc15::initialize()
{
    // Initialize variables
    numSent = 0;
    numReceived = 0;
    WATCH(numSent);
    WATCH(numReceived);

    hopCountStats.setName("hopCountStats");
    hopCountStats.setRangeAutoUpper(0, 10, 1.5);
    hopCountVector.setName("HopCount");

    // Module 0 sends the first message
    if (getIndex()==0)
    {
        // Boot the process scheduling the initial
        TicTocMsg15 *msg = generateMessage();
        scheduleAt(0.0, msg);
    }
}

```

.cc und .h Dateien

OMNeT++ - Simulationsbeispiel

2. Funktionalitäten der einfachen Modulen implementieren (2)

```

void Txc15::handleMessage(cMessage *msg)
{
    TicTocMsg15 *ttmsg = check and cast<TicTocMsg15 *>(msg);

    if (ttmsg->getDestination()==getIndex())
    {
        // Message arrived
        int hopcount = ttmsg->getHopCount();
        EV << "Message " << ttmsg << " arrived after " << hopcount << endl;
        bubble("ARRIVED, starting new one!");

        // update statistics.
        numReceived++;
        hopCountVector.record(hopcount);
        hopCountStats.collect(hopcount);

        delete ttmsg;

        // Generate another one.
        EV << "Generating another message: ";
        TicTocMsg15 *newmsg = generateMessage();
        EV << newmsg << endl;
        forwardMessage(newmsg);
        numSent++;
    }
    else
    {
        // We need to forward the message.
        forwardMessage(ttmsg);
    }
}

```

```

TicTocMsg15 *Txc15::generateMessage()
{
    // Produce source and destination addresses.
    int src = getIndex();
    int n = size();
    int dest = intuniform(0,n-2);
    if (dest>=src) dest++;

    char msgname[20];
    sprintf(msgname, "tic-%d-to-%d", src, dest);

    // Create message object and set source and destination.
    TicTocMsg15 *msg = new TicTocMsg15(msgname);
    msg->setSource(src);
    msg->setDestination(dest);
    return msg;
}

void Txc15::forwardMessage(TicTocMsg15 *msg)
{
    // Increment hop count.
    msg->setHopCount(msg->getHopCount()+1);

    // Same routing as before: random gate.
    int n = gateSize("gate");
    int k = intuniform(0,n-1);

    EV << "Forwarding message " << msg << " on gate " << k << endl;
    send(msg, "gate$o", k);
}

```

OMNeT++ - Simulationsbeispiel

3. .ini Datei erstellen

```
[Config Tictoc8]
network = Tictoc8

[Config Tictoc9]
network = Tictoc9

[Config Tictoc10]
network = Tictoc10

[Config Tictoc11]
network = Tictoc11

[Config Tictoc12]
network = Tictoc12

[Config Tictoc13]
network = Tictoc13

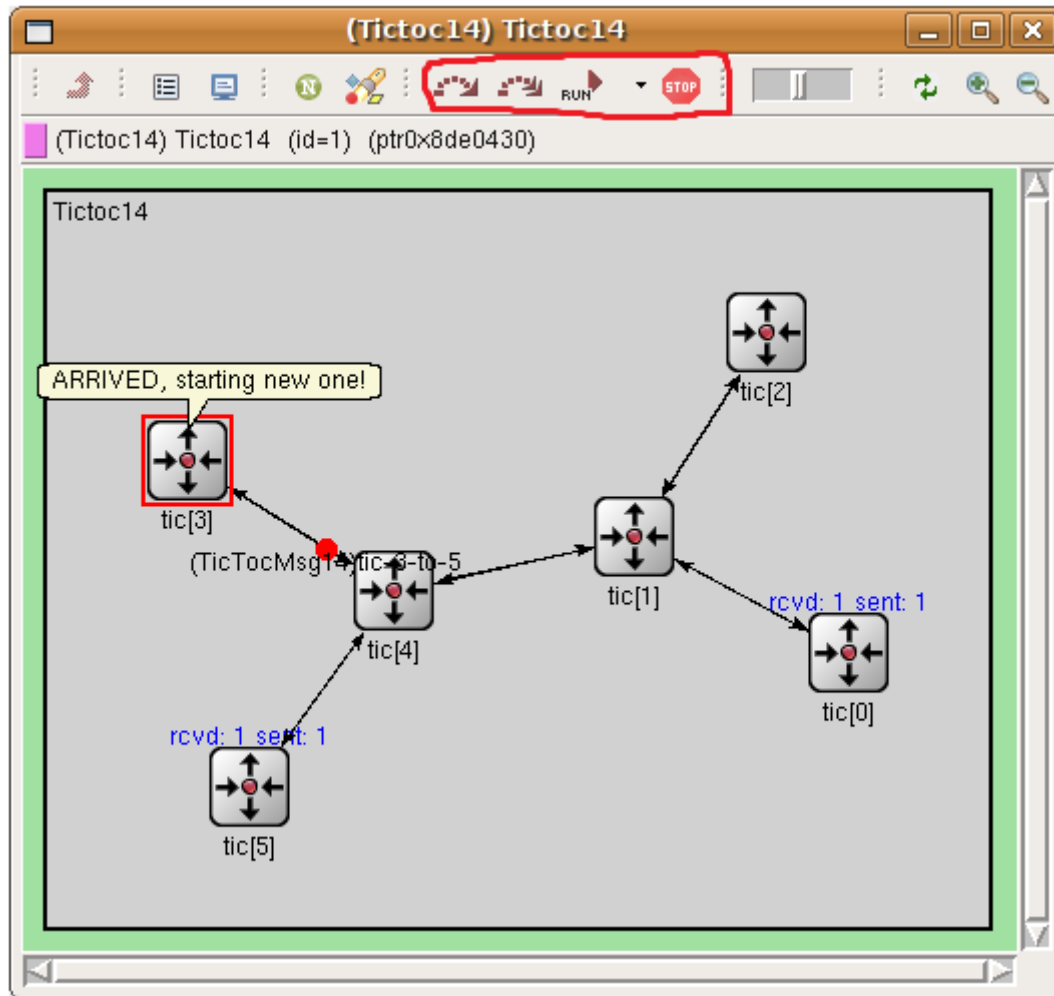
[Config Tictoc14]
network = Tictoc14

[Config Tictoc15]
network = Tictoc15
record-eventlog = true
```

omnetpp.ini Datei

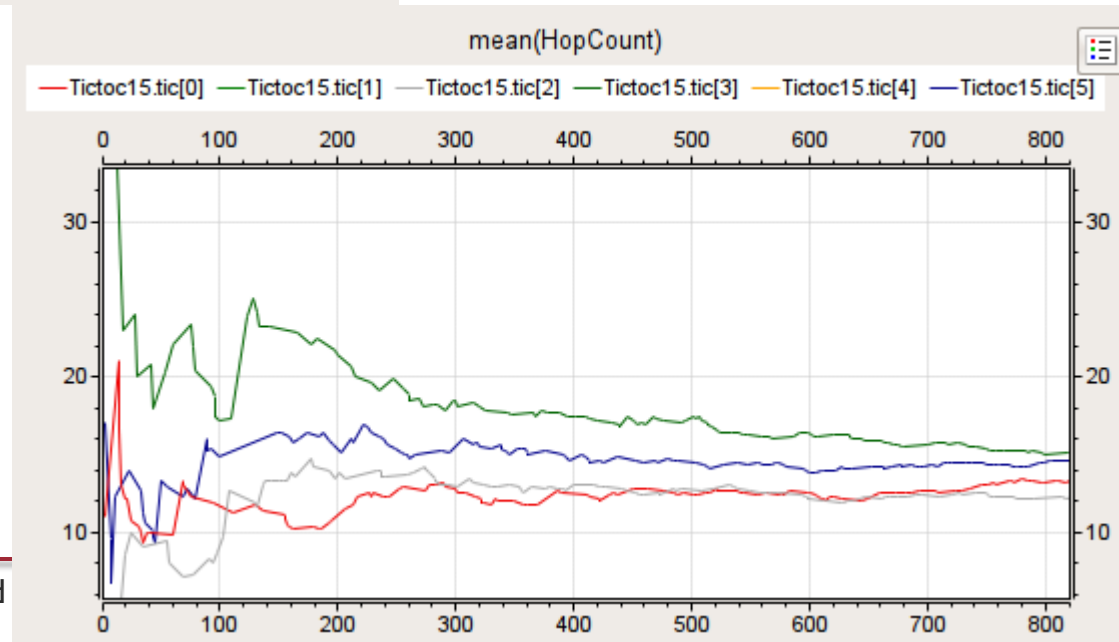
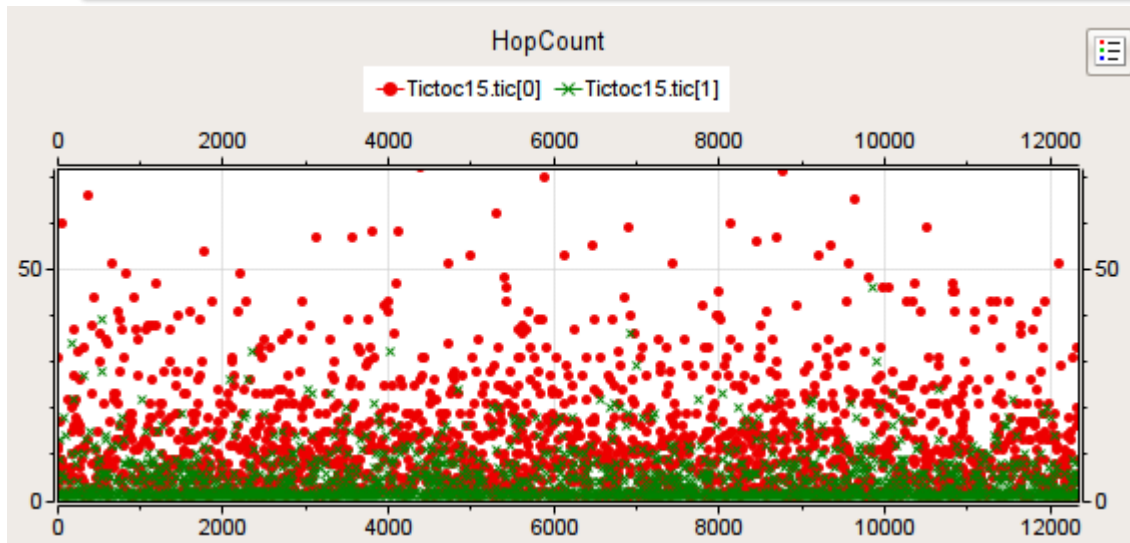
OMNeT++ - Simulationsbeispiel

4. Simulation durchführen



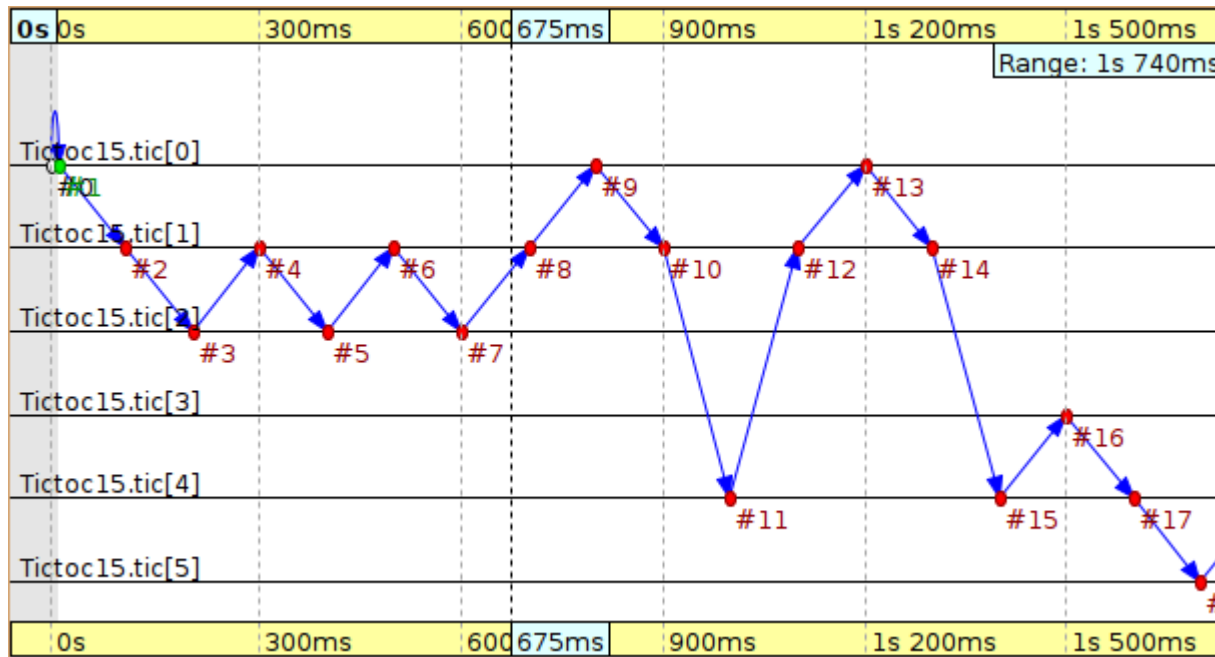
OMNeT++ - Simulationsbeispiel

5. Ergebnis analysieren



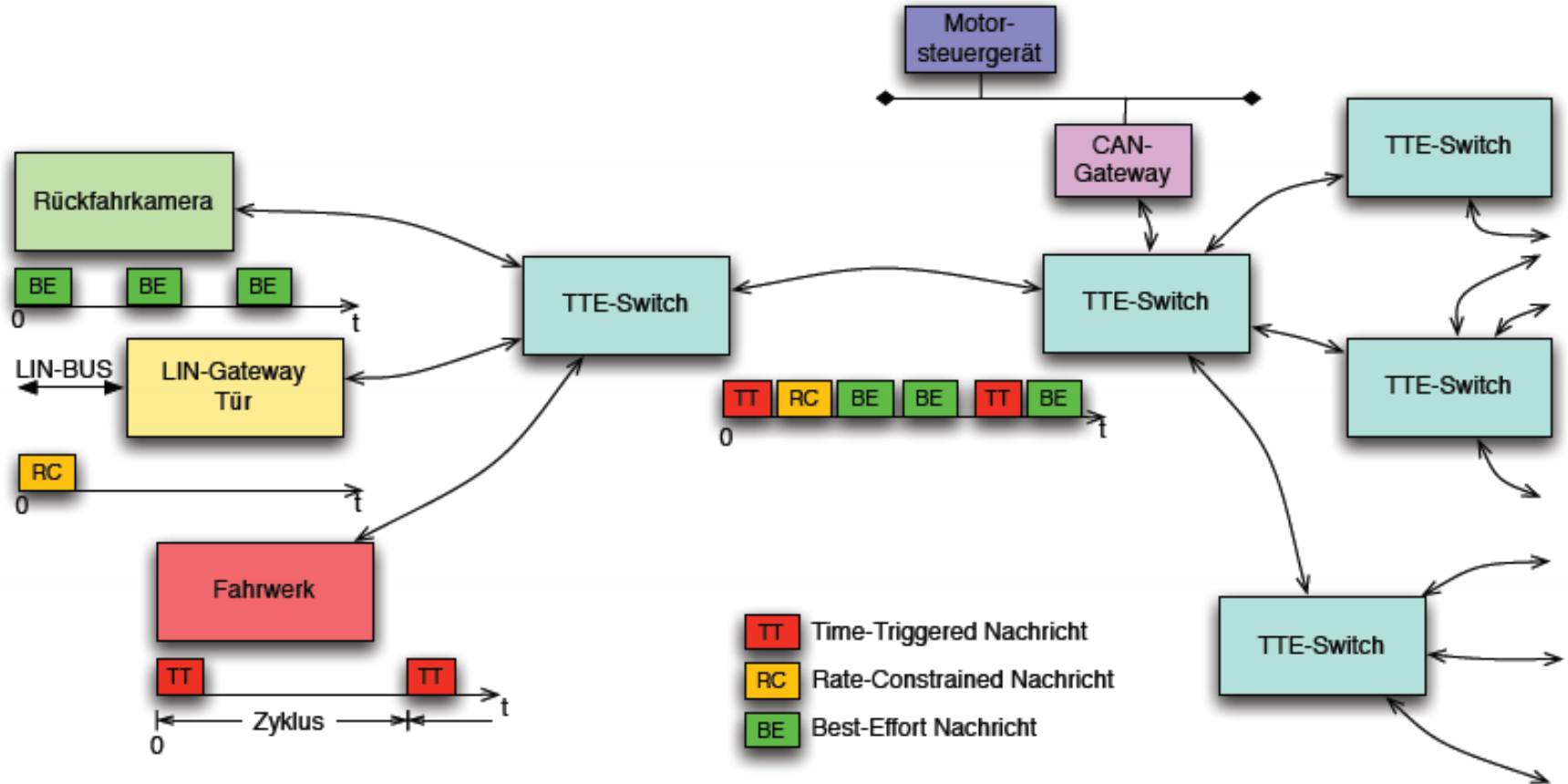
OMNeT++ - Simulationsbeispiel

5. Ergebnis analysieren (2)



Sequence Chart

Modellierung des Netzwerks - Integrationsbeispiel



Fazit und Ausblick

- ▶ Entwurf eines Modells für typische Kraftfahrzeug-Netzwerke
 - ▶ Verifizieren und Validieren
- ▶ Simulation des Modells im OMNeT++
 - ▶ Implementierung von TTEthernet-Vermittlungskomponenten
 - ▶ Simulation des Modells mit echten Topologien und charakteristischen Datenströmen
 - ▶ Analyse und Messdatenerfassung
- ▶ Implementierung eines Verkehrslast-Generators
- ▶ Konfiguration der Switches
- ▶ Dokumentation

Literatur

- ▶ [1] Mischo, Stefan; Powolny, Stefan; Zündel, Hanna; Löchel, Norbert; Stuphorn, Jörn; Constapel, Rainer; Häußermann, Peter; Leonhardi, Alexander; Holtkamp, Heiko; Robert Bosch GmbH (Hrsg.): Vernetzung im Kraftfahrzeug. Germany: 1. Ausgabe, April 2007. ISBN-13 978-3-86522-276-3
- ▶ [2] <http://www.real-time-ethernet.de/>
- ▶ [3] T. Steinbach: Ethernet als Bus für Echtzeitanwendungen im Automobil. Dec 2008, Bericht. URL- <http://papers.till-steinbach.de/s-ebea-08a.pdf>
- ▶ [4] T. Steinbach: Time-Triggered Ethernet in Fahrzeugnetzwerken: related work. Jun 2009, Bericht. URL - <http://papers.till-steinbach.de/s-ttefr-09a.pdf>
- ▶ [5] Omnet++. URL- <http://www.omnetpp.org/>
- ▶ [6] Wilfried, Steiner: TTEthernet specification. TTTech Computertechnik AG, Nov 2008. URL - <http://www.tttech.com>

Vielen Dank für ihre Aufmerksamkeit

