An example based course curriculum for Performance Evaluation in Distributed Real Time Systems

Henrik Schiøler and Thomas Kølbæk Jespersen
Aalborg University
Denmark
WoNeCA 2018
Purpose

• Theoretical framework -> Engineering Community

• user friendly tools
• co-existence with other methodologies
  • Analysis vs Simulation
• cooperation with other methodologies
  • Queueing theory vs Network Calculus
• dissemination to the educational sector
  • Course curriculum for engineering
  • Cooperation projects with industry
Course Curriculum:  
*Distributed Real Time Systems*

- Network Calculus
- Queueing theory
- Discrete event (stochastic) simulation
- (Reliability Modelling and Analysis)

- Mixed traffic on in-car network: Ethernet vs CANbus.

- RTC toolbox for MATLAB (ETH)
- CyNC toolbox for SimuLink
- OmNet++
- TrueTime for SimuLink. (Lund Uni)
- Mock-up network Arduino Teensy + FlexCAN library.

3 pillars:
- Analysis
- Simulation
- Experiments
DNC Curricula (AAU vs DISCO)

AAU
• Arrival curves: periodic (wj), spacing, affine, T-spec
• Service: Strict vs abstract, fluid, delay, rate-delay
• Theory: inf+, sup-,sub (super) additive closure
• Delay-, backlog-, output-bounds
• Prioritized ((non)pre-emptive) service.

DISCO
• MinPlusAlgebra
• Network Calculus Part I
• Network Calculus Part II
• Network Calculus Part III
• Timed Automata
Example Networks
Communication Pattern

Network elements
• 1MBps CANbus
• W{1,2,3,4}: wheel sensors measuring wheel position for ABS and EPS.
• EPS: Electronic Power Steering
• EC: Engine Controller
• HUD: Head Up Display
• MM: Multimedia system
• RC: Rear Camera

Transmission pattern
• W{1,2,3,4} -> ESP
• ESP -> EC
• MM -> HUD
• RC -> HUD
## Traffic characteristics

<table>
<thead>
<tr>
<th>Flow</th>
<th>Period / Mean period</th>
<th>Packet Size</th>
<th>Pattern</th>
</tr>
</thead>
<tbody>
<tr>
<td>W(1-4)</td>
<td>10 mS</td>
<td>20B</td>
<td>Periodic</td>
</tr>
<tr>
<td>ESP</td>
<td>10 mS</td>
<td>8B</td>
<td>Periodic</td>
</tr>
<tr>
<td>RC</td>
<td>40 mS</td>
<td>1400B</td>
<td>Poisson</td>
</tr>
<tr>
<td>MM</td>
<td>40 mS</td>
<td>1400B</td>
<td>Poisson</td>
</tr>
</tbody>
</table>
Ethernet Switch model

- Switch fabric is assumed infinitely fast
- Only store and forward delay in input queues
- When stored packets are moved instantly to output queues
- Queueing disciplines affect only output queues
- FIFO or FP scheduling
CANbus Model

- Bandwidth: 1Mbps
- CSMA/CA with prefix priority
- Non-preemptive Fixed Priority Scheduling
- Non-preemptive -> Head of Line Blocking (HoL)
Cyclic Network Calculus (CyNC) a toolbox for MATLAB SimuLink
Cyclic Network Calculus (CyNC)

- Cyclic dependence
- Cyclic flow patterns
- Service/flow counter propagation
RTC/CyNC implementation of CANBus FP scheduling

\[
a_{\text{in}} = \text{rtcpjd}(3, 0, 0);
b_{\text{in}} = \text{rtctdma}(2, 7, 4);
ed = 3;
[a_{\text{out}} \ b_{\text{out}} \ \text{del} \ \text{buf}] = \text{rtcgpc}(a_{\text{in}}, b_{\text{in}}, ed);
\]
RTC/CyNC implementation of EtherNet FIFO scheduling

a1\_in = rtcpjd(3, 0, 0); a2\_in = rtcpjd(7, 0, 0);
b\_in = rtctdma(3, 7, 4);
ed1 = 3; ed2 = 1;
[a1\_out del1 buf1 a2\_out del2 buf2 b\_out] = …
rtcfifo(a1\_in, ed1, a2\_in, ed2, b\_in);
Aperiodic Streams MM and RC

- Modeled as Poisson arrivals !?
- Filtered through token/leaky buckets
- One token – one packet

Separation of Domains

\[ \alpha(t) = t \times \frac{P}{T} + M \times P \]

Probabilistic /stochastic domain

Non-deterministic domain
Delays and Backlogs

\[ d_{RC} = 111,984 \text{ ms} \]
\[ \text{backlog}_{RC} = 5 \cdot P = 96,250 \]

\[ d_{MM} = 266,2640 \text{ ms} \]
\[ \text{backlog}_{MM} = 113,056 \]
\[ > 2 \cdot P = 38,500 \]
Probabilistic Analysis

- Mean backlogs
- Mean waiting times
- Packet loss probabilities
- $M/M/1/L+M$ queueing system with warping
  - Erlang B formula: $\pi_n = (\lambda T)^n / (n! (\sum_i (\lambda T)^i / i!))$
  - $P_{\text{loss}} = \pi_{L+M}$
- Discrete Time embedded Markov chain with warping
Discrete Time Embedded Markov Chain for Periodic Transfer

\[ H = \begin{bmatrix} A_0 + A_1 & A_2 & A_3 & A_4 & \cdots & \cdots & A_{L+M} & 1 - (A_0 + A_1 + \cdots + A_{L+M}) \\ A_0 & A_1 & A_2 & A_3 & \cdots & \cdots & A_{L+M-1} & 1 - (A_0 + A_1 + \cdots + A_{L+M-1}) \\ 0 & A_0 & A_1 & A_2 & A_3 & \cdots & A_{L+M-2} & 1 - (A_0 + A_1 + \cdots + A_{L+M-2}) \\ 0 & 0 & A_0 & A_1 & A_2 & \cdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & A_0 & 1 - A_0 \end{bmatrix} \]
Probabilistic Analysis

Warping

\[ \pi'_0 = \pi_0 + \pi_1 + \ldots + \pi_M \]
\[ \pi'_n = \pi_{n+M} \]
\[ P_{\text{loss}} = \pi_{L+M} = \pi'_L \]
\[ Q = \sum_i i \pi'_i \]
\[ W = (1 - P_{\text{loss}})Q / \lambda \]

- \( P_{\text{loss}} \) depends only on \( L+M \)
- Higher \( L \) more waiting less \( M \) (burst)
- Lower \( L \) less waiting higher \( M \) (burst)
Simulation with TrueTime

1 Schedule
TrueTime Network (CAN 1)
11-bit identifier
1 Mbit datarate

1:5
TrueTime Kernel
Periodic transmitter
Electronic Stability Controller

1:6 DIA
TrueTime Kernel
Receiver
Electronic Stability Controller

1:8 DIA
TrueTime Kernel
Receiver
Engine Controller

1:12 DIA
TrueTime Kernel
Receiver
HUD

W(1)
W(2)
W(3)
W(4)
Package delay for Wheel packages

W
Package delay for RC and MM

delays

1:10 DIA
TrueTime Kernel
Poisson process with Token Bucket Filter
For Rear camera

RC
Tokens
Queue

MM
Tokens
Queue

RC

Token bucket RC

RC

Package statistics RC

RC

Token bucket MM

RC

Package statistics MM

MM

Package drop statistics

MM

Token bucket MM

MM

Package statistics MM

MM

Package drop statistics
The TT kernel

function generator_init

% Initialize TrueTime kernel
ttInitKernel('prioFP'); % scheduling policy - fixed priority

starttime = 0.0;
% Poisson generator task
ttCreateTask('generator_task', starttime, 'generator_code');
%First job
ttCreateJob('generator_task', ttCurrentTime)
The TT task

function [exectime, data] = generator_code(seg)

%independent exponentially distributed inter arrival time
lambda=10; %intensity parameter
u=rand();
T=-log(1-u)/lambda; %inverse fct method

%Send message on CANbus interface with highest priority to node ID 4
priority = 0;
msg = [ttCurrentTime]; % message with timestamp for E2E delay statistics
ttSendMsg([14], msg, 250, priority);

%order next transmission
ttCreateJob('generator_task',ttCurrentTime+T)
exectime = -1; %job done no CPU resources used
TT Backlog and Tokens
TT CANbus Network Schedules
TT E2E delays
CANbus Test Delays

Delay

The mean package delays from the M/P/1 queueing model were calculated to:

$\tilde{W}_{RC} = 98.3 \text{ ms}$

$\tilde{W}_{MM} = 356.8 \text{ ms}$
Loss Probabilities

The loss probabilities from the M/P/1 queueing model were calculated to:

\[
\pi_{Rc} = 23.31\%
\]

\[
\pi_{MM} = 22.27\%
\]
CANbus Delays - MM