Optimizing Network Calculus for Switched Ethernet Network with Deficit Round Robin

Aakash SONI$^{1,2}$  
Jean-Luc SCHARBARG$^1$  
Xiaoting LI$^2$  
Christian FRABOUL$^1$

$^1$IRIT Toulouse  
$^2$ECE Paris

aakash.soni@irit.fr

STORE - 2019

February 18, 2019
Context: Evolution of Avionic Network

- **Traditional aircraft network ARINC 429. (Airbus A320)**
  - Mono-transmitter buses with limited performances (100 Kbits/s).
- **Avionics Full DupleX (AFDX) network.**
  - Switched Ethernet ARINC 664. (Airbus A380)
  - A backbone network for the avionics platform.
  - 100 Mbps.
  - FIFO/SP queues.
Context: The Problem

- **ARINC 664:** Indeterminism at Switch level.
  - Competition for the use of the resources.
    - Congestion = frame losses
    - Frame storage in queues = Latency and jitter.
  - Need of guaranteed bounds for certification.
Context: The Problem

- Inefficient use of available bandwidth.\[1\]
  - Lightly loaded network (up to 10% only).
  - Possibility to share bandwidth among critical (avionic) and non-critical flow.
  - Example:
    > Audio message from cockpit to cabin.
    > Parking video.

- Solution: Quality of Service (QoS) mechanism.

How to make better use of available bandwidth?

- **QoS:** Share bandwidth using Round Robin Scheduler.
  - **Deficit Round Robin (DRR) scheduling.**
  - **Weighted Round Robin (WRR) scheduling.**[1]

Performance analysis of DRR and WRR scheduler in AFDX network.

- Worst-case end-to-end delay.
- Fairness.
Context: Objective 3

Improve delay bound computation.

- Reduce pessimism in analysis approach.

![Diagram showing worst-case delay, sure upper bound of worst-case delay, and pessimism.](image-url)
1 Context
   - Objectives
   - Switched Ethernet Network
   - DRR Algorithm
   - Network Calculus

2 Contribution
   - Optimization
   - DRR vs WRR

3 Conclusion
Table of Contents

1. Context
   - Objectives
   - Switched Ethernet Network
   - DRR Algorithm
   - Network Calculus

2. Contribution
   - Optimization
   - DRR vs WRR

3. Conclusion
Switched Ethernet Network: AFDX Network Model

AFDX network model

- End-Systems ($e_x$)
- Switches ($S_x$)
- FIFO output ports
- Statically defined flows.

![AFDX Network Model Diagram]

Existing AFDX Model

AFDX Perspective
Avionic flows are characterized as virtual links;

- Statically defined: predictable deterministic behavior.
- Maximum frame length: $S_{max}$
- Minimum delay between two consecutive frames: $BAG$ (Bandwidth Allocation Gap)
- Multi-cast routing
Table of Contents

1. Context
   - Objectives
   - Switched Ethernet Network
   - DRR Algorithm
   - Network Calculus

2. Contribution
   - Optimization
   - DRR vs WRR

3. Conclusion
The DRR Algorithm

DRR

Active flow buffers

\begin{array}{|c|c|}
\hline
\text{bytes} & \\
40 & C_1 \\
30 & v_2 \\
20 & v_3 \\
10 & v_4 \\
0 & v_5 \\
\hline
\end{array}

\begin{array}{|c|c|}
\hline
\text{bytes} & \\
40 & C_2 \\
30 & v_5 \\
20 & v_6 \\
10 & v_7 \\
0 & \\
\hline
\end{array}

\[ Q = 20 \]

\[ \Delta : \text{Deficit} \]

\[ Credit = Q + \Delta \]

\[
\begin{array}{|c|c|c|c|}
\hline
Q_{C_1} & \Delta_{C_1} & Q_{C_2} & \Delta_{C_2} \\
\hline
\end{array}
\]
DRR Algorithm

Active flow buffers

\[ Q = 20 \]
\[ \Delta : \text{Deficit} \]
\[ \text{Credit} = Q + \Delta \]

<table>
<thead>
<tr>
<th>(Q_{C_1})</th>
<th>(\Delta_{C_1})</th>
<th>(Q_{C_2})</th>
<th>(\Delta_{C_2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
DRR Algorithm

Active flow buffers

bytes

40
30
20
10
0

C₁
C₂

v₁
v₂
v₅
v₆
v₇

Q = 20

Δ : Deficit

Credit = Q + Δ

<table>
<thead>
<tr>
<th>Q_{C₁}</th>
<th>Δ_{C₁}</th>
<th>Q_{C₂}</th>
<th>Δ_{C₂}</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**DRR Algorithm**

Active flow buffers

![Diagram of DRR algorithm](image)

<table>
<thead>
<tr>
<th></th>
<th>(Q_{C_1})</th>
<th>(\Delta_{C_1})</th>
<th>(Q_{C_2})</th>
<th>(\Delta_{C_2})</th>
</tr>
</thead>
<tbody>
<tr>
<td>v3</td>
<td>20</td>
<td>0</td>
<td>20</td>
<td></td>
</tr>
</tbody>
</table>

\(Q = 20\)

\(\Delta : Deficit\)

\(Credit = Q + \Delta\)
DRR Algorithm

Active flow buffers

bytes
40
30
20
10
0

$C_1$
$v_1$
$v_2$

$C_2$
$v_5$
$v_6$

$Q = 20$

$\Delta : Deficit$

$Credit = Q + \Delta$

<table>
<thead>
<tr>
<th>$QC_1$</th>
<th>$\Delta C_1$</th>
<th>$QC_2$</th>
<th>$\Delta C_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0</td>
<td>20</td>
<td>10</td>
</tr>
</tbody>
</table>
DRR Algorithm

Active flow buffers

bytes

40
30
20
10
0

$v_1$
$v_2$
$v_5$
$v_6$

$C_1$

$C_2$

$Q = 20$

$\Delta : Deficit$

$Credit = Q + \Delta$

<table>
<thead>
<tr>
<th>$Q_{C_1}$</th>
<th>$\Delta_{C_1}$</th>
<th>$Q_{C_2}$</th>
<th>$\Delta_{C_2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0</td>
<td>20</td>
<td>10</td>
</tr>
</tbody>
</table>
DRR Algorithm

\[ Q = 20 \]
\[ \Delta : Deficit \]
\[ Credit = Q + \Delta \]

<table>
<thead>
<tr>
<th>( Q_{C_1} )</th>
<th>( \Delta_{C_1} )</th>
<th>( Q_{C_2} )</th>
<th>( \Delta_{C_2} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>20</td>
<td>10</td>
</tr>
</tbody>
</table>
DRR Algorithm

\[ Q = 20 \]
\[ \Delta : \text{Deficit} \]

\[ \text{Credit} = Q + \Delta \]

\begin{array}{|c|c|c|c|}
\hline
Q_{C_1} & \Delta_{C_1} & Q_{C_2} & \Delta_{C_2} \\
\hline
20 & 0 & 20 & 10 \\
20 & 0 & 20 & 10 \\
\hline
\end{array}
DRR Algorithm

$Q = 20$

$\Delta : Deficit$

$Credit = Q + \Delta$

<table>
<thead>
<tr>
<th>$Q_{C1}$</th>
<th>$\Delta_{C1}$</th>
<th>$Q_{C2}$</th>
<th>$\Delta_{C2}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>0</td>
<td>20</td>
<td>10</td>
</tr>
<tr>
<td>20</td>
<td>0</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>
Table of Contents

1 Context
   ■ Objectives
   ■ Switched Ethernet Network
   ■ DRR Algorithm
   ■ Network Calculus

2 Contribution
   ■ Optimization
   ■ DRR vs WRR

3 Conclusion
Network Calculus

Network calculus

- Computes **upper bounds** on:
  - End-to-end delay.
  - Jitter.

- **Pessimism:** models network based on **traffic envelops**.
  - Overestimated flow traffic.
  - Underestimated network service.
Network Calculus: Traffic Envelops

![Diagram of End System, Network (DRR), End System]
Network Calculus: Traffic Envelops: Arrival Traffic

![Diagram showing network system with end systems and network (DRR) and actual arrival flow.]
Network Calculus: Traffic Envelops: Arrival Curve

End System \[ \rightarrow \] Network (DRR) \[ \rightarrow \] End System

[Diagram showing a flow of data with variables \( S_{\text{max}} \), \( \alpha_{C_1}^h \), and \( \text{BAG} \).]

Aakash SONI
Optimizing NC for Networks with DRR 25/42
Network Calculus: Traffic Envelops: Network Service

Actual Network Service

$S_{max}$

$\alpha^h_{C_1}$

$t$ (µsec)

End System

Network (DRR)

End System

Optimizing NC for Networks with DRR 26/42
Network Calculus: Traffic Envelops: Delay

\[ t (\mu sec) \]

\[ \text{bits} \]

\[ S_{\text{max}} \]

\[ \Theta_{C_1}^h \]

\[ \rho_{C_1}^h \]

\[ \alpha_{C_1}^h \]

\[ \beta_{C_1}^h \]

Network (DRR)
Network Calculus: Traffic Envelops: Optimization

End System → Network (DRR) → End System

\[ t (\mu s) \]

\[ \text{bits} \]

\[ S_{\text{max}} \]

\[ \Theta_{C_1}^h \]

\[ \rho_{C_1}^h \]

\[ \alpha_{C_1}^h \]

\[ \beta_{C_1}^h \]
Network Calculus: Traffic Envelops: Optimization

![Diagram of network system with End System, Network (DRR), and End System]

- \( \alpha^h_{C_1} \)
- \( \beta^h_{C_1} \)
- \( S_{max} \)
- \( \Theta^h_{C_1} \)
- \( \rho^h_{C_1} \)
- \( t (\musec) \)
- \( \text{delay} \)
- \( \text{delay}' \)
- \( \text{BAG} \)

Aakash SONI

Optimizing NC for Networks with DRR
# Table of Contents

1. Context
   - Objectives
   - Switched Ethernet Network
   - DRR Algorithm
   - Network Calculus

2. Contribution
   - Optimization
   - DRR vs WRR

3. Conclusion
Table of Contents

1. Context
   ■ Objectives
   ■ Switched Ethernet Network
   ■ DRR Algorithm
   ■ Network Calculus

2. Contribution
   ■ Optimization
   ■ DRR vs WRR

3. Conclusion
Pessimism in computed network service

```
C_1: v6, v5, v4, v3, v2, v1
C_2: v12, v11, v10, v9, v8, v7
C_3: v18, v17, v16, v15, v14, v13

\begin{align*}
\Theta^{h}_{C_1} & & \alpha^{h}_{C_1} & & \beta^{h}_{C_1} \\
v_1 & & v_2, v_3 & & v_{4, v_5, v_6} \\
v_7, v_8, v_13, v_14 & & v_9, v_{10}, v_{15}, v_{16} & & v_{11}, v_{12}, v_{17}, v_{18}
\end{align*}
```

\[ D^{h}_{C_1} \text{ (\(\mu\)sec)} \]
Pessimism in computed network service

C1: v6 v5 v4 v3 v2 v1
C2: v12 v11 v10 v9 v8 v7
C3: v18 v17 v16 v15 v14 v13

DRR

C1: C2: C3:

v7 v8 v13 v14

v9 v10 v15

v11

v12 v17 v18

v11 v12 v17 v18

v2, v3

v1

v2, v3

v1

v13 v14 v15 v16

v7 v8 v13 v14

v9 v10

v11

v15 v14 v13

Aakash SONI

Optimizing NC for Networks with DRR
Upper bound on interfering traffic

\begin{align*}
C_2 \text{ Flow traffic in } \beta^h_{C_1} \quad &- \quad \text{Actual traffic from } C_2 \text{ before } D^h_{C_1} = \text{Extraload}_{C_2} \\
C_3 \text{ Flow traffic in } \beta^h_{C_1} \quad &- \quad \text{Actual traffic from } C_3 \text{ before } D^h_{C_1} = \text{Extraload}_{C_3}
\end{align*}

Aakash SONI  
Optimizing NC for Networks with DRR  
35/42
More accurate delay computation

C_2 Flow traffic in \( \beta_{C_1} \) - Actual traffic from C_2 before \( D_{C_1}^h \) = \( \text{Extraload}_{C_2} \)

\[ (\alpha_{C_2}^h(D_{C_1})) \]

C_3 Flow traffic in \( \beta_{C_1} \) - Actual traffic from C_3 before \( D_{C_1}^h \) = \( \text{Extraload}_{C_3} \)

\[ (\alpha_{C_3}^h(D_{C_1})) \]

\[ D_{C_1}^{\text{opt},h} = D_{C_1}^h - tr\{\text{Extraload}_{C_2} + \text{Extraload}_{C_3}\} \]
**Evaluation**

- Airbus A350 configuration
- 984 flows, 96 end systems, 8 switches, 6412 paths

**Average gain:** 48%.

**Max gain 75%.

<table>
<thead>
<tr>
<th>Class</th>
<th>Flows</th>
<th>$S_{max}$ (byte)</th>
<th>$Q_x$</th>
<th>Category</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>718</td>
<td>475</td>
<td>$2 \times L_{max}$</td>
<td>Critical</td>
</tr>
<tr>
<td>$C_2$</td>
<td>194</td>
<td>971</td>
<td>$L_{max}$</td>
<td>Multimedia</td>
</tr>
<tr>
<td>$C_3$</td>
<td>72</td>
<td>1535</td>
<td>$L_{max}$</td>
<td>Best-effort</td>
</tr>
</tbody>
</table>
Table of Contents

1 Context
   • Objectives
   • Switched Ethernet Network
   • DRR Algorithm
   • Network Calculus

2 Contribution
   • Optimization
   • DRR vs WRR

3 Conclusion
Performance Analysis

<table>
<thead>
<tr>
<th>class</th>
<th>No. of flows</th>
<th>DRR Quantum (bytes)</th>
<th>WRR weight (no. of packets)</th>
<th>frame size range</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_1$</td>
<td>718</td>
<td>$4 \times l_{max}$</td>
<td>4</td>
<td>415-475</td>
</tr>
<tr>
<td>$C_2$</td>
<td>194</td>
<td>$2 \times l_{max}$</td>
<td>2</td>
<td>911-971</td>
</tr>
<tr>
<td>$C_3$</td>
<td>72</td>
<td>$1 \times l_{max}$</td>
<td>1</td>
<td>1475-1535</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>class</th>
<th>DRR vs WRR</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>avg difference (%)</td>
</tr>
<tr>
<td>$C_1$</td>
<td>29.16</td>
</tr>
<tr>
<td>$C_2$</td>
<td>29.6</td>
</tr>
<tr>
<td>$C_3$</td>
<td>-35.4</td>
</tr>
</tbody>
</table>
Table of Contents

1 Context
   - Objectives
   - Switched Ethernet Network
   - DRR Algorithm
   - Network Calculus

2 Contribution
   - Optimization
   - DRR vs WRR

3 Conclusion
Conclusion

- NC on AFDX network with mixed criticality
- QoS: DRR scheduling.
- Evaluation of improved NC approach.
- Performance comparison of DRR and WRR schedulers.
- What’s next?
  - Exact worst case delay using model checking approach.
    - Classical MC Approach => upto 32 flows
    - Improved Approach => 300+ flows
  - Evaluation of pessimism of NC for avionic network with DRR and WRR scheduler.
  - Weight/Quantum allocation in Round Robin scheduler (WRR/DRR)
Thank you for your attention!

Aakash SONI - aakash.soni@irit.fr