Combining static priority and weighted round-robin like packet scheduling in AFDX for incremental certification and mixed-criticality support

Marc Boyer[1], Nicolas Navet[2], Marc Fumey[3], Jörn Migge[4], Lionel Havet[4]

[1] ONERA – The French aerospace
[2] University of Luxembourg, Luxembourg
[3] Thales Avionics, Toulouse, France
[4] RealTime-at-Work, Villers-lès-Nancy, France

5th EUropean Conference for Aeronautics and Qpace Sciences (EUCASS 2013)

ONERA
THE FRENCH AEROSPACE LAB

r e t o u r s u r i n n o v a t i o n

Outline

AFDX backbone

Network partitioning

Deficit Round Robin

RTaW-PEGASE AFDX network analyser

Applicability of DRR for AFDX network

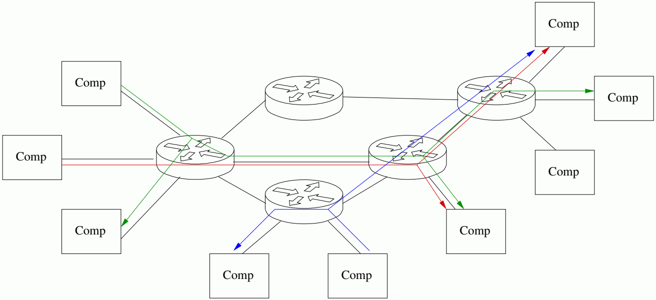
Applicability of SP/DRR for AFDX network

Conclusion

Context: AFDX backbone

Real-Time distributed systems:

- AFDX ≈ Ethernet technology for avionics
- ≈ hundred of computers
- 8 switches
- ≈ thousands of data flows



ONERA
THE FRENCH AEROSPACE LAB

Requirement: Worst Case Traversal Time

Requirement: bound on network latency

- Input contract traffic: Virtual Link
  - Static routing
  - Maximal frame size l^m
  - Minimal interval between two frames *BAG*
- Knowledge on switches:
  - bandwidth
  - scheduling policy: FIFO, Static Priority
- Analyse method
  - Network calculus
  - Trajectorial approach
  - Ad-hoc methods..

Outline

AFDX backbone

Network partitioning

Deficit Round Robin

RTaW-PEGASE AFDX network analyser

Applicability of DRR for AFDX network

Applicability of SP/DRR for AFDX network

Conclusion

Design challenge: system independence

Network: shared resource

- performance of one flow depends on all other flows
- hundred of systems (and system providers)
- need of early network design
- avoid "frequent" re-design / provisioning

Network independence:

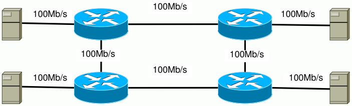
- Virtual Link: a mid-grained independence
  - allows frame content change
  - allows frame size reduction
  - allows period change ($\geq BAG$)
  - $\approx 10^4$ VLs
- A more coarse-grained partitioning ?

Network partitioning

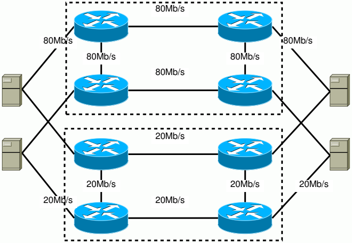
Challenge: cut the net into virtual independent partitions

- 4-10 partitions (coarse-grained)
- independent performances

One physical 100Mb/s network



Two virtual 20Mb/s 80Mb/s networks



Generalised Processor Sharing

GPS: Generalised Processor Sharing

- cut the service into n classes
- allocate a quantum q_i to each class
- each class receives fraction $\frac{q_i}{\sum_{i=1}^{n} q_i}$ of service
- ideal policy
- Packetised-GPS (P-GPS):
  - practical implementation of GPS
  - implement GPS "up to one packet size"
  - can hardly be implemented in real-time
- Deficit Round Robin (DRR)
  - other GPS implementation
  - efficient implementation

Outline

AFDX backbone

Network partitioning

Deficit Round Robin

RTaW-PEGASE AFDX network analyser

Applicability of DRR for AFDX network

Applicability of SP/DRR for AFDX network

Conclusion

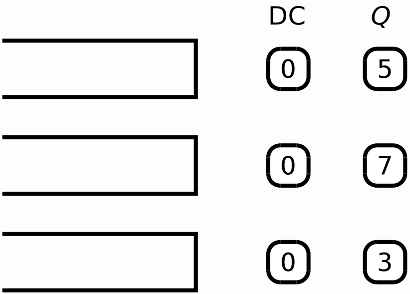
DRR idea

Deficit Round Robin:

- GPS implementation
- $O(1)$ implementation
- allocate one quantum Q_i per class
- one queue per class
- infinite loop: for each class/queue
  - increment credit by Q_i if active
  - sending packet of size s decrease credit by s
  - send packets as long as non null credit

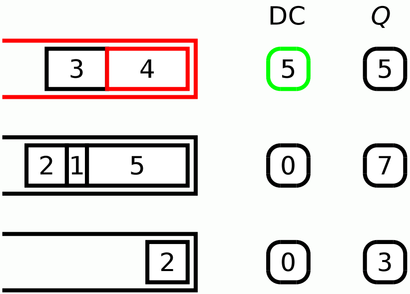
DRR run example

- 3 classes
- Quanta Q: 5, 7, 3
- CD: Deficit Counter



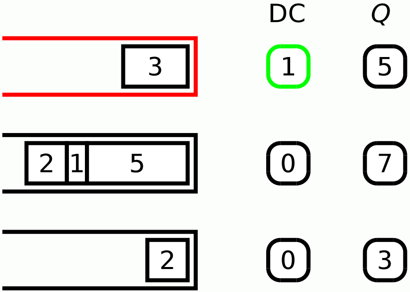
DRR run example

- 3 classes
- Quanta Q: $5, 7, 3$
- CD: Deficit Counter



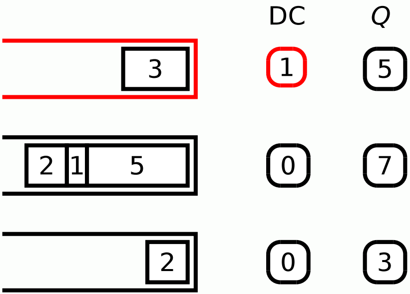
DRR run example

- 3 classes
- Quanta Q: $5, 7, 3$
- CD: Deficit Counter



DRR run example

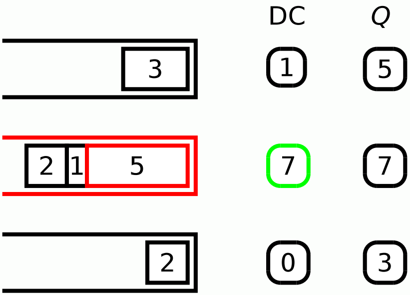
- 3 classes
- Quanta Q: $5, 7, 3$
- CD: Deficit Counter



ONERA
THE FRENCH AEROSPACE LAB

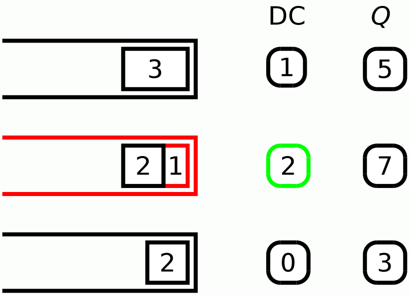
DRR run example

- 3 classes
- Quanta Q: 5, 7, 3
- CD: Deficit Counter



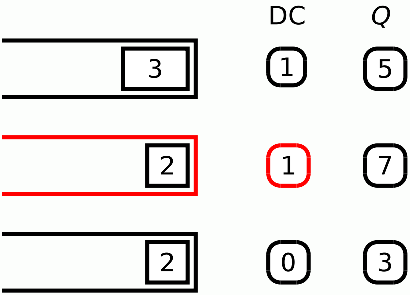
DRR run example

- 3 classes
- Quanta Q: $5, 7, 3$
- CD: Deficit Counter



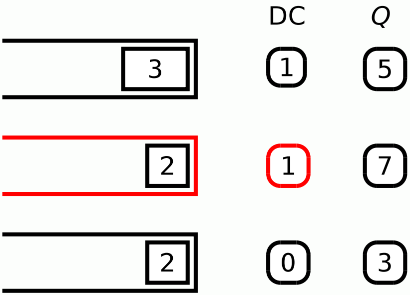
DRR run example

- 3 classes
- Quanta Q: $5, 7, 3$
- CD: Deficit Counter



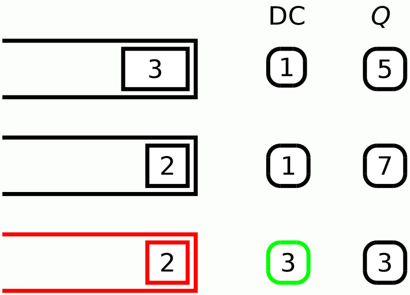
DRR run example

- 3 classes
- Quanta Q: $5, 7, 3$
- CD: Deficit Counter



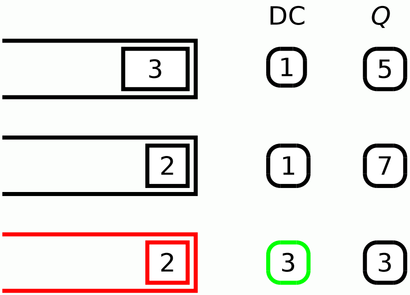
DRR run example

- 3 classes
- Quanta Q: $5, 7, 3$
- CD: Deficit Counter



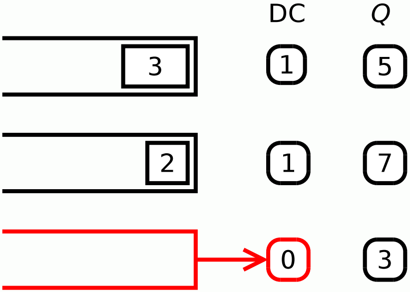
DRR run example

- 3 classes
- Quanta Q: 5, 7, 3
- CD: Deficit Counter



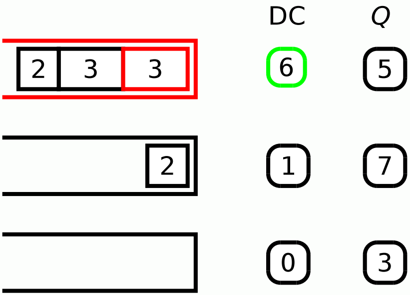
DRR run example

- 3 classes
- Quanta Q: $5, 7, 3$
- CD: Deficit Counter



DRR run example

- 3 classes
- Quanta Q: $5, 7, 3$
- CD: Deficit Counter



ONERA THE FRENCH AEROSPACE LAB

DRR property

- DRR allows per class WCTT bound
- DRR introduces some latency
- previous contribution: latency computation
- this latency depends on quanta Q_i and maximal frame sizes l_i^m

$$DRR\ latency = \frac{Q_i(L - l_i^m) + (F - Q_i)(Q_i + l_i^m)}{F}$$

with $F = \sum_{i=1}^{n} Q_i$, $L = \sum_{i=1}^{n} l_i^m$.

ONERA
THE FRENCH AEROSPACE LAB

Outline

AFDX backbone

Network partitioning

Deficit Round Robin

RTaW-PEGASE AFDX network analyser

Applicability of DRR for AFDX network

Applicability of SP/DRR for AFDX network

Conclusion

RTaW-PEGASE AFDX network analyser

- PEGASE:
  - french-founded project
  - 2009-2012
  - partners: 4 academics (ONERA, ENS, ENS Lyon, Inria), 1 SME (RealTime-at-Work), 1 major company (Thales)
  - Network Calculus for avionics networks
- RTaW-PEGASE:
  - prototype from PEGASE become product
  - developed by RealTime-at-Work
  - augmented with DRR analyse

Outline

AFDX backbone

Network partitioning

Deficit Round Robin

RTaW-PEGASE AFDX network analyser

Applicability of DRR for AFDX network

Applicability of SP/DRR for AFDX network

Conclusion

DRR for AFDX

- DRR pro and cons:
  - + network partitioning
  - - additional latency

DRR for AFDX

- DRR pro and cons:
  - + network partitioning
  - - additional latency
- Bandwidth should increase

DRR for AFDX

- DRR pro and cons:
  - \+ network partitioning
  - \- additional latency
- Bandwidth should increase

$\Rightarrow$ use bandwidth increase to "hide" latency increase

DRR for AFDX

- DRR pro and cons:
  + network partitioning
  - additional latency
- Bandwidth should increase

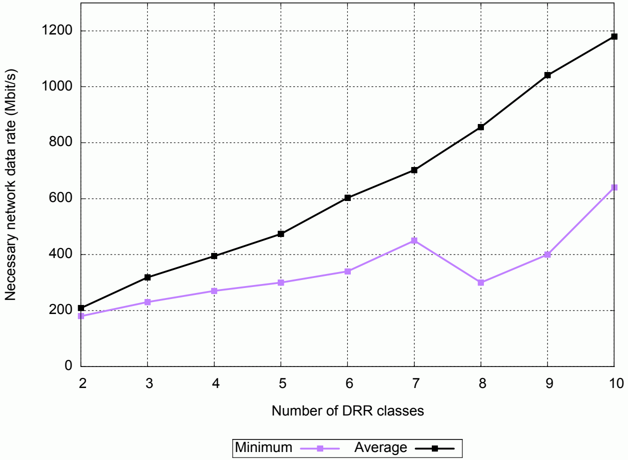
⇒ use bandwidth increase to “hide” latency increase

- does it works ?

DRR for AFDX experiment

- AFDX realistic configuration
- assume n classes
- random allocation of VL to classes
- quanta: proportional to class load
- what bandwidth to get same performance as 100Mb/s FIFO network?
- 100 configurations for each n

DRR for AFDX result



ONERA
THE FRENCH AEROSPACE LAB

AFDX/DRR conclusion

- DRR latency costs

AFDX/DRR conclusion

- DRR latency costs
- but bandwidth can compensate it

AFDX/DRR conclusion

- DRR latency costs
- but bandwidth can compensate it
- 1Gb/s network allows 6-8 independent classes/partitions

Outline

AFDX backbone

Network partitioning

Deficit Round Robin

RTaW-PEGASE AFDX network analyser

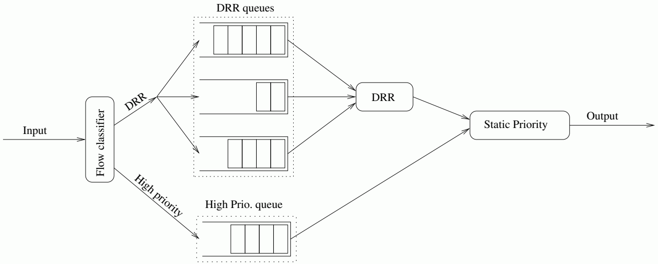
Applicability of DRR for AFDX network

Applicability of SP/DRR for AFDX network

Conclusion

SP/DRR for AFDX

- a few flows with very high constraints
- keep these flows in high priority queue

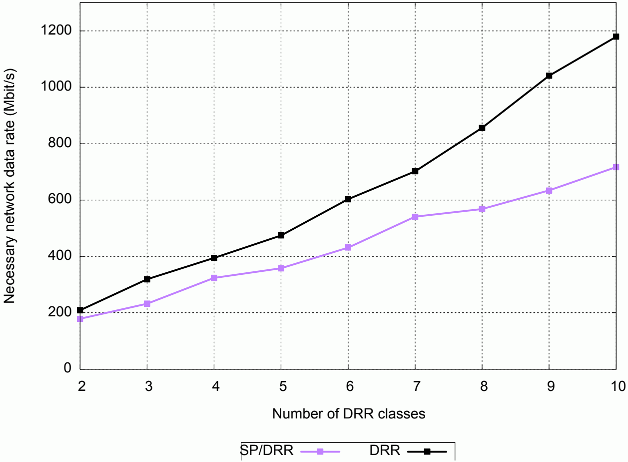


ONERA
THE FRENCH AEROSPACE LAB

SP/DRR for AFDX experiment

- same methodology as for DRR
- all VLs with latency constraints $\leq$ 2ms in high priority

SP/DRR for AFDX result



AFDX/DRR conclusion

- SP/DRR improves number of possible partition

AFDX/DRR conclusion

- SP/DRR improves number of possible partition
- no independence in high priority class

AFDX/DRR conclusion

- SP/DRR improves number of possible partition
- no independence in high priority class
- 1Gb/s network allows 10 independent classes/partitions

Outline

AFDX backbone

Network partitioning

Deficit Round Robin

RTaW-PEGASE AFDX network analyser

Applicability of DRR for AFDX network

Applicability of SP/DRR for AFDX network

Conclusion

Conclusion

- successful AFDX

Conclusion

- successful AFDX
- AFDX evolutions:

Conclusion

- successful AFDX
- AFDX evolutions:
  - bandwidth increase

Conclusion

- successful AFDX
- AFDX evolutions:
  - bandwidth increase
  - network coarse-grained partitioning

Conclusion

- successful AFDX
- AFDX evolutions:
  - bandwidth increase
  - network coarse-grained partitioning
- First experiment with DRR

Conclusion

- successful AFDX
- AFDX evolutions:
  - bandwidth increase
  - network coarse-grained partitioning
- First experiment with DRR
  - Implementation in RTaW-PEGASE

Conclusion

- successful AFDX
- AFDX evolutions:
  - bandwidth increase
  - network coarse-grained partitioning
- First experiment with DRR
  - Implementation in RTaW-PEGASE
  - Is pure DRR acceptable?

Conclusion

- successful AFDX
- AFDX evolutions:
  - bandwidth increase
  - network coarse-grained partitioning
- First experiment with DRR
  - Implementation in RTaW-PEGASE
  - Is pure DRR acceptable?
  - DRR-like solutions:

Conclusion

- successful AFDX
- AFDX evolutions:
  - bandwidth increase
  - network coarse-grained partitioning
- First experiment with DRR
  - Implementation in RTaW-PEGASE
  - Is pure DRR acceptable?
  - DRR-like solutions:
    - SP/DRR

Conclusion

- successful AFDX
- AFDX evolutions:
  - bandwidth increase
  - network coarse-grained partitioning
- First experiment with DRR
  - Implementation in RTaW-PEGASE
  - Is pure DRR acceptable?
  - DRR-like solutions:
    - SP/DRR
    - Modified DRR (cf Cisco, Juniper)

Conclusion

- successful AFDX
- AFDX evolutions:
  - bandwidth increase
  - network coarse-grained partitioning
- First experiment with DRR
  - Implementation in RTaW-PEGASE
  - Is pure DRR acceptable?
  - DRR-like solutions:
    - SP/DRR
    - Modified DRR (cf Cisco, Juniper)
    - Avionic-specific DRR ?