AFDX Emulator for an ARINC-based Training Platform

Jesús Fernández
Héctor Pérez
J. Javier Gutiérrez
Michael González Harbour

AFDX Emulator for an ARINC-based Training Platform

Jesús Fernández
Héctor Pérez
J. Javier Gutiérrez
Michael González Harbour

1. Introduction






2. Design and Implementation

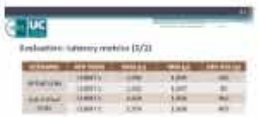






3. Evaluation



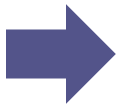



Motivation

- Mature standards for safety-critical applications
  - ARINC-653 for software partitioning
  - ARINC-664 for communication networks (Part 7, AFDX)

Motivation

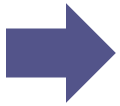
- Mature standards for safety-critical applications
  - ARINC-653 for software partitioning
  - ARINC-664 for communication networks (Part 7, AFDX)
    - Ethernet technology
    - Special-purpose switches



reduce costs and development times

Motivation

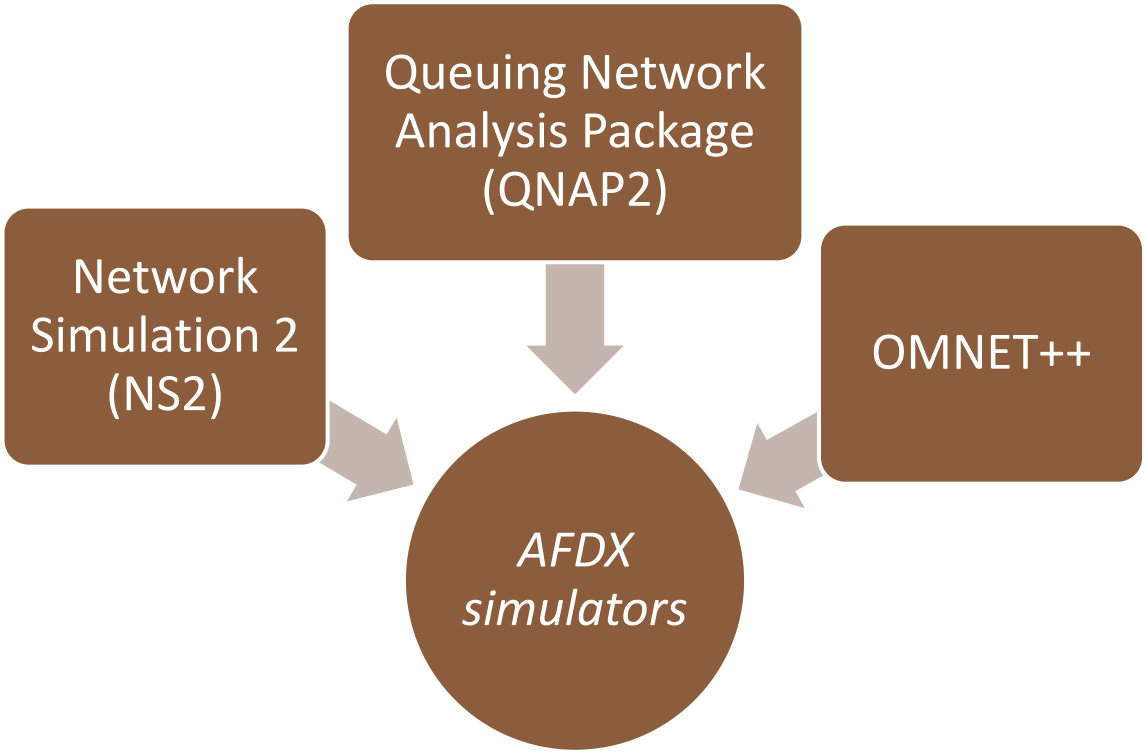
- Mature standards for safety-critical applications
  - ARINC-653 for software partitioning
  - ARINC-664 for communication networks (Part 7, AFDX)
    - Ethernet technology
    - Special-purpose switches



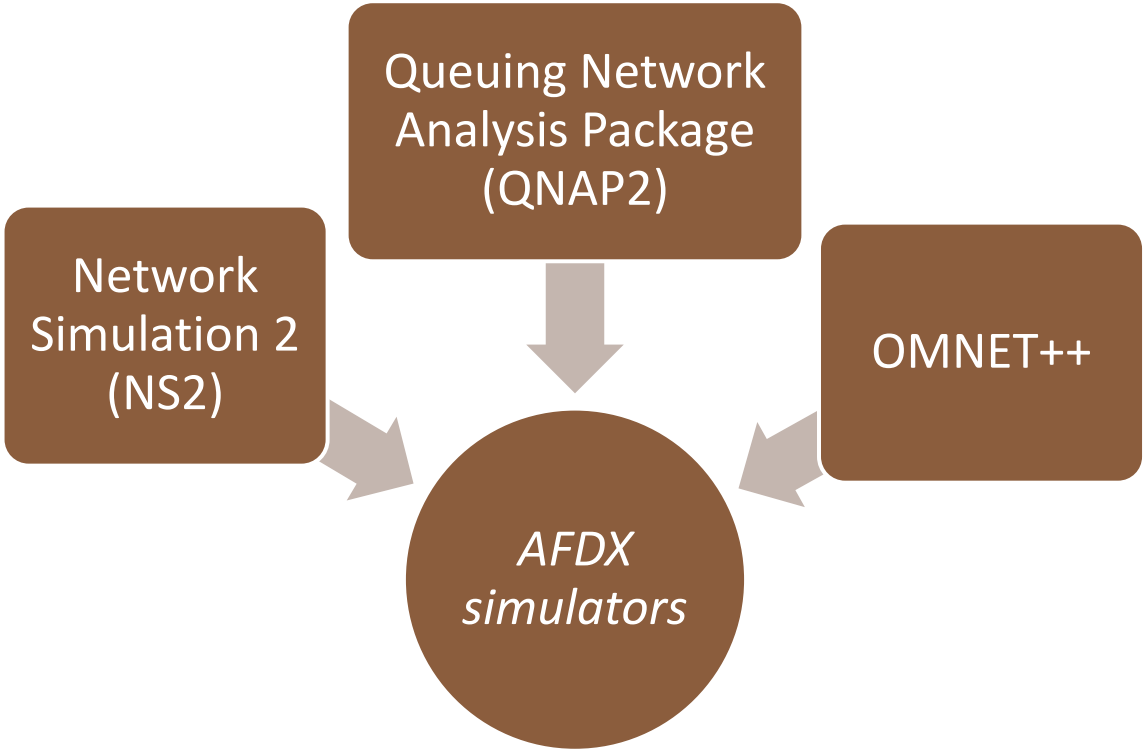
reduce costs and development times

enable access to AFDX technology at lower cost

Related work



Related work



Based on a complete simulation of the AFDX technology (including network cards and switches)

Objectives

- Development of an AFDX emulator
  - using ***real*** communication hardware
    - Ethernet network cards
    - industrial switches with QoS capabilities
  - implementation in Ada

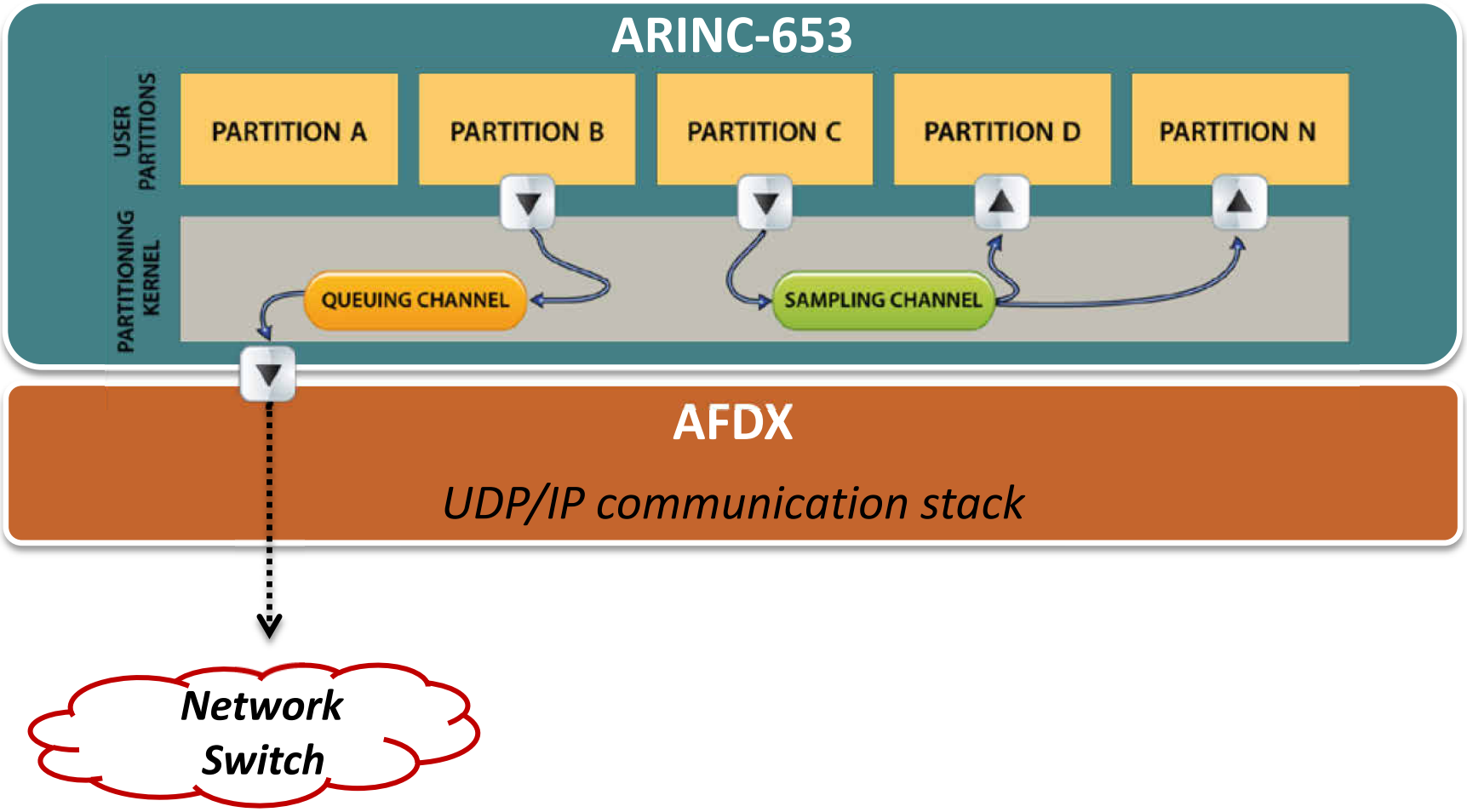
Objectives

- Development of an AFDX emulator
  - using ***real*** communication hardware
    - Ethernet network cards
    - industrial switches with QoS capabilities
  - implementation in Ada
- Integration with a real-time partitioned platform
  - XtratuM hypervisor to support ARINC-653 partitions
  - MaRTE OS to support Ada real-time features

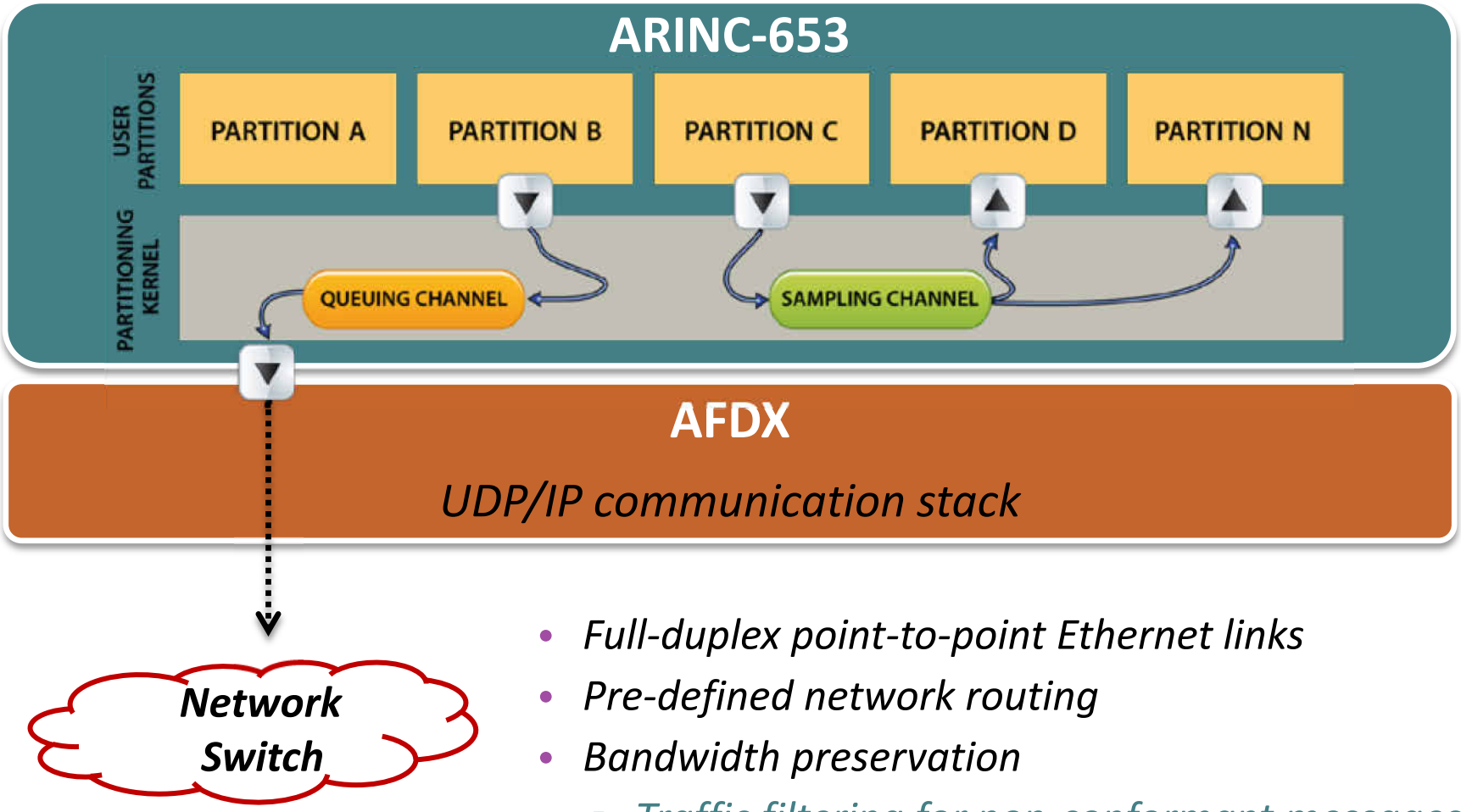
Objectives

- Development of an AFDX emulator
  - using ***real*** communication hardware
    - Ethernet network cards
    - industrial switches with QoS capabilities
  - implementation in Ada
- Integration with a real-time partitioned platform
  - XtratuM hypervisor to support ARINC-653 partitions
  - MaRTE OS to support Ada real-time features
- Suitable for training and research purposes

Background: AFDX architecture (1/3)

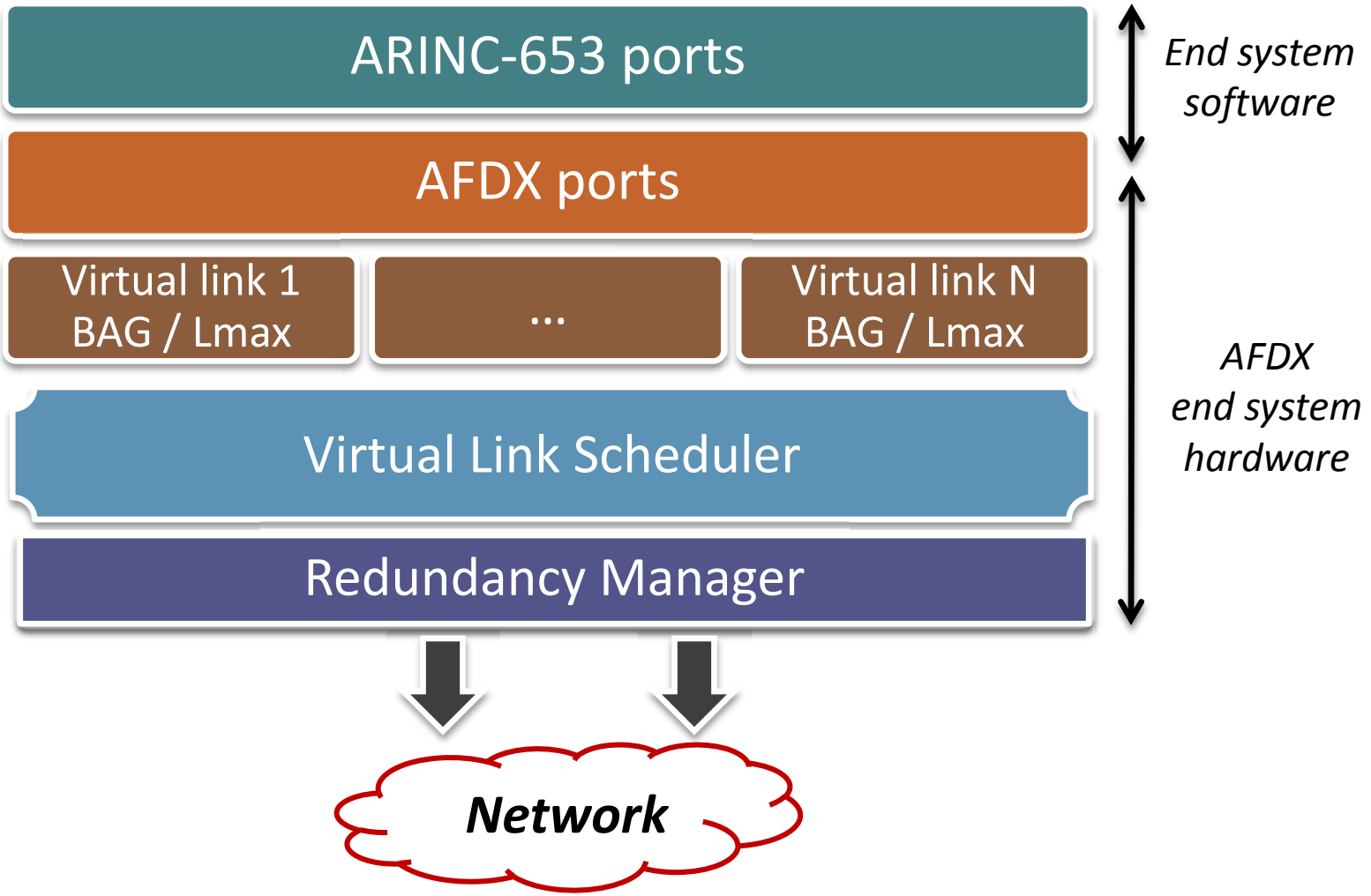


Background: AFDX architecture (1/3)

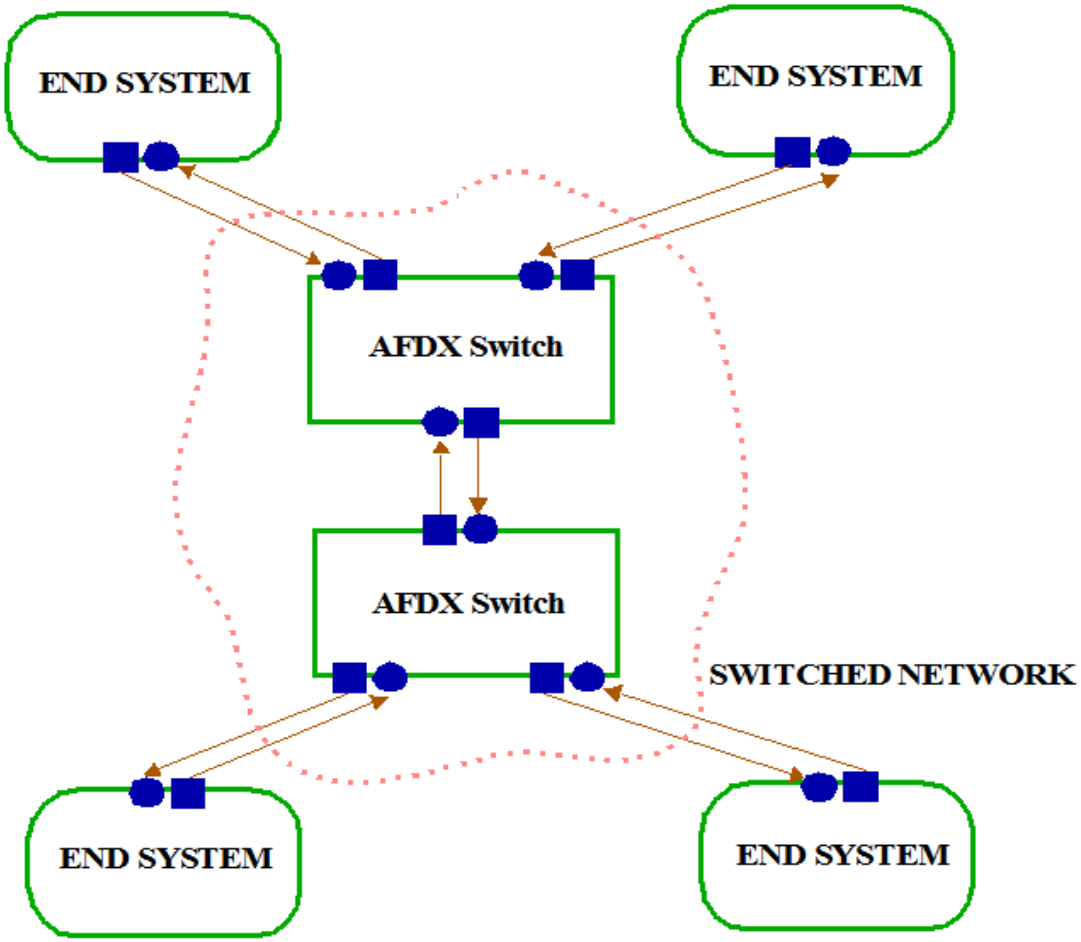


- *Full-duplex point-to-point Ethernet links*
- *Pre-defined network routing*
- *Bandwidth preservation*
  - *Traffic filtering for non-conformant messages*

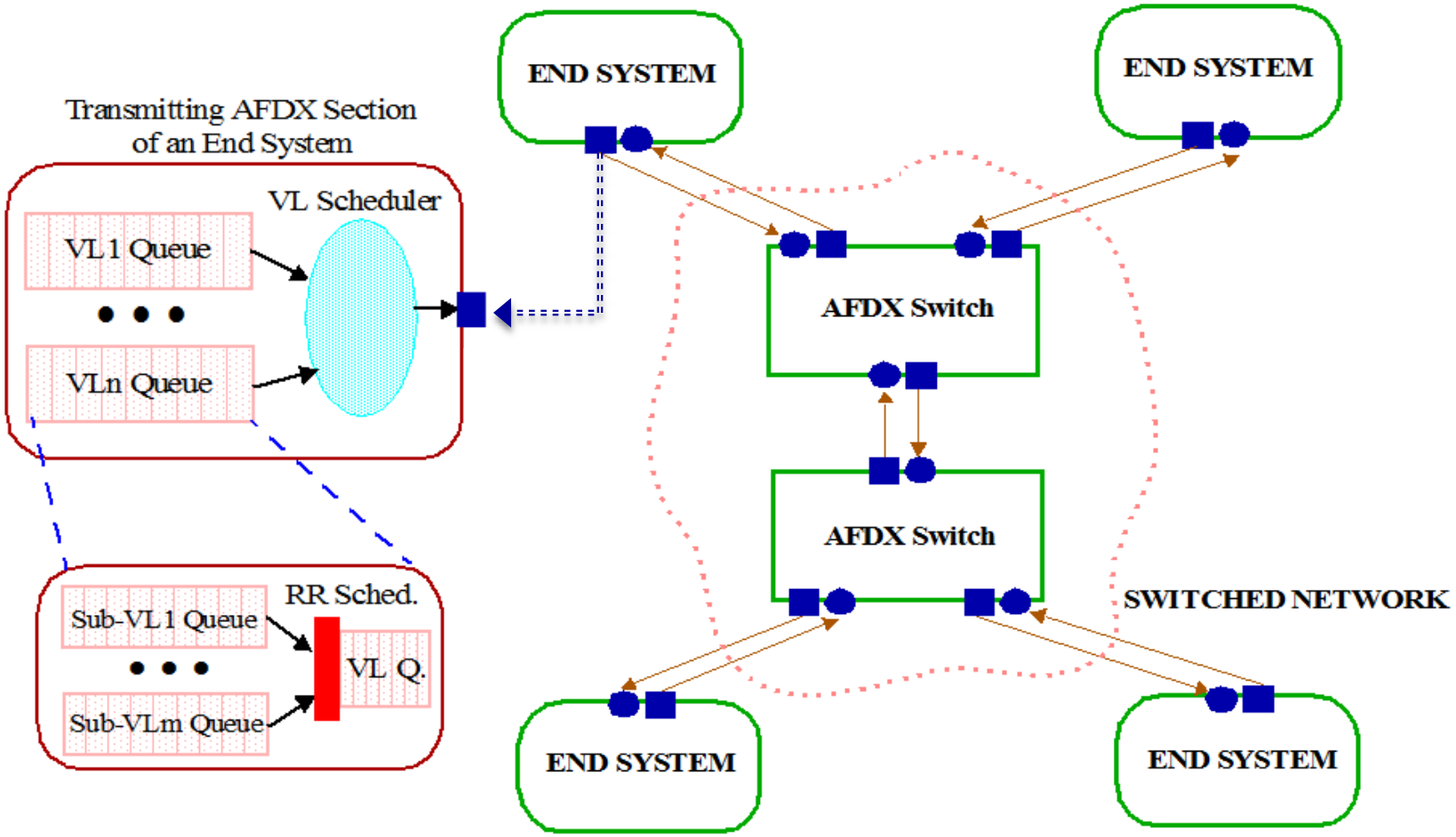
Background: AFDX architecture (2/3)



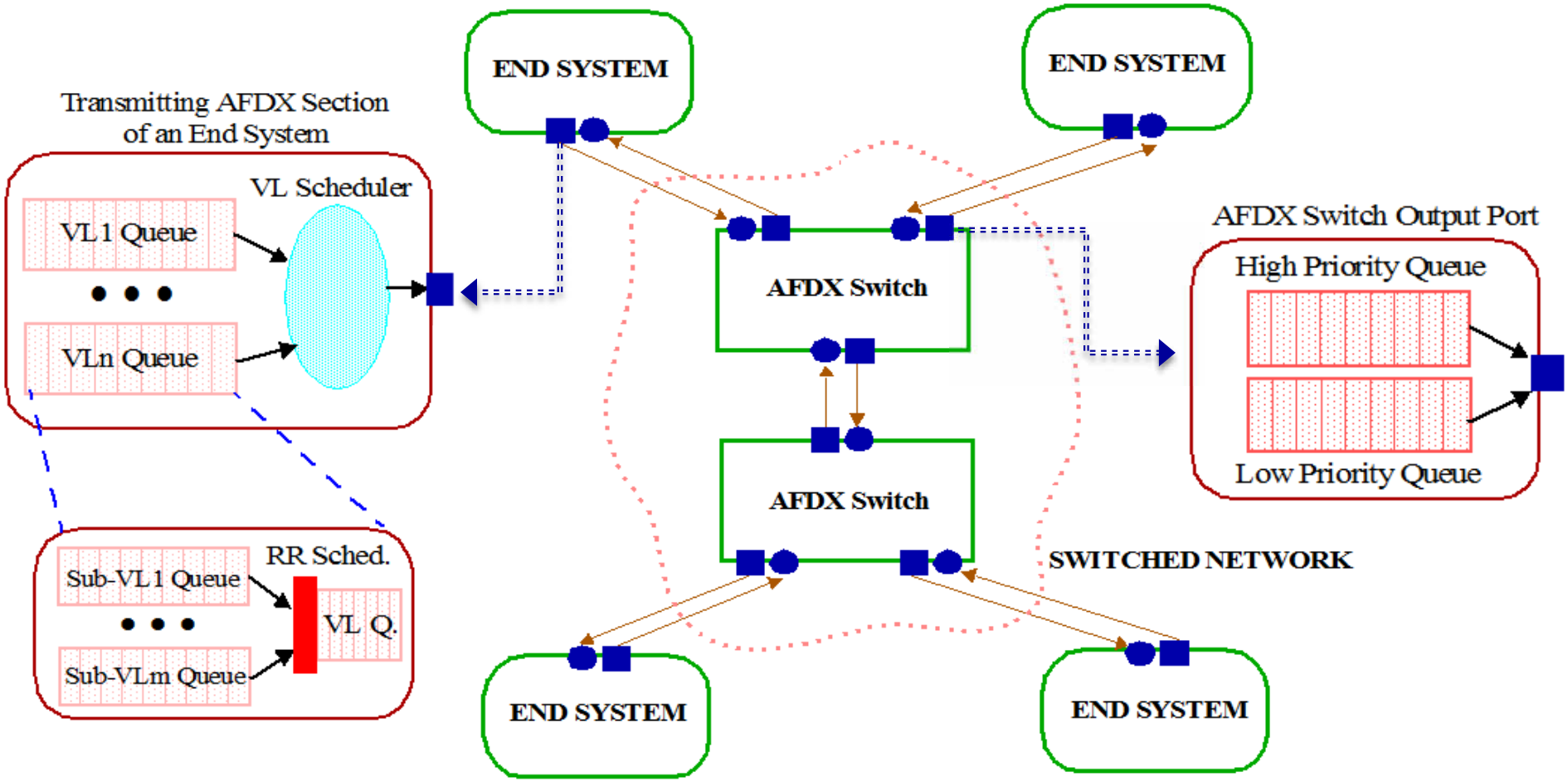
Background: AFDX architecture (3/3)



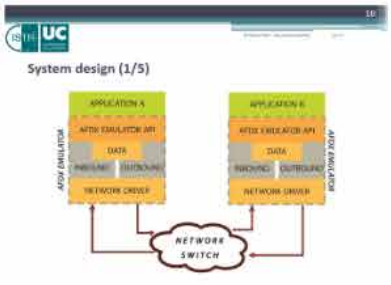
Background: AFDX architecture (3/3)

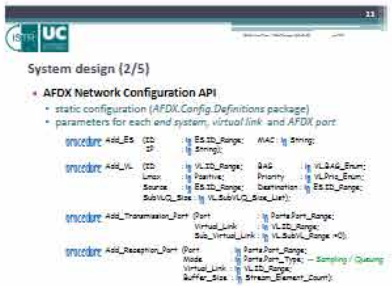


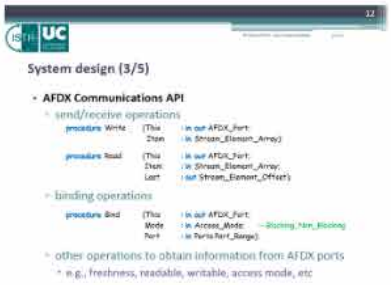
Background: AFDX architecture (3/3)

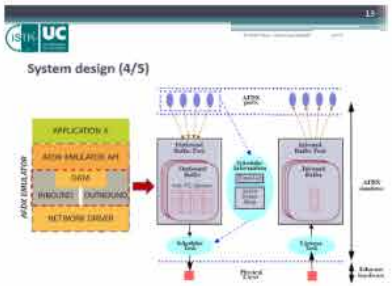


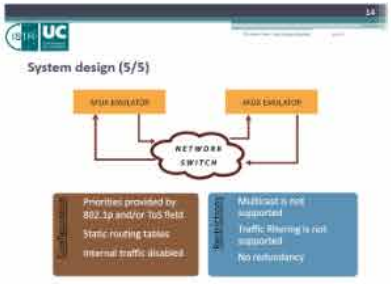
2. Design and Implementation

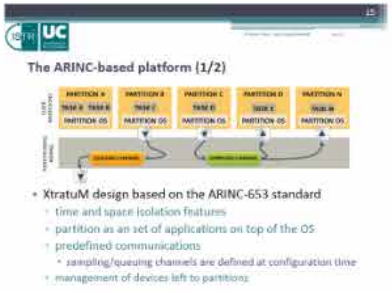


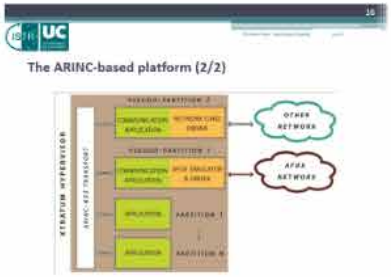




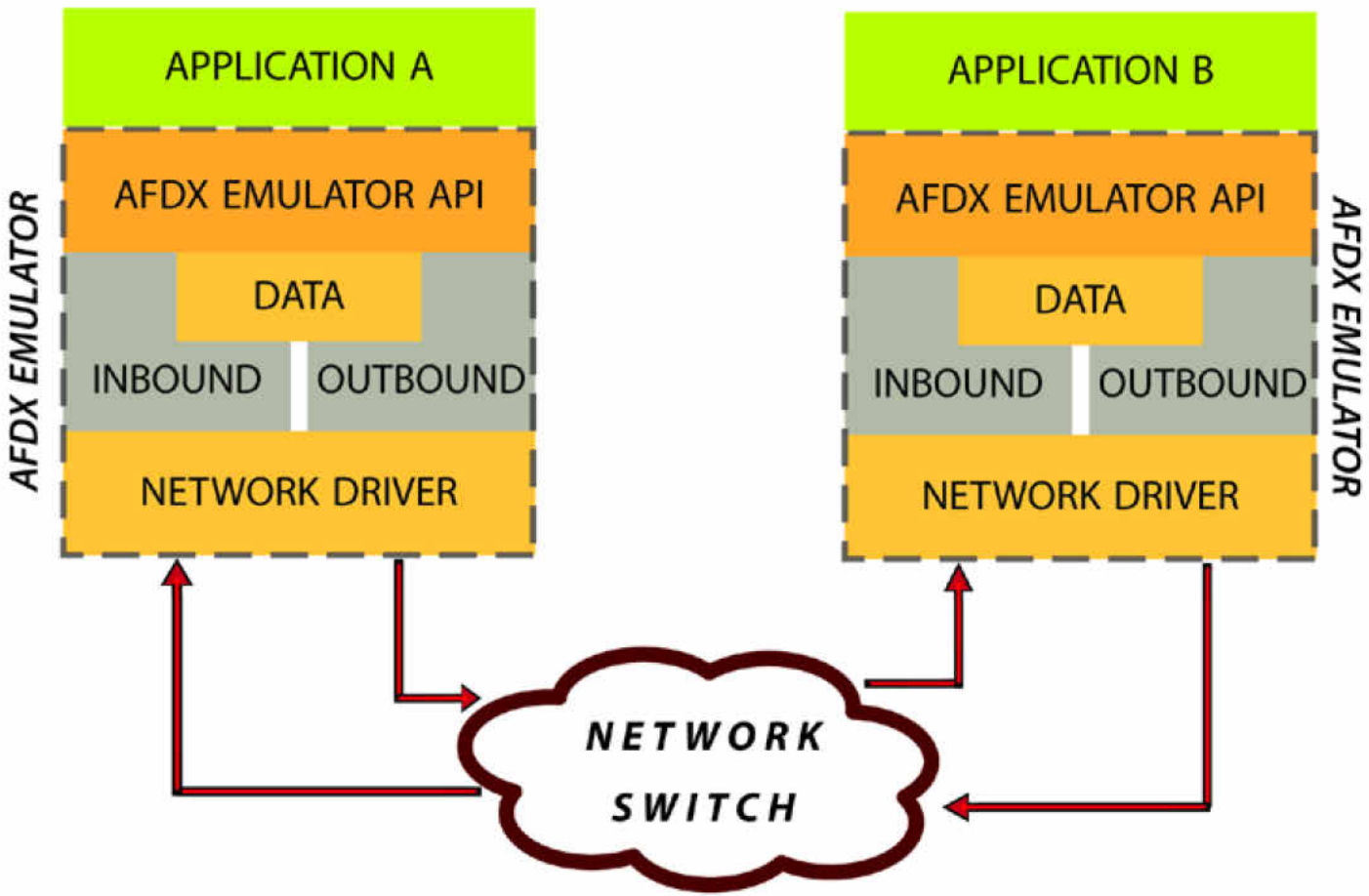








System design (1/5)



System design (2/5)

- **AFDX Network Configuration API**
  - static configuration (*AFDX.Config.Definitions* package)
  - parameters for each *end system, virtual link* and *AFDX port*

```
procedure Add_ES   (ID       : in ES.ID_Range;     MAC : in String;
                    IP       : in String);
```

System design (2/5)

- **AFDX Network Configuration API**
  - static configuration (*AFDX.Config.Definitions* package)
  - parameters for each *end system, virtual link* and *AFDX port*

```
procedure Add_ES  (ID        : in ES.ID_Range;     MAC : in String;
                   IP        : in String);

procedure Add_VL  (ID        : in VL.ID_Range;     BAG         : in VL.BAG_Enum;
                   Lmax      : in Positive;        Priority    : in VL.Prio_Enum;
                   Source    : in ES.ID_Range;     Destination : in ES.ID_Range;
                   SubVLQ_Size : in VL.SubVLQ_Size_List);
```

System design (2/5)

- **AFDX Network Configuration API**
  - static configuration (*AFDX.Config.Definitions* package)
  - parameters for each *end system, virtual link* and *AFDX port*

```
procedure Add_ES   (ID          : in ES.ID_Range;     MAC : in String;
                    IP          : in String);

procedure Add_VL   (ID          : in VL.ID_Range;     BAG          : in VL.BAG_Enum;
                    Lmax        : in Positive;        Priority     : in VL.Prio_Enum;
                    Source      : in ES.ID_Range;     Destination  : in ES.ID_Range;
                    SubVLQ_Size : in VL.SubVLQ_Size_List);

procedure Add_Transmission_Port (Port             : in Ports.Port_Range;
                                 Virtual_Link     : in VL.ID_Range;
                                 Sub_Virtual_Link : in VL.SubVL_Range :=0);

procedure Add_Reception_Port (Port        : in Ports.Port_Range;
                              Mode        : in Ports.Port_Type; -- Sampling / Queuing
                              Virtual_Link : in VL.ID_Range;
                              Buffer_Size : in Stream_Element_Count);
```

System design (3/5)

- **AFDX Communications API**
  - send/receive operations

```
procedure Write     (This     : in out AFDX_Port;
                     Item     : in Stream_Element_Array);

procedure Read      (This     : in out AFDX_Port;
                     Item     : in Stream_Element_Array;
                     Last     : out Stream_Element_Offset);
```

System design (3/5)

- **AFDX Communications API**
  - send/receive operations

```
procedure Write     (This     : in out AFDX_Port;
                     Item     : in Stream_Element_Array);

procedure Read      (This     : in out AFDX_Port;
                     Item     : in Stream_Element_Array;
                     Last     : out Stream_Element_Offset);
```

  - binding operations

```
procedure Bind      (This     : in out AFDX_Port;
                     Mode     : in Access_Mode;     -- Blocking, Non_Blocking
                     Port     : in Ports.Port_Range);
```

System design (3/5)

- **AFDX Communications API**
  - send/receive operations

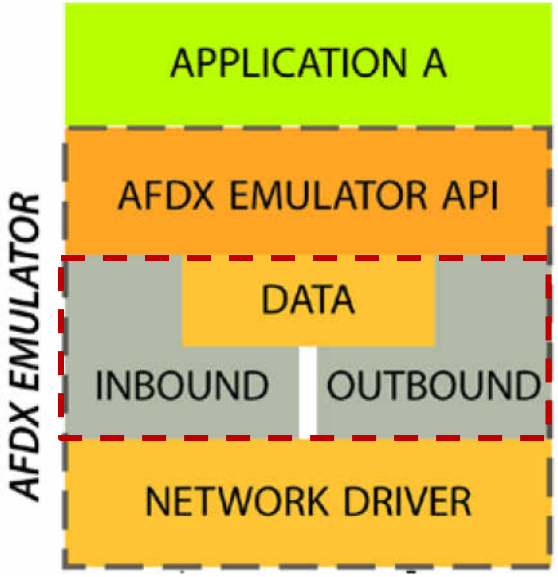
```
procedure Write     (This     : in out AFDX_Port;
                     Item     : in Stream_Element_Array);

procedure Read      (This     : in out AFDX_Port;
                     Item     : in Stream_Element_Array;
                     Last     : out Stream_Element_Offset);
```

  - binding operations

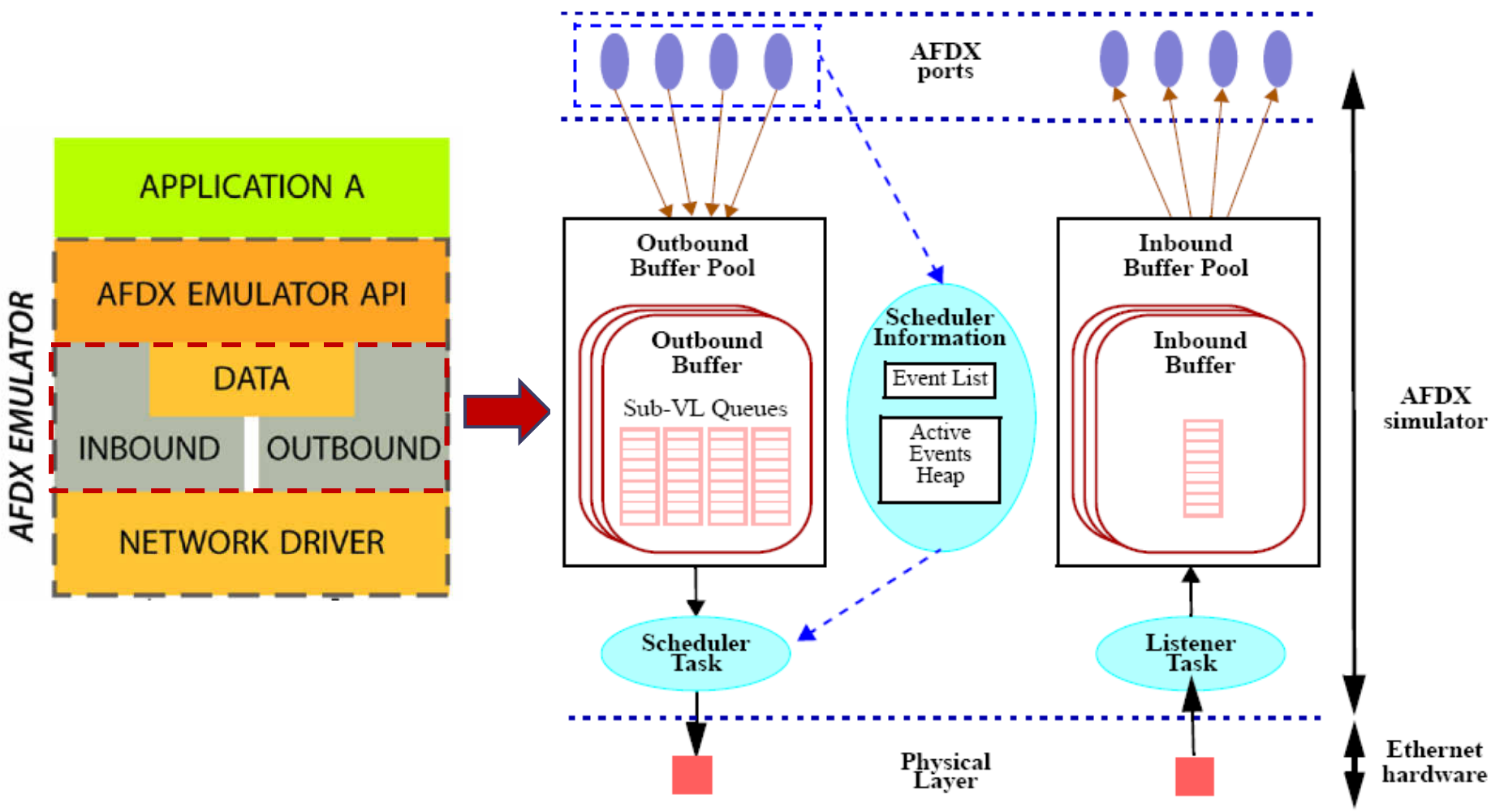
```
procedure Bind      (This     : in out AFDX_Port;
                     Mode     : in Access_Mode;     -- Blocking, Non_Blocking
                     Port     : in Ports.Port_Range);
```

  - other operations to obtain information from AFDX ports
    - e.g., freshness, readable, writable, access mode, etc

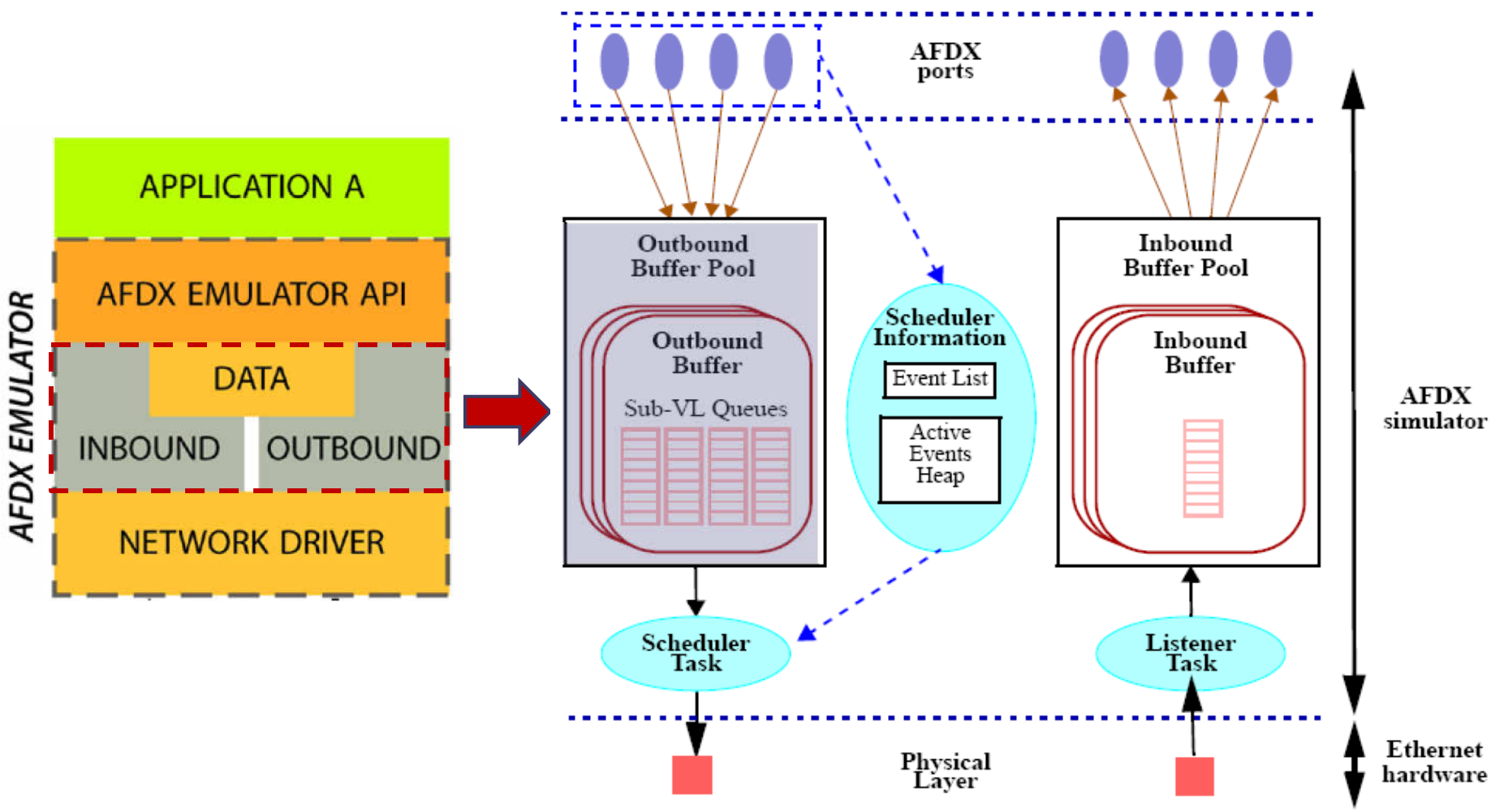
System design (4/5)



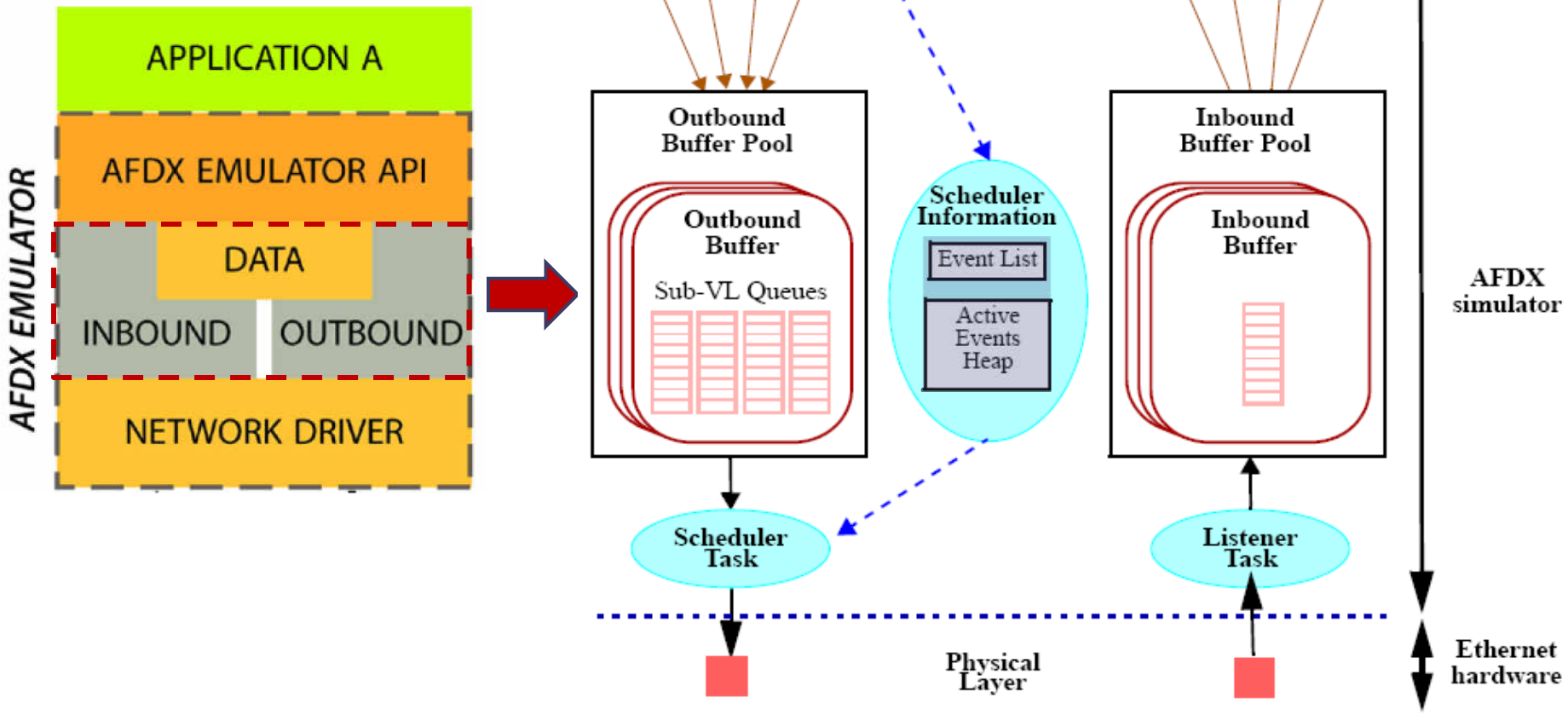
System design (4/5)



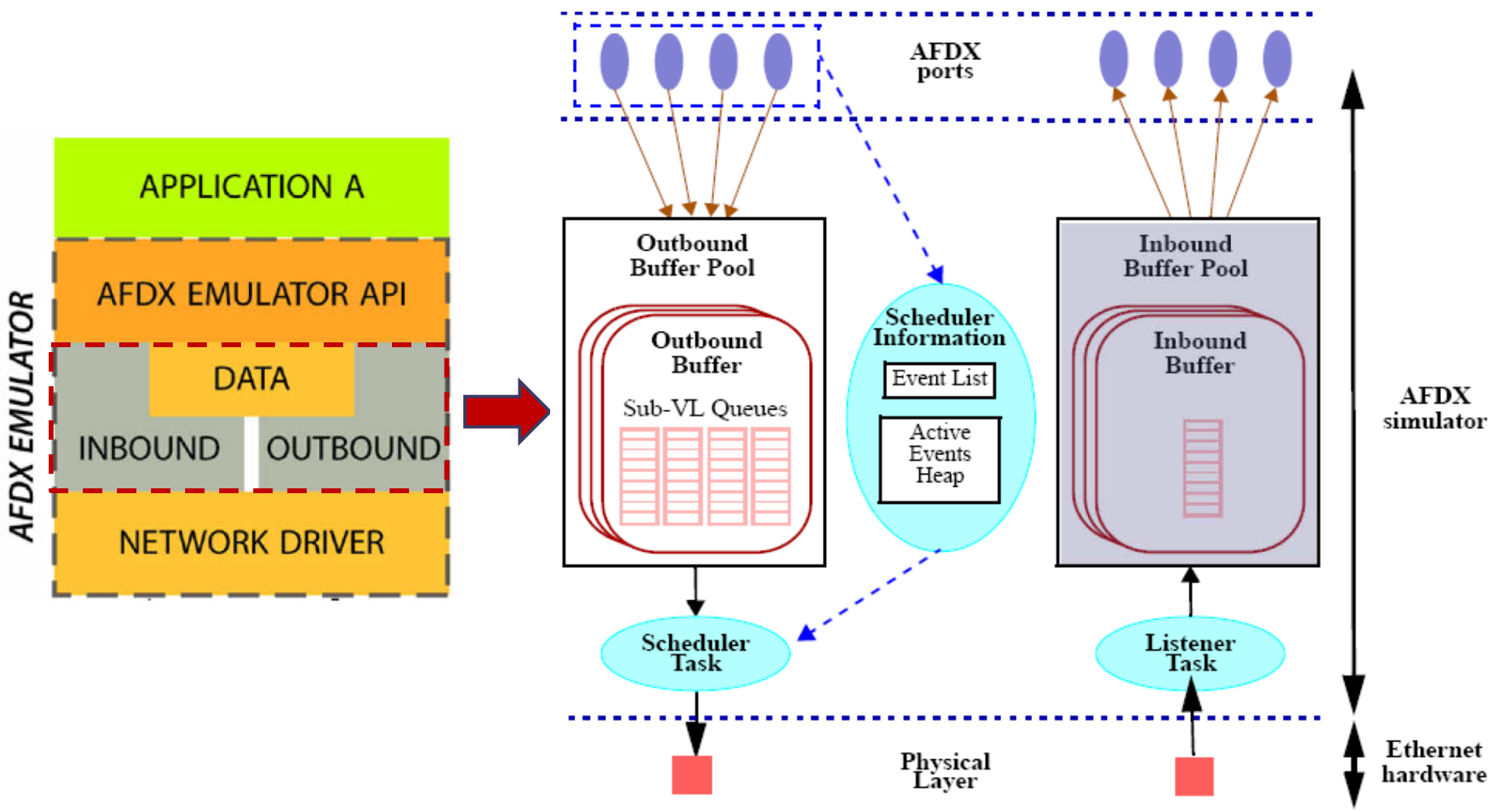
System design (4/5)



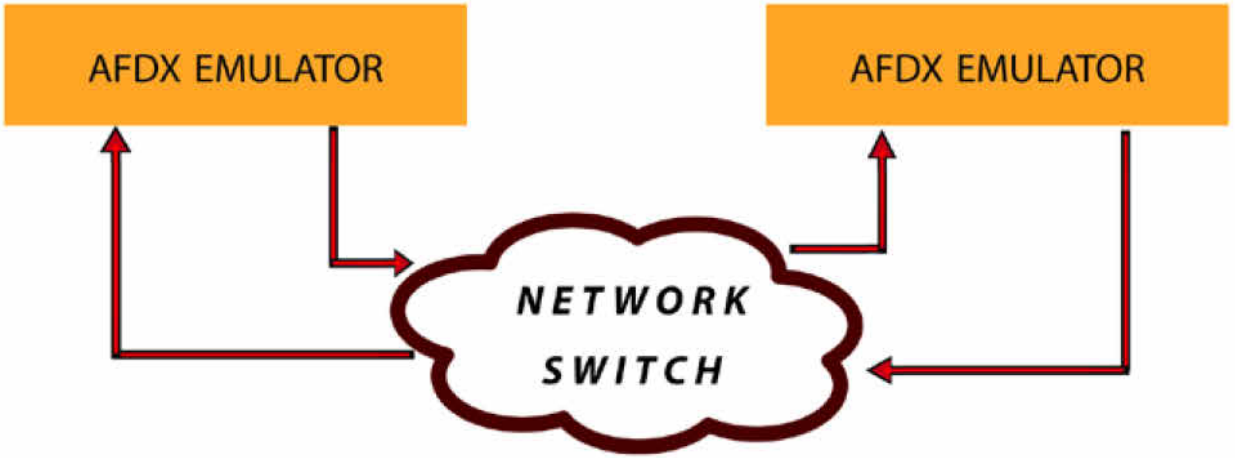
System design (4/5)



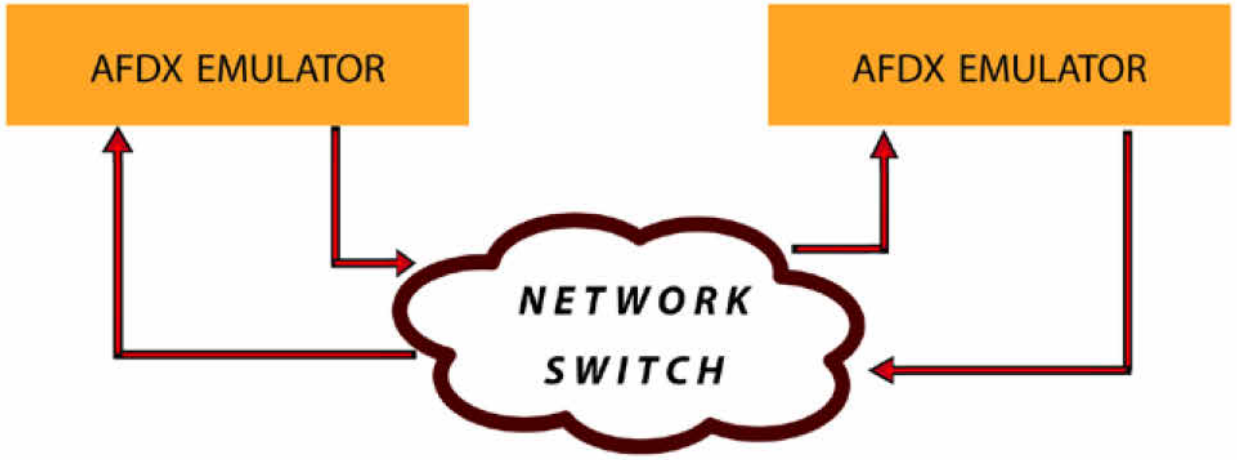
System design (4/5)



System design (5/5)



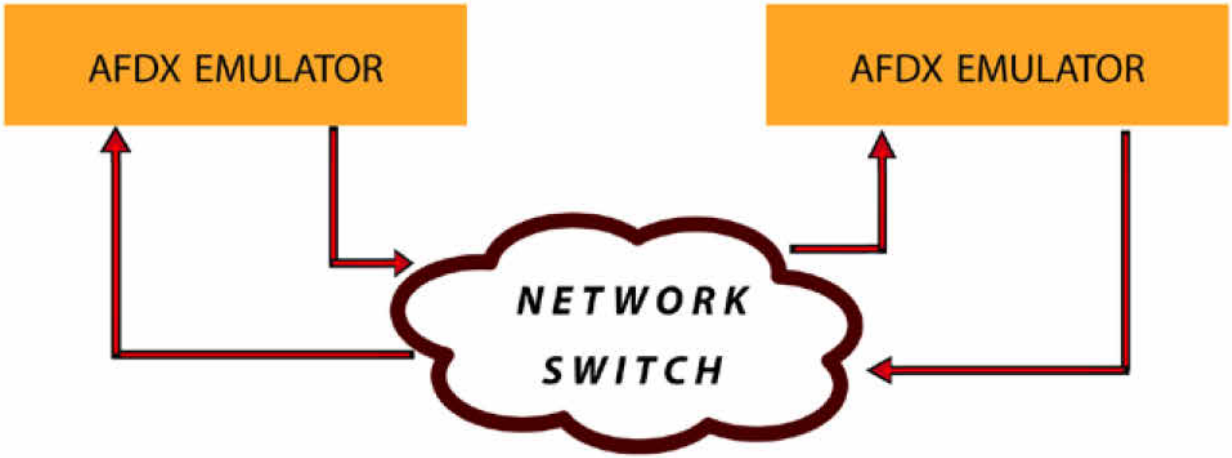
System design (5/5)



Configuration

- Priorities provided by 802.1p and/or ToS field
- Static routing tables
- Internal traffic disabled

System design (5/5)



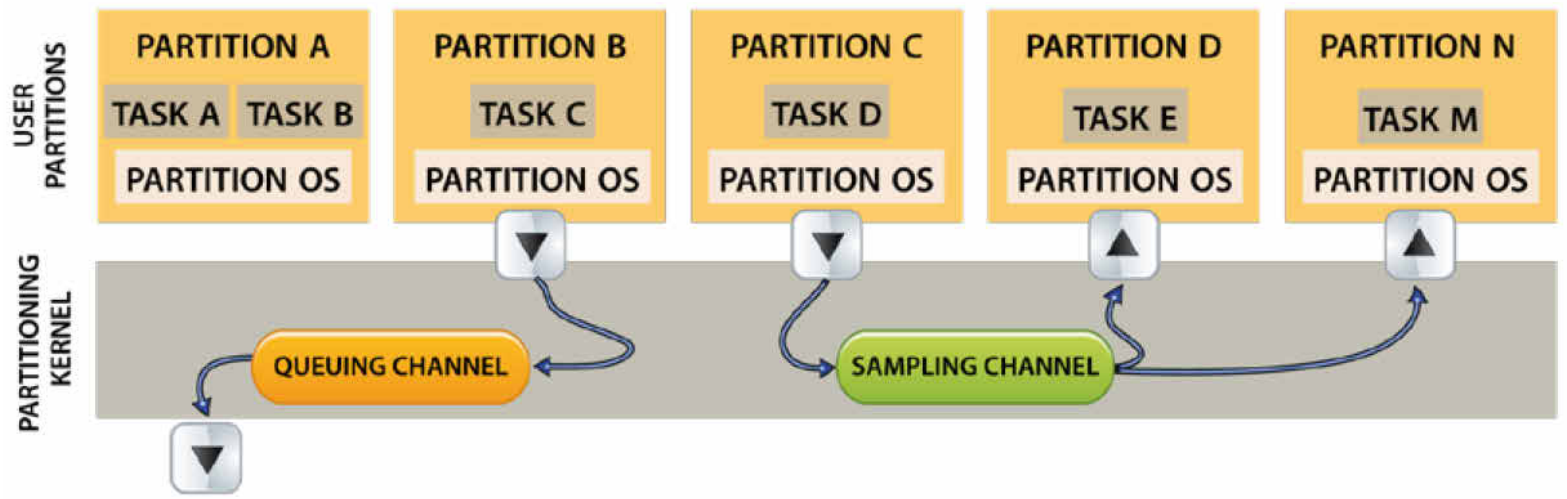
Configuration

- Priorities provided by 802.1p and/or ToS field
- Static routing tables
- Internal traffic disabled

Restrictions

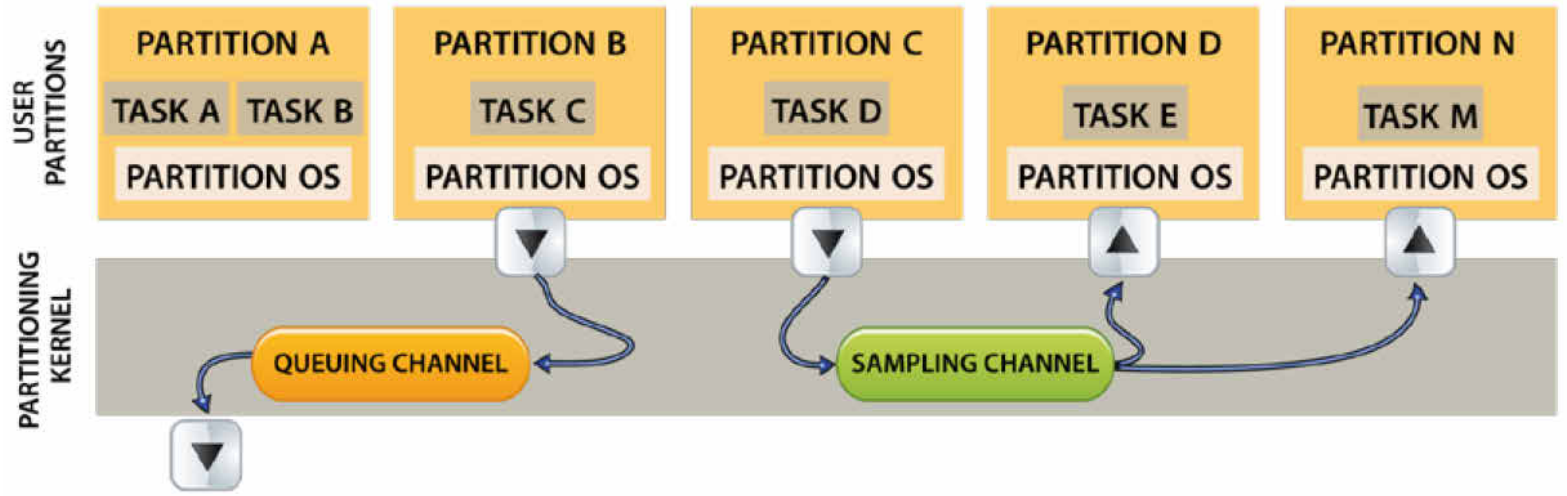
- Multicast is not supported
- Traffic filtering is not supported
- No redundancy

The ARINC-based platform (1/2)



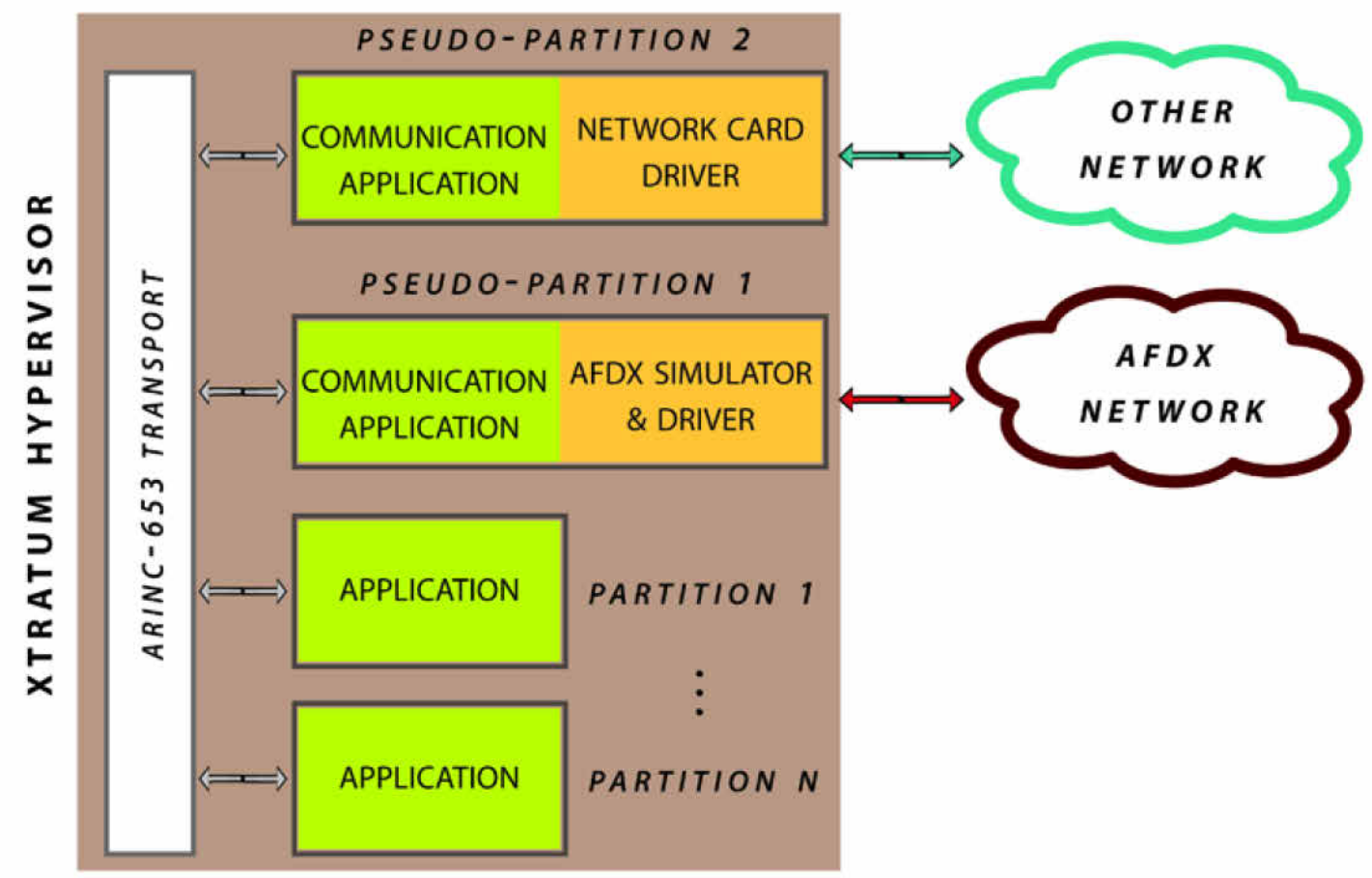
- XtratuM design based on the ARINC-653 standard
  - time and space isolation features
  - partition as an set of applications on top of the OS

The ARINC-based platform (1/2)



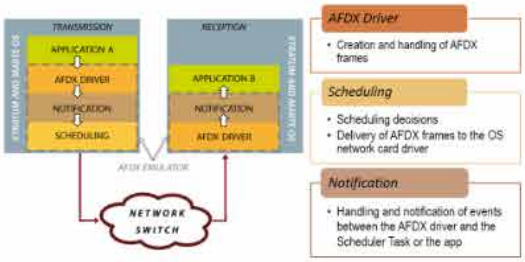
- XtratuM design based on the ARINC-653 standard
  - time and space isolation features
  - partition as an set of applications on top of the OS
  - predefined communications
    - sampling/queuing channels are defined at configuration time
  - management of devices left to partitions

The ARINC-based platform (2/2)

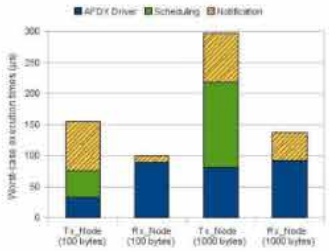


3. Evaluation

Evaluation: AFDX Emulator Characterization (1/2)

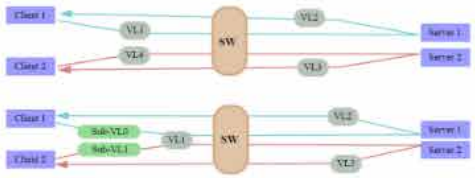


Evaluation: AFDX Emulator Characterization (2/2)



	AFDX Std	AFDX Emul
Tx Latency	40µ	☒
Max jitter in VLs	500µ	⊟
Switch	100µ	☑
Rx Latency	150µ	☑

Evaluation: Latency metrics (1/2)



Lmax	BAG	Tx payload	Rx payload
98 bytes	1 ms	120 bytes	56 bytes

Evaluation: Latency metrics (2/2)

SCENARIO	APP TASKS	MAX (µ)	MIN (µ)	DEV STD (µ)
Virtual Links	CLIENT 1	1,956	1,605	101
	CLIENT 2	1,932	1,607	95
Sub-Virtual Links	CLIENT 1	2,614	1,606	462
	CLIENT 2	3,374	1,606	865

- Virtual Links scenario
  - Minimum interval between frames (BAG)
- Sub-Virtual Links scenario
  - Use of Round-Robin scheduling for the same VL

Conclusions

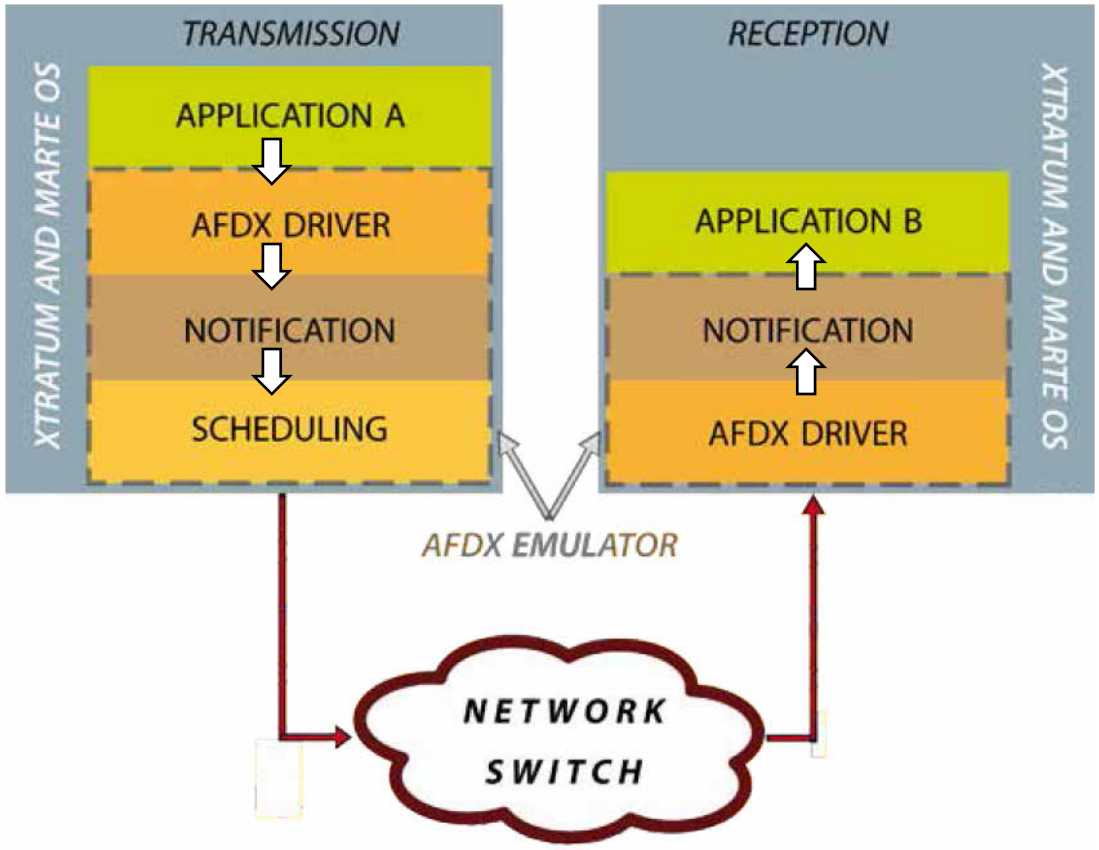
- Development of an AFDX emulator
  - provides applications with Ada interfaces
    - Configuration and Communication APIs
  - relies on standard ethernet hardware
    - thus enabling the use of an ARINC-based platform
- Integration with a real-time partitioned platform
  - XtratuM + MaRTE OS + Ada applications
  - Training/Educational platform with reasonable overheads
- Software available at ***http://marte.unican.es***
  - it can be used with or without hypervisor

everything will be okay
in the end.

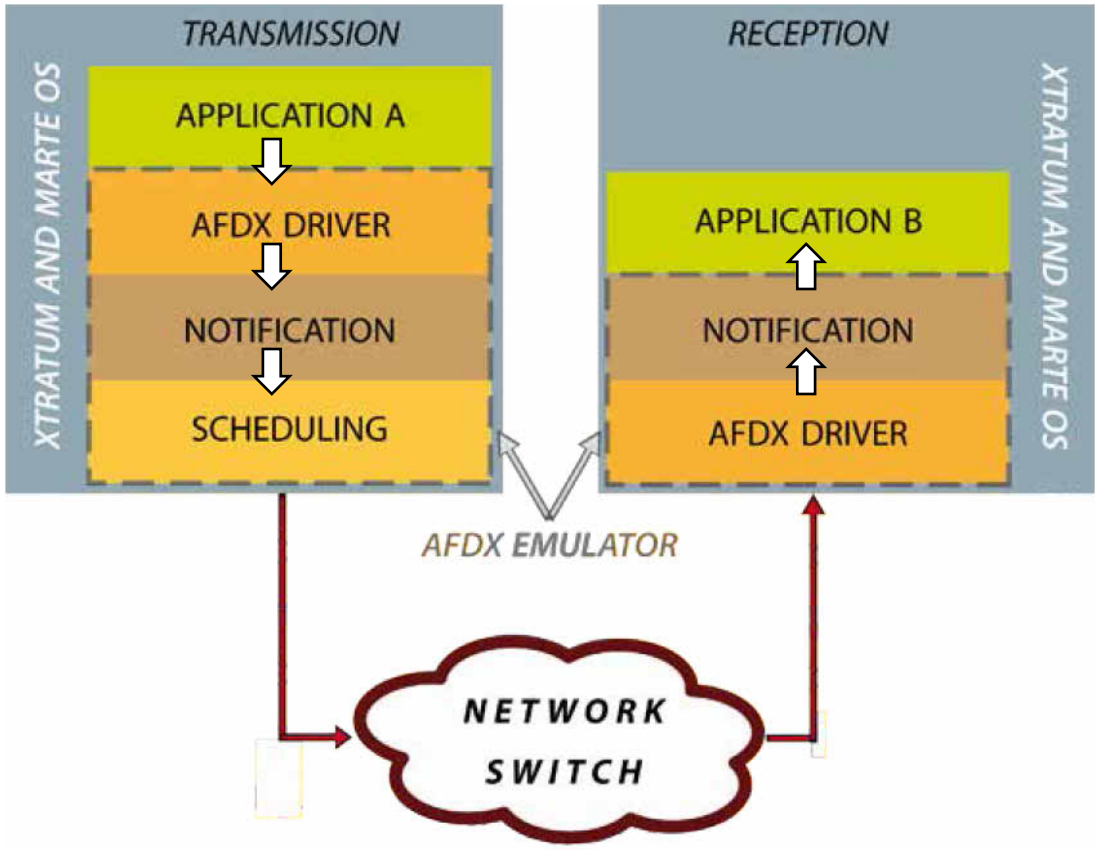
if it's not okay,
it's not the end.

(unknown)

Evaluation: AFDX Emulator Characterization (1/2)



Evaluation: AFDX Emulator Characterization (1/2)



AFDX Driver

- Creation and handling of AFDX frames

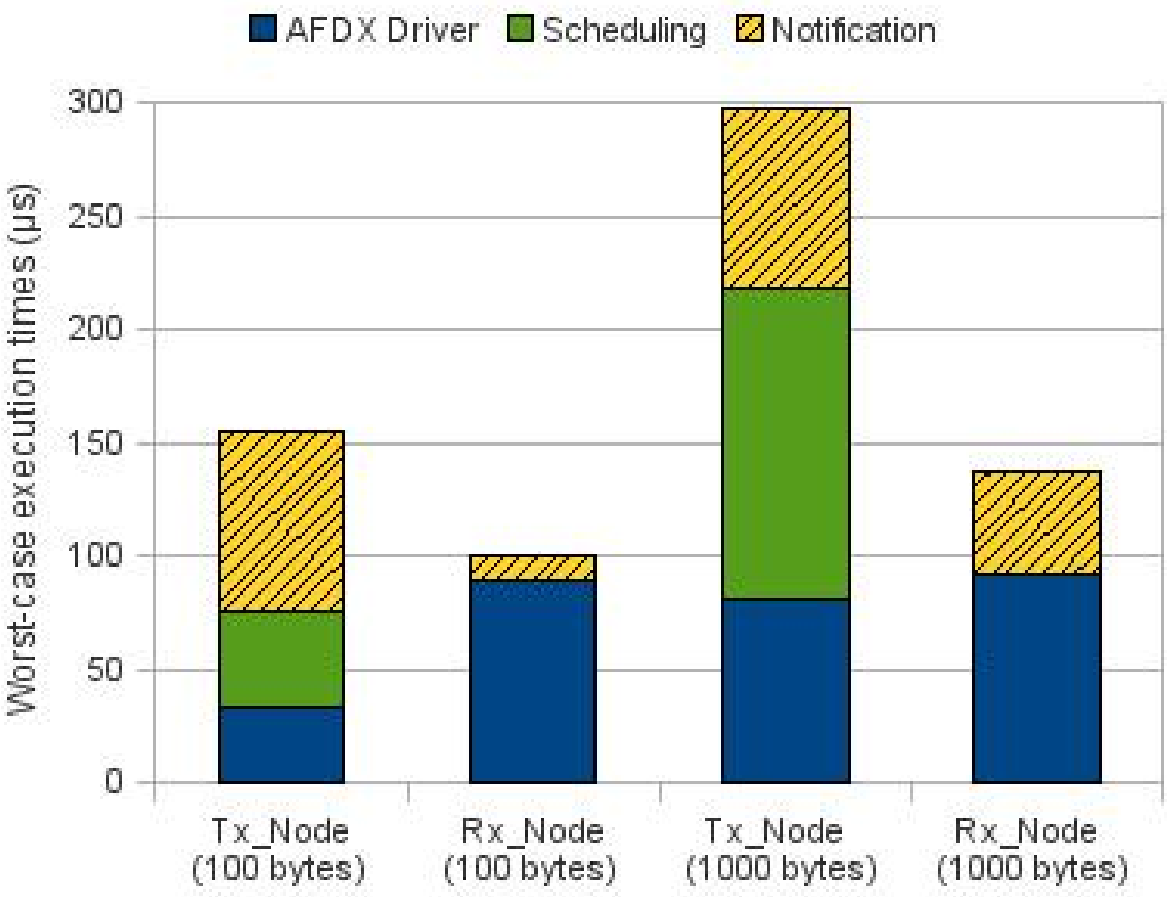
Scheduling

- Scheduling decisions
- Delivery of AFDX frames to the OS network card driver

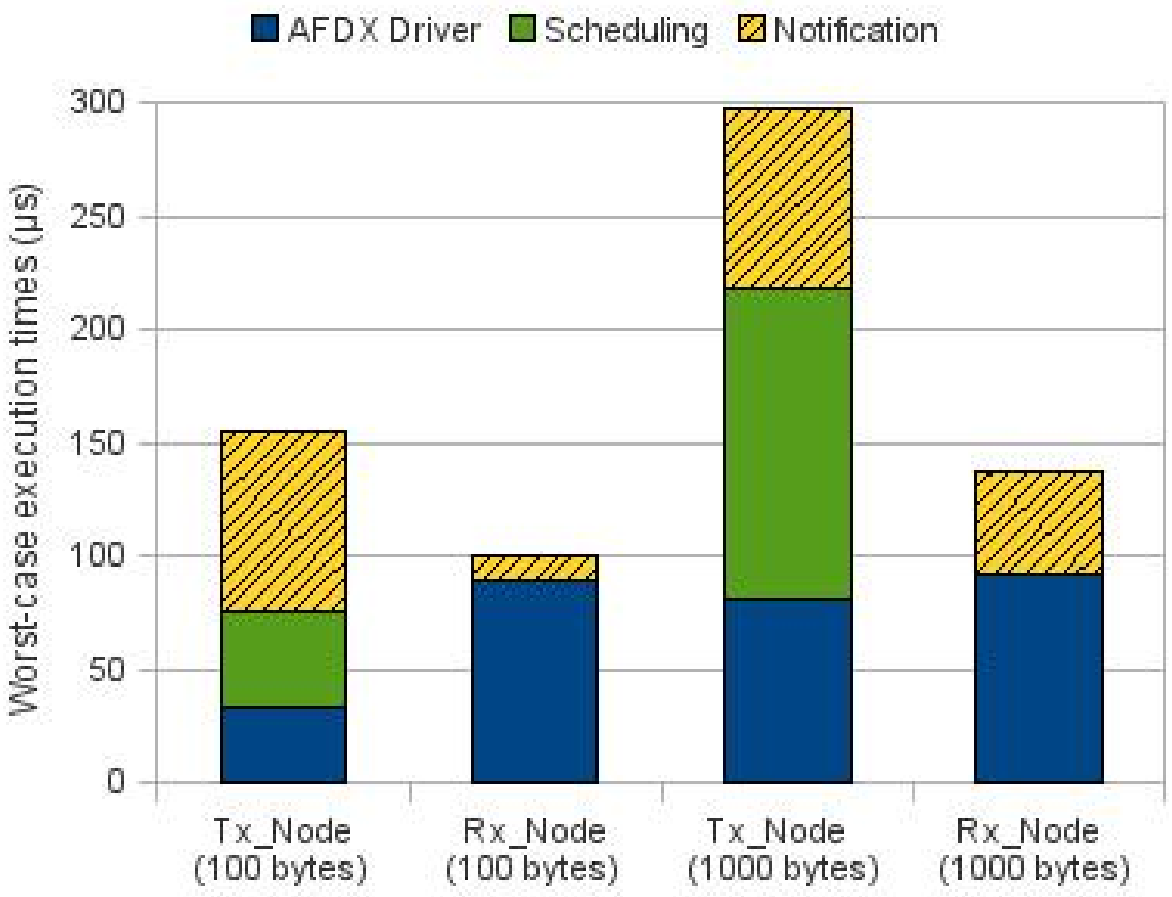
Notification

- Handling and notification of events between the AFDX driver and the Scheduler Task or the app

Evaluation: AFDX Emulator Characterization (2/2)

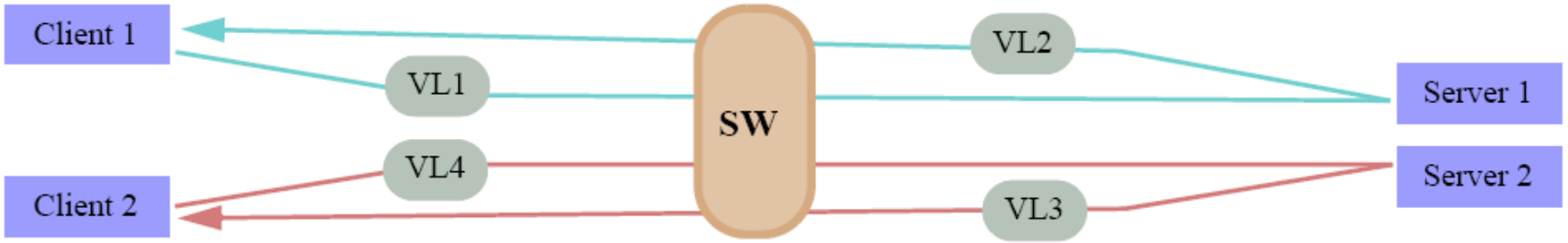


Evaluation: AFDX Emulator Characterization (2/2)

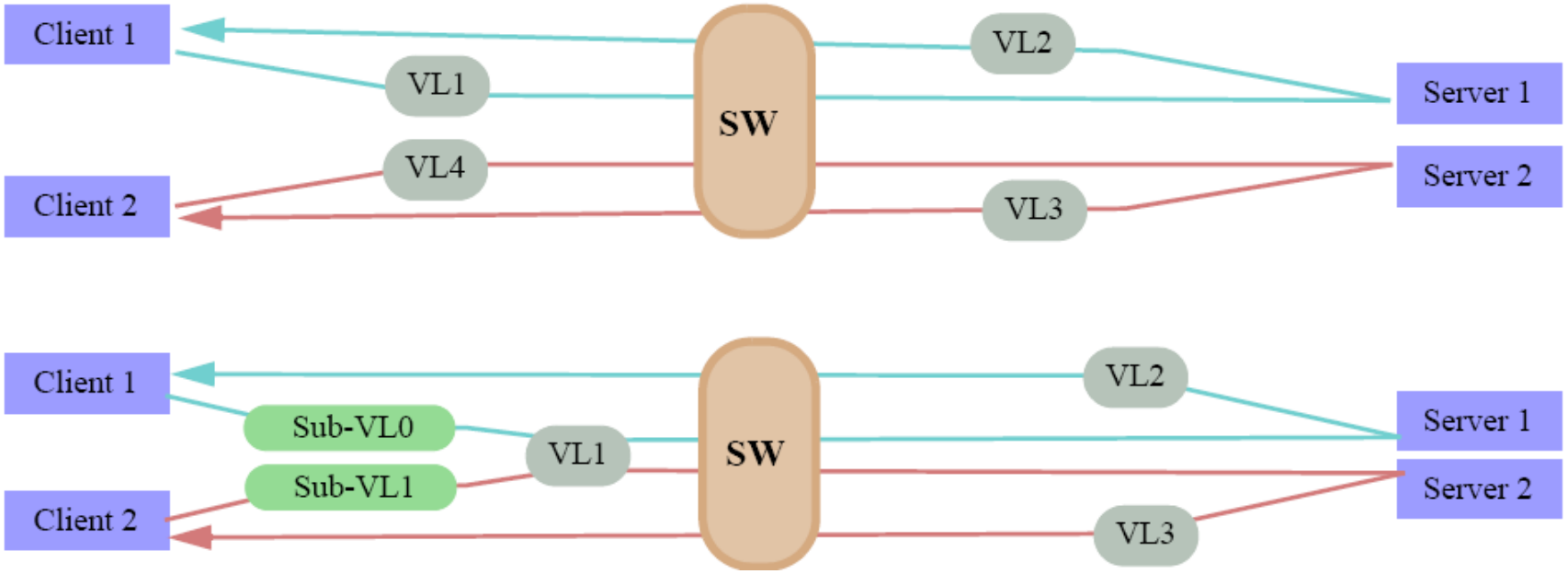


	AFDX Std	AFDX Emul
Tx Latency	40µ	☒
Max jitter in VLs	500µ	⊟
Switch	100µ	☑
Rx Latency	150µ	☑

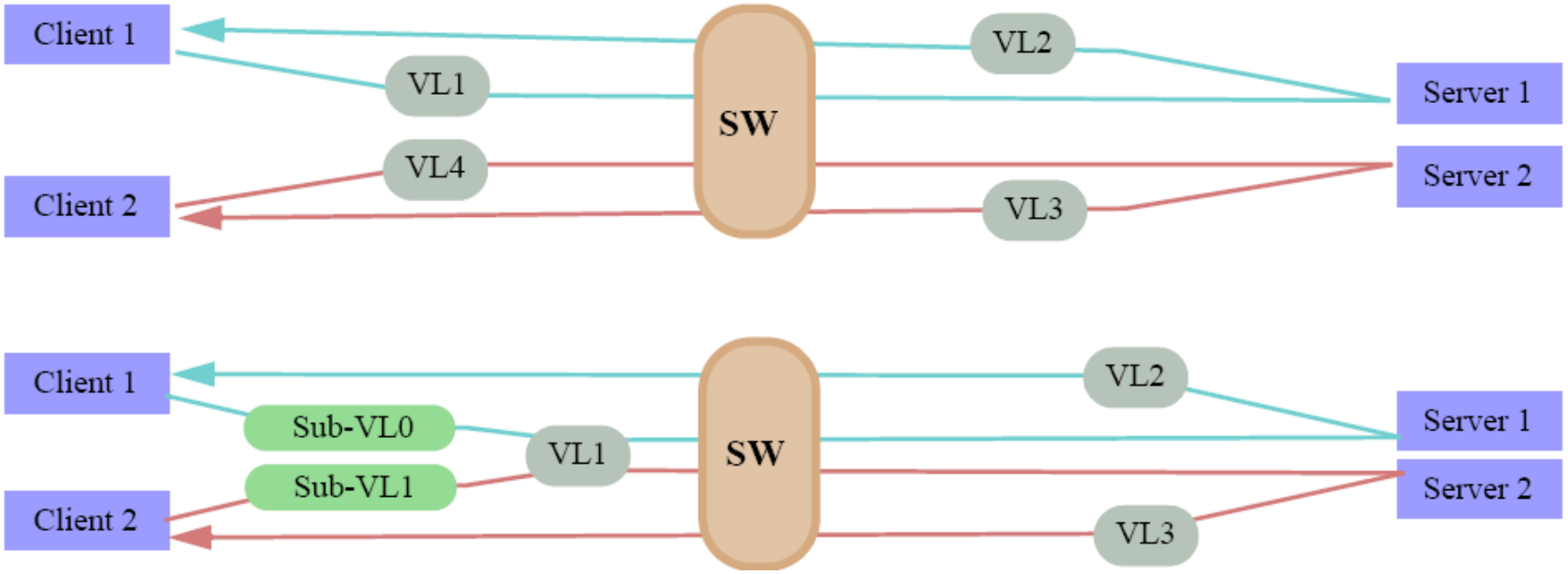
Evaluation: Latency metrics (1/2)



Evaluation: Latency metrics (1/2)



Evaluation: Latency metrics (1/2)



Lmax	BAG	Tx payload	Rx payload
98 bytes	1 ms	120 bytes	56 bytes

Evaluation: Latency metrics (2/2)

SCENARIO	APP TASKS	MAX (µ)	MIN (µ)	DEV STD (µ)
Virtual Links	CLIENT 1	1,956	1,605	101
	CLIENT 2	1,932	1,607	95

- Virtual Links scenario
  - Minimum interval between frames (BAG)

Evaluation: Latency metrics (2/2)

SCENARIO	APP TASKS	MAX (µ)	MIN (µ)	DEV STD (µ)
Virtual Links	CLIENT 1	1,956	1,605	101
	CLIENT 2	1,932	1,607	95
Sub-Virtual Links	CLIENT 1	2,614	1,606	462
	CLIENT 2	3,374	1,606	865

- Virtual Links scenario
  - Minimum interval between frames (BAG)
- Sub-Virtual Links scenario
  - Use of Round-Robin scheduling for the same VL

Conclusions

- Development of an AFDX emulator
  - provides applications with Ada Interfaces
    - Configuration and Communication APIs
  - relies on standard ethernet hardware
    - thus enabling the use of an ARINC-based platform

Conclusions

- Development of an AFDX emulator
  - provides applications with Ada Interfaces
    - Configuration and Communication APIs
  - relies on standard ethernet hardware
    - thus enabling the use of an ARINC-based platform
- Integration with a real-time partitioned platform
  - XtratuM + MaRTE OS + Ada applications
  - Training/Educational platform with reasonable overheads

Conclusions

- Development of an AFDX emulator
  - provides applications with Ada Interfaces
    - Configuration and Communication APIs
  - relies on standard ethernet hardware
    - thus enabling the use of an ARINC-based platform
- Integration with a real-time partitioned platform
  - XtratuM + MaRTE OS + Ada applications
  - Training/Educational platform with reasonable overheads
- Software available at ***http://marte.unican.es***
  - it can be used with or without hypervisor

everything will be okay

in the end.

if it's not okay,

it's not the end.

(unknown)