Networking in Modern Avionics: Challenges and Opportunities

ECRTS 2011 July 6-8 Porto, Portugal

RTN'2011

July 5th, 2011

Porto, Portugal

Sérgio D. Penna

Paulo Franklin M. R. Ferrua

Today's presentation

The introduction of "Integrated Modular Avionics" (**IMA**) by the Radio Technical Commission for Aeronautics (RTCA DO-297) in November 2005 gave focus to new industry standards. "Avionics Full Duplex Switched Network" (ARInc 644 Part 7 "**AFDX**"), "Time-Triggered Protocol" (TTA Group "**TTP**") and "Application Executive interface" (ARInc 653 "**APEX**") emerged offering new levels of modularity and communality to avionic systems. These standards present new **challenges** for system manufacturers and integrators, but offer new **opportunities** to improve current analytical methods for predicting system behaviour during the design phase. **This presentation provides a quick overview of these important standards and addresses challenges and opportunities arising from their adoption.**

In the aircraft industry...

Things are getting very complicated!

Previously...

- “Federated Architectures” featured independent units hosting one function each;
- Lots of dedicated cabling;
- Each unit was specifically designed for the purpose;
- Standard form-factor (ARINC 600);
- Only a few standard serial buses (ARINC 429, MIL 1553, ARINC 629).

Source: ARTIST2 – Integrated Modular Avionics A380

More recently...

- RADIO TECHNICAL COMMISION FOR AERONAUTICS (RTCA). RTCA DO-297: Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations. Washington D.C., USA, November 2005;
- Variations, such as “Distributed Integrated Modular Avionics” (DIMA);
- Faster digital buses, standard (AFDX, TTP) and proprietary (ASCB-D, HSDB);
- Standard computer platforms;
- Generalized use of COTS technology (such as Ethernet/IP networking);
- Standard RTOS services interface (ARINC 653);
- Much less cabling, but...

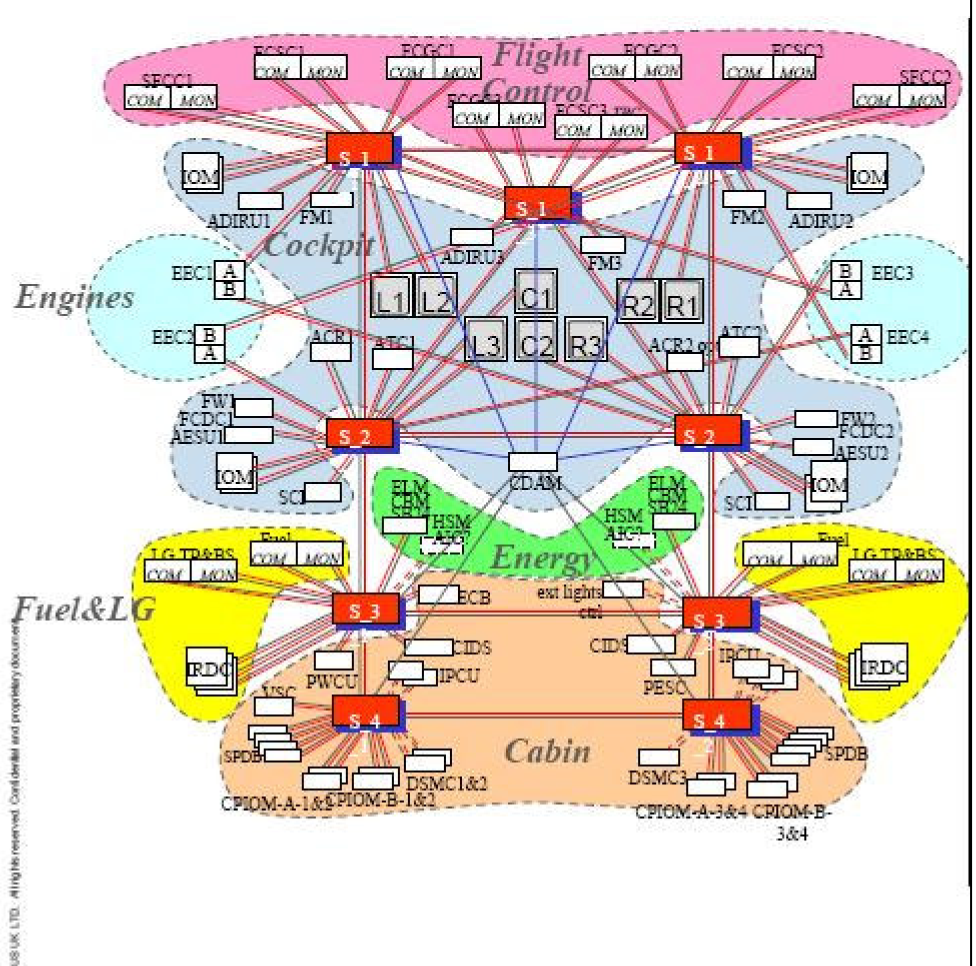
Not exactly less complex...

- In the AIRBUS A380:

AFDX Network:

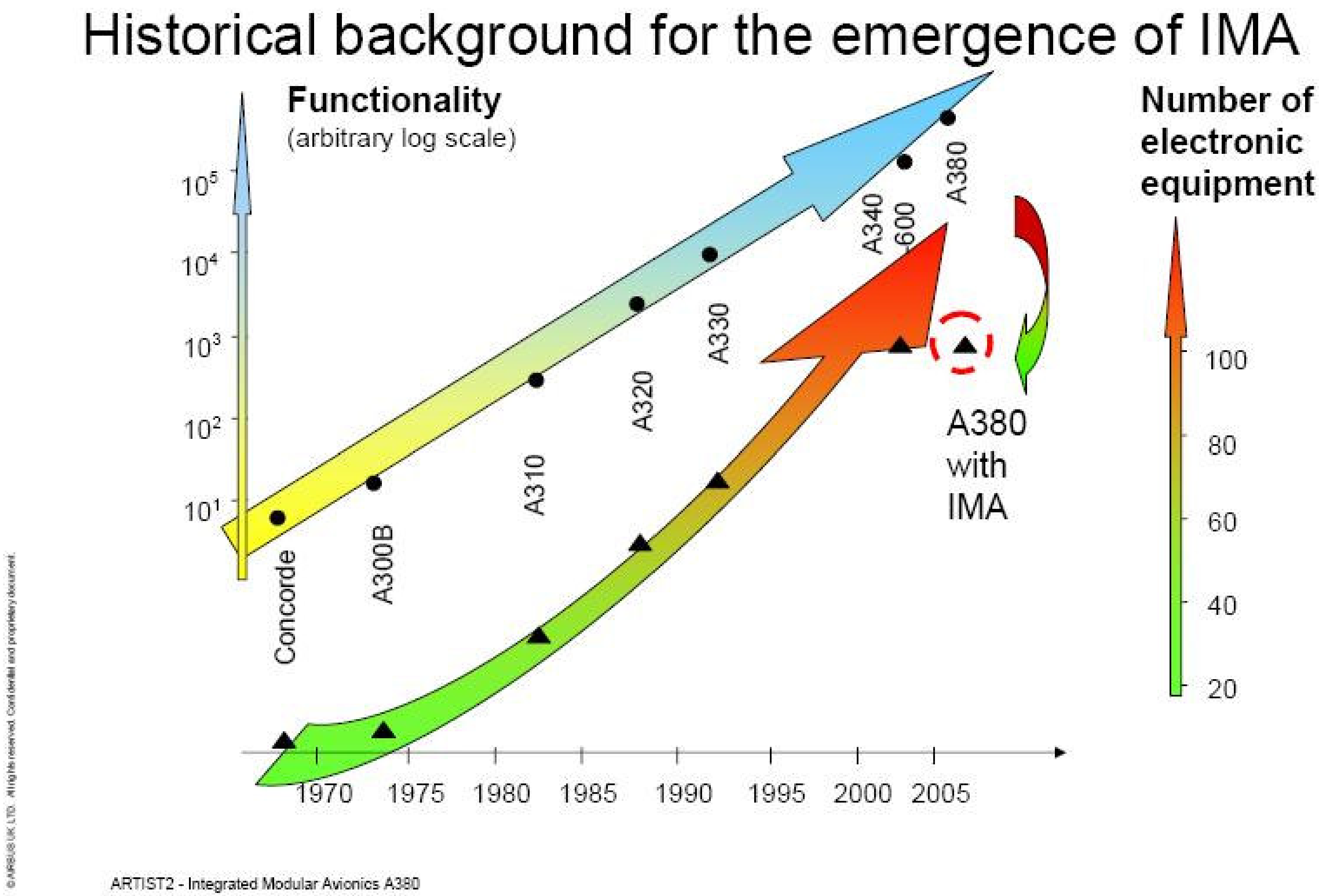
- 100 Mbits
- Redundant Network (A&B) with independent alimentation
- AFDX switches = 2 x 8
- NB of ports (connections) possible on each switch (20-24)
- MTBF of the switch is very high (100 000 hours expected)
- Up 80 AFDX subscriber

Source: ARTIST2 – Integrated Modular Avionics A380



ARTIST2 - Integrated Modular Avionics A380

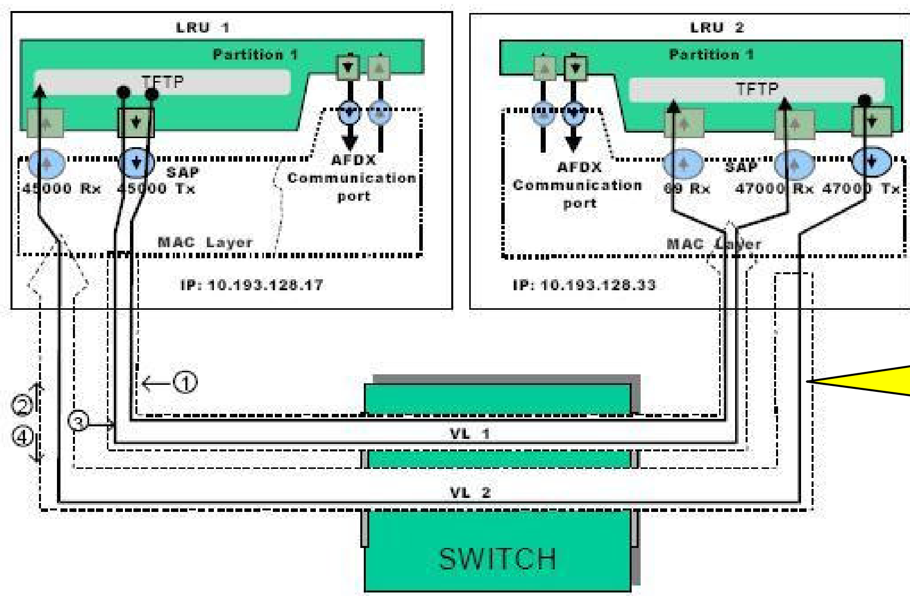
But why IMA?



Historical background for the emergence of IMA
Functionality
(arbitrary log scale)
10^5
10^4
10^3
10^2
10^1
Number of electronic equipment
100
80
60
40
20
Concorde
A300B
A310
A320
A330
A340 -600
A380
A380 with IMA
1970
1975
1980
1985
1990
1995
2000
2005
ARTIST2 - Integrated Modular Avionics A380
© AIRBUS UK LTD. All rights reserved. Confidential and proprietary document.

AFDX ("Avionics Full DupleX Switched Network")

- "End-System": a node in the AFDX (ARINC 664 Part 7) switched network;
- "Virtual Link" (VL): an unidirectional virtual communication channel;
- "AFDX Port": a virtual construct used for transmitting/receiving messages.
- "Bandwidth Allocation GAP" (BAG) and "Lmax": parameters used for network Traffic Shaping;
- The "AFDX Switch" is responsible for Traffic Policing and requires programming for maximum allowed jitter "Jswitch" in message inter-arrival time.

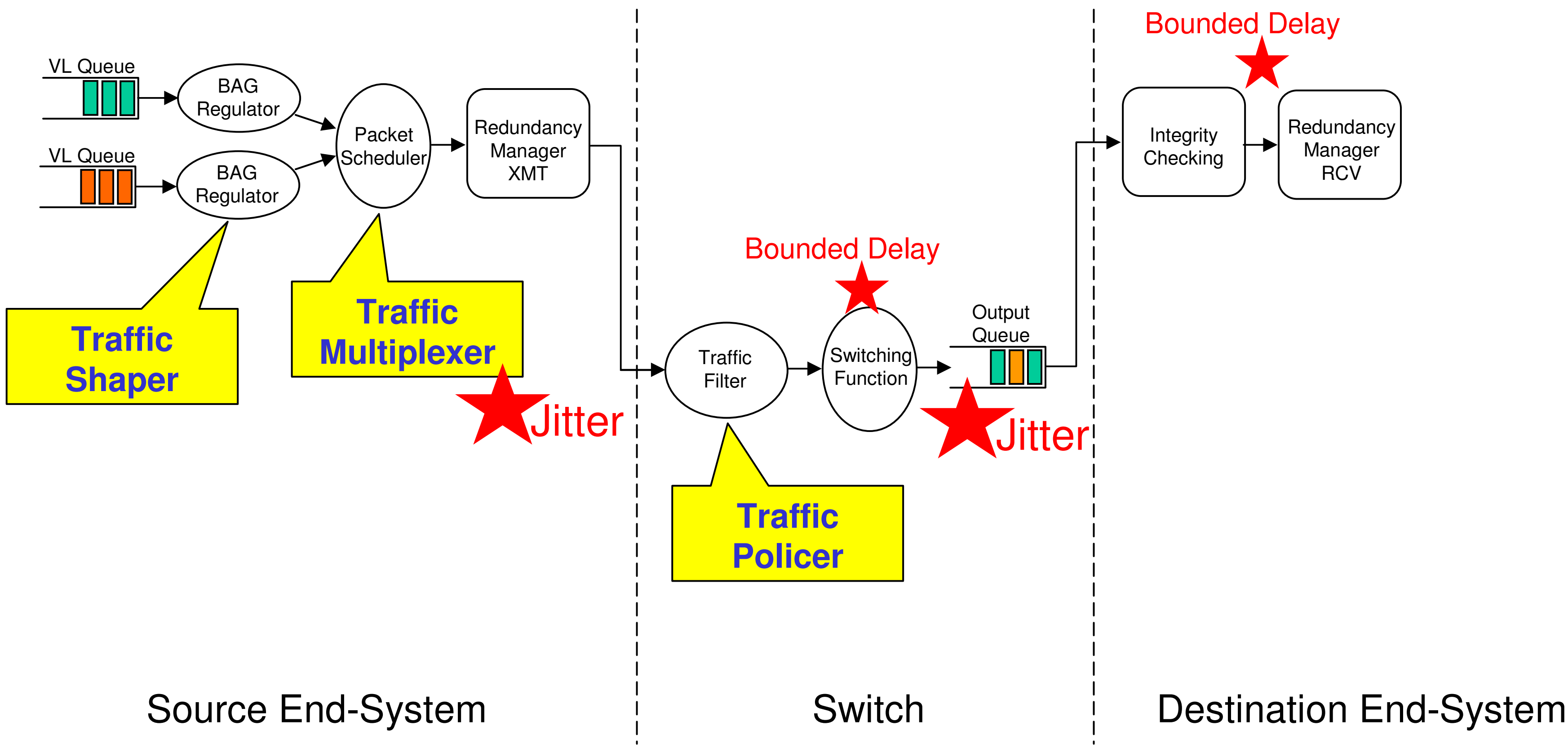


Physical medium is a full-duplex, dual-channel Ethernet 100Mbit/s

Source: ARINC SPECIFICATION 664 PART 7 - Page 35

AFDX Network

- The AFDX network is a combination of several components performing specific functions;
- Some of them introduce transmission jitter, others can be represented by a “bounded delay”.



AFDX Network (cont.)

- There can be more than one VL per physical link;
- VLs are made part (lower 16-bits) of the Ethernet Destination MAC Address which are inherently multicast (first byte is “xxxx xx11”);
- There can be a FIFO queue of “Sub-VLs” per VL;
- BAG can assume only a few values 2^k, k=[0,7], although smaller values are commonly adopted by system designers (e.g. 500µsec);
- AFDX switches usually follow the “store-forward”, “shared memory” architecture;
- AFDX switches can have two-priority queues per output port;
- IP fragmentation is allowed, although for just one ARINC 653 port type (“queueing port”);
- AFDX frames carry a “Sequence Number” (SN) in the last byte of the IP payload allowing the Redundancy Manager to discard duplicated frames.

AFDX Network (cont.)

- There is a strong connection between AFDX and ARINC 653:

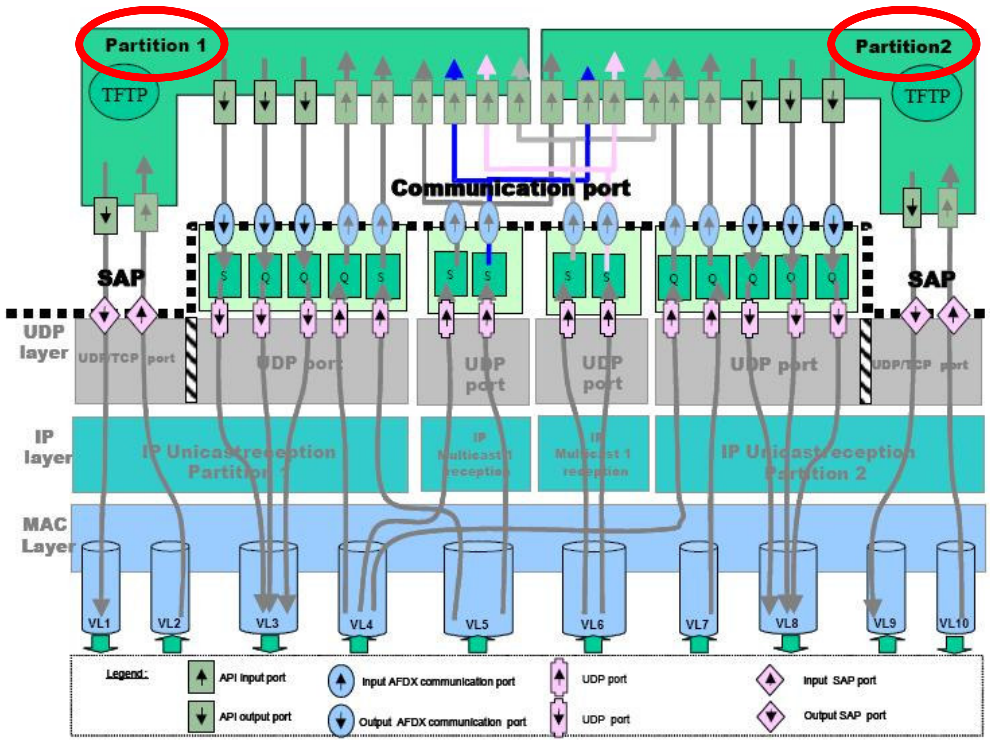


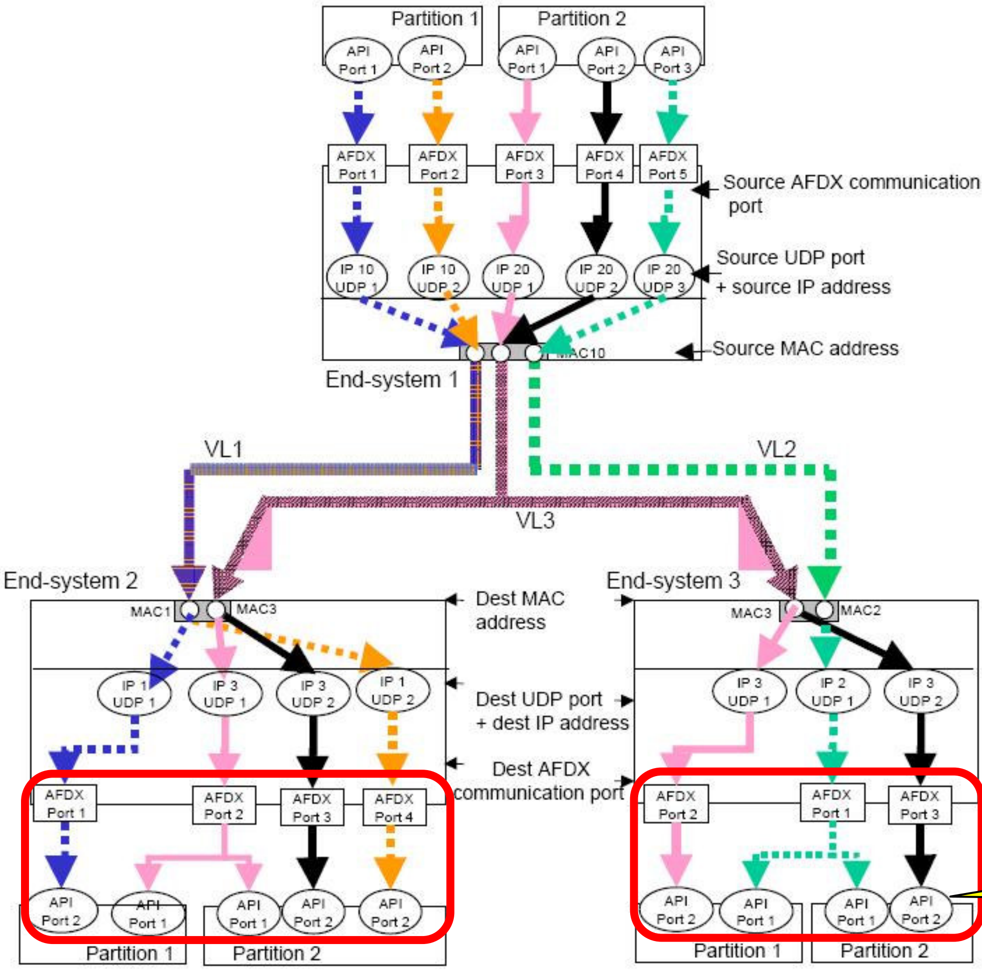
Figure 3-23 - Interface Between Partition and End System

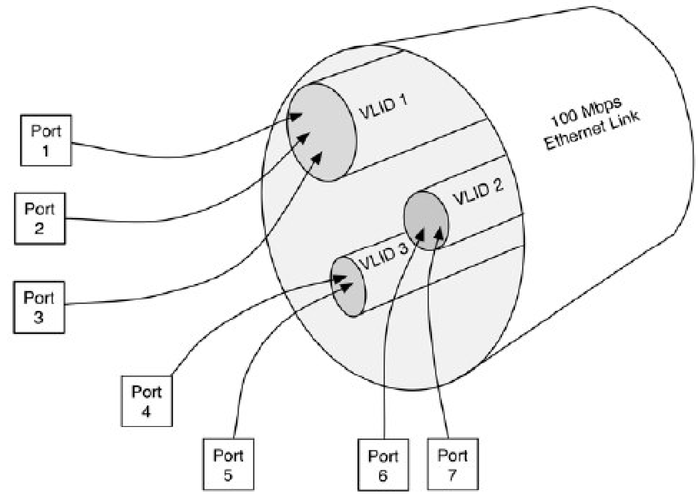
Figure 3-23 describes equipment which has two partitions (e.g., ARINC 653 for a definition of a partition) and an End System. Each partition has an IP address. To communicate with a partition, the End System uses two port types: Communication Port and SAP.

Source: ARINC SPECIFICATION 664 PART 7 - Page 29

AFDX Network (cont.)

- Sample addressing scheme:





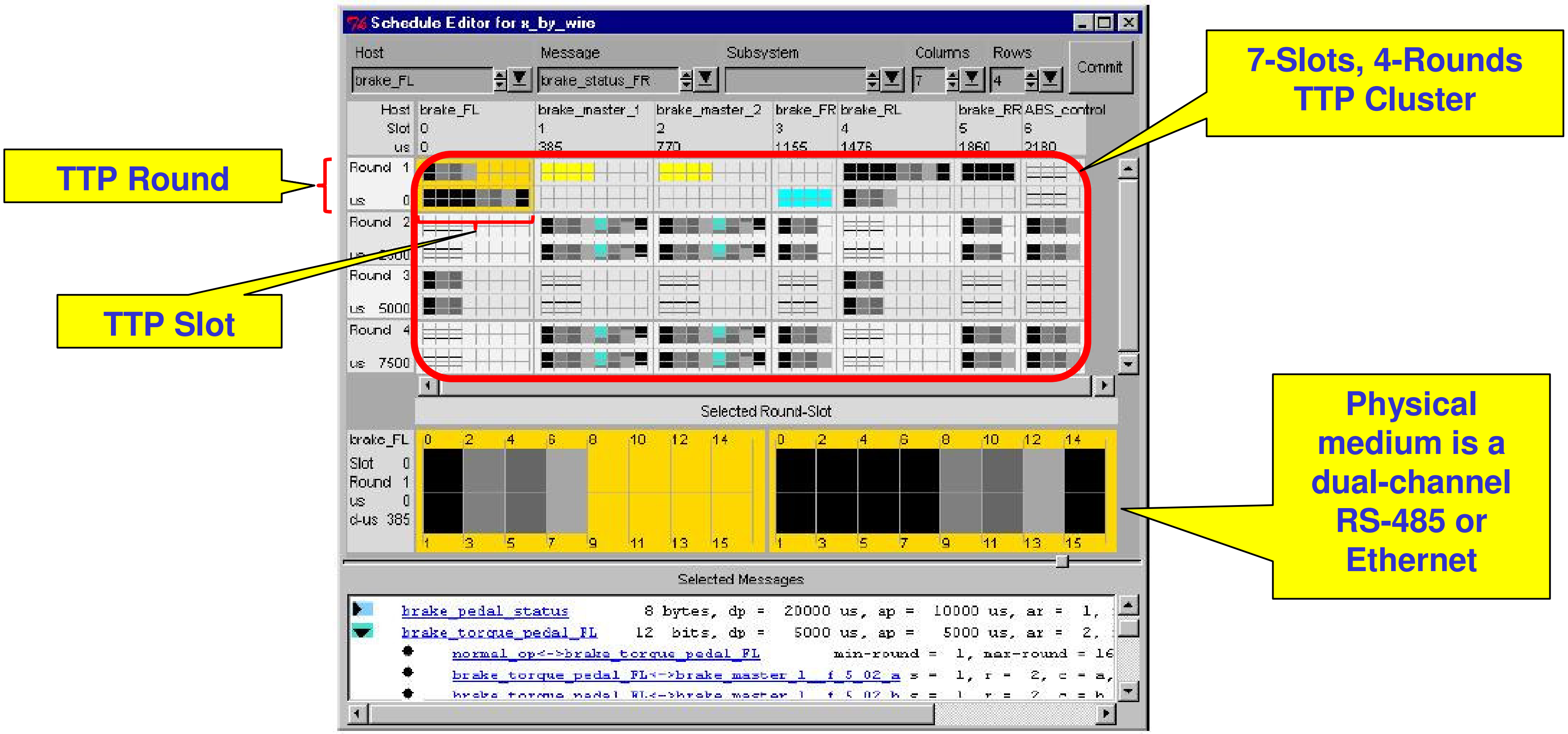
Source: Condor Engineering, Inc. - AFDX / ARINC 664 Tutorial - Page 21

AFDX-to-ARINC 653 port mapping is implementation specific

Source: ARINC SPECIFICATION 664 PART 7 - Page 38

TTP ("Time-Triggered Protocol" – SAE AS6003)

- Communication is done using "Time-Division Multiple Access" (TDMA);
- Each node has a reserved transmission "Slot" in a "Round";
- Many "Rounds" form a "TTP Cluster".



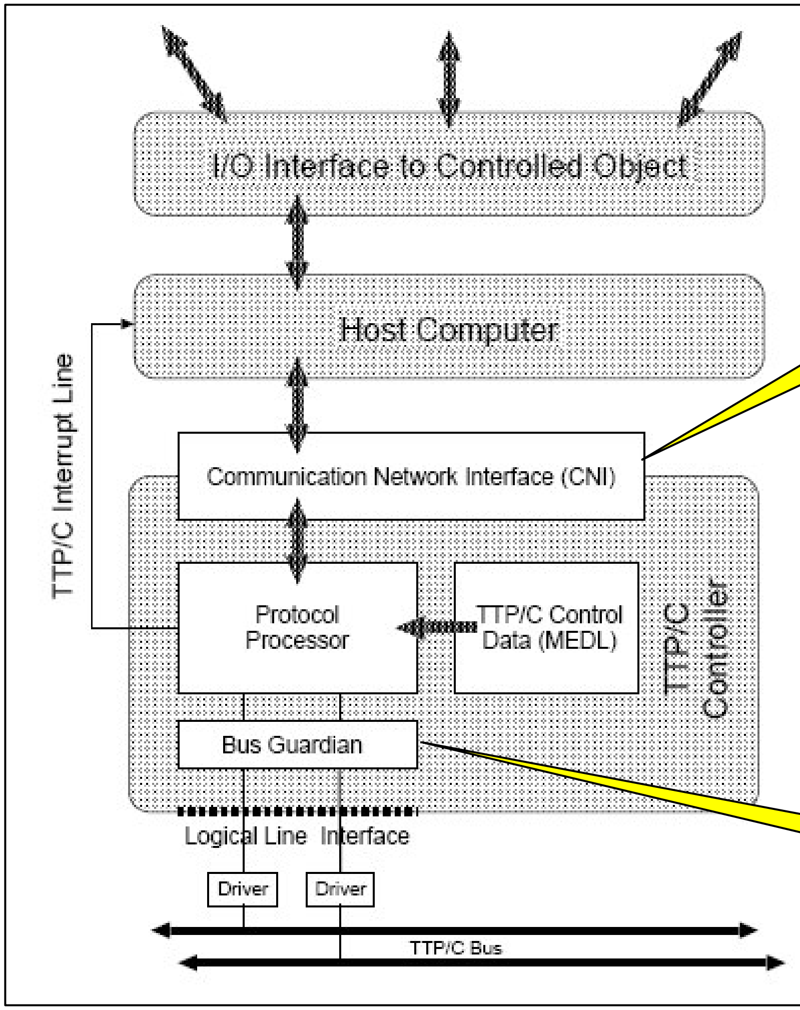
Source: TTTech Computer Technik AG

TTP Network

- TTP Rounds are usually short in time (n x 1ms) for Flight Control applications;
- TTP Clusters are equally short (n x 10ms);
- Messages (called "frames") in each transmission slot should be maximum 240 bytes long;
- Before network start-up, the "Message Description List" (MEDL) is loaded in each participant node containing information about transmissions of all nodes;
- Time synchronization is achieved using a clock correction algorithm based on a work by LUNDELIUS (later WELCH)-LYNCH "A new fault-tolerant algorithm for clock synchronization" (also called "Fault-Tolerant Midpoint" - FTM) published in 1984 and adapted by KOPETZ-OCHSENREITER in 1987 ("Fault-Tolerant Average" - FTA);
- The TTP network physical layer has two independent channels, which can be optionally used in redundancy;
- Network monitoring can be done using a passive, non-transmitting node;
- "Bus", "Star" and a combination of the two are supported network topologies.

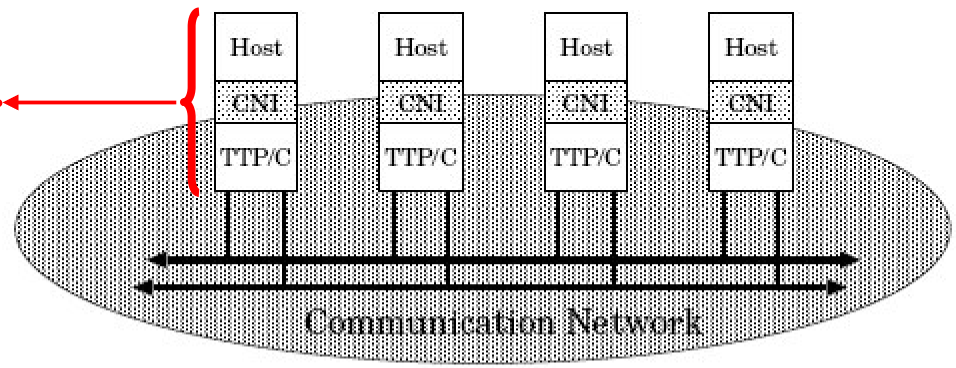
TTP Node

- The key element is the “TTP controller”:



Source: TTP/C Specification Version 1.4.3 - Page 22

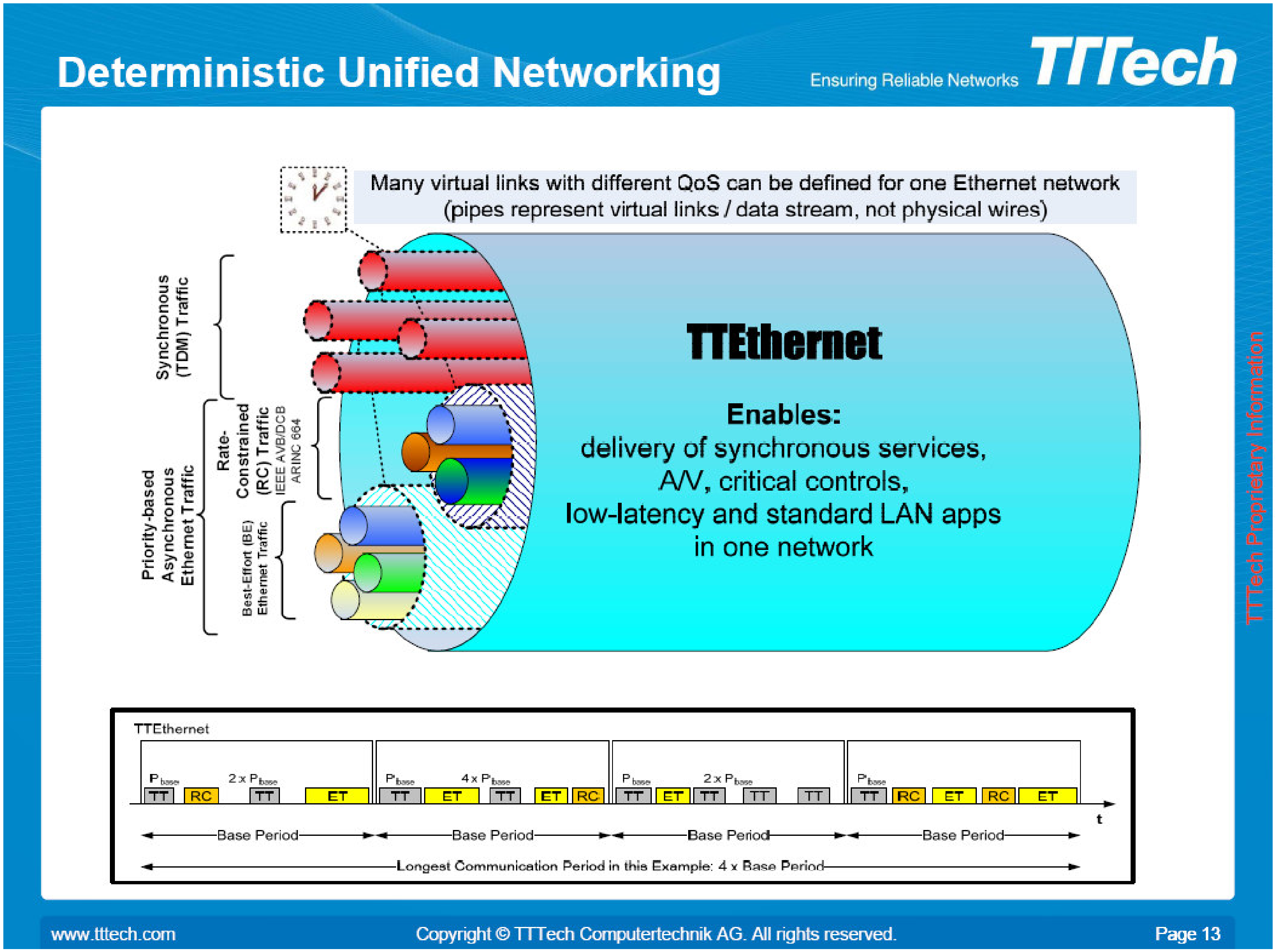
The CNI is a dual-port memory



Source: TTP/C Specification Version 1.4.3 - Page 19

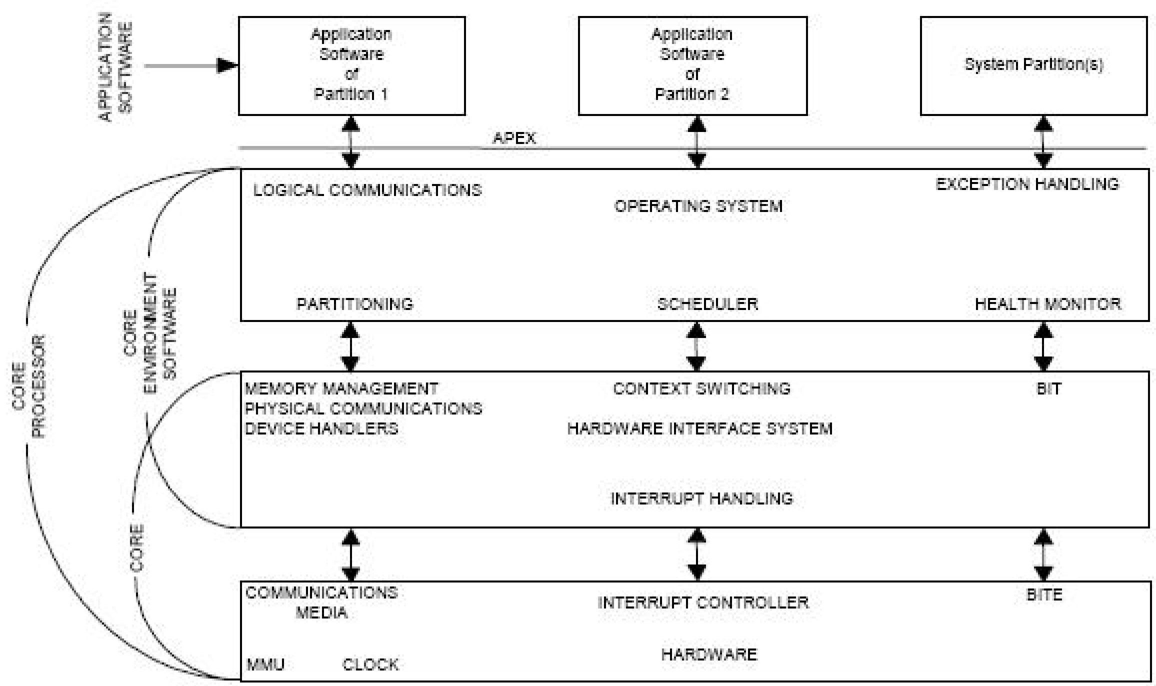
The “Bus Guardian” blocks medium access outside the node assigned Round-Slot

TT-Ethernet ("Time-Trigered Ethernet" – SAE AS6802)



ARINC 653 ("APplication/EXecutive")

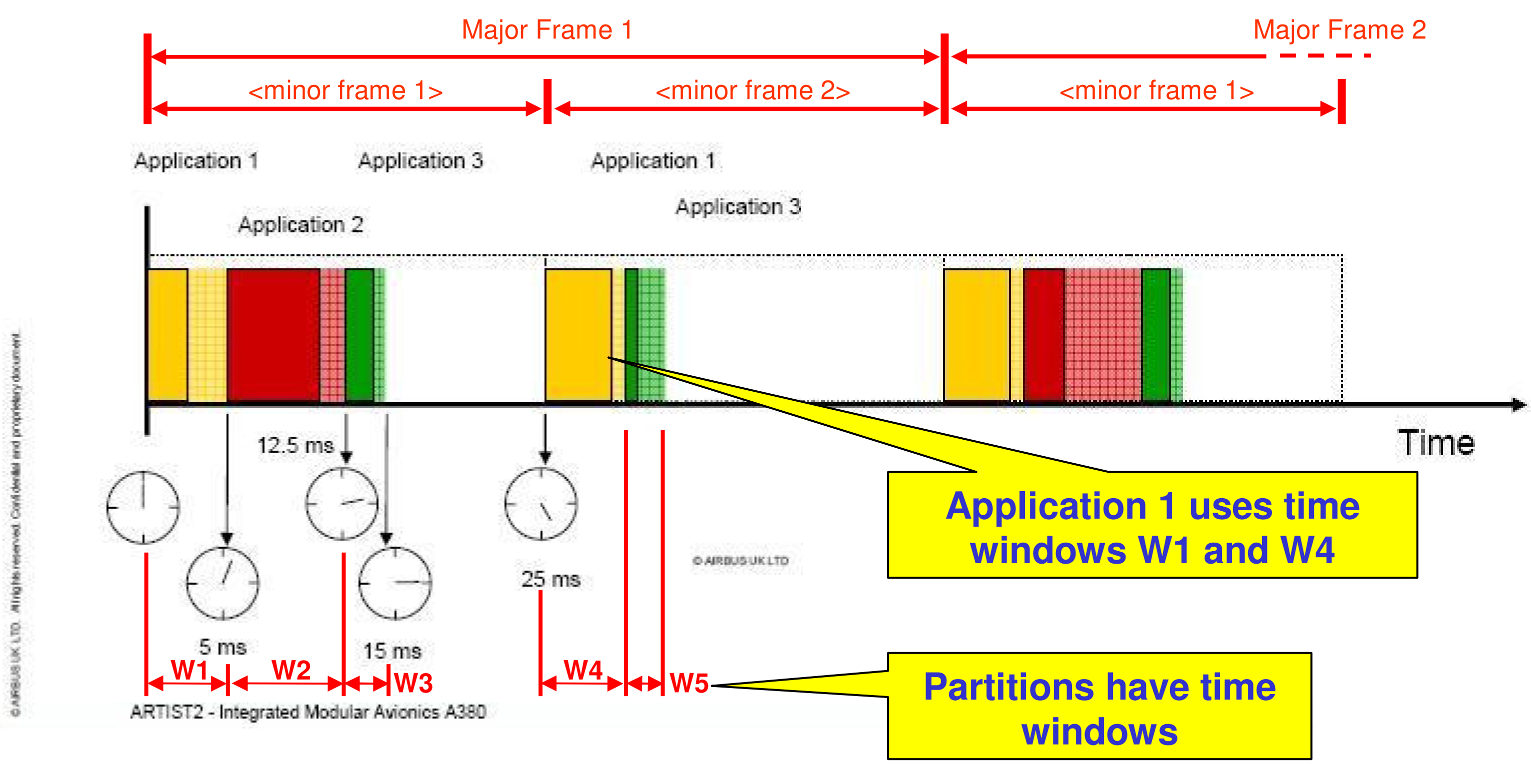
- "The primary objective of this Specification is to define a general-purpose APEX (APplication/EXecutive) interface between the Operating System (O/S) of an avionics computer resource and the application software" (Source: ARINC SPECIFICATION 653, PART 1 – Page 1)



Source: ARINC SPECIFICATION 653, PART 1 – Page 11

ARINC 653 (cont.)

- “Temporal Partitioning”: tasks run in partitions which are scheduled according to a fixed periodic scheduling configured at system start-up.
- “The major time frame (MAF) is defined as a multiple of the least common multiple of all partition periods in the module”.



ARINC 653 (cont.)

- The standard has a XML Schema defining partitions, time windows and other elements:

ARINC SPECIFICATION 653, PART 1 – Page 198

APPENDIX H
ARINC 653 XML-SCHEMA

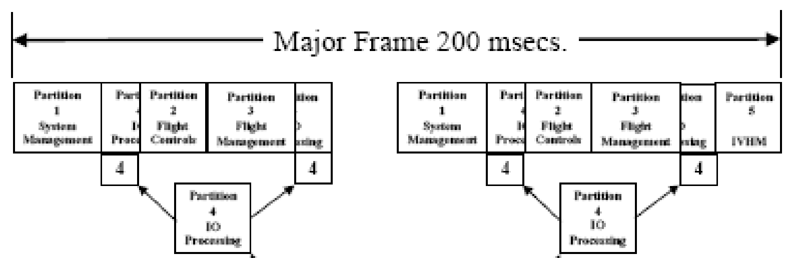

```
        <xs:documentation>The scheduling requirements for the module</xs:documentation>
      </xs:annotation>
      <xs:complexType>
        <xs:sequence>
          <xs:element name="Partition_Schedule" maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>The scheduling requirements for partitions within the
module.</xs:documentation>
            </xs:annotation>
            <xs:complexType>
              <xs:sequence>
                <xs:element name="Window_Schedule" maxOccurs="unbounded">
                  <xs:annotation>
                    <xs:documentation>The allocation of the partition to partition windows wi
frame.</xs:documentation>
                  </xs:annotation>
                  <xs:complexType>
                    <xs:attribute name="WindowIdentifier" type="DecOrHexValueType" use=
                    <xs:attribute name="WindowStartSeconds" type="xs:float" use="requirec
                    <xs:attribute name="WindowDurationSeconds" type="xs:float" use="requ
                    <xs:attribute name="PartitionPeriodStart" type="xs:boolean" default="fals
                  </xs:complexType>
                </xs:element>
                <xs:any namespace="Window_Sched_Ext" minOccurs="0" maxOccurs="unboun
                  <xs:annotation>
                    <xs:documentation>Implementer unique extensions.</xs:documentation
                  </xs:annotation>
                </xs:any>
              </xs:sequence>
              <xs:attribute name="PartitionIdentifier" type="IdentifierValueType" use="required"/>
              <xs:attribute name="PartitionName" type="NameType" use="optional"/>
              <xs:attribute name="PeriodSeconds" type="xs:float" use="required"/>
              <xs:attribute name="PeriodDurationSeconds" type="xs:float" use="required"/>
            </xs:complexType>
          </xs:element>
          <xs:any namespace="Part_Sched_Ext" minOccurs="0" maxOccurs="unbounded">
            <xs:annotation>
              <xs:documentation>Implementer unique extensions.</xs:documentation>
            </xs:annotation>
          </xs:any>
        </xs:sequence>
        <xs:attribute name="MajorFrameSeconds" type="xs:float" use="required"/>
      </xs:complexType>
```


ARINC SPECIFICATION 653, PART 1 – Page 206

APPENDIX I
EXAMPLE ARINC 653 XML INSTANCE FILE

Partition Window Requirements for XML Example

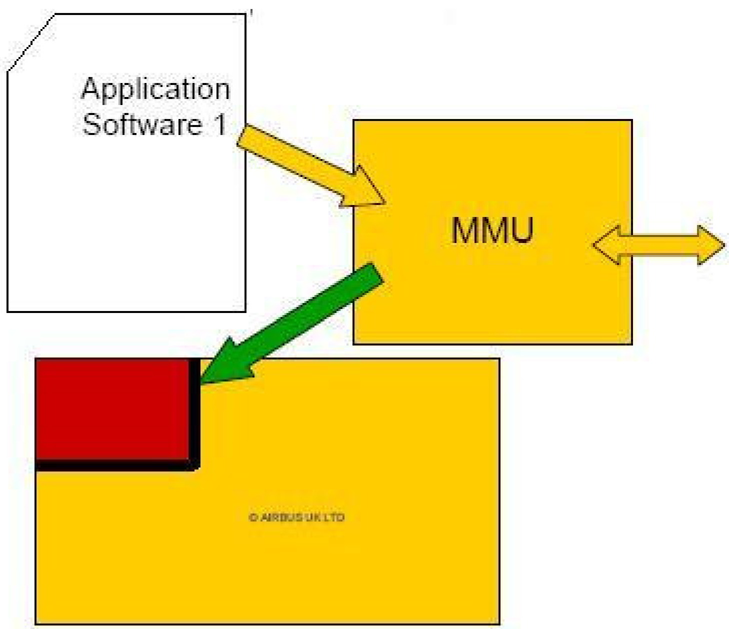
Window ID	1.1	4.1	2.1	3.1	4.2	1.2	4.3	2.2	3.2	4.4	5.1
Partition	P1	P4	P2	P3	P4	P1	P4	P2	P3	P4	P5
window offset	0.000	0.020	0.030	0.040	0.070	0.100	0.120	0.130	0.140	0.170	0.180
window duration	0.020	0.010	0.010	0.030	0.010	0.020	0.010	0.010	0.030	0.010	0.020



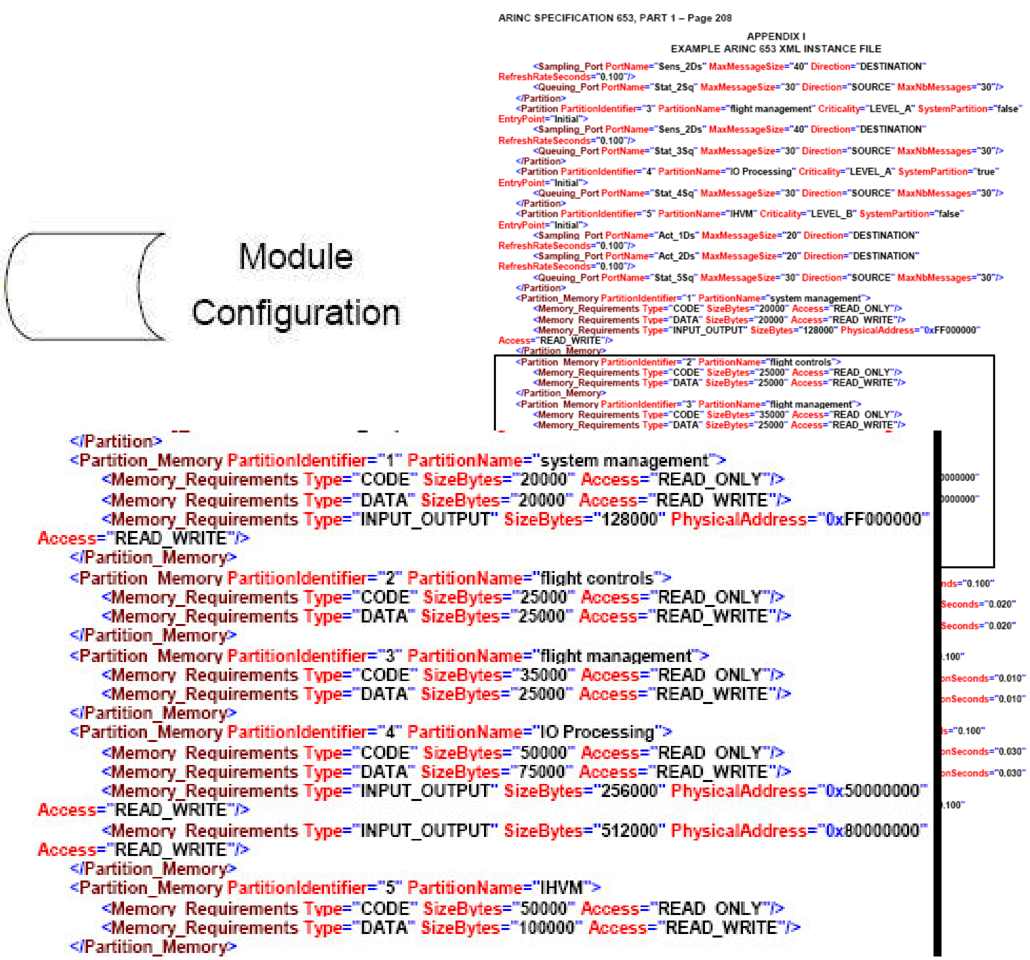
Partition 4 has two windows for each period and 4 windows within the major frame. Partitions 1,2 and 3 have two windows and partition 5 has one window within the major frame.

ARINC 653 (cont.)

- “Spatial Partitioning”: each partition has its own reserved virtual memory space configured at system start-up.



Source: ARTIST2 – Integrated Modular Avionics A380



ARINC SPECIFICATION 653, PART 1 – Page 208

APPENDIX I
EXAMPLE ARINC 653 XML INSTANCE FILE

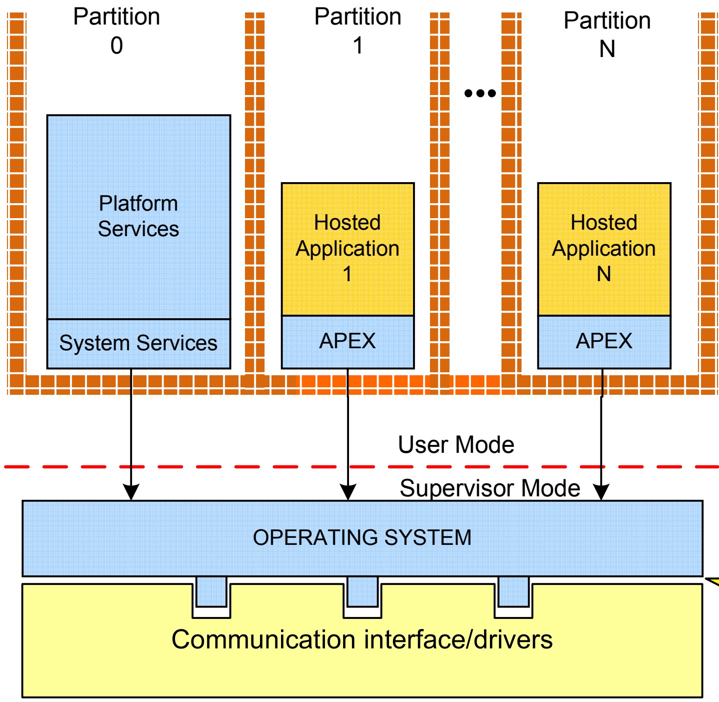

```
      <Sampling_Port PortName="Sens_2Ds" MaxMessageSize="40" Direction="DESTINATION"
RefreshRateSeconds="0.100"/>
      <Queuing_Port PortName="Stat_2Sq" MaxMessageSize="30" Direction="SOURCE" MaxNbMessages="30"/>
    </Partition>
    <Partition PartitionIdentifier="3" PartitionName="flight management" Criticality="LEVEL_A" SystemPartition="false"
EntryPoint="Initial">
      <Sampling_Port PortName="Sens_2Ds" MaxMessageSize="40" Direction="DESTINATION"
RefreshRateSeconds="0.100"/>
      <Queuing_Port PortName="Stat_3Sq" MaxMessageSize="30" Direction="SOURCE" MaxNbMessages="30"/>
    </Partition>
    <Partition PartitionIdentifier="4" PartitionName="IO Processing" Criticality="LEVEL_A" SystemPartition="true"
EntryPoint="Initial">
      <Queuing_Port PortName="Stat_4Sq" MaxMessageSize="30" Direction="SOURCE" MaxNbMessages="30"/>
    </Partition>
    <Partition PartitionIdentifier="5" PartitionName="IHVM" Criticality="LEVEL_B" SystemPartition="false"
EntryPoint="Initial">
      <Sampling_Port PortName="Act_1Ds" MaxMessageSize="20" Direction="DESTINATION"
RefreshRateSeconds="0.100"/>
      <Sampling_Port PortName="Act_2Ds" MaxMessageSize="20" Direction="DESTINATION"
RefreshRateSeconds="0.100"/>
      <Queuing_Port PortName="Stat_5Sq" MaxMessageSize="30" Direction="SOURCE" MaxNbMessages="30"/>
    </Partition>
    <Partition_Memory PartitionIdentifier="1" PartitionName="system management">
      <Memory_Requirements Type="CODE" SizeBytes="20000" Access="READ_ONLY"/>
      <Memory_Requirements Type="DATA" SizeBytes="20000" Access="READ_WRITE"/>
      <Memory_Requirements Type="INPUT_OUTPUT" SizeBytes="128000" PhysicalAddress="0xFF000000"
Access="READ_WRITE"/>
    </Partition_Memory>
    <Partition_Memory PartitionIdentifier="2" PartitionName="flight controls">
      <Memory_Requirements Type="CODE" SizeBytes="25000" Access="READ_ONLY"/>
      <Memory_Requirements Type="DATA" SizeBytes="25000" Access="READ_WRITE"/>
    </Partition_Memory>
    <Partition_Memory PartitionIdentifier="3" PartitionName="flight management">
      <Memory_Requirements Type="CODE" SizeBytes="35000" Access="READ_ONLY"/>
      <Memory_Requirements Type="DATA" SizeBytes="25000" Access="READ_WRITE"/>
```



```
    </Partition>
    <Partition_Memory PartitionIdentifier="1" PartitionName="system management">
      <Memory_Requirements Type="CODE" SizeBytes="20000" Access="READ_ONLY"/>
      <Memory_Requirements Type="DATA" SizeBytes="20000" Access="READ_WRITE"/>
      <Memory_Requirements Type="INPUT_OUTPUT" SizeBytes="128000" PhysicalAddress="0xFF000000"
Access="READ_WRITE"/>
    </Partition_Memory>
    <Partition_Memory PartitionIdentifier="2" PartitionName="flight controls">
      <Memory_Requirements Type="CODE" SizeBytes="25000" Access="READ_ONLY"/>
      <Memory_Requirements Type="DATA" SizeBytes="25000" Access="READ_WRITE"/>
    </Partition_Memory>
    <Partition_Memory PartitionIdentifier="3" PartitionName="flight management">
      <Memory_Requirements Type="CODE" SizeBytes="35000" Access="READ_ONLY"/>
      <Memory_Requirements Type="DATA" SizeBytes="25000" Access="READ_WRITE"/>
    </Partition_Memory>
    <Partition_Memory PartitionIdentifier="4" PartitionName="IO Processing">
      <Memory_Requirements Type="CODE" SizeBytes="50000" Access="READ_ONLY"/>
      <Memory_Requirements Type="DATA" SizeBytes="75000" Access="READ_WRITE"/>
      <Memory_Requirements Type="INPUT_OUTPUT" SizeBytes="256000" PhysicalAddress="0x50000000"
Access="READ_WRITE"/>
      <Memory_Requirements Type="INPUT_OUTPUT" SizeBytes="512000" PhysicalAddress="0x80000000"
Access="READ_WRITE"/>
    </Partition_Memory>
    <Partition_Memory PartitionIdentifier="5" PartitionName="IHVM">
      <Memory_Requirements Type="CODE" SizeBytes="50000" Access="READ_ONLY"/>
      <Memory_Requirements Type="DATA" SizeBytes="100000" Access="READ_WRITE"/>
    </Partition_Memory>
```

ARINC 653 Input-Output

- The standard provide services for inter-partition communication (“sampling” and “queueing ports”).
- It suggests a mechanism for abstracting physical I/O (“Pseudo Partition”), but lack of details leave implementations free for choosing different paths.

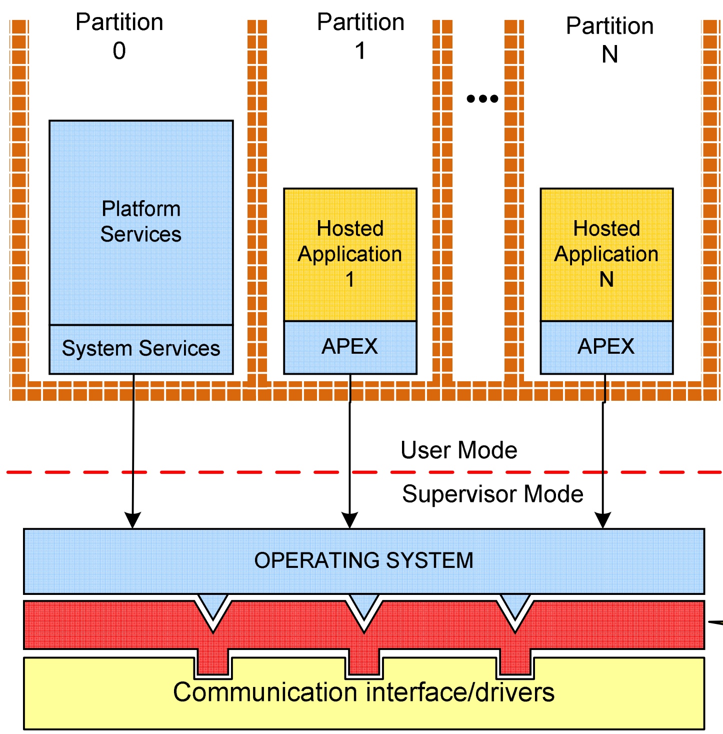


The concept of “ports” (“sampling” and “queueing”), as defined by ARINC653, fits perfectly to the Ethernet/IP/UDP transport protocol used by AFDX

ARINC 653 Input-Output (cont.)

EMBRAER

- The concept of "ports" does not fit well for avionic buses other than AFDX (such as ARINC 429, ARINC 629)…



Source: ARINC SPECIFICATION 653, PART 2 – Page 15

2.3.2.8.1 ARINC 429

Table 2.3.2.8.1-1 – ARINC 429 Mapping

Element	ARINC 653 Mapping	Notes
Octal Label/SDI	Use for port mapping	Note 1
SSM	Status mapped to FS Sign mapped to parameter sign	Note 2
BNR Data	Map to integer or float	Note 3
Discrete Data	Map to 32-bit discrete data block	Note 4
BCD		Note 5

Notes:

1. Use of only octal label and SDI may not provide sufficient information for unambiguous mapping. Physical port, equipment ID, or other means may need to be included.

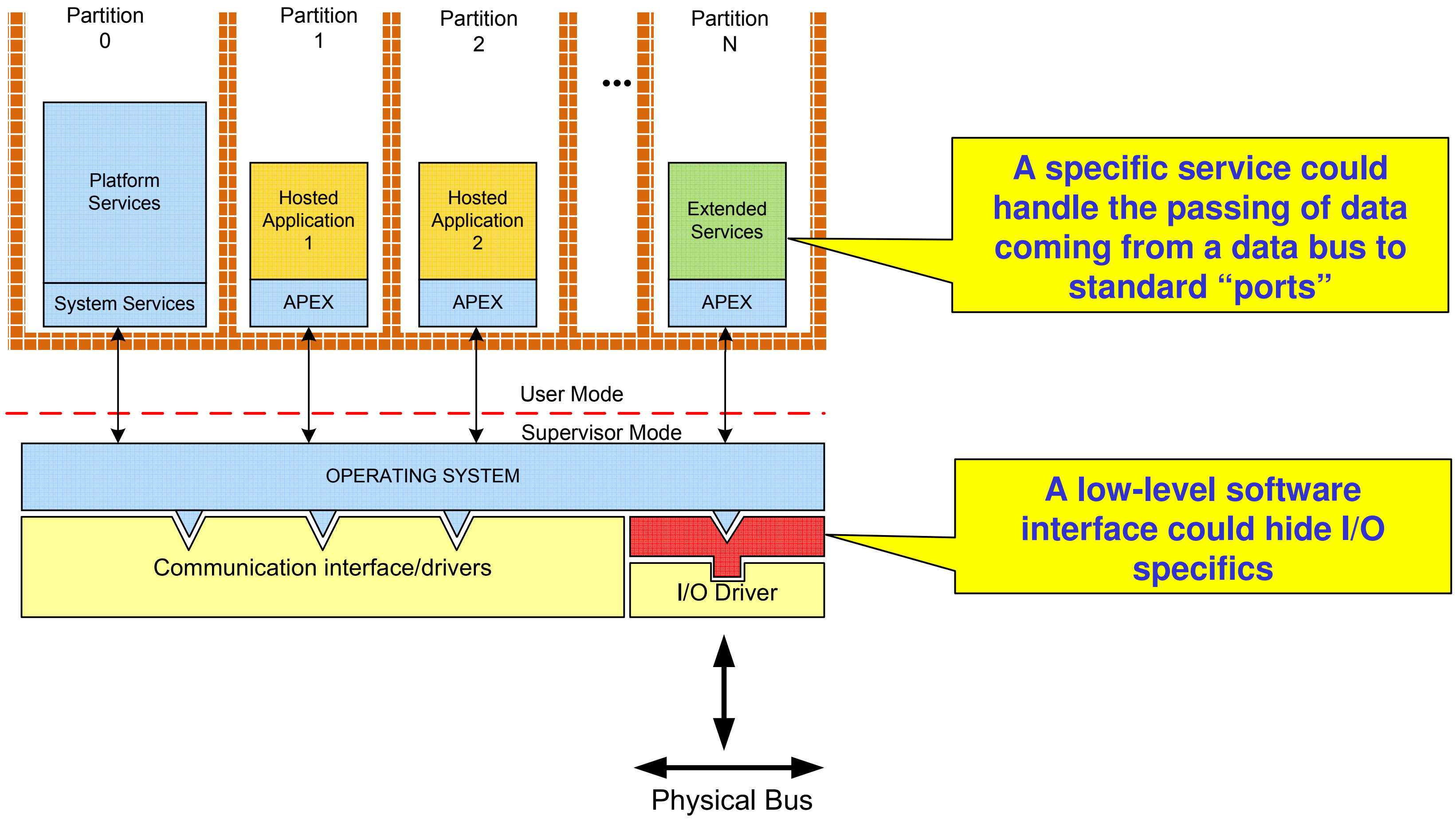
f the ARINC … lue should be … n, below). … INC 429 is … presentation

Input-Output of bus data such as ARINC 429 or ARINC 629 has to be handled below the APEX layer

Perhaps an abstraction layer to physical I/O could help...

ARINC 653 Input-Output (cont.)

- For buses other than AFDX such as ARINC 429, one possible solution could be…



ARINC 653 Sample Source-Code

sample.c

ARINC 653 and TTEthernet

TTECOM Layer ARINC 653 for VxWorks 653

The TTECOM Layer ARINC 653 integrated in Wind River's robust operating system VxWorks 653 for controlling ARINC 653 Integrated Modular Avionics (IMA) systems offers a powerful interface to applications running on TTEthernet-based networks. TTEthernet network technology enables high-bandwidth deterministic high-integrity real-time communication. TTEthernet integrates time-triggered, rate-constraint (ARINC 664 part 7), and COTS Ethernet traffic flows within one physical infrastructure.

KEY FEATURES/BENEFITS

- Integration of TTEthernet communication with proven Wind River's VxWorks 653 time and space partitioning RTOS
- Deterministic system communication using ARINC 653 API (APEX interface) sampling and queueing ports
- Enables growth path towards distributed IMA and mixed-criticality architectures

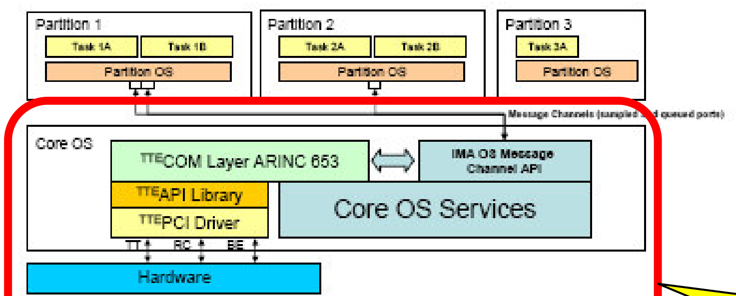


Figure 1: TTECOM Layer ARINC 653

Data bus and vendor specific

Application Areas

System and application design investment protection

Source: TTTech's Flyer on "TTECOM Layer ARINC\ 653 for VxWorks 653"

Challenges for System Designers

- AFDX network configuration:
  - How many Virtual Links? How many AFDX Switches? What Bandwidth Allocation Gaps (BAGs) to choose? Should IP fragmentation be used?
- TTP Cluster configuration:
  - How many Slots? How many Rounds? How big the messages should be? Should channel multiplexing be used?
- ARINC 653 processing load distribution:
  - How many processing modules? How many partitions per module? How long should be the duration of each partition?
- Task synchronization:
  - Should task sets running in different processor modules be synchronized?
- Time synchronization:
  - Is this a requirement? What protocol should be used? Hardware or software supported?

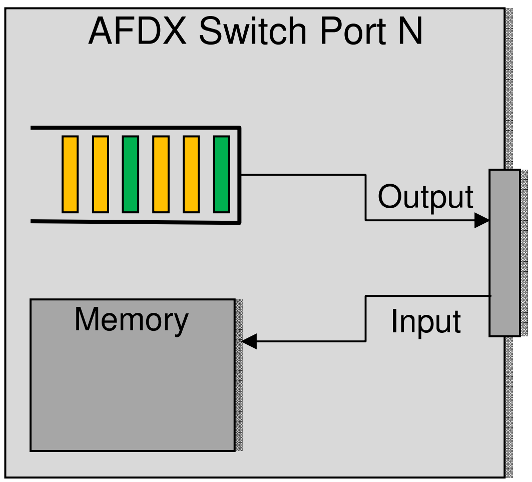
Opportunities for Researchers

- Extension of current analysis methods:
  - “Timing Analysis” should be able to analyse the entire distributed system, from task scheduling in processing modules to delays in message transmission and reception.
- Configuration and optimization tools:
  - The bigger the system, the more automation should be in place for configuring and optimizing it (a lengthy analysis process is never practical).
- Further studies on time synchronization in distributed systems:
  - Should hardware support for time synchronization always be in place?
  - How good time synchronization using exclusively software should be?
- Open issues in the industry standards:
  - ARINC 653 provides little guidance for “extra-cabinet” data communication other than AFDX (“Pseudo Partition”).
  - AFDX sometimes is used simply as a Transport Protocol (data type “opaque”).

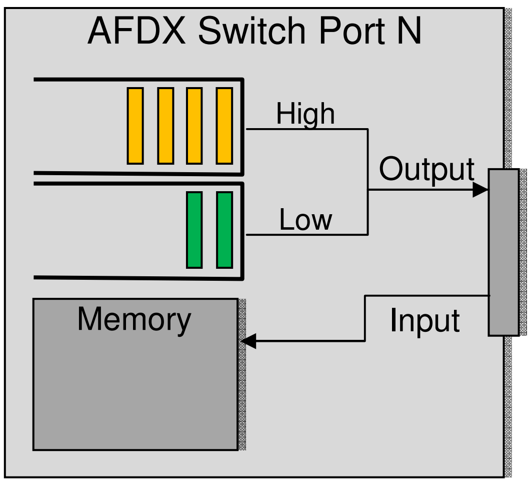
More than one solution...

- AFDX Switch output queue:

this...



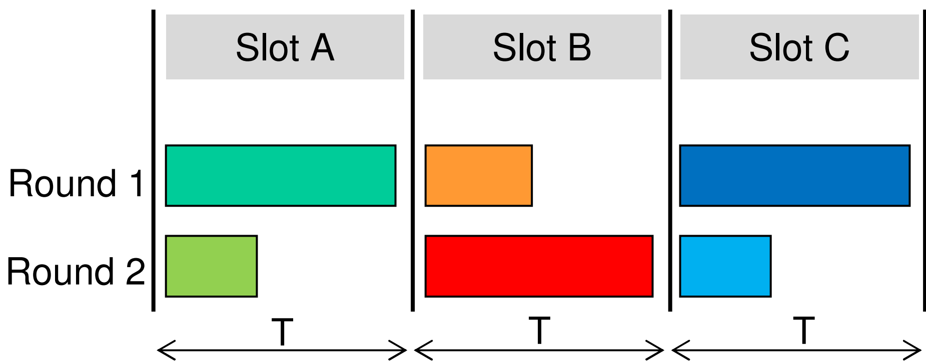
or this...



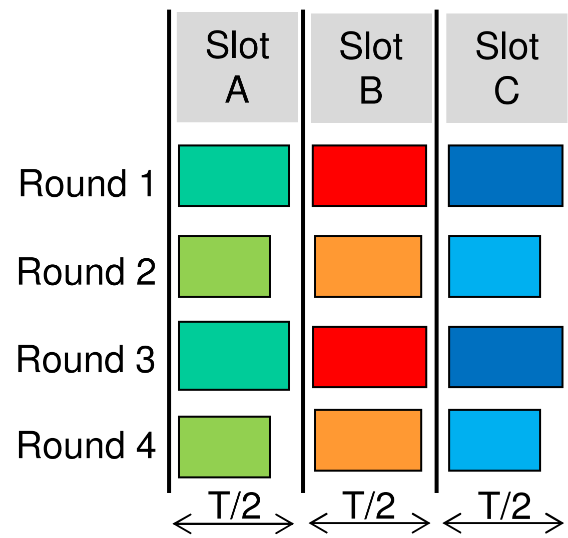
More than one solution...

- TTP Cluster design:

this...



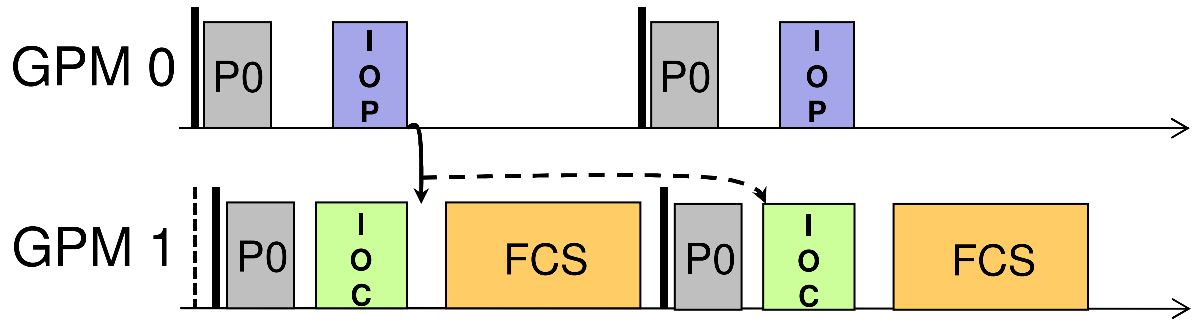
or this...



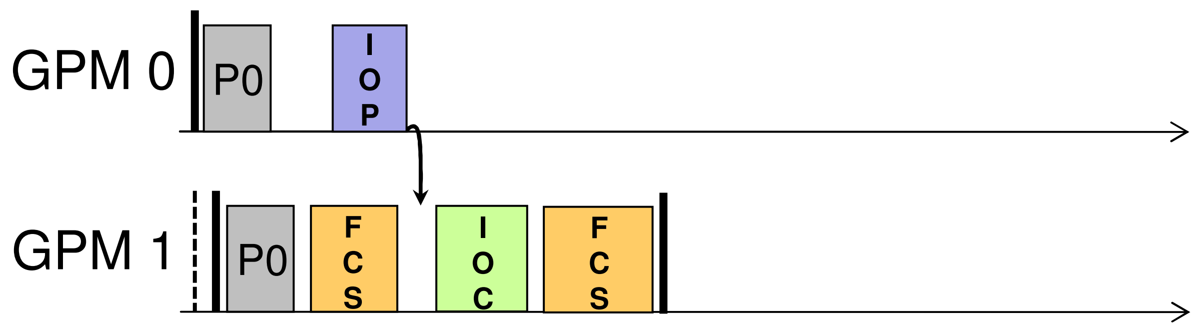
More than one solution...

- ARINC 653 Partition design:

this...



or this...



Conclusion

- To Do:
  - Composition of AFDX, TTP and ARINC 653, extending current “Holistic Timing Analysis” methods for evaluating task WCRT/BCRT and message transmission jitter;
  - Include “Data Aging” in the analysis;
  - “Fill-in-the-blanks” where ARINC 653 left physical I/O unmapped (materialize “Pseudo Partitions”);
  - Formally evaluate and recommend standard Time Synchronization Protocols for distributed systems such as NTP and IEEE 1588 in avionics applications;
  - Could a “Virtual Machine Hypervisor” be an alternative to ARINC 653?

Acknowledgements

- Part of the AFDX and ARINC 653 material used in this presentation was extracted from a presentation prepared by M. Jean-Bernard ITIER for the ARTIST2 project:
  - http://www.artist-embedded.org/docs/Events/2007/IMA/Slides/ARTIST2_IMA_Itier.pdf
- Part of the AFDX material was extracted from a tutorial by Condor Engineering, Inc. (today GE Intelligent Platforms).
- Other ARINC material was extracted directly from PDF documents published by each standard's committee.
- TTP material was extracted from TTA Group's "TTP Specification Version 1.4.3" (19-Nov-2003), TTTech's "TTP Plan" manual;
- TTEthernet material was extract from TTTech's presentation "Deterministic Ethernet for Aerospace and Defense Applications" (webinar on June 21st, 2011) and from TTTech's flyer on "TTECOM Layer ARINC 653 for VxWorks 653".

End of the presentation

Thank you!

Questions?

sdpenn@embraer.com.br

pferrua@embraer.com.br