IEC 62439-3 HSR/PRP Implementation on Sitara Processors using PRU-ICSS

Markets with Common Roots

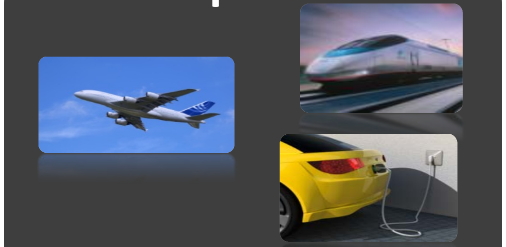
Energy

- Generation
- Transmission
- Distribution

Factory

- Automation
- Production
- Monitoring

Transportation



- Avionics
- Railway
- Automotive

RELIABILITY

ETHERNET PROTOCOLS

REDUNDANCY

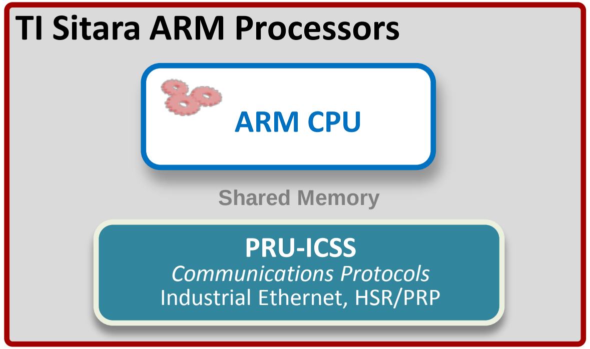
PRU-ICSS: Programmable Real-Time Industrial Communications

TI Sitara Processors simplify development for Industrial applications through the PRU-ICSS:

Deterministic RISC cores with dedicated I/O and memory used for industrial Ethernet and Fieldbus protocols

Hard real-time performance, fully deterministic

C Programmability



TI Sitara ARM Processors

ARM CPU

Shared Memory

PRU-ICSS
Communications Protocols
Industrial Ethernet, HSR/PRP

Benefits of TI's PRU-ICSS solution:

- System BOM **savings** from eliminating ASICs or FPGAs used for industrial Ethernet. Saves **power**, **size**, **cost**
- Software-based solution: Supports **multiple real-time protocols** with the same hardware, adapts to **changing standards**
- **Scalable solution** for master and slave industrial Ethernet, motor control, redundancy, and more.

Protocols Supported

(not full list)

EtherCAT.

PROFIBUS®

PROFI® INDUSTRIAL ETHERNET NET

EtherNet/IP™

SERCOS interface

ETHERNET POWERLINK

BiSS INTERFACE

PRP

HSR

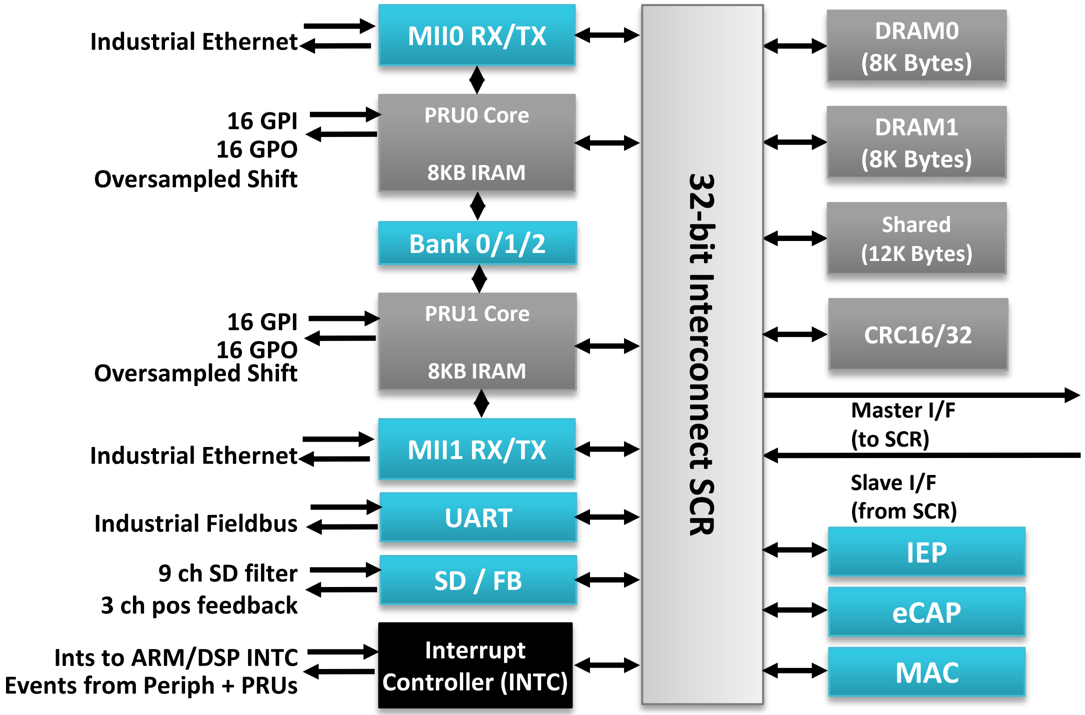
EnDat 2.2

HIPERFACE® DSL

Programmable Real-time Unit for Industrial Communication Subsystem (PRU-ICSS)

- Industrial Ethernet
- Serial Fieldbus
- Encoder Feedback
- Backplane Communication
- Sigma Delta filter
- Custom interfaces
- Signal Processing
- Application Synchronization

ICSS Functional Block Diagram



Industrial SDK: Current (AM335x/AM437x) SYSBIOSSDK-IND-SITARA

Customer Software

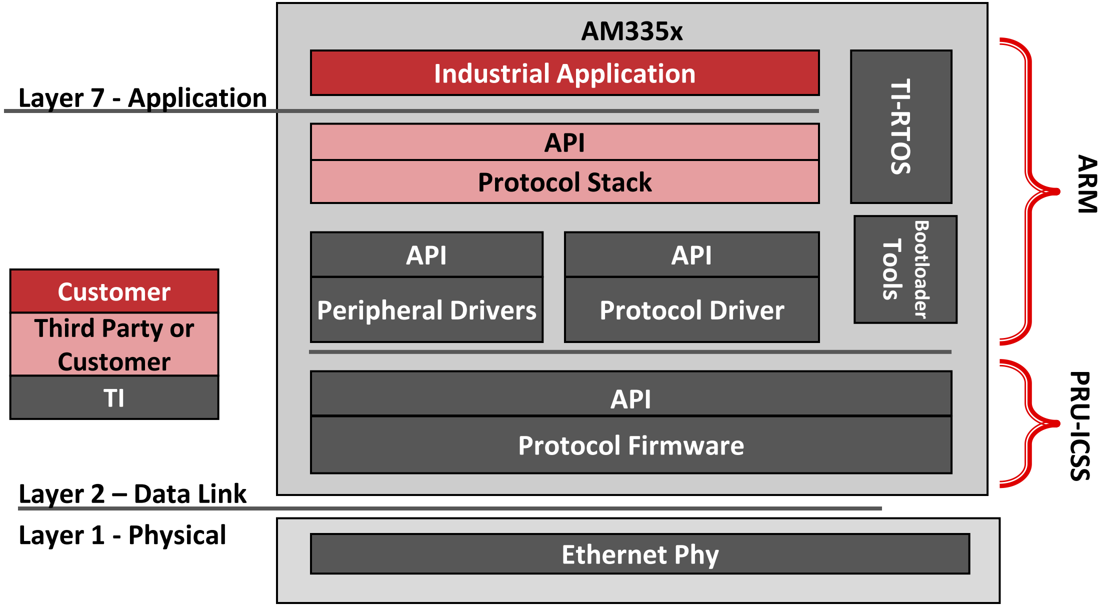
- Industrial application for end product (customer value add)
- BSP for customer boards/RTOS

Third-Party Software

- Free evaluation version of protocol stack
- For production, customers license stack from 3P, or develop/use own stack.

TI Software

- Includes protocol firmware, protocol and peripheral drivers, example slave application, TI-RTOS (SysBIOS), bootloader, and tools
- Some customers use these "as is" for production. Some use as a reference for their port.
- Available online as SYSBIOSSDK-IND-SITARA



Industrial Software for AM57x: PRU-ICSS-INDUSTRIAL-SW

Customer Software

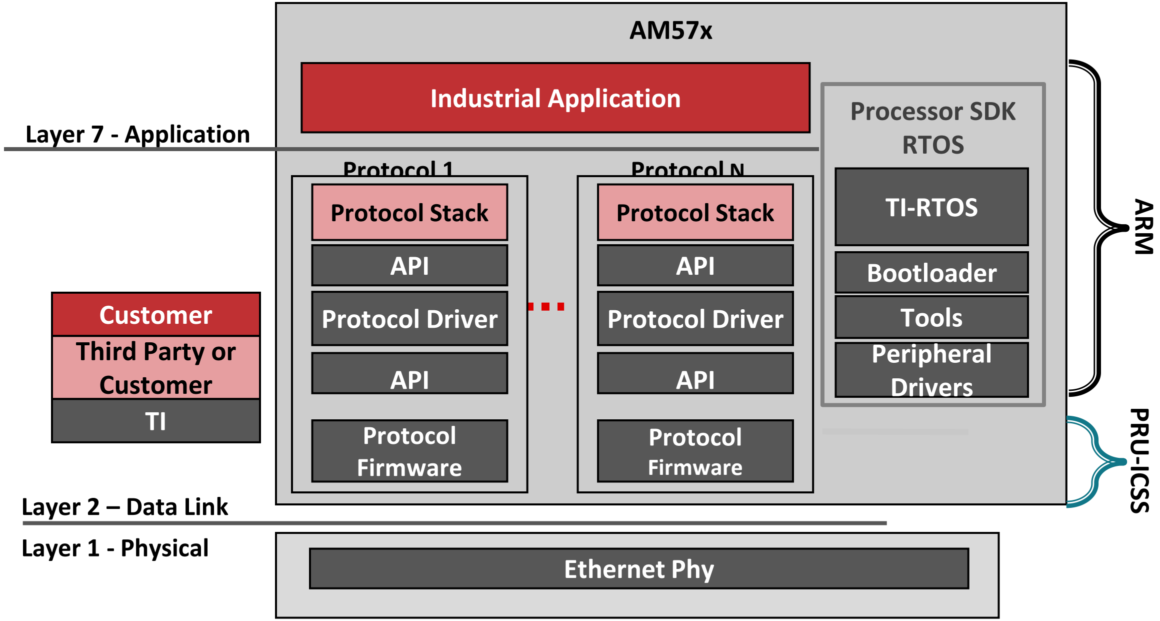
- Industrial application for end product (customer value add)
- BSP for customer boards/RTOS

Third-Party Software

- Free evaluation version of protocol stack
- For production, customers license stack from 3P, or develop/use own stack.

TI Software

- Includes protocol firmware, protocol and peripheral drivers, and example slave application provided per-protocol through PRU-ICSS-INDUSTRIAL-SW
- Industrial software layers on top of Processor-SDK-RTOS



HSR & PRP Redundancy Goals

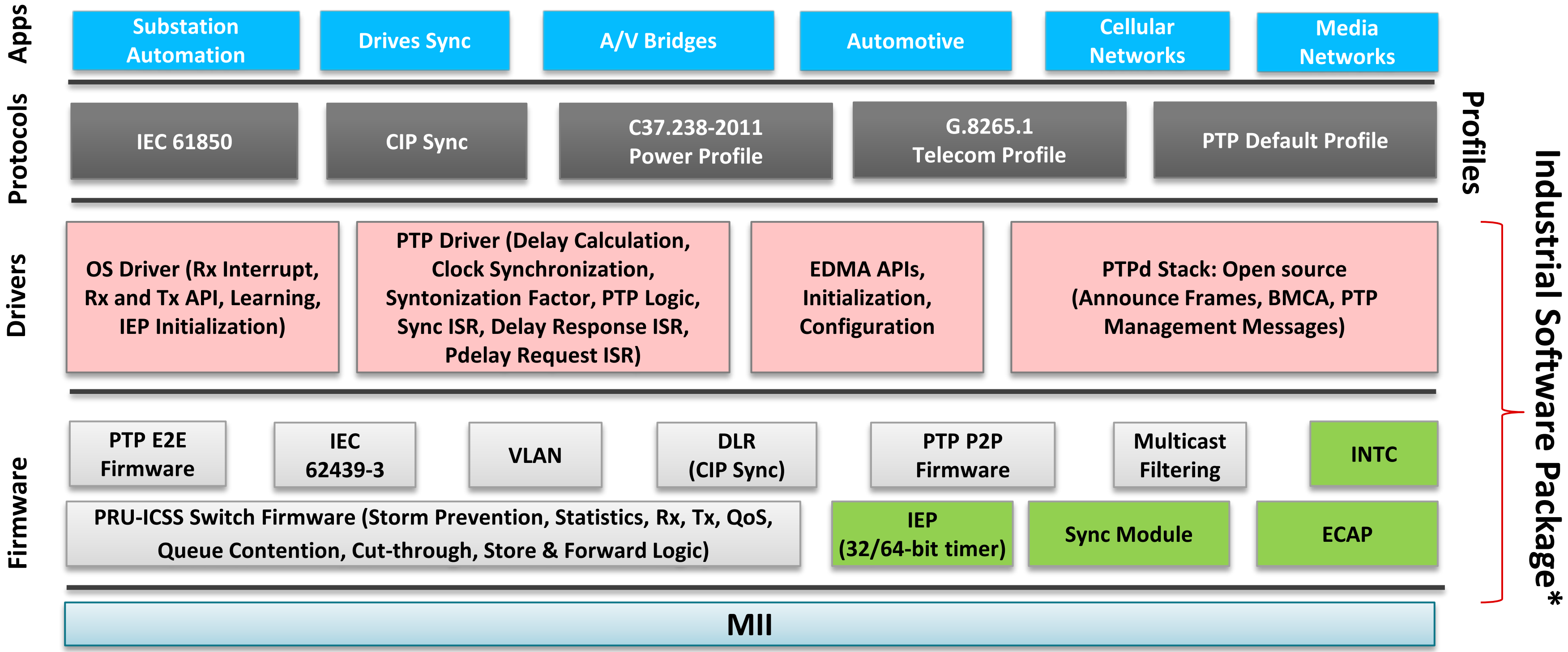
Redundancy goals for HSR (High-availability Seamless Redundancy) and PRP (Parallel Redundancy Protocol):

- Provide zero switchover time in case of failure
- Create a network with no single point of failure
- Fulfill dependability and real-time requirements of demanding applications such as:
  - Substation automation
  - Motion control
  - Avionics communication
- Allow for chaining of devices for cost-effective networking
- Allow for complex topologies such as rings and rings of rings
- Support protocol independence, which is the ability to be layered with any industrial Ethernet or other protocols

Ethernet Redundancy: IEC 62439 Standard

- IEC 62439 standard describes methods to implement Ethernet based network redundancy without packet loss.
- IEC 62439 standards covers different system requirements.
- General automation systems
  - The standard recommends to use **RSTP** (Base: IEEE standards, RSTP); No need for a new standard.
  - Grace time **< 500 ms**
- Benign real-time systems that are cost-sensitive:
  - Grace time **< 200 ms**
  - The standard shall define an adequate bridge redundancy scheme and redundant devices attachment.
  - Base: RSTP and further developments; Solution: **MRP, DRP, RRP**
- Critical real-time systems that require higher coverage:
  - Grace time **= 0 ms**
  - The standard shall define a parallel network solution and redundant device attachment.
  - Base: ARINC AFDX and similar; Solution: **PRP, HSR**

HSR PRP Switch Architecture



Apps

Substation Automation

Drives Sync

A/V Bridges

Automotive

Cellular Networks

Media Networks

Protocols

IEC 61850

CIP Sync

C37.238-2011 Power Profile

G.8265.1 Telecom Profile

PTP Default Profile

Profiles

Drivers

OS Driver (Rx Interrupt, Rx and Tx API, Learning, IEP Initialization)

PTP Driver (Delay Calculation, Clock Synchronization, Syntonization Factor, PTP Logic, Sync ISR, Delay Response ISR, Pdelay Request ISR)

EDMA APIs, Initialization, Configuration

PTPd Stack: Open source (Announce Frames, BMCA, PTP Management Messages)

Firmware

PTP E2E Firmware

IEC 62439-3

VLAN

DLR (CIP Sync)

PTP P2P Firmware

Multicast Filtering

INTC

PRU-ICSS Switch Firmware (Storm Prevention, Statistics, Rx, Tx, QoS, Queue Contention, Cut-through, Store & Forward Logic)

IEP (32/64-bit timer)

Sync Module

ECAP

MII

Industrial Software Package*

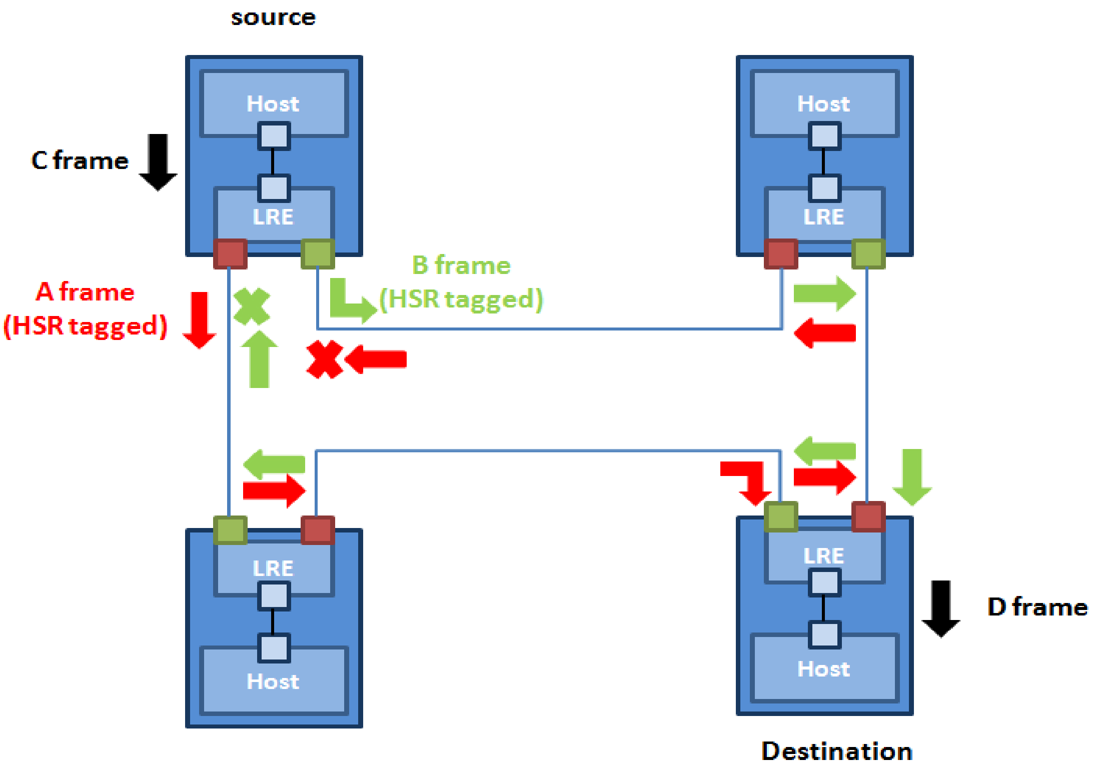
* Industrial software for AM335x, AM437x: SYSBIOSSDK-IND-SITARA. For AM57x: PRU-ICSS-INDUSTRIAL-SW

Real Time System Redundancy: HSR & PRP

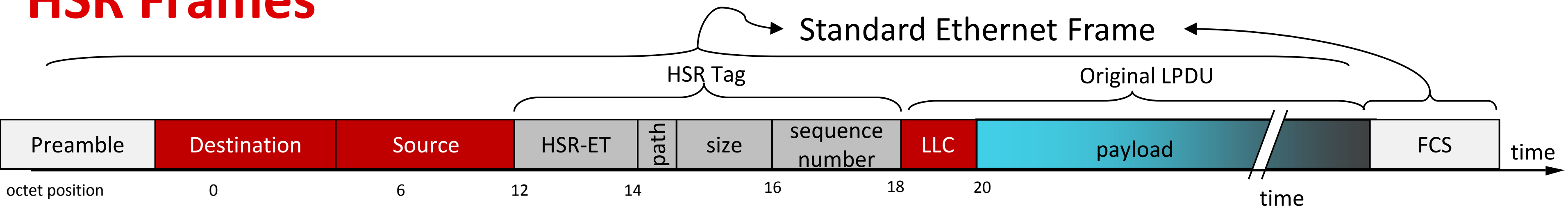
- The **Parallel Redundancy Protocol (PRP)** and **High-availability Seamless Redundancy (HSR)** protocols are suited for applications that demand high availability and low latency and thus gain momentum in the SCADA (Supervisory Control and Data Acquisition), energy, and transportation areas:
  - **PRP**: Seamless protocol suited for critical applications requiring **Doubly Attached Nodes (DAN).**
  - **HSR**: Seamless protocol suited for critical applications requiring a **cost efficient ring structure**
- High-availability Seamless Redundancy (HSR): IEC 62439-3 Clause 5:
  - Each node on the network has **two network ports** and is directly inserted in a network with **ring topology.**
  - Packets are **transmitted** on both network interfaces and thus transmitted **in both directions** of the ring **at the same time**.
  - Intermediate nodes forward the frames in the ring.
  - The receiver removes unicast-frames from the ring, the sender removes multi-cast and broad-cast frames.
  - In error-free conditions, the destination node receives two copies of the frame and **discards the duplicate.**
  - Since forwarding delays of all intermediate nodes sum up, device internal delays should be kept as small as possible.
  - This calls for **a cut-through forwarding mechanism** implemented in hardware. HSR supports this by using a header structure that is optimized for hardware parsing.

HSR Principle: Ring Network

- Nodes are arranged as a ring with two identical ports (A and B).
- For each frame, two copies are sent (A frame and B frame).
- The source node removes the frames it injected into the ring.
- Destination nodes consume the first frame of a pair and discard the duplicate.
- If the ring is broken, frames still arrive over the intact path. Loss of path is detected by duplicate not arriving.

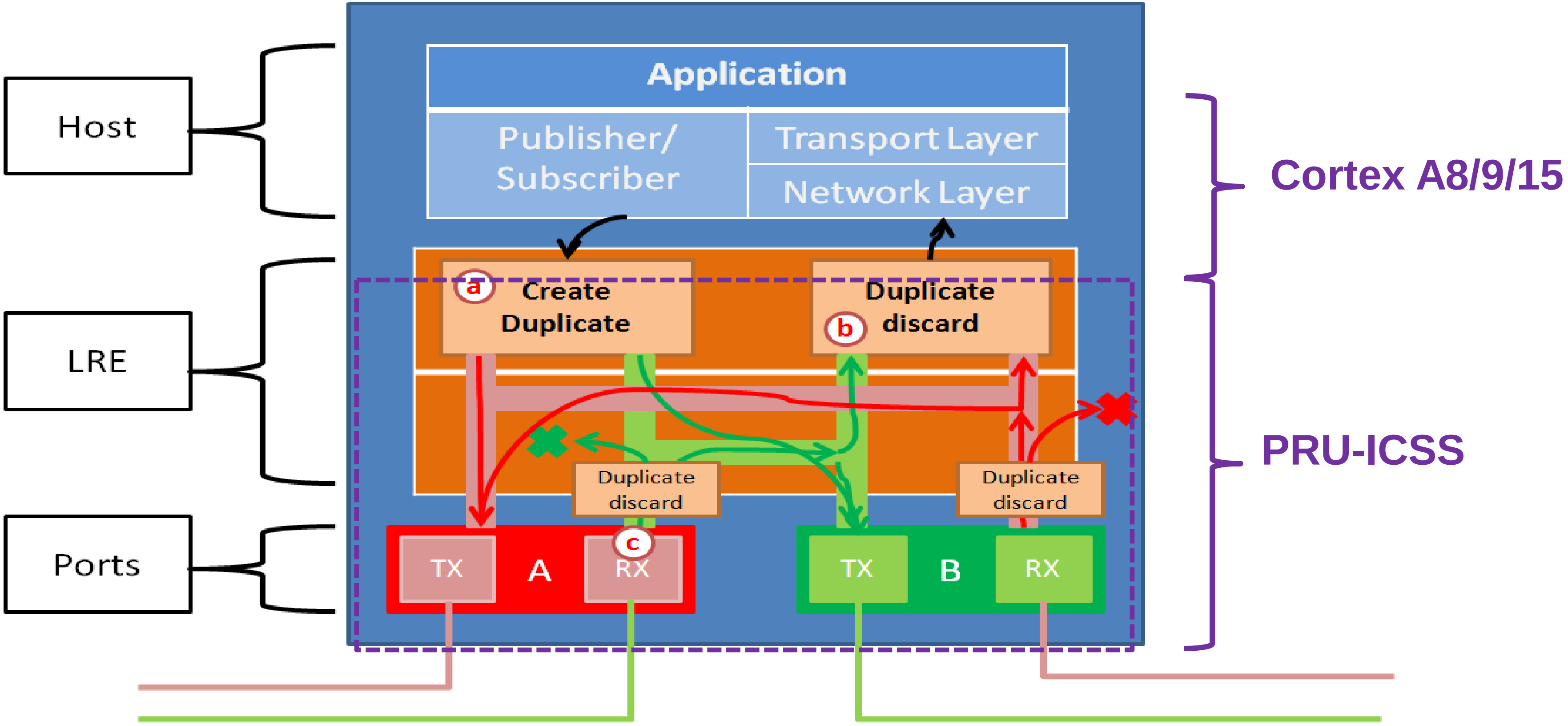


HSR Frames



- Each frame has an HSR Ethertype, a path indicator, a size field, and a sequence number.
- The sender inserts the same sequence number in both frames of a pair, and increments the sequence counter by one, for each sending from this node.
- The receiver keeps track of the sequence counter for each source MAC address from which it receives frames. Frames with the same source and sequence number value coming from different lines are discarded.
- To supervise the network, a node may keep a table of all other nodes in the network from which it receives frames. This allows for simultaneous detection of node absences and bus errors.
- A node recognizes the frame(s) sent through its source address and sequence number.

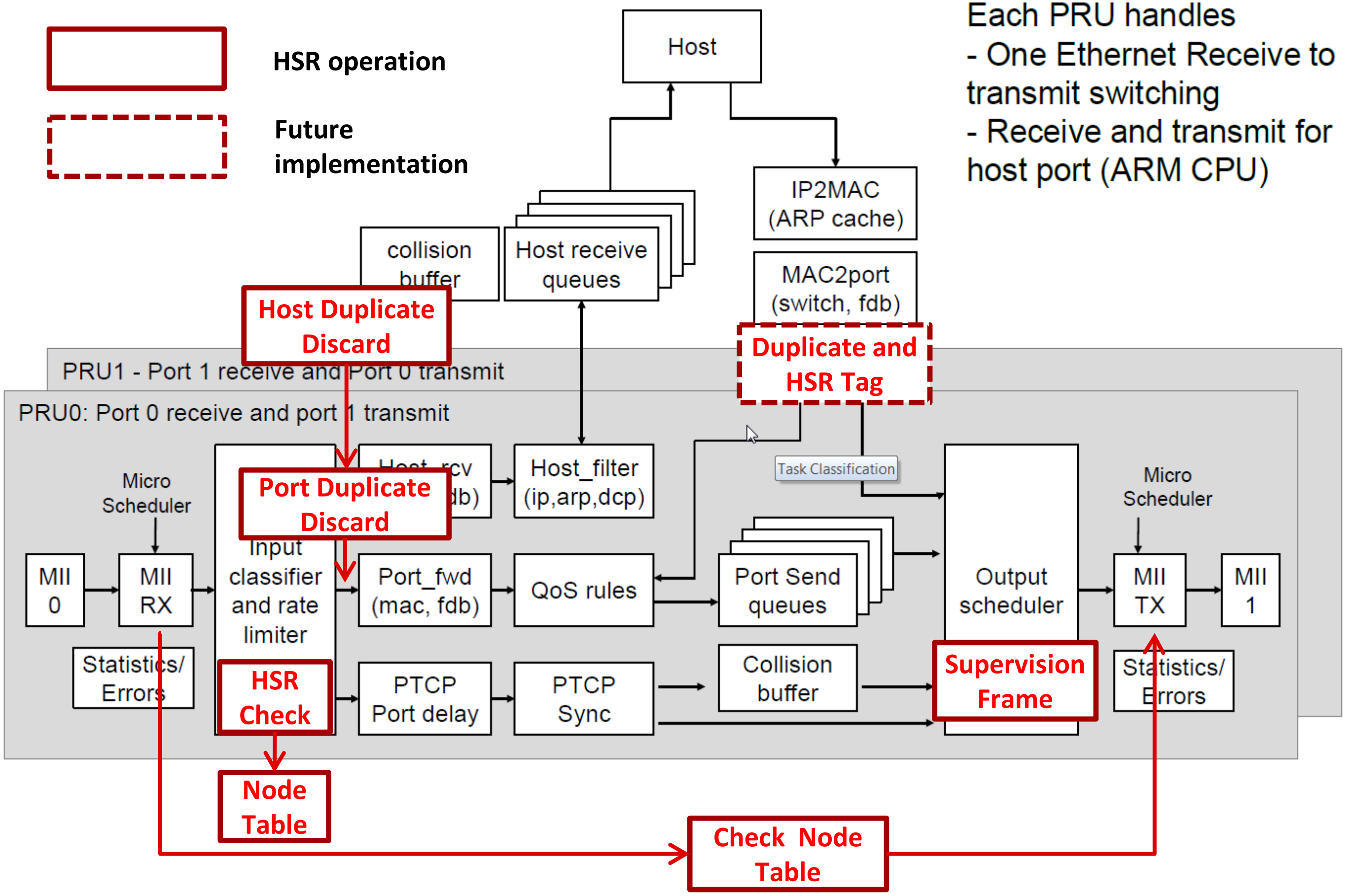
Overview with HSR Logic Scheme



HSR Software Architecture on PRU-ICSS

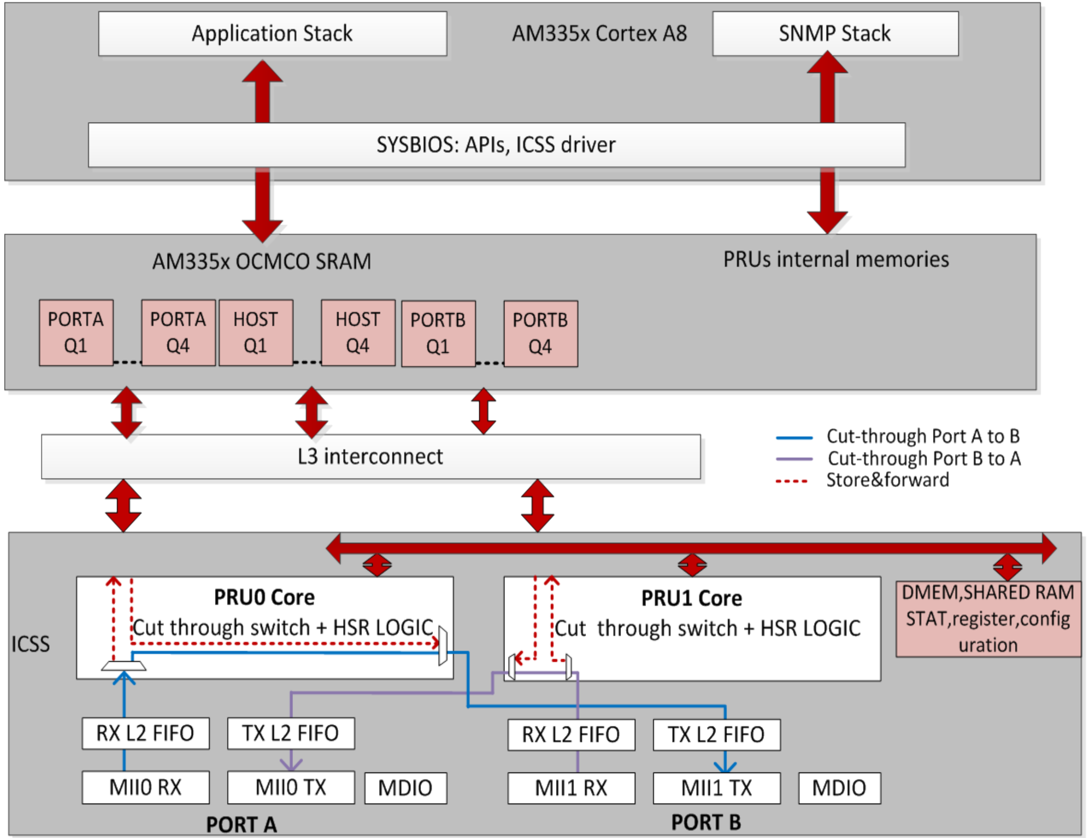
Leverage Ethernet Switch Implementation with Cut-through Operation on PRU-ICSS

Each PRU handles
- One Ethernet Receive to transmit switching
- Receive and transmit for host port (ARM CPU)



HSR + Ethernet Protocol Software Architecture

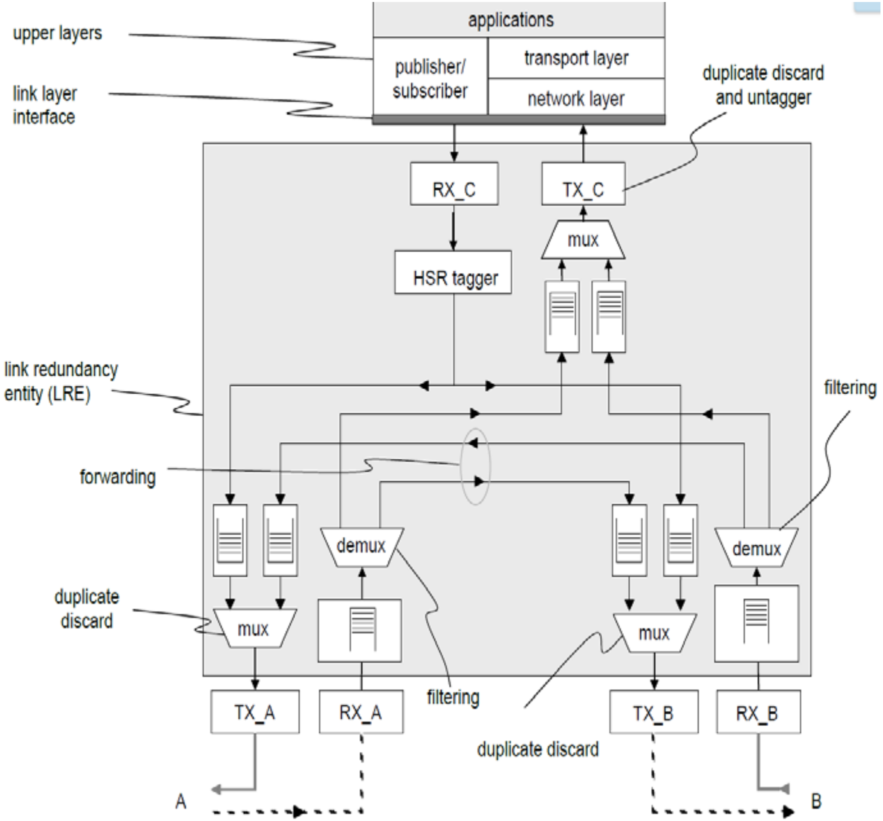
- Driver duplicates the frame received from an application and stores a copy in the queue of each PRU.
- The ICSS redundant switch firmware supporting HSR is integrated with the host application.
- The HSR firmware is transparent for the host applications and host IP stack.



HSR End Node

HSR End Node complying to IEC 62439-3 Clause 5:

- Implemented on two PRU cores of the PRU-ICSS subsystem.
- Host manages high-level stack, application, and platform initialization.
- Each node on the network has two network ports and is directly inserted in a network with ring topology.
- Supports 10/100 mbps full-duplex Ethernet interface.
- Support for HSR Modes H (default); N,T,U,M (configured during runtime).
- Cut-through switching time < 3 µs
- Continuously updates the node table.
- Configurable node table size up to 128.
- Support all statistics MIB variables defined in the IEC standard.
- Storm prevention for AM57x only.
- Supports IEEE1588 single-step, peer-to-peer transparent clock in Layer 2, as well as ordinary clock.



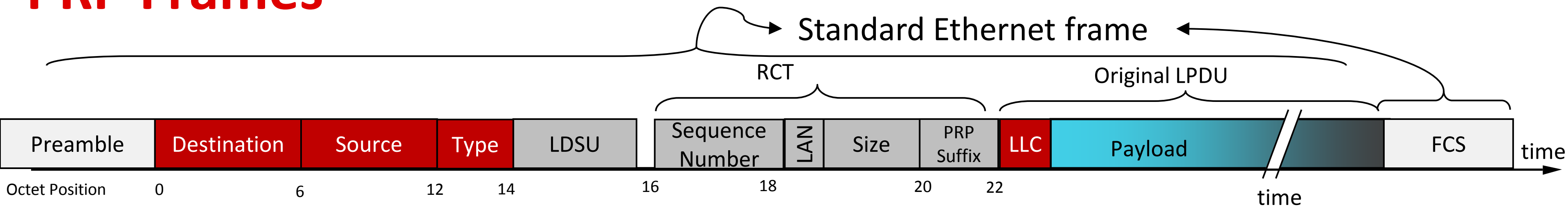
DANH Structure

The main difference compared to HSR is that no frames are forwarded from redundant port to redundant port.

PRP Operation

- PRP uses two LANs that are completely separate:
  - LANs will have independent failures
- Each PRP node has two Ethernet interfaces:
  - Same MAC address and IP address(es).
  - PRP is a layer 2 redundancy and is transparent to higher network protocols.
- PRP Operation:
  - A source node sends two simultaneous copies of a frame, one over each port; The two frames travel through each respective LAN until they reach the destination node.
  - The destination node accepts the first frame of a pair and discards the second; Each frame has a sequence number that is incremented for each frame sent.
- PRP Error Detection and Recovery:
  - Like HSR, provides a zero-time recovery.
  - Checks the redundancy continuously to detect lurking failures.
  - Node failures are not detected.

PRP Frames

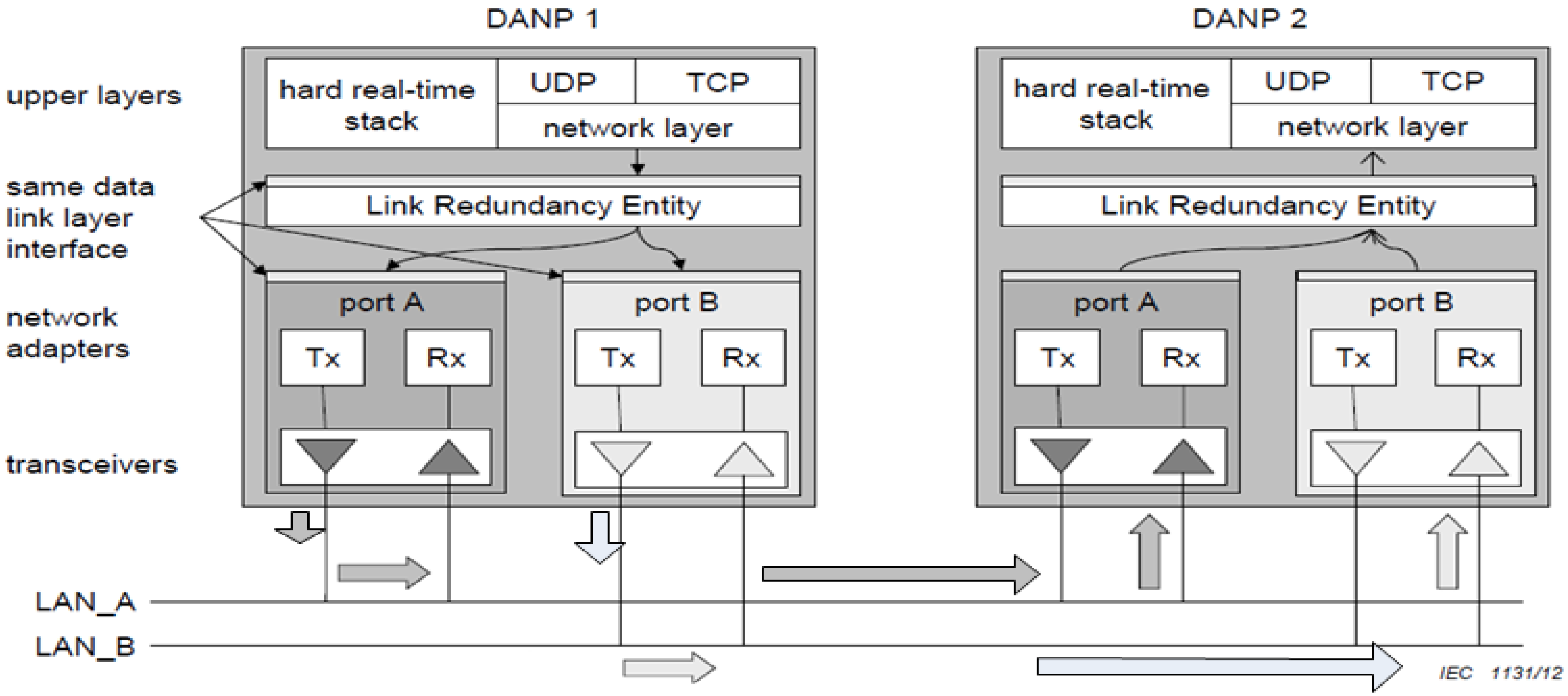


- Each frame has an Redundancy Control Trailer (RCT):
  - 16-bit sequence number
  - 4-bit LAN identifier, 1010 (0xA) for LAN_A and 1011 (0xB) for LAN_B
  - 12-bit frame size
- The sender inserts the same sequence number in both frames of a pair, and increments the sequence counter by one for each frame sent from this node.
- To allow the receiver to distinguish frames coming from PRP nodes vs non-PRP node, the sender appends the frame with the length of the Link Service Data Unit (LSDU) in octets in the 12-bit frame size field.
- The receiver tracks the sequence counter for each source MAC address it receives. Duplicate frames are discarded.

PRP Implementation

- The Link Redundancy Entity (LRE) connects the two ports operating in parallel to the upper layers of the communication stack.
- When the LRE receives a frame from the node’s upper layers:
  - Appends a Redundancy Control Trailer (RCT) containing a sequence number to the frame
  - Sends the frame through both ports at nearly the same time:
    - Both frame copies are identical except for the LAN identifier (and the checksum).
    - The two frames transit through the two LANs with different delays; If there are no errors, both arrive at the destination node.
- When the LRE receives frames from the network:
  - Forwards the first received frame to the node upper layers
  - Removes the RCT, if required
  - Discards the duplicate frame

PRP End Node



PRP with two DANPs communicating

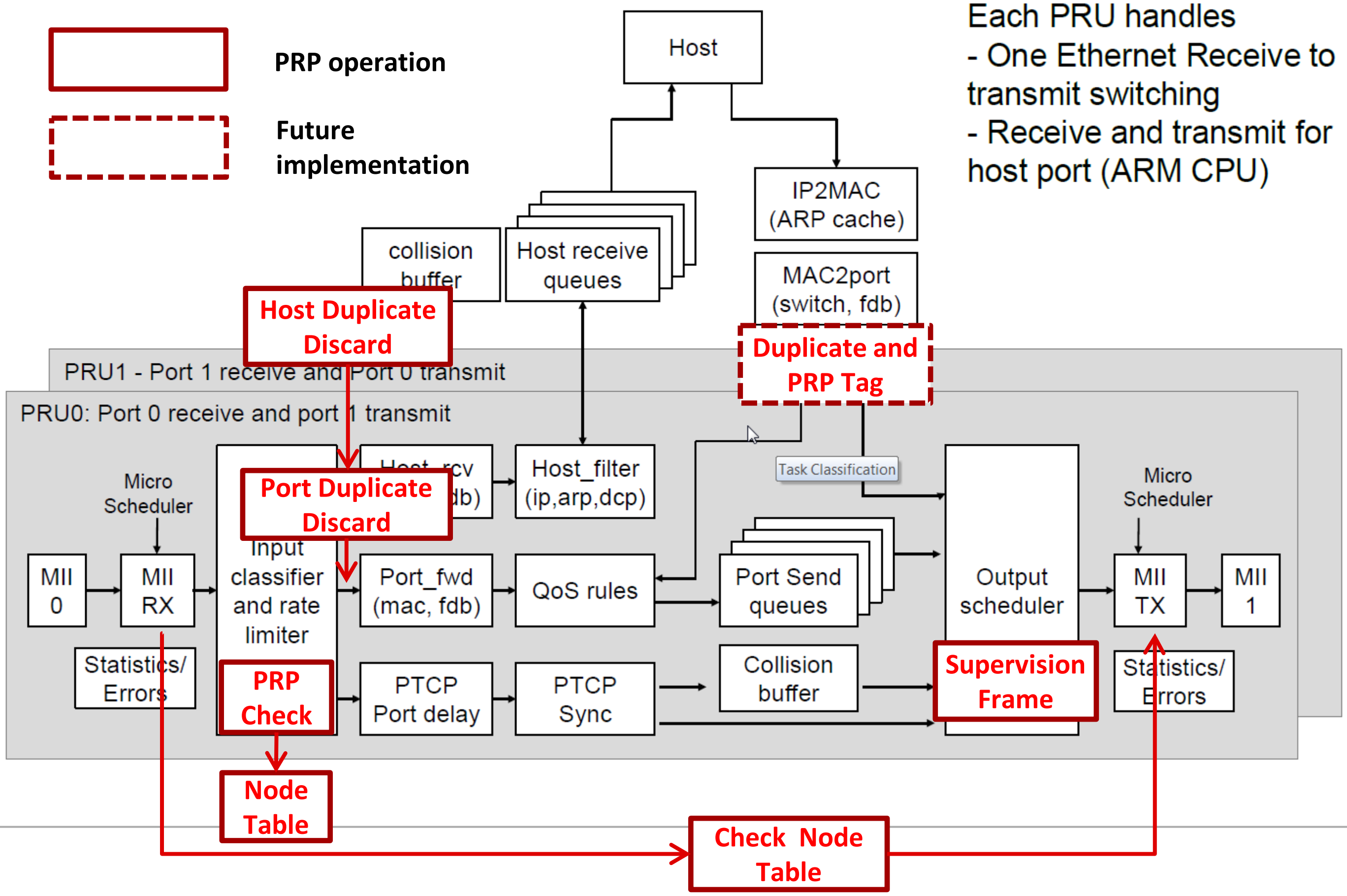
PRP main difference compared to HSR: There is no dependence on network topology (HSR uses a ring).

PRP Software Architecture on PRU-ICSS

Leverage Ethernet Switch Implementation with Cut Through Operation on PRU-ICSS

Each PRU handles

- One Ethernet Receive to transmit switching
- Receive and transmit for host port (ARM CPU)



PRP Compliant

PRP Node complying to IEC 62439-3 Clause 4:

- Implemented on two PRU cores of the PRU-ICSS subsystem
- Host-managed high-level stack, application, and platform initialization
- Seamless protocol suited for critical applications requiring Doubly Attached Nodes (DAN)
- Supports 10/100 mbps full-duplex Ethernet interface
- Store & forward with data integrity check
- Continuously updates the node table
- Configurable node table size up to 128
- Supports all statistics MIB variables defined in the standard
- Storm prevention for AM57x
- Supports IEEE1588 ordinary clock (coming soon)

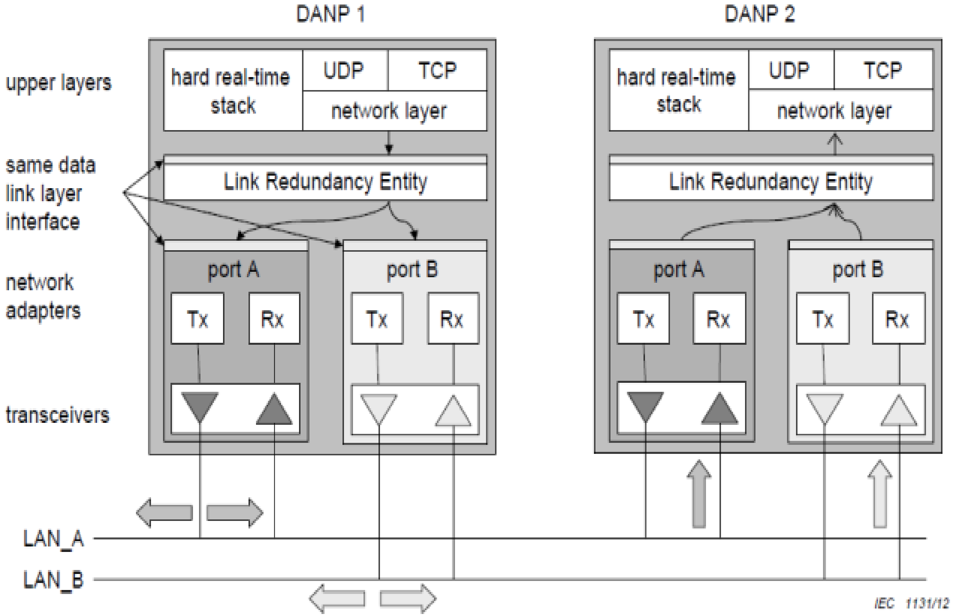


Figure 4 – PRP with two DANPs communicating

HSR/PRP Switch Features: PRU-ICSS Firmware + Driver

- Features:
  - HSR (IEC 62439-3 clause 5)
  - PRP (IEC 62439-3 clause 4)
  - PTP over IEEE 802.3, Peer to Peer mode (IEEE 1588 v2, Annex F)
  - PTP over HSR/PRP, Peer to Peer mode (IEC 62439-3 Ed 2.0 Annex A)
  - Ordinary clock (slave only, IEEE 1588)
- Restrictions:
  - Ordinary clock (master mode)
  - Boundary clock
  - Not backward compatible to PTPv1
  - Supports P2P with IEEE 802.3
  - No support for IPv6

Precision Time Protocol (PTP) Overview

- PTP Features
- PTP Implementation Details

PTP Terminology

- **E2E (End to End)** is a type of PTP delay measurement scheme in which clock measures delay with respect to the master. The packets used for this are Sync, Delay_Req and Delay_Resp.
- **P2P (Peer to Peer)** is a type of PTP delay measurement scheme in which clock measures delay with respect to it’s peer, the format for delay measurement scheme differs between P2P and E2E. The packets used are PDelay_Req and PDelay_Resp.
- **Cycle Time** is the time after which the internal nanosecond counter (IEP) resets itself. This is also the period for the periodic sync0 signal generated.
- **OC (Ordinary Clock)** is a clock that synchronizes it’s time base to a master.
- **TC (Transparent Clock)** is a clock that only performs adjustment for packets passing through it.
- **BMCA (Best Master Clock Algorithm)** determines which clock is the highest quality clock within the network.

PTP Features

- The Industrial SDK supports the following annexes and profiles of IEEE 1588v2:
  - Annex D: PTP over UDP over IPv4, End to End mode
  - Annex F: PTP over IEEE 802.3. Peer to Peer mode
  - Slave-only ordinary clock (No master at the moment)
  - PTPd stack integration
  - IEC 62439-3 Ed 2.0 Annex A – PTP over HSR. Peer to Peer mode
  - CIP Sync: ODVA Ethernet/IP specification Vol 1 Ed3
  - Supports only E2E with UDP and P2P with IEEE 802.3; Other combinations not supported.
  - Power Profile C37.238-2011 (Not supported right now)
  - Will not be backward compatible to PTPv1
  - IPv6 is not supported at the moment.

NOTE: All profiles and features will not be enabled on a single firmware.

PTP Task (PRU-ICSS)

- PTP Task is split into functionality performed on PRU-ICSS (Firmware) and Host (ARM CPU).
- PRU-ICSS is responsible for parsing the PTP messages and processing them. It decides whether to forward them to Host or go to Store and Forward mode.
- PRU-ICSS does the bulk of packet processing for PTP.
- PRU-ICSS is responsible for all time-critical operations like timestamping and bridge delay correction.
- PRU-ICSS provides intimation to Host about packets and provides timestamps to the Host.
- PRU-ICSS also checks if a sync packet is valid based on information provided by the Host. Thus, the PRU-ICSS is responsible for security.

PTP Task (Host)

- Host Task is split into driver and Stack functionality.
- The driver is responsible for doing all calculations like line and peer delay calculations.
- Driver performs all initializations like IEP and Sync0 hardware.
- Performing IEP adjustment by calculating the adjustment factor and triggering the EDMA.
- Stack performs all management related tasks and performing tasks like BMCA (Best Master Clock Algorithm).
- In conventional design, all packet processing is done in the PTP stack inside the Host processor:
  - This makes processing slow.
  - A majority of this task is offloaded to PRU-ICSS for better performance in TI’s design.
- Stack code is adapted from open source PTPd project.

For More Information

- Overview for Sitara ARM Processors: http://www.ti.com/sitara
- AM437x Sitara™ Processors Training Series: https://training.ti.com/am437x
- AM57x Sitara™ Processors Training Series: https://training.ti.com/am57x-sitara-processors-training-series
- For questions regarding topics covered in this training, visit the support forums at the TI E2E Community website.