

HSR/PRP Solutions on Sitara Processors for Grid Substation Communication



Kathryn Kalouf, Daolin Qiu

ABSTRACT

Texas Instruments AM64x/AM243x processors support the PRU_ICSSG subsystem, which mainly differs from the PRU_ICSS subsystem on AM3x/AM4x processors by being able to support up to 1 Gbps bandwidth as opposed to being limited to 100 Mbps. This paper explains in detail how the PRU_ICSSG subsystem implements HSR/PRP technology as well as introduces the main benefits of using PRU_ICSSG for applications that require the HSR/PRP protocol.

Table of Contents

1 Ethernet Redundancy in Smart Grid Substations	2
1.1 High Availability Seamless Redundancy (HSR).....	3
1.2 Parallel Redundancy Protocol (PRP).....	4
2 HSR/PRP Implementation in Sitara Processors	6
2.1 PRU_ICSSG Overview.....	6
2.2 PRU_ICSSG HSR/PRP Architecture.....	7
2.3 PRU_ICSSG HSR/PRP Features.....	7
2.4 PRU_ICSSG HSR/PRP Firmware.....	8
2.5 Benefits of PRU_ICSSG HSR/PRP.....	9
2.6 Linux HSR/PRP Solution.....	9
2.7 RTOS HSR/PRP Solution.....	10
3 HSR/PRP SDK Support	12
4 Summary	13
5 References	14
6 Revision History	14

List of Figures

Figure 1-1. Typical Electric Smart Grid.....	2
Figure 1-2. Process Bus vs Station Bus.....	3
Figure 1-3. Typical HSR Network Topology.....	4
Figure 1-4. Typical PRP Network Topology.....	5
Figure 2-1. PRU_ICSSG Functional Block Diagram.....	6
Figure 2-2. HSR/PRP Software Block Diagram.....	7
Figure 2-3. PRU_ICSSG HSR/PRP Firmware Block Diagram.....	8
Figure 2-4. Example of AM64x used in an HSR Network.....	9
Figure 2-5. Linux Hardware Offload HSR/PRP Block Diagram.....	10
Figure 2-6. RTOS Hardware Offload HSR/PRP Block Diagram.....	11

List of Tables

Trademarks

All trademarks are the property of their respective owners.

1 Ethernet Redundancy in Smart Grid Substations

Before digital and smart grid substations, traditional substations relied on analog systems and manual operations to monitor and control electricity distribution. Due to the rising need for a system that manages power distributions with greater efficiency, flexibility, and reliability, the industry transitioned to digital grid infrastructure. Digital grids use advanced sensors and networks to gather and process data. This industry update enabled real-time monitoring and automated control, unlike that of the previously used traditional grids. Smart grids are a type of digital grid that takes it a step further by having advanced analytics, management, and decision-making capabilities allowing for the optimization of grid performance across a wider scale.

A smart grid substation is a key component of the electricity-grid infrastructure, located everywhere from the high-voltage power generation facilities throughout the distribution network to the low-voltage feeders serving residences and businesses. Substations are a primary factor in transforming voltage levels for transmission and performing important functions such as switching, monitoring, and protecting sub-systems in order to maintain grid efficiency and reliability. To meet these important functions, fast, fault tolerant communication that helps operators achieve cost efficient and reliable operations are necessary. Substation networks require communications that have built-in redundancy and time synchronization with no down time due to a single fault in the network. If a failure does occur, the network must recover within a given short time. Redundancies create more than one path between the source and destination to reroute traffic at the time of failure.



Figure 1-1. Typical Electric Smart Grid

Operators need to continually monitor the health of networks and take action to maintain the operation with efficiency. This need leads to the requirement for reliable and low-latency communications between the control center of the operator and high-value nodes such as substations. Due to the different requirements of station and process bus networks, different protocols need to be implemented according to their specific performance characteristics.

Within a substation, there exists three main different levels of communication: the station level, bay level, and the process level. The 'station bus' handles communication between the station level and the bay level and the 'process bus' handles communications between the bay level and the process level. Each bus within the three main levels carry Ethernet networks that connect various Intelligent Electronic Devices (IEDs) - such as protection relays, bay controllers, etc. - and smart control/SCADA systems. The station bus network is mainly used to carry event-driven Ethernet messages for supervising the system and it interconnects the whole substation and provides connectivity between the central management and individual IEDs at the bay level. The station bus typically carries Generic Object Oriented Substation Event (GOOSE) traffic and TCP/UDP traffic. Traffic in the station bus can tolerate frame losses.

The process bus also carries GOOSE messages, but is also mainly used to carry measurement traffic, in the form of sampled values (SV) traffic. These are small Ethernet frames carrying measurement values, sent by merging units for example - a merging unit is a device within the substation that measures current and voltage

signals from instrument transformers. SV traffic does not tolerate frame loss, therefore, media redundancy must work without interruption.

The process bus is redundantly connected to the station bus through, for example, the High-availability Seamless Redundancy protocol (HSR) or Parallel Redundancy Protocol (PRP). The process bus traffic is sent to the station bus for analysis and for additional monitoring.

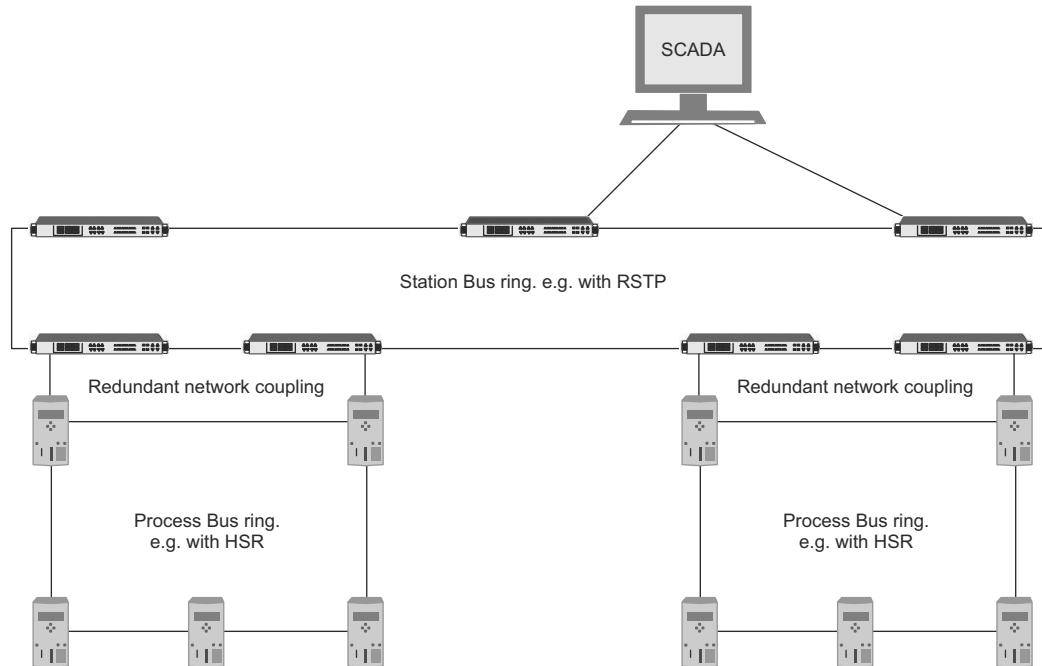


Figure 1-2. Process Bus vs Station Bus

The need for high performance, high reliability, and predictable Ethernet networks drove the motivation for the IEC 62439 standard with the goal to create low cost, easy to maintain, and interoperable common network infrastructure with built-in redundancy.

1.1 High Availability Seamless Redundancy (HSR)

IEC 62439-3 clause 5 defines the High Availability Seamless Redundancy (HSR) protocol as a redundancy protocol for substation automation. HSR is an independent redundancy protocol based on double transmission of message frames over ring-topology networks, so if one connection fails, the transmission on the second connection succeeds. No reconfiguration time or relearning of the communication paths is necessary for the network.

Since communication over an HSR network still continues without interruption when there is a network failure, HSR is a good fit for process bus applications.

HSR frames are identified uniquely by the HSR tag. Field devices are connected to a Double Attached Node HSR (DANH in a Ring topology and only DANH compliant nodes can be connected to the HSR network). Other standard Ethernet devices (Singly Attached Nodes, SANs) need to be connected through a Redundancy Box (RedBox) to work with HSR networks. The HSR tag is used to manage redundancy brought by duplicate frames in the HSR network.

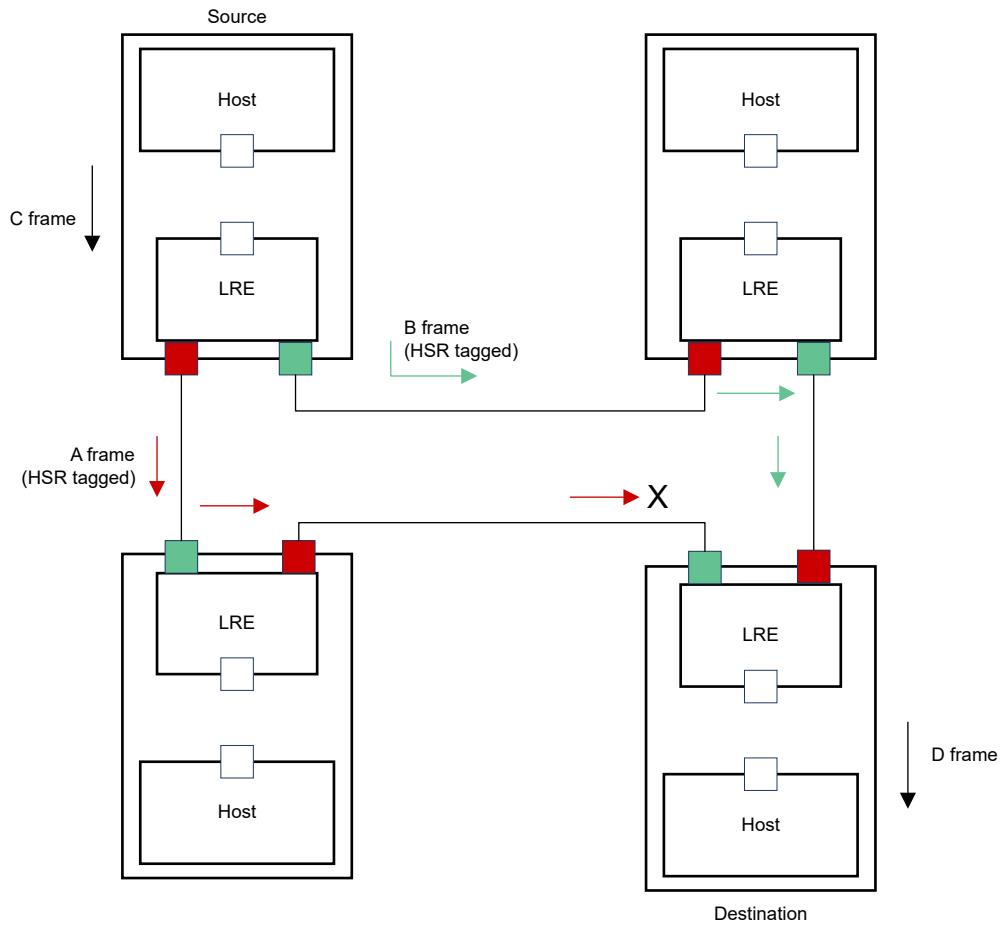


Figure 1-3. Typical HSR Network Topology

A DANH node has two ports operated in parallel and contains the upper layers of the communication stack (represented by "Host" in the diagram) and the Link Redundancy Entity (LRE), which helps present toward the upper layers the same interface as a non-redundant network adapter, so that the upper layers are unaware of redundancy. A source DANH prefixes a frame passed from its upper layers (in Figure 1-3, this is represented with "C frame") with an HSR tag to identify frame duplicates and sends the same copy of each frame over each port ("A frame" from the red port and "B frame" from the green port). Please note that the red and green ports on the figure simply represent two different ports on each DANH node.

A destination DANH receives, in the fault-free state, two identical frames (one from each port) within a certain interval. It removes the HSR tag of the first frame (in the example represented in Figure 1-3, this would be "B frame") before passing it to its upper layers (in the figure, this is represented by "D frame") and discards any duplicate frames (in the example represented by the figure, this would be "A frame").

The nodes support the IEEE 802.1D bridge functionality and forward frames from one port to the other, except if they have already sent the same frame in that same direction. In particular, the node will not forward a frame that it injected into the ring. A destination node of a unicast frame does not forward a frame for which it is the only destination, except for testing. Frames circulating in the ring carry the HSR tag inserted by the source, which contains a sequence number. For intermediate or "forwarding" nodes (i.e. neither the source nor destination node), frames are directly forwarded without removal of the HSR tag as removing the HSR tag is the responsibility of the destination node. The doublet (source MAC address, sequence number) uniquely identifies copies of the same frame. Discarding the duplicate frame prevents the network from being flooded with unnecessary traffic.

1.2 Parallel Redundancy Protocol (PRP)

PRP simply doubles the necessary network infrastructure and is defined in IEC 62439-3 Clause 4. All devices that need to take advantage of this redundant network infrastructure must be connected with a dual attachment

network interface that connects both interfaces, each to one of the two redundant Local Area Networks (LANs) A and B. Both networks are used simultaneously and both networks carry the same data that is sent redundantly by the Dual Attached Nodes PRP (DANP). Each DANP duplicates its entire network traffic that travels over both LAN A and LAN B.

The PRP supports star topology, which ensures a fixed-hop delay with cost of infrastructure.

When one of the two LANs experiences a fault, the network traffic still runs over the other LAN without interruption. The PRP is also an option to for the process bus, because in case of a failure in one network, the SV traffic can still safely travel through the other network to its destination.

A DANP node has two ports that operate in parallel and that are attached to the same upper layers of the communication stack through the Link Redundancy Entity (LRE). For the basic communication, the LRE presents toward its upper layers the same interface as a non-redundant network adapter, so the upper layers are unaware of redundancy. The LRE has two tasks: handling of duplicates and management of redundancy. When receiving a frame from the node's upper layers, the LRE appends a Redundancy Check Trailer (RCT) containing a sequence number to the frame and sends the frame through both ports at nearly the same time. The two frames are nearly identical except for the LAN identifier (and the checksum).

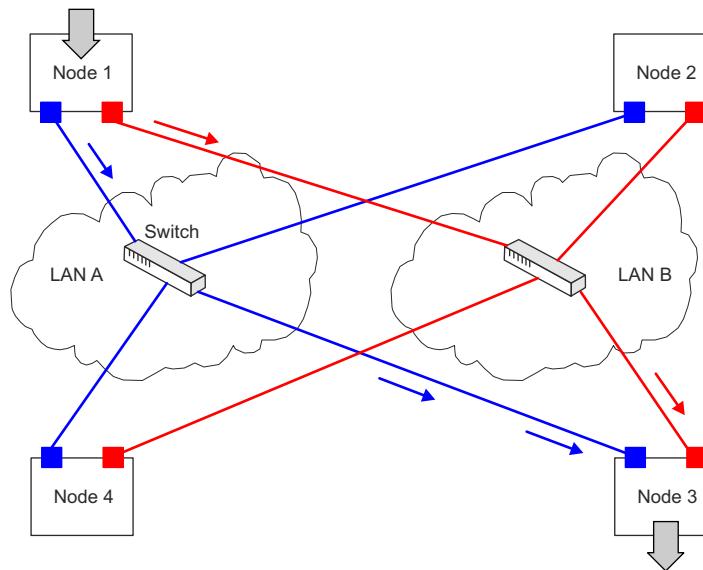


Figure 1-4. Typical PRP Network Topology

The two frames transit through the two LANs with different delays, ideally they arrive at nearly the same time at the destination node. When receiving frames from the network, the LRE forwards the first received frame of a pair to its node's upper layers and discards the duplicate frame (if it arrives). It removes the RCT, if required.

Figure 1-4 is a simplified view of an example PRP network consisting of 4 DANP nodes. It can be assumed that each node contains an LRE, which as mentioned is essential to presenting toward its upper layers the same interface as a non-redundant network adapter, and a host, to represent the upper layers of the communication stack.

2 HSR/PRP Implementation in Sitara Processors

The focus of this paper is to describe how the PRU_ICSSG subsystem on Texas Instruments AM64x/AM243x microprocessors implement HSR/PRP features. The following sections will start with an overview of the PRU_ICSSG subsystem and the HSR/PRP architecture, features, and firmware. At the end, a summary of the main benefits of offloading the PRU_ICSSG implementation of HSR/PRP will be given, along with how the Texas Instruments Linux and RTOS software development kits (SDKs) implement HSR/PRP.

2.1 PRU_ICSSG Overview

AM64x/AM243x SoCs integrate PRU_ICSSG technology that enables customers to add HSR-PRP Dual Attached Node support to the system. The Programmable Real-Time Unit and Industrial Communication Subsystem - Gigabit (PRU_ICSSG) consists of 6, 32-bit RISC cores (Programmable Real-Time Units, or PRUs), data and instruction memories, internal peripheral modules, and an interrupt controller (INTC). The programmable nature of the PRU_ICSSG, along with its access to pins, events and all SoC resources, provides flexibility in implementing fast real-time responses, specialized data handling operations, custom peripheral interfaces, and in offloading tasks from the other processor cores of the system-on-chip (SoC).

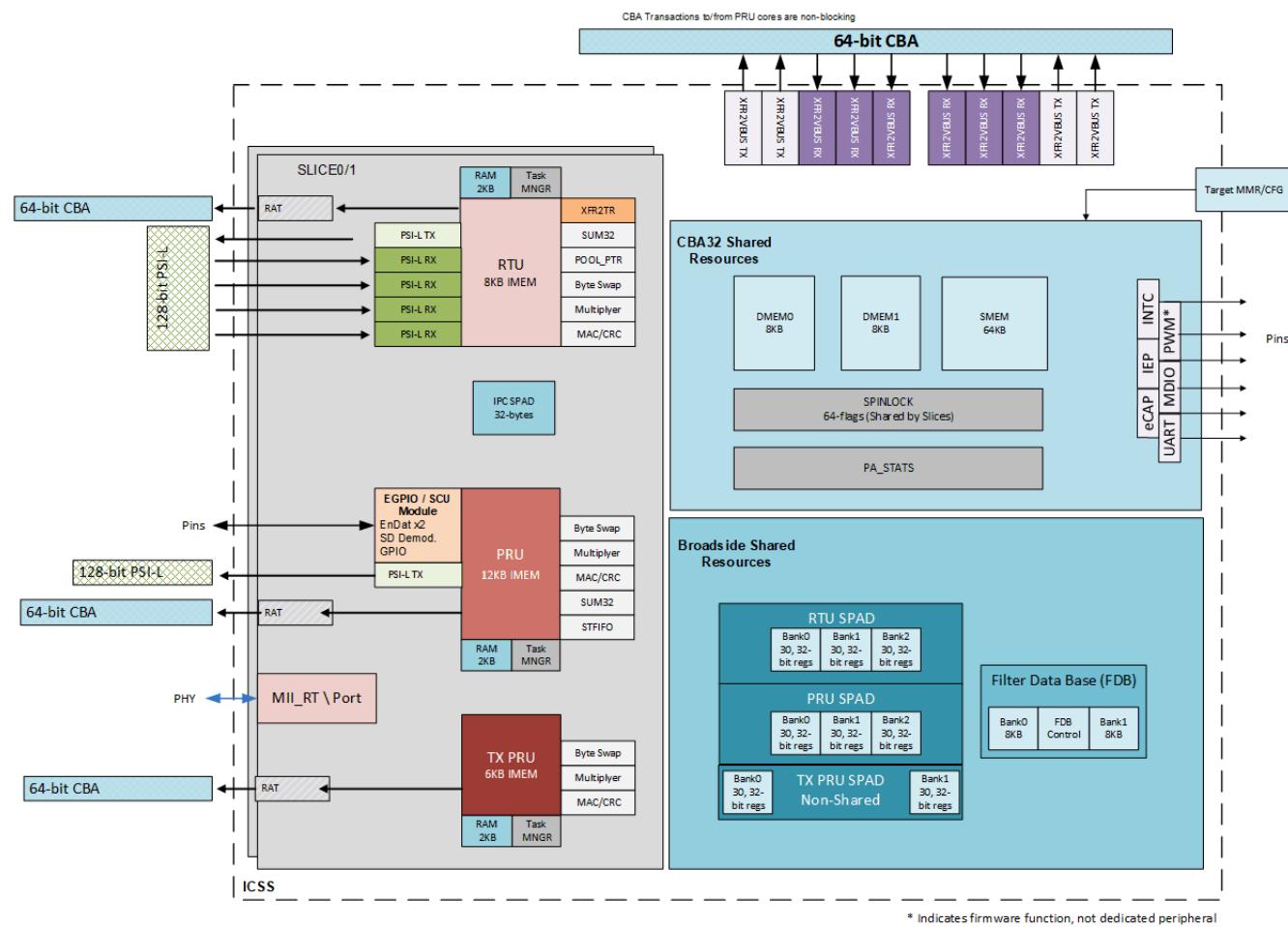


Figure 2-1. PRU_ICSSG Functional Block Diagram

The cores within each PRU_ICSSG have access to all resources on the SoC through the VBUSM Interface Controller port. Additionally, the external host processors can access the PRU_ICSSG resources through the VBUSP Interface Target port.

The use of XFR2VBUS allows BroadSide 32Bytes of data transfer to/from SoC CBASS0 Interconnect at 256-bit bursts using the VBUSM Controller port. The 32-bit Internal CBASS Interconnect bus will be the primary interconnect between all components internal to the PRU_ICSSG.

There are two equally symmetrical halves in each PRU_ICSSG known as SLICE0 and SLICE1. Each slice will share several resources while capable of working independently of each other. There are two sets of XFR2VBUS for each Slice. The XFR2VBUS hardware accelerator is shared between PRU0 and RTU_PRU0 for SLICE0 and the same configuration is valid for SLICE1. The TX_PRU0 and TX_PRU1 cores have also attached XFR2VBUS hardware accelerators.

The INTC handles system input events and posts events back to the device-level host CPU. The PRU cores are programmed with a small, deterministic instruction set. Each PRU can operate independently or in coordination with each other and can also work in coordination with the device-level host CPU. This interaction between processors is determined by the nature of the firmware loaded into the PRU's instruction memories.

The PRU_ICSSG also contains components such as FDB (Filter Database), XFR2PSI, and MII_G_RT (Real-time Media Independent Interface), which are key components to implementing HSR and PRP functions.

Refer [AM64x/AM243x Technical Reference Manual Programmable Real-Time Unit and Industrial Communication Subsystem - Gigabit \(PRU_ICSSG\)](#) for complete details on PRU_ICSSG. For more details on PRU, please refer to the [PRU Academy Training Module](#).

2.2 PRU_ICSSG HSR/PRP Architecture

The implementation consists of PRU_ICSSG firmware, Drivers (Linux/RTOS), application, and SNMP support. LRE is mostly handled in PRU_ICSSG firmware and driver. The main cores (Cortex-Ax and Cortex Rx) run drivers and application code. TI provides PRU_ICSSG HSR firmware and drivers (Linux and FreeRTOS); however, customers develop the protocol and applications. Please note that the terms "RTOS" and "FreeRTOS" is used interchangeably throughout this paper.

Below shows HSR/PRP switch architecture on Sitara devices that are made of firmware, drivers, protocols and applications.

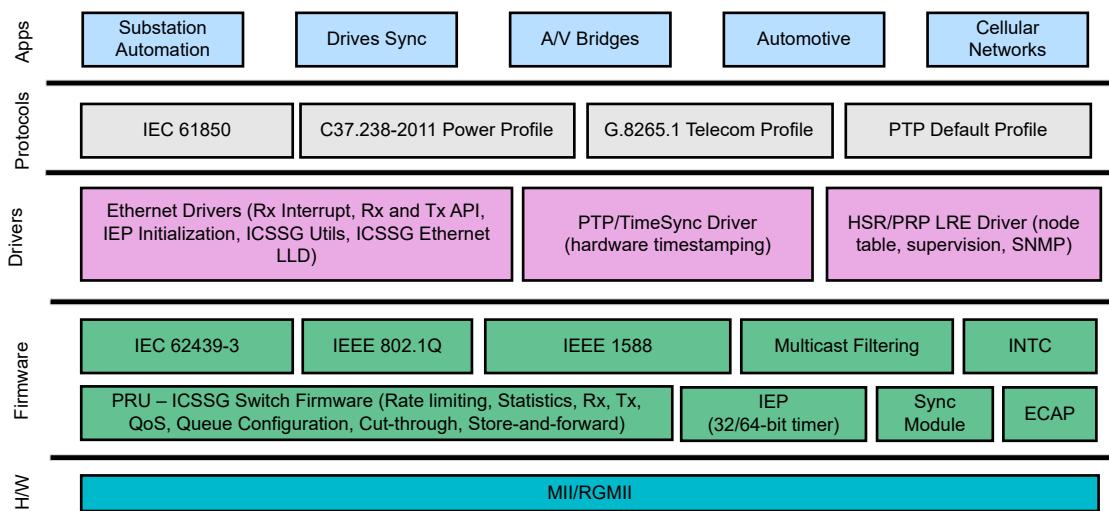


Figure 2-2. HSR/PRP Software Block Diagram

The firmware layer is run on the PRU_ICSSG and currently implements features such as cut-through forwarding, duplicate generation and discard logic, and HSR tag insertion and removal to enable HSR and PRP. The PRU_ICSSG is also capable of implementing support for IEEE 802.1CB (Frame Replication and Elimination for Reliability - FRER). The driver layer connects firmware to protocol layer and implement features like node table, SNMP support. The protocol layer runs features such as IEC 61850, PTP default profile, and interfaces with the application layer. Please note that the protocol layer is not provided by TI.

2.3 PRU_ICSSG HSR/PRP Features

- HSR End Node (DANH) compliant with IEC 62439-3 Edition 3 Clause 5
- PRP End Node (DANP) compliant with IEC 62439-3 Edition 3 Clause 4
- 10/100/1000 Mbits/s Full Duplex Ethernet Interface
 - 1000 Mbits/s exists for Linux from SDK 10.1 and newer SDKs

- Enhanced ICSSG cut-through switch with HSR capability
 - Achieving switching time < 2.5µs at 1 Gbps link speed
- IEEE 802.1Q QoS traffic prioritization maintained on both egress (host-to-Ethernet) and ingress (Ethernet-to-host) paths
- IEEE 802.1Q bridging rules support, including Filter Database (FDB) - static and ageable entries, and RSTP port states (port block, disable, learning)
- IEEE 1588 - PTP timescale timer with 1-PPS capability
- Rate limiting to prevent storm traffic by limiting the rate of best-effort traffic
- Node table enhanced with traffic capture (statistics) for all frames received by the host

2.4 PRU_ICSSG HSR/PRP Firmware

A common firmware is used across FreeRTOS and Linux implementations of HSR/PRP. PRU_ICSSG firmware handles features like cut-through forwarding, port to host/port duplicate detection and elimination, HSR/PRP tag insertion and removal, and packet duplication.

The PRU_ICSSG runs the Link Redundancy Entity (LRE) using all the 6 PRU cores letting the host manage the high level stack and user application as well as the necessary platform initialization.

Figure 2-3 shows more details of how the PRU_ICSSG firmware implements HSR/PRP logic.

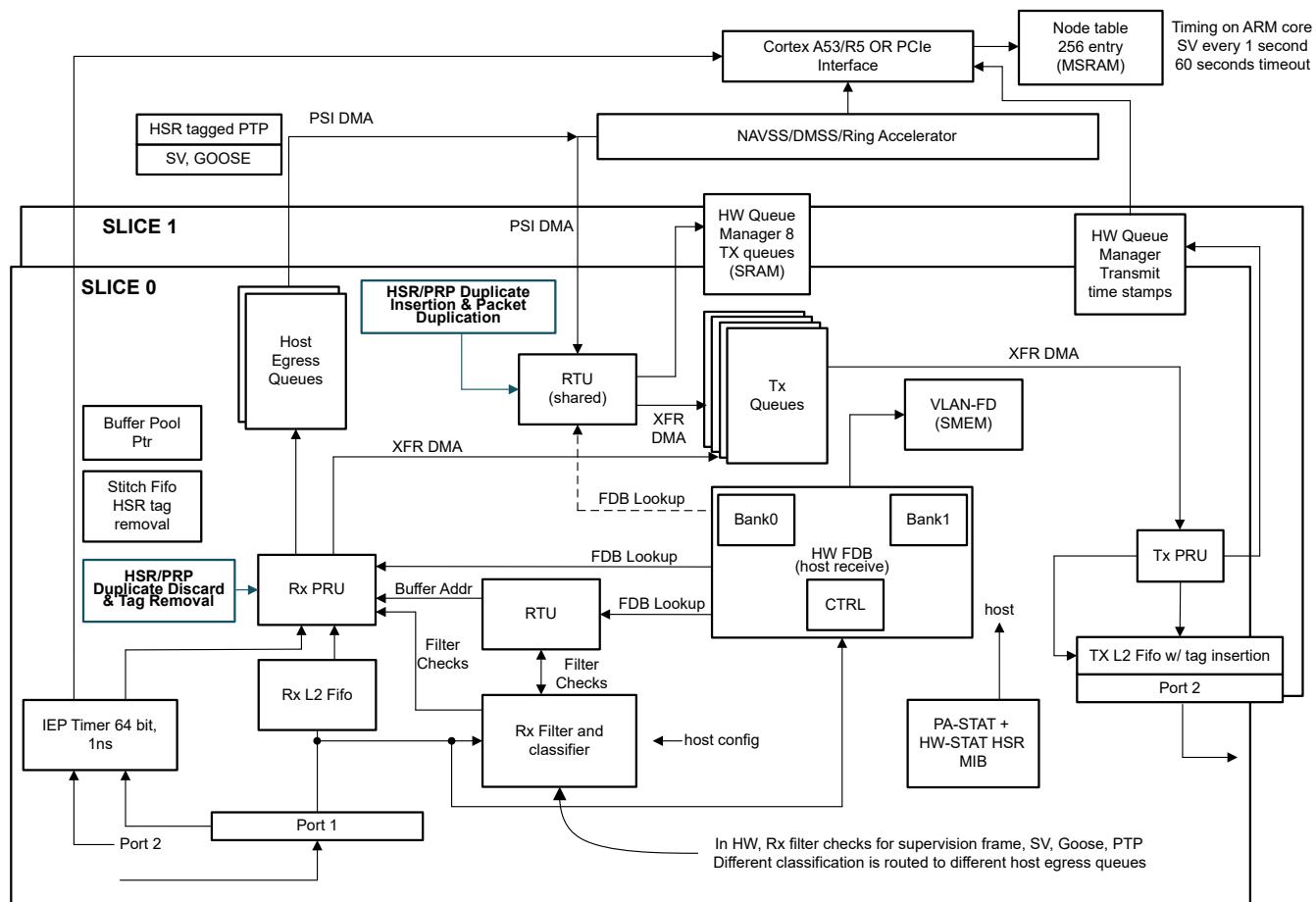


Figure 2-3. PRU_ICSSG HSR/PRP Firmware Block Diagram

2.5 Benefits of PRU_ICSSG HSR/PRP

The main benefit of using the PRU_ICSSG subsystem for HSR/PRP is in its capability to offload the main functions of HSR/PRP to the PRU cores. For example for HSR, the AM64x/AM243x has the ability to offload the following tasks: port-to-port forwarding, duplicating outgoing HSR frame, inserting unique tag to identify the HSR frame, and removing the HSR tag. This results in improvements including a reduction in CPU loading on the application cores (A-cores or R-cores on the AM64x/AM243x microprocessors) on the forwarding devices in an HSR network. In addition, in offloaded mode, forwarding latency is also decreased on the forwarding devices, which is especially beneficial in HSR networks.

Figure 2-4 shows a block diagram view of how offloading fits into an example system using an HSR topology. As mentioned in the introduction sections of this application note, an IED can be any device from protection relays to bay controllers. These devices handle functions such as protection, control, and metering. The ICSSG Module supports the previously described HSR offloaded tasks.

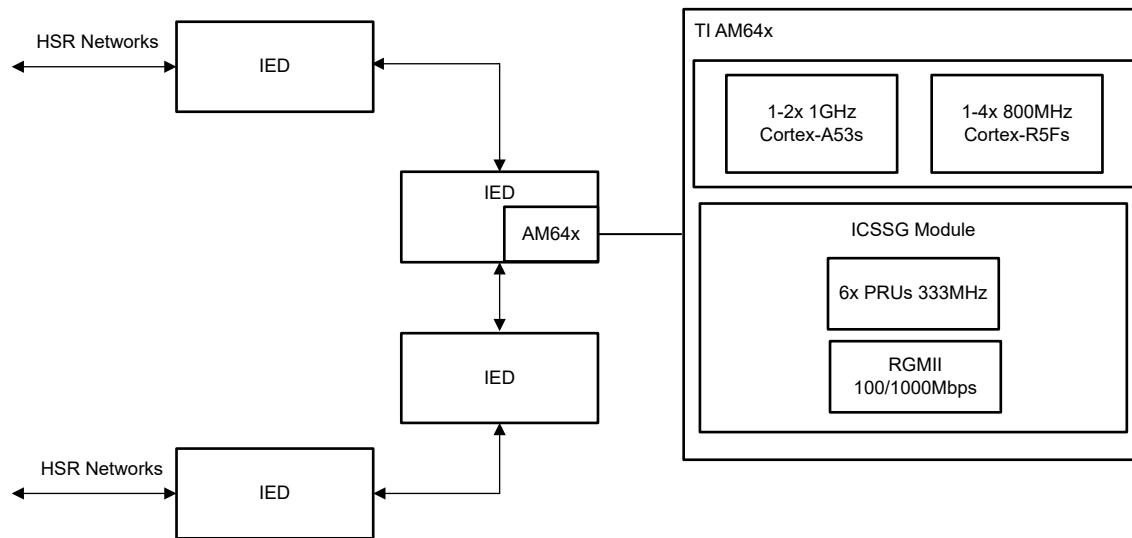


Figure 2-4. Example of AM64x used in an HSR Network

2.6 Linux HSR/PRP Solution

Since the AM243x microprocessor mainly features ARM cortex R-cores (does not contain A-cores) and TI offers Linux SDK support for A-cores, only the AM64x microprocessor (which features both A-cores and R-cores) supports Linux offering for HSR/PRP.

The [AM64x Linux Processor SDK](#) supports HSR/PRP hardware offload from the Linux PRU_ICSSG Ethernet driver to the PRU_ICSSG firmware. When the HSR/PRP firmware is loaded, the PRU Ethernet driver will perform the proper firmware configurations and indicate the offload capabilities in the `netdev` feature flag. Currently, the Linux HSR/PRP driver stack uses this feature flag to determine if hardware offload is supported or not at the lower level Ethernet driver and disables processing at its layer. Within the `netdev` feature flag, the Linux Kernel provides the following tags for hsr offloading: `hsr-dup-off`, `hsr-fwd-off`, `hsr-tag-ins-off`, and `hsr-tag-rm-off`. Please see the [Linux Kernel documentation](#) for detailed definitions of these tags.

The left part of [Figure 2-5](#), labeled "Hardware Non-offload HSR", shows components of the Linux HSR/PRP driver that implements the HSR/PRP functionality, without offloading any of the functions to the PRU_ICSSG. The right section of the figure, labeled "Hardware Offload HSR", shows greyed out Linux components that are not being used due to the respective HSR functions being offloaded to the PRU_ICSSG.

In general, the figures shows three different paths an Ethernet packet may take for both the non-offloaded and offloaded cases. Transmitted packets travel through the "Tx Packet Path" in the figure. Received packets travel through the "Host Rx Path". Finally, forwarded packets will travel through the "Forwarded Rx Path".

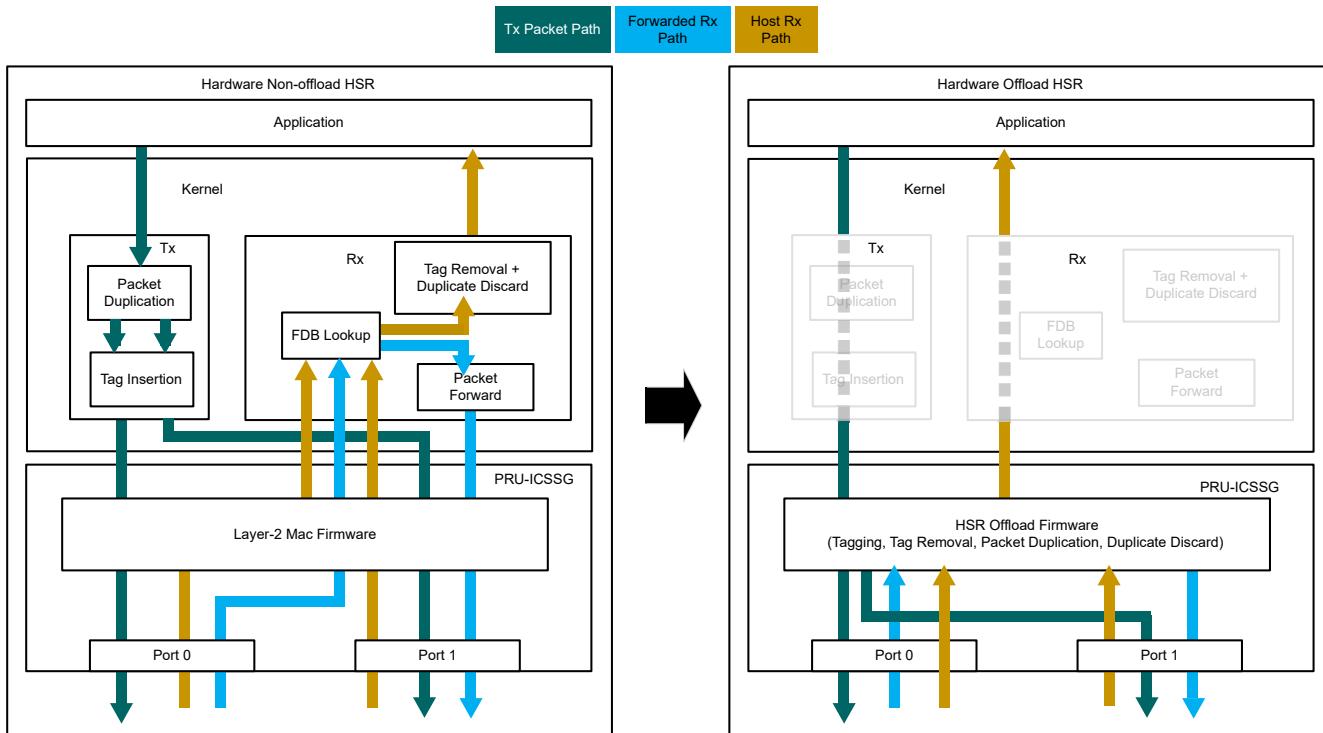


Figure 2-5. Linux Hardware Offload HSR/PRP Block Diagram

For more information on the Linux HSR/PRP offload solution regarding features, setup, test procedures, and performance; please see the latest AM64x Linux software documentation on [HSR Offload](#) or [PRP Offload](#).

2.7 RTOS HSR/PRP Solution

The [Industrial Communications SDK for AM64x](#) and the [Industrial Communications SDK for AM243x](#) features HSR and PRP support using RTOS. The Industrial Communications SDK packages are a single scalable software platform for Industrial Protocols and Drives designs that offers streamlined development across different TI Processors and MCUs. This package provides fundamental software for development of various HSR/PRP solutions. The packages also include sample demo applications that showcase the ability of the software/hardware to implement 1G HSR/PRP. For the Industrial Communications SDK to run effectively, the required MCU+ SDK package would need to be installed. Please see [1G HSR](#) and [1G PRP](#) for a quick start guide on how to test HSR and PRP with the Industrial Communications SDK.

Figure 2-6 shows how the SDK provided RTOS solution addresses offloading HSR/PRP functions to the PRU_ICSSG firmware. Please note that "Hardware Non-offload HSR" on the current SDK provided RTOS solution means that the HSR/PRP feature is completely disabled and there is no concept of running HSR/PRP directly on the R-cores. This is why the figure for "Hardware Non-offload HSR" does not show where the tagging, tag removal, packet duplication, duplicate discard is done. However, experienced RTOS developers are welcome to implement their own HSR/PRP stack to run on the R-cores.

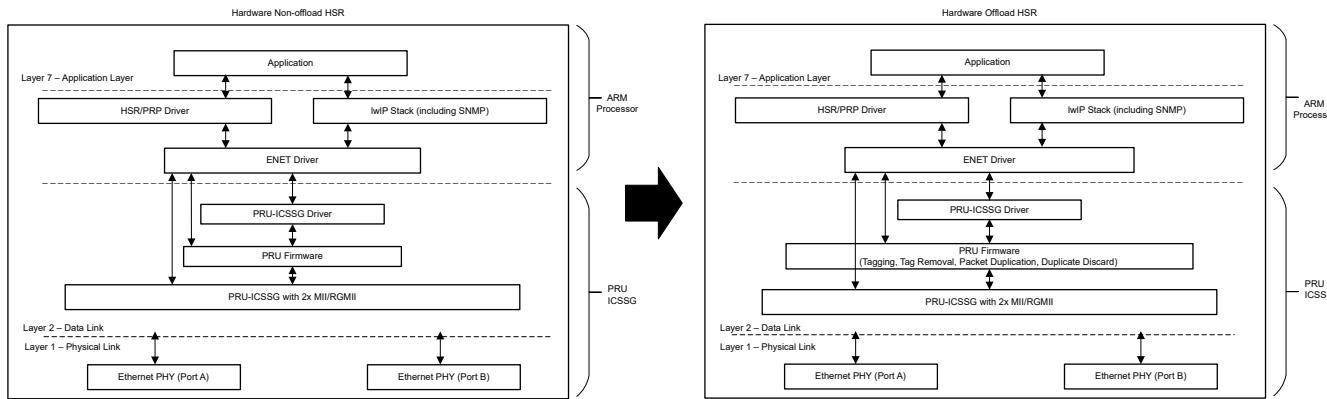


Figure 2-6. RTOS Hardware Offload HSR/PRP Block Diagram

3 HSR/PRP SDK Support

As mentioned in [Section 2.6](#) and [Section 2.7](#), TI provides a Processor Linux SDK and a Industrial Communications SDK for Linux and RTOS respectively. Both SDK packages include examples and documentation on how to implement and test HSR and PRP on AM64x and AM243x microprocessors.

Both SDK packages consists of the following:

- PRU_ICSSG firmware to implement 1G HSR/PRP LRE
- Associated drivers
- Stack adaptation layer
- Evaluation versions of protocol stack library for selected protocols
- Quick start examples

In regards to the AM3x/AM4x microprocessors, HSR/PRP functionality is also supported. These devices feature a PRU-ICSS subsystem. The main difference between PRU-ICSS and PRU_ICSSG is that the PRU-ICSS can only support up to 100Mbps bandwidth. Resources on the PRU-ICSS implementation of HSR/PRP can be found [here](#).

4 Summary

Texas Instruments supports High-availability Seamless Redundancy (HSR) and Parallel Redundancy Protocol (PRP) for real-time networking applications requiring high reliability of critical traffic. These protocols provide redundancy in smart grid substations where fast and fault tolerant communication is a key requirement.

Sitara microprocessors provides Linux and RTOS HSR/PRP solutions for AM64x/AM243x devices using the PRU_ICSSG subsystem. Using the PRU_ICSSG subsystem enables offloading key HSR/PRP functions, resulting in decreased CPU load and fowarding latency.

5 References

- AM64x/AM243x Technical Reference Manual
- PRU Academy Training
- AM64x Linux Processor SDK
- Linux Kernel Netdev Documentation
- Industrial Communications SDK for AM64x
- Industrial Communications SDK for AM243x
- PRU-ICSS HSR/PRP Firmware

6 Revision History

NOTE: Page numbers for previous revisions may differ from page numbers in the current version.

Changes from Revision * (April 2018) to Revision A (January 2026)	Page
• Included PRU-ICSSG Overview section.....	6
• Updated figures to reflect current technology support.....	6
• Added information on AM64x and AM243x processors supporting HSR/PRP on the PRU-ICSSG subsystem supporting 1Gbps bandwidth.....	7
• Expanded on the HSR/PRP Linux and RTOS support sections.....	7
• Added PRU-ICSSG HSR/PRP Benefits section highlighting the value of our offloading technology.....	9
• Updated HSR/PRP support section to include AM64x and AM243x families in addition to AM3x/AM4x microprocessors.....	12

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, regulatory or other requirements.

These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you fully indemnify TI and its representatives against any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to [TI's Terms of Sale](#), [TI's General Quality Guidelines](#), or other applicable terms available either on [ti.com](#) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products. Unless TI explicitly designates a product as custom or customer-specified, TI products are standard, catalog, general purpose devices.

TI objects to and rejects any additional or different terms you may propose.

Copyright © 2026, Texas Instruments Incorporated

Last updated 10/2025