Processor SDK7.1 – Ethernet LLD (enet_lld)

August 2020

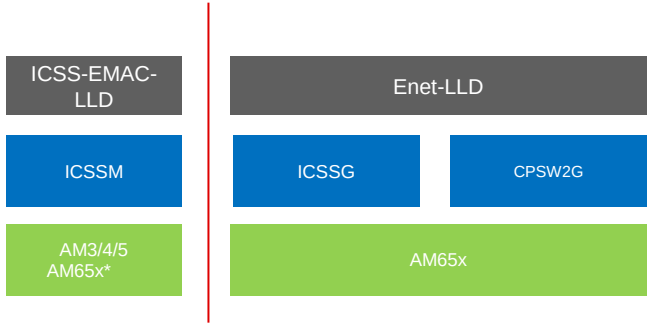
Platform SW Connectivity Team

Agenda

- Background
- Ethernet driver (Enet_LLD) -
  - Impact analysis
  - DMA improvements
  - API & IOCTL changes from emac_lld
- Migration path & Migration collaterals
  - Schedule

Background

- TI is migrating emac lld API and IOCTLS towards feature rich unified **Ethernet LLD (enet_lld)**
- This enables
  1. **DMA improvements** - Utilizing advanced UDMA features and enable performance improvements hooks like
     - Multiple packets submit/retrieve (batch processing)
     - Optimized data flow using UDMA – exposed ring mode, checksum offload, scatter gather, dedicated proxy per ring etc.
     - Faster packet processing by utilizing Low latency memory
     - **Multi-core data/DMA path** – enable direct packet routing to multiple SOC cores
  2. **PHY LLD –** better PHY mgt. with PHY driver supporting interrupt mode (for link detect), master slave config etc.
  3. Roadmap enablement to leverage/reuse high level protocol stacks like Timesync, TSN across on CPSW and ICSSG IPs for industrial and automotive apps
     - Define common interface catering to common industrial and automotive apps
  4. Scalability across product portfolio from DDR less devices to high end devices like J721E
     - Modular architectures that can work across SOC family
     - Performance entitlement & low Memory footprint to meet diverse set of use-cases
- This change **impacts AM65xx ICSSG and CPSW2G** and new devices going forward.
- Legacy devices and ICSSM applications (100M protocols) continue to use icss-emac lld with current feature set.



IP	Devices	Current	Going Forward
GMAC/ CPSW	AM3/4/5x	EMAC_LLD	EMAC_LLD
ICSSM	AM3/4/5x AM65x	ICSS EMAC_LLD	ICSS EMAC_LLD
CPSW2G	AM65x	EMAC_LLD	Enet_LLD
ICSSG	AM65x	EMAC_LLD	Enet_LLD

Fig – Ethernet driver – IP/SOC mapping

* when using 100M protocols

Impact Analysis – Applications

Sr.	Use-case	Impact	Remarks
1	a) NRT path	• Change in abstraction and interface layer	• TI will update stack abstraction layer with new APIs. • No impact on applications
	b) RT path	• NA	• No changes in RT path
2	TCP/IP with TI NDK	• None	• Change abstracted from users of NDK. TI would update NIMU as per new interface without impacting the apps
3	Third Party TCP/IP stack	• Abstraction layer needs to be adapted for new APIs	• The stack and driver abstraction layer needs to be updated for new APIs. • TI will provide reference abstraction layer for LwIP. For other 3rd party TCP/IP stacks migration guide can be referred.
4	L2 stacks – AVB, Ethernet/IP etc.	• Change in abstraction and interface layer	• The stack and driver abstraction layer needs to be updated for new APIs.
5	Time sync	• NA	• TI does changes in timestamping interface (psi dma)

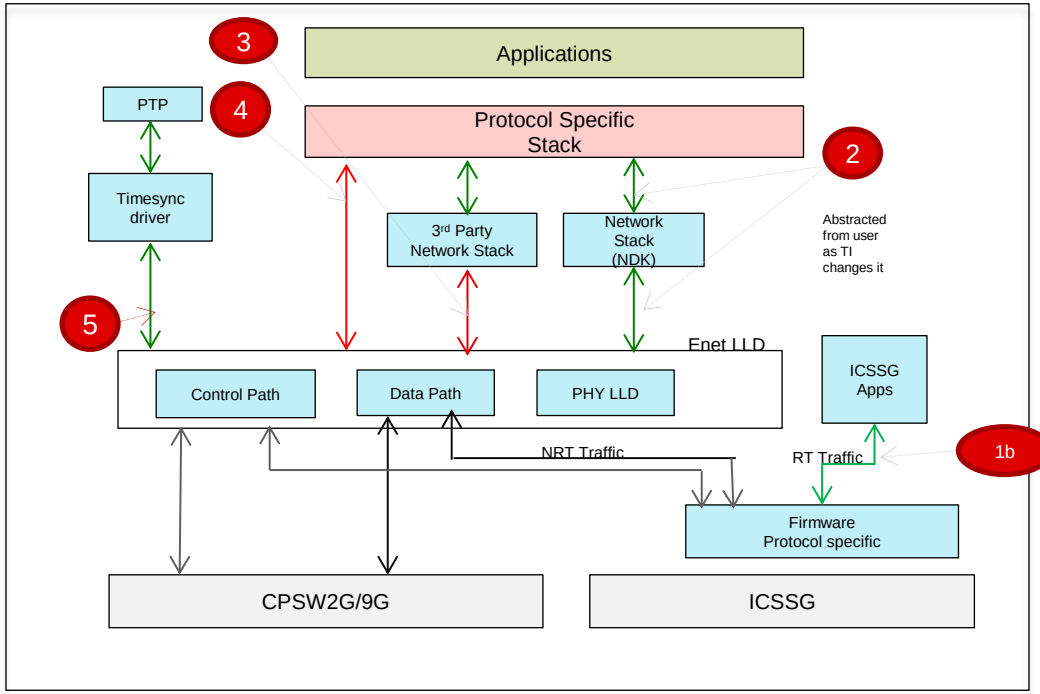
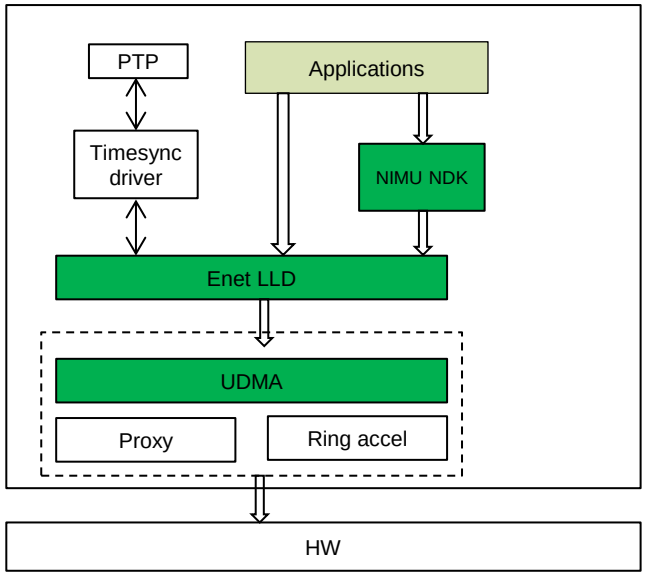


Fig – CPSW/ICSS-G RTOS stack

DMA Optimization Roadmap



DMA Optimizations Plan

Blocks optimized

	Module	SW Optimization	Improvement expected (w.r.t. baseline)	Remarks
	Baseline performance	TCP receive - **71.4Mbps@75%** R5 400MHz		
1	Core driver optimizations	Interrupt mode support	NA	Enables low latency path
		Interrupt Pacing	10%	SDK7.1
		Batch submit/retrieve APIs	10%	SDK7.1
		Checksum offload	40%	SDK7.1
		Eliminate SW queue (expose descriptor mode)	10%	SDK7.1
		Transmit deferred submission	5%	SDK7.1
		Low latency faster memory for packet data	10%	SDK7.1
2	UDMA optimizations	Exposed ring mode with doorbell (CQs)	8%	SDK7.1
		Dedicated proxy per flow	5%	SDK7.1
		UDMA multi-packet wrapper for ring mode	5% (estimated)	SDK7.1
	Improvement enabled (estimated)	TCP receive - **140Mbps@75%** R5 400MHz		

- Proposed LLD migration path comprehends above DMA optimization plan
- With this, 2X performance improvement would be achieved
- Goal is to make this available by SDK7.1

Enet LLD – Schedule & Migration plan

Sr. No.	Milestones	Timeline	Comments
1	Publish detailed API/IOCTLs Flow/sequence diagrams	**26th Aug 2020**	Documentation milestone.
2	Enet LLD – Sanity test complete	**21st Sept 2020**	Documentation milestone
3	Documentation - user, migration guide etc.	**28st Sept 2020**	Documentation milestone
4	SDK release with Enet LLD	**6th Nov 2020**	SDK7.1 release

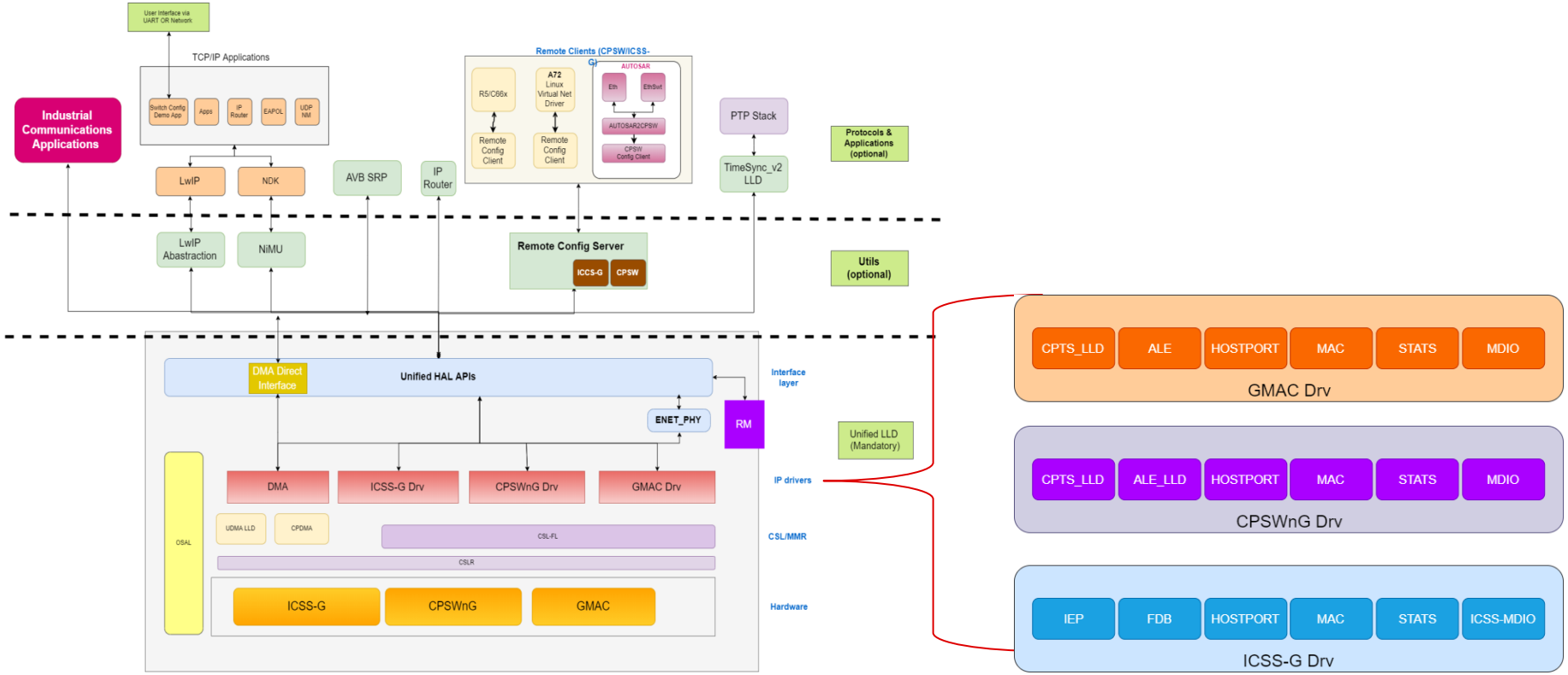
- **Migration documents**
  - API and IOCTL mappings – mapping document with emac lld API and IOCTLs describing the changes needed.
  - API guide and design document
  - Reference unit test and examples
    - Enet lld Unit test
    - Reference industrial apps examples
  - Migration videos
    - Video describing the changes and reference migration
- **Migration Support**
  - Migration sessions to walk through changes
    - Debug sessions if needed.
  - Support via e2e

- Documentation milestones collaterals would be uploaded to FAQ e2e on the timeline date.
- For additional information and documentation feedback, please contact TI representative.

Ethernet LLD – Folder Structure (Tentative)

```
enet/
|- docs
|- src
|   |- core
|   |- common
|   |- dma
|   |- per
|   |   |- cpsw.h
|   |   |- iccsg.h
|   |   └- gmac.h
|   |- mod
|   |   |- cpsw_*.h
|   |   └- icssg_*.h
|   └- phy
|- include
|   |- core
|   |   |- enet_types.h
|   |   |- enet_per.h
|   |   |- enet_mod.h
|   |   └- enet_mod_*.h
|   |- common
|   |- dma
|   |   |- enet_udma.h
|   |   └- enet_cpdma.h
|   |- per
|   |   |- cpsw.h
|   |   |- iccsg.h
|   |   └- gmac.h
|   |- mod
|   |   |- cpsw_*.h
|   |   └- icssg_*.h
|   └- phy
|- examples
|   |- enet_nimu_example
|   └- enet_loopback_example
|- tools
|- unit_test
└- lib
    |- enetcore.lib
    |- enetsoc.lib
    |- enetcpsw.lib
    |- eneticssg.lib
    |- enetdma.lib
    └- enetphy.lib
```

Ethernet LLD – Block Diagram (tentative)

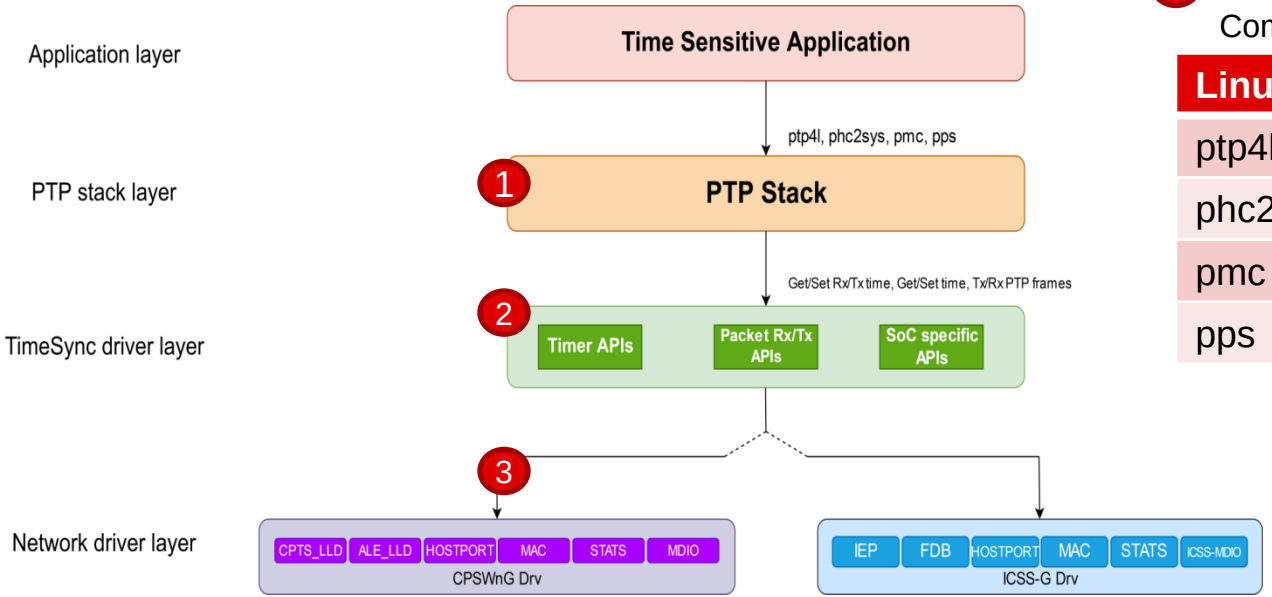


Version History

Version	Author	Date	Revision
0.1	TI Internal	Aug 24th 2020	Initial version

Backup

Time Sync stack diagram



1 **PTP stack to application APIs**
Comparison with Linux PTP

Linux PTP	TimeSync PTP
ptp4l	TimeSync_initPtp
phc2sys	TimeSync_coupleTimer
pmc	TimeSync_ptpMgmt
pps	TimeSync_configPps

TimeSync driver layer refers to Time sync HAL abstracting underlying network driver layer