Processor SDK7.1 – Ethernet LLD (enet_lld)

August 2020

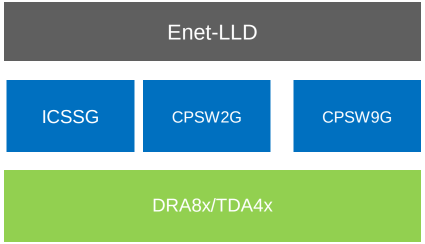
Jacinto Team

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. **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
  2. 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
  3. DMA improvements - Utilizing advanced UDMA features and enable performance improvements hooks like
     - Improved performance on gateway applications and TCP/IP stack.
       - » Optimized data flow using UDMA – scatter gather etc.
       - » Faster packet processing by utilizing Low latency memory
- This change **impacts J721E CPSW & ICSSG** RTOS drivers and **new devices** going forward.
- Legacy devices emac lld with current feature set.

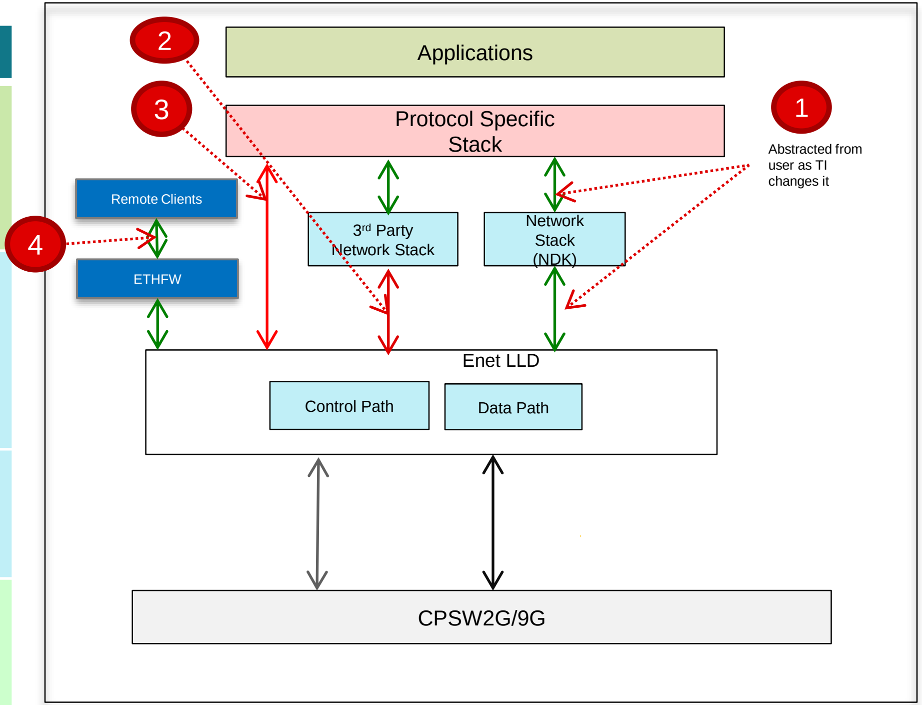


IP	Devices	Current	Going Forward
CPSW2G	TDA4xx/ DRA8xx	CPSW_LLD	Enet_LLD
CPSW9G	TDA4xx/ DRA8xx	CPSW_LLD	Enet_LLD
ICSSG	TDA4xx/ DRA8xx	EMAC_LLD	Enet_LLD

Fig – Ethernet driver – IP/SOC mapping

Impact Analysis – Applications

Sr. No.	Use-case	Impact	Remarks
1	TCP/IP with TI NDK	• None	• Change abstracted from users of NDK. TI would update NIMU as per new interface without impacting the apps
2	TCP/IP with TI NDK or 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.
3	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.
4	CPSW remote clients – MCAL, Linux etc.	• No impact.	• Change abstracted by proxy layer



Abstracted from user as TI changes it

CPSW/ICSS-G RTOS stack

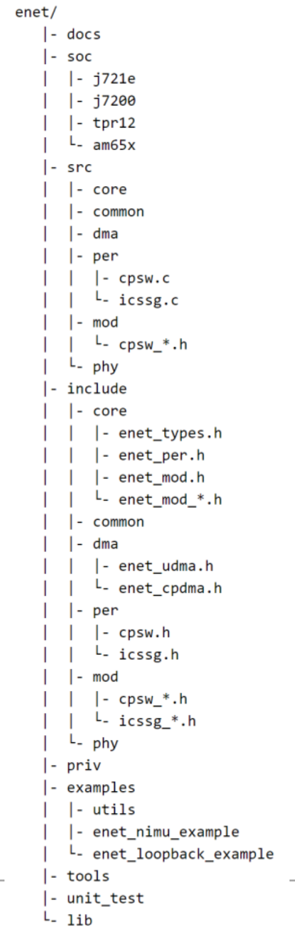
Enet LLD – Schedule & Migration plan

Sr. No.	Milestones	Comments
1	Publish detailed API/IOCTLs Flow/sequence diagrams	Documentation milestone.
2	Enet LLD – Sanity test complete	Documentation milestone
3	Documentation - user, migration guide etc.	Documentation milestone
4	SDK release with Enet LLD	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

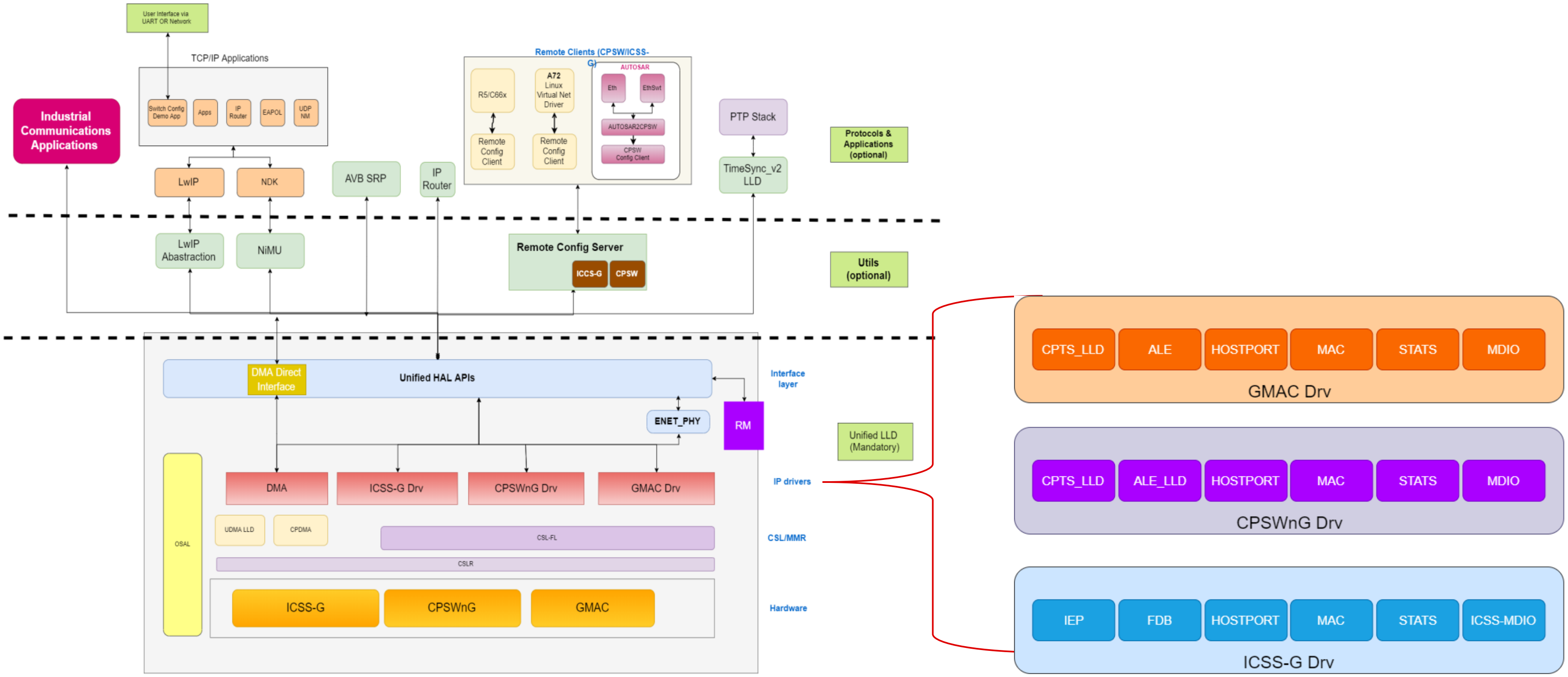
- Documentation milestones collaterals would be uploaded to FAQ e2e on the timeline date.
- For additional information and documentation feedback, please contact TI representative.

Ethernet LLD – Folder Structure



```
enet/
   |- docs
   |- soc
   |  |- j721e
   |  |- j7200
   |  |- tpr12
   |  └- am65x
   |- src
   |  |- core
   |  |- common
   |  |- dma
   |  |- per
   |  |  |- cpsw.c
   |  |  └- icssg.c
   |  |- mod
   |  |  └- cpsw_*.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
   |  |  └- icssg.h
   |  |- mod
   |  |  |- cpsw_*.h
   |  |  └- icssg_*.h
   |  └- phy
   |- priv
   |- examples
   |  |- utils
   |  |- enet_nimu_example
   |  └- enet_loopback_example
   |- tools
   |- unit_test
   └- lib
```

Ethernet LLD – Block Diagram (tentative)

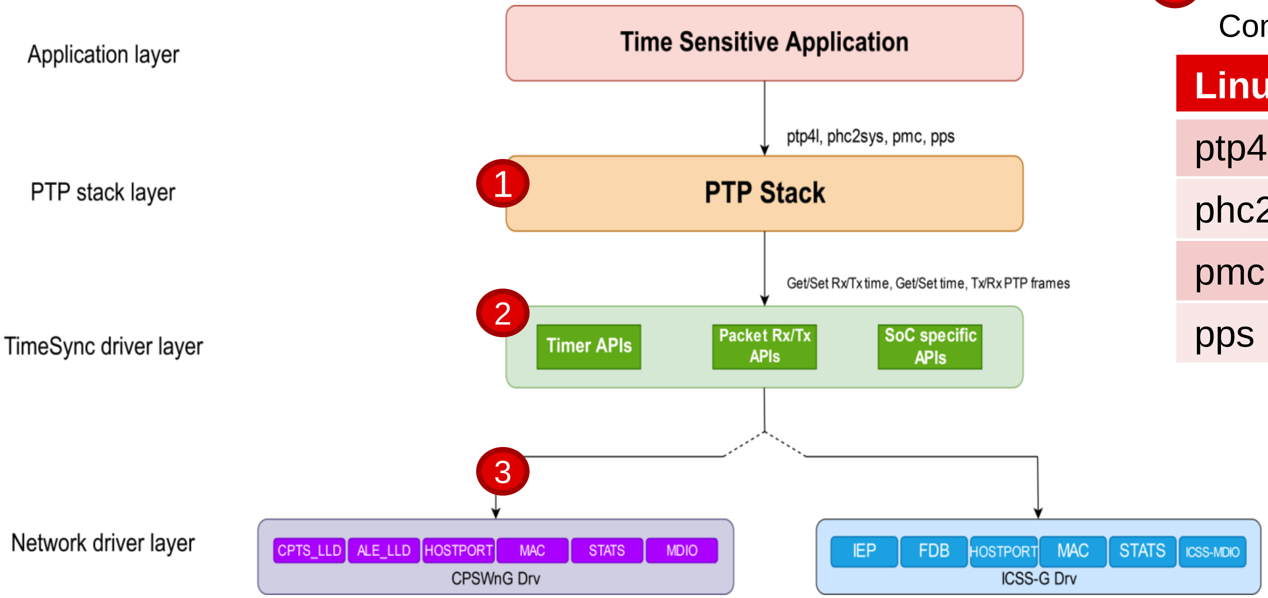


Version History

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