THE LINUX FOUNDATION

Embedded Linux Conference

SAMSUNG

Automotive Ethernet: Future of Connected Vehicles

Ravi Patel, Sriranjani P

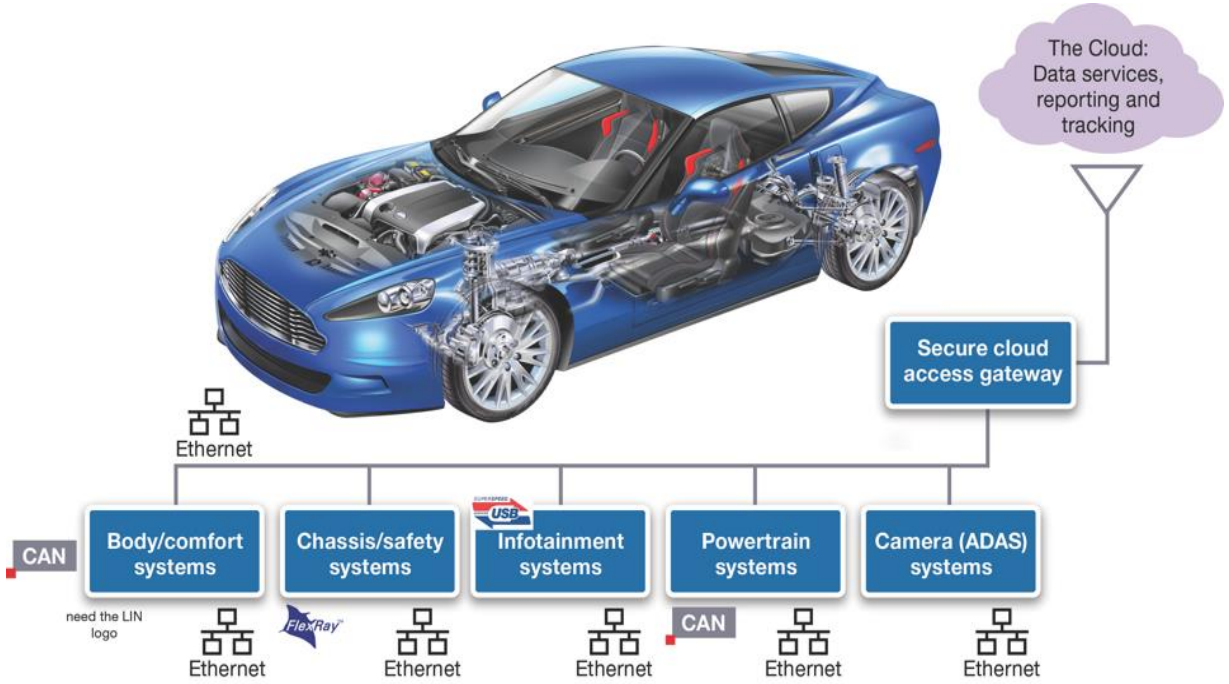
SSIR Samsung Semiconductor India R&D Center

Agenda

- Automotive Systems and Network
- Current Trends in Automotive Networks
- The Future Requirements
- Motivation behind Automotive Ethernet
- Why not Conventional Ethernet?
- History and Evolution of Automotive Ethernet
- Protocol formats
- Different Technologies
- Support of Automotive Ethernet in Linux
- Automotive Ethernet PHY
- Conclusion

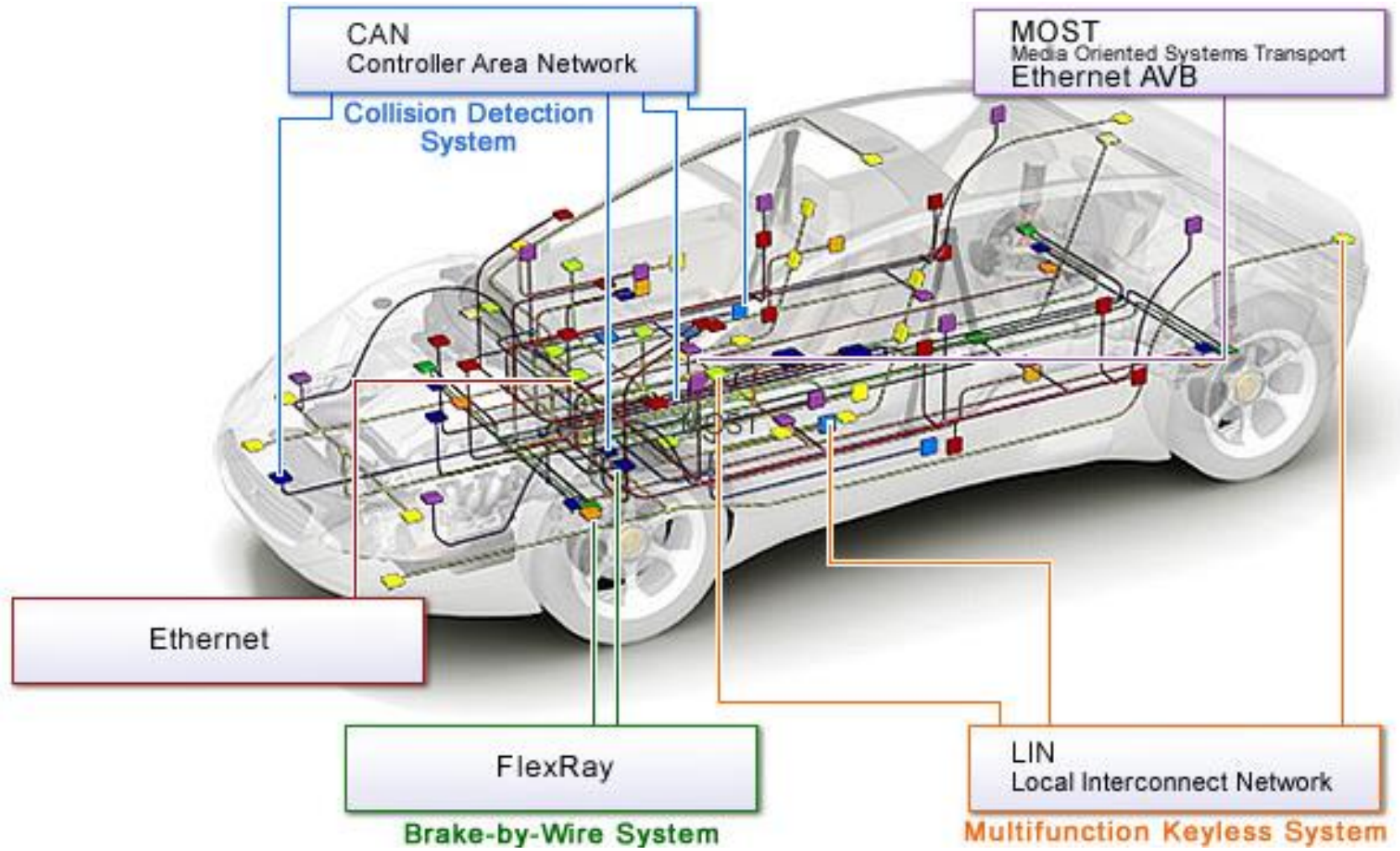
Automotive Systems and Network

- An automotive vehicle contains lot of sensors, actuators and controllers
- Connected by simple wire during initial days
- Evolutions of components and its complexity increased over time
- Requirement of serial bus or fieldbus instead of simple wire to fulfill requirements
- Wiring harness
  - 3rd highest cost component in a car
  - comprise 50% of the cost of labor for the entire car



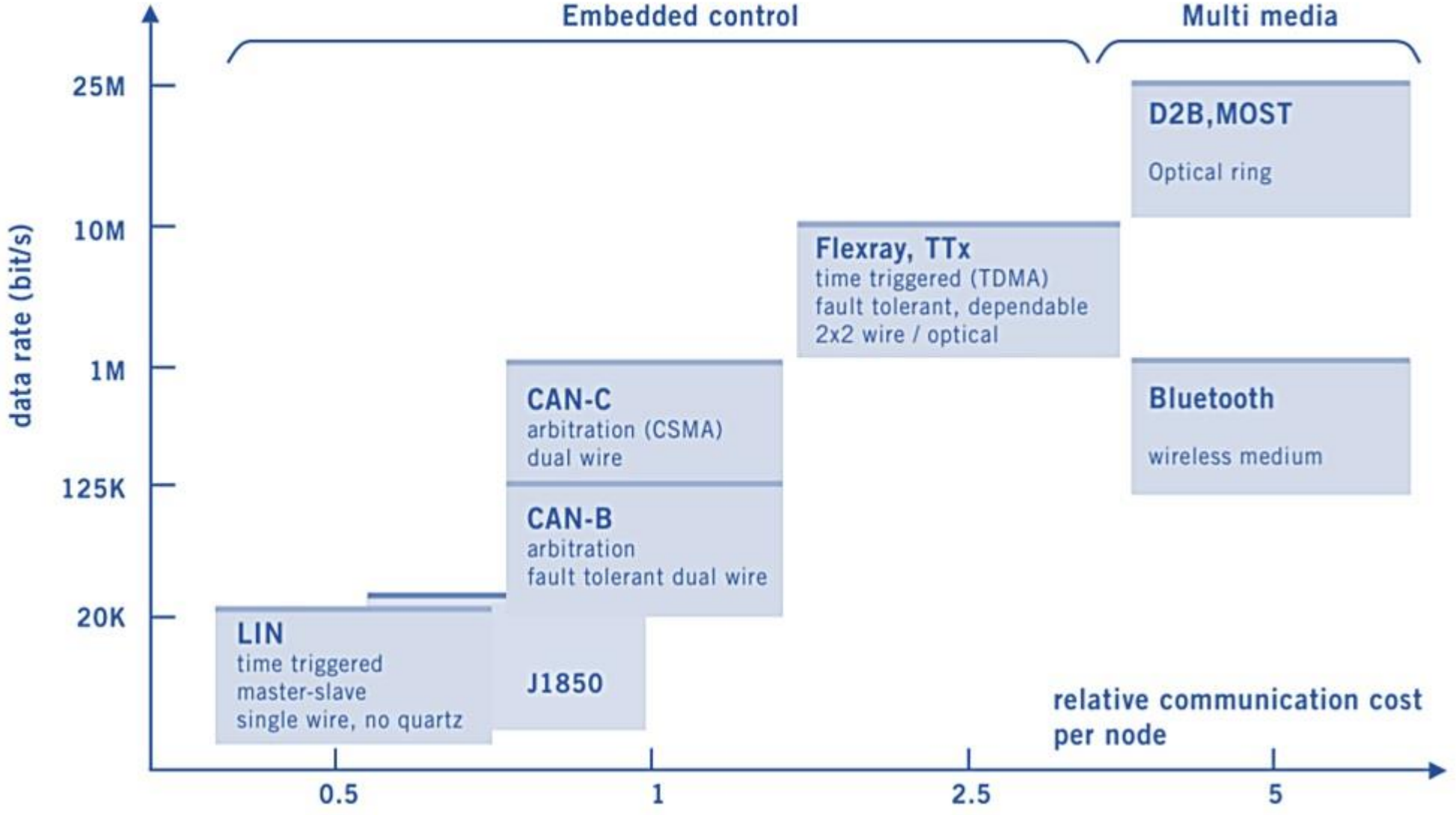
Source: https://bit.ly/39XyFfs

Current Trends in Automotive Networks



Source: https://bit.ly/3yMvfXo

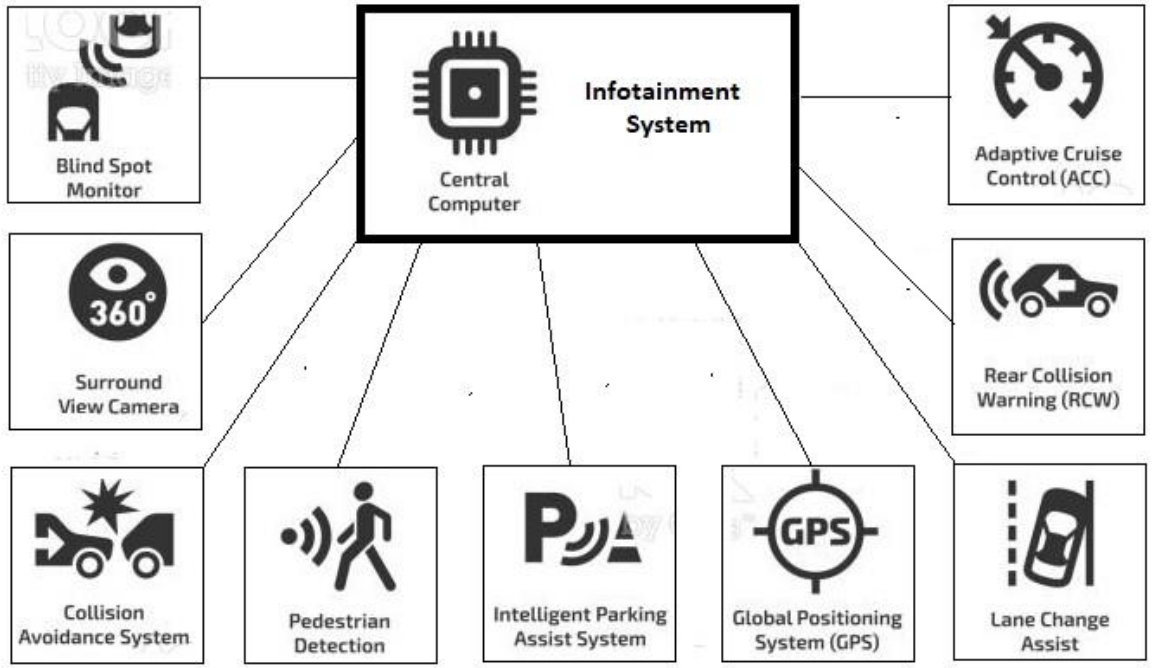
Current Trends in Automotive Networks



Source: https://bit.ly/3MwqBRl

The Future Requirements

- More bandwidth needed for new systems like Adaptive Cruise Control, Lane Detection, Around view monitoring etc.
- Keep number of wires less to reduce the wire complexity
- Software driver support in open source OS (e.g. Linux) for easy integration with infotainment system
- Low latency, reliable and real time to support systems like Cruise Control, Emergency Breaking, Stability Control system etc.



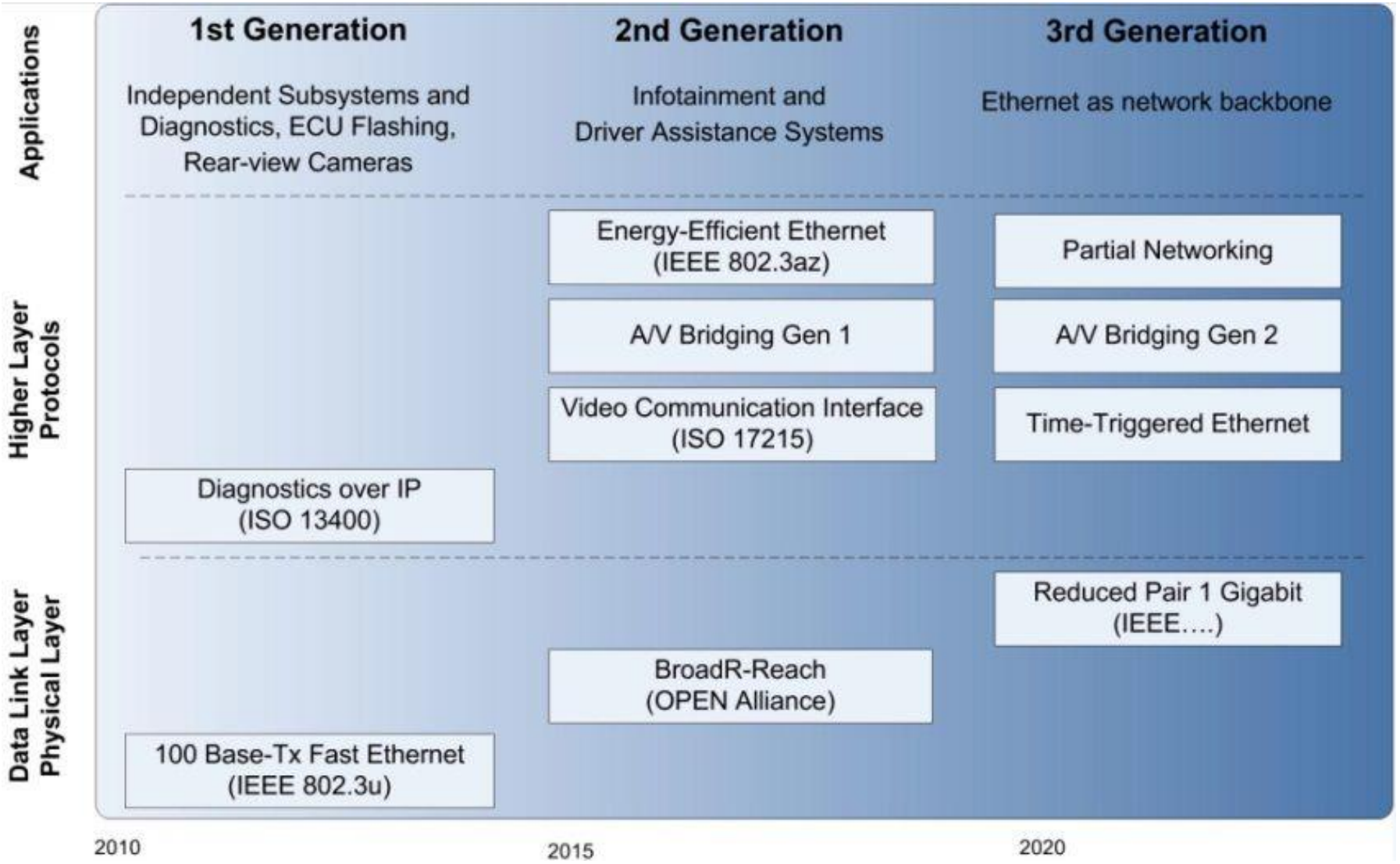
Motivation behind Automotive Ethernet

- High Bandwidth requirement
  - Row camera data
  - Data Logging (Government Regulations)
  - Map Data
  - High Resolution Displays
  - For instance, the LIDAR sensors needed for lane detection and other driver assistance applications require a 70 Mbps connection just for one sensor
- Reliable and cost effective network
- Minimum latency and precise timing
- Redundancy and security
- Precise Time Awareness
- Easy integration with TCP/IP protocol
- Lower silicon costs and space

Why not Conventional Ethernet?

- Higher number of wires
- Does not meet the OEM EMI/RFI requirements
- Does not provide the guaranteed latency in us range
- No support for bandwidth control for different streams
- No time synchronism mechanism
- Harsher environment conditions
  - Operating temperatures. (-40°C to 85°C for body/cabin and up to 125°C for chassis/powertrain)
  - Mechanical accelerations (up to 4G)
  - Automotive EMC requirements
- Safety/ASIL compliance
- Reliability (high MTBF)
- Very low standby power requirements
  - Standby power << 100uA
  - Wakeup time < 100-500ms

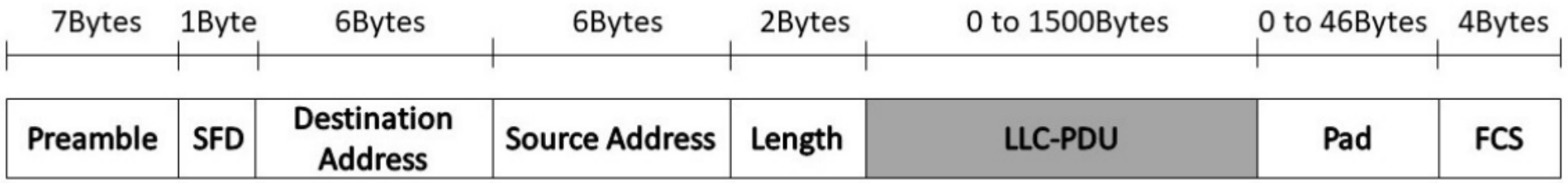
History and Evolution of Automotive Ethernet



Source: https://bit.ly/3lmGjmb

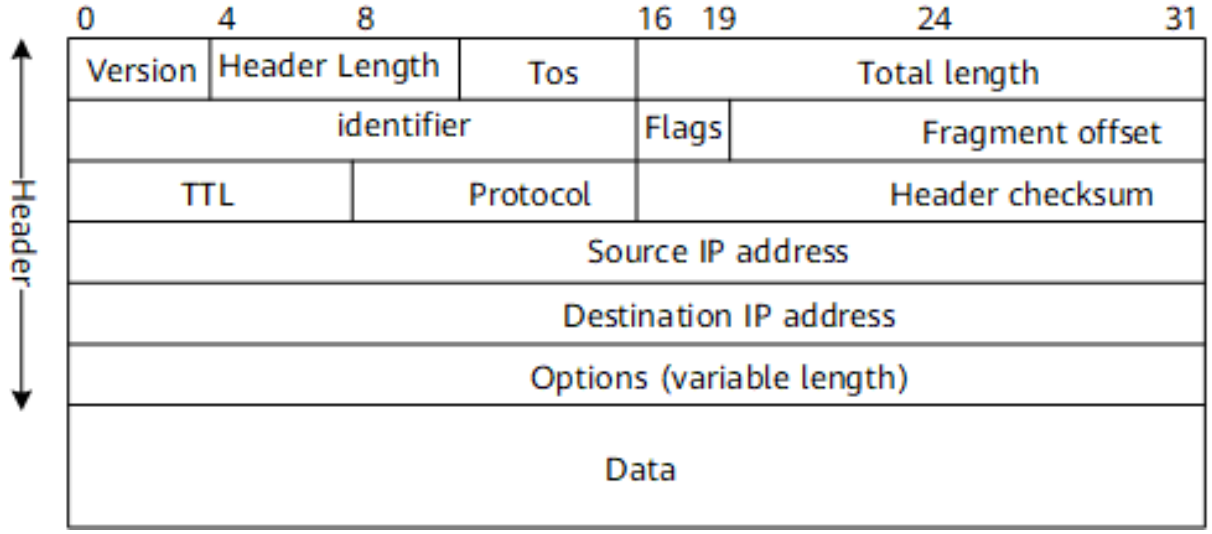
Protocol formats

- Link-Layer Frame (same as conventional Ethernet)



Source: https://bit.ly/38C0xFN

- IP layer (same as conventional Ethernet)



Source: https://bit.ly/38uxOml

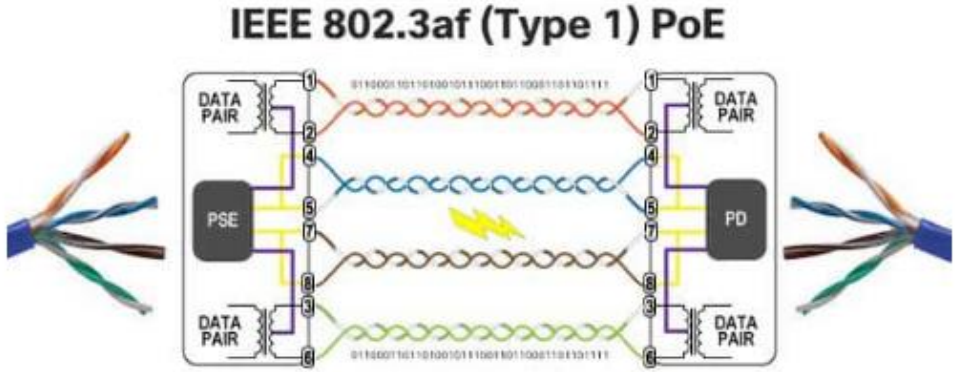
Different Technologies

- AUTOSAR (Automotive Open System Architecture)
  - Formed in 2003 by major automotive OEMs to promote open standard automotive architecture
  - Provides specifications of basic modules, application interfaces and data exchange standard
  - Helps in establishing common ECU software architecture
  - Uses three layer architecture: Basic Software, Runtime Environment and Application Layer

- OPEN (Open Pair Ether-Net)
  - A non-profit Special Interest Group (SIG) to encourage wide scale adoption of Ethernet based communication for in-vehicle networks
  - Formed in 2011 by BMW, Broadcom and NXP
  - Has introduced standards for testing the in-vehicle Ethernet systems in switches, ECUs.
  - Helped in deploying 100BASE-T1, 1000BASE-T1, and 1000BASE-RH physical layers to be used within Automotive Ethernet.

Different Technologies

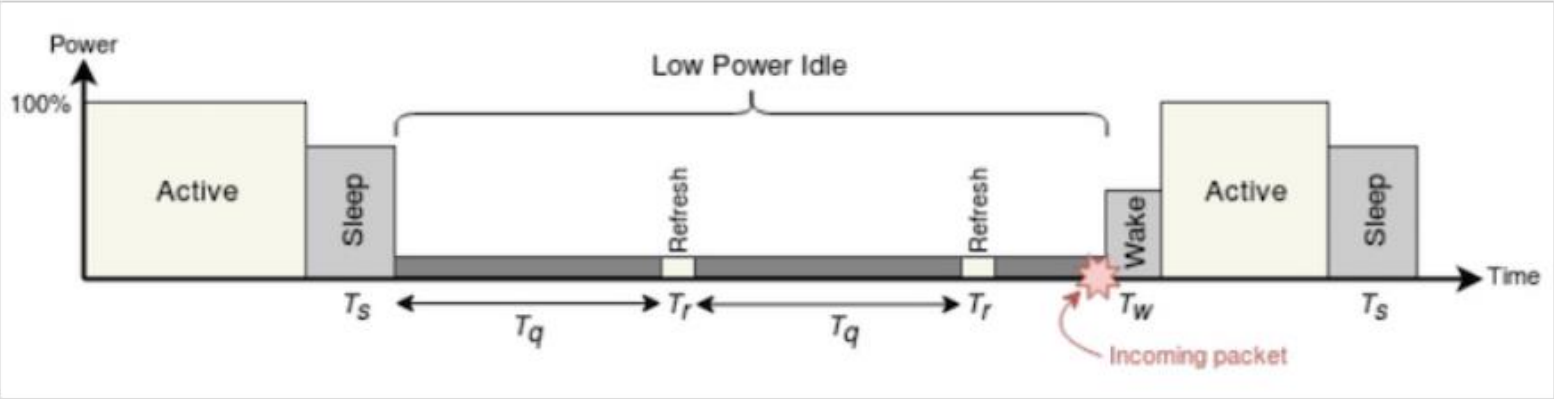
- PoE (Power over Ethernet)
  - Originated in 2003 as IEEE 802.3af and introduced as IEEE 802.3bu in 2016 for Single Pair Ethernet which is intended for automotive applications.
  - Powers the devices in the vehicle and eliminates the requirements of additional power sources.
  - Reduces the wiring and its complexity.
  - Protects the device from overload, under-powering and incorrect installation.



Source: https://bit.ly/3yK1GFT

Different Technologies

- EEE (Energy Efficient Ethernet)
  - Specified as IEEE 802.3az in 2006 which attempts to save power on inactive Ethernet links.
  - Node sends idle packet over link at specific interval when no data.
  - Helps to save battery when engine is off.



Source: https://bit.ly/3sIRKZs

Different Technologies

- Time Synchronization
  - Known as Time Sensitive Networking (TSN) which was formed in 2012 by renaming existing Audio Video Bridging (AVB) group.
  - Standardized as IEEE 802.1AS which defines a protocol to synchronize reference time between distributed nodes in network
  - Best clock determined by a best master clock algorithm which distributes clock information to all other capable nodes.
  - Used by time critical automotive applications like ADAS.

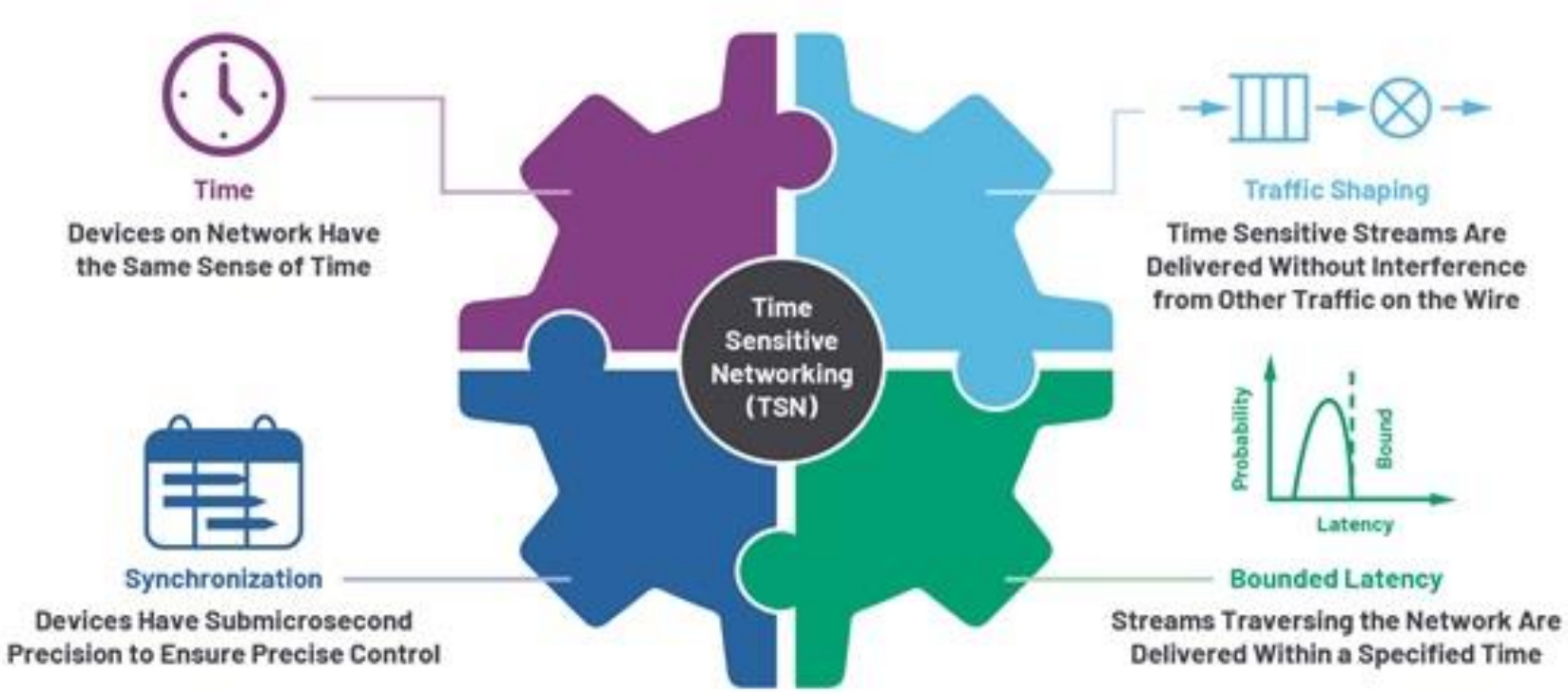
- Diagnostics Over IP
  - Vehicle diagnostics protocol based on ISO 13400 standard to analyze data from on-board computers and update firmware
  - Enables remote vehicle diagnostics by managing communication between external tester tool and ECUs.
  - Uses dedicated diagnostics Ethernet connections and runs over TCP/IP.
  - Allows much data rates at lower cost compared to conventional CAN based diagnostics.

Support of Automotive Ethernet in Linux

Time Sensitive Networking (TSN)

Time Sensitive Networking (TSN)

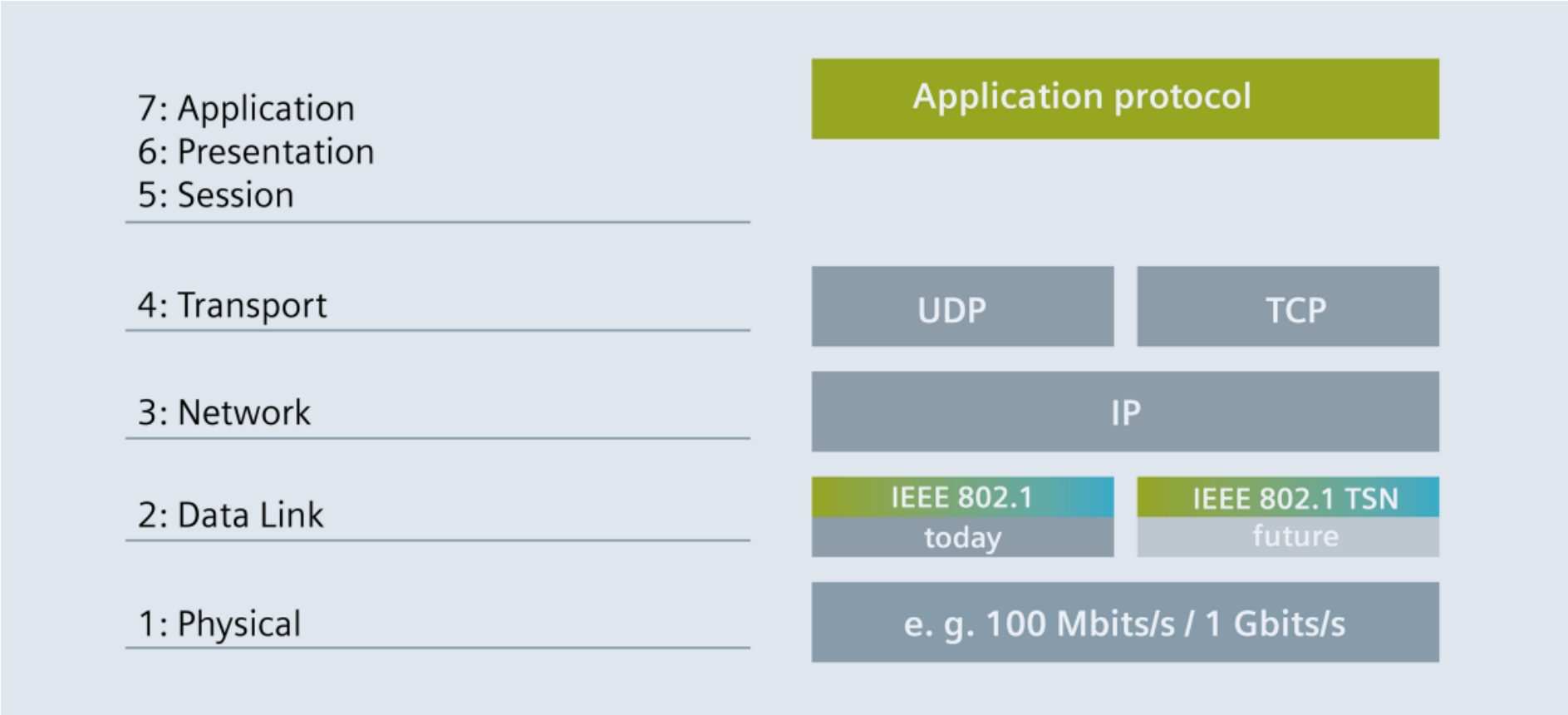
- Set of Standards developed by IEEE Time-Sensitive Networking Task Group.
- Formed in 2012 from existing Audio/Video Bridging (AVB) Task Group.



Source: https://bit.ly/3FVe3k1

Time Sensitive Networking (TSN)

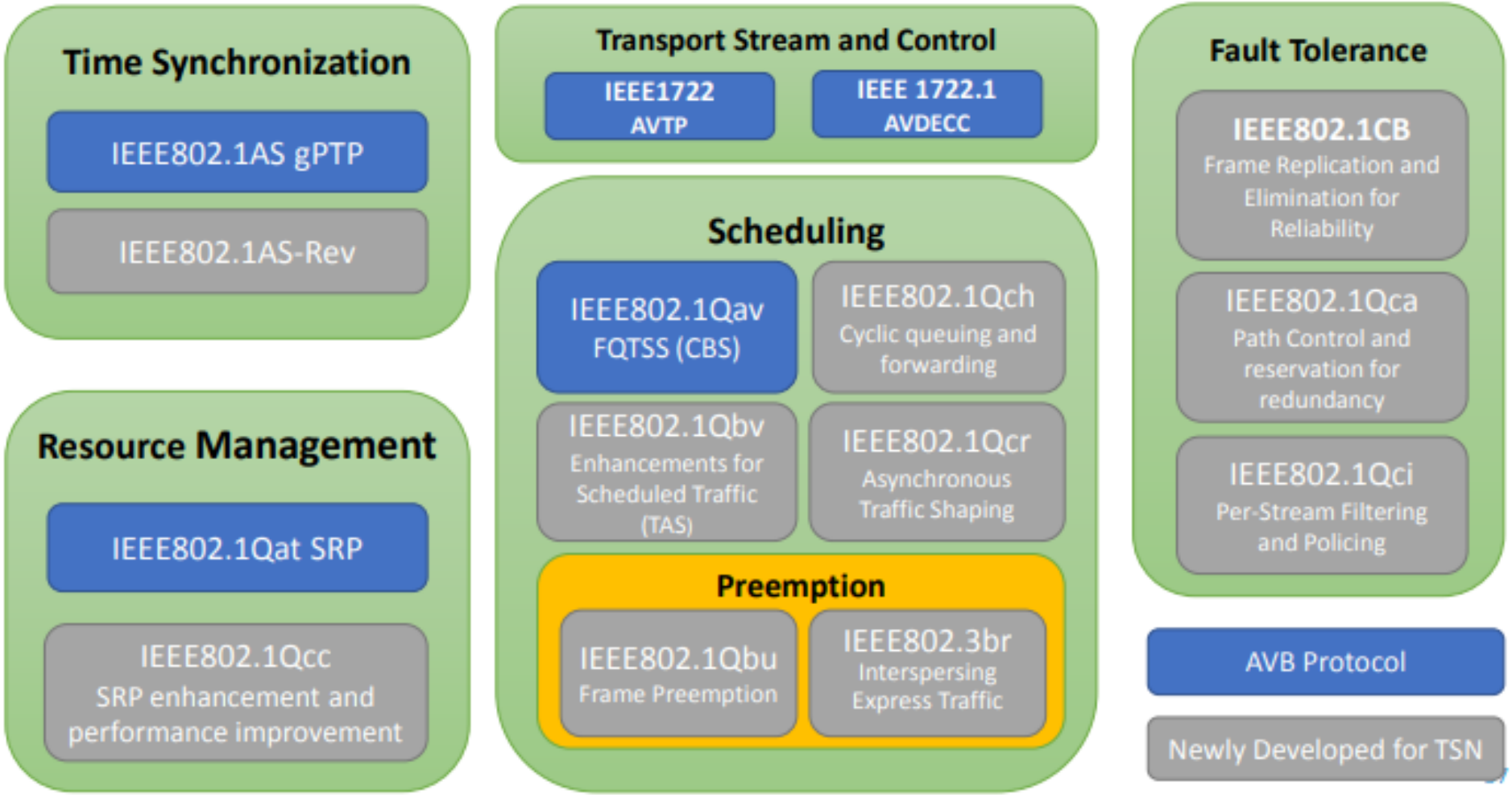
- Applicable to Data link layer.



Source: https://sie.ag/39znc5x

Time Sensitive Networking (TSN)

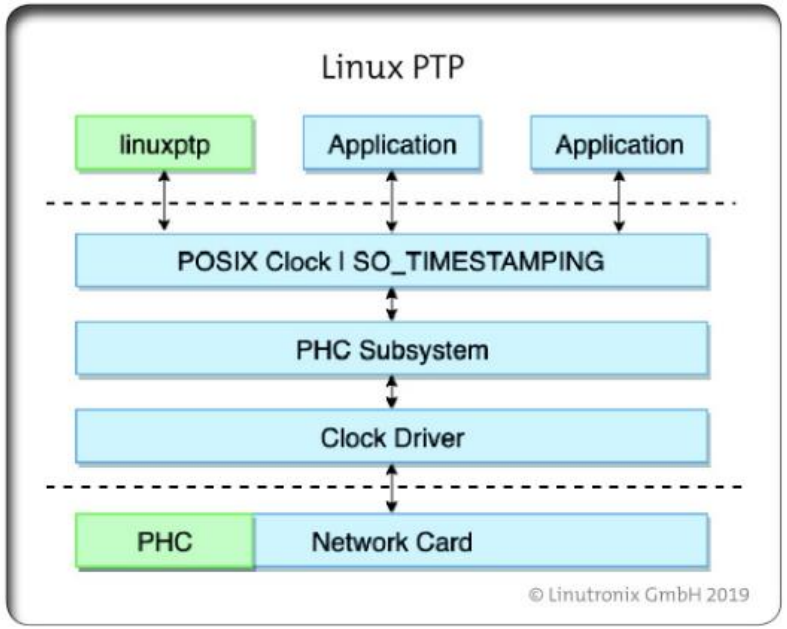
- TSN and AVB Protocols



Source: https://bit.ly/3FTTEMh

Time Synchronization

- Known as PTP (Precision Time Protocol)
- Much better then NTP (Network Time Protocol)
- Allows synchronization with an accuracy in nanoseconds
- Protocol:
  - Master clock and node is selected by Best Master Clock Algorithm (BMCA)
  - Master node provides clocking information to other nodes
  - All nodes must support PTP for effective time synchronization

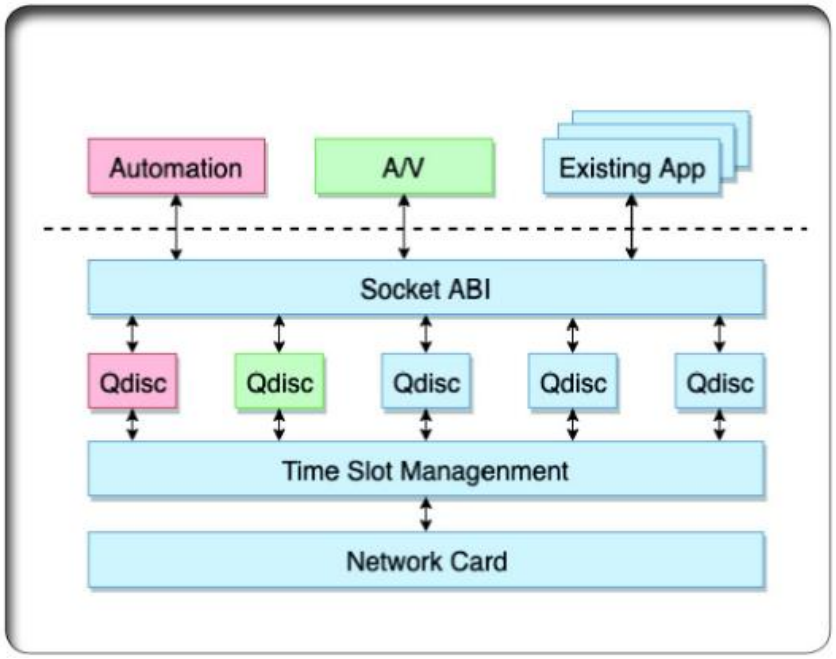


Source: https://bit.ly/3PtGWbq

- Kernel offers its own subsystem for controlling PTP hardware clocks (PHC).
- PTP at user level, kernel just provides hardware access to clocks
- Linuxptp – most popular user space PTP stack
- Linuxptp applications:
  - Ptp4l – Implementation of PTP
  - Ptp2sys – synchronize two clocks
  - Pmc – Send PTP management messages to PTP nodes

Traffic Scheduling

- 802.1Qbv
- TSN control plane is implemented through Linux Traffic Control (TC) System.
- Supported via TC Queuing Disciplines (Qdiscs).
- Qdisc – A packet scheduler which decides time when packet is given to network hardware or application.
- Linux currently provides below qdiscs for TSN:
  - CBS qdisc : Credit Based Shaper introduced by 802.1 Qav
  - Time-Aware Priority Shaper (TAPRIO) qdisc: Implements simplified version of 802.1Qbv standard
  - Earliest TxTime First (ETF) Qdisc : enables the Lunchtime feature present in some NICs



Source: https://bit.ly/3G4Gex6

Switches and switched end devices

- TSN does not have only end devices. Switches and switched terminals also present.
- Linux offers two frameworks for this:
  - The Distributed Switch Architecture (DSA)
  - Switchdev

- DSA
  - Introduced in 2008 to support Marvell switches
  - Evolved to support other venders also
  - Concept of combine several individual switches to common logical component

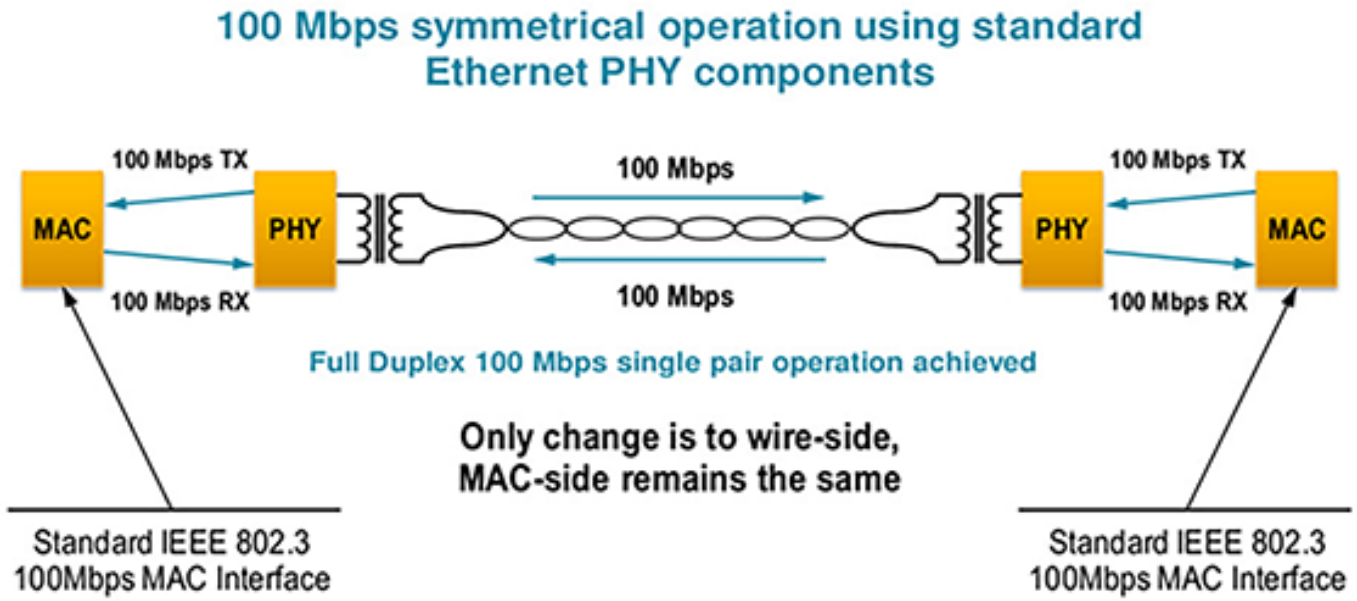
- Switchdev
  - Focus on offloading as much work as possible to the hardware
  - Not a Linux device model in traditional sense

Summary of TSN in Linux

Standard	Alias	Linux Support	Linux Alternatives
802.1AS	Network Timing & Synchronization	In parts	Yes
802.1Qav	Credit Based Shaping	Yes	Yes
802.1Qbv	Traffic Scheduling	In parts	Yes
802.1Qbu	Frame Preemption	In progress	Yes
802.1Qbr	Frame Preemption	No	Yes
802.1Qca	Path control & Reservation	No	Yes
802.1Qcc	Stream Reservation	No	Yes
802.1Qch	Cyclic Queuing	No	Yes
802.1Qci	Pre-Stream Filtering	No	Yes
802.1CB	Frame Replication & Elimination	No	Yes

Automotive Ethernet PHY

- BroadR-Reach automotive Ethernet standard – release in 2011
- Later on IEEE 802.3bp (100BASE-T1) and 802.3bw (1000BASE-T1) was standardized



Source: https://bit.ly/3FWjUFD

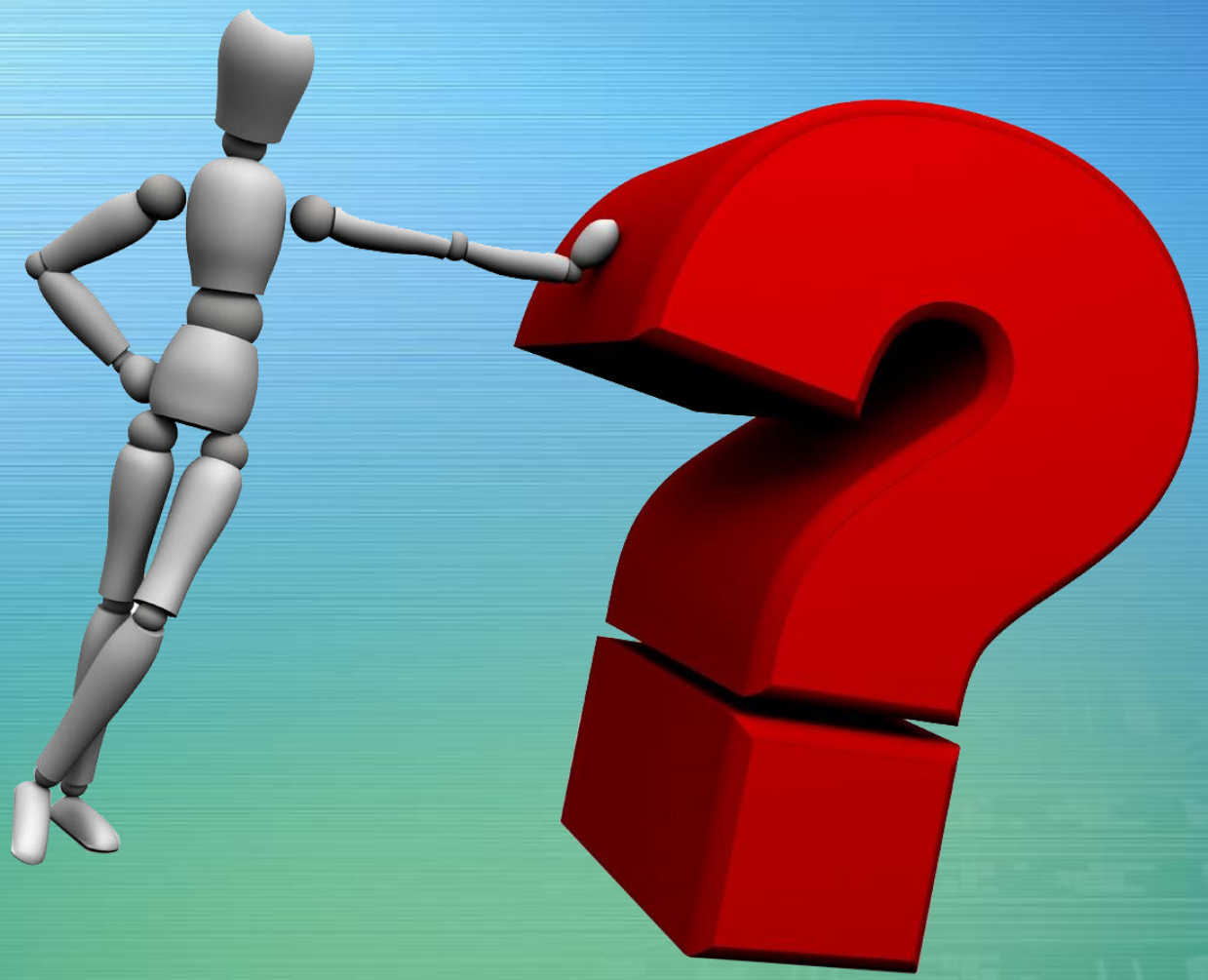
Automotive Ethernet PHY

100Base-T1	1000Base-T1
Developed for 100Mbps	Developed for 1000Mbps
Multi Level PAM-3 coding	Multi Level PAM-3 coding
600MHz bandwidth	600MHz bandwidth
Specified in 802.3bp	Specified in 802.bw
Supports single twisted pair upto 15 meters	Supports single twisted pair upto 15 meters with copper and upto 40 meters with optical fiber
Full duplex mode	Full duplex mode
Supports EEE (Energy Efficient Ethernet) as a optional	-

Conclusion

- Automotive Technology evolved from simple IC engine to a moving combination of integrated computer systems like ADAS, Adaptive Cruise Control, hybrid engines, smart infotainment etc.
- Cabling in a vehicle is 3rd highest cost and 3rd heaviest component. Simplifying and reducing cabling reduces:
  - Fuel consumption
  - Repair issues
  - Manufacturing cost
  - Production time
- The recent advancement in automotive Ethernet is driving the reality of deploying Ethernet in automobile.
- The industry is highly motivated by significant benefits of bandwidth, cost and weight.
- Challenges:
  - Development and Testing of Ethernet compatible ECUs and components
  - Security of a system.
  - Robust cabling to handle high electromagnetic interference (EMI).
  - Tradeoff between data error correction and effective bandwidth.

Any Questions ?



THANK YOU