Vehicular Networks

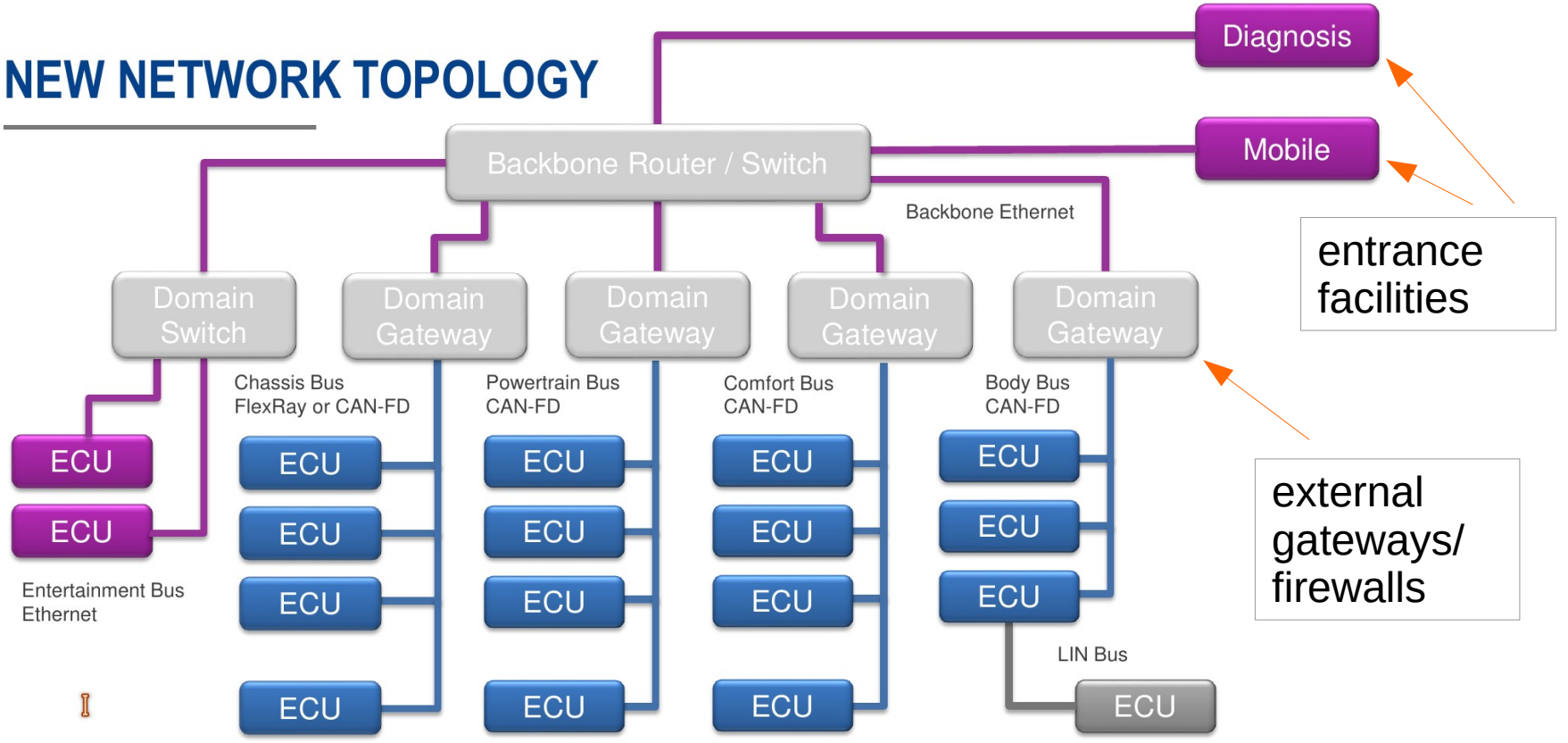
Alison Chaiken
alison@she-devel.com
Presented to the UCSC Extension course
“Networking Fundamentals"
Aug 9, 2021

Vehicular Networks

Common in automated systems and robotics:

- Precision Timing Protocol (IEEE 1588), to sync clocks of hosts
- Time-Sensitive Networking (IEEE 802.1 and 1722), to prioritize critical messages
  - CAN or Internet payload
- CV2X and DSRC wireless (not 802.11)
- Controller Area Network

Renesas Network for Modern Autos

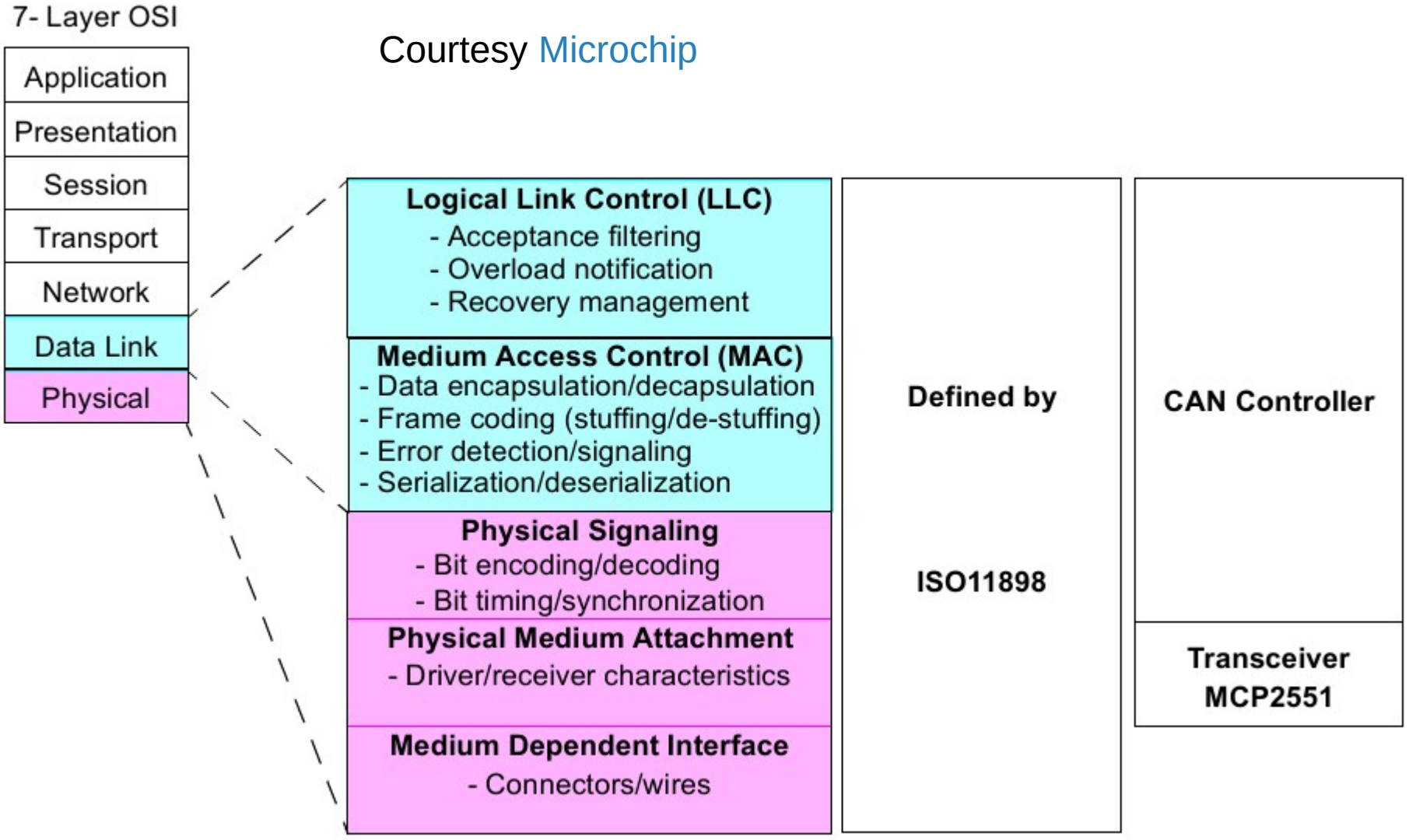


Each vehicle is an Autonomous System.

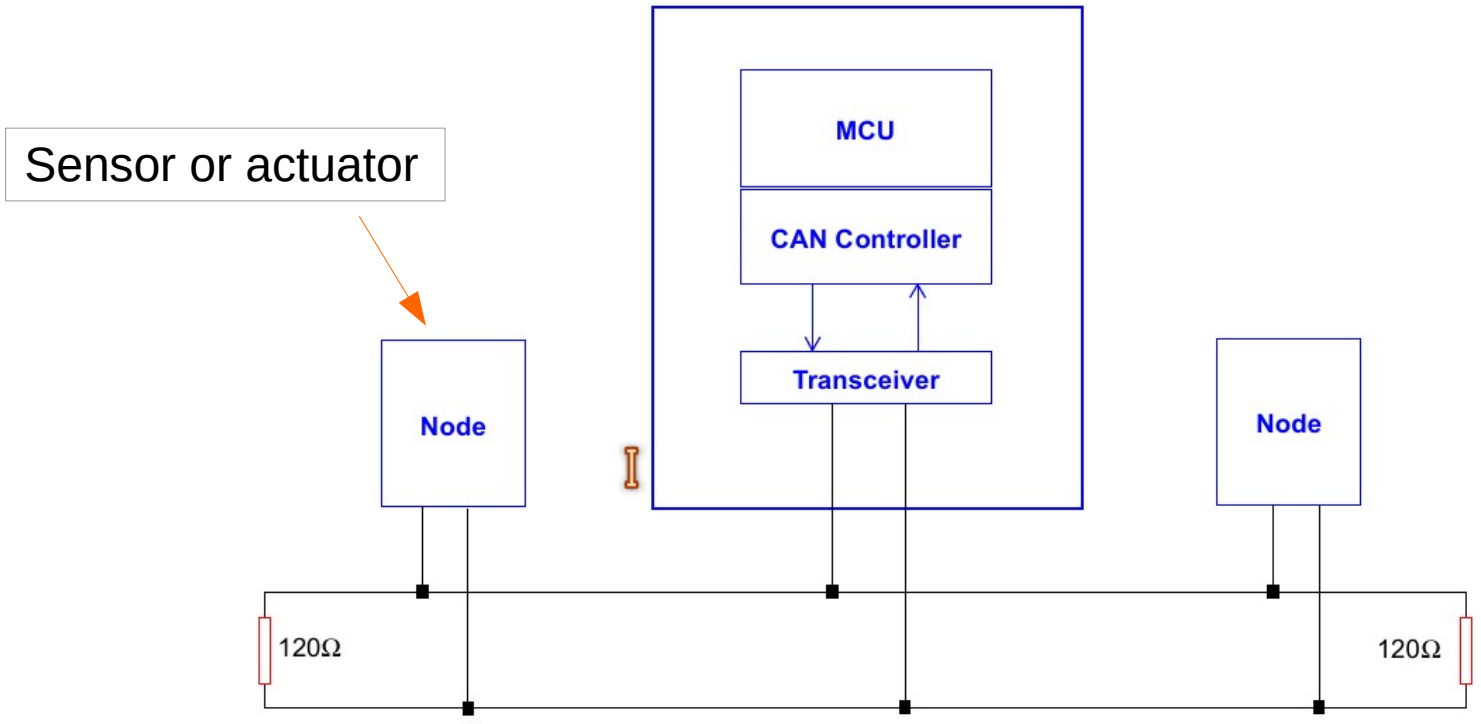
Motivation: Vehicular Subnetworks

- Different bus speeds and cabling.
- Different security domains.
  - Problem: interconnection of various domains.
  - Example: driver displays with brake and music-player info.
- CAN gateways interconnect subnetworks.

Courtesy Microchip

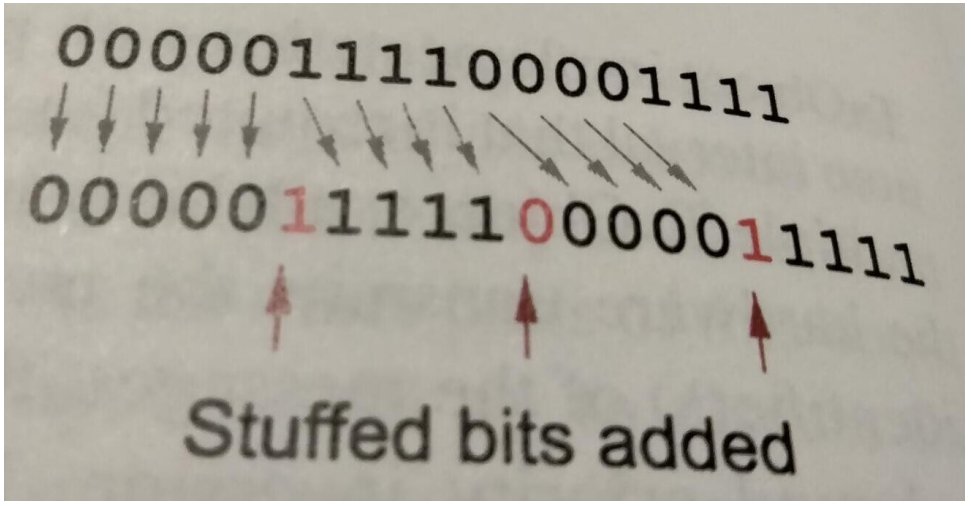


Courtesy Microchip



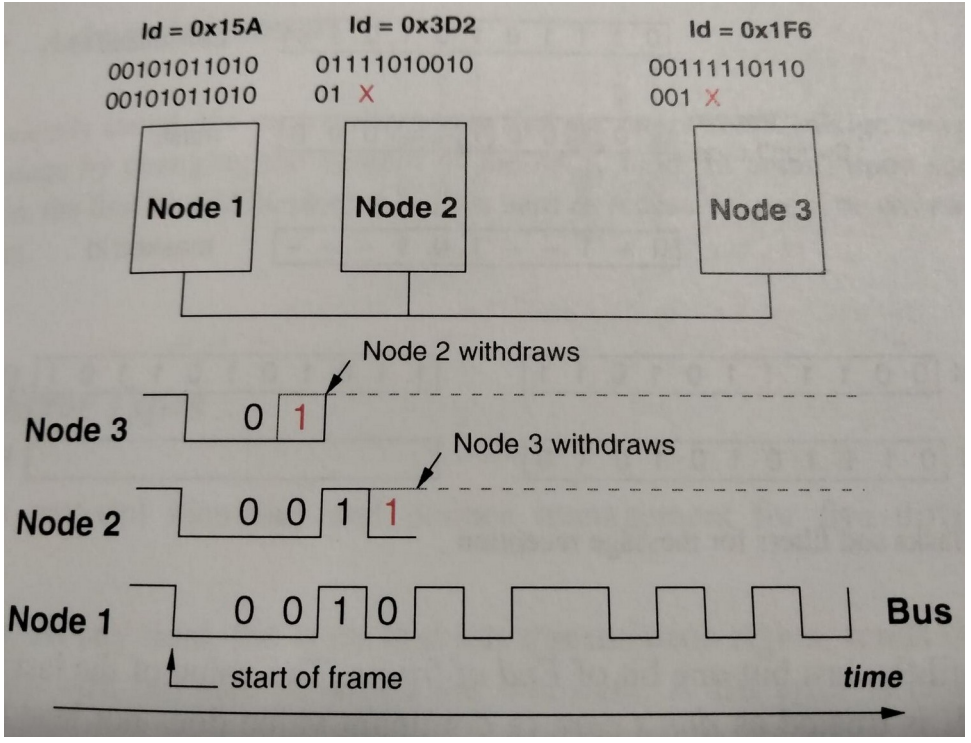
CAN Clock

Clock is derived from data → "bit stuffing."



From
Understanding and Using the Controller Area Network Communication Protocol

Bus Arbitration: CSMA/Collision Resolution



From
Understanding and Using the Controller Area Network Communication Protocol

CAN Data Frame Basics

- 11- or 29-bit ID determined by “claiming” procedure.
- Messages are acknowledged (like WiFi).
- Protocol supports Remote Request and Error frames.
- Classic CAN: 8B MTU, up to 1 Mb/s.
- CAN FD: 64B MTU, up to 5 Mb/s.
- All messages are broadcast: connectionless.
- HW filters determine which messages an ECU (electronic control unit) processes.

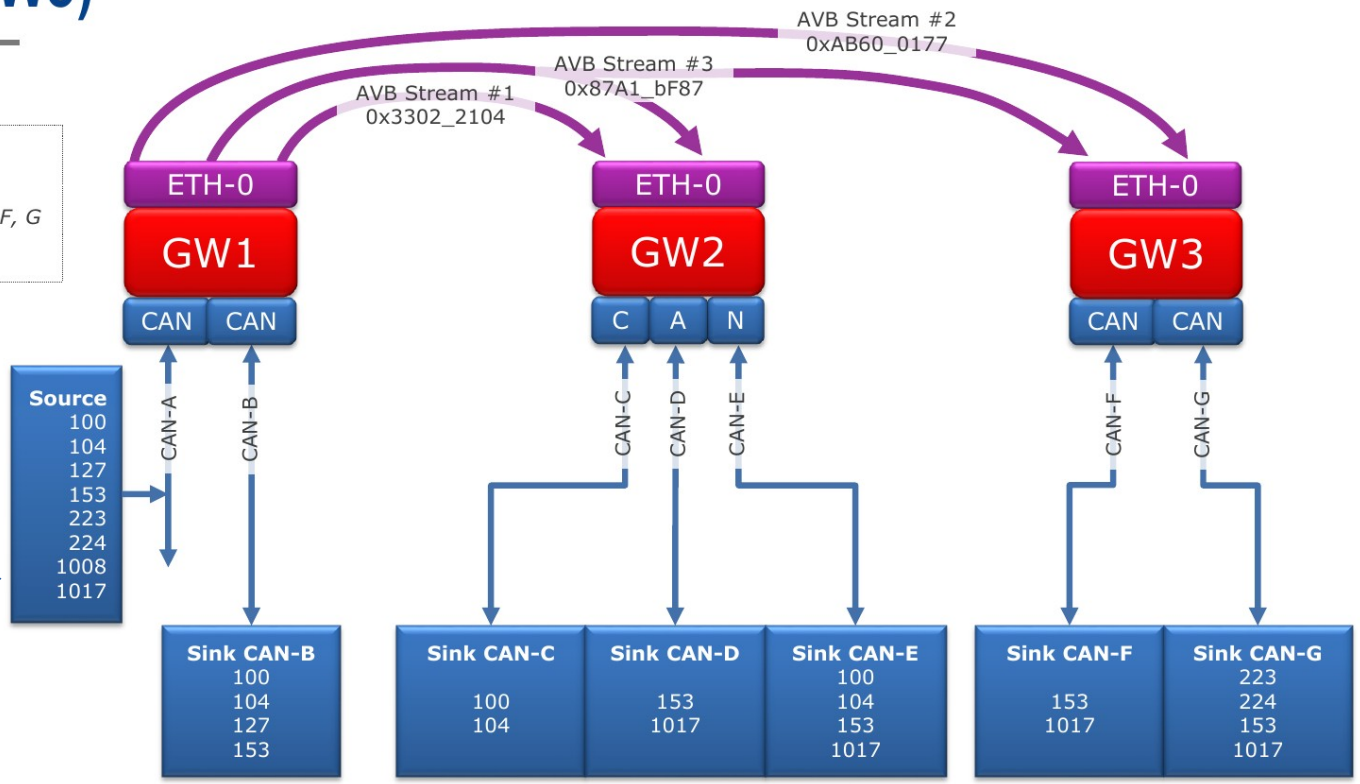
Renesas Gateway Routing: static packet filtering

(GW1, GW2, GW3)

System Definition
CAN-A 100 to 110 – **AVB#1** → *CAN-C, E*
CAN-A 200 to 2FF – **AVB#2** → *CAN-G*
CAN-A 153, 1017 – **AVB#3** → *CAN-D, E, F, G*
CAN-A 100 to 1FF → CAN-B

AVB = audio-video bridging (IEEE 1722)

CAN IDs



Summary

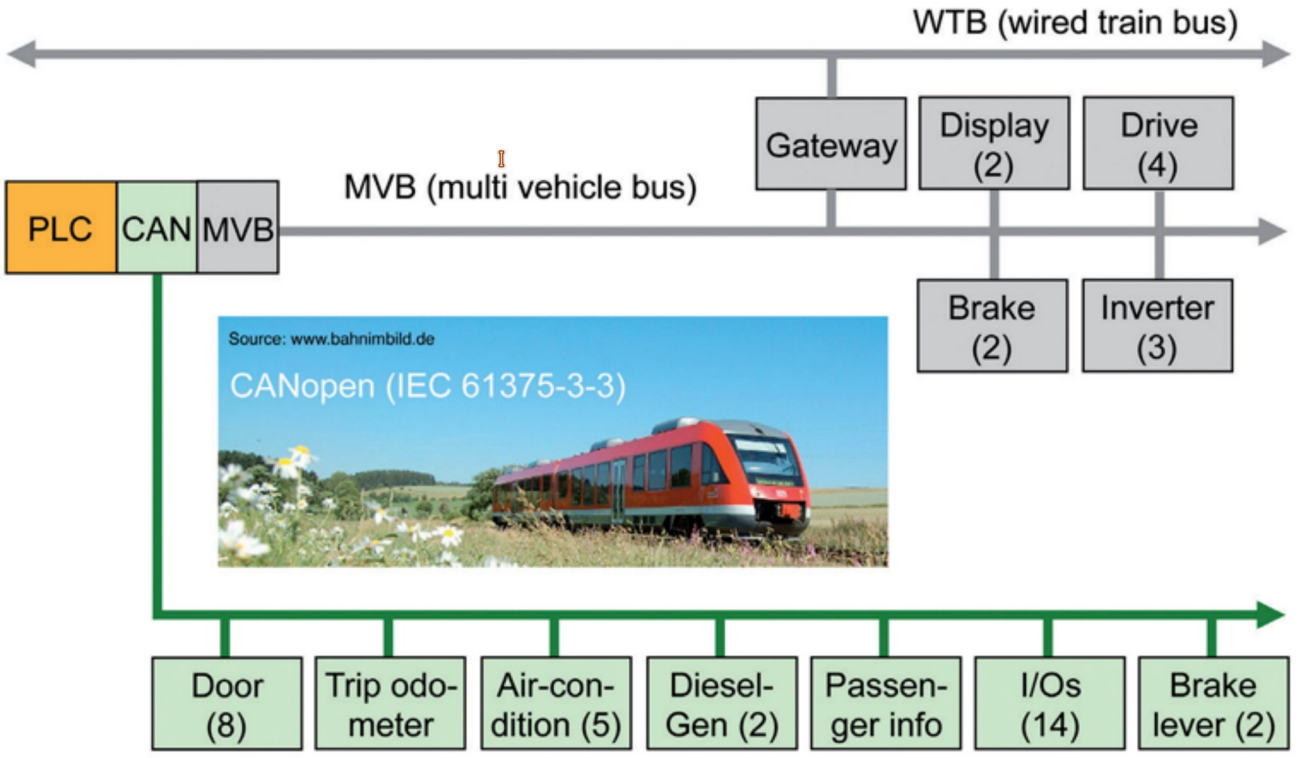
- Like Ethernet and IP, is old but evergreen.
- Differs from TCP/IP at every protocol layer.
- CAN classic and CAN FD are PHY and Link Layer protos.
- High layers have many standards and vendors.
- Wide application in vehicles, robotics, industrial applications, agriculture . . .

```
#!/bin/bash
set -e
set -u
echo "****Existing networks before VCAN addition****"
ip link show
echo "************************************************"
echo ""
# tell Linux about virtual CAN nodes: necessary one time
#sudo ip link add type vcan
sudo ip link add dev vcan0 type vcan
sudo ip link set up vcan0
echo "****Networks after VCAN addition****"
ip link show
echo "************************************************"
echo ""
```

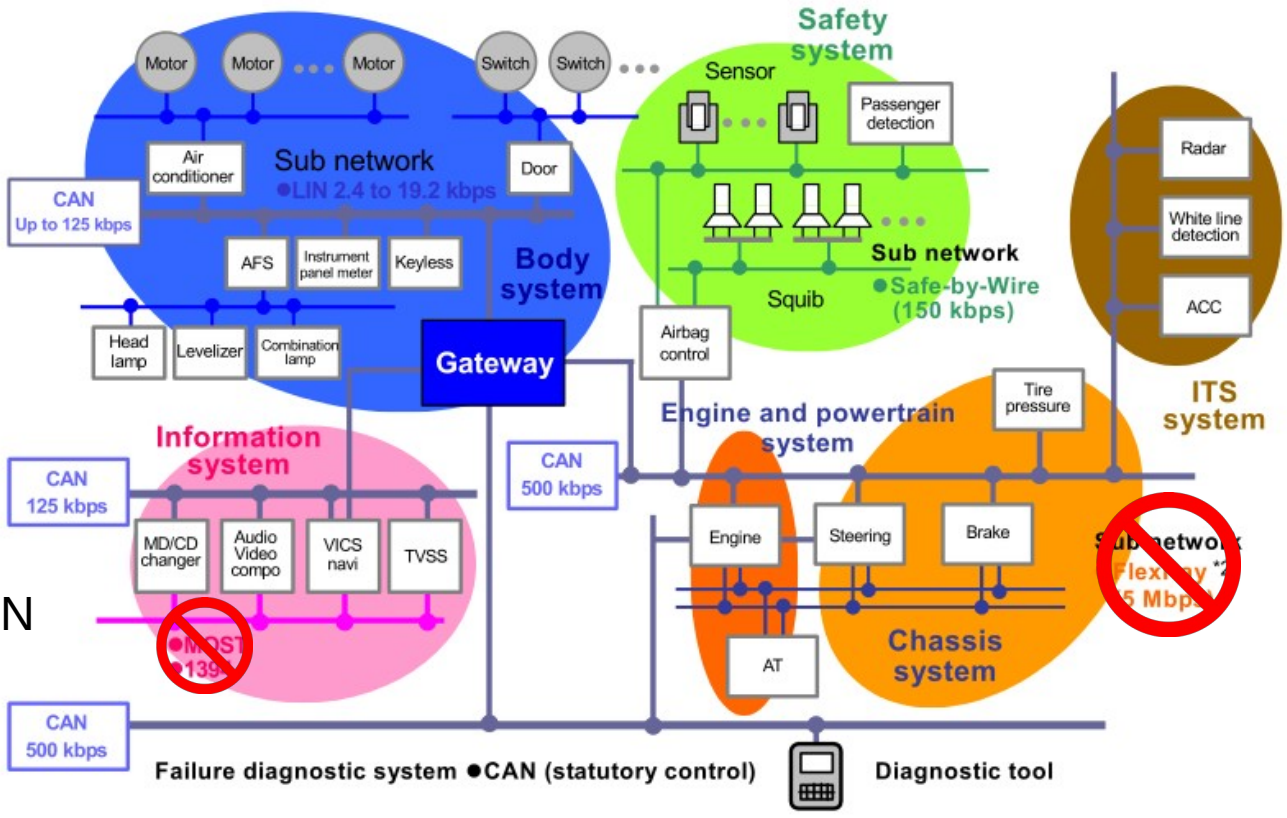
Then $ cangen -n 5 -m

```
#!/bin/bash
set -e
set -u
count=$(ip link show | grep vcan | wc -l)
echo "Removing ${count} vcan devices"
echo ""
while (( --count > -1 )); do
  echo "removing vcan${count}"
  sudo ip link del dev vcan${count}
  echo ""
done
echo "****Removed vcan devices:****"
ip addr list
echo "************************************************"
```

Example: railroad control system



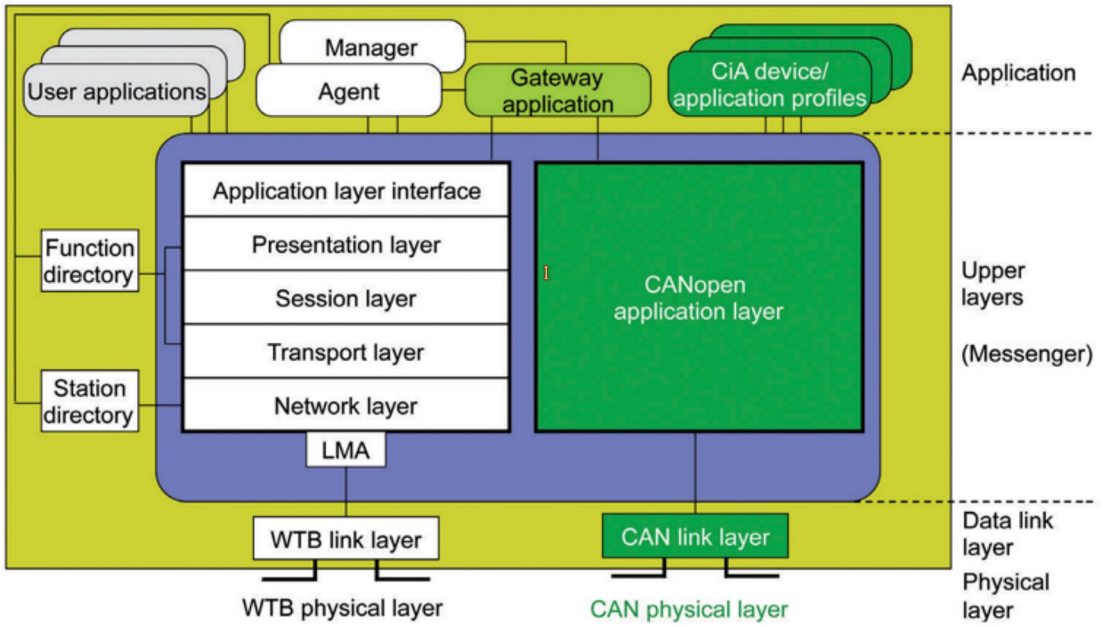
Older image of passenger vehicle network



TSN

Cameras and LIDARS: TSN

CANopen and ISO-TP provide higher levels



Example: railroad gateway