Heterogeneous Vehicular Networks and the Controller Area Network Bus

Alison Chaiken, 15 Aug 2021
Submitted to the UCSC Extension course "Networking Fundamentals"

Vehicular Network Architecture

Embedded devices such as building security controllers or motor vehicles often contain internal networks as well as hosting connections to the broader Internet. The internal networks may connect a wide variety of sensors, actuators, specialized processors, storage devices or displays, many of which are unfamiliar from general-purpose computing. The requirements of these internal networks dictate whether they employ TCP/IP like the broader Internet or implement different protocol stacks like the Controller Area Network. Primary requirements for embedded devices' internal networks are high reliability, predictability, and mechanical and thermal robustness rather than raw performance. Complex systems like motor vehicles include multiple different network types, notably Time-Sensitive Networking (IEEE 802.1AS and 1722a) as enabled by the Precision Timing Protocol (IEEE 1588) in addition to the Controller Area Network and standard TCP/IP. Vehicles may also make use of transportation-specific wireless communication devices that implement CV2X or DSRC protocols in addition to LTE.

Modern vehicles are from the network architecture point of view "autonomous systems" that include TCP/IP and TSN networks as well as CAN. From the outside, current-generation vehicles look like a roving mobile phone due to the presence of an embedded LTE modem, but that apparent outward simplicity masks considerable complexity within.

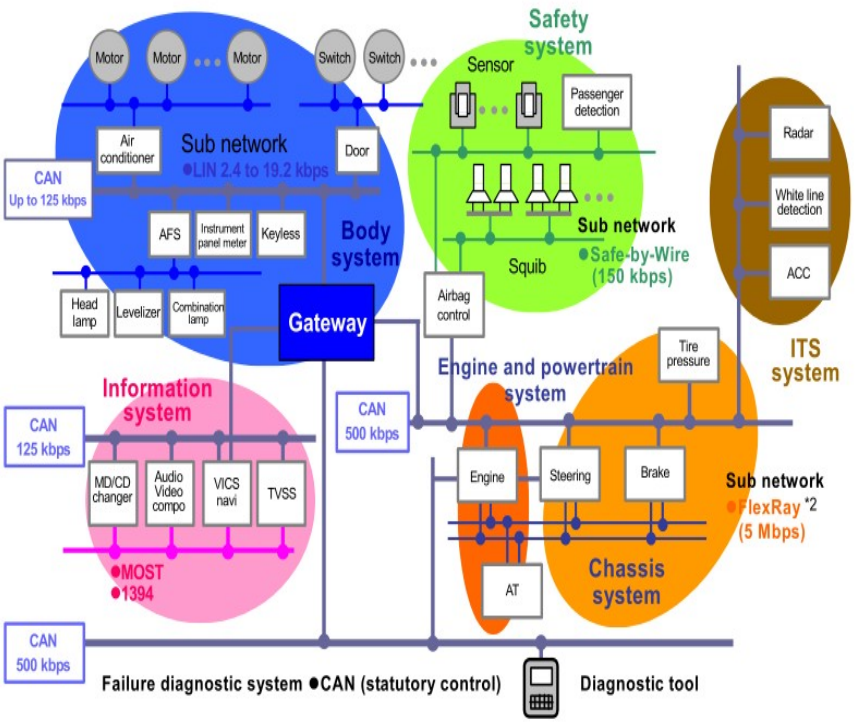


Figure 1: Network architecture typical for a current-generation high-end passenger vehicle, courtesy Renesas.

Controller Area Network Design and Protocols

The original CAN protocols were developed in the 1960's as the increasing complexity of automotive electronics made simple star-topology connection of electronic-control units (ECUs) increasingly unworkable. While CAN, like TCP/IP itself, is decades old, it has remained relevant by adopting a protocol stack architecture that has allowed it to adapt to increasingly stringent demands. A consideration of CAN from the point of view of the OSI Model therefore illuminates how it differs from TCP/IP.

Strictly speaking the CAN stack (ISO 11898) defines only the PHY and Link Layer protocols, although multiple compatible high-level protocols that are particular to CAN networks are in widespread use. The original CAN standard from 1991 dictated a simple twisted pair bus with nodes (ECUs) connected in parallel, kb/s data rates, 8-byte data length and 11-bit IDs. Later extensions supported 29-bit IDs, and the widely adopted CAN FD protocol permites 64-byte messages and up to 5 Mb/s data rates.

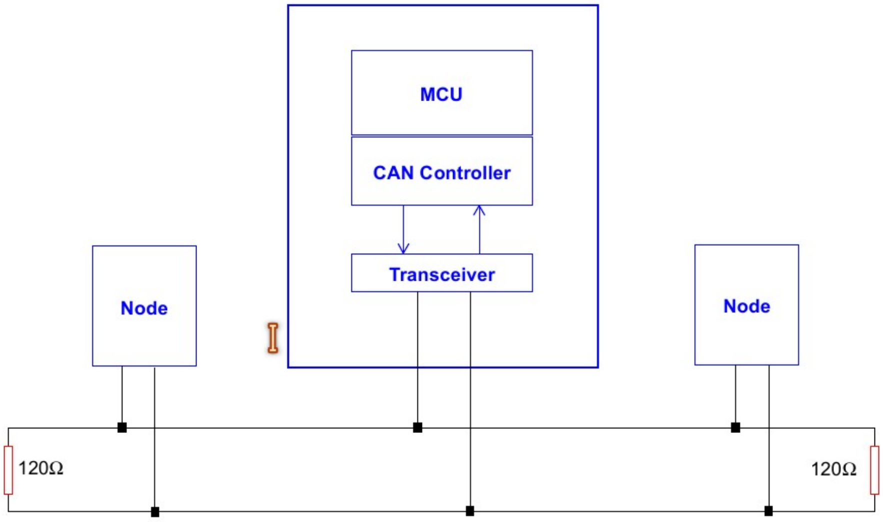


Figure 2: Topology of CAN, courtesy Microchip.

CAN is a multi-master, half-duplex serial bus. Messages are all broadcast, but hardware filters inside the transceivers of individual ECUs narrow the subset to which they listen. Each message frame begins with a unique ID that describes message content. Standard IDs and message format for vehicles are defined in standards documents, for example SAE J1939 for commercial vehicles, and SAE J1979 for OBDII emissions testing. Most ECUs also transmit frames with proprietary content whose format specification and interpretation are only available from their manufacturer. As with WiFi, CAN messages are acknowledged, which presents considerable overhead. CAN also supports "remote request" frames that query all listeners about a given message ID. If multiple listeners respond, the bus may enter an error state.

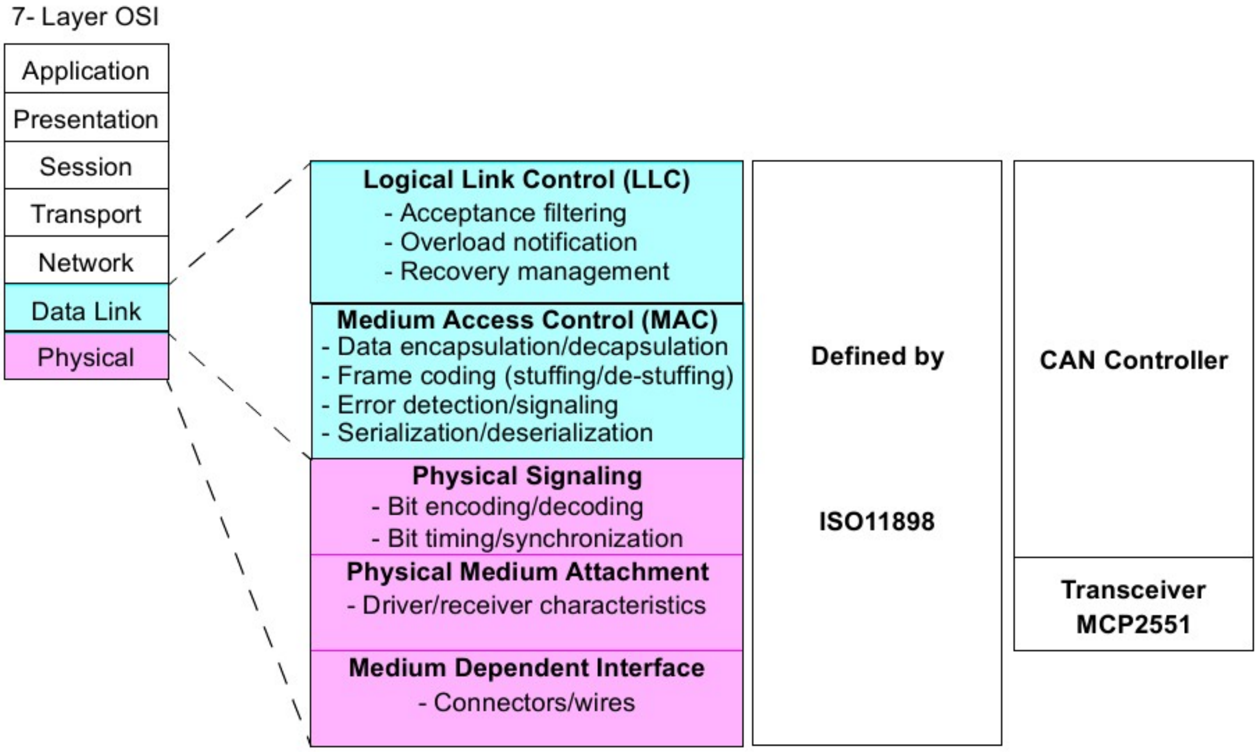


Figure 3: CAN's implementation of the PHY and LL levels of the OSI stack (courtesy Microchip).

Salient features of CAN's lower layers include clock-signal propagation via bit-stuffing and medium-access control via Collision Sensing Multiple Access with Collision Resolution, also known as "bus arbitration. "Bit stuffing" refers to a transmitting ECU's insertion of single low or high bits after a long sequence of highs or lows in order that other nodes can keep their clocks in synchrony. The CSMA/CR arbitration is based on the message ID that begins every frame, as illustrated in the Figure below. The conflicting transceivers that wish to transmit will yield the channel to other transceivers whose message ID begins with a longer string of low values. Thus 0x1 is the highest-priority message, and a sequence of 1's is the lowest.

(continued next page!)

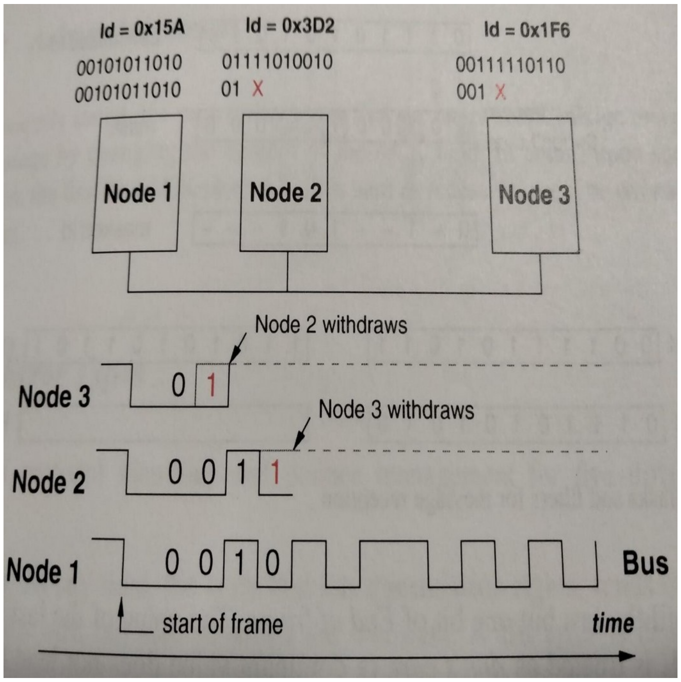
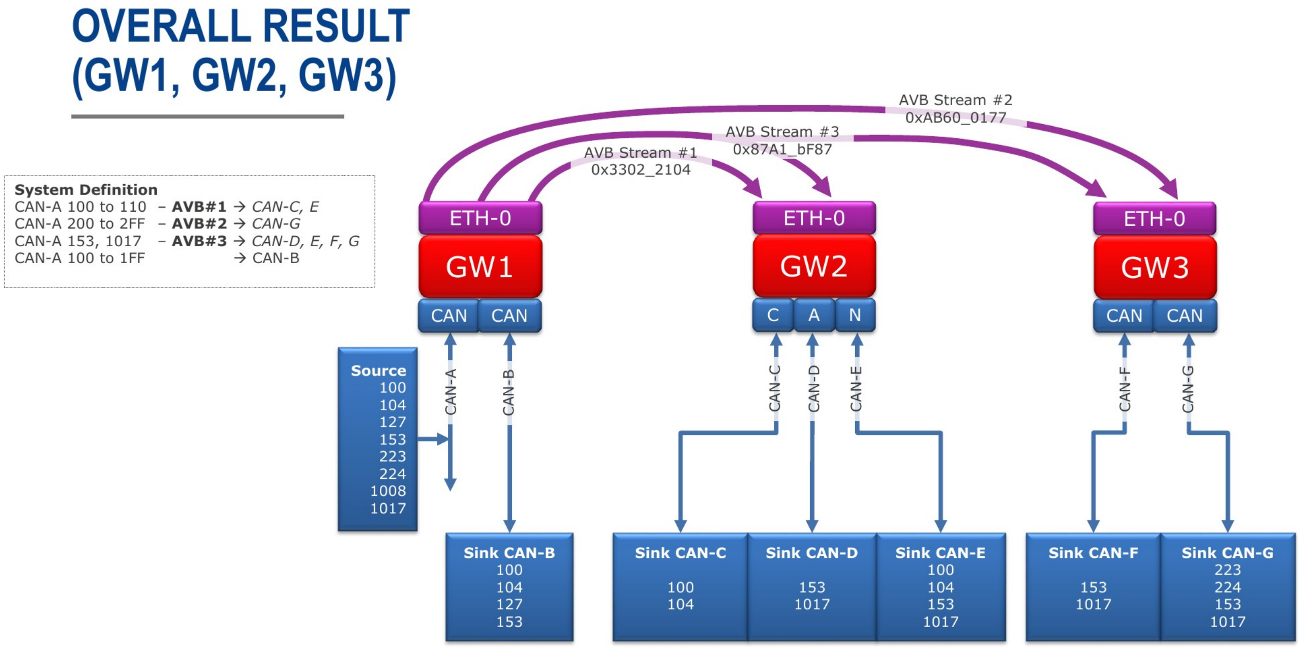


Figure 4: CAN bus message-ID-based collision resolution scheme. The message with the lowest ID has highest priority. From Understanding and Using the Controller Area Network Communication Protocol

Vehicular Subnetworks and Gateways

As the first Figure illustrates, CAN networks are typically broken out into subnets for many of the same reasons as traditional computer networks, for example differing speeds, differing security levels or capacity limitations. The presence of multiple networks can be confirmed by checking the number of populated lines in the OBDII port of a modern vehicle. Beyond the ground connection, each pair of lines indicates a distinct subnet.

Network designers utilize sophisticated simulation software that informs them if all the ECUs connected to a particular subnet will be able to their deliver high priority messages in a timely fashion. As in traditional computer networks, particular ECUs may need to communicate on multiple subnets, often with multiple interfaces, here CAN transceivers. In some cases interconnection of multiple networks of the same security level and speed indicates the need for a gateway. In the case where a dumb device needs only to read CAN messages and not to send them, an interposed firewall is appropriate. A more sophisticated network layout that is representative of the next generation of vehicles is illustrated in the Figure below. Gateways with multiple interfaces statically route messages between different subnets based on their message IDs. Gateways communicate with each other via TSN, which is referred to the in the diagram by the older name "AV/B", which stands for "Audiovideo bridging" (IEEE 1722).



OVERALL RESULT (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

Figure 5: A more modern vehicular network architecture, courtes Renesas.

Summary

The Controller Area Network was designed in response to a set of requirements that are quite different from those of traditional computer networks. The original CAN protocol is quite old, but the disaggregation of CAN implementations into a protocol stack has allowed CAN to evolve while continuing to support legacy ECUs. Next-generation vehicular designs increasingly incorporate TSN and PTP in addition to CAN. CAN networks are an integral part of many products besides vehicles, notably in military, scientific, agricultural and industrial applications.

Recommended Reference: CAN-CIA website