The IEC/IEEE Train Communication Network

Railway operators and manufacturers have standardized a data communication network that interconnects programmable equipment between and within rail vehicles. This data bus architecture offers a basis for standardization of future railways applications.

Hubert Kirrmann
ABB Corporate Research

Pierre A. Zuber
DaimlerChrysler
Rail Systems

Automatic coupling of railway vehicles has existed since the mechanical Jenny coupler at the turn of the 19th century. The railway industry's next challenge is automatic coupling of the vehicle's electronic equipment through a data bus. This requires a worldwide standardization of onboard data communication. A joint effort by the International Railways Union (Union Internationale des Chemins de Fer, or UIC), Utrecht, Netherlands, and the International Electrotechnical Committee (IEC), Geneva, Switzerland has laid the groundwork for this standardization. The UIC groups all national rail operators worldwide and ensures cross-border traffic by standardizing track profiles, pneumatic hoses, traction voltages, operating procedures, and so on. The IEC is well known to IEEE members for its impressive collection of standards in the electric world, and as the "electric sister" of the ISO.

Deputies from over 20 countries, including many European nations, the US, Japan and China representing major railways operators and manufacturers, worked several years within the IEC's Working Group 22 (WG22) on the definition of the Train Communication Network. The TCN was adopted as the international standard IEC 61375 in 1999.[1] The IEEE Rail Transit Vehicle Interface Standards Committee Working Group 1 contributed to this work in the late phase and adopted TCN as IEEE Std. 1473-1999 Type T with no modifications the same year.[2]

An international standardization of data communication is necessary at both the train and vehicle levels. Trains with varying composition during daily service—such as metros, or suburban and international trains—need a standard form of data communication for train control, diagnostics, and passenger information. Such communication should configure itself when vehicles are coupled on the track.

At the vehicle level, a standard attachment of equipment would serve manufacturers, suppliers, and operators. Manufacturers could assemble pretested units, such as doors manufactured by subcontractors, which include their own computers. Parts suppliers who interface with different manufacturers could reduce development costs by adhering to one standard. Railroad operators could reduce spare parts and simplify maintenance and part replacement.

General architecture

The TCN architecture addresses all relevant configurations found in rail vehicles. It comprises the train bus connecting the vehicles and the vehicle bus connecting the equipment

0272-1732/01/$10.00 © 2001 IEEE

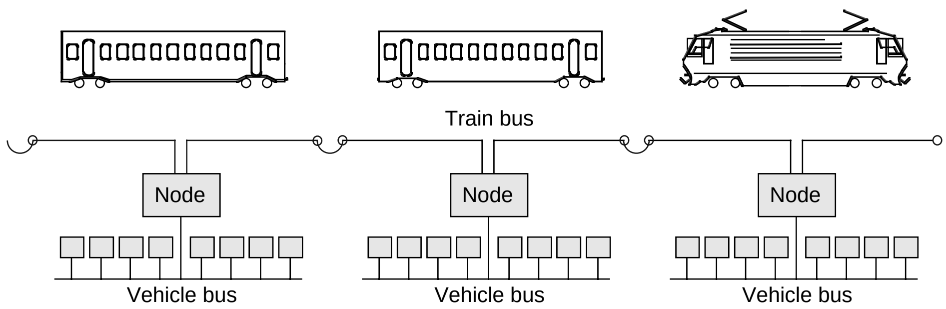

Figure 1. Train communication network.

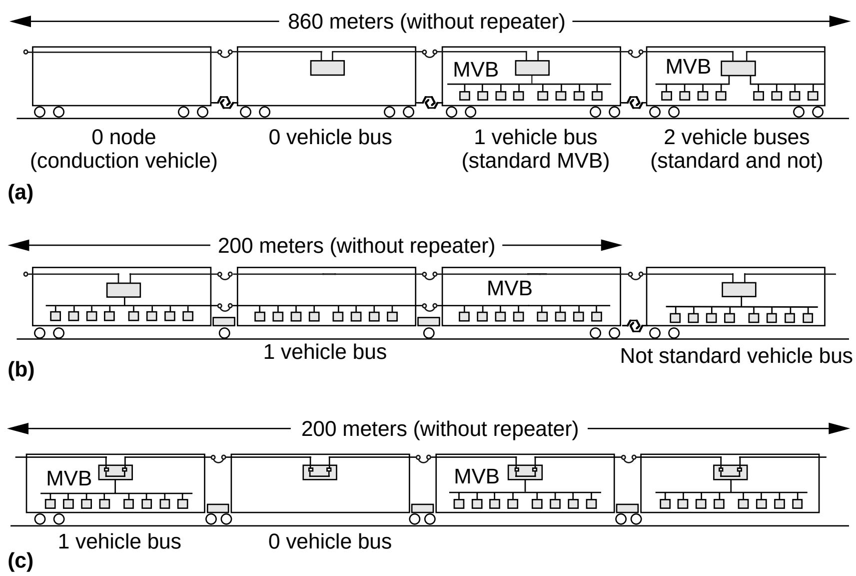

Figure 2. Open train with the Multifunction Vehicle Bus as vehicle bus (in some vehicles) and the Wire Train Bus as train bus (a); connected train sets with the Wire Train Bus as the train bus and the Multifunction Vehicle Bus interconnecting an (inseparable) vehicle set (b); closed train, such as a tilting train, with the Multifunction Vehicle Bus both as a train bus and as a vehicle bus—a nonstandard bus can also be integrated as vehicle bus (c).

aboard a vehicle or group of vehicles, as shown in Figure 1.

A vehicle may carry none, one, or several vehicle buses. The vehicle bus may span several vehicles, as in the case of mass-transit train sets (multiple units) that are not separated during daily use. In closed train sets where the train bus needs no sequential numbering of nodes, the vehicle bus may serve as a train bus, as shown in Figure 2.

Wire Train Bus

To respond to the demand for train-level standardization, WG22 specified the Wire Train Bus (WTB) as part of the TCN architecture. The WTB interconnects vehicles over hand-plug jumper cables or automatic couplers, as shown in Figure 3.

WG22 considered several media. It rejected coaxial cable because of its poor mechanical resistance to shock and vibration. Optical fiber was also dismissed because of difficulties in building automatic couplers that could withstand shock and vibration as well as harsh weather conditions. Therefore, as its name implies, the WTB uses a twisted shielded-wire pair, which has demonstrated its reliabili-

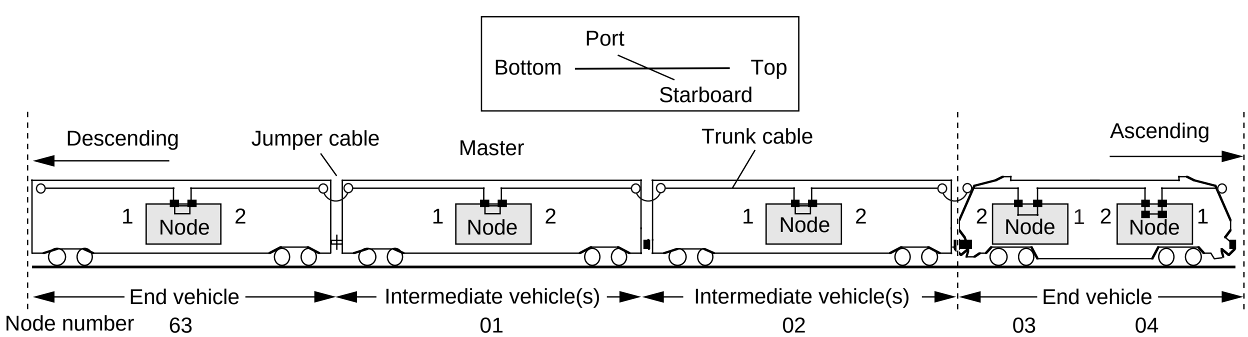

Figure 3. Wire Train Bus.

ty in several European trains. Originally the WTB shared the UIC cable with the standard wires carrying the DC signals for controlling lights, loudspeakers, and doors in international vehicles. Due to these wires limited capacity and in view of future requirements, the UIC decided to add to the UIC cable a dedicated, twisted shielded-wire pair capable of carrying data at 1 Mbps. The WTB layout is by principle redundant; one cable runs on each side of the vehicle, as shown in Figure 4.

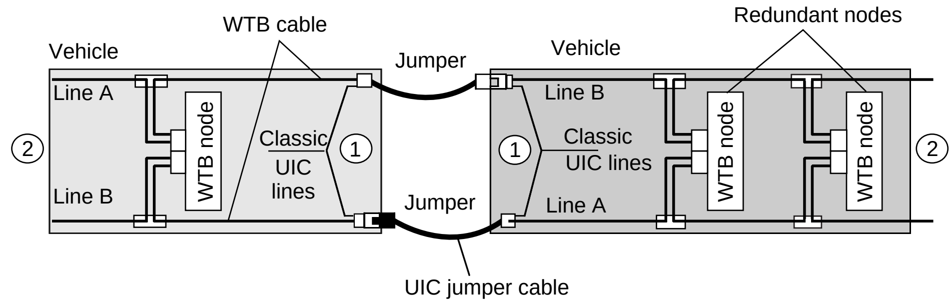

Figure 4. WTB cabling (top view).

The WTB can span 860 meters, a distance corresponding to 22 UIC vehicles, without a repeater. This requirement allows connecting of older vehicles not equipped with the new data bus onto a train. It also allows bypassing of vehicles with a low battery voltage—a major concern because of battery discharge when vehicles are in the marshaling yard. The WTB may have to operate under harsh environmental conditions where oxidation of contacts can occur. To clean oxidized connectors or contacts, a fritting voltage (cleaning action of the coupler's contact) can be superimposed on the lines.

The binary data are not transmitted over the cable as a sequence of 1s and 0s, technically known as nonreturn to zero. Instead, the bits have a Manchester encoding scheme, offering several advantages (see "Manchester encoding" sidebar).

The WTB's most salient feature (and a unique trait in the railroad industry) is that it automatically numbers nodes in sequential order and lets all nodes distinguish between the train's right and left sides and aft and fore directions. Each time the train composition changes, for example, after adding or removing vehicles, the train bus nodes execute the inauguration procedure, which connects electrically and assigns a sequential address to each node. In general, there is one node per vehicle, but, as shown in Figure 3, there may be more than one or none at all.

At the end of the inauguration, all vehicles recognize the train topography, including

- their own address, orientation (right and left), and position with respect to the bus master (aft and fore);
- other vehicles' number and position in the train;
- other vehicles' type and version (locomotive, coach, and so on) and their supported functions; and
- their own and other vehicles' dynamic properties (for example, the presence of a driver).

Manchester encoding

Manchester encoding is a robust, synchronous encoding scheme used by several buses such as Ethernet. It encodes bits in fixed time slots (cells); a "1" represented as a positive transition in the middle of a cell and a "0" as a negative transition (or the reverse). Since there is always one transition per bit, the signal clock may be easily recovered from the signal.

In its simplest form, Manchester is decoded by sensing the zero crossings of the signal. This uses inexpensive RS-485 transceivers, such as those used by the MVB. Sensitivity is increased by sampling the signal at its peaks, see Figure A. A clock synchronized to the signal by a phase-locked loop evaluates the position of the peak. To allow the phase-locked loop to adjust itself, useful data must be preceded by a preamble with a known sequence, consisting usually of alternating 0s and 1s.

In WTB, the phase-locked loop is enhanced by signal processing techniques, similar to those used in DSL.

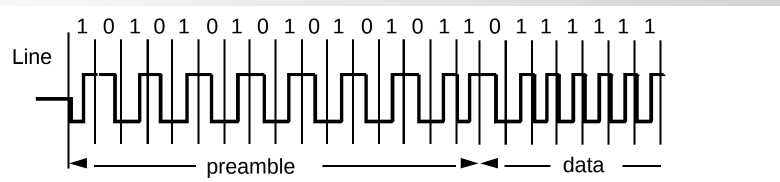

Figure A. The Manchester encoding scheme signal sequence.

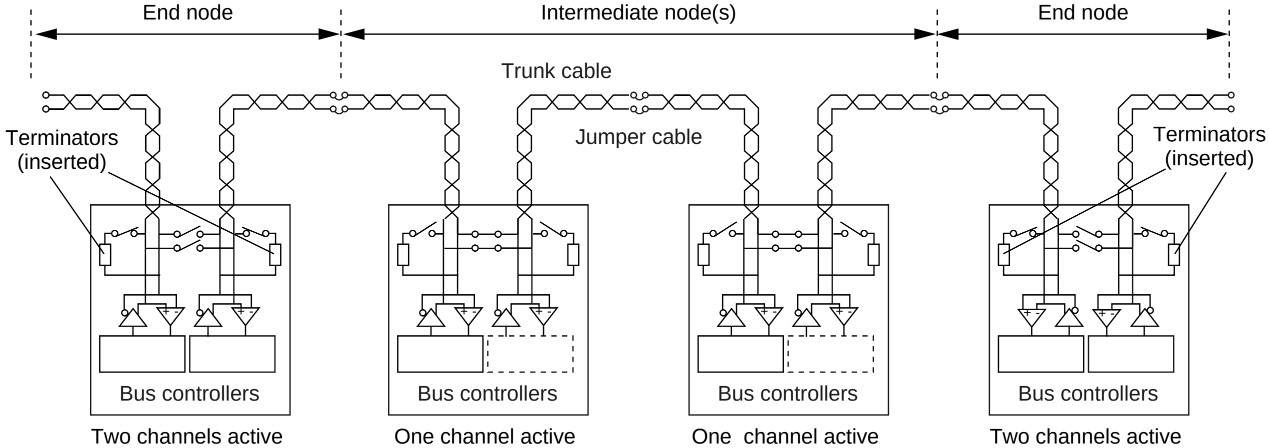

Figure 5. Detailed view of WTB.

Each node comprises two high-level data link (ISO 3309) control channels, one for each direction (forward, backward) as shown in Figure 5. During operation, the end nodes insert their termination resistors to close the bus, while the intermediate nodes establish bus continuity between the end nodes. On the end nodes, two channels are active, one for the bus traffic and one for detecting additional nodes. On the intermediate nodes, only one channel is active, the other is isolated to reduce bus load.

When a train composition consisting of *N* nodes is operating, its end nodes send a "We are *N* nodes" frame every 50 ms over the open extremity. The rest of the time they "listen" for additional nodes. When a second composition consisting of *M* nodes is coupled, the end node of the first composition detects the "We are *M* nodes," while the second composition detects the "We are *N* nodes" of the first composition. What follows depends on the respective number of nodes: If the second composition has more nodes ($M > N$), then the first composition disbands. If both compositions have the same strength, the disbanding decision is random. The winning composition integrates the nodes of the disbanded composition one by one. Each time the winning bus integrates a node, the node receives its address and becomes the next end node, while the former end node switches to an intermediate position.

The principle is simple, but inauguration is complex since it requires correct node numbering and identification in many situations. For instance, nodes may wake up from low-power sleep mode to active mode in the middle of an inaugurated composition, nodes could start operating as backup in cases where a working node fails, or one of the redundant lines might fail (only one line is shown in Figure 5) and this may not affect numbering. For a fast recovery after bus disruption, every node can become bus master. In such an event, mastership automatically transfers to a neighboring node. The dining car, for example, can become the bus master, but since all TCN traffic is slave-to-slave, it will not control the train. The worst-case recovery time is less than 1 second for 32 nodes.

Once inauguration is finished, the nodes broadcast their configuration to each other, indicating, for instance, that they represent a locomotive, a motor coach, or a driver coach. They also broadcast properties, such as the length between buffers and their weight. This requires a strict definition of the data exchanged and builds on the expertise of railway experts. The WTB data traffic and the exact meaning of each variable and each bit is standardized in UIC leaflet 556.[3]

Multifunction Vehicle Bus

To simplify assembly, commissioning, and subsystem reuse, the TCN architecture specifies the Multifunction Vehicle Bus (MVB) as a vehicle bus. The MVB connects equipment within a vehicle or within different vehicles in closed train sets. Figure 6 shows what subsystems it could connect in a locomotive. The

MVB operates at 1.5 Mbps and over the following media:

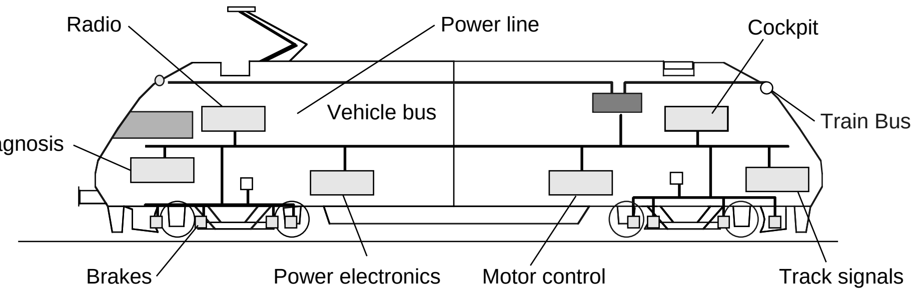

Figure 6. MVB layout in a locomotive.

- Optical fibers for distances over 200 meters and for environments sensitive to electromagnetic interference (in locomotives). MVB specifies 240-μm fibers, which are more robust against cracks and vibrations than standard telecom fibers.
- Transformer-coupled, 120-ohm twisted-wire pairs for distances of up to 200 meters to connect two or three vehicles in a train set. These specifications resemble those of IEC 61158 but use 120 ohm for robustness and low attenuation.
- RS-485/120-ohm cable for cost effective device connections within the same cabinet or on the same backplane with no galvanic separation. When galvanically separated, this cable can connect equipment in different vehicles in closed train sets.

These different media can by directly connected with repeaters, since they operate at the same speed with the same signaling.

The MVB is based on the bus pioneered on the Swiss Locomotive 460 and is used in over 600 vehicles worldwide. The MVB enables considerable reduction in the amount of cabling and increased reliability with respect to conventional wiring.

A dedicated master controls the MVB and can have backup from redundant masters to increase availability. The MVB controller provides redundancy at the physical layer: A device transmits on the redundant lines, but listens to only one while monitoring the other. Other features include high integrity against data corruption and, due to its robust Manchester encoding and checksums, fulfillment of the IEC 60870-5 FT2 class. The Hamming distance is 8 when using fiber optics.[4]

Common protocols

Despite differences at the physical and link layer, the WTB and MVB adhere to the same operating principles.

Data traffic

TCN buses transport two types of data: process variables and messages. Process variables reflect the train's state, such as speed, motor current, and operator's commands. The transfer time for process variables must be short and deterministic (see the "Determinism for time-critical data transmission" sidebar). Railways require that the train communication network guarantee less than 100 ms of delivery delay from a device on a first vehicle bus to a device on a second vehicle bus, both vehicle buses being connected by the train bus. Traction control over the vehicle bus requires guaranteed delivery from application to application for all critical variables within less than 16 ms. To guarantee these delays, the train communication network transmits all process variables periodically.

Message data carry infrequent, but possibly lengthy information, for instance, diagnostics or passenger information. Message length varies between a few bytes to several kilobytes. Messages transmission delay must be short on the average, but the application tolerates delays up to several seconds. This slackened requirement lets the TCN transmit messages on demand.

Medium access control for periodic and sporadic traffic

All buses pertaining to the TCN provide two basic medium accesses:

- periodic (for data like process variables and
- sporadic (for on-demand data traffic, such as messages).

Periodic and sporadic data traffic share the same bus, but devices treat each separately.

Determinism for time-critical data transmission

Controversy rages in the automation community between those who think that deterministic operation is required and those who think a weaker constraint is sufficient.[1]

Determinism means that the time between the detection of a change and response to that change is bound by a maximal value, even when including some fault conditions (for example, transient communication error). These systems are also called *hard real time*.

In this context, *nondeterminism* means that the system cannot provide an upper bound for its response time but will normally react fast enough for all practical purposes. Such systems are sometimes called *soft real time*.

The distinction between deterministic and nondeterministic systems is visible in a plot of the probability of response versus the response time, as shown in Figure B.

While a deterministic system responds before the deadline under all circumstances, the nondeterministic system has a nonzero probability of missing its deadline, although it usually reacts faster under normal conditions.

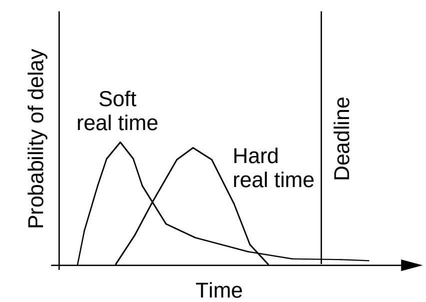

Figure B. Response delay probability for deterministic or hard real-time systems and for nondeterministic or soft real-time systems.

Determinism is a system property. Every component (whether for data acquisition, processing, transmission, or storage) must be deterministic for the whole system to be deterministic. A deterministic bus is no guarantee that the whole system will be hard real time.

Of course, no system behaves deterministically under all failure scenarios. But a nondeterministic system introduces temporal errors because of its very nature and without any external influence.

Example systems

A safety system reads an emergency stop signal and should stop the train before reaching a switch. Safety-critical systems operate in negative mode, meaning that the brake computer applies the brakes if it doesn't receive confirmation that the emergency stop is not activated. This protects against communication disruption.

The deterministic bus will transmit the no_stop signal cyclically every 0.2 s. If transmission fails, the brake computer will wait for three cycles before applying the brakes, its timer being set to 0.6 s, which leaves sufficient headway to stop the train. Emergency braking takes place in the unlikely case of three garbled transmissions in a series (an external cause).

The nondeterministic system lets the brake computer periodically ask for the emergency stop's status. If it does not receive a positive response, it applies the brakes. Although the response usually comes within 0.1 s, in some cases response time increases to 0.6 s. In this situation, the train will suffer an emergency stop because of data packet corruption or network congestion—neither are external causes. Increasing the time-out does not change this situation, but requires longer rails and headway.

Both systems are safe, but availability—how often the train stops because of false alarms—becomes the issue.

Achieving determinism

Deterministic systems reserve system resources before operation, which prevents resource contention.

Communication systems usually achieve determinism by cyclic operation, using time division multiple access under either a master-controlled, token-passing, or clocked operation. All TCN buses are deterministic, a philosophy shared by the field buses, such as IEC Std. 61158.

Systems can also achieve determinism in processing by enforcing time-bounded tasks and cyclic operation. Examples of such systems include industrial programmable-logic controllers programmed in IEC Std. 61131 function block language.

Nondeterministic systems have no fixed preallocation of resources. Examples include collision-based medium access buses, such as Ethernet; databases with semaphore access; and preemptive operating systems, such as Unix or Windows.

The controversy

This controversy over the need for determinism is reflected in everyday traffic. For example, commuters accept delays in the morning rush hours as a price for using a nondeterministic, event-driven transportation: their car.

Conversely, commuters expect scheduled public transportation to behave deterministically. If delays occur, commuters expect the operator to tell them about an external cause for the delay, such as heavy snowfall, but too much traffic will never be an issue (airlines excepted). The amount of reserve time that commuters plan on to reach a destination will influence their decision on the type of transportation, with a clear advantage to the scheduled public transportation with tight time constraints and heavy traffic. Estimating delays for the nondeterministic automobile transportation requires an analysis of the situation, such as listening to traffic news.

Reference

1. P. Koopman, "Tracking down Lost Messages and System Failures," *Embedded Systems Programming*, Oct. 1996.

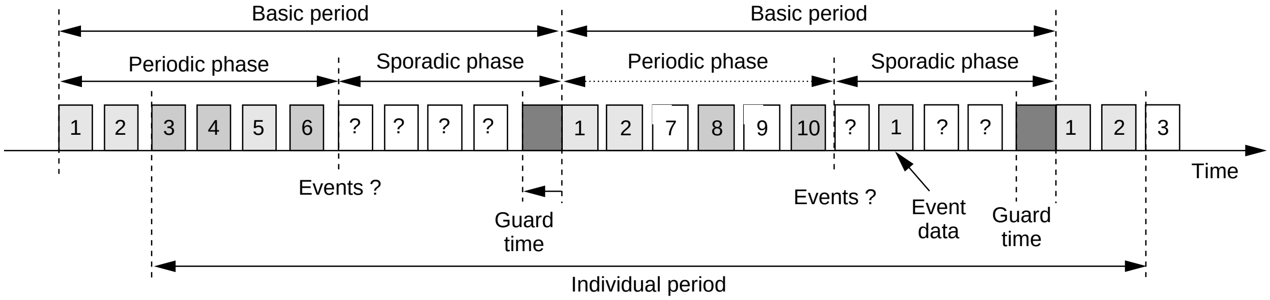

Figure 7. Alternating periodic and sporadic data transmissions lets a single bus transmit both types of data. Process variables are transmitted at regular intervals (1 ms) and after transmission, the bus checks for sporadic traffic demand and transmits, if requested, a message packet, if the guard time is respected.

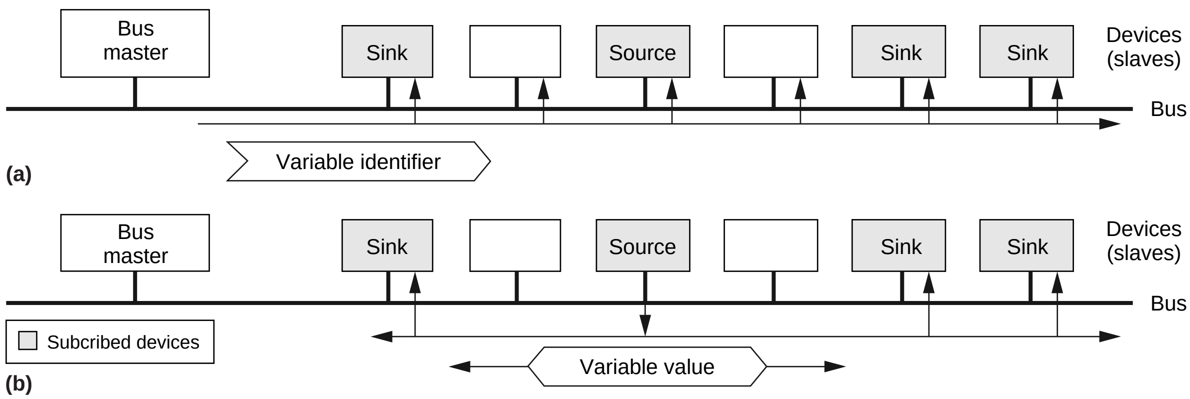

Figure 8. Source-addressed broadcast. In the first phase (a), the master broadcasts a short, master frame with the identifier of a variable, taken from its poll list. In the second phase (b), the source device sends the variable's value in a slave frame, all devices interested in that value receive it. The master is normally neither source not sink of the variables.

One device acting as master controls periodic and sporadic data transmission, which guarantees deterministic medium access. To accomplish this, the master alternates periodic and sporadic phases, as shown in Figure 7.

Traffic is divided into basic periods of fixed duration—either 1 or 2 ms on the MVB and 25 ms on the WTB. At the start of a period, the master polls the process variables in sequence during a certain time period—the periodic phase. To reduce traffic, urgent data are transmitted every period and less urgent variables are transmitted with an individual period every second, forth, eight, and so on basic period, with the longest period being 1,024 ms.

After transmitting the process variables, the bus master checks for sporadic data to transmit. On the WTB, a flag in the periodic data signals that a node has sporadic data pending. On the MVB, an arbitration procedure ensures that one of several devices gets serviced. If there are no sporadic data to transmit, the sporadic phase remains unused. If there are data, the master checks that sufficient time remains until the start of the next period (it respects the guard time), and if so, invites a device to transmit its sporadic data. A highly precise start of the next period is needed because the first master frame of a period serves to synchronize all clocks with a jitter of some microseconds.

Process variable transmission

In the first phase of process variable transmission, the master broadcasts a frame to trigger transmission of a certain variable without specifying the source device. In a second phase, the source device answers by broadcasting a frame containing the requested value to all devices. Each device interested in this variable picks up the value, as shown in Figure 8.

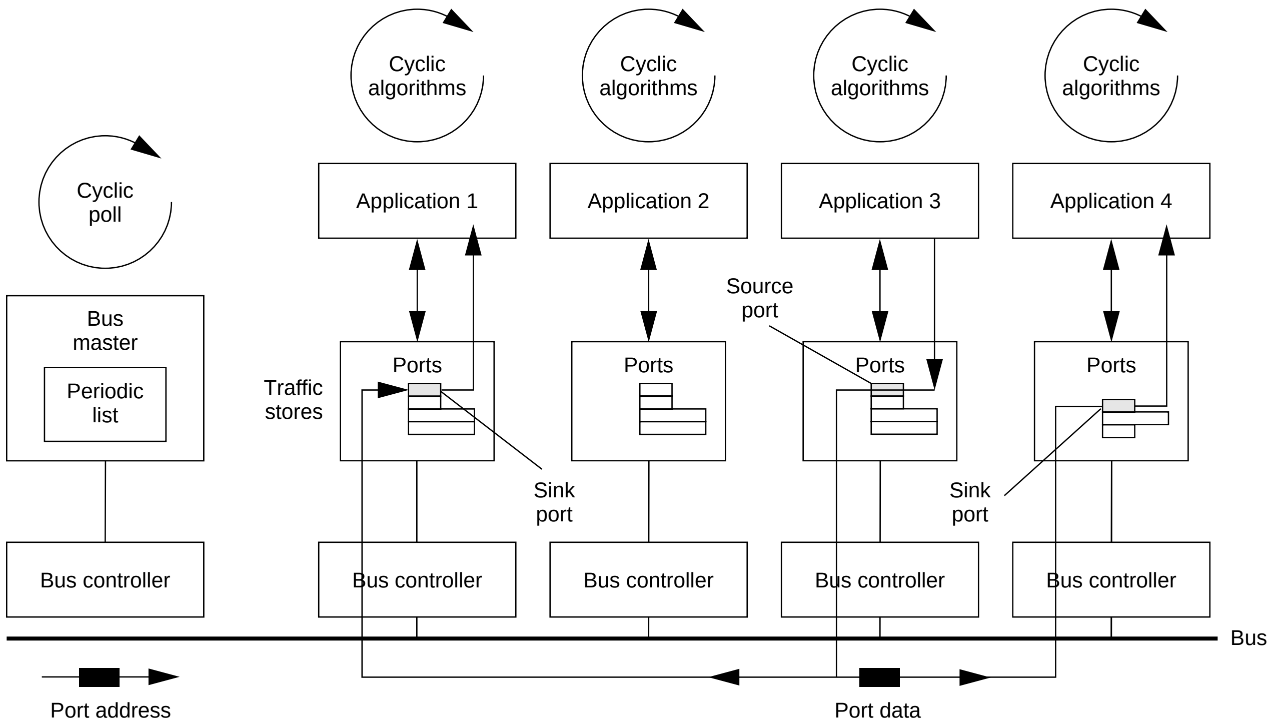

Figure 9. Broadcast of process variables.

To increase efficiency, slave frames carry numerous variables having the same period, called a *data set*. A data set contains values and check bits, but no addresses. Each variable is identified by its offset from the data set's beginning, each set is identified by the master frame. To maintain determinism, the configuration tools define the frame format and the poll lists before operation starts, after this, the traffic pattern cannot change. On the MVB, each device can subscribe (as either a source or sink) to up to 4,096 data sets. On the WTB, each node has only one data set to broadcast, but it can receive up to 32 data sets from other nodes.

The sources or sink data sets with the values of the variables are stored in a shared memory, called the *traffic store*. The application processor and the bus controller can simultaneously access the traffic store on a device. The traffic stores implement conjointly a distributed database, as shown in Figure 9, which the bus keeps synchronized. For the application programmers, the bus behaves like a shared memory.

Source-addressed broadcast lets applications and the bus operate independently. The application processor is only interrupted on reception or transmission for time synchronization. End-to-end determinism is ensured by the periodic nature of the application processes and the bus.

Since the master periodically requests the transmission of process variables, there is no need for an explicit retransmission after an occasional loss. To cope with persistent faults, the bus controller maintains a counter for each variable, indicating how long ago the bus refreshed the variable. In addition, the application can transmit a check variable for each variable to certify the variable's timely and correct production.

The application accesses process variables either individually or (more efficiently) by clusters. The process data application layer marshals transmitted data to the individual application variables. It also converts data types to the representation used by the consumer. The gateway between the WTB and MVB copies variables from one bus to the other and synchronizes the cycles. The gateway can also combine variables, for example, it can build a compound variable indicating that all doors are closed in its vehicle.

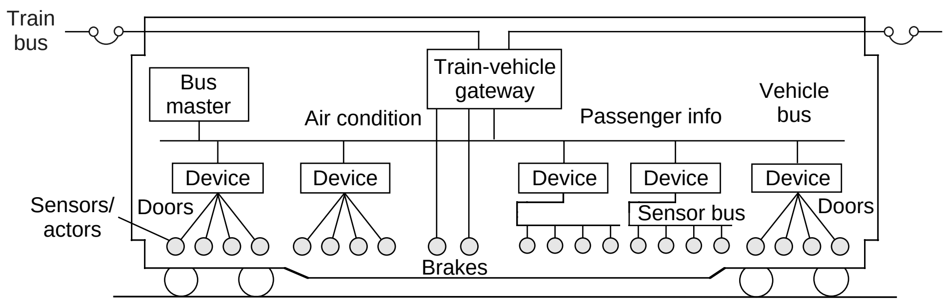

Figure 10. Map of logically addressed application functions that are typical in a railway car.

Message transfer

Applications exchange messages transparently over the TCN. An application cannot determine whether its peer resides on the same bus, the same station, or anywhere else on the network.

To cope with a variety of vehicles and equipment, the TCN uses logical addressees for messages. Every node of the train bus supports several application functions, as shown in the map in Figure 10.

From an outside node on the WTB, the internal organization of a vehicle is not detectable, it seems as if the train bus node executes all the functions. One or more devices or the train bus node can execute the application functions. A device might execute several functions, or different devices may execute a function. The same principle applies to functions communicating over the vehicle bus—the application need not recognize where the other function resides.

Applications communicate on a client-server basis. A conversation consists of a call sent by the client and a reply sent by the remote server. The network retains no memory of a conversation once transmission is successful or timed out. This is more efficient in terms of memory and timers than TCP-like stream-oriented protocols and suits the dominant diagnostics traffic.

The communication layer divides these call or reply messages into small packets for transmission. Each packet carries a full address, which identifies its source and destination. The train bus nodes route the packets using a function directory that indicates which device is executing which function. This function directory is static. A dynamic actualization would have been more analogous to plug-and-play, but would have caused major fault-recovery delay. A classical sliding window retransmission protocol implements flow control and error recovery. Only end devices execute this transport protocol; intermediate nodes only intervene in exceptional cases (during inauguration, for instance).

Network management

Network management helps configure, commission, and maintain the TCN. A network manager can connect to the TCN, for instance, as a vehicle device. The network manager has access to all devices—in any vehicle—connected to the TCN

The network manager can inspect and modify other devices through an agent (an application task running in each station). The agent has local access to managed objects such as process variables, protocols, memory, tasks, and clocks. The standard specifies the management services to read and write the managed objects, as well as the format of network manager messages.

Conformance testing

Interoperation will only succeed if manufacturers can validate that devices conform to the TCN specifications. Conformance testing guidelines let manufacturers test their products against the standard. In particular, this requirement applies to WTB nodes, which must operate without adjustment when vehicles of any origin are coupled. The MVB has similar requirements when it comes to plug-in interchangeability.

To address conformance testing, WG22 developed a set of guidelines. This is only the first step toward a full program of conformance testing that an independent agency, such as the

European Railways Research Institute, would perform. The IEC set up Working Group 34 to develop a full suite of conformance tests as a second part of the standard.

State of the work

Standardization has prompted numerous railway manufacturers to support TCN-compliant product development. Applications include signaling, radio communication, and Web access to rolling stock.

Development

A joint development project by a group of manufacturers—Adtranz, Siemens, and Firema-Ercole Marelli—supports the TCN and has helped to demonstrate its technical capabilities.

The joint development project members combined forces to develop a complete TCN with all the necessary hardware and software. This group intends to make the components available to any interested party for their own implementations under reasonable conditions.

ERRI test train

Although the working group derived the WTB and MVB from existing, railways-proven solutions, important modifications made in response to user requirements demanded a complete test of the TCN. The UIC, through the European Railways Research Institute (ERRI), sponsored a full-scale TCN implementation from May 1994 to September 1995. They tested the TCN on a special test installation in the lab and on an existing track.

The study used test train equipped by different manufacturers (Adtranz, Siemens, Firema, and Holec) and with coaches from Italy, Switzerland, Germany, and the Netherlands. The ERRI put this train into revenue service between Interlaken, Switzerland, and Amsterdam, Netherlands.

This test validated the interoperation of a mixed system and confirmed the standard documents' completeness. The valuable experience gained on this train improved the standard, especially in relation to its impact on existing systems and on exploitation issues (for example, the necessity for personnel to verify that the two cables are plugged).

Standardization

Although the technical standard work was nearly complete after the ERRI test train, it took four more years to meet the quality requirements for a standard. This long delay is not uncommon in standards work: While the original documents tend to focus on technical aspects, the final documents focus on interface aspects to ensure that standard-compliant devices are interoperable. This approach differs from the current tendency to base standards on product specifications, allowing different variants, profiles, and incompatible options. For instance, the conformance test lists several device properties that must exist to bear the name of IEC Std. 61375. These properties range from the connector to the type of messages that the device must send and receive. The result is that the IEC passed this standard in 1999 with nearly unanimous approval.

The IEEE Rail Transit Vehicle Interface Standards Committee adopted the TCN as IEEE Std. 1473 for onboard data communication. Here, the focus of standardization was defining which applications the TCN covers, assuming that other bus types aboard the same vehicle exist. The IEC shares this focus, stating that the WTB and MVB should be used in situations requiring interoperability and interchangeability. The IEC does not want to force manufacturers to use TCN where optimized solutions already exist.

Eurocab project

A full-size test rig in Brussels demonstrated the TCN's application to safety tasks—in particular, automatic train operation. Here, signaling equipment from different manufacturers with different safety philosophies interoperated on a simulated train. This test took place within the Eurocab project, and was part of the larger European Train Management System. The network's deterministic nature let the safety analysis focus on hardware failures and disturbances.[5]

ROSIN project

A common communication protocol is necessary but not sufficient to ensure interoperation. Applications must also have standardized ways to exchange data, so that applications can access equipment and subassemblies regardless of manufacturer.

To address this need, the European Union

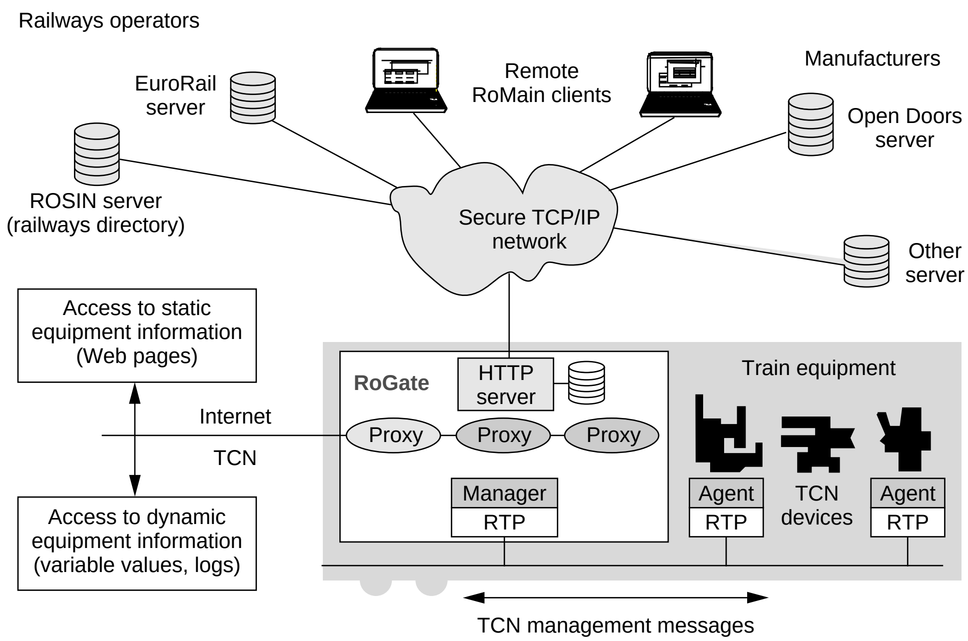

Figure 11: RoMain architecture for Web access to moving trains over radio links.

set up the three-year ROSIN (Railways Open System Interconnection Network) project.[6] About 20 different firms collaborated on this project to define device profiles for different applications such as

- passenger trains with locomotives,
- freight trains,
- mass transit,
- equipment interfaces (for propulsion, brake, doors, air conditioning, and so on),
- radio links, and
- signaling.

ROSIN defined the exchanged data down to the individual bit level. Standardized data representation definitions exist, but it would have been unrealistic to force all existing equipment to switch to a new data-encoding scheme, ignoring the installed base. Therefore, project members defined a notation (ROSIN notation or retrofit notation) that can describe arbitrary bit fields. For instance, say that certain equipment exports a variable representing the vehicle speed; ROSIN's preferred measurement is in SI units, representing meter/second as a 32-bit real number. A manufacturer may however specify a different representation in its device description file, such as speed_mph as a 16-bit integer from 0 to 200 miles per hour in little-endian format. By looking at this specification, the user of this device knows how to map this variable to other devices.

The ROSIN project concluded with a demonstration of Web access to the vehicles, called RoMain (ROSIN Maintenance). The group equipped a local commuter train between France and Spain with a radio link and a Web server. This demonstrated that a PC-based Web server could understand the data traffic on the MVB just by reading the equipment description files.

The demonstration was impressive—users could inspect vehicle data while the train was running with a standard browser from anywhere in the world, via the system architecture shown in Figure 11. The main challenge was database management. Indeed, because of the radio links' limited bandwidth, the devices' static information isn't located on the devices themselves but on the railway operator's Web server. This arrangement arose from the fear that mergers and sales among device manufacturers would rapidly make the Web links obsolete. This makes updating handbooks and maintenance manuals easy, but requires a rather high administrative effort.

US initiatives

After the IEEE Rail Transit Vehicle Interface Standard Committee accepted the IEEE Std. 1473, work continued in the IEC Working Group 9 to define the equipment interface. This working group considered the experience of the American Public Transportation Association and the work of the Transit Communication Interface Profile project. In parallel, Working Group 01 is developing an open TCN stack as a clean room implementation. The first WTB-equipped train in the US should be the New Jersey's Comet 5 train.

The IEC/UIC standardization of the TCN ensures a good base for actual and future developments. The number of TCN equipped vehicles is growing rapidly. All new projects by Adtranz, Firema, Siemens, and several other manufacturers are TCN based.

Railways are now specifying TCN conformance in their public bids. The standardization of application functions is an indispensable further step to achieve plug-in interchangeability of equipment and vehicles.

The TCN technology has spread outside of the railways community. It is used in high-voltage substation control and in printing machines, where real-time constraints are as demanding as in railways. MICRO

References

1. *Train Communication Network*, IEC 61375, International Electrotechnical Committee, Geneva, 1999; http://www.iec-tcn.org.
2. IEEE Std. 1473-1999, *IEEE Standard for Communications Protocol Aboard Trains*, IEEE, Piscataway, N.J., 1999.
3. *Information Transfer in Trains*, UIC B 108.3, leaflet 556, 11th ed., Union Internationale des Chemins de Fer [International Railways Union], Utrecht, Netherlands, 1996.
4. P. Koopman and T. Chakravarty, *Analysis of the Train Communication Network Protocol Error Detection Capabilities*, ECE Department and ICES, Carnegie Mellon University, Pittsburgh, Pa.
5. B. Eschermann et al., *Fail-Safe On-Board Data Bus for Automatic Train Protection*, Railtech, Birmingham, England, 1994; http://www.cordis.lu/transport/src/eurosig.htm.
6. *Telematics Applications for Transport Research Information*, 4th Framework Research Projects ROSIN TR 1045 Railway Open System Interconnection Network, European Union, Brussels, Belgium, 1999; www.labs.it/rosin.

Hubert Kirrmann is a senior scientist at ABB Corporate Research, Baden, Switzerland. His research interests include industrial automation computer buses and fault-tolerant systems. Kirrmann has a PhD from the Swiss Institute of Technology, Zurich. He is a member of the IEEE and of several interest groups.

Pierre A. Zuber is a fellow engineer at DaimlerChrysler Rail Systems in Pittsburgh, Pa. He participates in research, development and design of industrial control for automatic train control and train communication equipment. Zuber has a BSEE in automated control from the Geneva School of Engineering, Switzerland. He holds several patents and is a member of the IEEE.

Direct questions and comments about this article to Hubert Kirrmann, ABB Corporate Research, CH 5405 Baden, Switzerland; hubert.kirrmann@ch.abb.com.