

Time-triggered exchange of train route data between train control systems

M. Montigel

Department of Computer Science, University of New Orleans, USA.

Abstract

The increased heterogeneity and decentralisation of train control systems, often requiring the collaboration of several suppliers, calls for a simplification of the involved interfaces. Time-triggered protocols form a favourable alternative to the event-triggered protocols, which are still used in most vital computer-based applications involving data exchange. The advantages of time-triggered protocols include an increased robustness against transmission failures and a simplification of the underlying definitions.

To take full advantage of time-triggered protocols, the used telegrams should be of fixed length whenever possible. This poses a problem in transmitting state information of ambiguous train routes, i.e., if there exist several such routes between a given pair of start and end point. To overcome this problem, an algorithm to derive a unique route index is introduced. The proposed approach relies exclusively on geographical track layout data, without the exchange of explicit route tables between collaborating suppliers. As a means to represent track topology data and to introduce the route indexing algorithm, Double Vertex Graphs are used, which – in contrast to ordinary graphs – allow for a correct representation of the topological properties of railway tracks.

The indexing scheme proposed here has been successfully applied in the pilot project of an ERTMS Level 2 system currently being commissioned by Swiss Federal Railways.

1 Introduction

In the recent past, the architectures of computer-based interlockings and other Train Control Systems (TCS) have moved from centralised solutions – provided by one supplier – to heterogeneous, distributed systems supplied by several manufacturers. As a result, *well-defined interfaces* for the data exchange between such systems have to be specified, in order to achieve the correct functionality and the necessary level

to be specified, in order to achieve the correct functionality and the necessary level of safety. Usually, the Safety Integrity Level 4 for vital electronic systems, as defined in the CENELEC norm [1], is applied.

In the following considerations, two systems involved in a data exchange are referred to as the *sender* and the *receiver* respectively, depending on the role they currently play in a data exchange. The *relevant state* denotes the state information contained in the sender which is relevant for the receiver. Consequently, the relevant state is subject to be transmitted to the receiver in a timely manner, as defined in the specifications.

To define protocols for the data exchange between a sender and a receiver over an interface, mainly two basic mechanisms are used: the *event-triggered* and the *time-triggered* approach. In the former, a data transmission is triggered by an *event*, i.e., by an occurrence of a change in the relevant state of the sender. In the latter concept, the sender transmits its full relevant state *periodically* to the receiver, i.e., the transmission is triggered by *time*, not by events.

While in many current computer-based interlockings and other TCS event-based protocols are applied for the data exchange, time-triggered protocols have more favourable properties concerning the overall system stability and the simplicity of the protocol definitions. The latter argument gains importance if several different suppliers are involved in the system development, because experiences show that complicated protocol definitions are likely to lead to misunderstandings, system integration problems and delays in commissioning.

To take full advantage of the beneficial properties of time-triggered protocols, *fixed-length* telegrams should be used whenever possible. In this article, a successful application of a time-triggered protocol is discussed. The protocol definition includes a novel indexing scheme of train routes, allowing for the representation of a route's state with a fixed length, even in the ambiguous case in which several routes between a given pair of start and end point exist.

The remainder of this article is organised as follows:

As a basic mathematical means to introduce the proposed indexing scheme of train routes, Double Vertex Graphs are employed, which have considerable advantages over ordinary graphs in describing track layout data. Since this method may not be commonly known, a brief introduction is given in *Section 2*.

In *Section 3*, the properties of event-triggered and time-triggered protocols in railway applications are compared, leading to the conclusion that fixed-length telegrams should be used in time-triggered protocols. This contradicts to the usual way of denoting ambiguous train routes by means of points lists, whose length depends on a particular train route.

In *Section 4*, an indexing scheme for train routes is presented which preserves the fixed length telegrams. The ambiguous case in which several routes between a given pair of start and end point exist is solved by a unique train route index. As a further simplification, the presented approach allows each involved party to determine the train route indexes uniquely by means of an algorithm, which is purely based on basic track layout data. As a result, the tedious and error-prone exchange of route tables between involved parties can be avoided.

The *Conclusion* provides a review of the achieved results. Furthermore, future research in protocols for train control systems should be directed towards a much higher level of standardisation, which could involve the definition of standardised, XML-based protocols.

2 Double Vertex Graphs for the representation of track layouts

As a basic mathematical means to introduce the proposed indexing scheme for ambiguous train routes, Double Vertex Graphs are used (see [2] for a complete definition). Too often, ordinary graphs (see, e.g., Wilson's introduction [3]) are used to represent track topologies, in spite of the fact that they are not well suited for this purpose. In particular, they fail to accurately represent the special topological characteristics of railway points, by introducing physically impossible paths. Such a path would require a train to turn back on a trailing points (see [2], pp. 360–361 for details). In Double Vertex Graphs this problem does not exist. Let

$$G := \langle V, E \rangle \quad (1)$$

be an ordinary graph consisting of the set of vertices V and the set of edges E . Now, a Double Vertex Graph

$$H := \langle G, \circ \rangle \quad (2)$$

consists of an ordinary graph, G , and a bijective mapping of the set of vertices onto itself, \circ , called the *joining mapping*. This mapping serves to form pairs of vertices, i.e., no vertex may exist alone, but each one has exactly one partner. Two vertices forming a pair are called *joined vertices*.

The edges of a Double vertex Graph serve to represent pieces of track of a given track topology, while the pairs of vertices denote the "connectors" between these pieces.

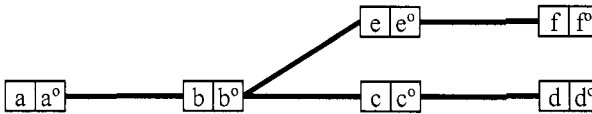


Figure 1: Example of a Double Vertex Graph containing a points

In *Figure 1*, an example of a Double Vertex Graph is shown, consisting of a points and adjacent track sections. In this case, we have

$$V = \{a, b, c, d, e, a^\circ, b^\circ, c^\circ, d^\circ, e^\circ\} \quad (3)$$

and

$$E = \{ \langle a^\circ, b \rangle, \langle b^\circ, c \rangle, \langle b^\circ, e \rangle, \langle c^\circ, d \rangle, \langle e^\circ, f \rangle \} \quad (4)$$

Examples of joined vertex pairs are: a and a° , or e and e° .

The most important definition, however, is the definition of *paths* in Double Vertex Graphs. A path defines how trains are allowed to move along the elements of a track topology, observing the path rule stated below:

Path rule: A path always starts at a vertex. Before using an edge connected to that vertex, however, a train has to move *first to the joined vertex*. It may then use a connected edge, ending up at another vertex. Here, the same rule applies: first, move to the joined vertex.

Paths are denoted as sequences of vertices, starting with the first, ending with the last, but leaving out all joined vertices (since they are implicitly clear). For instance,

$$\langle a, b, e, f \rangle, \langle d^\circ, c^\circ, b^\circ \rangle \quad (5)$$

are two legal paths in the Double Vertex Graph shown in *Figure 1*, while

$$\langle d, c, b, a \rangle, \langle c^\circ, b^\circ, e \rangle \quad (6)$$

are two illegal paths, because they violate the path rule stated above. In the latter case, a train would illegally turn back on a trailing points. In an ordinary graph, such a movement could not be prevented, because in the mathematical definition, there is no notion of an "angle" between edges, as it is suggested when the graph is drawn.

The stated path rule imposes a natural notion of *direction* on each vertex. For instance, again using the example shown in *Figure 1*, a train starting at vertex *b* is forced to move to the right, while a train starting at *b*[°] will need to proceed to the left.

In fact, it turns out that Double Vertex Graphs are a very appropriate means to represent track topologies. For instance, Hürlimann uses Double Vertex Graphs as the basic data structure in his train traffic simulation tool OpenTrack [4]. Double Vertex Graphs will be used as a basis to introduce the proposed indexing scheme route for ambiguous train routes.

3 Event-triggered protocols in vital train control systems

In many computer-based interlockings and other TCS, event-based protocols are applied for the data exchange. However, this protocol type involves two serious drawbacks, caused by interruptions of the transmission link and floods of events respectively. In this section, these drawbacks are examined and compared to the considerably more favourable properties of time-triggered protocols.

3.1 Event-triggered vs. time-triggered transmission protocols

If an interruption of the transmission link in an event-triggered protocol exceeds the tolerated time span, one or several – possibly vital – events may be lost. Hence, the protocol has to be restarted from scratch, leading to the necessity of retransmitting the full relevant state from the sender to the receiver, which may delay the resumption of normal operations. As a second consequence, the protocol definitions must contain a start-up sequence – which covers the initial transmission of the relevant state – and the regular, event-based transmission protocol, thereby leading to an increased complexity of the resulting definitions.

As a second drawback, a large number of relevant state changes simultaneously occurring in the sender result in a flood of events to be transmitted to the receiver. Such cases are normally triggered by the instantaneous failure of several subcomponents in the sender. In the worst case, this flood may exceed the capacity of the transmission system, leading to a violation of timeliness requirements, i.e., the receiver may not be able to process some events in time. In most cases the result will be a safety reaction, which, again, leads to a disruption of normal operation. Since it is hard to anticipate the maximum number of events generated in the sender at any given time point, the dimensioning of a transmission system's required capacity can be a very difficult task.

In time-triggered protocols, on the other hand, the full relevant state is periodically transmitted from the sender to the receiver. Consequently, this type of protocol is considerably more robust towards interruption of the transmission link, since the receiver is updated with the full state information with the *first* arriving telegram after the interruption, without the need to restart the protocol. This applies to the start-up phase of both systems as well. Therefore, a special start-up phase of the protocol becomes superfluous, thereby even simplifying the protocol definitions.

Secondly, a time-triggered protocol can tolerate any number of instantaneous changes of the sender's relevant state, because – based on its principle – it is dimensioned to transmit the full state for any case.

These observations lead to the conclusion, that it is appropriate to use time-triggered protocols for the communication between vital train control systems.

3.2 Time-triggered protocols and train-route data

An sample application of a time-triggered protocol is the data exchange interface between Alcatel's computer-based interlocking ELEKTRA and Bombardier's Radio Block Centre (RBC). These systems participate in an ERTMS Level 2 system [4] without lineside signals in Swiss Federal Railways' pilot project Zofingen-Sursee (see Montigel & Scheck [6] and Montigel [7]).

The protocol definition of the interface between the two systems is based on a time-triggered protocol, leading to the favourable simplifications and stability properties discussed in *Section 3.1*. To maximise the data exchange frequency, thereby further increasing the stability of the system, the amount of transmitted data has to be minimised. Secondly, the length of the exchanged telegrams should be constant, if possible, because this simplifies the dimensioning of the transmission system's capacity.

In the discussed application, the relevant state to be transmitted between the sender (an interlocking) and receiver (an RBC) consists of train route data only. If each telegram representing the state of a train route has a fixed length, the required capacity c [bits/sec] of the transmission system can easily be stated as

$$c = f(h + nr) \quad (7)$$

where f [1/sec] denotes the required frequency of the periodic transmission of the relevant state, h [bits] the size of a telegram header, n the maximum number of concurrently active train routes and r [bits] the size of a route state representation.

Remark: For the following discussion, it is assumed that the geographical principle is used for the representation of track-, route- and signal-related data, rather than the route table principle. Therefore, a globally available table itemising all train routes – as, e.g., used in British signalling philosophy – is not assumed to exist. Contrarily, it is assumed that all route-related data can be derived from the data representation of the track elements, their neighbours, the signals, etc.

Under this assumption, a pair of $\langle start, end \rangle$ can serve to denote a particular train route in a telegram. As *start* and *end*, signal or track element identifiers can be used. However, a problem arises, if several routes are possible between *start* and *end* in the given track layout. In this ambiguous case, an additional means is necessary to render the route specification unique.

Common approaches to solve this problem consist of including either a list of the positions of all facing points contained in the route, or some artificial route index consistently known to both involved systems. Yet, neither of these approaches is entirely satisfactory for the purposes of time-triggered protocols: The first contradicts the desired fixed-length property, while the second violates the geographical principle, which interdicts such artificial route indexes, since routes are not supposed to be enumerated.

As a more satisfactory solution for this problem, a natural indexing scheme for train routes is presented in the next section, allowing for the transmission of unambiguous fixed-length telegrams, which can be interpreted by the receiving system without any exchange of a route table.

4 Indexing scheme for ambiguous train routes

Below, an indexing scheme for ambiguous train routes is introduced. The main objective of this scheme is to uniquely denote a particular train route if for a given pair of start and end point several routes exist, but without

- using points position lists of variable length
- exchanging explicit route tables containing lists of routes between suppliers.

In the proposed approach, a sender system creates a unique index for each route by applying a coding algorithm, whose input is the data of the given track topology. The receiver system is able to determine the denoted train route, given its index, using the same algorithm and its own representation of the same track topology. For the procedure discussed herein a patent is pending [8].

For the representation of track topologies, Double Vertex Graphs as introduced in *Section 2* are used. As an example the following track topology is considered:

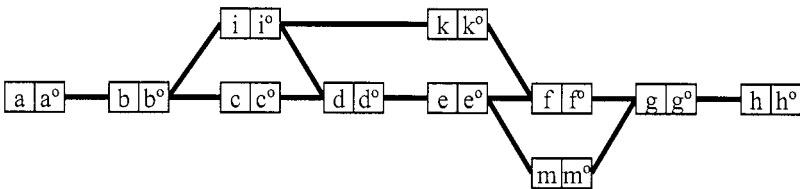


Figure 2: Example of Track Topology with ambiguous train routes

Assuming vertex *a* as the start point and *h* as the end point, the 5 train routes shown in *Table 1*, left column, exist in this track topology. In the table, they are represented as paths of the corresponding Double Vertex Graph. The train routes are shown in random order.

Table 1. Determination of position strings of facing points

Route	Position string of facing points	Derived index
<a, b, i, d, e, m, g, h>	LRR	?
<a, b, c, d, e, f, g, h>	RL	?
<a, b, i, k, f, g, h>	LL	?
<a, b, c, d, e, m, g, h>	RR	?
<a, b, i, d, e, f, g, h>	LRL	?

The first step to derive the envisaged unique index consists of determining the so called *position string* for each route: This string is composed of the letters L and R. Starting with the first vertex of the route, each vertex is examined as follows:

- If the *joined* vertex of the current vertex *does not* denote the root of a facing points, no character is added to the position string
- If the *joined* vertex of vertex *does* denote the root of a facing points, the character L is added, if the points is used in its left position, the character R otherwise

For instance, taking the first route in *Table 1* as an example, *a* does not contribute to the position string, *b* contributes an L, *i* an R, *d* nothing, *e* an R and the rest of the vertices nothing.

Likewise, the position strings of all routes are calculated.

The next step consists of sorting the table alphabetically in ascending order of the position strings, as shown in *Table 2*.

As a last step, the numbers 1 to n are inserted in the Index column in ascending order, resulting in the unique indexing scheme of the so far ambiguous train routes: Now, a triple <*start,end,index*> denotes a train route uniquely.

Table 2. Derivation of route index based sorted position strings

Route	Position string of facing points	Derived index
<a, b, i, k, f, g, h>	LL	1
<a, b, i, d, e, f, g, h>	LRL	2
<a, b, i, d, e, m, g, h>	LRR	3
<a, b, c, d, e, f, g, h>	RL	4
<a, b, c, d, e, m, g, h>	RR	5

It has to be pointed out that the determination of the index requires exclusively

- the general agreement on the algorithm
- track element data

No additional agreements are needed to ensure a consistent interpretation of the route index in all systems participating in an exchange of train route data.

5 Conclusion

It has been shown, how ambiguous train routes can be indexed in a unique manner, without any agreements on explicit route tables. This index proves beneficial in time-triggered protocols for the data exchange of train route state information, because it allows for route telegrams of fixed length.

Consequently, such train-route telegrams fit well into the time-triggered approach, thereby simplifying the protocol specifications, increasing the robustness of the data exchange and decreasing the time to reinitialise the participating systems in case of serious failures of other nature.

However, the definition of interfaces between systems provided by different suppliers still involves a significant effort because of a complete lack of standardisations of the underlying protocols. Secondly, more often than not, the transmitted data consist of binary data, which require specialised tools to debug in case of inconsistencies or failures.

In recent developments in the standardisation of data exchanges between computer-based systems, the eXtended Mark-up Language (XML) [9] has gained considerable importance. Currently, several refined XML versions for specialised domains are being defined, e.g., the Geographic Mark-up Language (GML, [10]). Yet, this standardisations come at a price: All XML-based protocols are character-oriented and involve a quite extensive parsing process on the receiver's side, while binary data – which allow for a much more compact data representation – tend to map directly into the data structures of most modern programming languages.

In spite of these apparent obstacles, the investigation of the applicability of XML-based protocols in vital railway applications seems to be an interesting research topic. On one hand, the definition of a Railway Mark-up Language (RML) would create an enormous potential for effort reductions in protocol definitions, in the development of basic components of transmission systems and the accompanying debugging tools. On the other hand, the trade-offs concerning the parsing issue and the less efficient use of the capacity of transmission systems will have to be considered very carefully.

References

- [1] CENELEC Norm ENV 50129, Railway Applications, Safety Related Electronic Systems for Signaling, Mai 1998.
- [2] Montigel, M., Formal Representatio of Track Topologies by Double Vertex Graphs. *Proceedings of Computers in Railways*, pp. 359–370, 1992.
- [3] Wilson, R.J., *Introduction to Graph Theory*, Academic Press, New York, 1979.
- [4] Hürlimann, D., *Objektorientierte Modellierung von Infrastrukturelementen und Betriebsvorgaengen im Eisenbahnewsen*, Dissertation Swiss Federal Institute of Technology, Zurich, 2001.
- [5] UNISIG Consortium, European Rail Traffic Management System. <http://www.ertms.com/>.
- [6] Montigel, M. & Scheck, O., Interface between Interlocking and Radio Block Center for an ERTMS System without Lineside Signals. *Proceedings of Aspect'99*: London, 1999.

- [7] Montigel, M., Die Schnittstelle des elektronischen Stellwerks ELEKTRA zur Führerstandssignalisierung (FSS) der SBB am Beispiel der Pilotstrecke Olten – Luzern. *Signal+Draht*, 12/1998, pp. 8-10, 1998.
- [8] Montigel, M., *Verfahren zur rechnergestützten Kennzeichnung von Fahrstrassen, Verfahren zur Übertragung von sich auf Fahrstrassen beziehende Daten und Kommunikationssystemen hierfür*, German Patent DE 197 50 465 A1, 1997.
- [9] World Wide Web Consortium, *Extensible Markup Language (XML) 1.0* (Second Edition), <http://www.w3.org/XML>.
- [10] Open GIS Consortium, *The Geographic Markup Language (GML)*, Recommendation 2.0, <http://www.opengis.org/techno/specs.htm>.