RESEARCH ARTICLE OPEN ACCESS

Location identification and Route identification in MANET

P.Latha
Department of Mathematics
AMK Technological Polytechnic College, Sembarambakkam. India

Abstract:

The area and verification of routes is a vital part of vehicle interlocking rationale plan. The interlocking directions are typically listed manually by experienced signalling engineers, and this task is both time-consuming and expensive. In this paper, an automatic, graph theory-based approach to route location and verification is presented. In this new approach, a component-based model is used to represent the topology of the path layout, and a modified matrix algorithm based on graph theory is used to locate all of the routes in a given location. This algorithm exhibits superior performance in the position and verification of routes and is universally applicable, irrespective of the station layout. When a station is modified, the designers can merely update the topological data for the station, and the new route information can be automatically.

Keywords: **MANET, Location Detection, Route generation, Path Selection, Graph Theory**

INTRODUCTION

In early decades, wireless communications have turned into an essential piece of individuals' lives everywhere throughout the glove. Currently, commercial high data rate mobile services are pro-vided with a peak data rate of 28Mbps using a bandwidth of 5MHz. However, this peak data rate is insufficient and even higher transmission rates are already desired, especially when multiple users and multimedia traffic are considered. Inside the vehicle, the MSs are connected to the wireless access point to communicate. The APs are connected to one transceiver antenna on the top of the vehicle by wired connections. In the outside of the vehicle, the BSs for the highway is designed differently. The Omni-directional antennas are used for the vehicle to vehicle communication BSs for less energy consumption. The scope zone of one BS is covered with the closest BS as an element of the average expected speed of the vehicle. The vehicles Base Stations are associated with the BSC, MSC and PSTN as like the traditional cell framework and also there is a point to point association line between two contiguous BSs to exchange the control data right off the bat. At the end when the vehicle goes through the covering region of the phones, the handoff happens, and every one of the controls is exchanged from the past BS to the following BS to the moving heading of the prepare.

Rapid transportation conveys comfort to people groups' lives and is for the most part considered as a standout amongst the most feasible advancements for ground transportation. One of the vital parts of HSR development is the flagging framework, which is likewise called the " activity control framework," where wireless communications assume a crucial role in the transmission of prepare control information. We talk in detail the primary contrasts in relevant research for wireless connection between the HSR activity situations and the normally open and portable situations. The latest research progress in wireless channel modelling in viaducts, cuttings and tunnels scenarios are discussed. The characteristics of the non-stationary channel and the line-of=sight (LOS) spare and LOS multiple-input-multiple-output channels, which are the typical channels in HSR scenarios, are analysed. Some novel concepts such as composite transportation and key challenging techniques such as train-to-train communication, vacuum maglev train techniques, the security for HSR, and the fifth-generation wireless communications related techniques for future HSR development for safer, more comfortable, and more secure HSR operation are also discussed.

A calculus is developed for obtaining bound on delay, and buffering requirements in a communication network operating in a packet switched mode under a solid routing strategy. The theory created is not the same as conventional ways to deal with breaking down postponement because the model used to portray the passage of information into the system is non-probabilistic. It is gathered that the information stream went into the system by any given client fulfils burstiness requirements. An information stream is said to meet a burstiness requirement if the amount of information from the stream contained in any interim of time is not as much as

esteem that relies upon the length of the provisional. A few system components are characterised that can be utilised as building squares to show a wide assortment of communication systems.

METHODOLOGY

Algorithm for finding simple paths in a graph

From the simplified station topology graph (STG) obtained in the previous section, the route information for the interlocking system can be generated using graph theoretic techniques.

In graph theory, a simple path refers to a path along which no vertex is traversed more than once (Hart, Nilsson, & Raphael, 1968; Ore, 1962). A non-situated graph can be spoken to the in-network frame, and the more significant part of the primary ways in the chart would them be able to be gotten utilising lattice tasks(Danielson, 1968; Duckhm, 2003; Fisher & Wing, 1996; Ponstein, 1966; Sedgwick, 2003). Routes in the interlocking system have the same characteristics as simple paths, and methods designed for the location of simple paths in graphs can, therefore, be applied to the problem of route generation based on the STG.

Given an m-node graph G, A={a(i, j)} is the m _ m adjacency matrix of G, where a(i, j)=1 if an edge occurs between nodes i and j and is equal to 0 otherwise. Another m _ m network, B=b[b(i, j)], can likewise be developed, where b(i, j)=j if an edge exists between hub I and j and is equivalent to 0 generally. Moreover, assume that we have a network Pn=[pn(i, j)], where pn(i,j) is a polynomial used to portray the arrangement of inward hub results of the greater part of the basic ways of length n between hubs I and j. Each term in pn(i, j) is an essential hub item demonstrating straightforward ways of length n between hubs I and j. If the set of paths is/, then pn(i,j)=0. Also, we set pn(i, j) equal to 0 if i=j because closed routes should not occur in railway interlocking systems. There are two distinct simple paths with lengths of 3 from node a to node e, a-b-c-d and a-db-d-e. From hub a to hub e, the arrangement of ways spoke to by inward hubs is {bc, bd}. Consequently, p3(a, e) = bc+bd in the relating lattice P3.considering the prodect BPn=[cn+a(i, j)], we have cn+1(i, j)=P kb(i, k)pn(k,j) as indicated by the framework task rules. We see that b(i,k) – 0 when an edge connects node i to node k \, and pn(k,j) – 0 when a simple path of length n exists between nodes k and j. Therefore,cn+1(i,j) – 0 indicates that there is at least one possible simple path of length(n+1) that begins at node i and ends at node j. In fact, the nonzero terms cn+1(i, j) describe the set of internal nodes of all possible simple paths of length (n+1) between nodes i and k. A value of cn+1(i, J)= 0 indicates that no simple paths of length (n+1) exists between nodes i and j. As the paths represented by cn+1(i, j) are constructed by adding the head node i to the simple paths represented by pn(k, j), the product paths may not be simple unless node i is required not to be among the terms pn(k, j). The simple paths (including simple circuits) can be extracted by replacing i with 0. In the line of framework BPn. The primary slanting term cn+1(i,i) in BPn speaks to the shut ways (circuits) through hub I, and the circuits can be rejected from the arrangement of straightforward ways of length (n+1) by setting principle corner to corner terms of BPn equivalent to 0. These two activities are consolidated into one task known as basic().That is , Pn+a=Simple(Bon). The matrix P2-= Simple(BA) must be considered as a special case because P1 does not exist according to the definition of Pn. The Matrices A,B, BA and P3. Unmistakably to acquire the greater part of the straightforward ways, we can just emphasize utilizing the equationPm+1 = Simple(BPn) until Pm = [0] for all q>m_1,Pq is also equal to [0].

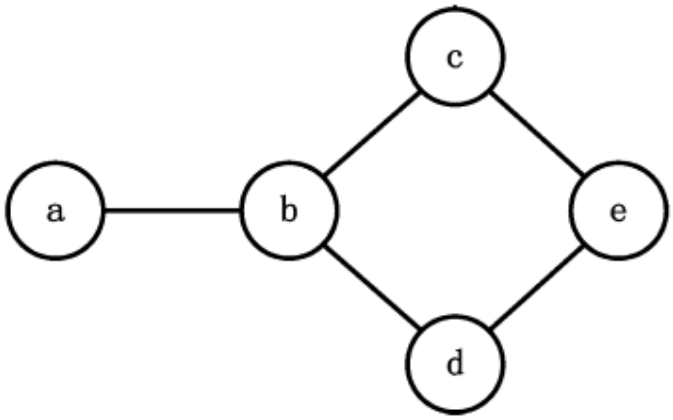


Figure 1: simple graph path

$$A=\begin{array}{c|ccccc} & a & b & c & d & e \\ a & 0 & 1 & 0 & 0 & 0 \\ b & 1 & 0 & 1 & 1 & 0 \\ c & 0 & 1 & 0 & 0 & 1 \\ d & 0 & 1 & 0 & 0 & 1 \\ e & 0 & 0 & 1 & 1 & 0 \end{array} \qquad B=\begin{array}{c|ccccc} & a & b & c & d & e \\ a & 0 & b & 0 & 0 & 0 \\ b & a & 0 & c & d & 0 \\ c & 0 & b & 0 & 0 & e \\ d & 0 & b & 0 & 0 & e \\ e & 0 & 0 & c & d & 0 \end{array}$$

$$BA=\begin{array}{c|ccccc} & a & b & c & d & e \\ a & b & 0 & b & b & 0 \\ b & 0 & a+c+d & 0 & 0 & c+d \\ c & b & 0 & b+e & b+e & 0 \\ d & b & 0 & b+e & b+e & 0 \\ e & 0 & c+d & 0 & 0 & c+d \end{array} \qquad P_2=\begin{array}{c|ccccc} & a & b & c & d & e \\ a & 0 & 0 & b & b & 0 \\ b & 0 & 0 & 0 & 0 & c+d \\ c & b & 0 & 0 & b+e & 0 \\ d & b & 0 & b+e & 0 & 0 \\ e & 0 & c+d & 0 & 0 & 0 \end{array}$$

The modified algorithm to obtain the route information

Because the switches have specified branching directions and the semaphores have specified guard directions, not all of the simple paths in an STG represent valid routes. The connectivity between certain components must be analysed, and the simple paths algorithm requires modification.

The connectivity of switches

Each switch component has three connecting points, which are not equivalent in their connectivity. Consider the switch w, as an example. Contingent upon the situation of the switch (typical or invert), a prepare can move from direct X toward either point Y or point Z and from either point Y or guide Z toward point X through w. Consider the paths Y-w-Z and Z-w-Y. These paths are invalid regardless of the position of the switch, and all of the simple paths including these sub-paths are wrong routes. Positions similar to point X are defined as switch root positions, denoted by root, and positions identical to point Z are defined as straight

branch positions of the switch, denoted by bn. Positions similar to point Y are determined as diverted positions of the switch and indicated by br The bn and b positions are equivalent in their connectivity and can, therefore, be known as b positions collectively.

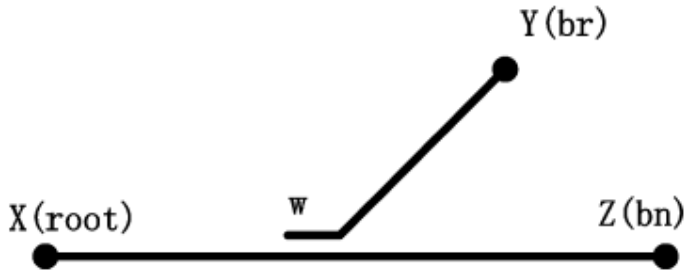


Figure 2: The connectivity of switches

To test the validity of the path, two flags associated with switches are introduced. One flag is referred to as flag in; if a path enters a switch from the root position, then flag in=1, and if a path opens a switch from the b position, then flag in=1. The other flag is denoted by flag out; if a path leaves a switch at the b position, then dwindle out=1, and is a path goes a switch at the root position, then flag out=a. The values of these flags are listed. After the path has crossed the switch, a new flag indicating the validity of the path is obtained by multiplying flag in and flag out. The product flag is denoted by flag v. A value of 1 for flag v indicates that the path is valid, while a value of _1 suggests that the path is invalid.

Improvement of the simple paths algorithm

To reduce the number of invalid routes among the simple paths obtained using the algorithm introduced, we make several modifications to the algorithm.

Table 1
Validity of paths through switches.

	Leave position	
	root	***b***
Enter position		
root	Invalid	Valid
b	Valid	Invalid

Table 2
Two types of flags associated with switches.

	root position	***b*** position
flag_in	+1	−1
flag_out	−1	+1

Definitions of the modified matrices

For the extraction of the correct routes from the set of simple paths given. By the matrix Pn. Several additional matrices and formulae must be defined. Based on the matrix B introduced, the additional matrices Bout and Bin out are introduced. The matrices Bout is constructed by adding the flag out flag to matrix B. For every row of

matrix Bout that is marked with the switch ID row header, each component that is connected at the root position is multiplied by_1, based on the matrix B. The matrix Bin, out is constructed by adding the flag in flag to the matrix Bout. For every column of Bin, out that is marked with the switch ID column header, each component that is connected at a b position is multiplied by _1, based on the matrix Bout. The corresponding B, Bout and Bin out matrices are provided. Similarly, based on the matrix A introduced, the matrix A out can be constructed by adding the flag out flag to matrix A. For every row of the matrix A out that is marked with the switch ID row header, each component connected at the root position is multiplied by _1, based on the matrix A. The A and A out matrices corresponding.

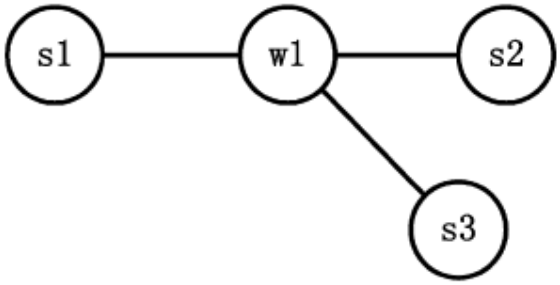


Figure 3: Simple TG having the single switch

$$B = \begin{array}{c} \\ s1 \\ s2 \\ s3 \\ w1 \end{array} \begin{array}{c} \begin{array}{cccc} s1 & s2 & s3 & w1 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & w1 \\ 0 & 0 & 0 & w1 \\ 0 & 0 & 0 & w1 \\ s1 & s2 & s3 & 0 \end{bmatrix} \end{array}$$

$$B^{out} = \begin{array}{c} \\ s1 \\ s2 \\ s3 \\ w1 \end{array} \begin{array}{c} \begin{array}{cccc} s1 & s2 & s3 & w1 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & w1 \\ 0 & 0 & 0 & w1 \\ 0 & 0 & 0 & w1 \\ -s1 & s2 & s3 & 0 \end{bmatrix} \end{array}$$

$$B^{in,out} = \begin{array}{c} \\ s1 \\ s2 \\ s3 \\ w1 \end{array} \begin{array}{c} \begin{array}{cccc} s1 & s2 & s3 & w1 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & w1 \\ 0 & 0 & 0 & -w1 \\ 0 & 0 & 0 & -w1 \\ -s1 & s2 & s3 & 0 \end{bmatrix} \end{array}$$

Matrices B, Bout and Bin out corresponding to the TG

$$A = \begin{array}{c} \\ s1 \\ s2 \\ s3 \\ w1 \end{array} \begin{array}{c} \begin{array}{cccc} s1 & s2 & s3 & w1 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

$$A^{out} = \begin{array}{c} \\ s1 \\ s2 \\ s3 \\ w1 \end{array} \begin{array}{c} \begin{array}{cccc} s1 & s2 & s3 & w1 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ -1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

Matrices A and A out corresponding to the TG

Route computation

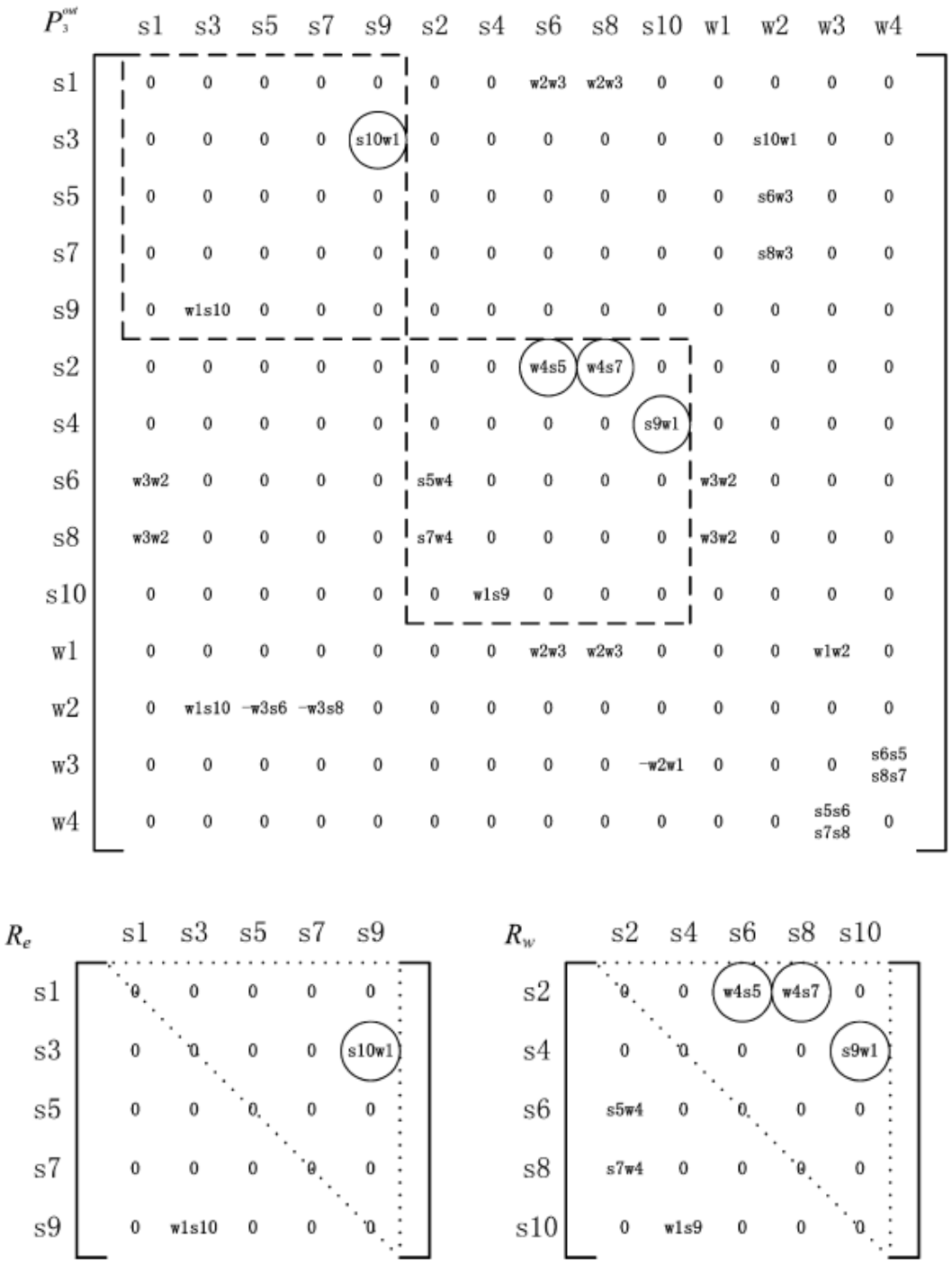
Example to illustrate the rout information computation process,

Step 1,Describe the station plane layout using a component-based topology model and simplify it (to obtain the STG

Step 2, Represent the k-node STG in matrix form. Generate the following k_h_matrices: B , A, Bout, Bin out and A out. In this case, K=14.

Step 3, Perform iterative computations using formulae obtain the matrices P n, Poutpvn, Pout n until n=m, where for any q>m_1, Pout q 1/4 ½ 0- It can be seen easily that m 6 Lenghmax<k, where length max is the length of the longest simple path in the STG. In this case, we can see that Pout 9 is equal to [0], and the iterative computation, therefore, terminates at n=9. Save all of the Pout n matrices generated by the iterative computation. The time complexity of the computations is K_K.

Sep 4, Identify the available routes in each of the Pout n matrices saved. The matrix Pout is used as an example to illustrate the identification process. Semaphores s1,s3.s5,s5.s7 and s9 all have a guard direction of E (east), and semaphores s2, s4, s6,s8, and s10 all have a guard direction of W(west). Each group of semaphores(east- and west-directed) can be used to generate a submatrix of Pout n: these sub-matrices are denoted by Re and Rw. All of the semaphores with a guard direction of E are contained in the matrix Re, and all of the semaphores with a guard direction of W are contained in the matrix Rw. If the semaphores in these submatrices are arranged in east-to-west or west-to-east order, then it is easiest to obtain the route information from the top right or bottom left elements of the sub-matrices, using the diagonal terms as a reference, The circles in routes of length 3. One eastbound route, [s3-s10-W\w1-s9], and three westbound routes, [s2-w4-s5-s6], [s2-w4-s7-s8] and [s4-s9-w1-s10], are obtained from Re and Rw.



The matrix Pout 3 and its two sub-matrices, Re and Rw

Table 3
Route information for the station shown in Fig. 2.

SN.	Route	Direction	Length	Referenced matrix
1	s3s10w1s9	E	3	P_3^{out}
2	s2w4s5s6	W	3	P_3^{out}
3	s2w4s7s8	W	3	P_3^{out}
4	s4s9w1s10	W	3	P_3^{out}
5	s1w2w3s6s5	E	4	P_4^{out}
6	s1w2w3s8s7	E	4	P_4^{out}
7	s6w3w2w1s10	W	4	P_4^{out}
8	s8w3w2w1s10	W	4	P_4^{out}
9	s3s10w1w2w3s6s5	E	6	P_6^{out}
10	s3s10w1w2w3s8s7	E	6	P_6^{out}
11	s2w4s5s6w3w2w1s10	W	7	P_7^{out}
12	s2w4s7s8w3w2w1s10	W	7	P_7^{out}

Route verification

To open a particular route from the route list, the interlocking system must first perform several verifications. More specifically, a route must meet the following necessary conditions to be opened.

I No conflicting route is opened.

ii. The switched\s (if they exist) in the target route must be in the correct positions.

Iii The sections in the target route must be in the clear(not occupied). It is straightforward to verify whether the sections of a route are clear. Once the route information is generated, the primary tasks of the interlocking system before opening a given route are conflict detection and verification of the switch positions.

Conflict detection

For a given railway station, all of the route information can be generated using the algorithm introduced. Certain routes cannot be opened at the same time. It is clear that routes sharing the same component nodes conflict. Consider the routes listed. The 1st and 8th routes conflict as they both contain s10 and w1 in their route information. The two routes, therefore, cannot be opened at the same time. Normally, the conflict information between routes is included in the interlocking logic, which is edited by experienced signalling. The conflict detection algorithm for target route k proceeds as follows

I Calculate the conflict matrix C using the operation R_RT

Ii Locate all of the nonzero terms in row K of matrix C and check whether the different routes indicated by these nonzero terms are open. If anyone of the different routes is opened. The matrices R and C corresponding to the routes. Take c(2,9) in the matrix C as an example, c(2,9) = 2 means the 2nd and 9th routes ([s2w4s5s6] and [s3s10w1w2w3s6s5] are in conflict as they contain two common elements(s5and s6].

R

	s1	s3	s5	s7	s9	s2	s4	s6	s8	s10	w1	w2	w3	w4
1	0	1	0	0	1	0	0	0	0	1	1	0	0	0
2	0	0	1	0	0	1	0	1	0	0	0	0	0	1
3	0	0	0	1	0	1	0	0	1	0	0	0	0	1
4	0	0	0	0	1	0	1	0	0	1	1	0	0	0
5	1	0	1	0	0	0	0	1	0	0	0	1	1	0
6	1	0	0	1	0	0	0	0	1	0	0	1	1	0
7	0	0	0	0	0	0	0	1	0	1	1	1	1	0
8	0	0	0	0	0	0	0	0	1	1	1	1	1	0
9	0	1	1	0	0	0	0	1	0	1	1	1	1	0
10	0	1	0	1	0	0	0	0	1	1	1	1	1	0
11	0	0	1	0	0	1	0	1	0	1	1	1	1	1
12	0	0	0	1	0	1	0	0	1	1	1	1	1	1

The matrix R corresponding to the routes

C

	1	2	3	4	5	6	7	8	9	10	11	12
1	4	0	0	3	0	0	2	2	3	3	2	2
2	0	4	2	0	2	0	1	0	2	0	4	2
3	0	2	4	0	0	2	0	1	0	2	2	4
4	3	0	0	4	0	0	2	2	2	2	2	2
5	0	2	0	0	5	3	3	2	4	2	4	2
6	0	0	2	0	3	5	2	3	2	4	2	4
7	2	1	0	2	3	2	5	4	5	4	5	4
8	2	0	1	2	2	3	4	5	4	5	4	5
9	3	2	0	2	4	2	5	4	7	5	6	4
10	3	0	2	2	2	4	4	5	5	7	4	6
11	2	4	2	2	4	2	5	4	6	4	8	6
12	2	2	4	2	2	4	4	5	4	6	6	8

The matrix C corresponding to the routes

CONCLUSION

In this paper, another approach is presented for acquiring course data in interlocking frameworks. This approach employs a component-based topology model and graph theoretic matrix method. The component-based topology model describes the station as a group of independent components, and a set of data structures is used to represent these components. To improve the route-finding efficiency given the station topology graph, a modified matrix algorithm, is introduced. An automatic verification method to e applied before a route is opened is also introduced. This solution is a universal method that is applicable to variety of station layouts. For each new layout, the user can obtain the route information automatically using a single application. At this stage, the structured data of components in a station are generated from the station layout manually. This will become a bottleneck especially in the large railway station layout. Therefore, we plan to develop a graphical tool for drawing station layouts and generating structured data. The planners of railways can use this tool to edit the station layout and the structured data can be automatically generated. In addition, there is further work remaining to be done. For example, the coding and configurations of the interlocking logic software and compiled manually, leading to difficulties in updating the software when the layouts of stations change. It is therefore useful to provide designers with an integrated framework in which the route information generator and auto-coding suite for the interlocking logic are combined

REFERENCES

[1]. G. Barbu, "E-Train—Broadband communication with moving trains technical report—Technology state of the Art," Int. Union Railways, Paris, France, Tech. Rep., Jun. 2010.

[2]. Coraiola and M. Antscher, "GSM-R network for the high speed line Rome-Naples," Signal Draht, vol. 92, no. 5, pp. 42–45, 2000.

[3]. "Requirements for Further Advancements for Evolved Universal Terrestrial Radio Access (E-UTRA) (LTE-Advanced) (Release 11)," 3GPP TR Specification 36.913, Technical Specification Group Radio Access Network, Sep. 2012.

[4]. B. A. et al., "Challenges toward wireless communications for high-speed railway," IEEE Trans. Intell. Transp. Syst., vol. 15, no. 5, pp. 2143–2158, Oct. 2014.

[5].K. Abboud and W. Zhuang, "Stochastic analysis of a single-hop communication link in vehicular ad hoc networks," IEEE Trans. Intell. Transp. Syst., vol. 15, no. 5, pp. 2297–2307, Oct. 2014

[6].R. Cruz, "A calculus for network delay—I: Network elements in isolation," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 114–131, Jan. 1991.

[7].Y. Jiang and Y. Liu, *Stochastic Network Calculus*. London, U.K.:Springer-Verlag, 2008.

[8].A. Burchard, J. Liebeherr, and S. Patek, "A min-plus calculus for end-to-end statistical service guarantees," *IEEE Trans. Inf. Theory*, vol. 52, no. 9,pp. 4105–4114, Sep. 2006

[9]. R. Cruz, "A calculus for network delay—I: Network elements in isolation," *IEEE Trans. Inf. Theory*, vol. 37, no. 1, pp. 114–131, Jan. 1991.

[10]. Y. Jiang and Y. Liu, *Stochastic Network Calculus*. London, U.K.:Springer-Verlag, 2008.