

Railway Signalling and Delay Estimation via Simulation

Arjun Krishna

ARK@CSE.IITM.AC.IN

*Department of Computer Science and Engineering
Indian Institute of Technology, Madras*

Abstract

The paper describes a method for delay estimation and congestion estimation in the railway network using simulations. The key idea is to model the railway network as a computer network with various stations being nodes and trains being packets and run a Discrete Event Simulator SimJava with well defined signalling protocols to observe some useful statistics. In this paper simulations are run on the static schedule of the Southern Indian railways; a dataset curated by scraping the indiarailinfo site. The simulations results in statistics about delays at various nodes and contention for a particular link in the network indicating congestion in the network.

The simulation results for schedule improvements are presented using the $O(\text{congestion} + \text{dilation})$ packet routing described by Leighton, Maggs and Rao [1] and also present a schedule improvement strategy which introduces small delays at some intermediate stations than just delay at source.

Keywords: Discrete event simulation (SimJava) , Delay estimation, Congestion estimation, $O(\text{congestion} + \text{dilation})$ packet routing

1. Introduction

Train Delay is the measure of deviation from the fixed schedule of the train, the delays can occur due to a variety of reasons like signalling delays, train interference etc., some of these reasons are controlled by the railway operators and some others indeterminable. The goal of the paper is to present a simulation method for estimating the train delays assuming a approximate model of how the railways actually performs.

Simulation methods are very useful when it is very hard to estimate some variables analytically. In the literature for delay estimation there has been a lot of analytical, history data based and simulation methods. The simulation method is apt for this setting as the railway network is a very complex entity and via simulations one can attempt to model the complex network as close as possible without a lot of relaxations, thereby enabling us to estimate the variables accurately.

Train Delay estimation plays an important role in railway optimizations, as it gives a better knowledge of the system and various factors that cause delays so enabling the railway operators to strategize and operate efficiently. The paper discusses some simple signalling protocols that are used by the railways and modelling them as a computer network proto-

col, enabling the use of discrete-time simulator SimJava to run simulations under certain assumptions of train speed, signal placement etc., to estimate the delays caused at various nodes.

The network simulation is run for a fixed schedule of the Southern Indian Railways which is curated by scraping the indiarailinfo site. The simulation is run for a weekly schedule, however this can be easily extended based on the computational resource available. The simulation logs the delay faced by a train at a particular node and also if there is contention it logs the array of trains contending for a link. This log is analysed to present the results in this paper, which gives a fair indication of how the schedule performs.

Schedule improvement can be done via the $O(\text{congestion} + \text{dilation})$ suggested by Leighton, Maggs and Rao [1], where all the trains are delayed at the source by a random amount of time. It is empirically observed that there is a significant reduction in average delay at a node and the average end-to-end delay a train faces. Also, it is shown that instead of an initial delay at the source of the train, a similar performance output can be obtained by performing small delays at random nodes on the path of the train.

2. Railway Signalling Protocols

Railway Signalling Protocols ensure rail safety and control railway traffic. The protocols dictate how the train has to travel between two stations. There are two main types of signalling methods

Time Interval Method : The trains are spread over the length of the track in such a manner that if the train ahead stops, the following train would be able to stop without collision. This method works if the traffic is less and would be risky when considering passenger trains.

Space Interval Method : In this method the track is divided into blocks, the protocol ensures that a train can enter a block only if there is no other train in that block.

Signalling Systems

Semaphore Signal : This was the earliest form of fixed railway signal, the information is conveyed to the locomotive driver from the inclination of the pivoted arm on a vertical post. This was the widely used form of mechanical signalling. Colour Light signals have replaced most of these Semaphore Signals today.

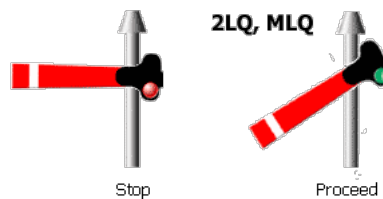


Figure 1: Semaphore Signal

Colour Light Signals : In this system colour lights are used to convey information to the train driver, the key advantage over Semaphore signal is that it is visible over longer ranges and there is no mechanical movements involved.

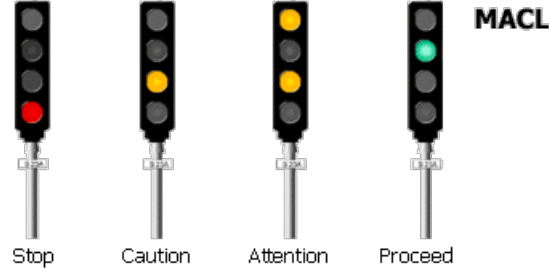


Figure 2: Colour Light Signal

The red signal indicating the train has to stop before moving to the next block. The green signal indicating that the train has acquired permission to proceed further. A single yellow (caution) conveys to the train driver to proceed and be prepared to stop at the next block. A double yellow signal (attention) conveys to the locomotive driver to pass the next signal at a restricted speed.

These protocols ensure that any two trains are never in the same block. In this paper the above protocols are modelled as a computer network protocol to enable us to simulate the train movement using the SimJava simulator running the Signalling Protocol.

3. Protocol for Modelled Railway Network

Modelling the Semaphore Signals

The Semaphore Lock Signal is modelled with the following event based algorithm. This is the protocol on which the simulation results are based on. The image below is a visual depiction of the protocol.

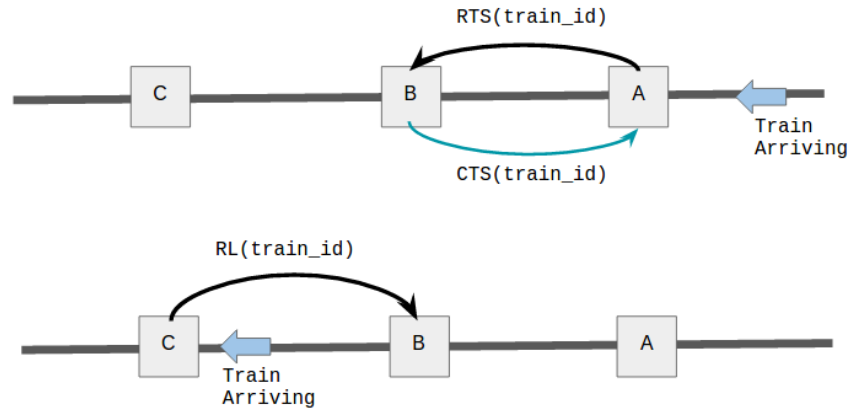


Figure 3: Semaphore Signal based protocol visualization

RTS : Request To Send | **CTS :** Clear To Send | **RL :** Release Lock

SIGNALLING PROTOCOL

- **Event** : Train Packet Received

Action :

- Send RL (train_id) to the previous node in the route.
- Send RTS (train_id) to the next node in the route.
- Store the packet in the buffer.

- **Event** : RTS (train_id)

Action :

Definition

- if node is a STATION, then L is a Semaphore Lock with maximum count being number of platforms.
 $M = \text{platform_count}$
- if node is a SIGNALLING NODE, then L is a binary Semaphore Lock.
 $M = 1$

if ($L < M$) :

 acquire(L)

 Send CTS(train_id) to the requesting Node.

else :

 Buffer RTS packet in requesters queue.

- **Event** : CTS (train_id)

Action :

- Send the buffered train packet identified by train_id to the next node.

- **Event** : RL (train_id)

Action :

if (requesters.empty()) :

 release(L)

 clear all train_id related storage in buffers.

else :

 Pop the RTS packet from queue

 Send CTS (train_id) based on the RTS packet to the requester.

Modelling the Colour Light Signals

Each signalling unit is a node in the modelled network and has a lock associated with it, a train that acquires this lock will be granted permission to move on to the next signal. If the train is reaching a signalling node A , The node A anticipates this and would send a message to the next signalling node on the track, say B , and the node next to B say C . If the lock at node B is acquired then the signal at A is red, else If the lock at C is not acquired then the signal at node A is green, allowing the train to proceed further. If it is acquired then the signal would be yellow.

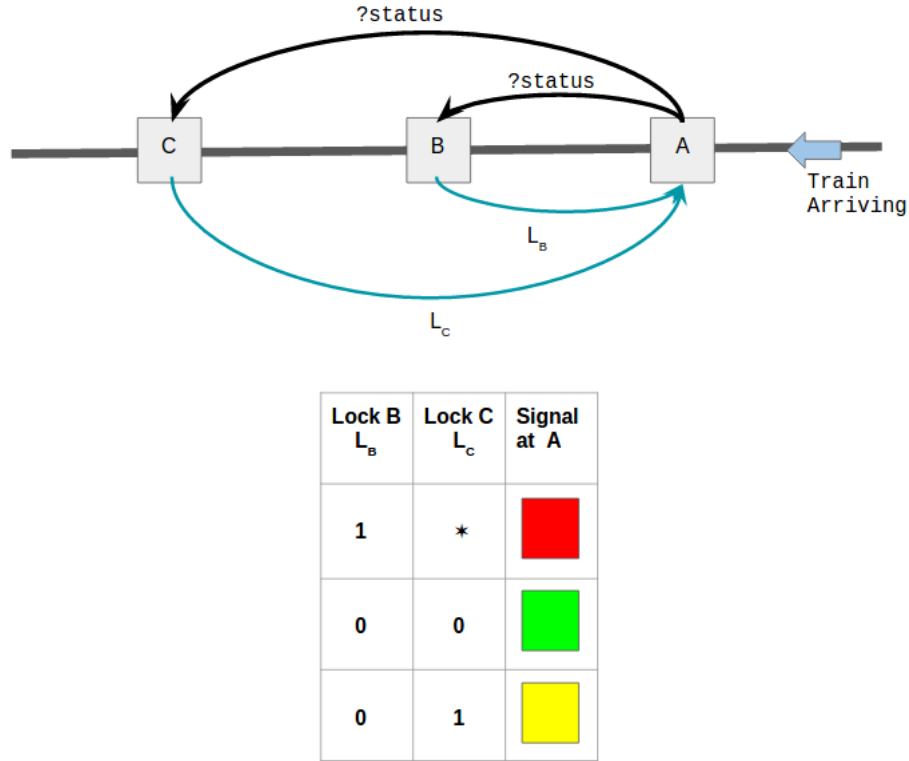


Figure 4: Signalling protocol based on the Colour Light Mechanism
(Lock is 1 if it is acquired by some other train already)

If the train is signalled red, then it waits at that block until the next block sends a message that it's lock is free. The signalling element is designed such that whenever permission for a train to acquire the lock is denied the event is logged and the requester gets a clearance once the lock is available.

Through this protocol train speed can be varied based on observed signal eg. if the train sees an yellow signal the packet speed can be decreased. Also speed can be varied based on the history of signals seen by the train. This protocol is fairly complex and is a part of the future work, the simulation results are obtained using the Semaphore based signalling scheme.

4. Train Delay Estimation

The train delay is the measure of deviation from the schedule, it is quite important to record these train delays or estimate them so that railway operations can be more efficient, with appropriate strategies developed to tackle delays.

There are a lot of reasons for delays to occur in the railway operations some of them are listed below

- Track and Signal Delays : caused due to track maintenance or signals that ensure reduced speeds for safe operation.
- Train Interference Delays : caused due to other trains in the route, includes delay caused in switching tracks.
- Equipment Delays : caused predominantly due to equipment failure or unscheduled maintenance.
- Weather Delays : caused due to the weather condition.
- Passenger Delays : accounts for the waiting time for passenger boarding and de-boarding, and delays in passenger service.
- Operational Delays : Delays due to crew operation of the train, such as delay in the starting of the train, detour etc.,

Though some factors like Signal Delays, Train Interference Delays can be obtained by direct simulations, the other delay factors have to be modelled as probability distributions based on historical data to include them in the simulation. Train delay estimation is an important problem and documenting delays at various parts of the network would be useful for the railway operations to strategize better. There are three widely studied techniques for train delay estimation in the literature: Analytical Method, Data-based Approach, Simulation.

Analytical Method

In this method the railway network is mathematically modelled, assuming a few parameters of the network and provide a probabilistic model of the trains. Using the mathematical formulation delay estimates can be obtained. Though, it is possible to get a clear mathematical formulations, it is hard to capture a variety of complex interactions between the trains and some initial mathematical assumptions might not be realistic. Hence, Analytical Methods do not capture the real-life complex interactions in the railway network, thus would not provide good delay estimates.

Data-based Approaches

With the availability of a lot of historic data about train journeys, Data-based approach of delay estimation is very practical and provides good estimates. This allows to consider various parameters that determine delay on a network and build a regression model with

these features and the train travel records to perform delay estimation. These models can also help in identifying important features that cause a lot of delay. The major drawback of this method is that there is no sufficient data available on the Indian Railways operation and hence the data based approach would be hard to apply.

Simulation methods

Simulation methods actually simulate the railway network in-order to extract actual train performance and delays. The simulations gives scope to handle various parameters of the network and hence can model the task on hand with flexibility and produce fairly accurate delay estimates. The simulated methods can embed a lot of complex interactions that are analytically hard to conceptualize and with a properly tuning of parameters the train delay statistics are easily obtained. Simulation models are intuitively good for delay estimation but it is still an approximation to the actual railway operation because it is very hard to capture all the complex features of operation.

In this paper a discrete time simulation model for the Railway Network (modelled as computer network) using SimJava is presented. The presented Network model simulates a static timetable of the Southern Indian Railway Schedule obtained by scraping the indi-arailinfo website. The model runs the Semaphore based Signalling protocol, and the delay a train occurs at a node is the time delay in getting a CTS packet once an RTS packet is fired, this delay is logged. Any contention for accessing a block is identified when buffering requesters and this is also logged as it directly indicates the congestion in the network.

5. $O(\text{congestion} + \text{dilation})$ packet routing

$O(\text{congestion} + \text{dilation})$ is the best one can do when it comes to packet routing, as both the factors act as a lower bound and have a grip on the end-to-end routing time. Leighton, Maggs and Rao [1] prove that it is possible to produce a $O(\text{congestion} + \text{dilation})$ schedule for routing.

The following online algorithm which does a random delay at the source of the train is adopted from Leighton, Maggs and Rao. Following are a few definitions after which the algorithm used is presented, for proofs please refer to the paper.

Definition :

1. Dilation d , is the longest routing path on the network. This is an obvious lower bound on the total routing time
2. Congestion c , is the maximum of the number of routing path crossing a edge.

Parameters :

- Let c be the congestion in the network
- Let N be the number of packets.
- Let d be the dilation of the network.

- Let α be a constant.

The following algorithm provides an $O(c + d \log(Nd))$ length schedule. It produces near optimal schedule for high c , though this is not the case in the Railway Network, schedule improvements using the following technique is observed experimentally.

Randomized on-line algorithm

1. Each packet is assigned a delay chosen randomly, independently and uniformly from the interval $[1, \frac{\alpha c}{\log(Nd)}]$
2. The packet waits for delay chosen at source, and is routed without stopping.

The following procedure produces a $O((c + d)2^{O(\log^*(c+d))})$ length schedule.

Definition :

1. A frame is a sequence of T time steps
2. Congestion of a frame, C is the maximum number of packets crossing any edge in the frame

Lemma : For every set of packets that has congestion c and dilation d , there exists a schedule of length $O(c + d)$ such that no packet ever waits in queues once it starts and the relative congestion of any frame of length less than $\log d$, is at most 1.

In the lemma, a schedule is created by the randomized on-line algorithm, which is used to recursively build a schedule to obtain the desired $O((c + d)2^{O(\log^*(c+d))})$ length schedule, which is done as follows.

1. By using Lemma, produce a schedule S where the number of packets that use an edge in any frame of length $\log d$ is at most $\log d$.
2. Break the schedule into frames of size $\log d$.
3. Consider each frame of length $\log d$ as a new routing problem that has dilation and congestion $\log d$, and solve it recursively, until the frames have constant size and at most a constant number of packets use each edge in each frame.

The choice of the parameter α should be quite high to be effective, but it also indicates the maximum delay that can be inflicted in the schedule. Results are obtained for simulations done using the randomized on-line algorithm provided by LMR.

The following is another alternative algorithm which unlike the randomized on-line algorithm that inflicts random delays at the source, induces smaller random delays at random

intermediate nodes on the route of the packet. In this case a frame is defined to be the subset of nodes in the route of the packet whose size is less than a fixed length L . Let β be a fixed constant, which is similar to parameter α but would be lower.

Random Delay at Intermediate Nodes

1. For each packet assign a source delay randomly chosen in the interval $[1, \frac{\beta c}{\log(Nd)}]$
2. As the packet is routed in its assigned path, with a probability $p = \frac{1}{L}$ pick a random delay in the interval $[1, \frac{\beta c}{\log(Nd)}]$ for a node in the path.

It is empirically observed that the above algorithm like LMR offers a significant reduction in end-to-end delays, and the frequency of seeing large delays.

6. Results

The modelled Network simulation is run on the Southern Indian Railway Schedule. The following are the details of the network

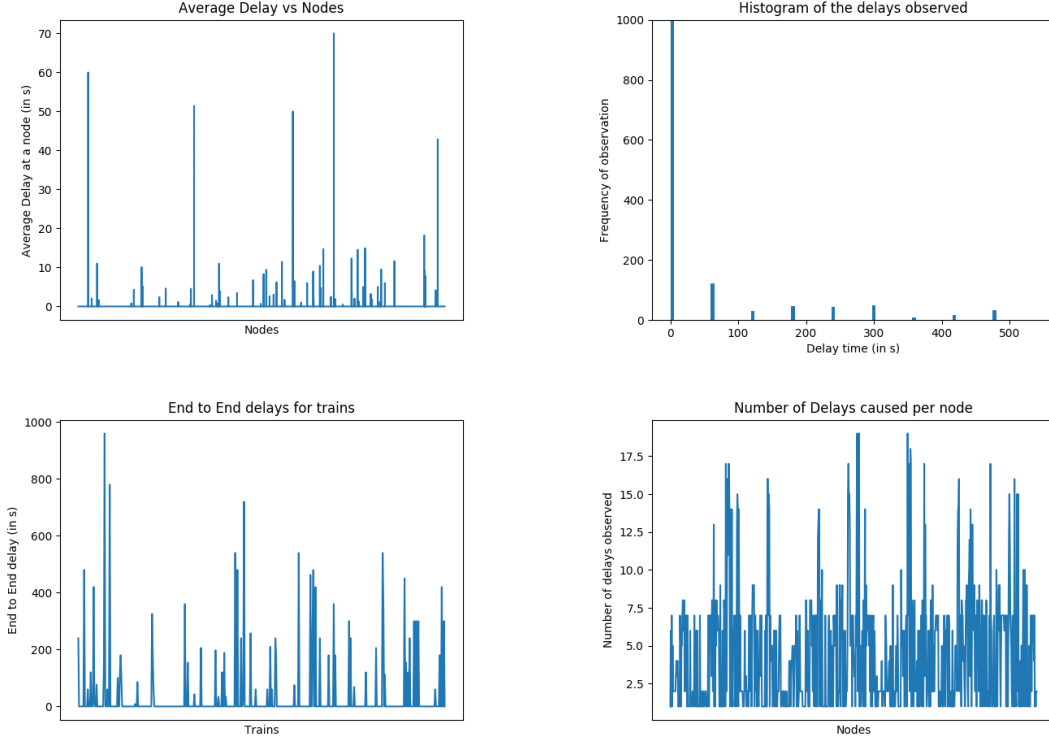
- Number of stations : 565
- Number of Signalling Nodes : 10766
- Dilation (d) : 356
- Number to trains Scheduled (N) : 2712

The time measured in simulation is the number of seconds since 00:00 AM, Monday. Using this base line all the departure timing of the train is converted to simulator time in-order to schedule it accordingly.

The result organization is as follows

- **Default** : Simulations run with default parameters as specified by the modelling paper. The congestion c is estimated from this run as it is more realistic than calculating the worst case congestion. Using this c the algorithms described in the previous section are run. It is observed from the data collected that $c = 3$.
- **LMR** : Simulations run with train packets randomly delayed at the source as specified by the previous section. The results are presented for the on-line randomized algorithm with parameter $\alpha = 3000$ which dictates the maximum delay at source to be $\frac{\alpha c}{\log(Nd)}$ which turns out to be about 10 min.
- **RDIN** : Simulations run with introducing small random delays in some intermediate stations in the path of the train. The results are presented for the Random Delay at Intermediate Nodes algorithm, with $\beta = 100$ which indicates that the delay at an intermediate node is at most 20 seconds ($\frac{\beta c}{\log(Nd)}$)

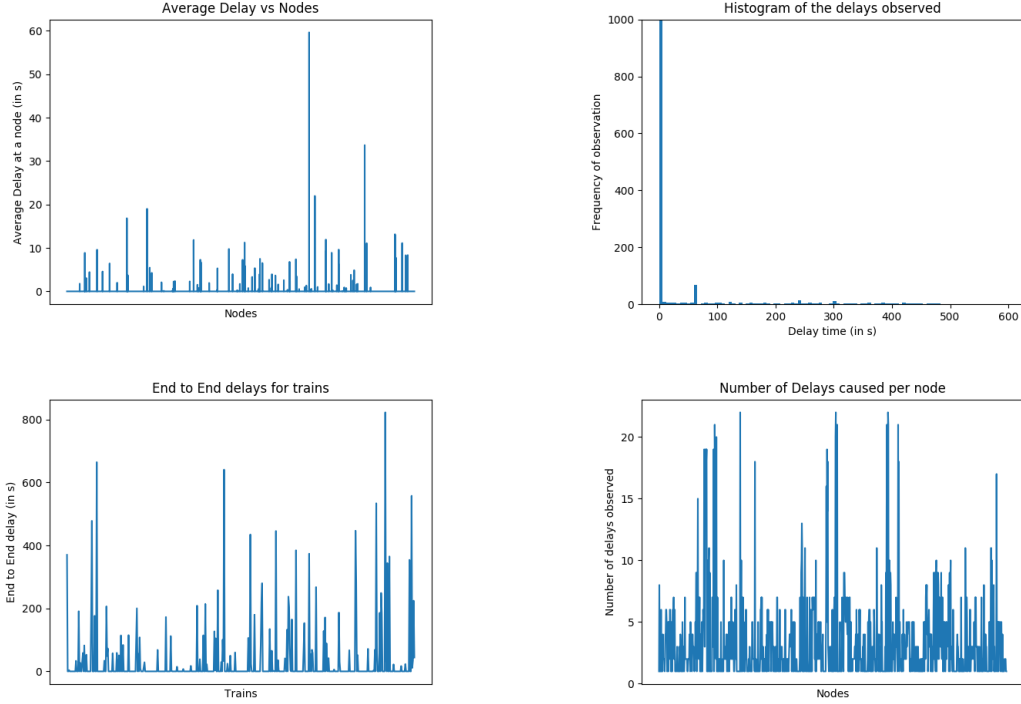
6.1 Default



The above plots indicate the simulation results for the default run of the schedule on the modelled network. The maximum contention observed is 3. The total end-to-end delay is at most 980 seconds (≈ 16 min). In the histogram of delays observed as expected most frequent are the low delays, but some taxing delays of upto 8 minutes are also observed.

The maximum congestion observed $c = 3$ is used as the parameter need in applying the LMR, RDIN schedule improvement procedure.

6.2 LMR

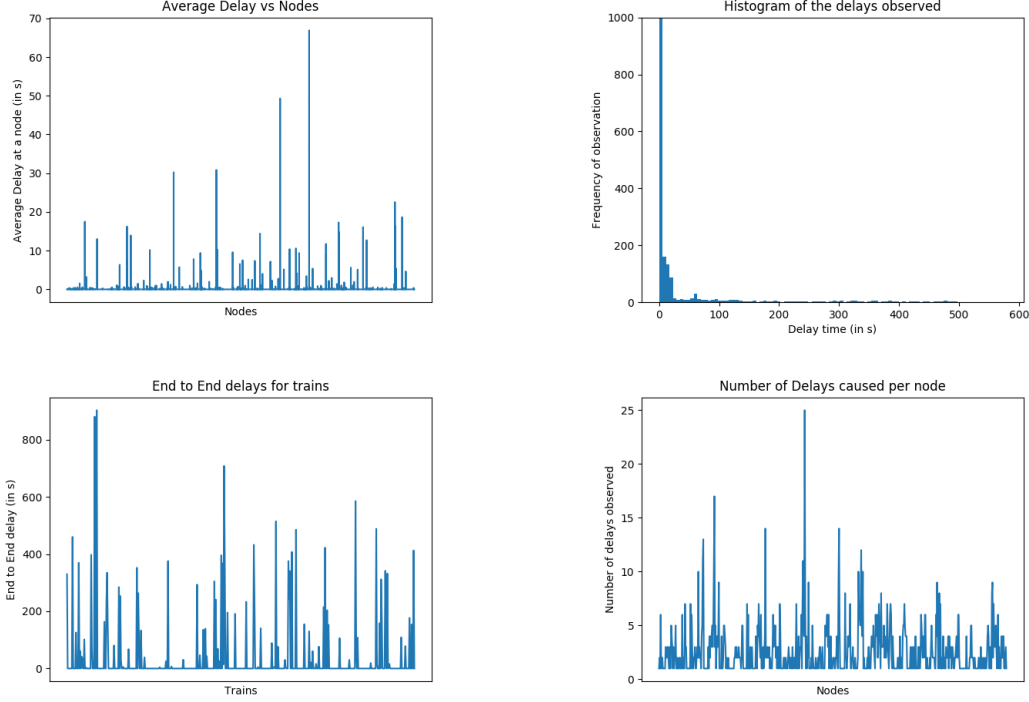


The above plots indicate the simulation results for the LMR randomized algorithm applied to the Southern Railway Schedule. The congestion $c = 3$ and parameter $\alpha = 3000$ indicates the maximum a train can get delayed at source is 10 minutes.

A significant reduction of the maximum end-to-end delay of a train is observed, reducing the maximum delay by around 3 minutes. The average delay that occurs at node is also significantly reduced. The histogram depicts that very large delays occur very rarely even less frequent than the default run. Though the number of delays caused by nodes might have slightly increased, the histogram convinces the fact that the delay caused is probably lower.

Therefore, as expected from theory randomized on-line algorithm presented by Leighton, Maggs and Rao [1] offers a significant reduction in major delays that occur due to contention and offer an improved schedule.

6.3 RDIN



The above plots indicate the simulation results for the RDIN algorithm applied to the Southern Railway Schedule. The congestion $c = 3$ and parameter $\beta = 100$ indicates the maximum a train can get delayed at any node on its path is at most 20 seconds.

A significant reduction of the maximum end-to-end delay is observed and is very similar to LMR algorithm. The average delay that occurs at node does not change much. The histogram still indicates that very large delays do not occur as often as the default case, however in performance is slightly poorer than LMR. The number of delays seem to have increased, but since it is rare to see high delays it is presumably very minor delay occurrences.

This method is in fact useful for schedule improvement as it is feasible to induce small train delays at an intermediate stations which brings down the end-to-end delay therefore can be easily incorporated in the schedule.

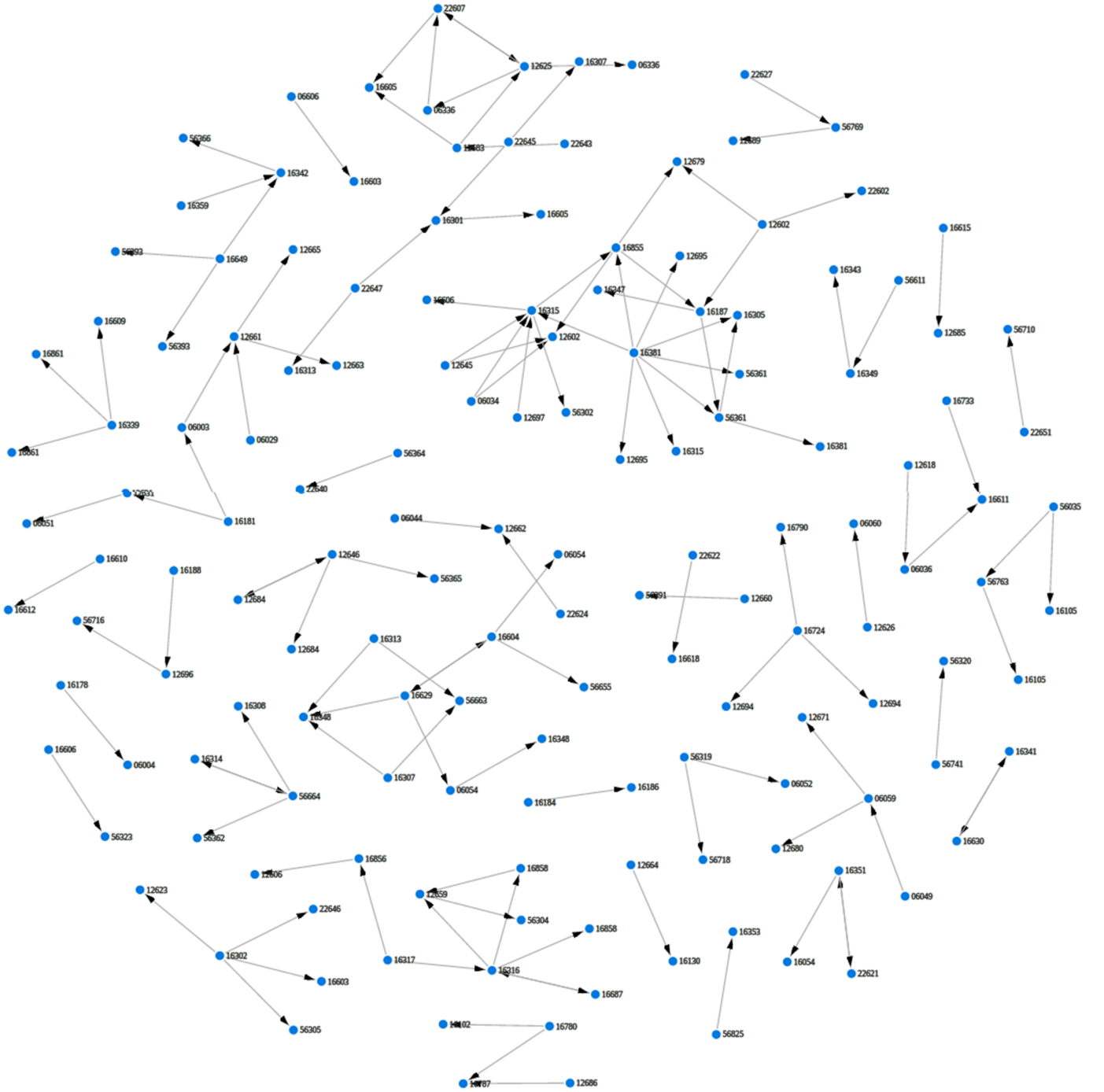


Figure 5: visualization of the reason for a train delay (*for the Default run*)
 Directed edge $a \rightarrow b$ indicating a is delayed due to b

7. Conclusion

In this paper a Discrete Event Simulation Program is presented, that can basically evaluate the performance of a provided input static-schedule. The schedule is simulated over a modelled network using a semaphore signalling scheme as the underlying signalling protocol. This program would be useful to evaluate schedules and help in optimizing it if necessary. Though the presented simulator makes a crude approximate of the underlying complex interactions, it is a very powerful technique that can be extended to adapt to more complex models. Experimentally it is observed that the delays caused due to signalling and the way the static schedule is designed can be reduced by using the on-line randomized algorithm suggested by Leighton, Maggs and Rao. The introduced alternative algorithm Random Delay at Intermediate Nodes [RDIN] also provides a schedule that has lower end-to-end delays, but is not as effective as the randomized on-line algorithm of LMR.

Source code : <https://gitlab.iitm.ac.in/cs14b058/Railway-Network-Simulation>

8. Future Work

The simulation model presented here is just a scratch on the surface of a plethora of things that can be considered as a simulation parameter. Railway operations being a complex interaction, modelling it exactly is near impossible. However with sufficient historic data of the operation simulation parameters can be derived for delays such as passenger boarding delay, weather conditions etc.,

The signalling in the railways is more complex as there are warning signals that ensure the train runs at a lower speed than the usual. These complex signalling mechanisms can also be modelled and implemented in the simulator.

Extraneous delay causing factors such as operational delays, equipment delay can also be modelled with the availability of historic data about such occurrences.

The throttling factor for modelling these parameters is the lack of availability of data. To overcome this one must develop some clever analytical approximations that can be incorporated with the existing simulation model to provide a better evaluation of the schedule input to the program.

Acknowledgments

I would like to thank Prof. Narayanaswamy for all the guidance and enriching discussions regarding the project. Also, I would like to thank Prof. Krishna Sivalingam for helping to choose SimJava as the choice of Discrete Event Simulator, which made the work considerably easier than using Network simulator ns3.

References

- [1] F. T. Leighton, Bruce M. Maggs, and Satish B. Rao. Packet routing and job-shop scheduling in (congestion+dilation) steps. *Combinatorica*, 14(2):167–186, 1994.

- [2] Introduction to railway signalling. <https://www.slideshare.net/sandeepatalay/introduction-to-railway-signalling>.
- [3] Visual depiction of signalling elements. <http://www.irfca.org/faq/faq-signal2.html>.
- [4] Causes of delays in railways operation. https://www.amtrak.com/servlet/Satellite?c=AM_Content_C&cid=1241245662251&pagename=am/Popup.
- [5] R. Wang and D. B. Work. Data driven approaches for passenger train delay estimation. In *2015 IEEE 18th International Conference on Intelligent Transportation Systems*, pages 535–540, Sept 2015.