

Cascading Behavior of delay in Dutch Train Transportation: Network patterns and a model

Koen Frankhuizen, Yun Li, Hanbin Liu

Abstract

With an increasing necessity for public transportation, the complexity of the train traffic network increases along with its vulnerability. A review of current work shows that a lot of work has been done on network topology of transport networks and capacity. We have applied these approaches to the Dutch transit network. We show that the Dutch network is a very dense network and the node distribution of the regular topology follows a power law distribution. By reversing the topology, we gain insight in the interaction between the routes on the network.

Using these two topology files we use a SIR algorithm to simulate the 'spread' of the delay throughout the network. We start with a theoretical approach taking a fixed infection and recovery rate to study the behavior of the SIR model for the train network. Second we extend the algorithm with edge dependent infection rates derived from real delay data. We use a heuristic approach to estimate the infection rates: we train on a set of five days of data and use correlation between delay patterns to infer edge delay propagation probabilities. The results we compare using a similarity analysis of the delay data. We find that our model simulates large delay events quite well. Furthermore we investigate the difference in delay patterns from nodes with high centrality and low centrality.

Section I: relevance and review

Introduction

With an increasing necessity for public transportation, the complexity of the train traffic network increases along with its vulnerability. Similar to many other real-world networks such as blogs, water distribution, etc., information travels in a train network. For example delay at one stop could cause to delay at other stops. Most analysis of vulnerability in a train network focus on performance per line or station. In our current work, we use a network analysis approach to characterize the transit traffic network in the Netherlands. We focus on the mainline train network, which is a dense network. By using two different network topologies, we show that the routes are highly connected. The goal of this work is to be able to model propagation of delay through the network and use model to determine crucial points in the network for delay vulnerability.

To train the network, dynamic data of spreading delay over this network is obtained by parsing the operation logs. We study the patterns and dynamics of delay

events spreading among the network too.

Understanding how delay propagated in the train transit network is useful for a number of reasons. First, it allows us to understand how delay can flow through the network and how to reduce the probability of cascade events from spreading. Second, it provides valuable information to optimize the train operations.

This article is organized in the following way: this section focuses on the relevance of the project and a review of related work. Section II describes the dataset and data preparation. Section III describes our current approach and results and Section IV includes future work / discussion.

Relevance of proposed work

In modern western countries, train networks are a common feature of the transport network. Although these networks have been extended over time, the demand for mobility have grown faster over time and therefore the intensity of use has increased. Among the most intensive used passenger networks are India, Netherlands, and the UK (1), measured as passenger-

kilometers per kilometers of rail track. With higher intensity the vulnerability of the network will grow. We focus on an important indicator of the service quality: the punctuality (delay) of the train service. Most analysis of this KPI focus on performance per line or station. In the current proposed work, we will use a different perspective and use a network analysis approach to study the properties of rail track network, and most importantly, to gain insight in the vulnerability of train networks to delay events.

Review of previous work

Network analysis has been applied to transportation systems before. Kurant and Thiran reviewed the topology of transportation systems (2), with stops as the node. In their paper, three different topology models were evaluated. As shown in figure 1(b), in the space of changes, two stations are considered to be connected by a link when there is at least one vehicle that stops at both stations; as shown in figure 1(c), in the space of stops, two stations are connected if they are two consecutive stops on a route of at least one vehicle; and in the space of stations (shown in figure 1(d)), two stations are connected only if they are physically directly connected with no station in between. In addition, the author proposed an algorithm for extracting both the real physical topology and the network of traffic flows from timetables of the transportation systems. They applied their algorithm to analyze three large transportation networks.

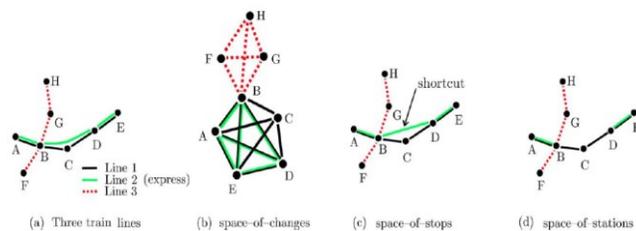


Fig 1. Different topological representations of transportation systems.

Similar to rare cascade events in random network, large scale delay of the transportation system could be triggered by small initial delay. Watts developed a simple binary decision model (3) with externalities which captures features with neighbor's nodes to study global cascades on random network.

Another approach to simulate propagation of information is the Markov chain, as described by the work of Crisostomi et al. (4) In their work, Crisostomi et al (4) use the dual approach to model the road network, where nodes correspond to roads and junctions are the edges, which is similar to the space of changes from Kurant and Thiran (2).

Although cascading behavior has been modeled in many real-world networks (5) (6), to our best knowledge, there is no publication modeling the cascade delay events using network pattern and simulation. There is no clear model examples how delay propagates through a transportation network using time ordered data. In current work, we will combine the transport network approach from above with methods from a different field within network theory: spreading of infections within a network. D. Easley and J. Kleinberg show a practical application of the SIR algorithm (7).

Section II: Data Sets and Data Preparation

The data source for the transport network and the delay information is acquired from Dutch open source website: www.ndov.nl. Specifically, two datasets have been downloaded and parsed in the current work. The first dataset contains detailed route and schedule information (timetables) and the second one is text information from operation logs which contained historical information on every trip, for every route, on every station per day in the past. The full train network in the Netherlands consists of approximately 392 stations. To prepare the data for network analysis, we applied the following filters / assumptions:

Timetable data:

We look at the Main rail network only (run by the national railways) and removed international / night trains from the dataset.

Even for the regular routes, a lot of irregularities are found. For example, a route might follow a certain pattern 9 out of 10 trips of the day, and stop at a certain stop only once (for example the latest or first trip of the day). Therefore, we only include stops which are addressed at more than 75% of the route trips.

The resulting timetable was cross-checked with route information from the National Railway website (NS.nl) and showed to be accurate. The next step was to convert the time table into network table.

Note the route is loaded for both directions so the mirrored edge-list is also created (which effectively generates an undirected network).

Delay data

The delay data is based on logs which deliver a network update every 5 minutes, for all trains departing in the Train Network for the next hour. Such an update consists of the Route ID, Stop ID, Time of departure/arrival according to Timetable and, if applicable, current delay with respect to the timetable. Similar filters have been applied for parsing the log data as we did with the timetable: we only look at the National Rail network and removed international /night trains.

Because updates were delivered every 5 minutes, many duplicates occurred. If duplicate rows (consisting of a RouteID, Stop and Departure/Arrival Time) contained different delay information for the same route and same trip, the latest delay entry was kept.

Section III: Approach and Results

The result section is divided into 4 subsections: first we assess the network by looking at general network characteristics such as degree and centrality. Second we use a theoretical approach to assess how the SIR model behaves using the regular and reverse topology. In the last two subsections we explain our correlation method to derive delay propagation probabilities and we use the edge infection rates to test our model to real data.

(1) Network analysis

In our first stage, we study transportation networks using exploratory data analysis. Two network topology models have been created to represent the train transportation system from the data sets mentioned above, following the approach described in our review of previous work (Kurant and Thiran (2), Crisostomi et al. (4)). The first topology consists of nodes being train stations and edges being the train route connecting two train stations. This is very intuitive and labelled as Normal Transit Network. The second topology consists

of nodes representing the train's route. Edges exist in the second topology if two routes share at least one train station. We refer to the second topology representation as Reverse Transit Network. Two network data are loaded to snap.py as undirected graphs. In addition, for the purpose of comparison, two Erdos-Renyi random networks are generated having the same number of nodes and edges as the above two networks.

Table 1 summarizes the train station time tables being used to generate both the normal and reverse transit network graph. Figure 1 shows the general picture of the transit network. Figure 1 is generated using the time tables and the geo spatial information of each train station in our graph.

# of Routes	# of Stations	# of Paths	Avg Paths per Route
74	252	1255	14.8

Table 1: Summary of Train Station Time Tables

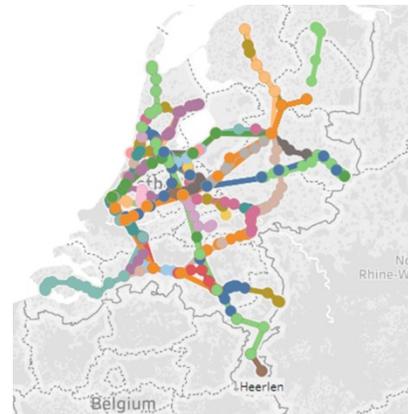


Figure 2: the train transit network

As shown in the figure 2, the transit network is a dense network. Most of the train stations are connected in some way, with only one route left without direct connection to the remaining part of the graph.

Comparing the two representations of the network, table 2 summarizes the basic graph information. Further we examine the degree of nodes of each graph and comparing the degree distributions of each train transit graphs along with their random counterparts in figure 2.

As shown in the degree distribution, the regular topology is close to a power law distribution (with fitted $\alpha = 2.5$), indicating that the network has a few high degree nodes central in the network and a lot of small

degree nodes. This can be explained by the observation that the train network consists of a few central stations with a lot of small stops in between. Reversing the topology, by seeing a route as a node and the connecting nodes as an edge, actually delivers an interesting pattern: we get a much more connected network: the ratio Nodes / Edge drops strongly (table 2). The resulting edge distribution does not follow the Power Law distribution anymore as shown in figure 3.

Regular Topology		Reverse Topology	
# of Nodes	# of Edges	# of Nodes	# of Edges
252	352	77	983

Table 2: Two network topology representations

The Reverse network graph shows a broad degree distribution (Figure 3b). It almost looks like there are two separate distributions, unlike the random graph with only one distribution. Unlike the reverse bus graphs shown in reference (8), which also followed a power distribution, we believe the binomial-like phenomena might indicate most routes are connected somewhere in the network through some key stations, which forms a large connection cluster. This finding is supported by plotting the cluster coefficient of those two network graphs which are shown in figure 4.

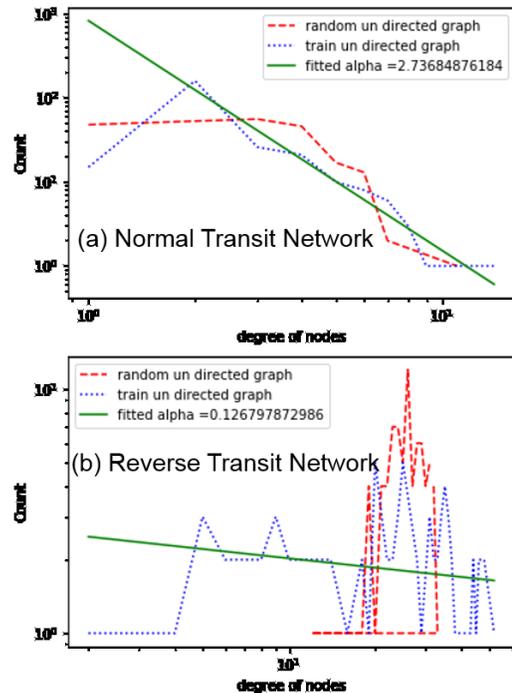


Figure 3: Degree distribution for normal and reverse transit network topology

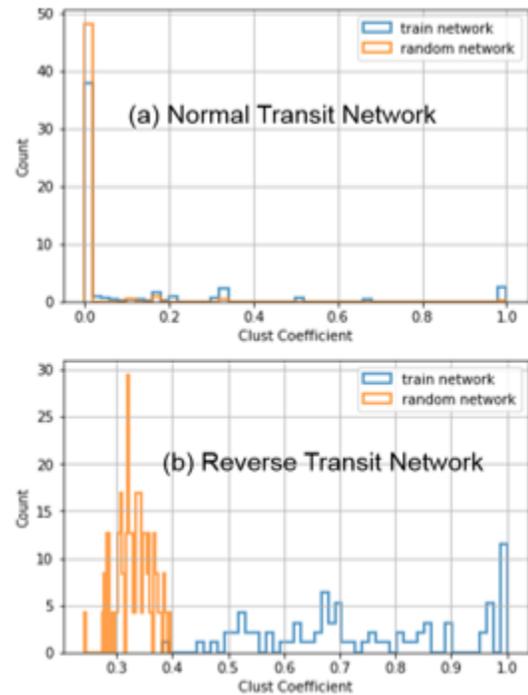


Figure 4. Cluster coefficient distribution

Other network properties such as the nodes' centrality is also calculated to help to identify the top 10 nodes with the highest degree and highest centrality. The results are summarized in supporting table 3(a) and (b).

Node	Station	Degree	Node	Station	Centricity
304	UT	14	304	UT	15119.3
356	ZL	10	11	AMF	8842.4
263	RTD	9	171	HT	7898.7
197	LEDN	8	24	ASD	7615.3
115	GD	8	30	ASS	7468.4
11	AMF	8	115	GD	5939.9
2	AH	7	263	RTD	5312.1
171	HT	7	356	ZL	4574.7
7	ALM	7	105	EHV	4090.5
94	DVD	7	22	ASA	3942.3

Table 3(a): node ranking normal network topology.

Node (=route)	Degree	Node (=route)	Centricity
800	52	800	234.25
500	48	3600	149.79
600	48	3500	147.76
4000	46	3000	114.51
3000	46	2200	96.77
1700	45	9200	89.73
7400	44	500	86.69
3500	44	600	86.69
9200	43	4000	74.11

Table 3(b): node ranking reversed network topology.

(2) Theoretical approach: SIR model with Fixed Infection Rate and Recovery Rate

Similar to the algorithm from lecture, we run a SIR model to simulate the cascade effect. Under the SIR model, every node can be either susceptible, infected, or recovered and every node starts off as either susceptible or infected. This is similar to the delay events. The SIR model assumes the distance between each nodes to be equal and a node (which is a station) to be fully 'infected' or not. In SIR model, every infected neighbor of a susceptible node infects the susceptible node with probability β , and infected nodes can recover with probability δ .

Our SIR simulations is performed as following:

- For each node in the graph, we initialize it as a delayed node (infected node) and started to run the SIR simulations until all nodes are recovered.
- We repeat the same process 100 times to get statistics on the infection rate.
- Since we have a fairly small network, we infect every single node in the graph and repeat the SIR simulations 100 times.
- The infection rate is set between 0.1 and 0.5 and recovery rate are set as 0.5 for the Normal Transit Network and for the Reverse Transit Network.

SIR simulations average infection rate vs node degrees are shown in Figure 5 a-f

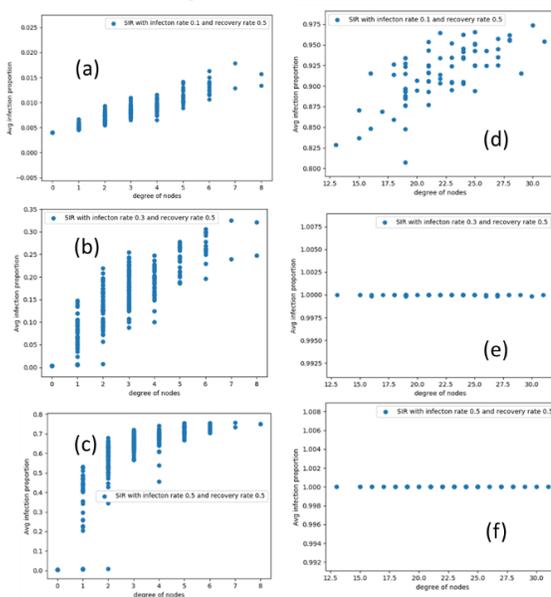


Figure 5 a – c: SIR analysis for normal transit network, d – f: SIR analysis for reverse transit network

We see that, as expected, the SIR analysis is strongly influenced by the infection rate. The very high density of the reversed network topology causes a relatively low infection rate (0.1) to reach almost 100% of the nodes, regardless which starting node is chosen. The normal network is much more robust – if we consider 50% of nodes reached as a threshold for a cascade event, cascades emerge at an infection rate of 0.5. In addition, we found that infecting the highest degree nodes does not lead to the highest infection rate.

(3) Deriving delay propagation probability: Delay similarity across the network

Delay is key information to estimate how the actual transit graph structure affected the network. We parse and aggregated the delay information into daily sets per node, where we binned the delay events (see figure 6). In figure 6, we show an aggregated binned delay distribution on May 31 2017 for all nodes. Most stops are on time and the distribution is strongly skewed (the y-axis is on a log scale).

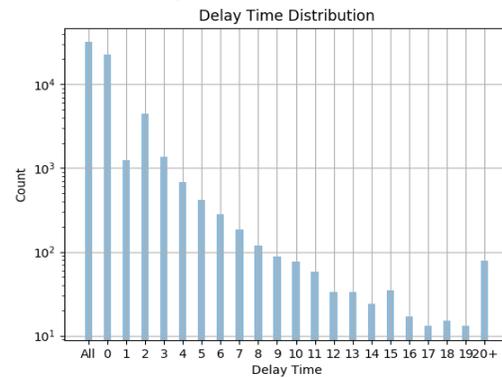


Figure 6: delay time distribution for all nodes

For each node we measure the delay rate as the number of delays divided by the number of stops. The resulting distribution is shown in figure 7. The outliers are 3 stations that had no delays and 1 stop which was delayed for every passing train. The distribution peaks at 22%.

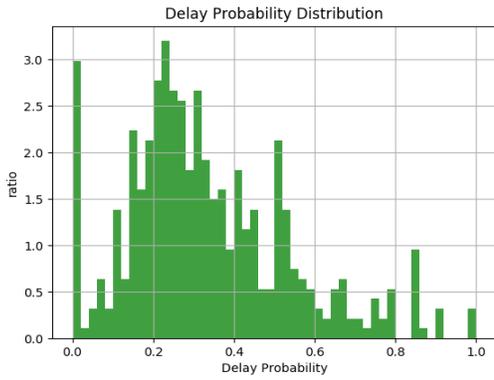


Figure 7: the distribution of Delay probability

Using these aggregations, we break down the distribution in figure 5 at station level and calculate the correlation between different stations. We show an example in figure 8. This figure displays the “similarity” distribution for the station (WD, node 335) against all other nodes. The correlation between two connected nodes can give us hint on how strongly the delay spreads from one station to another.

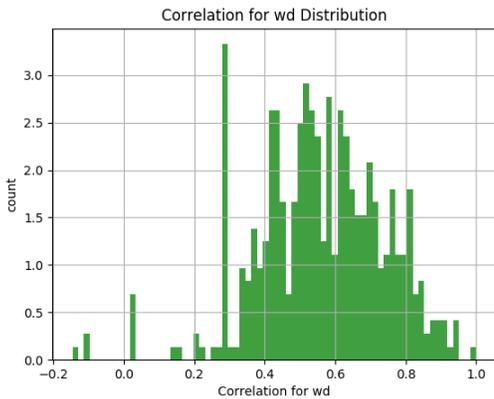


Figure 8: the distribution of correlations between station WD (node 335) and all other stations.

In our initial approach calculated the correlations between to stops with aggregated delay signature within one day. This provides us the distribution of all correlations between nodes for all existing network edges, shown in figure 9. As we can see most coefficients are very close to 1.

The correlation value from the one day aggregation therefore crowds around 1, which makes it less useful for the SIR model. Using a set of five days of train data shifted the distribution only slightly to the left (peaking at 0.9).

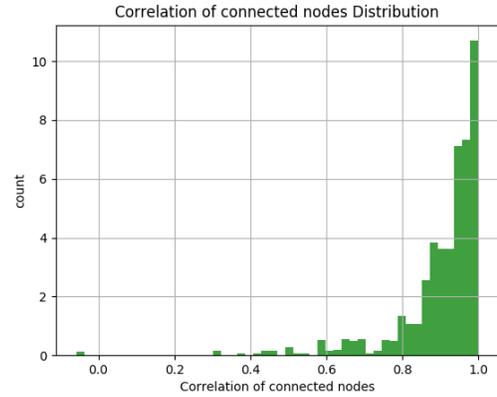


Figure 9: the distribution of correlations of delays on one day between nodes of edges

The high correlation factors can be explained from the fact that lots train arrivals may delayed by couples of minutes. Thus the delay events with small amount of delays, dominating the signature, will generate high correlation values. To get a clear signal, we only looked at delay events with a delay larger than 5 minutes. This is also the leading norm used as measure by the Dutch Railways (9).

The results, averaging over 5 days of training data, are shown in figure 11. Now the distribution has a much broader spread pattern and contains only a few nodes with high correlation. Most correlation values are close to 60%. The correlation along an existing edge will be used as the infection rate in the SIR model.

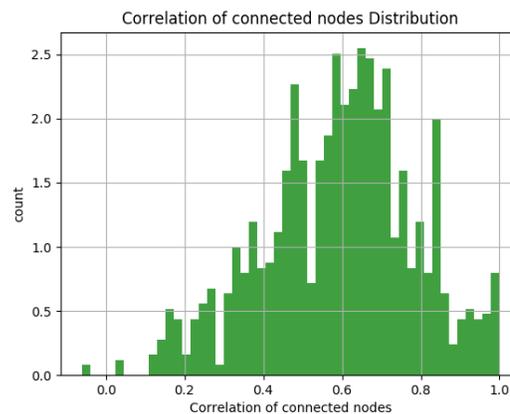


Figure 10: the distribution of average correlations of delays from 5 days between nodes of edges, with 5 mins or more delays.

(4) Testing the algorithm to real data:
Probability of Delay infection

The train transit network are heavy traffic network. As shown in figure 6, on average, there is about 100 passing per station per day. Under the general assumption that the delay event does not carried over to following day, the similarity score of the delay signal can be used as the delay infection probability.

Here we build our probability of SIR model using the Pearson correlations coefficient (see previous section). Five days correlation coefficient of connected nodes have been calculated and average results were used in the SIR model as infection probabilities to each train passing event independently. The recovery rate is kept at 0.5.

Furthermore, delay signals are assumed to propagate only to nodes connected via a route. The influence of this assumption is small however. This reverse topology showed that the routes are highly connected (almost every routes interacts with every route). Therefore requiring the delay to propagate along routes has limited influence.

The performance of the SIR model with individual probabilities have been measured in the following ways:

- 1) We isolate key delay events for the daily delay date. A key delay event is determined as more than 4 passing trains delayed for equal to or more than 10 minutes at one station (node). 17 of such events where found.
- 2) From the time that events happens, we search all the delay data in the train network within the 1 hour time frame. The set of delay stations as well as percentage of the infections were saved as the true outcome.
- 3) We simulated the major delay event using the probability SIR model and compare the results with the true outcome. We run the simulation for each event 50 times, starting from the node identified as the start-node of the event.
- 4) We use the purity score and the average precision score to measure how well the simulations results agree with the true outcome.

The results of the probability SIR model, as well as the initial infections station, size of infections in both the true outcome and the SIR simulations are shown in Table 4.

Date of real event	Start-node	Inf.rate true outcome	Inf. rate SIR	Purity score	Avg. Precision score
2-May	7	0.65	0.61	0.66	0.69
2-May	24	0.73	0.64	0.73	0.76
6-Apr	24	0.75	0.67	0.75	0.78
13-Apr	24	0.82	0.66	0.82	0.85
16-May	32	0.51	0.64	0.60	0.57
2-May	87	0.59	0.68	0.62	0.64
2-May	94	0.63	0.64	0.65	0.67
24-May	128	0.61	0.68	0.63	0.65
5-Apr	171	0.62	0.64	0.64	0.66
13-Apr	263	0.77	0.65	0.77	0.81
6-Apr	279	0.77	0.66	0.77	0.80
12-May	279	0.68	0.67	0.68	0.73
28-Apr	279	0.68	0.68	0.68	0.72
23-May	304	0.73	0.66	0.73	0.79
3-Apr	356	0.50	0.62	0.59	0.55

Table 4: comparison of simulations with ‘true outcome’ running 50 simulations for each ‘true event’.

Table 4 show that major events can be fairly well simulated – the average purity score (from 50 runs) lies between 0.6 and 0.8. Major events are covered quite well by the simulation model. This was too be expected, as we inferred the infection rates by using more severe delay events. The model can therefore be used to measure the vulnerability of the network to major delays.

To measure the vulnerability of the network to major delays, we also modelled the station with key network properties: looking at nodes ranking high or low for

betweenness centrality. We used the infection rates inferred from the correlation analysis. The results are shown in figure 11a and 11b.

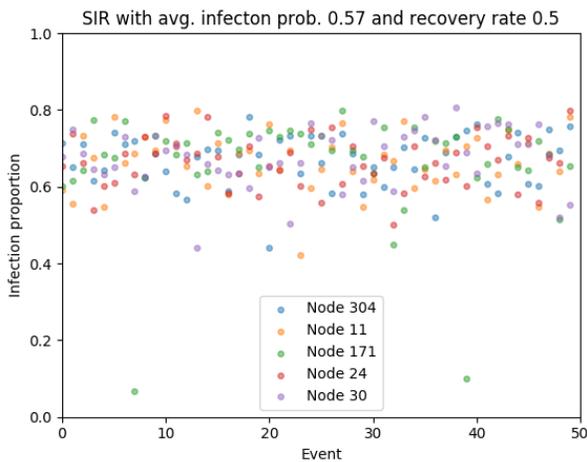


Figure 11a: infections starting at the top 5 nodes ranked to centrality

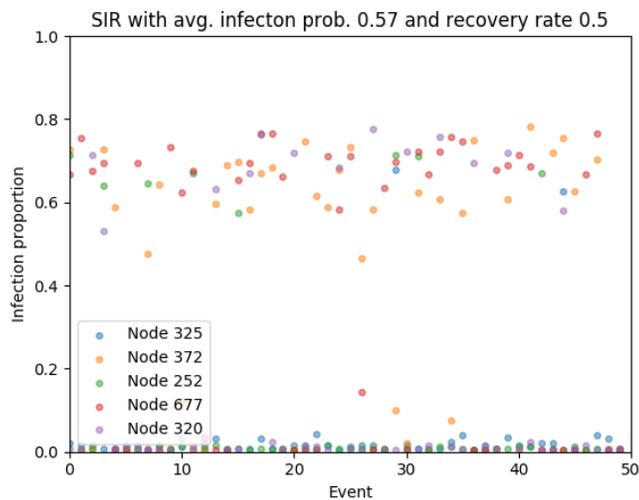


Figure 11b: infections starting at the top 5 nodes ranked to centrality

As shown in figure 11a and 11b, infections starting at nodes with high centrality have on average a higher infection rate than infections starting at low centrality nodes. Nodes with high centrality (with one exception for node 11) always trigger a cascade. However nodes with low centrality show on/off behavior: either a cascade is triggered (and then a fair amount of the network is reached), or (almost) no cascade is triggered. This is probably because these nodes lie at the edge of the network – the delay has to propagate across a few nodes before the ‘main’ network can be reached. The simulations are sensitive to the recovery rate. When we

set the recovery rate to 0.75 the high centrality nodes have to an average infection rate of 50% and when setting it to 0.9 few events infect more than 50% of the network. This might provide insight for train network developers – delays might difficult to prevent but by keeping the recovery rate high delays can be prevented to spread.

Section IV: Discussion / Further research

The SIR model has as advantage that it is able to capture infection spreading very well and has a few, very clear parameters which we could adjust based on real data. Therefore we are able to simulate major delay events.

The disadvantage however is that it assumes all edge distances to be constant, while in the actual network distances between nodes vary (shown in figure 2). Also not all routes stop at all nodes (e.g. some routes connect main ports). Furthermore, an infection event ‘infects’ the node completely. In the real situation, in particular busy stations consists of many routes passing at the same time. Therefore some routes might be delayed whereas others aren’t - and the delay should then only propagate across the infected route. The SIR model is not able to capture this. Also nodes are assumed to recover and not become susceptible afterwards. Station can be infected more than once. Using a SIS model might deliver different (interesting) results as well.

Our probability model considered both the network structure and the individual properties of the train station themselves. The probabilities of delay progressing along the edges in time are estimated using the statistical value of correlation coefficient. The correlation was based on delay data from a full day and therefore capable of distinguishing major events. A more elaborate approach might base the correlation factors on smaller timeframes (for example, 3-hour windows) to find the propagation probabilities between edges for smaller delay events. Also the correlation approach to infer edge probabilities does not directly use the network topology. However we show that using a SIR model is a relevant approach to model delay events in a train network and can be extended further

to capture a wider array of delay events (small and larger events).

Our SIR model used a fixed recovery rate of 0.5 which showed pretty consistent results (higher recovery rates for example lead to quickly dying infections, even at the most central nodes). However we did not derive the recovery rate from the data and we assumed it to be fixed across the nodes. Extending the delay data analysis to derive a good approximation for the recovery rate per node could make the model resemble the real situation more closely and could be an interesting topic for further research.

Timeseries are also often used in cascade analysis (e.g. Leskovec et al. (5)). The delay data turned out to be quite complex to analyse (each day consisting of approx 1.2M rows). By removing the duplicate messages we are able to reduce the amount of messages to approx 50 thousand messages a day, which still adds to 1.6M rows a month. This was our main argument to use the heuristic approach to infer delay probabilities. A more elaborate but very interesting approach would be to use time series to analyse the delay events and infer propagation probabilities. This approach might as well help to capture smaller events and for example provide multiple edge probabilities for different types of delays.

Bibliography

1. **Worldbank.** *www.worldbank.org.* s.l. : Worldbank.
2. *Extraction and analysis of traffic and topologies of transportation.* **Kurant, M. and Thiran, P.,** s.l. : Physical Review E , 74 (3), p.036114, 2006.
3. *A simple model of global cascades on random networks.* **Watts, D. J.** 2002.
4. *Google-like Model of Road Network Dynamics.* **Crisostomi, E., Kirkland, S., Shorten, R.** 2010.
5. *Cascading Behavior in Large Blog Graphs.* **J. Leskovec, M. McGlohon, C. Faloutsos, N. Glance, M. Hurst.** s.l. : Proc. SIAM International Conference on Data Mining, 2007.
6. *Tracking information epidemics in blogspace.* **E. Adar, L. Adamic.** s.l. : Proc. Web intelligence, 2005.
7. **Kleinberg., David Easley and Jon.** Chapter 21, Epidemics. *Networks, Crowds, and Markets: Reasoning*

about a Highly Connected World. s.l. : Cambridge University Press, 2010.

8. *Scaling and correlations in three bus-transport networks of China.* **Xinping Xu, Junhui Hu, Feng Liu, Lianshou Liu.** s.l. : Physica A, 2007, Vol. 374. 441-448.

9. **Railways, Dutch.** NS News. *NS News; NS Punctuality.* [Online] NS, January 2017. <http://nieuws.ns.nl/punctualiteit-ns-in-2016-licht-gestegen/>.