

# CONGESTION-AWARE ROUTING (CAR): VEHICULAR TRAFFIC ROUTING BASED ON REAL-TIME ROAD OCCUPANCY ESTIMATES

M. WILSON\* AND K. O. BOATENG

(M. W.: *Institute of Scientific and Technological Information (INSTI), Council for Scientific and Industrial Research (CSIR), P. O. Box M32 Accra-Ghana, Cantonment, Accra, Ghana; K. O. B.: Department of Computer Engineering, Kwame Nkrumah University of Science and Technology, PMB University Post Office, Kumasi, Ghana).*

\*Corresponding author's email: [michaelw@csir-insti.org.gh](mailto:michaelw@csir-insti.org.gh)

## Abstract

This work addresses the problem of routing vehicular traffic on road networks. Fair routing is effected using real-time data acquired from a sensor network superimposed on road networks. Routing information is in the form of which route provides the fastest set of interlinked road segments between any departure-destination pair of nodes. The work adopts Dijkstra's Shortest Path First (SPF) routing algorithm and derives a suitable routing metric from road occupancy data as a major contribution of this work that makes the SPF algorithm applicable to vehicular traffic routing on road networks. Also, a hypothetical road network and a corresponding Mobile App is used to illustrate our novel vehicular traffic routing algorithm. It is shown in this work that the method is more practical and easier to realize than a method in literature — Spatial and Traffic Aware Vehicular Routing (STAR).

## Introduction

Lots of efforts have been made in different countries with different expertise in attempt to either eliminate, control or reduce traffic congestion. The truth is, the number of cars in many countries keep increasing and is same in Ghana as we strive to attain a middle-level income status. Vehicular traffic flow ( $q$ ) is traditionally defined as the number of vehicles ( $n$ ) passing through a given cross-section of a road network per unit time ( $T$ ) and road occupancy ( $k$ ) as the average number of vehicles occupying a unit distance of that road subsection (Kornhauser, 2005) (Heydecker, 2011);

traffic congestion on vehicular networks occurs when the occupancy of a road's subsection or the entire road-network exceeds the routing capabilities. If decongestion strategies are not employed quickly in such situations, the network may encounter congestion collapse where the capacity of the road network may seem to be lower. This effect is considerable, and values of a performance reduction up to 30 percent is recorded (Kornhauser, 2005).

This work reviews existing work in the scope of traffic management on both road and data networks. It follows by examining some similarities between data and vehicular traffic

congestion and some data network routing techniques and outlines the congestion aware routing algorithm as a strategy to address the problem of decongesting vehicular traffic on road networks using transport telematics to compute the instantaneous least cost path with regards to congestion state and travel time for departure-destination nodes (junctions) of a journey. The research results are presented, discussed and the work is finally concluded. Traditional attempts to solve the problem of road traffic congestion have concentrated on either building more roads or expanding existing road networks. These traditional methods have mostly failed, and this failure is explained by the principle of Induced Traffic (Litman, 2001). Litman concluded that about 60 percent to a little over 90 percent of improved road capacity in urban areas is consumed with new traffic in less than six years. Work by Edward Leigh also classifies congestion reduction methods as either temporary or virtuous and agrees that road capacity freed up by temporary measures are filled by induced demand (Leigh, 2017). His work discusses Road pricing as the closest of temporary solutions to a

one-hit solution to traffic decongestion. Other approaches discussed in Leigh's work include optimized traffic light management, the use of CCTV to monitor road conditions and the use of inbound flow control. With the inbound flow control technique, vehicles are deliberately queued released from gates at a rate that matches the roads outflow rate. This however requires that all traffic flows freely beyond the gate and is not guaranteed to be so.

A review of decongestion strategies on data network (Bergamasco *et al.*, 2005; Sridharany *et al.*, 2013) reveal that data networks encounter data traffic congestion usually as a result of data packets from multiple sources needing to go through the same path with limited bandwidth to their destination and mismatch between system parts where low bandwidth lines are unable to forward packets at the same or better speed than the arriving speed of packets. Vehicular networks are plagued with traffic congestions for similar reasons. Improving a subsection of a road network in an attempt to reduce congestion usually results in mismatch between road subsections as shown in Fig. 1.

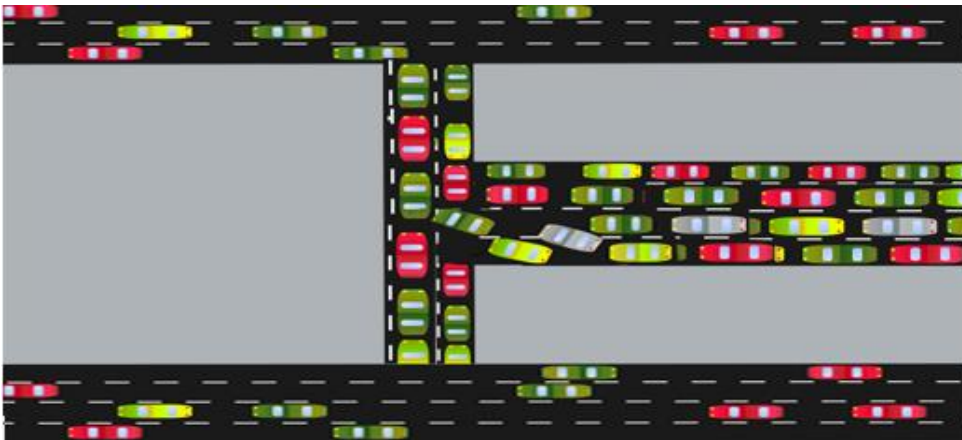


Fig. 1: Traffic bottle neck due to capacity mismatch.

Work by Guidici & Pagani (2005) addresses the issue by suggesting that vehicles be equipped with network interface cards that will allow them to partake in vehicular network infrastructure and making routing decisions based on the vehicular network state coupled with the geographical layout of the road networks. That, however, requires that all vehicles within area of interest are connected to yield accurate results. The solution also is limited by the coverage area of the wireless technology used by the vehicles. This work proposes a strategy to decongesting vehicular traffic by re-engineering a data network algorithm and using transport telematics to compute the instantaneous least cost path with regards to real-time congestion state and travel time for departure-destination nodes (junctions) of a journey. The main contributions of this work include the modeling and re-engineering of a data network algorithms for its application in the area vehicular networks. To achieve our overall goal the following specific objectives were achieved.

- i. The work selects a suitable routing algorithm for modification and adoption.
- ii. Derives and simulates a suitable metric for the application of the dynamic algorithm on vehicular networks.
- iii. Formulates the method of congestion-aware routing and illustrate its use based on an assumed sensor network infrastructure.
- iv. Implements the method as a mobile application and demonstrate its ease of use.

### Experimental

Our proposed solution identified Dijkstra's shortest path first algorithm as suitable for adoption and re-engineering to device a fair routing for vehicles on road networks. We employed the concept of classical Proportional

Integral Derivative (PID) controllers to adjust road occupancy data acquired from an assumed sensor network superimposed on the road network. Real-time vehicular occupancy data on road networks is collected and converged to a central database. We exploit principles of graph theory to attain topological design of a sub-section of the road network as a data structure (A MAP) implemented using the JAVA programming language. This map served as a routing table for the reengineered SPF algorithm with nodes of a graph representing road intersections and the road links representing the edges of the graph. Estimated travel time for each road link is computed based on occupancy-speed relationship and speed/distance/travel-time dependencies. That allowed fair routing to be effected using real-time data. Routing information is presented in the form of which route provides the fastest set of interlinked road segments between any departure-destination pair of nodes of road network. Routing traffic in such a way minimizes the overall traffic congestion on road networks.

#### *The Primary Metric and its estimation*

To be able to apply the Shortest Path First algorithm to road networks, there was the need to derive a performance metric close to a true measure of traffic behavior on the road network. We use the study of occupancy-speed and speed/distance/travel-time dependencies to compute the estimated travel time of vehicles on the road network based on the degree of deviation of occupancy.

Greenfields postulate (Kornhauser, 2005; Heydecker & Addison, 2010) in 1934 describes a fundamental association between speed and density/occupancy as  $v = f(D)$ , which represents a descriptive model of traffic behavior depending on the choice of function  $f(\cdot)$ . We

first considered a way in which a value (occupancy) that relates to the fundamental equation can be measured practically on road segments. We then analyzed different functional forms that could be used to represent the significant relationship between the measured occupancy value and average speed of vehicles. The direct and inverse relationship between the speed, distance and time thus, gave us a way of predicting estimated travel time for any subsection of a road network under consideration. This estimated travel time is then substituted for the routing metric of the adopted SPF algorithm.

#### Road occupancy measurement

Road occupancy data for the various road sub sections is obtained from electronic devices fitted with ultrasonic sensors and mounted at strategic entry and exit points on road networks as shown in Fig. 2.

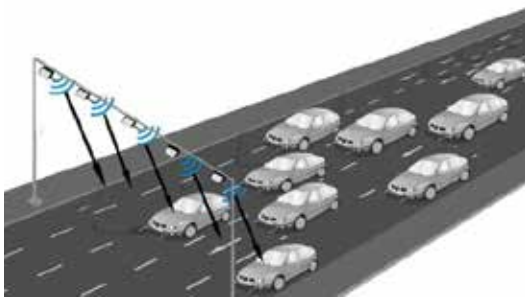


Fig. 2: Integrated traffic data gathering setup with ultrasonic sensors.

These devices count the inflow and outflow of vehicles on a road sub-sections and compute the road subsection occupancy as the difference between the inflow count and outflow count. However, for occupancy values below jam density/occupancy, flow-rate is high and the occupancy values reported by the sensors lag behind the actual value due to transmission delays as a few more vehicles join the

road segment over the transmission period. On the contrary, for occupancy values above jam density, flow-rate is very slow, and the sensors count vehicles which are only partly within the segment as vehicles within the road segment. These vehicles, however, take quite some time actually to join the road segment in full. The instantaneous occupancies, in this case, are, therefore, likely to be different from the actual occupancy.

Fine tuning the instantaneous road occupancy data. To refine the road occupancy data, we model the traffic flow on vehicular networks as a classic Proportional Integral Derivative (PID) controlled system considering the current vehicular flowrate (inflow and outflow), road capacity, threshold occupancy and the current occupancy level of the road as factors for computing the likelihood degree of congestion.

The modeled PID controller (Fig. 3) computes and refines an error value as the deviation of the measured instantaneous occupancy value from a threshold value that marks the onset of traffic congestion for any particular road subsection.

In summary, the PID control system as shown in Fig 3(b), receives instantaneous occupancy ( $dt$ ) and the threshold occupancy ( $d_{max}$ ) as input and computes the deviation ( $dt - d_{max}$ ) of the current occupancy value from the threshold

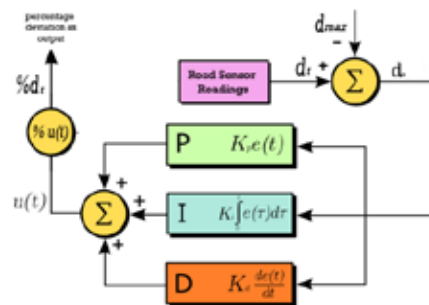


Fig. 3: The PID compensator.

value as the error ( $d$ ). This deviation (error) value(s) is used by the Proportional Congestion Deviation unit (Pcd), Derivative Congestion Deviation unit(Dcd) and Integral Congestion Deviation unit(Icd) units to compute the proportional, derivative and integral deviations of the measured occupancy value. A summation of the respective deviations mentioned above is a deviation value (U(t)) that factors in both deviation history and a predictive measure of deviation.

*Proportional Congestion Deviation (PCD)*

In the PCD unit, the occupancy deviation ( $d$ ) is scaled using a Proportional gain ( $K_p$ ) whose value range from 0 – 1. The value of  $K_p$  for any road sub section is carefully chosen to either intensify or relax the  $d$  value based on the relative impact of traffic congestion of that road subsection on other road sub sections. Thus proportional congestion deviation (Pcd) is computed as shown in [1]:

$$Pcd\ output = K_p * error(t) \dots\dots [1]$$

The “error” is the occupancy deviation value  $d$ .

*Derivative Congestion Deviation (DCD)*

The DCD unit computes the rate of change of deviation of the  $d$  value from the threshold value as the derivative deviation and is scaled using a Derivative gain ( $K_d$ ). In our case, the derivative deviation is the difference between the two preceding deviations, i.e.,  $d'(t)$  and  $d(t-1)$  where  $d(t)$  is the current computed deviation. The derivative congestion deviation (Dcd) is computed as shown in [2]:

$$Dcd\ output = K_d * \frac{d}{dt}(d) \dots\dots [2]$$

The derivative deviation is a predictive measure being used to predict future congestion

because we know the slope (the rate of change of deviation).

*Integral Congestion Deviation (ICD)*

The ICD unit accumulates instantaneous past errors over a period. It factors in both the magnitude and duration of congestion deviation into the value of the generated metric giving the controller an accumulated offset of deviation that should have been corrected in the past. The Integral Congestion Deviation (ICD) is thus computed as shown in [3]:

$$Icd\ output = K_i * \int_0^t d(\tau) d\tau \dots\dots [3]$$

The constant  $K_i$  can be seen as the forgetful factor which is kept as a fraction of a whole. The final output of the PID model is a summation of the proportional, integral and derivative deviations from the desired congestion value as shown in [4]:

$$U(t) = K_p * error(t) + K_p * \frac{d}{dt}(error(t)) + K_i * \int_0^t error(\tau) d\tau \dots [4]$$

The aberrancy in the instantaneous occupancy value is handled by subtracting the PID output from the reported instantaneous occupancy as a conciliatory value for transmission delays and vehicular behavior at different occupancy levels. The refined occupancy value is thus computed as:

$$d'_i = d_i - U(t) \dots\dots [5]$$

where  $d'_i$  = refined occupancy,  $d_i$  = instantaneous occupancy and  $U(t)$  = PID Output.

*Speed - Occupancy Relationship*

By Greenshields’s postulate, 1934 of a linear relationship between speed and occupancy, the relationship between speed and occupancy  $v = f(d)$ , can be formulated as observed values of occupancy and speed as shown in [6]:

$$v_i = f(d_i) + e_i \dots\dots [6]$$

Where  $e_i$  is an error term for the estimated speed. The error term introduced into this relationship is to make up for the dynamic and rapid changes in occupancy values over time.

An alternative view is that the traffic assumes a certain density and hence occupancy depending on the prevailing speed. That leads to a statistical model formulation in [7]:

$$d_i = g(v_i) + \epsilon_i \dots\dots [7]$$

Where  $\epsilon_i$  is an error term for the estimated occupancy. In this case, the form of the function  $g()$  is inverse to that of  $f()$ .

A detailed version of Greenfield's postulate of a linear relationship in line with [6] is mathematically stated in [8]:

$$f(d) = v_0 (1 - d/d_j) \dots\dots [8]$$

$d$  = density/occupancy (measured in vehicles per length of road);  $v_0$  = free-flow speed; and  $D_j$  = jam density

Though this linear model is simple and useful, other non-linear models in work (Kornhauser, 2005) have shown better results. Among these includes Greenburg's model (1959) of a relationship between speed and occupancy which was intended for use at high densities. This model is stated mathematically in [9]:

$$f(d) = v_m \log_e (d_j/d) \dots\dots [9]$$

where  $d$  = density/occupancy (measured in vehicles per length of road);  $v_m$  = free-flow speed and  $D_j$  = jam density.

Results from this model was proven (Kornhauser, 2005; Heydecker & Addison, 2010) to be better than the linear model but this model

will not fit our purpose of computing travel time from this relationship without imposing certain constraints especially when dealing with lower occupancy values because the speed, in theory, approaches infinity as the occupancy approaches zero. Underwood's (1961) modeled a relationship between speed and density but was this model was intended primarily for use in free-flow conditions. Underwood's model is expressed as an exponential decay curve mathematically shown below and is most effective for non-congested traffic:

$$f(d) = v_0 \exp(-d/d_j) \dots\dots [10]$$

Where  $d_j$  is the jam occupancy. This model has an explicit free-flow speed  $v_0$ , but there is no value of occupancy that will cause the speed to be zero, implying traffic can travel at a non-zero speed even when it is at jam density. The Eddie model(1961) employs two equations; an exponential equation (similar to Underwood's model) used when occupancy values are below the threshold value that marks the onset of congestion and a logarithmic equation(similar to Greenburg's model) used for occupancy values greater than the threshold occupancy value. These two equations are as follows:

$$\text{Speed } (v) = V_m * \exp(-d/d_j), \text{ for } d \leq d_t \dots\dots [11]$$

$$\text{Speed } (v) = V_m * \ln (d_j/d), \text{ for } d > d_t \dots\dots [12]$$

Where  $d_j$  is the jam density which in this case is same as the capacity of the road segment and  $d_t$  is threshold density, which refers to the occupancy or density value that marks the onset of congestion.

Throughout our work, the occupancy values used are the refined instantaneous occupancies measured as explained earlier. Thus, if the refined occupancy values accurately compensate for the errors due to transmission

delays then the modeled speed values will need no error term reducing [6] to:

$$v_i = f(d_i') \dots [13]$$

where  $d_i'$  is the adjusted instantaneous occupancy

We adopt and refine [11] and [12] for this work as it is most useful for situations of occupancy values being either above or below a threshold value. With the adopted equations having no way of compensating for the error due to transmission delays, we achieve this compensation by replacing the occupancy value “ $d$ ” with our refined occupancy “ $d_i'$ ”. Therefore, our adopted and refined model can be stated as in [14] and [15].

$$\text{Speed}(v) = V_0 * \exp(-d_i'/d_j), \text{ for } d_i' \leq d_t \dots [14]$$

$$\text{Speed}(v) = V_0 * \ln(d_j/d_i'), \text{ for } d_i' > d_t \dots [15]$$

### Speed – Travel Time Conversion

The fundamental relationship between speed, distance and time are used to estimate the travel time on a road link. The travel time is computed from the predicted speed using the relation:

$$\text{Travel Time } \alpha = \frac{\text{distance}}{\text{Speed}}$$

$$\Rightarrow \text{Travel Time} = \begin{cases} \frac{\text{distance}}{v_0 * \exp(-d_i'/d_j)} & \text{for } d_i' \leq d_t \\ \frac{\text{distance}}{v_0 * \ln(d_j/d_i')} & \text{for } d_i' > d_t \end{cases} \dots [16]$$

The computed travel time is used as a routing metric for the SPF algorithm.

### Traffic Decongestion Routing For Vehicular Networks

To route vehicles along least congested sections of the vehicular network, we modeled vehicular routing as the shortest path problem with constraints on computational metrics like

traffic congestion and travel time.

In computing the least congested path for a journey, we treat the vehicular network as a directed graph  $G(V,E)$ . Vertices  $v \in V$  represents all junctions or nodes on the vehicular network and Edges  $e \in E$  represents connections between these junctions or nodes. The length, speed limit and the reported congestion metric (from ultrasonic sensor devices mounted on each edge) for each edge are used to compute the cost of traversing that edge  $c(E)$  in terms of estimated travel time.

For each request of a departure-destination pair route, a Java web service dynamically generates a sub-map of all possible routes for the selected departure-destination nodes.

The SPF algorithm is then executed over the generated map using the estimated travel time on each edge as the routing metric of computation. The best path  $P$  is a sequence of  $n$  vertices ( $v_1, v_2, v_3 \dots, v_n$ ) returned by the algorithm after computation.

The Path Cost  $C(p)$  is a summation of the estimated travel time for each edge on the best path where travel time for each edge is the time required by a vehicle travelling at the maximum possible speed within the speed limits of the edge at a point in time subject to real-time road occupancy.

### Implementation (Kwanso Mobile App)

The mobile app allows a user to request for the least congested path from a departure node to a destination node. For each request, the application connects to an oracle database through a Java Enterprise web service which dynamically generates a sub-map of all possible routes for the selected departure and destination nodes. The application then computes the Congestion Aware Routing algorithm over the generated map using the traffic congestion metric

values dynamically fed to the oracle database from a superimposed data network. The computed least congested route from the selected departure to destination is then presented to the user in two forms:

- i. A directional map of the route superimposed on Google Maps
- ii. A list of nodes/junctions to traverse with journey details (estimated travel time, journey distance, estimated fuel consumption)

## Results

### Simulations

We confirm the behavior of our routing metric by simulating varying occupancy values for a 5km stretch of road segment with a capacity of 1200 vehicles and a congestion threshold value of 1000. Find below a summary of the simulated data captured within five minutes interval for our study. Table 1 simulates a PID controller with a Proportional gain of 0.0009 a derivative gain of 10 and an integral gain of 0.0005.

TABLE 1  
PID Simulation for occupancy values (Full table attached as an appendix)

$di$	$d$	$\frac{d}{dt}(d)$	$\int_0^t d(\tau)d\tau$	$PID$ $Out$	$di'$	$PID$ Occupancy
3	-997	0	0	-0.8973	3	3.8973
4	-996	-1	-997	-2.3949	5	6.3949
8	-992	-4	-1993	-5.8893	12	13.8893
10	-990	-2	-2985	-4.3835	12	14.3835
...	...	...	...	...	...	...
891	-109	6	-489	5.6574	885	885.3426
877	-123	14	-504	13.637	863	863.3627

Table 2 simulates the occupancy-speed-travel time relationship using the refined occupancy values from Table 1. The portion of the study

data above reveals a steady rise of travel time as occupancy increases from zero to some value just below the threshold of congestion onset.



TABLE 2  
Simulation of occupancy-speed-travel time relationship for  
Adjusted Occupancies below the Threshold Value

$di'$	$-di' / dj$	$\exp(-di' / dj)$	$V_0 * \exp(-di' / dj)$	Travel Time
3	-0.003	0.998	79.800	0.0627
5	-4166667.000	0.996	79.667	0.0628
12	-0.010	0.990	79.204	0.0631
...	...	...	...	...
934	-778333333.000	0.459	36.734	0.1361
935	-779166667.000	0.459	36.703	0.1362
936	-0.780	0.458	36.672	0.1363
943	-785833333.000	0.456	36.459	0.1371
944	-786666667.000	0.455	36.429	0.1373
956	-796666667.000	0.451	36.066	0.1386
965	-804166667.000	0.447	35.797	0.1397
970	-808333333.000	0.446	35.648	0.1403
985	-820833333.000	0.440	35.205	0.1420
990	-0.825	0.438	35.059	0.1426
999	-0.833	0.435	34.797	0.1437

### *Discontinuity*

The remaining section of the simulation data in Table 3 reveals the exponential increase in travel time as the occupancy grows beyond the

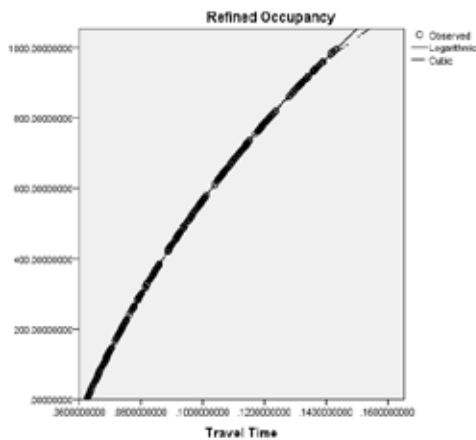
threshold value of 1000 which marks the onset of congestion in accordance with the Eddie model of speed-density relation.

TABLE 3  
Occupancy-Speed-travel Time for Adjusted Occupancies above the Threshold Value.

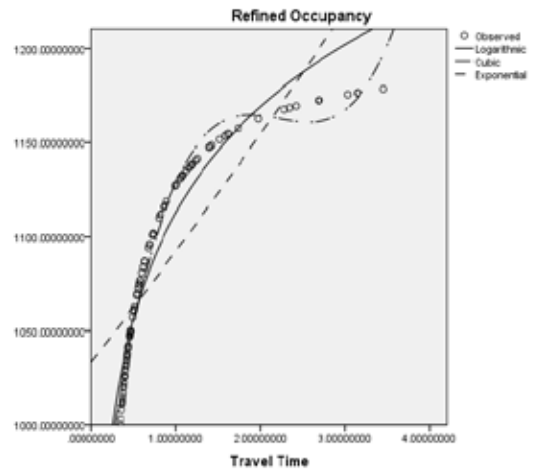
$di'$	$dj/ di'$	$\ln (dj/ di')$	$V_m * \ln (dj/ di')$	$TravelTime$
1000.165	1.200	0.182	14.572	0.3431
1003.119	1.196	0.179	14.337	0.3488
1008.170	1.190	0.174	13.935	0.3588
1011.090	1.187	0.171	13.703	0.3649
1012.014	1.186	0.170	13.630	0.3668
1013.005	1.185	0.169	13.552	0.3689
1015.030	1.182	0.167	13.392	0.3733
1017.022	1.180	0.165	13.235	0.3778
1020.048	1.176	0.162	12.998	0.3847
1022.004	1.174	0.161	12.844	0.3893
1025.031	1.171	0.158	12.608	0.3966
1029.014	1.166	0.154	12.298	0.4066
1030.975	1.164	0.152	12.145	0.4117
1031.936	1.163	0.151	12.071	0.4142
1033.964	1.161	0.149	11.914	0.4197
1035.954	1.158	0.147	11.760	0.4252
1036.917	1.157	0.146	11.686	0.4279
...	...	...	...	...
1181	1.016	0.016	1.277	3.9160
1200	1.000	0	0	$\infty$
1221	0.983	-0.017	-1.388	-3.6026

Our simulated data from Table 3 reveal an abnormal/infinite travel time if the road segment network is allowed to hit its full capacity (1200). That confirms and coincides with the

point of congestion collapse described by Todd Littman (Litman, 2001). Analyzing the simulation data with the SPSS analytic tool results in the following graphs:



Occupancy Vs Travel-Time before threshold Occupancy



Occupancy Vs Travel-Time before threshold Occupancy

**Model Summary and Parameter Estimates**

Dependent Variable: Refined\_Occupancy

Equation	Model Summary					Parameter Estimates			
	R Square	F	df1	df2	Sig.	Constant	b1	b2	b3
Linear	.987	34842.070	1	443	.000	32.036	.994		
Quadratic	.987	17384.972	2	442	.000	30.806	1.000	-4.929E-6	
Cubic	.987	11564.386	3	441	.000	31.612	.992	1.169E-5	-9.199E-9

The independent variable is Instantaneous\_Occupancy.

Fig. 4: SPSS analytics on Instantaneous occupancy VRS Refined occupancy.

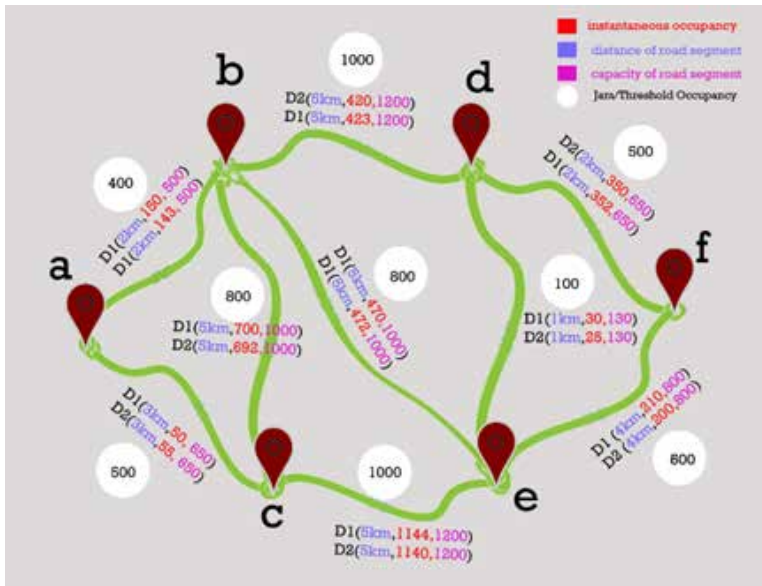


Fig.5: Example Road Network for Illustration

Using the dummy road map shown in Fig. 4, find in Table 4 the derivative outputs computed for the various road links of the road map and in

Table 5&6, the estimated travel times modeled from the measured occupancy values assuming a speed limit of 80km/h for all the links.

TABLE 4  
Occupancy Fine-tuning for Routes

Occupancy	Differential error	Dout [ $d(e)/dt$ * capture interval]	Derivative Occupancy
NODE A-B			
143	0	0	143
150	-7	-7	157
NODE A-C			
50	0	0	50
55	-5	-5	60
NODE B-C			
700	0	0	700
692	8	8	684
NODE B-D			
423	0	0	423
420	3	3	417
NODE B-E			
472	0	0	472
470	2	2	468
NODE C-E			
1144	0	0	1144
1140	4	4	1136
NODE D-E			
30	0	0	30
25	5	5	20
NODE D-F			
352	0	0	352
350	2	2	348
NODE E-F			
210	0	0	210
200	10	10	190

Note that the implementation delineated here does not apply the proportional and integral deviation values. It only considers and corrects the derivative errors over the transmission period.

TABLE 5  
Travel time computation from occupancy values below the threshold density

Road Link	OCCUPAN CY	(-Occupancy/ Capacity)	EXP (- Occupancy/ Capacity)	SpeedLimit * EXP (-Occupancy/ Capacity)	Travel Time
A- B	157	-0.3925	0.675366	54.02930771	0.03701
A- C	60	-0.12	0.8869	70.953	0.0423
B- C	684	-0.855	0.4253	34.022	0.1469
B- D	417	-0.417	0.659	52.721	0.0948
B- E	468	-0.585	0.5571	44.568	0.1121
D- E	20	-0.2	0.8187	65.498	0.0152
D- F	348	-0.696	0.49856	39.886	0.050143
E- F	190	-0.3166	0.7286	58.2859	0.068627

TABLE 6  
Travel time computation from occupancy values above the threshold density

Road Link	OCCUPANCY	(Occupancy/ Capacity)	ln (-Occupancy/ Capacity)	SpeedLimit * ln (- Occupancy/Capacity)	Travel Time
C- E	1136	1.0563	0.05481	4.385	1.140

The computed travel times then become a routing metric for the SPF routing algorithm used in this work. Find in Fig.5. A reconstruction

of the road map replacing congestion situations with estimated travel times computed for use in the proposed algorithm for vehicular routing.

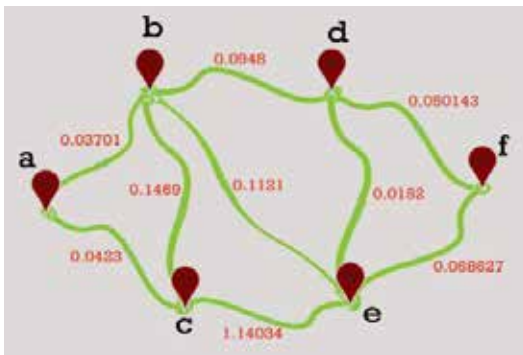


Fig. 6: Road Network with Estimated Travel Time Metrics for routes

Using the brute force method, the travel times for all possible paths in from node 'a' to node 'f' is as shown in Table 15 indicate the best route to be {a, b, d, f} with a travel time of 0.181953 units.

TABLE 7  
Possible routes with associated route cost in minutes

PATH	Route Cost (Travel Time in minutes)
{a, b, c, e, f}	1.392877
{a, b, d, f}	0.181953
{a, b, e, f}	0.217737
{a, b, d, e, f}	0.215637
{a, b, c, e, d, f}	1.389593
{a, b, e, d, f}	0.214453
{a, c, e, f}	1.251267
{a, c, b, e, f}	0.369927
{a, c, b, d, f}	0.334143
{a, c, b, d, e, f}	0.367827

The iterations through the algorithm used in our proposed routing system for the road network under discussion is as detailed in the Table 8.

TABLE 8  
Least travel time route computation from Node A to F

S	L(a)	L(b)	L(c)	L(d)	L(e)	L(f)
a	∞	∞	∞	∞	∞	∞
{a}		a	a	∞	∞	∞
		<b>0.03701</b>	0.0423			
		a	a	b	b	
{b}		<b>0.03701</b>	<b>0.0423</b>	0.13181	0.14911	∞
			<del>B:0.1839</del>		b	
		a	a	b	0.14911	∞
{c}		<b>0.03701</b>	<b>0.0423</b>	<b>0.13181</b>	<del>C:1.1826</del>	
		a	a	b	d	d
{d}		<b>0.03701</b>	<b>0.0423</b>	<b>0.13181</b>	<b>0.14701</b>	0.1819
		a	a	b	d	d
{e}		<b>0.03701</b>	<b>0.0423</b>	<b>0.13181</b>	<b>0.14701</b>	<b>0.181953</b>
						<del>E:0.2156</del>
{f}						

After the iteration it is found that the least cost path in terms of travel time from node 'a' to node 'f' is by routing through {a, b, d, f} and has a path cost of 0.181953-time units. This

path is confirmed by the brute force tables and is a proof that the algorithm used can compute the least cost path in terms of travel time.

### Discussion

There is STAR (Guidici & Pagani, 2005) solution in the literature which exploits both street topology information achieved from geographic information systems and information about the spatial distribution of vehicles along the street and vehicular traffic, to perform routing decisions.

The work by Guidici & Pagani (2005) suggest that vehicles be equipped with network interface cards enabling them to partake in a vehicular network infrastructure so these vehicles can share traffic data with each other. This solution however fails to distinctively provide a metric capable of being used by any algorithm to predict the cost of a journey. The solution leaves this to the intuition of the driver to decide on which road segment will be best to travel along given the traffic congestions on the various road links. In our case, however, data captured is not delivered to road users as raw traffic status data but converted into a travel time cost and fused into an algorithm (in our prototype mobile app) making the solution more useful. The STAR solution also depend directly on vehicles actively connected to the road network requiring either all vehicles to have active Network Interface Cards to yield accurate results. Finally, the solution is limited by the coverage area of the wireless technology used by the vehicles. Our solution overcomes these limitations as the detectors that capture the occupancy values are not attached to any particular vehicle but rather fitted to the road network system. The “KWANSO” mobile application was designed as prototype mobile application that minimizes traffic congestion on the Liberation Road in Accra making use of the modeled routing metric and algorithms presented in this study. The mobile app allows a

user to request for the least congested path from a departure node to a destination node. For each request made, a computed best route from the selected departure to the destination is then presented to the user in two forms:

- i. A directional map of the route
- ii. A list of nodes/junctions to traverse with journey details (estimated travel time, journey distance, estimated fuel consumption).

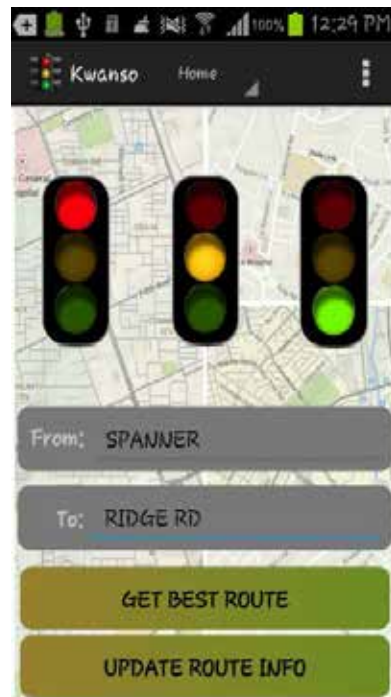


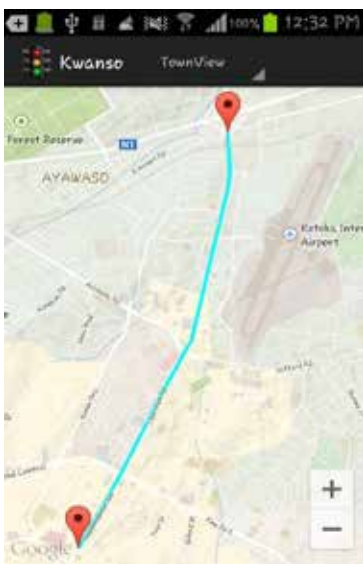
Fig. 6: Rout Request Nodes



Fig.7: Least Cost Path



Fig. 9: Least Congested Path - Satellite View

Fig. 8:  
Default Path- Satellite View

## Conclusion

This study has proposed a strategy to address the problem of decongesting vehicular traffic on road networks using transport telematics to compute the instantaneous least cost path with regards to congestion state and travel time for departure-destination nodes (junctions) of a journey.

The proposed strategy identified the Dijkstra's shortest path first algorithm as suitable for adoption and re-engineering to device a fair routing of vehicular routing to manage congestion on road networks. The adopted algorithm has been re-engineered to use estimated travel time as the only metric. Travel Time is computed based on occupancy-speed relationship and speed/distance/travel-time dependencies that called for the adjustment of instantaneous occupancy for error. Instantaneous occupancy data is acquired from a sensor network superimposed on the target road



network. Adjusting the instantaneous occupancy has been effected by estimating the error based on the classical PID controller model. Based on the re-engineered Dijkstra's shortest path first algorithm, a method of congestion-aware routing of vehicular traffic has been formulated. The method assumes the existence of sensor network that feeds road occupancy data into a dynamic database. From the data, the time metric is computed for each road segment. The re-engineered algorithm has been analyzed to a time complexity of  $n^2 + m$  with  $n$  representing the total number of nodes and  $m$  representing the total number of edges. The method has been implemented as a server-assisted mobile application and demonstrated. Also the method has been evaluated against a similar research work reported in the literature. The proposed method proves to be easier to use by its server-assisted nature. The mobile application makes use of processed data that is meticulously adjusted for reliability through a non-mobile sensor network. Coverage of the network can be made as wide as possible and constant. Overall, it can be stated that the objectives of the research work has been achieved. In the future, we envision to analyze other data network algorithms that could be re-engineered to solve the same problem. We intend to prototype and compare the performance of multiple re-engineered algorithms and select the best for our final implementation. In a future work, we are looking forward to a study on a telematics approach to computing the relative congestion contribution of each road sub-section to the overall traffic congestion for a chosen road segment. This study will help refine values for proportional, derivative and integral gains used in this study resulting in a more tailor-made decongestion solution to the problem for different localities.

## References

- ALESIANI, F. & NITIN, M. (2014) Optimization of Charging Stop for Fleet of Electric Vehicles: A Generic Approach. Intelligent transportation Group. NEC Europe Labs, NEC Europe Ltd, Heidelberg, Germany. IEEE Intelligent Transportation Systems Magazine Volume 6 Number 3.
- BHATTACHARYYA, S. & DIOT, C. (2012) On the Impact of Aggregation on the Performance of Traffic Aware Routing; A. Sridharany, U. Pennsylvania Philadelphia, PA. Sprint Labs Burlingame, CA. R. Guerin U. Pennsylvania Philadelphia, PA. J. Jetchevaz, N. Taf Sprint Labs Burlingame, CA.
- EDWARD, L. (2017) Smarter Cambridge Transport. Reducing Traffic Congestion and Pollution in Urban Areas.
- GIUDICI, F. & PAGANI, E. (2005) Spatial and Traffic-Aware Routing (STAR) for Vehicular Systems. Information and Communication Department Universit'a degli Studi di Milano, Italy e-mail: {fgiudici, pagani}@dico.unimi.it
- HEYDECKER, B. G. & ADDISON, J. D. (2010) Analysis and modelling of traffic flow under variable speed limits. Centre for Transport Studies, University College London, Gower Street, London WC1E 6BT, UK.
- HEYDECKER, B. G. & ADDISON, J. D. (2011) Analysis and modelling of traffic flow under variable speed limits. Centre for Transport Studies, University College London, Gower Street, London WC1E 6BT, UK.
- JACK, W. (2006) "The Role of Transportation in the U.S. Economy," Chief Economist, U.S. Department of Transportation. Presentation to the National Surface Transportation Policy and Revenue Study Commission on June 26, 2006.
- KATABI, D. HANDLEY, M. ROHRS, C. (2002) "Internet Congestion Control for High Bandwidth-Delay Product Networks." ACM Sigcomm, Pittsburgh.

- KORNHAUSER, A. L. (2005) Traffic Congestion: How Predictable? Discovering Volume Trends across Time and Confirming Fundamental Speed-Flow-Density Relations – Megan Lynn Bernard Princeton University.
- LARMILEY, B. (2013) Daily Graphic reporting on statistics by DVLA and Metropolitan Roads Department.
- LITMAN, T. (2001) Generated traffic: implications for transport planning. *ITE Journal* 71.
- MAHAMA, F. (2011) Industrial mathematics – KNUST. Study of vehicular traffic congestion in the Sekondi-Takoradi metropolis.
- WEE, V. B., ANNEMA, J. A. & BANISTER, D. (2013) The Transport System and Transport Policy by, Chapter 7 - Traffic flow theory and modelling.
- SALVADORI, E., BATTITI, R. & SABEL, M. (2002) A Reactive Scheme for Traffic Engineering in MPLS Networks. Universita' di Trento, Dipartimento di Informatica e Telecomunicazioni via Sommarive 14, 38050 Povo (TN), Italy.

Received 23 Jul 18; revised 24 Oct 18.