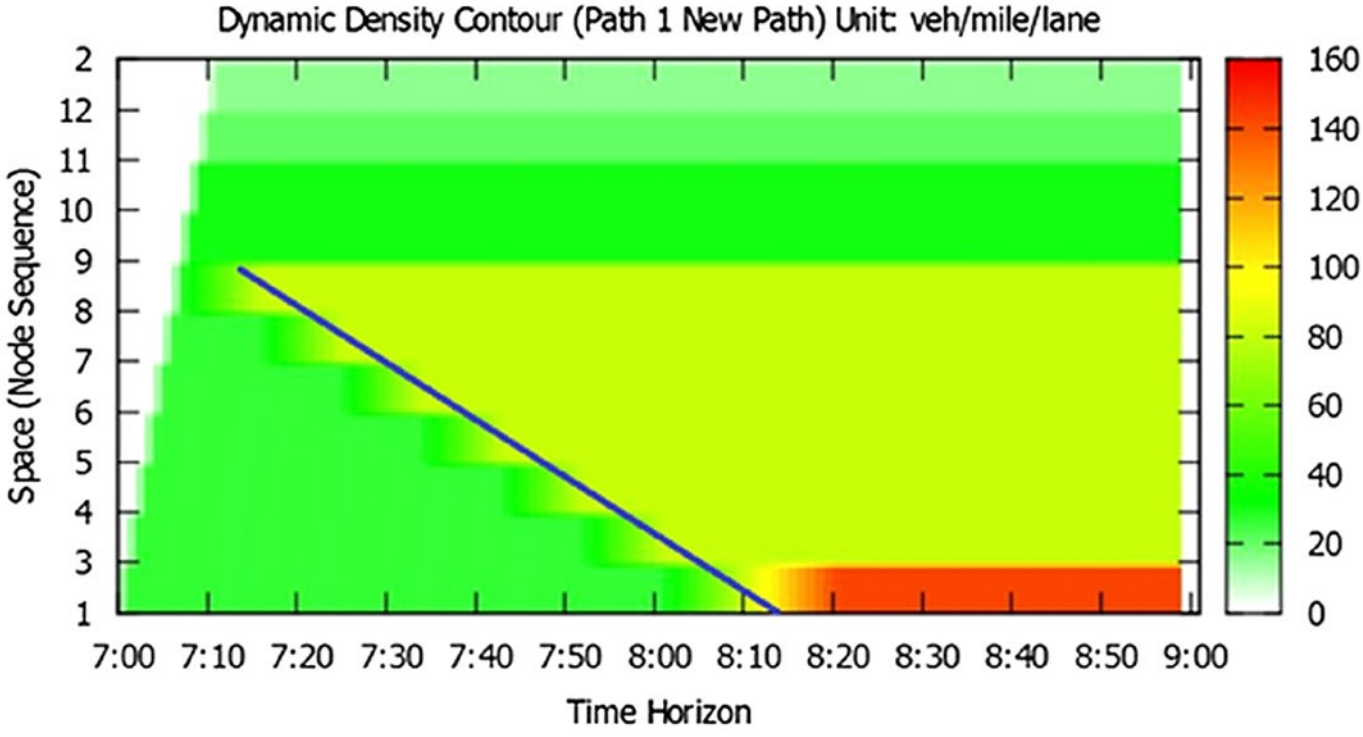
cogent engineering



CIVIL & ENVIRONMENTAL ENGINEERING | SOFTWARE

DTALite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration

Xuesong Zhou and Jeffrey Taylor

Cogent Engineering (2014), 1: 961345

cogent oa

Zhou & Taylor, *Cogent Engineering* (2014), 1: 961345
http://dx.doi.org/10.1080/23311916.2014.961345

cogent engineering

CIVIL & ENVIRONMENTAL ENGINEERING | SOFTWARE

DTALite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration

Xuesong Zhou[1*] and Jeffrey Taylor[2]

Received: 08 May 2014
Accepted: 26 August 2014
Published: 01 October 2014

*Corresponding author: Xuesong Zhou, School of Sustainable Engineering and the Built Environment, Arizona State University, Tempe, AZ 85287, USA.
E-mail: xzhou74@asu.edu; xzhou99@gmail.com

Reviewing editor:
Filippo Pratico, University Mediterranea of Reggio Calabria, Italy

Additional article information is available at the end of the article

Abstract: A number of emerging dynamic traffic analysis applications, such as regional or statewide traffic assignment, require a theoretically rigorous and computationally efficient model to describe the propagation and dissipation of system congestion with bottleneck capacity constraints. An open-source light-weight dynamic traffic assignment (DTA) package, namely DTALite, has been developed to allow a rapid utilization of advanced dynamic traffic analysis capabilities. This paper describes its three major modeling components: (1) a light-weight dynamic network loading simulator that embeds Newell's simplified kinematic wave model; (2) a mesoscopic agent-based DTA procedure to incorporate driver's heterogeneity; and (3) an integrated traffic assignment and origin–destination demand calibration system that can iteratively adjust path flow volume and distribution to match the observed traffic counts. A number of real-world test cases are described to demonstrate the effectiveness and performance of the proposed models under different network and data availability conditions.

Subjects: Intelligent & Automated Transport System Technology, Systems Integration, Transport Engineering

Keywords: transportation network modeling, traffic simulation, traffic demand estimation, dynamic traffic assignment

ABOUT THE AUTHORS

Xuesong Zhou is an associate professor in the School of Sustainable Engineering and the Built Environment at Arizona State University. He has been the leading developer for the DTALite/NeXTA packages since he worked as an assistant professor at the University of Utah. His current research interests include analytical and computational modeling of transportation systems. He serves as a subcommittee chair for TRB Committee on Transportation Network Modeling (ADB30), Network Equilibrium Subcommittee. He is a member of IEEE and INFORMS.

Jeffrey Taylor received his BS in Civil and Environmental Engineering from the University of Utah in 2010 and he obtained his master's degree at the University of Utah in 2014. While most of his research work has been devoted to supporting and enhancing the DTALite/NeXTA packages, his research interests also extend to topics in transportation planning/modeling, transportation safety and traffic flow modeling. He holds memberships with IEEE, INFORMS, and ITE.

PUBLIC INTEREST STATEMENT

This paper describes the internal functions of DTALite, an open-source, light-weight dynamic traffic assignment (DTA) software package. DTALite can be used for large-scale transportation modeling applications to help planning/engineering organizations and public officials make transportation infrastructure investment decisions. Particularly important applications may include modeling the traffic impacts of work zones, proposed freeways/highways, and tolling facilities. DTALite's route choice model also allows organizations to test the effects of multiple strategies to improve traffic operations and manage travel demand. One of the highlights of this paper is DTALite's support for multi-threaded processing, which significantly reduces model run-times. This could allow users to evaluate more alternative strategies to solve a problem within a limited amount of time, providing more information to help decision-makers find better solutions to difficult problems.

1. Introduction

Motivated by a wide range of application needs, such as region-wide traffic analysis and route guidance provision, dynamic traffic assignment (DTA) models have been increasingly recognized as an important tool for assessing operational performance of those applications at multiple spatial resolutions (e.g. network, corridor, and individual segment levels). The advances of DTA in this aspect are built upon the capabilities of DTA models in describing the formation, propagation, and dissipation of traffic congestion in a transportation network.

DTALite, an open-source mesoscopic DTA simulation package, in conjunction with the Network eXplorer for Traffic Analysis (NeXTA) graphic user interface, has been developed to provide transportation planners, engineers, and researchers with a theoretically rigorous and computationally efficient traffic network modeling tool. This fully functional, open-source DTA model can be downloaded from http://code.google.com/p/nexta/. In general, the software suite of DTALite + NeXTA aims to:

(1) Provide an open-source code base to enable transportation researchers and software developers to expand its range of capabilities to various traffic management application

(2) Present results to other users by visualizing time-varying traffic flow dynamics and traveler route choice behavior in an integrated 2D/3D environment

(3) Provide a free, educational tool for students to understand the complex decision-making process in transportation planning and optimization processes.

Additionally, DTALite also adopts a new software architecture and algorithm design to facilitate the most efficient use of emergent parallel (multi-core) processing techniques and exploit the unprecedented parallel computing power newly available on both laptops and desktops.

This paper is organized as follows. Section 2 first introduces the overall system design and model structure of DTALite. This is followed by the related literature review and traffic flow modeling implementation in Section 3, and a queue-based traffic state estimation framework in Section 4. The paper closes with a case study using real-world networks (Section 5) and overall conclusions (Section 6).

2. System design and model structure

The software architecture designed in DTALite aims to integrate many rich modeling and visualization capabilities into an open-source DTA model. Using a modularized design, the software suite of simulation engine + visualization interface can also serve future needs by enabling transportation researchers and software developers to continue to build upon and expand its range of capabilities. The streamlined data flow from static traffic assignment models and common signal data interfaces aims to allow planners and engineers to rapidly apply the advanced DTA methodology, and further examines the effectiveness of traffic mobility, reliability, and safety improvement strategies. The overall structure, illustrated in Figure 1, integrates the four major modeling components highlighted in yellow.

DTALite's four major modeling components include:

(1) Time-dependent shortest path finding, based on a node-link network structure

(2) Vehicle/agent attribute generation, which combines an origin–destination (OD) demand matrix with additional time-of-day departure time profile to generate trips

(3) Dynamic path assignment module, which considers major factors affecting agents' route choice or departure time choice behavior, such as (i) different types of traveler information supply strategies (e.g. historical, pre-trip, and/or en route information, and variable message signs) and (ii) road pricing strategies where economic values are converted to generalized travel time

(4) A class of queue-based traffic flow models that can accept essential road capacity reduction or enhancement measures, such as work zones, incidents, and ramp meters.

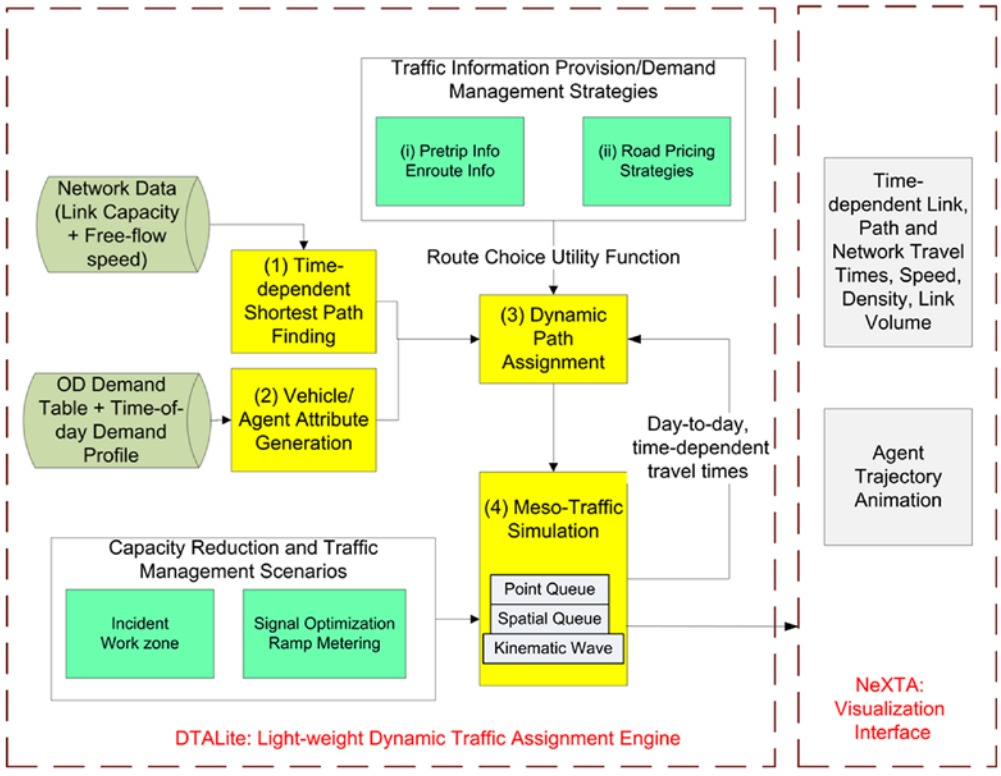


Figure 1. Software system architecture with key modeling components.

The major focus of this paper will be in describing the queue-based traffic simulation model and its application in an integrated origin–destination demand estimation procedure. The queue-based traffic simulation model in DTALite only requires basic link capacity and free-flow speed for operation, which are readily available from static traffic assignment models. Using simple input parameters, in addition to possible connections with common signal data interfaces, the proposed simulation package may enable state DOTs and regional MPOs to rapidly apply advanced DTA methodologies for large-scale regional networks, subareas, or corridors. Additionally, the modularized system design may help serve future needs by simplifying the process for transportation researchers and software developers to continue to build upon and expand its range of capabilities.

3. Queue-based traffic flow simulation model

3.1. Literature review on traffic simulation in a DTA context

In general, a simulation-assignment DTA model needs to read time-dependent OD demand matrices and then assign vehicles to different paths based on link travel time. This procedure is illustrated by the diagram in Figure 2. Given trip demand data and path information, the dynamic network loading (DNL) module in the DTA model then simulates the movement of vehicles through the network. The physical process of moving vehicles through a transportation network works in a two-step process. First, a vehicle is moved across a link by a link traversal model, and then moved between links at the node by a node transfer function. In particular, the link traversal model typically enforces a speed–density relationship and hard outflow capacity constraints, which are determined by link properties (e.g. number of lanes, link type). The node transfer model involves specific left-turn or through movement capacity determined by green time allocation at signalized nodes or other attributes. Link travel times from the traffic simulator are fed to the time-dependent shortest path model for path selection through a specified route choice utility

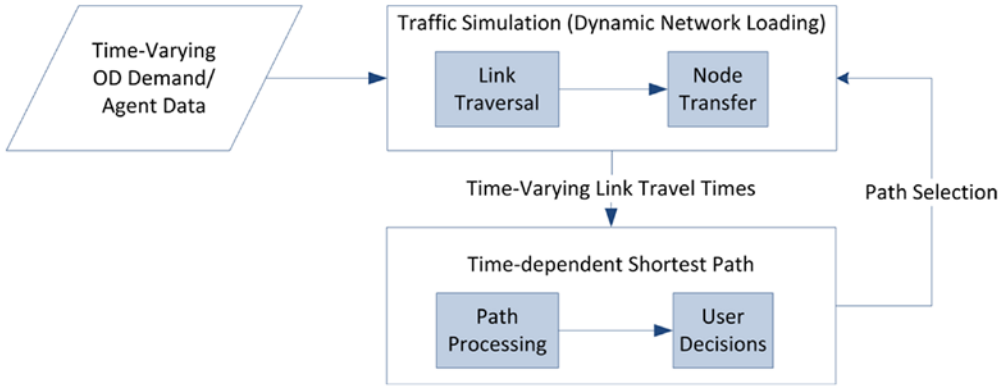


Figure 2. General simulation-based DTA modeling framework.

function or traffic assignment rules. The new paths are then fed back into the traffic simulator in the next iteration.

There are macroscopic, mesoscopic, and microscopic simulation-based methods for generating time-dependent travel time measures in DTA. The macroscopic models use continuum fluid representations, and the microscopic simulation models typically rely on detailed car following and lane changing models. In contrast, a mesoscopic model considers vehicles (with specific origin, destination, departure time, and routes) individually and moves vehicles according to a number of macroscopic traffic flow relations. Speed–density functions, like those used by Mahmassani, Hu, Peeta, and Ziliaskopoulos (1994), Mahmassani (2001), and Ben-Akiva, Bierlaire, Burton, Koutsopoulos, and Mishalani (2002) are commonly found in application.

To offer a computationally efficient traffic state simulator and estimator, this study focuses on how to use a queue-based mesoscopic traffic simulation methodology to capture complex traffic processes in realistic networks. Interested readers are referred to the survey paper by Peeta and Ziliaskopoulos (2001) for different models such as analytical optimization-based and optimal control-based formulations, and Nie, Ma, and Zhang (2008) for a comprehensive discussion on different types of DNL models and their trade-offs in capturing travel time.

3.2. Capacity and traffic flow models

To capture queue formation, spillback, and dissipation through simplified traffic flow models, DTALite uses a number of computationally simple but theoretically sound traffic queuing models (e.g. point queue model, spatial queue, and Newell's kinematic wave model) to track forward and backward wave propagation in its light-weight mesoscopic simulation engine. By doing so, traffic simulation in DTALite only requires a minimal set of traffic flow model parameters, such as outflow, inflow capacity, and storage capacity constraints, which are illustrated in Figure 3.

To capture the queue dynamics at typical bottlenecks (e.g. lane drop, merge, and weaving segments), the classical kinematic wave theory needs to be integrated with (1) flow conservation constraints, (2) traffic flow models that represent speed (or flow) of traffic as a function of density, and (3) partial differential equations. The flow conservation constraints typically follow a hyperbolic

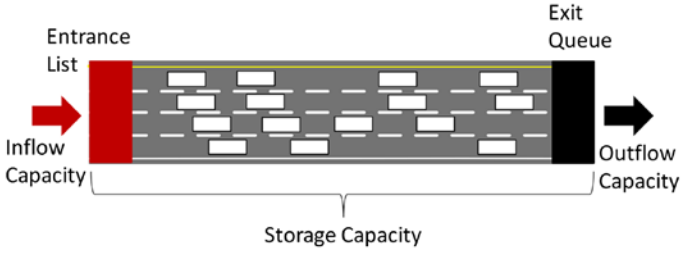


Figure 3. Modeling traffic dynamics through essential outflow, inflow capacity, and storage capacity constraints.

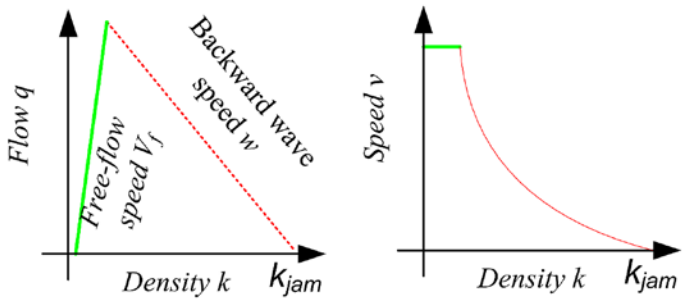


Figure 4. Triangular fundamental diagram.

system of conservation laws $\frac{\partial q}{\partial x}+\frac{\partial k}{\partial t}=g(x,t)$ where q and k are flow and density, respectively, and $g(x, t)$ is the net vehicle generation rate.

Various finite-difference approximation methods have been presented to solve these equations numerically. Based on a triangular flow–density relationship in Figure 4, there are two closely related finite deference-based numerical solution schemes to solve the first-order kinematic wave problem: (1) Newell's simplified model (1993) that keeps track of shock wave and queue propagation using cumulative flow counts on links and (2) Daganzo's cell transmission model (1995) that adopts a "supply–demand" or "sending–receiving" framework to model flow dynamics between discretized cells. DTALite implements Newell's solution, using cumulative flow counts to keep track of traffic entering and leaving each link. These cumulative flow counts are also useful when applied with queueing models.

3.3. Simple point queue and spatial queue models

The most simple queue model implemented in DTALite is the point queue model. By imposing a single outflow capacity constraint on each link, a point queue model aims to capture the effect of traffic congestion at major bottlenecks, although it does not take into account the queue spillback and the resulting delay due to storage capacity. Using a point queue model in the first few iterations and then applying a simplified kinematic wave model in the late assignment process, one can avoid unrealistic and unnecessary gridlock in the initial assignment process, and further allow agents to learn travel times from previous iterations and switch routes to achieve a smooth and close-to-reality traffic pattern.

Figure 5 represents a point queue as a vertical queue with a stack of vehicles, where some of the vehicles are mapped or "rotated" from the physical link (shaded) to the vertical stack queue. The other vehicles on the physical link (not shaded) correspond to the vehicles that move at free-flow speed. In this case, the length of the queue segment in this point-queue model is zero and

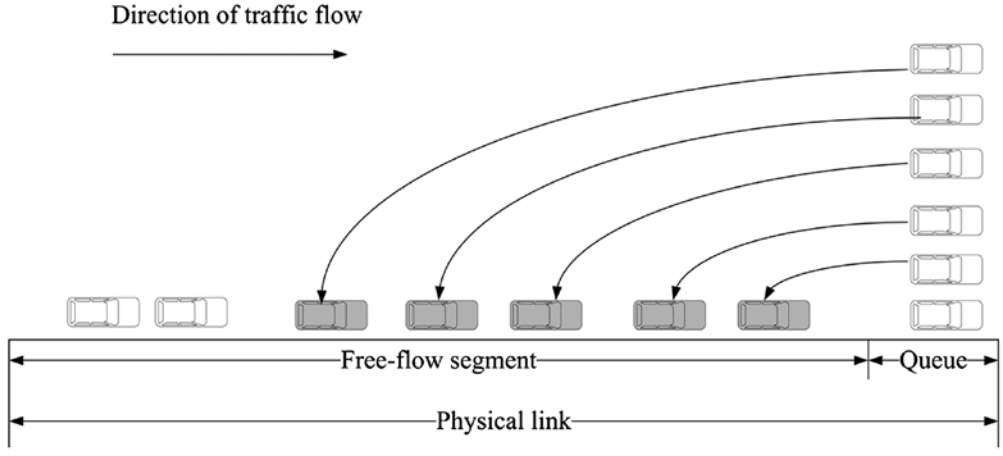


Figure 5. Point queue model represented as a vertical stack queue.

the link has unlimited storage capacity. Interested readers are referred to the paper by Hurdle and Son (2001) to examine the connection between physical spatial queues and vertical stack queues.

The realism of a point queue model can be enhanced by adding spatial storage capacity constraints, so that the resulting spatial queue model can capture queue spillbacks. This is accomplished in DTALite using link-specific jam densities, identifying how many vehicles can be stored on a link when no traffic is moving. Furthermore, by explicitly using the cumulative arrival and departure curves to track kinematic waves, Newell's flow model provides an effective means to realistically represent traffic congestion propagation and capture shockwaves as the result of bottleneck capacities.

Compared to other cell-based models that need to subdivide a long link into segments with short length, Newell's model can handle reasonably long links with homogeneous road capacity. Its simple form of traffic flow models and computational efficiency make it particularly appealing in establishing theoretically sound and practically operational DTA models for large-scale networks. There are a number of related studies on Newell's kinematic model, including model calibration research by Hurdle and Son (2000) and extensions to node merge and diverge cases by Yperman, Logghe, and Immers (2005) and Ni, Leonard, and Williams (2006), to name a few.

3.4. Simple queue-based DNL model

We first present a point-queue-based DNL model, implemented in DTALite, for a sequence of consecutive links of a freeway corridor that cover links $l = 1, 2, \ldots, L$. In the corridor without entrances or exits, traffic of a single OD pair is loaded to the first link $l = 1$ and leaves the network from the last link L.

Notation

N	number of nodes in a corridor
n	index of nodes, $n = 1, 2, \ldots, N$
L	number of links in a corridor
l	index of links, $l = 1, 2, \ldots, L$
Δt	length of simulation interval (e.g. 6 s)
Δx	link length (e.g. 1 miles)
$k_{l,t}$	prevailing density during the tth time step on link l
$q_{l,t}$	transfer flow rate from link l to link $l + 1$ during the tth time interval $[t, t + \Delta t]$
$\text{cap}_{l,t}^{\text{out}}$	outflow capacity on link l during the tth time interval $[t, t + \Delta t]$
v_{free}	free-flow speed
k_{jam}	jam density

3.4.1. Point queue model computational procedure

For time $t = 0$ to T

For link $l = 1$ to L

Step 1: Calculate flow ready to move from link l: $v_{\text{free}} \times k_l(t)$

Step 2: Calculate transfer flow from link l to link $l + 1$

$$q_{l,t} = \text{Min}\{v_{\text{free}} \times k_{l,t}, \text{cap}_{l,t}^{\text{out}}\}$$

Step 3: Update prevailing density at link l

$$k_{l,t+1} = k_{l,t} + q_{l-1,t} - q_{l,t} \times \Delta t / \Delta x$$

End for //link

End for //time

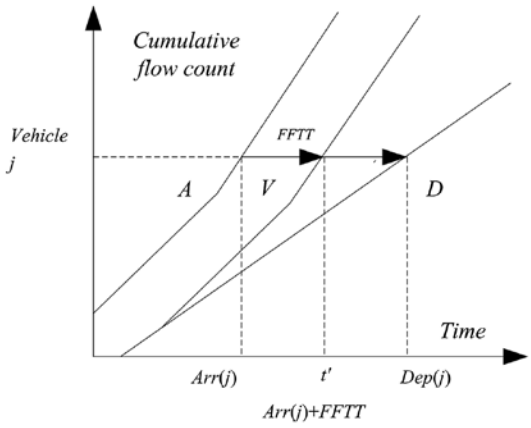


Figure 6. Event-based simulation process; A, D: cumulative arrival and departure flow counts; V: cumulative flow count of the vertical queue; FFTT is free-flow travel time of the link.

Within a mesoscopic simulation framework, the above fluid-based numerical computation scheme can be viewed as a pseudo event-based simulation process, as we do not simulate how a vehicle moves inside the link (at free-flow speed). As shown in Figure 6, at each simulation step, vehicle j is moved into a link and sets its arrival time at the upstream node of the link as Arr(j). Accordingly, the time entering the vertical queue at the stop bar is determined by Arr(j) + FFTT. As the simulation clock advances to t' = Arr(j) + FFTT, we need to check if the outflow link capacity $\text{cap}_{l,t'}^{\text{out}}$ is still available and vehicle j is in the beginning of the vertical queue. If the vehicle is at the beginning of the queue, this vehicle moves to the next link and its departure time stamp Dep(j) = t'. Otherwise, the vehicle must stay in the queue waiting for the available outflow capacity, which results in a departure time Dep(j) > t'. Thus, the entire link travel time of the vehicle is Dep(j) − Arr(j), which can be further used to infer space-mean speed of the link.

Similarly, the numerical calculation scheme of the spatial queue can be enhanced by changing Step 2 by including an additional storage capacity constraint as follows:

$$q_{l,t} = \text{Min}\left\{ v_{\text{free}} \times k_{l,t}, \text{cap}_{l,t}^{\text{out}}, \left(k_{\text{jam}} - k_{l+1,t}\right) \times \Delta x \right\}$$

where $(k_{\text{jam}} - k_{l+1,t}) \times \Delta x$ is the physical space availability at downstream link l + 1.

As the proposed model strictly satisfies the first-in-first-out (FIFO) constraint, if a vehicle in the beginning of the vertical queue is blocked due to unavailable outflow capacity (in the case of spatial queue or kinematic wave model), then the vehicles arriving later in the queue will be blocked as well. To model complex geometric features such as short left-turn bays on a multi-lane facility, one needs to decompose a link into multiple connected cells with each cell satisfying FIFO constraints (Reynolds, Zhou, Rouphail, & Li, 2010).

3.5. Modeling forward and backward wave propagation using Newell's simplified kinematic wave model

To describe traffic congestion propagation realistically by including the phenomena of queue build-up, spillback, and dissipation along freeway corridors, DTALite incorporates Newell's (1993) simplified kinematic wave model, which is built upon the assumption of a triangular flow–density relationship. Newell's model is implemented using cumulative flow counts on each link. As illustrated in Figure 7, $A(t)$ represents the cumulative number of vehicles moving into a link, $D(t)$ represents the cumulative number of vehicles leaving a link, and $V(t)$ is the cumulative number of vehicles in the vertical queue, waiting to exit the link.

Hurdle and Son's framework (2000) is adopted in this research to explain how Newell's model is able to model forward and backward waves using the cumulative flow counts. Let x be the location

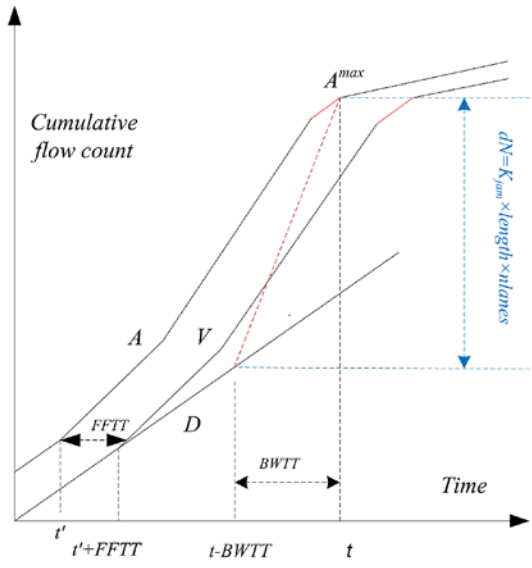


Figure 7. Illustration of cumulative arrival and departure curves *A*(*t*) and *D*(*t*), and the shifted arrival curve *V*(*t*).

along the corridor, and $N(x, t)$ is the cumulative flow count at location x and time t of a link. The change of $N(x, t)$ along a characteristic line (wave) is represented as follows:

$$dN(x,t)=\frac{\partial N}{\partial x}dx+\frac{\partial N}{\partial t}dt=qdt-kdx \tag{1}$$

A wave represents the propagation of a change in flow and density along the roadway, and the wave speed is the slope of the characteristics line $w=\frac{\partial q}{\partial k}=\frac{dx}{dt}$. Along the movement of a wave, we substitute $dt=\frac{dx}{w}$ into the above equation, so that we can link the difference of cumulative flow counts together through

$$dN(x,t)=qdt-kdx=\left(-k+\frac{q}{w}\right)dx \tag{2}$$

For the triangular-shaped flow–density relationship with constant forward and backward wave speeds, it is easy to verify that, when the speed of the forward wave is v_f, the general cumulative flow count updating formula reduces to $-k+\frac{q}{v_f}=-k+k=0$. Under congested traffic conditions with a constant backward wave speed w_b, we have $-k+\frac{q}{w_b}=-k_{\text{jam}}$, and this equation can be rewritten as:

$$dN=\left(-k+\frac{q}{w_b}\right)dx=-k_{\text{jam}}(a)\times\text{length}(a)\times n\text{lanes}(a) \tag{3}$$

The above equation is used to describe how a backward wave travels through the link. Under the queue spillback condition, the difference of $k_{\text{jam}}(a)\times\text{length}(a)\times n\text{lanes}(a)$ between the cumulative arrival and departure counts is illustrated by the vertical dashed line in Figure 7. Now consider a situation with two consecutive links, upstream link a and downstream link b, as shown in Figure 8. Newell's model is concerned about three-state variables on upstream link a: (1) cumulative flow count $A(a, t)$ for vehicles moving into link a through the upstream node; (2) cumulative flow count $V(a, t)$ for vehicles waiting at the vertical queue of the downstream node of link a at time t; and (3) cumulative flow count $D(a, t)$ for vehicles moving out of link a through the downstream node.

As shown in Figure 8, when a queue spills back from the downstream to the upstream, the arrival and departure cumulative flow counts at two ends of a link (at timestamps t and time t-BWTT(a)) need to ensure a constant difference of $dN = k_{\text{jam}}(a)\times\text{length}(a)\times n\text{lanes}(a)$, and the capacity restriction is propagated throughout the link using a time duration of $\text{BWTT}(a) = \text{length}(a)/w_b(a)$.

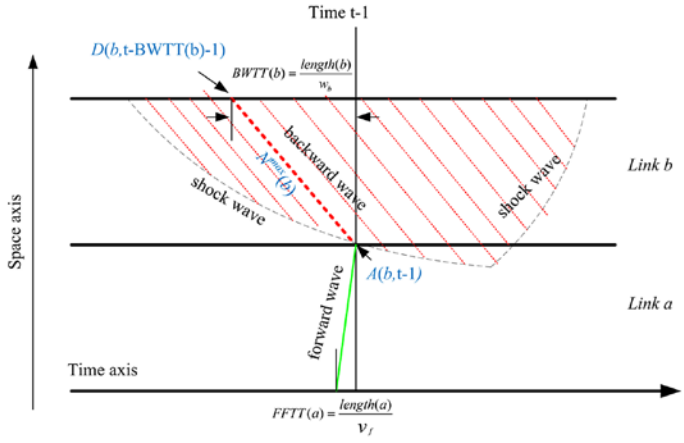


Figure 8. Illustration of forward and backward wave representation in Newell's simplified KW model.

The space–time plot in Figure 8 clearly demonstrates how Newell's model uses cumulative counts to model the forward and backward waves in describing queue phenomenon on links *a* and *b*. Exactly at time $t-1$, the tail of the queue at the downstream link *b* propagates to its upstream node. That is, if the cumulative outflow count at a lagged time stamp $t-\text{BWTT}(b)-1$ on link *b* is equivalent to the cumulative inflow count at time $t-1$, then a queue spillback occurs as a backward wave is able to propagate through the congested time–space "mass" of the link. Compared to the conventional vertical queue model, where the inflow rate of a link is not constrained by the available physical length, the inflow rate in Newell's model is governed by the discharge flow of a link through the backward wave propagation process. Furthermore, the number of vehicles on a lane should not be greater than $k_{\text{jam}} \times \text{length}(b)$. As a result, the proposed model is able to detect and capture queue spillbacks to upstream links.

3.6. Implementing Newell's simplified kinematic wave model as an advanced queue model

Notation

Link attribute variables

length(l)	length of link l
*n*lanes(l)	number of lanes on link l
$q^{\max}(l, t)$	maximum flow rate on link l between time $t-\Delta T$ and time t
$v_f(l)$	free-flow speed (or forward wave speed) on link l
$w(l)$	constant backward wave speed for link l
$k_{\text{jam}}(l)$	jam density on link l
FFTT(l)	free-flow travel time on link l, i.e. length(l)/$v_f(l)$
BWTT(l)	backward wave travel time on link l, i.e. length(l)/w(**l**)

Capacity constraint variables

$\text{cap}^{\text{in}}(l, t)$	inflow capacity of link l between time $t-\Delta T$ and time t
$\text{cap}^{\text{out}}(l, t)$	outflow capacity of link l between time $t-\Delta T$ and time t

Cumulative flow count variables

$A(l, t)$	cumulative number of vehicles entering link l at time t
$V(l, t)$	cumulative number of vehicles waiting at the vertical queue of link l at time t
$D(l, t)$	cumulative number of vehicles departing/leaving link l at time t

For simplicity and clarity, we present the proposed DNL model for a sequence of consecutive, directional links of a freeway corridor that covers links $l = 1, 2, \ldots, L$. Each link l is attached to an upstream node and a downstream node. In the corridor, traffic of a single OD pair is loaded to the first link $l = 1$ and leaves the network from the last link L.

Step 1 (Initialize variables and boundary conditions)

For each link, initialize cumulative flow counts for the entering queue, vertical queue, and exit queue.

$$A(l, t=0)=0, V(l, t=0)=0, D(l, t=0)=0$$

Link attributes, including $q^{max}(l, t)$, $k_{jam}(l)$, $n$lanes(l), FFTT(l), and BWTT(l), are assumed to be known for all links. $q^{max}(l, t)$ is dependent on node flow management rules described in the following section.

Two boundary conditions are assumed to be known for all time intervals: the demand flow, $A(1,t)$, on the first link, and the destination capacity $q^{max}(L, t)$ on the last link.

Step 2 (Update time-dependent traffic states at time t)

For current time stamp t, given $A(l, t')$, $D(l, t')$, and $q^{max}(l, t)$, such that $t' = 0, \Delta T, 2\Delta T, \ldots, t-\Delta T$, perform Steps 2.1–2.3 sequentially for each link $l = 1, 2, \ldots, L$.

Step 2.1 (Forward wave propagation): For each link l, move cumulative arrival flow counts of link l to its vertical queue.

$$V(l,t)=A(l,t-FFTT(l))$$

Step 2.2 (Backward wave propagation): For each link l, calculate the maximum possible cumulative arrival count for the flow to move into link l at time t.

$$A^{max}(l, t)=D(l, t-\text{BWTT}(l))+k_{jam}(l)\times\text{length}(l)\times n\text{lanes}(l)$$

Step 3 (Calculate capacity constraints)

Determine the inflow and outflow capacity constraints for each link at time t.

Step 3.1 (Determine inflow capacity): For each link $l = 1, 2, \ldots, L$, determine the inflow capacity of link l at time t.

$$\text{cap}^{in}(l, t)=A^{max}(l, t)-A(l, t-\Delta T)$$

Step 3.2 (Determine outflow capacity): For links $l = 1, 2, \ldots, L-1$, determine the outflow capacity of link l at time t.

$$\text{cap}^{out}(l, t)=\min\{q^{max}(l, t), \text{cap}^{in}(l+1, t)\}$$

$\text{cap}^{out}(L, t)=q^{max}(L, t)$ for the last link L

Step 4 (Transfer flow between links)

For each node n = 1, 2, ..., N, identify incoming and outgoing links, and identify agents/vehicles ready to move between those incoming and outgoing links at time t.

Step 4.1 (Update cumulative departure count): For each link upstream from node n, update the cumulative departure counts.

$$D(l, t)=\min\{V(l, t), D(l, t-\Delta T)+\text{cap}^{out}(l, t)\}$$

Step 4.2 (Update cumulative arrival count): For each link downstream from node n, update the cumulative arrival counts.

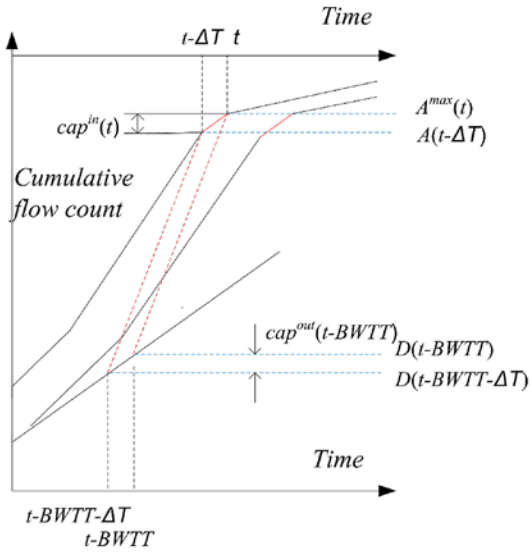


Figure 9. Illustration of queue spillback and propagation of outgoing flow constraint from the downstream end at time *t*-BWTT to the upstream *end* at time *t*.

$$A(l+1,t)=D(l,t)$$

Step 5 (Advance simulation clock):
Advance $t = t + \Delta T$, go back to Step 2.

Given existing traffic states in the previous time interval $t-\Delta T$, the DNL model follows a multi-step process for moving agents/vehicles along a given path between specific origin and destination nodes, subject to capacity constraints and time-dependent traffic states defined by shockwave propagation in Newell's simplified kinematic wave model. The DNL first updates the values of traffic state variables in time interval t in Step 2, and determines the maximum number of agents/vehicles which can be transferred between links (at nodes) in Step 3. Those agents/vehicles are then moved between links in Step 4.

As a detailed illustration, Figure 9 demonstrates how to calculate the maximum inflow capacity and outflow capacity in a queue spill-back case. The maximum inflow capacity $\text{cap}^{\text{in}}(t)$ at the upstream end of a link is calculated using the difference of cumulative arrival flow counts at two consecutive time stamps $t-\Delta T$ and t, which is determined by the outflow capacity $\text{cap}^{\text{out}}(t-\text{BWTT})$ at the downstream end between time stamps $t-\text{BWT}T-\Delta T$ and $t-\text{BWTT}$.

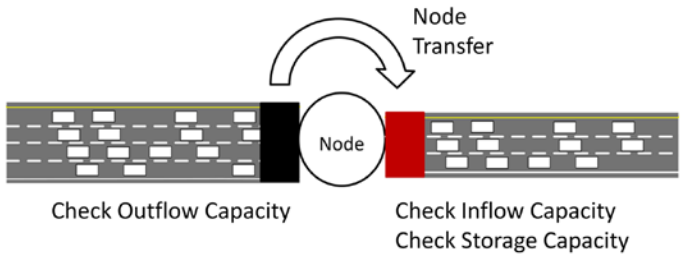
3.7. Node management/control models

Within a typical network, different node types (i.e. origin, merge, and signalized intersections) may require unique methods for moving agents between different links. In the proposed DNL model, agents are loaded into the network at the origin node using a loading buffer, rather than being loaded directly into link $l = 1$. Three separate models, based on previous studies in the literature, can be used to allocate capacity at merge junctions. Lastly, outflow capacity for incoming links at signalized intersections is allocated based on a simplified model using cycle times and link effective green times. Diverge junctions require no special handling because the paths are known for agents/vehicles traveling through the network (Figure 10).

3.7.1. Origin nodes and the loading buffer

As stated in kinematic wave-based DNL algorithm, the demand flow, $A(1, t)$, on the first link is assumed to be known for all time intervals. As a result, each agent/vehicle is assigned to enter the network at a specific departure time based on some semi-random time interval between arrivals. At the origin node, located upstream from the first link, agents/vehicles first enter the loading buffer, which acts as a temporary storage queue. In each time interval t, if an agent's departure time $\leq t$ and

Figure 10. Illustration of node transfer process.



there is inflow capacity available, those agents/vehicles are moved from the loading buffer to the entrance queue on link $l = 1$.

3.7.2. Merge junctions

Consider a merge junction with incoming link 1 and 2 merging into downstream link 3, as shown in Figure 11. If the total demand from all incoming links (denoted as $d1$ and $d2$ on the two links) is greater than the available inflow capacity on link 3, denoted as cap^{in}, then the available inflow capacity must be distributed to each incoming link.

To determine the outflow capacity for mainline link 1 and onramp link 2, namely $\text{cap}^{\text{out}}(1)$ and $\text{cap}^{\text{out}}(2)$, we use Daganzo's priority-based merge model (1994) in this study to maximize the utilization of downstream inflow capacity, while upstream outflows satisfy feasibility constraints.

If $d1 + d2 < \text{cap}^{\text{in}}$, then $\begin{cases} \text{cap}^{\text{out}}(1) = \text{mid}\{d1, \text{cap}^{\text{in}} - d1, p1 \times \text{cap}^{\text{in}}\} \\ \text{cap}^{\text{out}}(2) = \text{mid}\{d2, \text{cap}^{\text{in}} - d2, p2 \times \text{cap}^{\text{in}}\} \end{cases}$ where

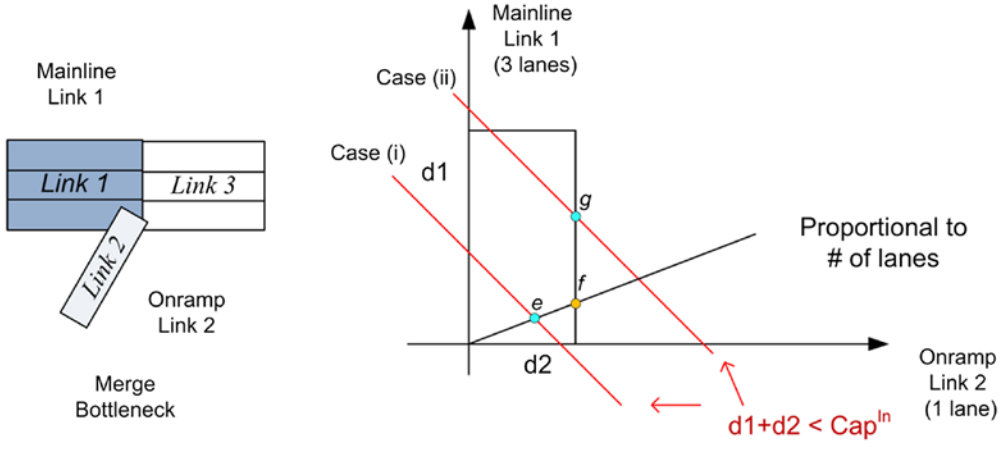
$p1 = \dfrac{n\text{lanes}(1)}{n\text{lanes}(1) + n\text{lanes}(2)}$, $p2 = \dfrac{n\text{lanes}(2)}{n\text{lanes}(1) + n\text{lanes}(2)}$ according to a lane-based proportional distribution rule, and the mid function takes the middle value of all three input parameters.

Illustrated in Figure 11, in case (i), the solution corresponds to point e. The solution to case (ii) should be point *g* rather than point f. While point *g* optimizes the overall inflow capacity allocation, point *f* underutilizes the inflow capacity by strictly following the lane-based distribution rule.

3.7.3. Flow conservation at diverge junctions

In continuous flow-based traffic simulation, many macroscopic fluid-based models use continuous ratios of destination distribution to decide the amount of traffic flow to different destinations. As the mesoscopic DNL model moves agents/vehicles with OD and path information, the proportions of vehicles moving out of a link to individual outgoing links are in fact determined directly by the paths (downstream node sequences) associated with vehicles at this link, so it is very easy to decide which

Figure 11. Capacity allocation at merge bottleneck.



sets of mesoscopic agents to be moved to a specific destination at any intersection or diverge point associated with an off ramp.

3.7.4. Signalized intersections

The proposed DNL model considers the influence of traffic signals using a simplified method for estimating the link outflow capacity. Assuming that the effective green time and saturation flow rate are known for link *l*, upstream from node *n*, and the cycle time is known for the signal at node *n*, the link outflow capacity on link *l* is calculated using the equation below.

$$\text{cap}^{\text{out}}(l,t) = q^{\text{saturation}}(l,t)\frac{\text{effective green time}(l)}{\text{cycle length}(n)} \tag{4}$$

To achieve an effective coupling of a signal timing estimation model and DTA in feedback loops, Zlatkovic and Zhou (2014) integrated a quick estimation method to link critical movement analysis methodologies with the DTALite simulation package.

4. Queue-based dynamic OD demand estimation

To allow consistent OD demand estimation for fast model calibration, DTALite embeds a path flow-based optimization model that utilizes sensor data (i.e. observed link flows and densities) and target OD demands $\bar{d}(i,j,\tau)$to estimate a set of path flow volumes. This approach combines OD estimation/adjustment with traffic assignment seamlessly. That is, under this ODME model, DTALite first runs *K* assignment iterations (e.g. 40 iterations) to generate likely paths, and then performs anther *K′* iterations (e.g. 100 iterations) with ODME enabled to provide the final solution as a set of path flow patterns satisfying "tolled user equilibrium." In our discussion below, we want to highlight the use of the approximate gradient method in DTALite, which utilizes a queue model to calculate link flow–density change due to incoming path flow change. The detailed mathematical formulation and solution algorithm can be referred to a recent paper by Lu, Zhou, and Zhang (2013).

4.1. Mathematical model formulation

The problem statement and notations of ODME can be summarized as follows. Using path flow vector $\mathbf{r}(i,j,\tau,k)$ for OD pair (i,j) at departure time at time k, the model minimizes a set of deviation functions with respect to target OD demands $\bar{d}(i,j,\tau)$ and observed link flows and densities, subject to (i) DNL constraints that describe traffic flow propagation with multiple OD pairs and (ii) a gap function-based constraint that measures the deviation from the dynamic user equilibrium conditions.

$$\text{Min } Z = \beta_d \sum_{i,j,\tau}\left[d(i,j,\tau)-\bar{d}(i,j,\tau)\right]^2 + \sum_{a\in S}\sum_{t\in H_o}\left\{\beta_q[q(a,t)-\bar{q}(a,t)]^2+\beta_k[k(a,t)-\bar{k}(a,t)]^2\right\} \tag{5}$$

where $q(a,t)$ and $\bar{q}(a,t)$ are the simulated and observed link flows, respectively, on link a at time interval $[t, t+1]$; $k(a,t)$ and $\bar{k}(a,t)$ are the simulated and observed link densities, respectively, on link a at time t; S is the set of links with observations; H_o is the set of link intervals with observations; and β_d, β_q, β_k is the weights reflecting different degrees of confidence on target OD demands and observed link flows and densities, respectively.

In the estimation step, within a column generation-based framework, a gradient-projection-based descent direction method (Lu, Mahmassani, & Zhou, 2009) is used to update path flows $r^{(m+1)}$ at step $m+1$. Specifically,

$$\begin{aligned} r(i,j,\tau,k)^{m+1} \quad = \quad & \text{Max}\Big\{0, r(i,j,\tau,k)^m - \gamma^{(m)}\Big[\beta_d\nabla h^d(r)|_{r=r^{(m)}} \\ & +\beta_q\nabla h^q(r)|_{r=r^{(m)}}+\beta_k\nabla h^k(r)|_{r=r^{(m)}}+\lambda^{(n)}\nabla g(r,\pi)|_{r=r^{(m)}}\Big]\Big\} \end{aligned} \tag{6}$$

where $\gamma^{(m)}$ is the step size, and the gradients that consist of the first-order partial derivatives with respect to a path flow variable $r(i, j, \tau, k)$ are discussed as follows, based on an queue-representation in Figure 13.

The solution framework integrates a gradient-projection-based path flow adjustment method. The simple queuing model allows this tool to also derive analytical gradient formulas for the changes in link flow and density due to the unit change of time-dependent path inflow in a general network under congestion conditions, as shown in Figure 12.

In this study, the link partial derivative is referred to as the change in link flow and density due to an additional unit of link/path inflow. Ghali and Smith (1995) presented an analytical approach to evaluate the (local) link marginal travel time (or delay) on a congested link, based on link cumulative flow curves of a simple point queue model. We illustrate the use of the queue-based link partial gradient calculation in Figure 13, which shows the cumulative arrival and departure curves $A(t)$ and $D(t)$, and the dashed line represents the cumulative vertical queue. Given (outflow) capacity of the link as c, the additional change in the A/D curve is equivalent to the gray area. In this example, FFTT(l) = 15 min. The queue starts at t_l^{qs} = 7:00, and an additional unit of flow, vehicle n' enters link l

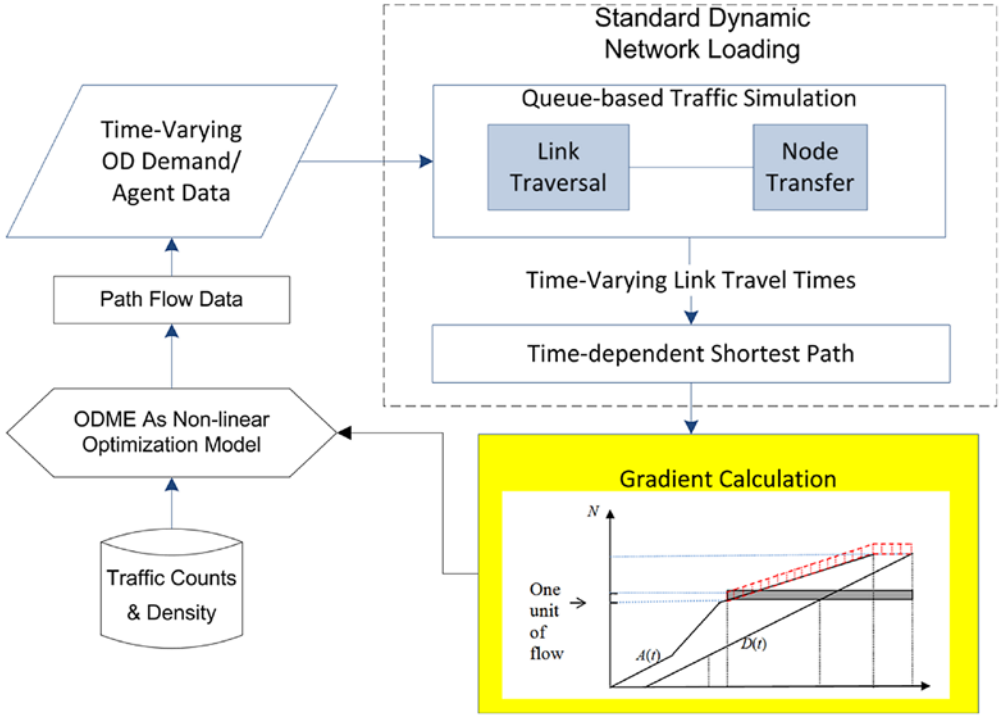


Figure 12. Single-level Origin -Destination Matrix Estimation (ODME) flowchart.

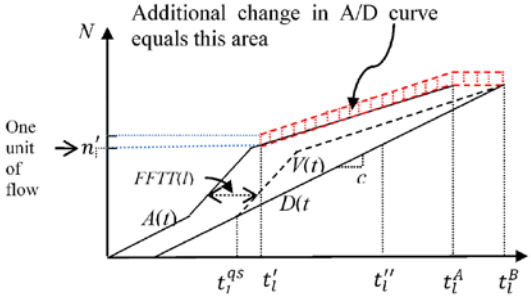


Figure 13. Illustration of link partial gradient on a congested link.

at time $t_l' = 7{:}10$. This vehicle leaves the link at time $t_l'' = 7{:}50$, the queue dissipates on link l at $t_l^B = 8{:}30$, and $t_l^A = 8{:}15$.

We can easily analyze the marginal effects of an additional unit of flow on link flow (inflow and outflow) and the number of vehicles (and corresponding density). In particular, under partially congested conditions shown in Figure 13, the link inflow and outflow increase by one at times t_l' and t_l^B, respectively, and the flow rates at the other time intervals do not change due to the assumed constant capacity from t_l' to t_l^B, while the number of vehicles in the link increases by one for the same time period.

5. Numerical experiments

5.1. Verifying shock wave propagation speed on a hypothetic corridor

The proposed queue-based mesoscopic DNL model and the ODME method have been implemented in DTALite in C++, using an OpenMP as the application programming interface for parallel computing. We first consider a hypothetic corridor with a lane drop bottleneck. Based on the flow–density relationship in Figure 14, the theoretical shock wave propagation speed is (1560−1200)/(80−26) = 6.67 mph. Figure 15 shows the dynamic density contour generated from DTALite simulation results. The congestion starts at 7:12 at node 9 and propagates to the upstream node 1 at around 8:15 along this 7 mile stretch. Accordingly, we can calculate the simulated shockwave propagation speed from DTALite as 7 mile/63 min = 6.67 mph, which is nicely consistent with the theoretical value.

5.2. Large real-world network calibration using OD demand calibration

As shown in Figure 16, this real-world network is the Triangle Regional Model network containing most of Raleigh, North Carolina, USA. This large-scale regional network has 2,389 zones, 20,259 links, and about 2,000 signalized intersections. Provided by the local metropolitan planning agency, the

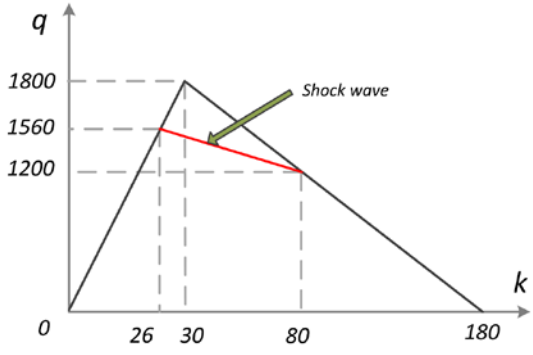


Figure 14. Flow–density relationship and shock wave on a lane drop bottleneck.

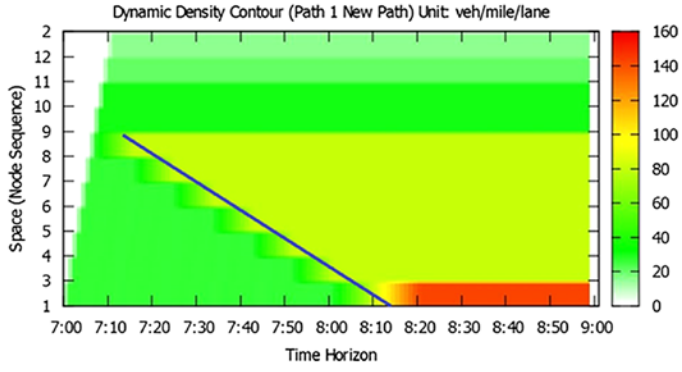


Figure 15. Dynamic density contour on a hypothetic corridor with a lane drop bottleneck on node 9.

Figure 16. Triangle Regional Model, NC network with 2,389 zones, 20,259 links, about 2,000 signalized intersections, 1.06 million vehicles in morning peak hours.

morning peak-hour demand matrix has about 1.06 million vehicles, covering a time period from 6 am to 10 am. There are about 16 and 14 sensors, respectively, on freeway and arterial links, producing a total of 120 hourly link count observations. The experiments were performed on a PC with 16 GB memory and 4-core processors with hyperthreading, running at 2.70 GHz. Using the open-source DTA package DTALite, the running time is about 2 min and 45 s per iteration. When incorporating the additional path flow adjustment process, the average running time increases to 5 min and 3 s per iteration. The iterative sequential adjustment converged after 140 iterations, which required a total of 12 h of CPU time. The scatter plots in Figures 17 and 18 show additional MOEs—R^2, total estimated vs. observed flow ratio. The average absolute link flow deviations are 435.15 and 212.21 vehicles per hour per link, respectively, on freeway and arterial links, which further

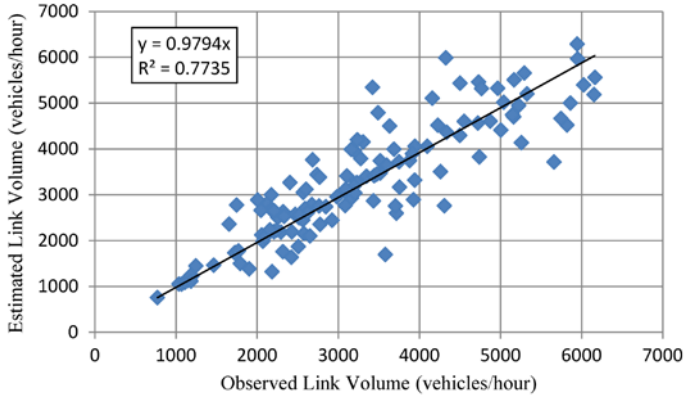


Figure 17. Observed vs. estimated link volume on freeway links on the Triangle Regional Model, NC network.

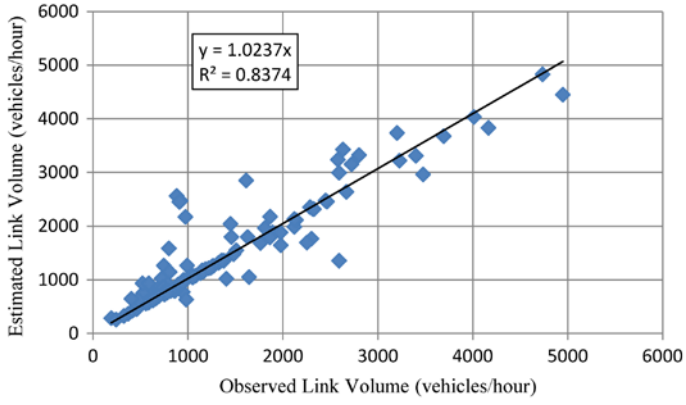


Figure 18. Observed vs. estimated link volume on highway and arterial links on the Triangle Regional Model, NC network.

demonstrate the effectiveness of the proposed method on both freeway and arterial links of the large-scale network.

6. Conclusions

In this paper, a queue-based mesoscopic DNL model is proposed to allow rapid simulation of traffic congestion propagation in a network with various bottlenecks. This mesoscopic DNL model, based on point queues, spatial queues, and Newell's simplified kinematic wave theory, is proposed to realistically estimate dynamic flow and travel time performances for an estimated OD demand table.

Based on the cumulative flow counts from the queueing simulation results, we can analytically derive partial derivatives of link flow and density with respect to path inflow perturbations, and these results can provide essential gradient information to determine the feasible descent direction in the OD demand estimation algorithm. Based on a class of queueing models, the seamless integration between the OD flow estimation and DNL model offers an efficient way to utilize available sensor data sources.

The presented queue-based mesoscopic traffic simulation model has considerable potential for generalizing the modeling framework into the field of real-time traffic state estimation and prediction. In our future research, we will further examine different ways for calibrating the maximum queue discharge rates, utilizing end-to-end travel times, and considering an agent-based learning framework to fully consider behavioral heterogeneity.

Acknowledgments
The first author appreciates Prof. Hani S. Mahmassani's instructions and guidance for the fundamental knowledge of DTA. The first author would also like to thank Jason Chung-Cheng Lu and Kuilin Zhang for their help in developing the dynamic origin destination demand estimation model. Both authors, when working at the University of Utah, were partially supported through a FHWA project titled "An Open-Source DTA Tool for Assessing the Effects of Roadway Pricing and Crash Reduction Strategies on Recurring and Non-Recurring Congestion." Special thanks to our colleagues Nagui Rouphail and Anxi Jia at the North Carolina State University, Wayne Kittelson and Brandon Nevers at Kittelson & Associates, Inc, Brian Gardner from FHWA, for their constructive comments. The work presented in this paper remains the sole responsibility of the authors.

Funding
Xuesong Zhou and Jeffrey Taylor, when working at the University of Utah, were partially supported through a FHWA project titled "An Open-Source DTA Tool for Assessing the Effects of Roadway Pricing and Crash Reduction Strategies on Recurring and Non-Recurring Congestion."

Author details
Xuesong Zhou[1]
E-mail: xzhou74@asu.edu; xzhou99@gmail.com
Jeffrey Taylor[2]
E-mail: jeff.d.taylor@utah.edu
[1] School of Sustainable Engineering and the Built Environment, Arizona State University, Tempe, AZ 85287, USA.
[2] Department of Civil and Environmental Engineering, University of Utah, Salt Lake City, UT 84112-0561, USA.

Citation information
Cite this article as: DTALite: A queue-based mesoscopic traffic simulator for fast model evaluation and calibration, X. Zhou & J. Taylor, *Cogent Engineering* (2014), 1: 961345.

References

Ben-Akiva, M. E., Bierlaire, M., Burton, D., Koutsopoulos, H. N., & Mishalani, R. (2002). Real-time simulation of traffic demand–supply interactions within DynaMIT. In M. Gendreau, & P. Marcotte (Eds.), *Transportation and network analysis: Current trends. Miscellenea in honor of Michael Florian* (pp. 19–36). Boston, MA: Kluwer.

Daganzo, C. F. (1994). The cell transmission model: A dynamic representation of highway traffic consistent with the hydrodynamic theory. Transportation Research Part B: Methodological, 28, 269–287. http://dx.doi.org/10.1016/0191-2615(94)90002-7

Daganzo, C. F. (1995). The cell transmission model, part II: Network traffic. Transportation Research Part B: Methodological, 29, 79–93. http://dx.doi.org/10.1016/0191-2615(94)00022-R

Ghali, M. O., & Smith, M. J. (1995). A model for the dynamic system optimum traffic assignment problem. *Transportation Research Part B, 29,* 155–170. doi:10.1016/0191-2615(94)00024-T

Hurdle, V., & Son, B. (2000). Road test of a freeway model. *Transportation Research Part A: Policy and Practice, 34*, 537–564.

Hurdle, V., & Son, B. (2001). Shock wave and cumulative arrival and departure models: Partners without conflict. *Transportation Research Record: Journal of the Transportation Research Board, 1776*, 159–166. http://dx.doi.org/10.3141/1776-21

Lu, C.-C., Mahmassani, H. S., & Zhou, X. (2009). Equivalent gap function-based reformulation and solution algorithm for the dynamic user equilibrium problem. *Transportation Research Part B: Methodological, 43*, 345–364.

Lu, C.-C., Zhou, X., & Zhang, K. (2013). Dynamic origin–destination demand flow estimation under congested traffic conditions. *Transportation Research Part C: Emerging Technologies, 34*, 16–37. http://dx.doi.org/10.1016/j.trc.2013.05.006

Mahmassani, H. S. (2001). Dynamic network traffic assignment and simulation methodology for advanced system management applications. *Networks and Spatial Economics, 1*, 267–292. http://dx.doi.org/10.1023/A:1012831808926

Mahmassani, H. S., Hu, T.-Y., Peeta, S., & Ziliaskopoulos, A. (1994). *Development and testing of dynamic traffic assignment and simulation procedures for ATIS/ATMS applications*. McLean, VA: US DOT, Federal Highway Administration.

Newell, G. F. (1993). A simplified theory of kinematic waves in highway traffic, part I: General theory. *Transportation Research Part B: Methodological, 27*, 281–287. http://dx.doi.org/10.1016/0191-2615(93)90038-C

Ni, D., Leonard, J. D., & Williams, B. M. (2006). The network kinematic waves model: A simplified approach to network traffic. *Journal of Intelligent Transportation Systems, 10*(1), 1–14. http://dx.doi.org/10.1080/15472450500455070

Nie, Y., Ma, J., & Zhang, H. M. (2008). A polymorphic dynamic network loading model. *Computer-Aided Civil and Infrastructure Engineering, 23*, 86–103.

Peeta, S., & Ziliaskopoulos, A. K. (2001). Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics, 1*, 233–265. http://dx.doi.org/10.1023/A:1012827724856

Reynolds, W. L., Zhou, X., Rouphail, N. M., & Li, M. (2010). Estimating sustained service rates at signalized intersections with short left-turn pockets. *Transportation Research Record: Journal of the Transportation Research Board, 2173*, 64–71. http://dx.doi.org/10.3141/2173-08

Yperman, I., Logghe, S., & Immers, B. (2005). *The link transmission model: An efficient implementation of the kinematic wave theory in traffic networks* (pp. 122–127). Paper presented at the Proceedings of the 10th EWGT Meeting, Poznan, Poland.

Zlatkovic, M., & Zhou, X. (2014). *Effective coupling of signal timing estimation model and dynamic traffic assignment in feedback loops: System design and case study*. Paper presented at the Transportation Research Board 93rd Annual Meeting, Washington, DC, USA.

***Cogent Engineering* (ISSN: 2331-1916) is published by Cogent OA, part of Taylor & Francis Group.**