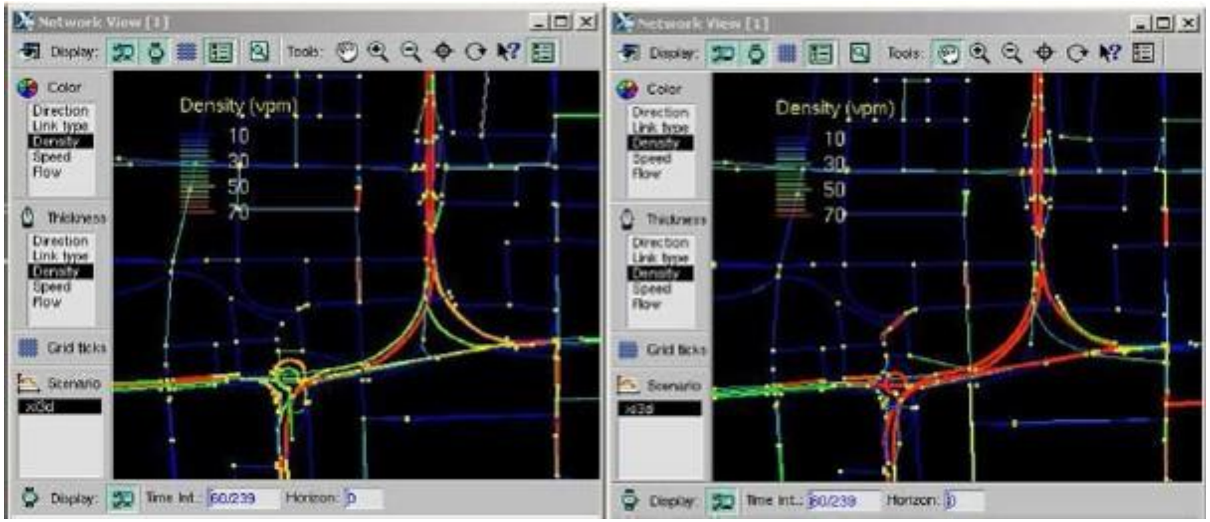
SMART
Singapore-MIT Alliance for Research and Technology

FUTURE URBAN MOBILITY

DynaMIT/SM Mid Term – Supply Simulator



Contents

- Introduction
- Network Representation
  - Static Components
  - Dynamic Components
  - Output Capacity
- Traffic Dynamics
  - Introduction
  - Deterministic Queuing model
  - Speed Model
  - Vehicle Movement Model
- Simulation Process

Introduction

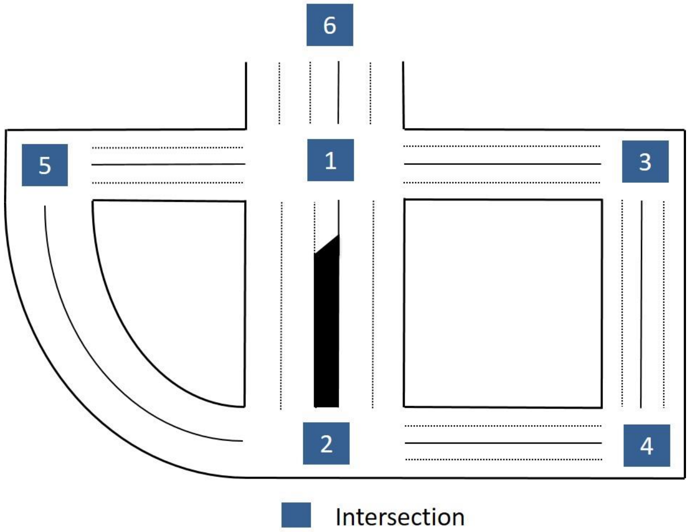
- Traffic/Supply simulator
  - Simulation of vehicle movements on a given network
  - Infer traffic flows/speeds/densities/queue lengths, etc.
- DynaMIT: Mesoscopic Supply simulator
  - Does not simulate individual movements such as lane-changing and car following
  - Represents traffic dynamics using speed–density relationships and queuing theory
    - Formation and dissipation of queues
    - Spillback effects
    - Impacts of incidents
    - Bottlenecks

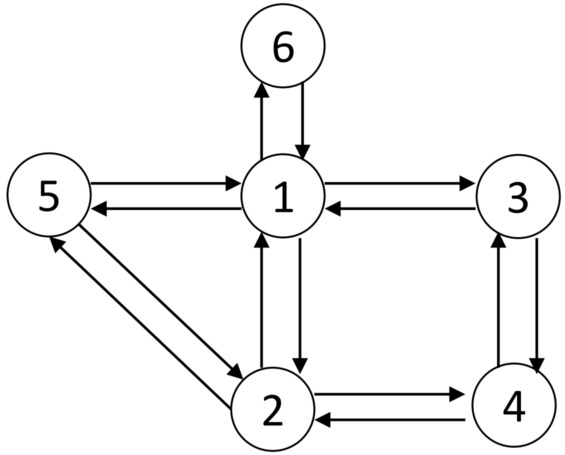
Network Representation

Static Components

- **Nodes** – Intersections on the actual network
- **Links** – Unidirectional pathways between nodes
- **Loading Elements** (Loaders) – Areas where traffic is generated and/or attracted; associated with the node to which they are connected and vehicles can be loaded onto any downstream links of the associated node

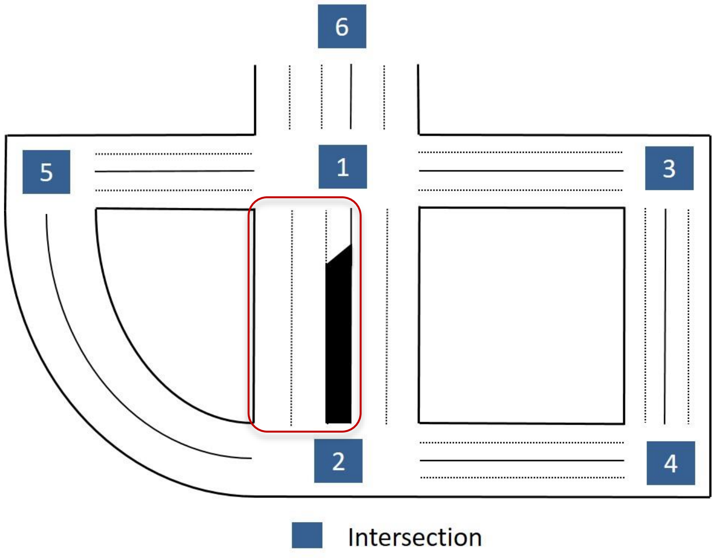
Example Road Network

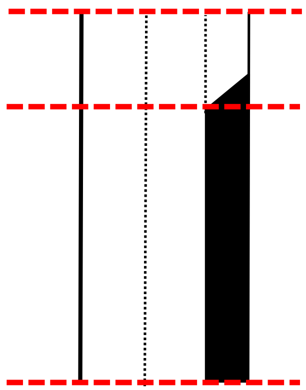




Dynamic Components: Segments

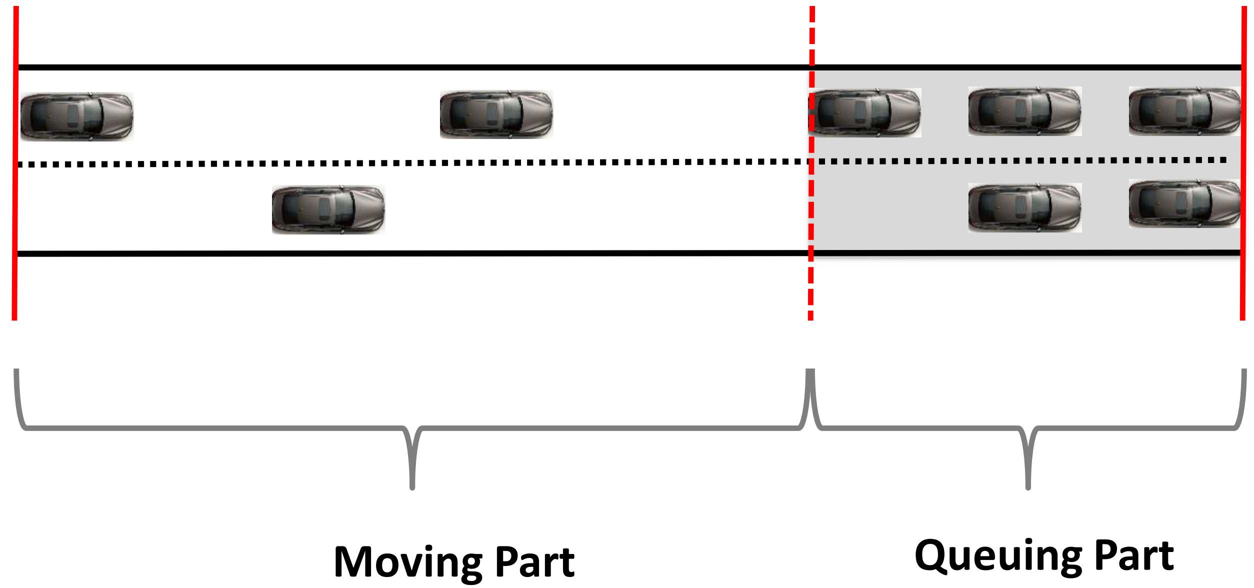
- Segments – Each link is divided into segments that capture variations of geometry and traffic conditions along the link. While most segments are defined in advance, additional segments can be dynamically created to capture the presence of incidents.
- Segments are associated with a downstream capacity constraint caused by:
  - Roadway characteristics
  - Incident
  - Specific control device





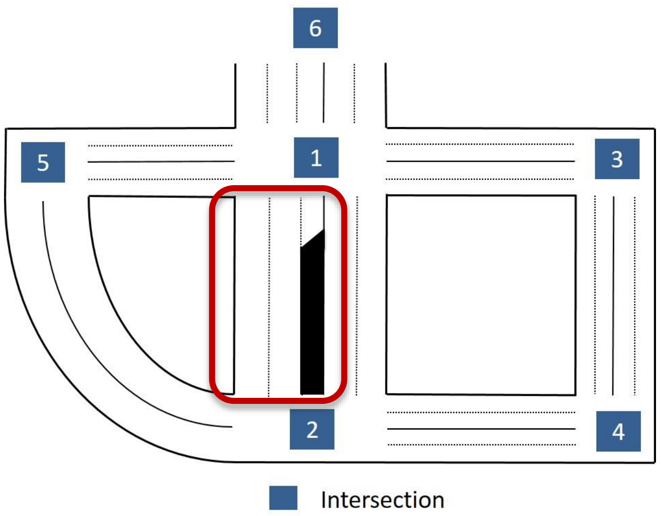
Dynamic Components: Segments

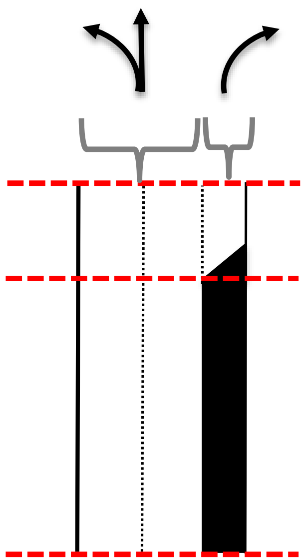
- Segments – Each segment is also divided into a
  - **Moving part**: represent vehicles that can move with some speed
  - **Queuing part**: represent vehicles that are queued



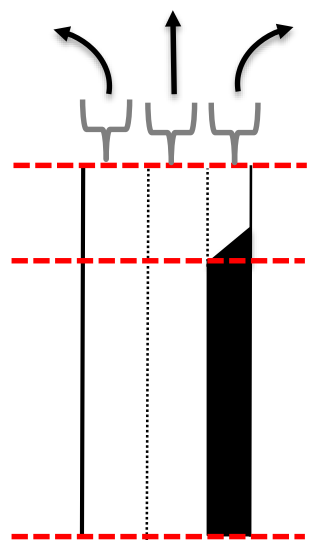
Dynamic Components: Lane Groups/Lanes

- Segments are composed of:
  - **Lane Groups**: Set of direction streams that operate at the same time (collection of lanes); *a lane in DynaMIT cannot belong to multiple groups*
  - **Lanes**: Single physical traffic lane





Option 1



Option 2

Dynamic Components: Attributes

Segment Attributes
Identifier of the associated link
Position of the beginning of the segment
Length
List of component lane groups
Free flow speed
Jam density
Density (Moving + Queuing)
Density (Moving)
Speed at upstream end
Acceptance capacity
Acceptance rate

Segment Attributes
Number of lanes
Greenshields speed-density parameters
List of moving packets
Time of last accept
Time of last release
Output Capacity
Segment Processed ?

Dynamic Components: Attributes

Lane Group Attributes
Identifier of the associated segment
Number of Lanes
Output Capacity
Output Counter
Speed at upstream end
List of component lanes
Next Link
Speed at upstream end
Acceptance rate
Last Accept

Lane Attributes
Identifier of the associated lane group
Queue Length
Initial Queue Length
List of Queuing packets
List of next links

Output Capacity

- Output Capacity: the determination of output capacities is based on recommendations from the HCM*; Two cases can be distinguished:
  - Control at signalized intersections: Intersection signal control can be separated into pretimed and traffic actuated. An output capacity is assigned to each lane group of a segment. The capacity for each lane group at the intersection is defined as a function of the effective green time, cycle time, and saturation flows, determined according to the geometry of the intersection.

- Highway Capacity Manual, 2010. Transportation Research Board of the National Academies.
- NCHRP Report 599 (2008) Default Values for Highway Capacity and Level of Service Analyses, Transportation Research Board of the National Academies.

Lane Group Definitions

Number of Lanes	Movements by Lanes	Number of Possible Lane Groups
1	LT + TH + RT	1 (Single-lane approach)
2	EXC LT TH + RT	2
2	LT + TH TH + RT	1 OR 2
3	EXC LT TH TH + RT	2 OR 3

Source: Highway Capacity Manual, 2000. Transportation Research Board of the National Academies.

Output Capacity

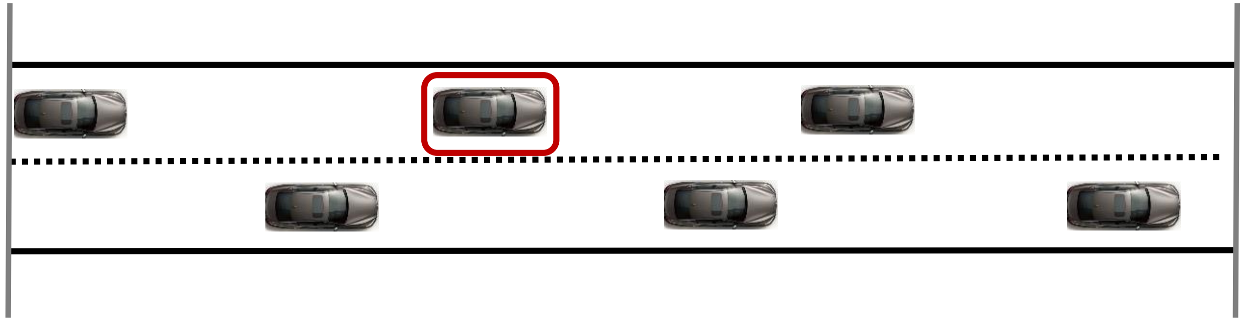
- Control at unsignalized intersections: Unsignalized intersections are either uncontrolled or controlled by stop and yield signs. The approaches that are not controlled or controlled by yield signs correspond to major links; the approaches that are controlled by stop signs correspond to minor links. The capacity for the turning movements from major links is calculated based on the geometric characteristics of the intersection and a correction factor defined as a function of total approach flow from minor links. The capacity for the minor links is calculated using a model based on the acceptable gap concept.

Traffic Dynamics

Introduction

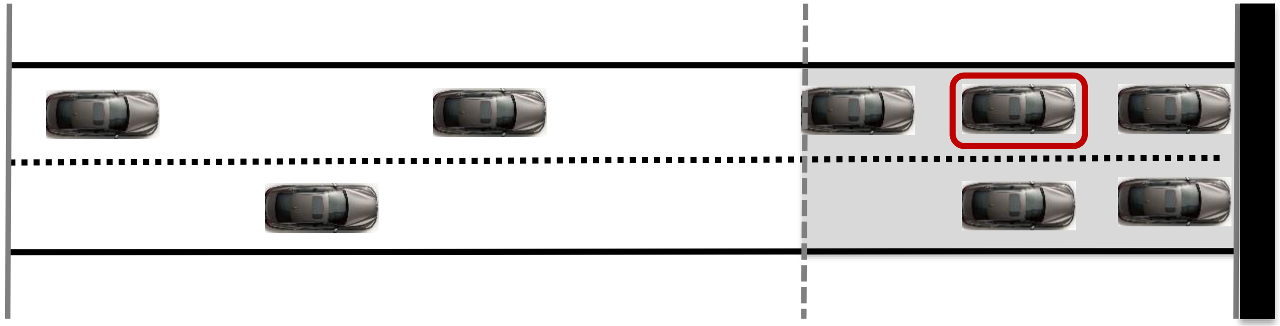
- Traffic dynamics captured through two models
  - Deterministic Queuing model
  - Speed model
- Queuing/Moving Scenarios

1. No Queue

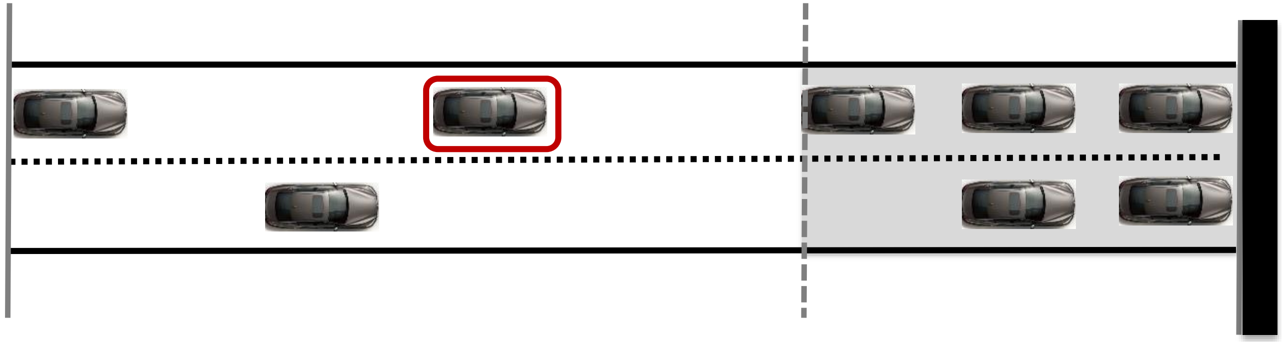


Introduction

2. Blocked Queue (Vehicle in queue)

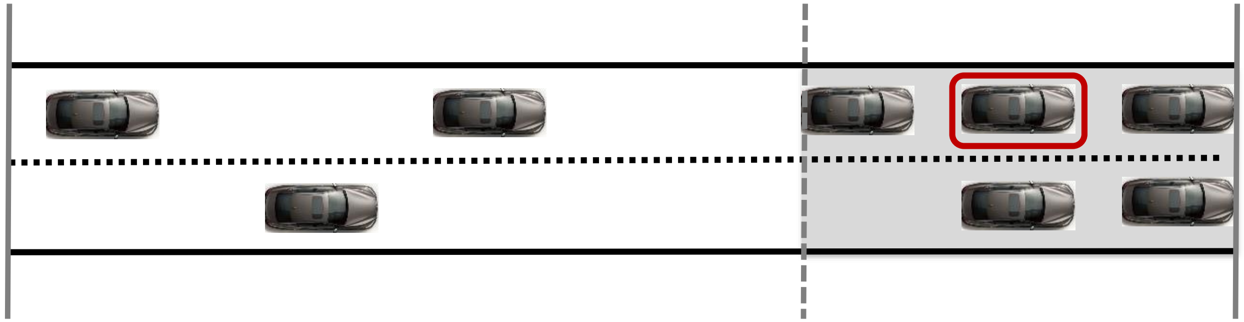


3. Blocked Queue (Vehicle moving)

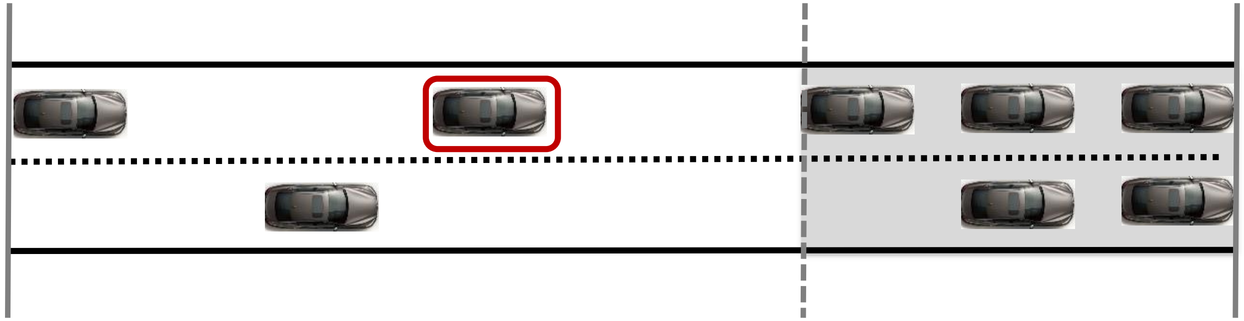


Introduction

4. Moving Queue (Vehicle in queue)

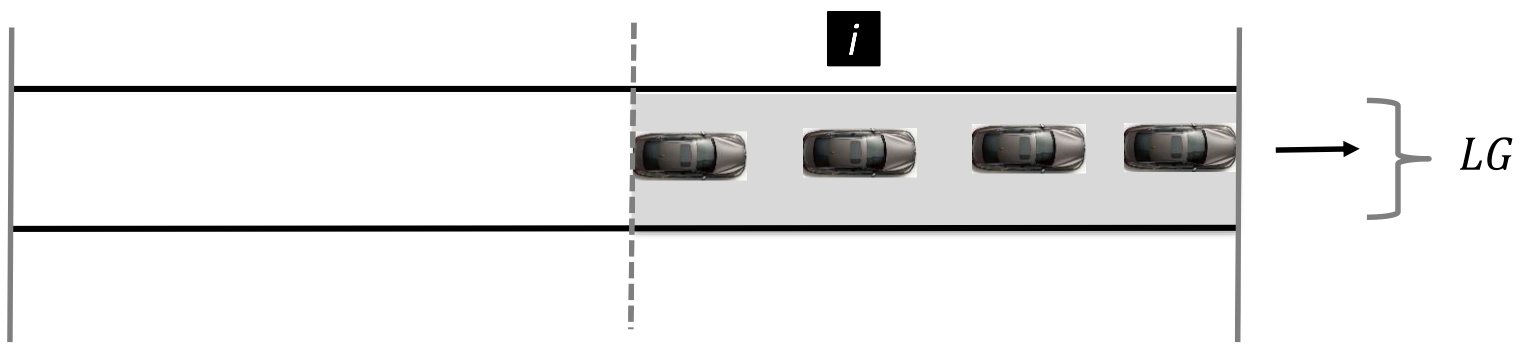


5. Moving Queue (Vehicle moving)



Deterministic Queuing Model

- Simulation captures the formation of queues and represents them at the lane level
- Treatment of queue lengths is on a per lane basis
- Let c (vehs/second) represent the output capacity of the lane group LG

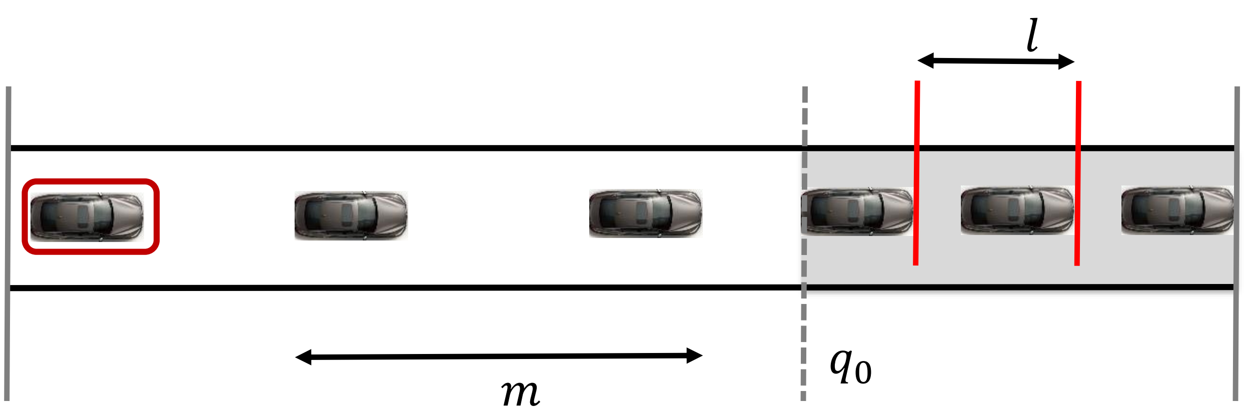


- Queing delay of the i^{th} vehicle is given by $i \times \left(\frac{1}{c}\right)$

Deterministic Queuing Model

- Previous model can be used to determine vehicle position
- Consider the case of a moving vehicle with a queue at the end of the segment
- During a time period t, ct vehicles leave the queue
- If a moving vehicle reaches the end of the queue at a time t, its position and therefore, the position of the end of the queue is given by

$$q(t) = q_0 + l(ct - m)$$



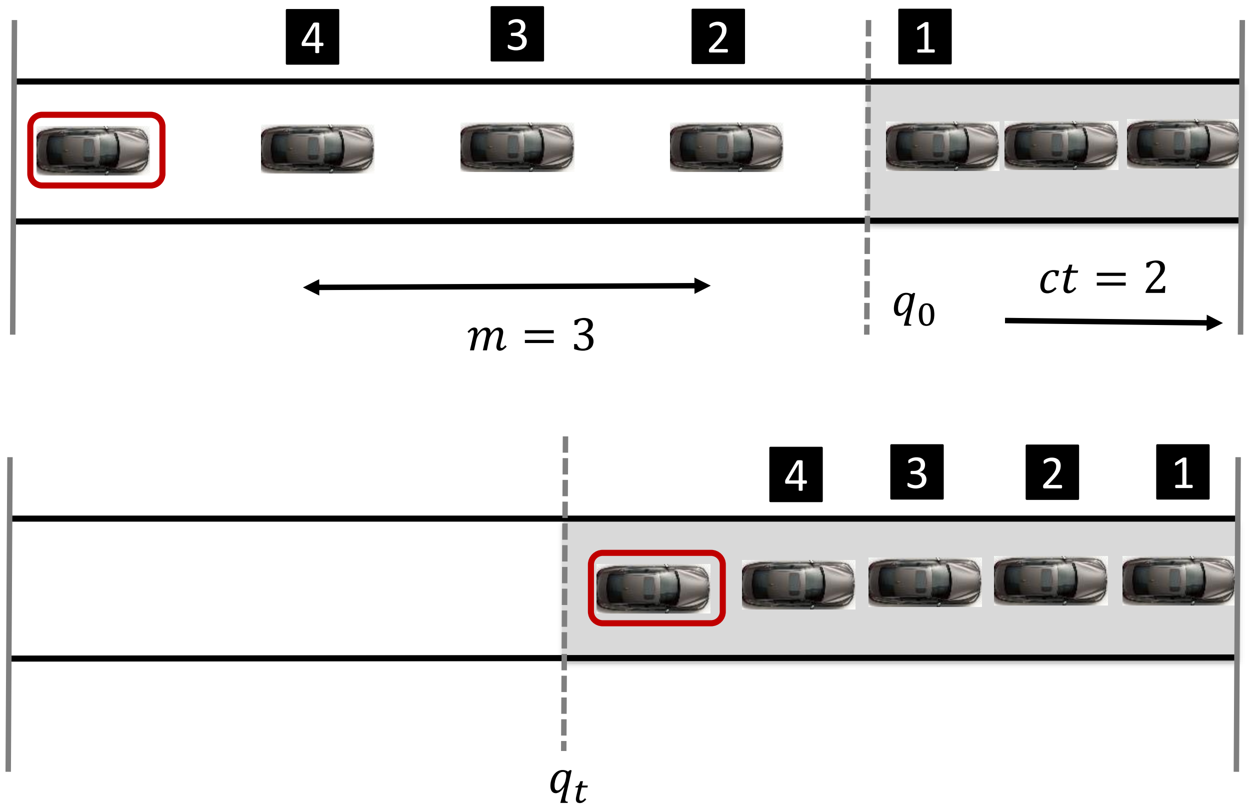
$q(t)$: Distance measured from the start of the segment

q_0: Queue position at $t = 0$

l: Vehicle length including headway

m: Number of moving vehicles between considered vehicle and queue at $t = 0$

Deterministic Queuing Model

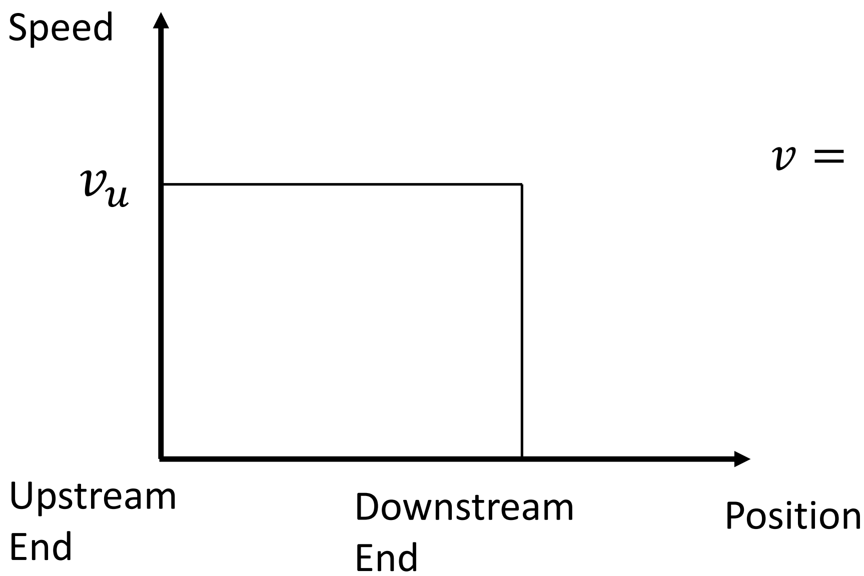


$q(t) < 0$ will never occur as spillbacks are explicitly modeled

$q(t) = L$ implies that the queue has completely dissipated

Speed Model (I)

- Uniform Velocity Profile along the length of the segment
- Constant speed along the length of the segment for moving vehicles



$$v = \begin{cases} v_{max} & , k \leq k_{min} \\ v_{max}\left[1 - \left(\frac{k - k_{min}}{k_{jam}}\right)^{\beta}\right]^{\alpha} & , k > k_{min} \end{cases}$$

$$v_u = \max(v, v_{min})$$

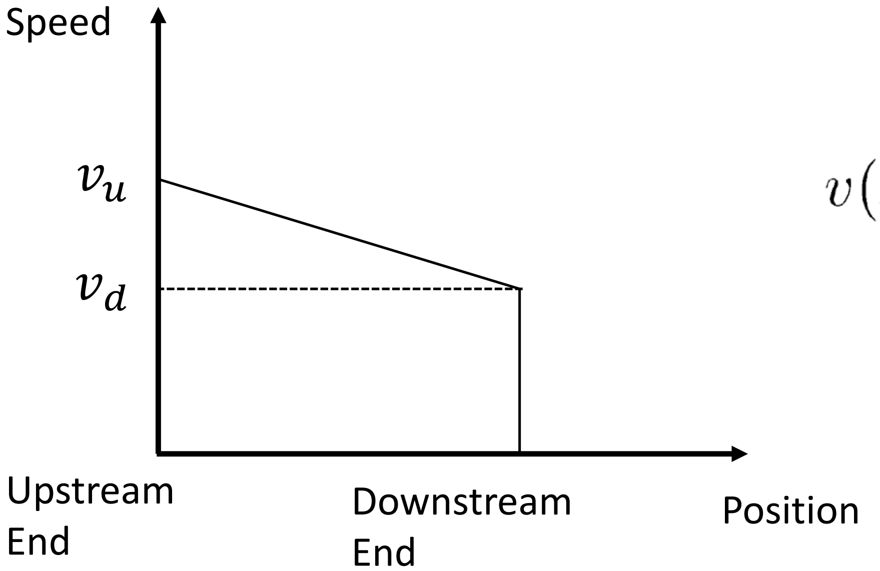
$Parameters: v_{max}, k_{min}, \alpha, \beta, v_{min}, k_{jam}$

Vehicle Movement Model (I)

- Queuing and speed models used to simulate the movement of vehicles on the network
- Assume at time $t = 0$, the vehicle is at position $x = x_0$
- Absence of a queue
  - Vehicle will reach position x at time: $t(x) = \dfrac{x - x_0}{v_u}$
  - Position at time t: $x(t) = v_u t + x_0$
- Presence of a queue (dissipating)
  - Time taken to process through the queue: $t = \dfrac{\frac{L - q_0}{l} + m}{c}$
- Presence of a queue (blocked)
  - Time taken to reach the queue: $t = \dfrac{q - x_0}{v_u}$

Speed Model (II)

- Speed varies linearly from v_u (upstream speed) to v_d (downstream speed)



$$v(x) = \lambda x + v_u, \; 0 \leq x \leq L$$

$$\lambda = \frac{v_d - v_u}{L}$$

- v_u is computed based on the density of the moving part of the segment
- v_d is computed based on density of the moving part of the downstream segment (if there is no queuing) or is based on the queue discharge characteristics if there is queuing

Vehicle Movement Model (II)

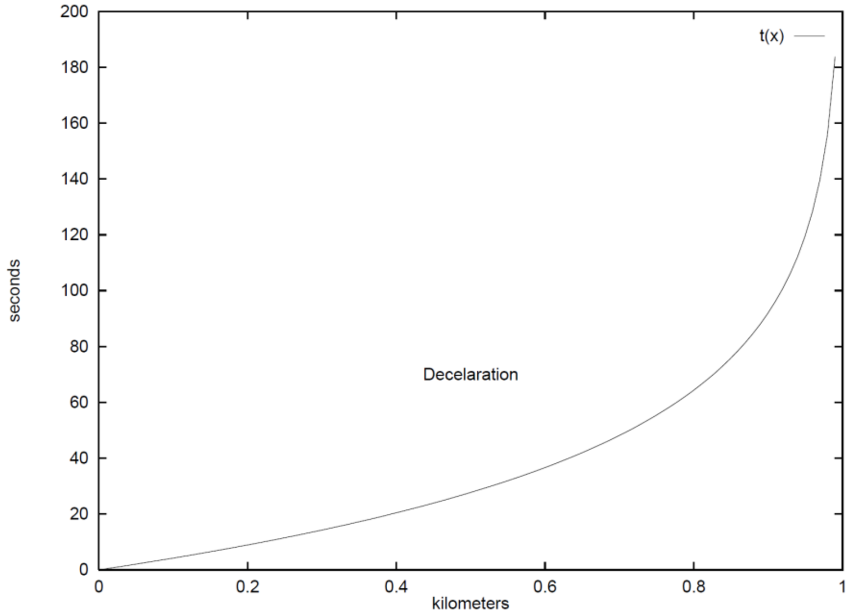
- Assume at time $t = 0$, the vehicle is at position $x = x_0$
- Vehicle will reach position x at time:

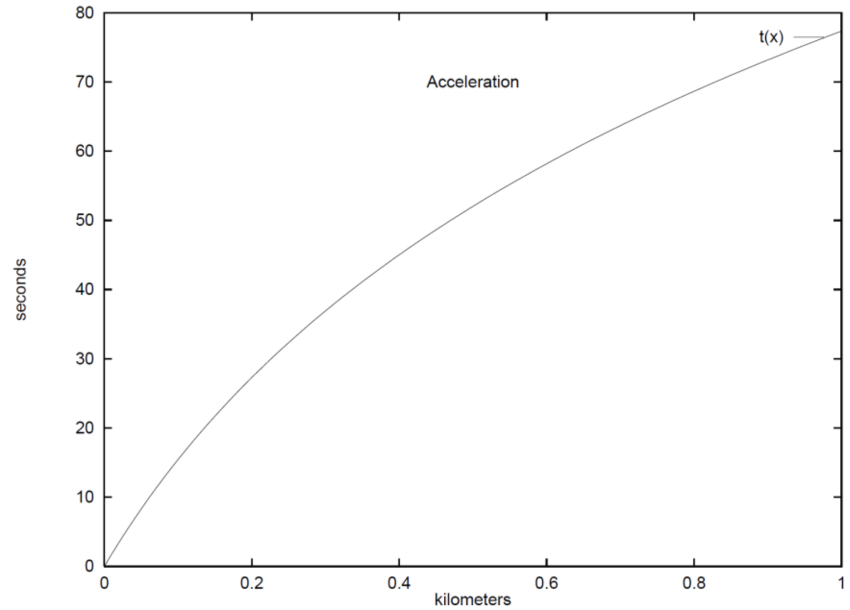
$$t(x) = \begin{cases} \frac{1}{\lambda} \log \frac{\lambda x + v_u}{\lambda x_0 + v_u} & \text{if } v_u \neq v_d \\ \frac{x - x_0}{v_u} & \text{if } v_u = v_d \end{cases}$$

- Position at time t:

$$x(t) = \begin{cases} e^{\lambda t}(x_0 + \frac{v_u}{\lambda}) - \frac{v_u}{\lambda} & \text{if } v_u \neq v_d \\ v_u t + x_0 & \text{if } v_u = v_d \end{cases}$$

Vehicle Movement Model (II)





$v_u = 90\ kmph; v_d = 20\ kmph$

$v_u = 20\ kmph; v_d = 90\ kmph$

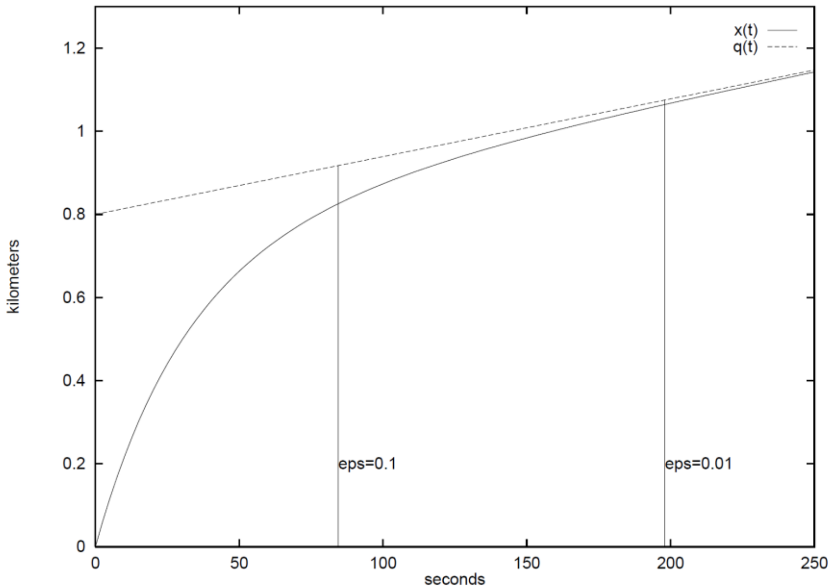
$$x_0 = 0; \quad L = 1\ km$$

Vehicle Movement Model (II)

- In the presence of a queue, the position of a vehicle at time t:

$$x(t) = e^{\lambda(t)t}\left(x_0 + \frac{v_u}{\lambda(t)}\right) - \frac{v_u}{\lambda(t)},$$

$$\lambda(t) = \frac{-v_u}{q_0 + l(ct - m)},$$



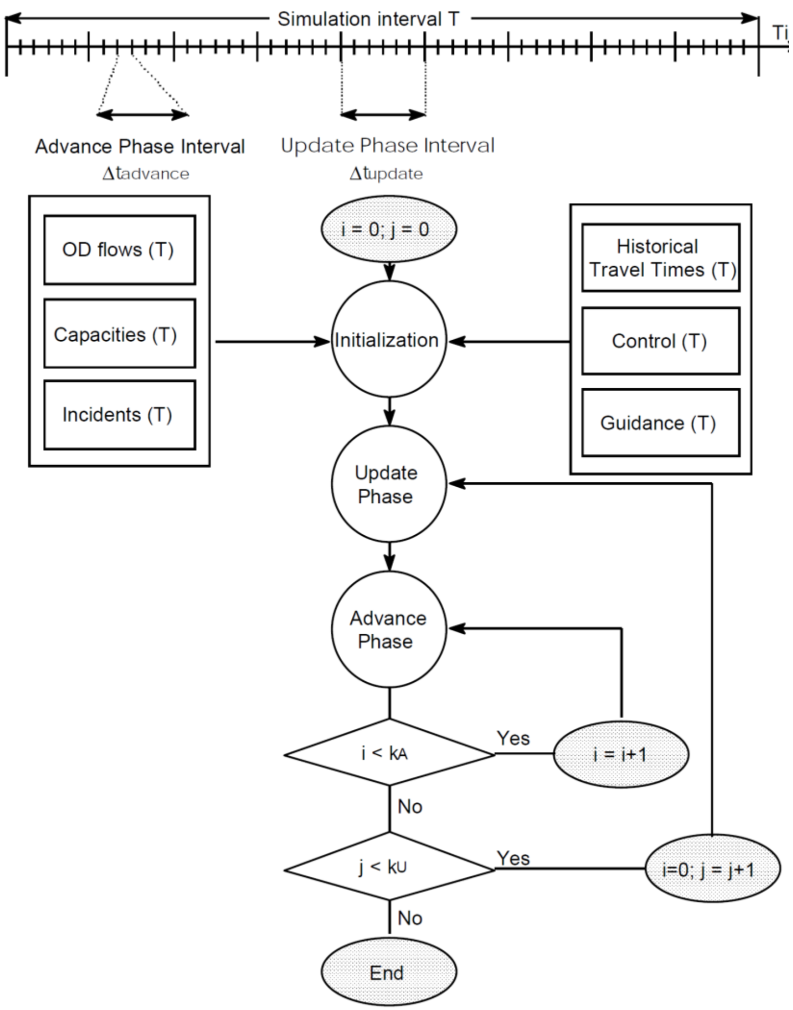
$$x(t) = \begin{cases} e^{\lambda(t)t}\left(x_0 + \frac{v_u}{\lambda(t)}\right) - \frac{v_u}{\lambda(t)} & \text{if } t < t^* \\ q_0 + l(ct - m) & \text{if } t \geq t^* \end{cases}$$

$v_u = 90\ kmph; v_d = 20\ kmph$
$x_0 = 0; m = 0; L = 1.2\ km;$
$q_0 = 0.8\ km; c = 1000\ vph$

Simulation Process

Overview

- Simulation of the traffic network operations proceeds in two phases:
  - Update Phase
  - Advance Phase.
- Update Phase is used for updating the traffic dynamics parameters (densities, speeds, etc.) during each simulation cycle
- Advance Phase is used for advancing the vehicles to their new positions at the end of the simulation cycle



Inputs

- Network Description
- Path Description
- Simulation Horizon
  - Simulation Interval
  - Update/Advance intervals
- Location of incidents; reduced capacities and interval
- List of Packets
  - Path
  - Departure Time

Outputs

- At the end of each update phase, the simulator reports the following data
  - Travel Times
  - Flows
  - Speeds
  - Densities
  - Queue Lengths
- Travel time is the time taken by a packet to traverse a link which are used to compute average link travel times
- Speeds and flows are reported for every sensor and every segment
- Densities and queue lengths are reports for every segment

Update Phase Algorithm

Initialization :

- Segment definition updated according to incident description.
- Output capacities of all segments are updated.

For each segment, compute:

- speed
- density

For each lane-group of the segment,

- initialize the output counter to the output capacity,
- initialize the input counter to the acceptance capacity.

Initialize i=0.

Description of one iteration :

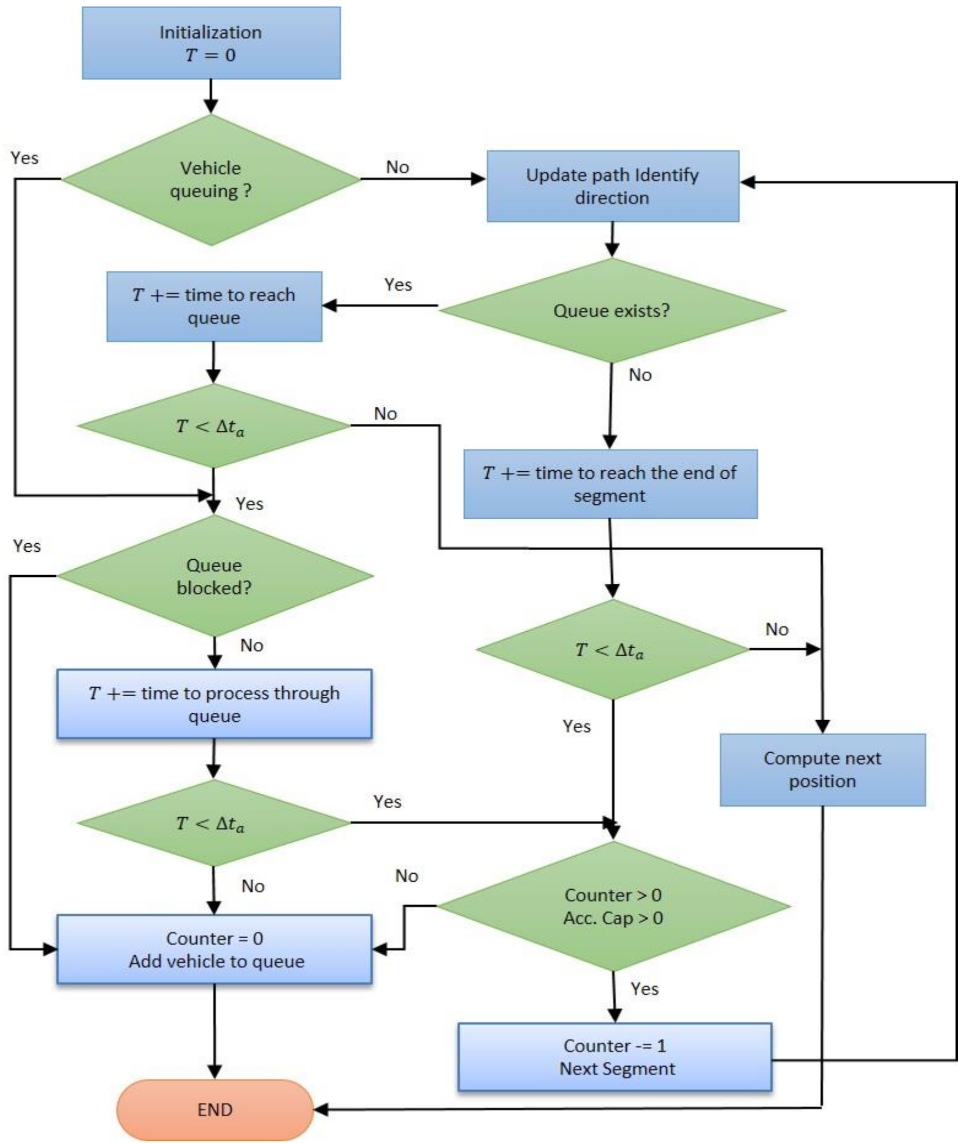
- Advance Phase algorithm is applied to the current interval
- Advance Phase counter is incremented: i := i + 1

Stopping criteria : The algorithm is stopped when all Advance Phase intervals have been processed, i.e. when i = kA.

Advance Phase Algorithm

- Consists of a loop over all vehicles
- Processing order of the vehicles is important; nodes are processed in the order defined by a topological sort of the network
- For a given node, we consider all incoming links, and sort all vehicles on these links according to their distance to the node, starting with the closest vehicle
- Each vehicle is then advanced, using the advance vehicle algorithm
- Following this all vehicles on virtual links are moved
- When all vehicles on the network have been processed, new vehicles are loaded using the Vehicle loading algorithm

Advance Vehicle



Advance Vehicle

- Vehicle Queuing ? If No, move to the next bullet
  - Is there sufficient output capacity; if not vehicle will continue to queue
  - If yes, will it have sufficient time to clear the queue ?
    - If yes, the *movetonextsegment* method is applied
    - If not, the vehicle continues to queue
- Queue Present ? If No, move to the next bullet

  Compute time to reach the queue

$$t_f = t_0 + \left(\frac{q - x_0}{v_u}\right)$$

  - If $t_f < \Delta t_{advance}$, then the vehicle will reach the end of the queue and join it
  - If $t_f \geq \Delta t_{advance}$, the vehicle moves for $\Delta t_{advance}$ seconds and its final position is given by $x_f = x_0 + (t_f - t_0)v_u$

Advance Vehicle

- Initial queue present ? If No, move to the next bullet

Compute

$$\begin{cases} t_A = t_0 + \left(\frac{m}{c}\right) + \left(\frac{L_{seg} - q_0}{cl}\right) \\ t_E = t_0 + \left(\frac{L_{seg} - x_0}{v_u}\right) \end{cases}$$

Time to process through the dissipating queue

q_0 - Queue length at the beginning of the advance interval

Time to reach the end of segment

Case 1: $t_A < t_E < \Delta t_{advance}$: Packet will not encounter queue and reach end of segment in time t_E

Case 2: $t_E < t_A < \Delta t_{advance}$: Packet reaches end of segment in time t_A after processing through the queue

Case 3: $t_E > t_A > \Delta t_{advance}$: Packet does not encounter queue and does not reach end of the segment (moves normally within segment)

Case 4: $t_E > \Delta t_{advance} > t_A$: Packet does not encounter queue and does not reach end of the segment (moves normally within segment)

Advance Vehicle

- None of the above ?

If none of the previous conditions are met, time taken to reach the end of the segment is

$$t_f = t_0 + \left(\frac{L_{seg} - x_0}{v_u}\right)$$

- If $t_f < \Delta t_{advance}$; If output capacity > 0, the *movetonextsegment* method is applied, if not it starts queuing
- If $t_f > \Delta t_{advance}$ it moves normally for $\Delta t_{advance}$ seconds

Advance Vehicle: Move to Next Segment

- Is there a next segment on the same link ?
  - If Yes, perform acceptance capacity checks; If AC checks fail queue on the current segment else move to the next segment
  - If No, check if the next link exists
    - If No, report link travel for the link and unload packet to the loader
    - If Yes, check if the next segment has been processed
      - If Yes, perform AC checks ; If AC checks fail queue on the current segment
      - If No, Check if the virtual queue associated with the downstream link has capacity
        - If Yes, add vehicle to the virtual queue
        - If No, queue on current segment

Advance Vehicle: Acceptance capacity check

- For an arbitrary segment *j*, the acceptance capacity is modelled in three ways
  - Physical Space

$$\left(N_j^{mov} + N_j^{que}\right) < \frac{L_j^{seg} \times n_j^{Lanes}}{l}$$

  - Acceptance Rate (Lane group *k*)

$$R_{accept} = \max\left(\frac{l}{n_{Lanes}^{k} \times v_u^k}, \frac{\delta^k}{c^k}\right)$$

  - Size of Virtual Queue in Virtual Link Processing

N_j^{mov} - Number of moving vehicles

N_j^{que} - Number of queuing vehicles

L_j^{seg}, n_j^{Lanes} - Segment Length, number of lanes

δ^k=1 if there is queuing on LG *k*

c^k - Capacity of LG *k*

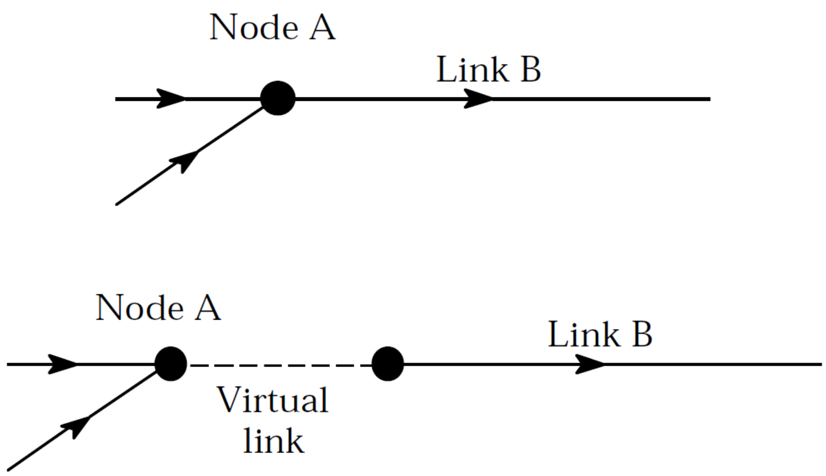
v_u^k - Speed of moving part on LG *k*

Vehicle Loading

- Loading of vehicles is controlled by two quantities
  - $rate = \frac{1}{c}$, where *c* is the output capacity of the loader, rate gives the time between loading of two vehicles as determined by the output capacity of the loader
  - $physicalRate = L_{veh}/v_u$ where *physicalRate* is the minimum time separation between vehicles as dictated by the time it takes the vehicle to move a distance equal to its length along a segment
- $releaserate = \text{MAX}(rate, physicalRate)$
- *Releaserate* determines the time at which the next vehicle can be loaded provided there is sufficient space on the segment to be loaded

Virtual Processing

- When a vehicle is moved to a new link it is assumed that all vehicles on this link have already been processed. This assumption is not always satisfied. The concept of a virtual link is used to deal with these situations.



- Assume that in the figure vehicles are processed through node A but vehicles on link B have not been processed. In that case the vehicles that leave node A are added to the virtual link and moved only after vehicles on link B have been processed. Virtual link capacity is set to the capacity of the upstream segment of the link at the beginning of each update phase

THANK YOU