ICAPS Summer School 2018: Sequential Optimization for Traffic Signal Control

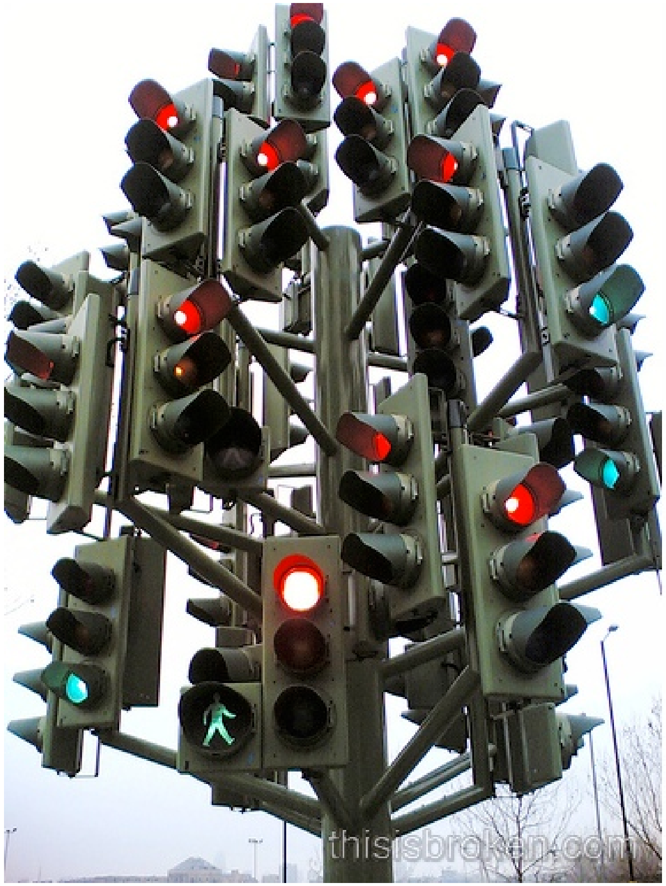
Scott Sanner

UNIVERSITY OF TORONTO

QTM slides courtesy of Iain Guilliard (ANU) and
Surtrac slides courtesy of Steve Smith (CMU)

Optimal Traffic Signal Control

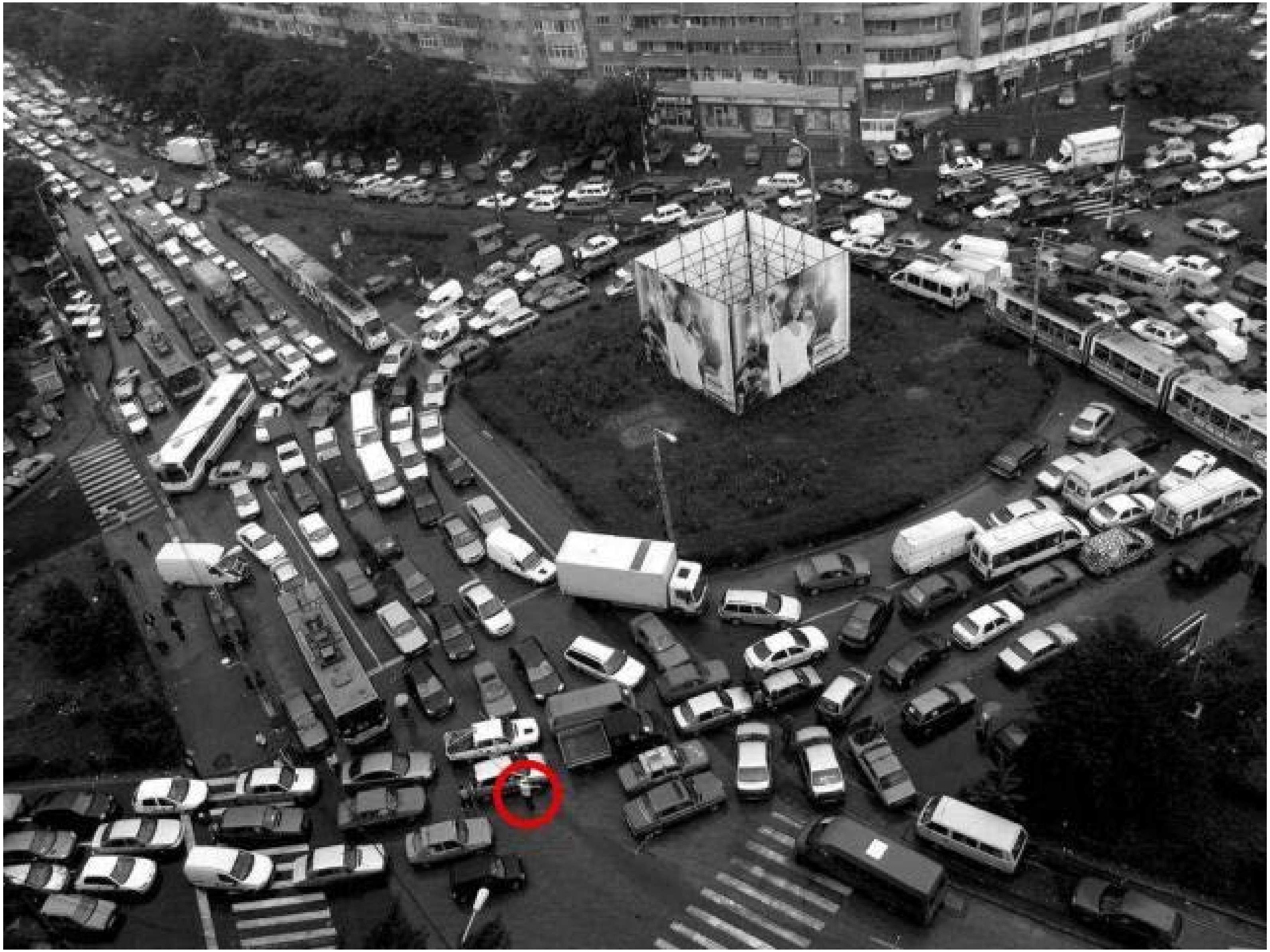
- Motivation
- Existing Approaches
  - Practice
  - Theory
- New approaches
  - QTM MILP optimization
  - Surtrac Scheduling



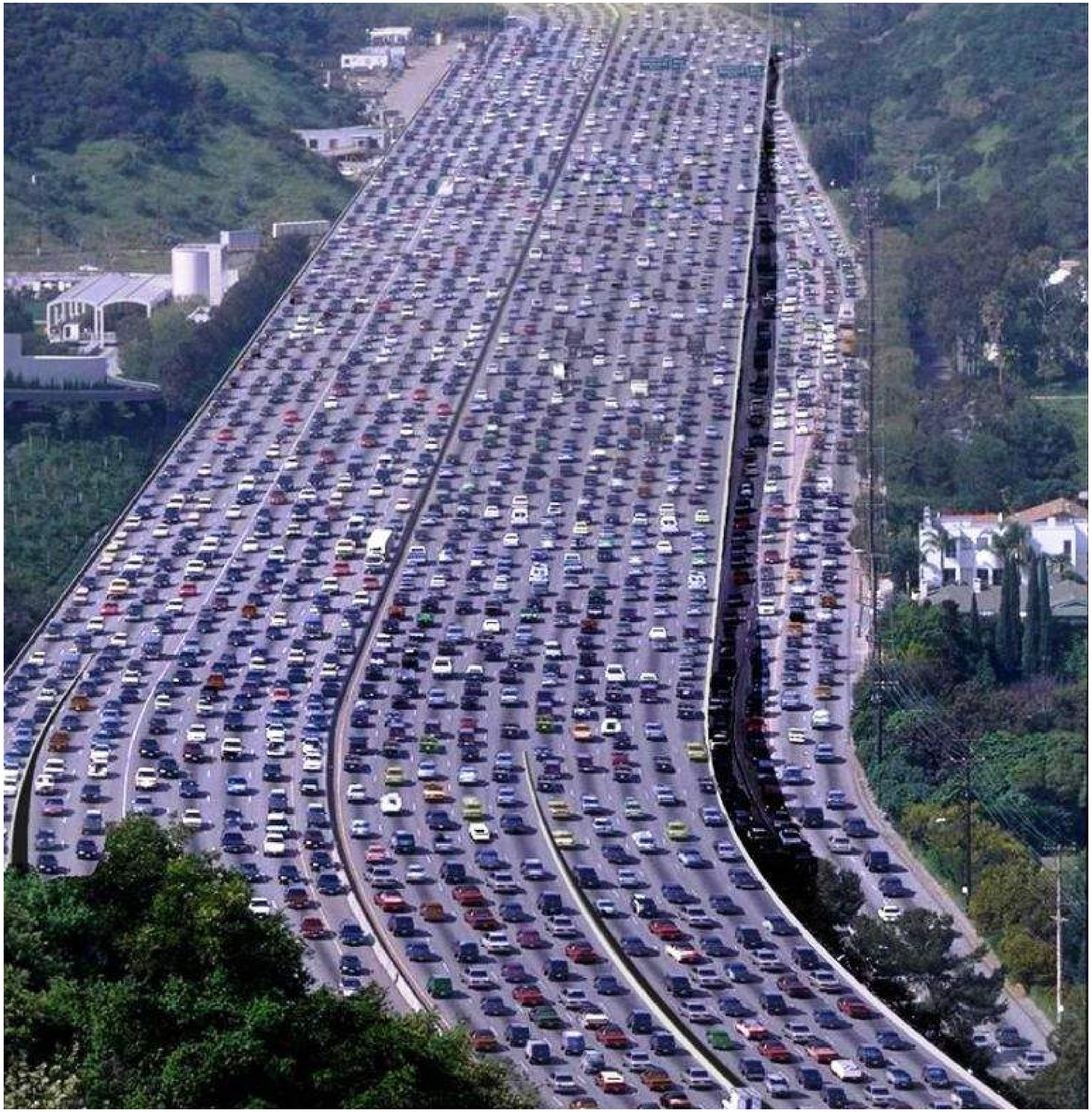
- Frontiers: connected and autonomous vehicles

Motivation

More Motivation



Unreal Motivation

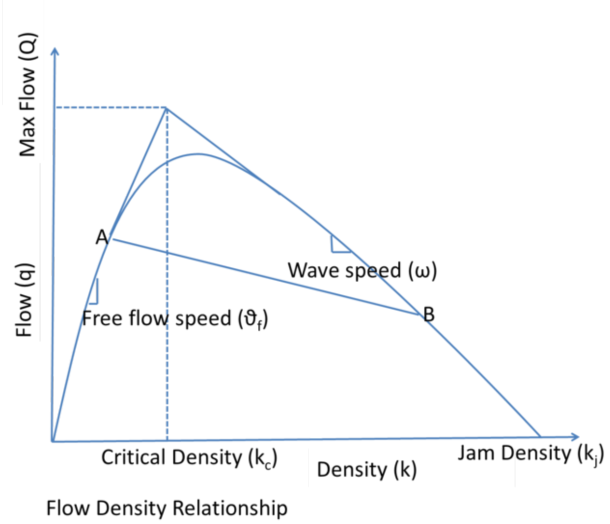


Traffic Impacts Everyone

- Not a problem I have to motivate
  - Economically, impact of better control is in billions of $$$ for large cities!
- Real & unsolved problem
  - Multidimensional state (integer / continuous)
  - Multidimensional concurrent actions
  - Stochastic
  - Building a high fidelity model is difficult
  - Optimizing it is just as hard

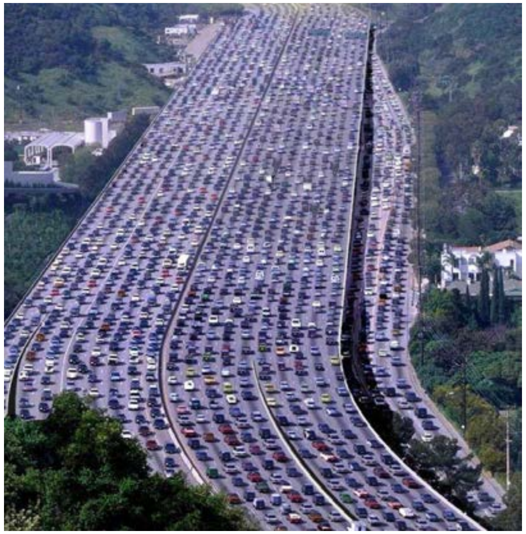
Theory vs. Practice

- Theory
  - Idealized
  - Models major phenomena
  - Good analytical techniques



Need a stronger connection!

- Practice
  - Control is rule-based
    - No models or optimization
  - Manually tuned



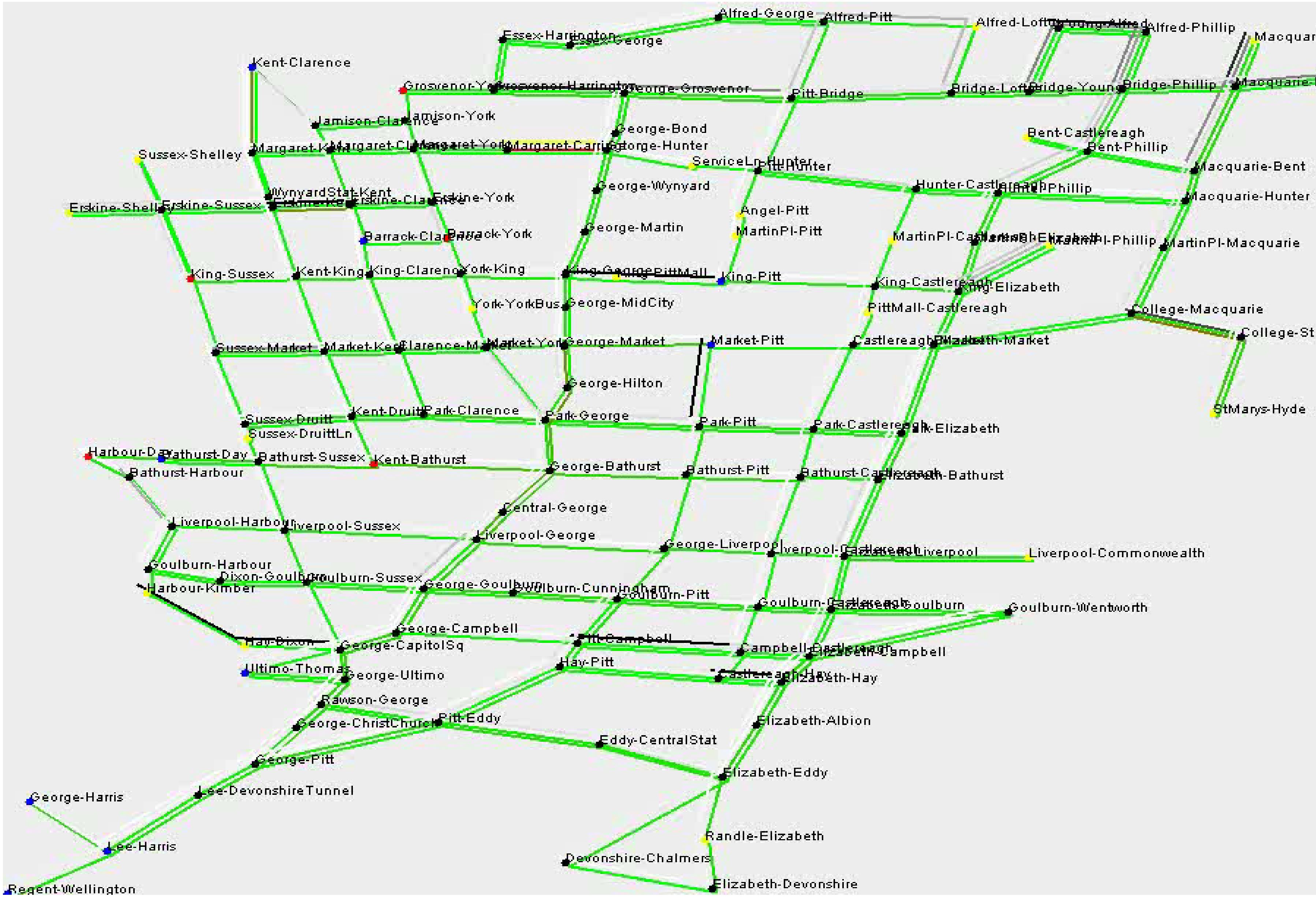
Practice: It's worse than you thought

- Billions of $$$ in **legacy** infrastructure
- Systems are **safety verified**
  - Difficult and expensive to replace
  - Figure out where to fit in for lowest cost
- Hardware/software limited, e.g., **1970's** era:
  - PDP-11 assembly on PDP-11 simulators!
  - 300 baud rate of infrastructure communication
  - Day divided into four time periods
    - Morning rush, mid-day, evening rush, other
    - Software allows four plan variations per period

Massive Opportunity for Change

- Not only is existing technology rooted in 70's era
  - But methodologies are often pre-70's
  - Data collection via human surveys
  - Flow modeling makes strong assumptions
    - Static Nash equilibrium (Wardrop and Whitehead, 1952)
  - Predictions often not validated against flow data
    - Gravity model!
- But now we collect and store masses of data!

And we have tons of data!



Vision: Optimized Traffic Control

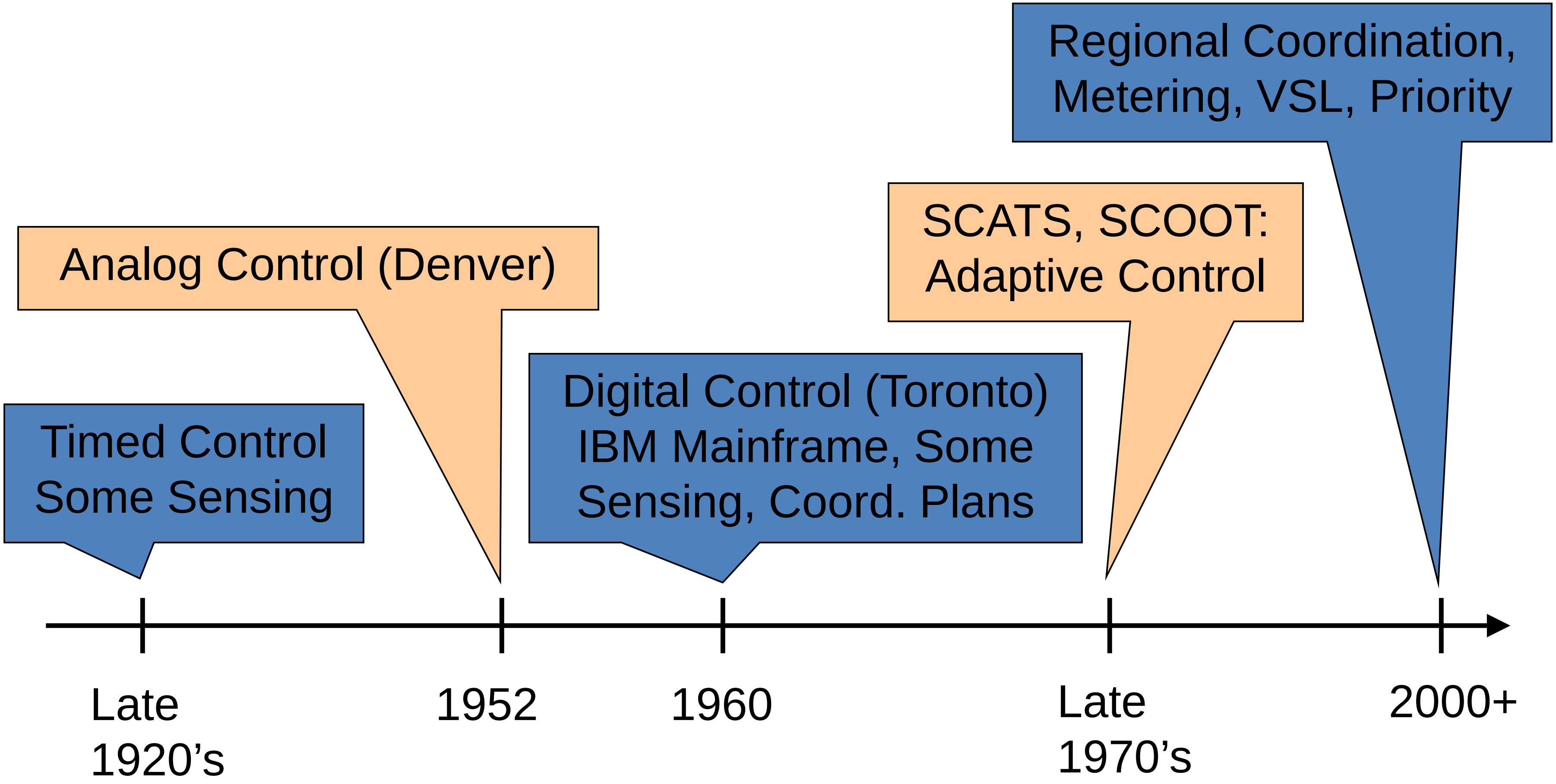
- Use predictive traffic model
  - Models traffic well based on existing theory
  - Ideally model parameters are learned from data
- Optimize future signals to maximize traffic flow (i.e., replan every 5 seconds wr.t. current state/model)
  - Use the online learned model for prediction
  - Use a MILP to optimally solve for signal changes

But first…

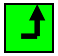
- We need to understand traffic flow modeling
- And existing methods for signal control
  - In practice
  - In theory
- What’s wrong with existing work?
  - We’ll see…

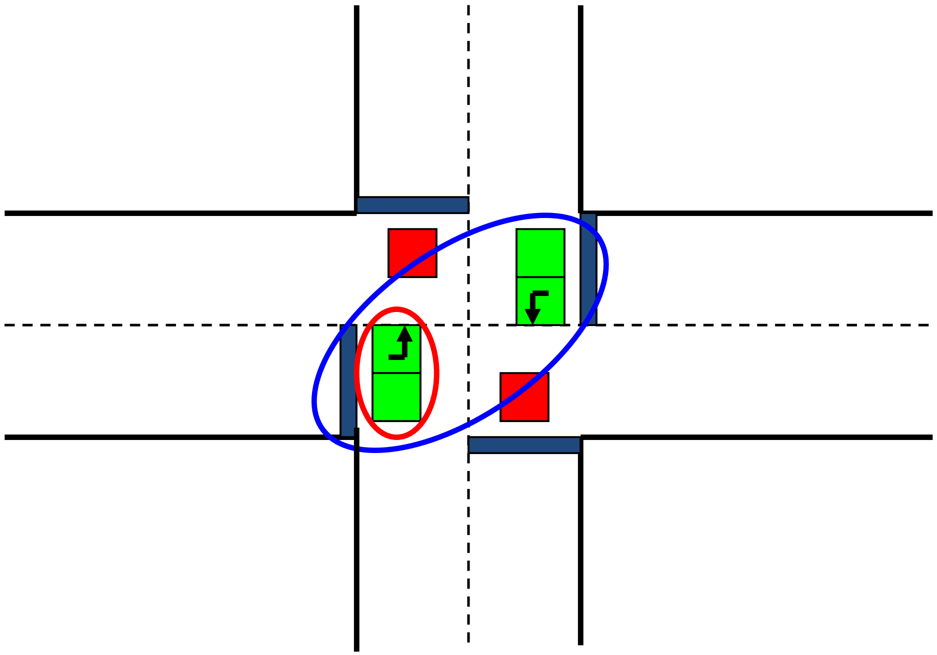
Traffic Control: In Practice

Signalized Control Timeline

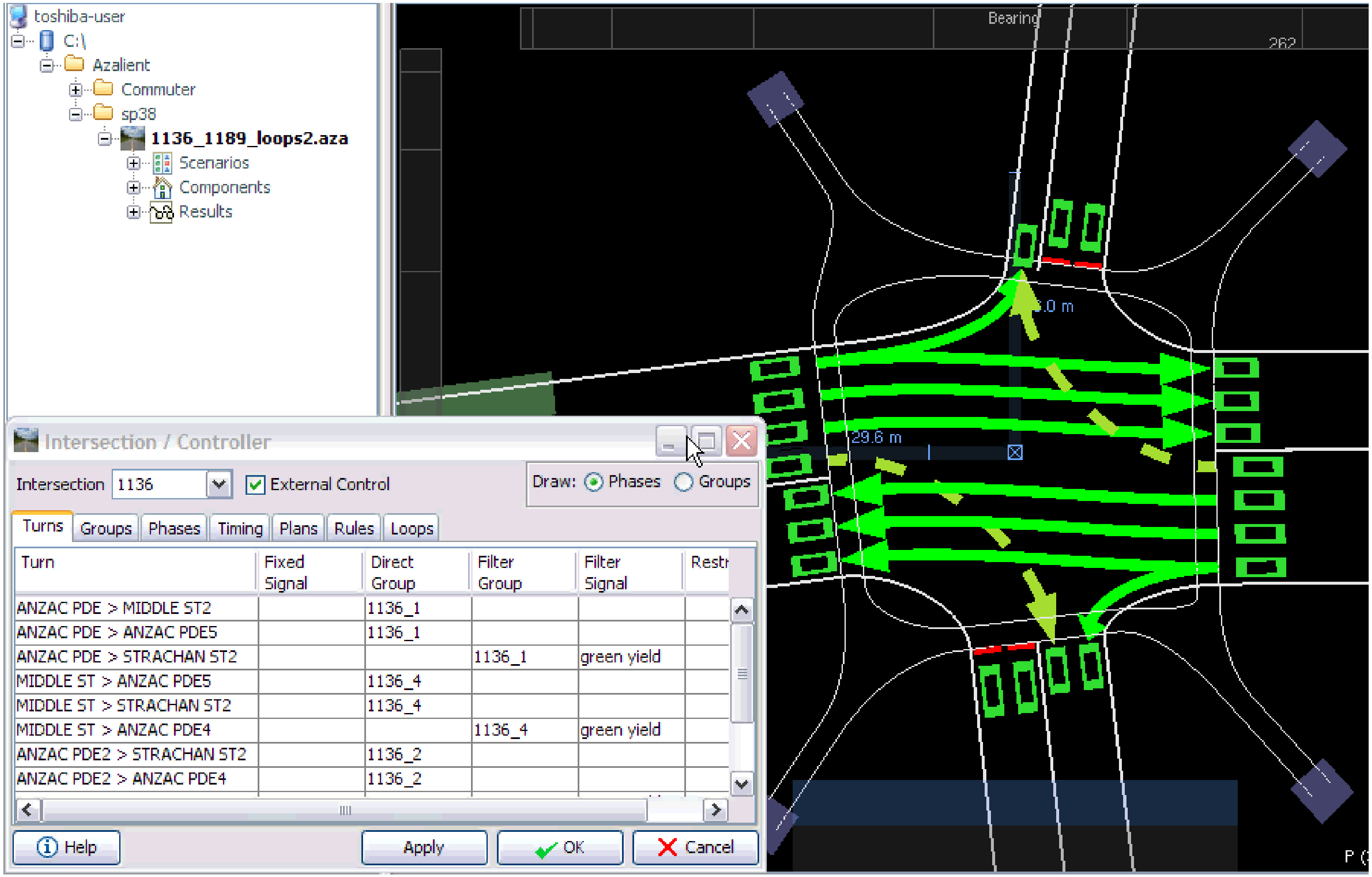


Terminology

- Signal, e.g., 
- Signal Group
- Phase
- Turns
  - Protected Turn
  - Filter Turn
    - unprotected

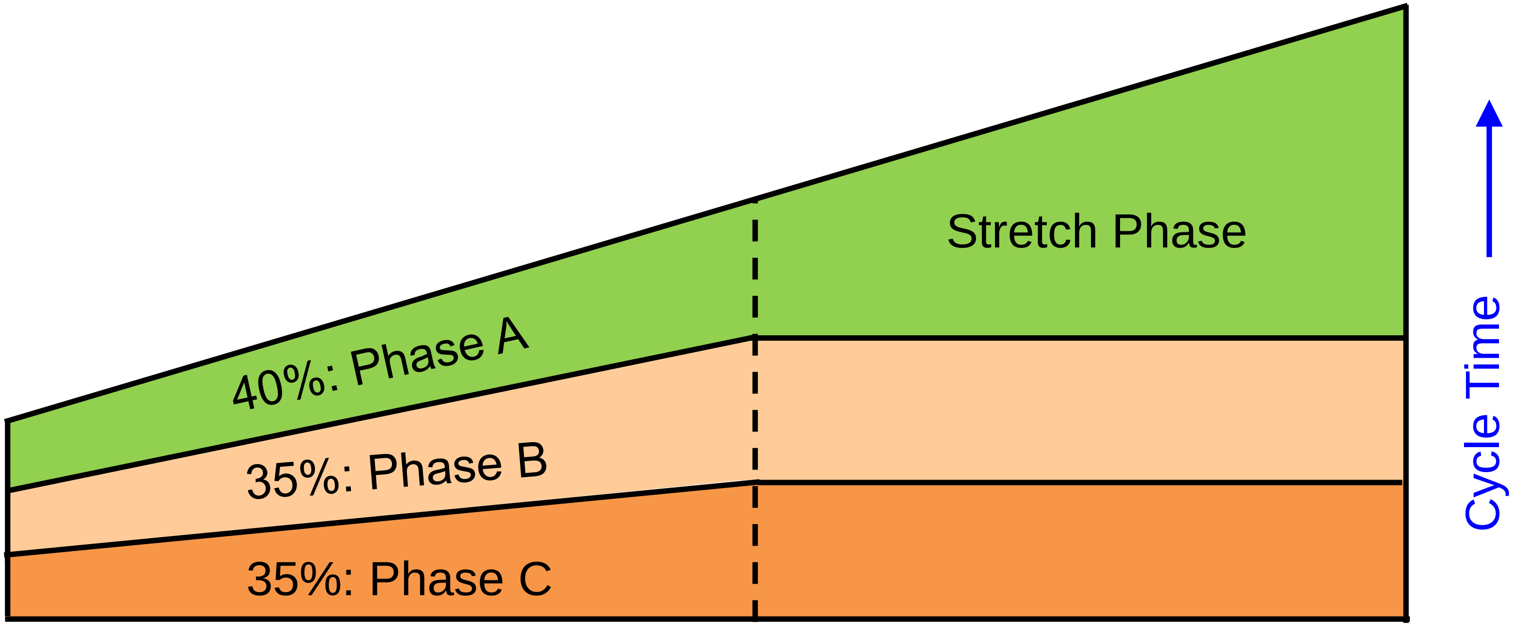


Phase Illustration in Commuter

- 

SCATS Phase Plans

- Each intersection has one or more **phase plans**
  - Each **phase** gets a **split** of the **cycle time**

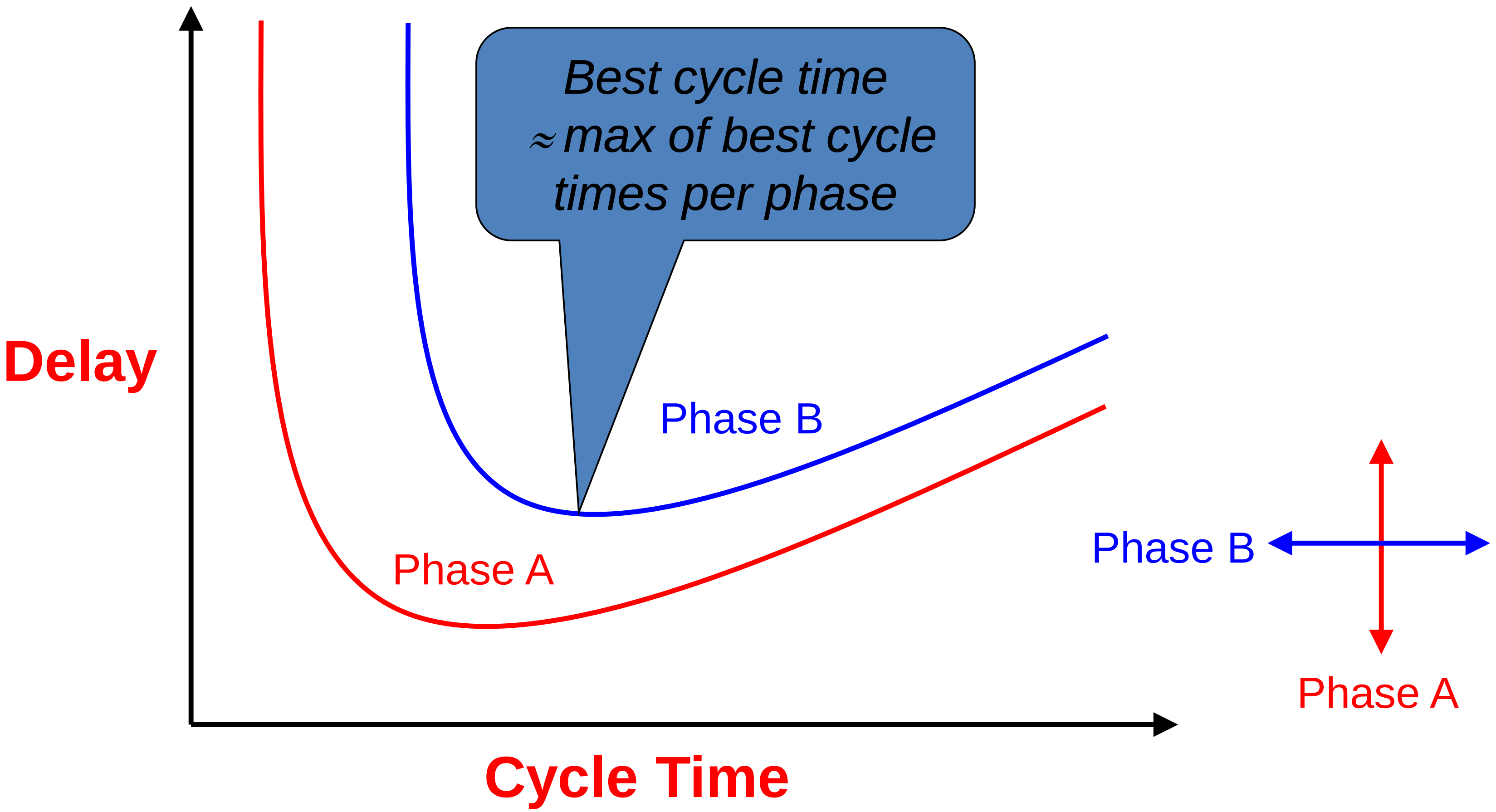


- Typically four plans per intersection
  - Heavy inbound / outbound, balanced, & light

Now just choose a plan and cycle time for one intersection!

Delay vs. Optimal Cycle Times

- Use maximum best cycle time of any phase



Optimal Cycle Times vs. Flow

- Light traffic
  - Short cycle times
  - Minimize delay for individual cars
- Heavy traffic
  - Long cycle times
  - Maximize steady-state flow

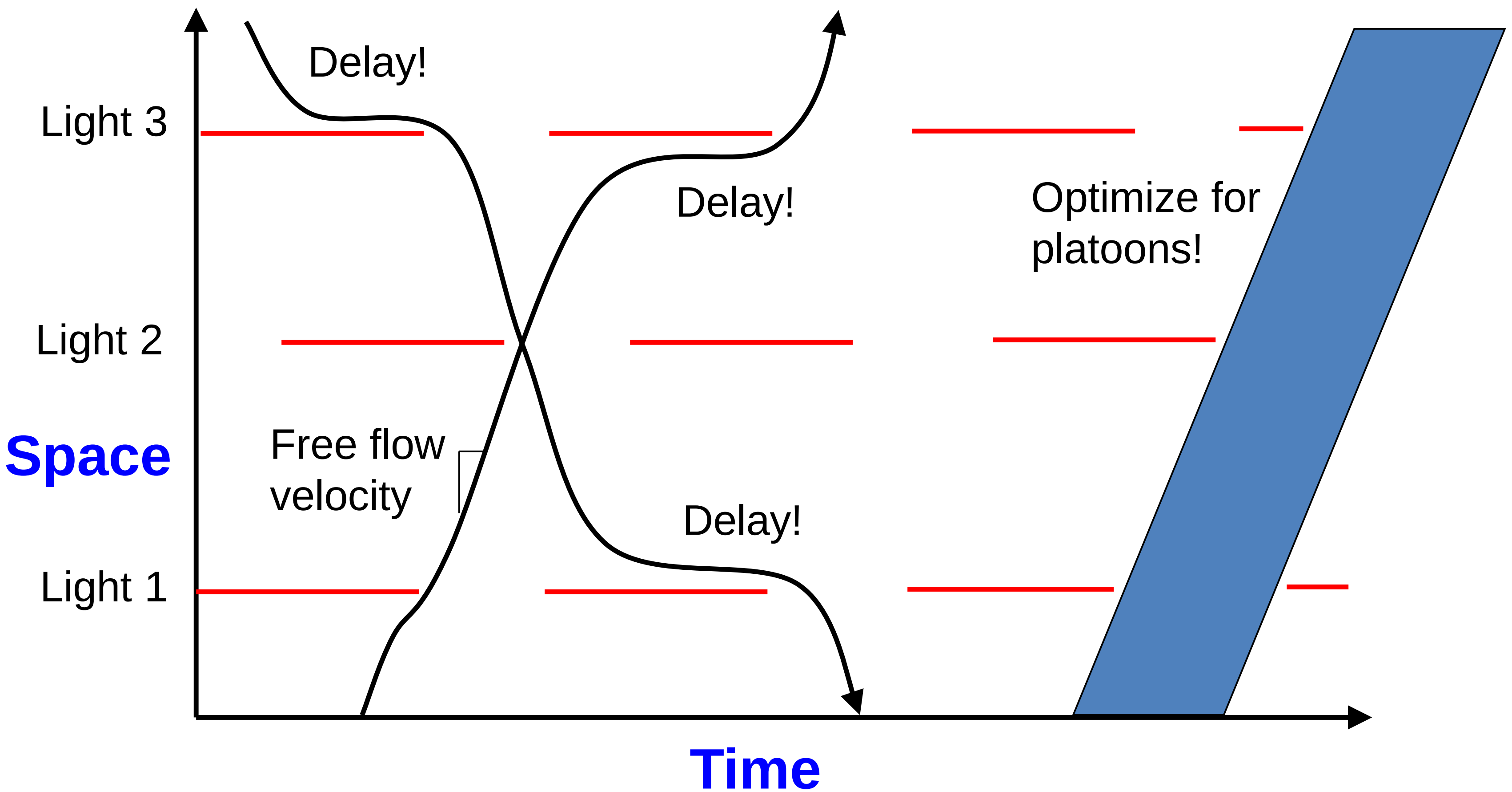
Problems with Local Control

- Intersections are not independent
  - In-flow of cars q_i is **not uniformly distributed!**
- Platoons

  - Cars tend to "clump" into platoons
    - Due to discharge from upstream queues
  - Best throughput with good platoon management
    - Careful timing needed

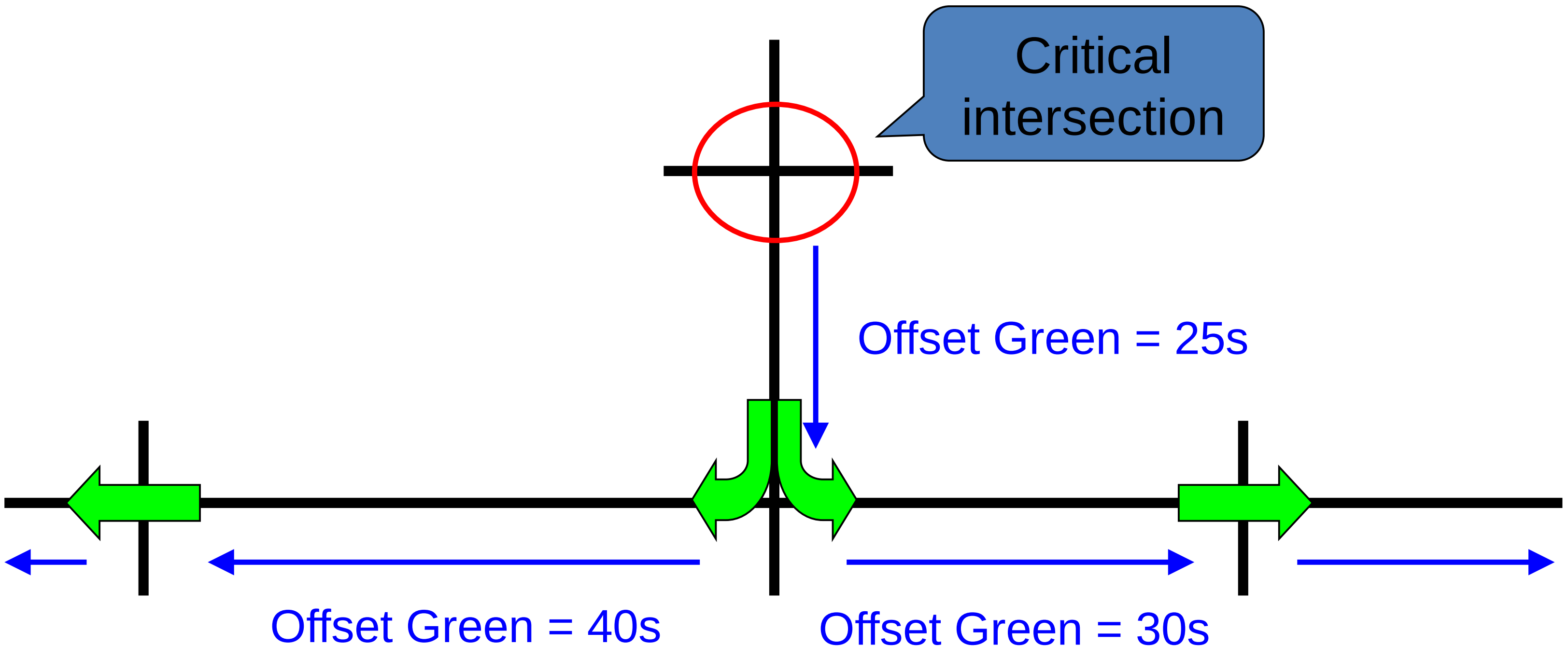
Multi-intersection Control

- Optimize phase offsets for platoon throughput:



Master/Slave Offset Control

- Fix timing offsets from critical intersections
  - Allows platoons to pass in dominant flow direction



Multi-intersection Control in Practice

- Split, Cycle, Offset Optimization (SCOOT, SCATS)
  - Decide on synchronized intersections
  - Decide on intersection offsets
    - Based on dominant flow direction
  - Decide on phase splits
    - W.r.t. offset constraints
    - Rules to modulate splits by observed flow
- Practical, but rule-based and very heuristic
  - Room for data-driven modeling & optimization!

That was practice… let's take a more theory driven approach

Traffic Theory: Modeling

Fundamental Diagram of Traffic Flow

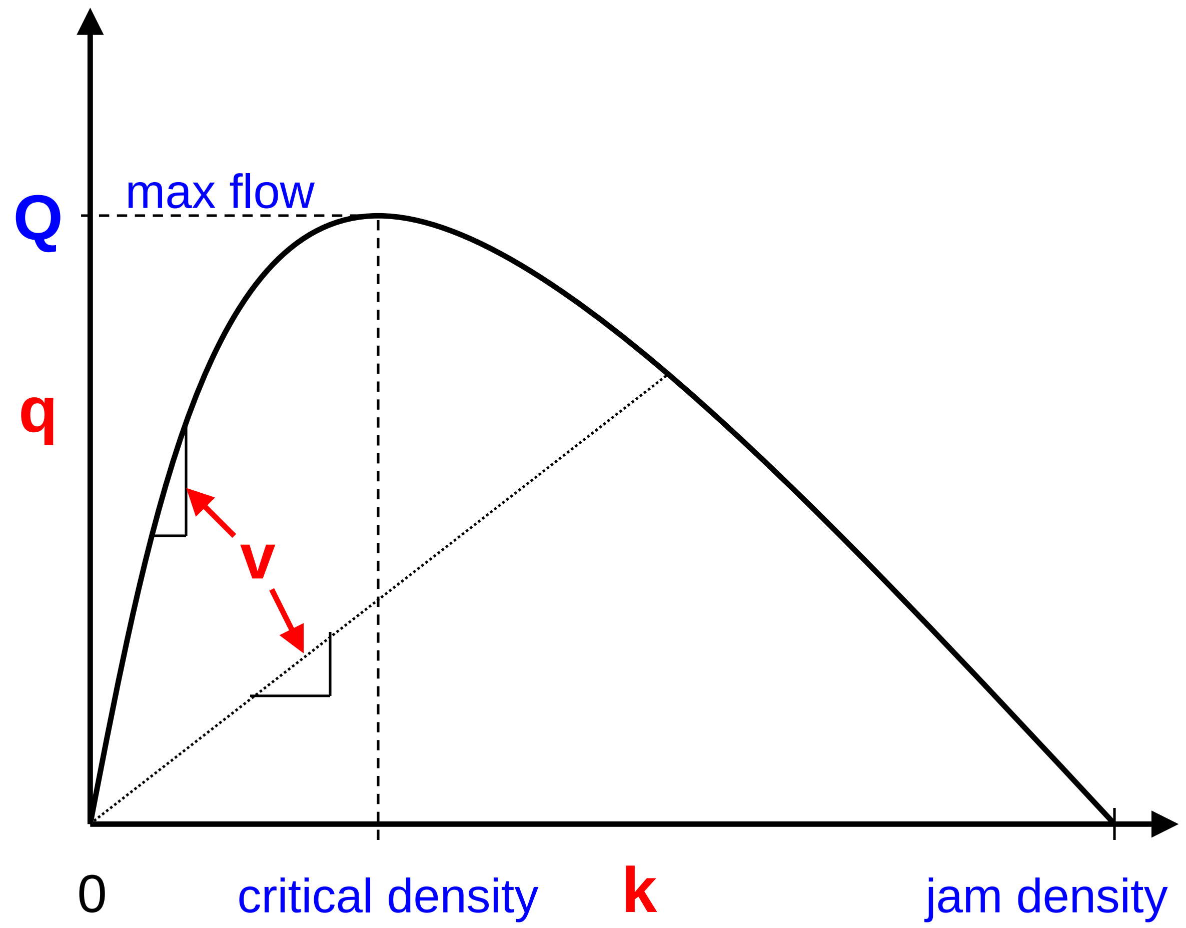
Flow q: cars/s

Density k: cars/m

Velocity v: m/s

q = kv

v = q/k



Types of Models

- Macrosimulation
  - Model aggregate properties of traffic
  - Average flow, density, velocity of cells
- Microsimulation
  - Model individual cars
  - Typically cellullar automata
- Nanosimulation
  - Model people (inside & outside of cars)

Human Factors in Microsimulation

- Microsimulation often involves driver choice:
  - Filter turns
  - Turns into flowing traffic
  - Lane merges
  - Lane changes

- Theories such as gap acceptance theory
  - Attempt to explain driver choices
  - e.g., gap size willing to accept on filter turn $\propto$ 1/time

- See Ch. 3 of Traffic-Flow Theory, Henry Lieu

Microsimulation Turn Models

Two ways to model turns:

1. Turn probabilities at each intersection

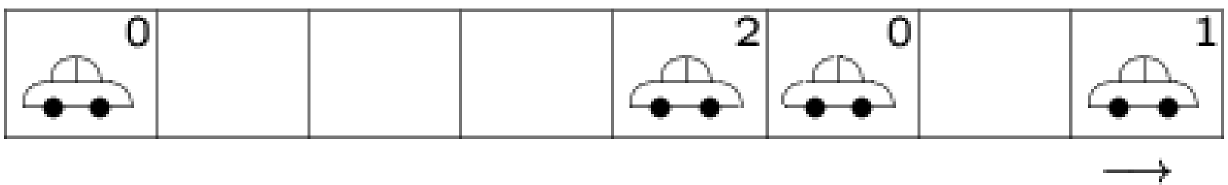
2. Frequencies in origin-destination (OD) matrix
   (routes predetermined for each OD pair)

Which is better?

Car may go in loops for 1, more realistic to choose 2!

Microsimulation

- Nagle-Schreckenberg
  - Cellular Automata Model
    - nominally each cell is 7.5m in length



  - Simplest model that reproduces realistic traffic behavior

Image and description from: http://www.thp.uni-koeln.de/~as/Mypage/traffic.html

Car Following in Microsimulation

- Nagel-Schreckenberg
- 4 Rules
  - Acceleration:
    $v_i := \min(v_i + 1, v_{max})$
  - Safety Distance:
    $v_i := \min(v_i, d)$
  - Randomization:
    prob p: $v_i := v_i - 1$
  - Driving:
    $x_i' = x_i + v_i$

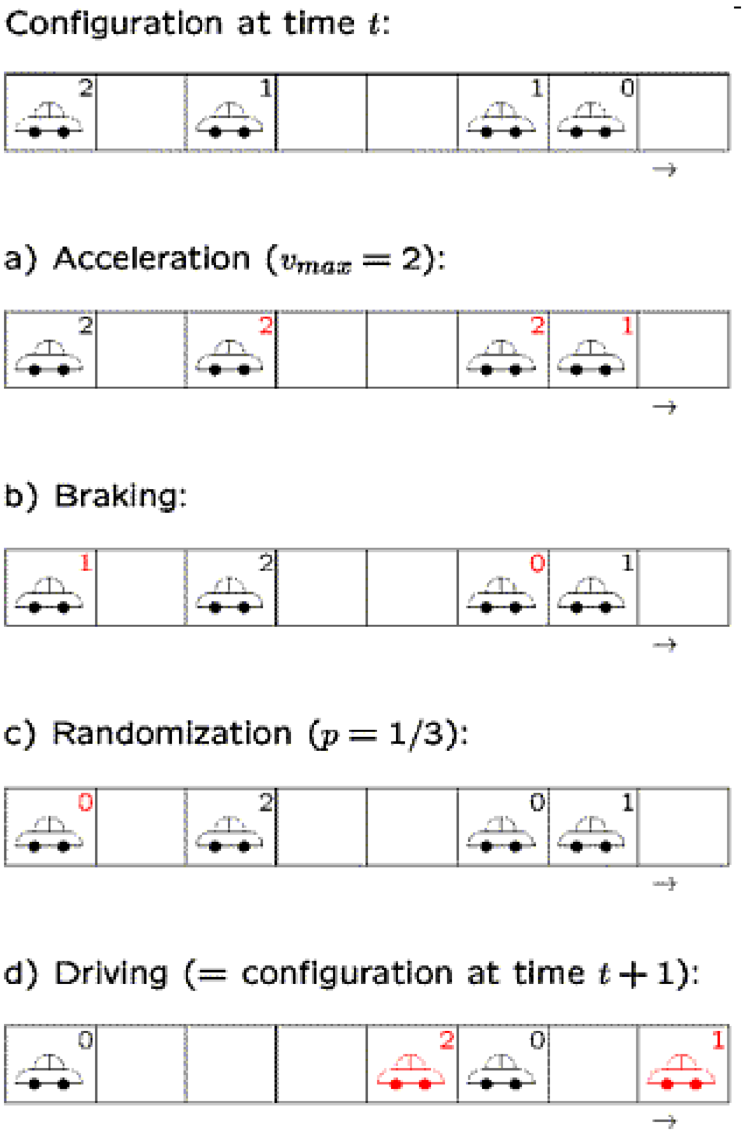
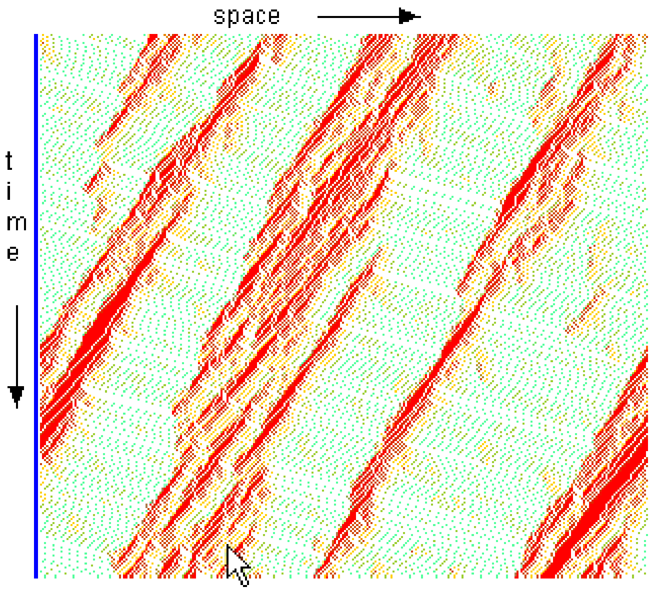
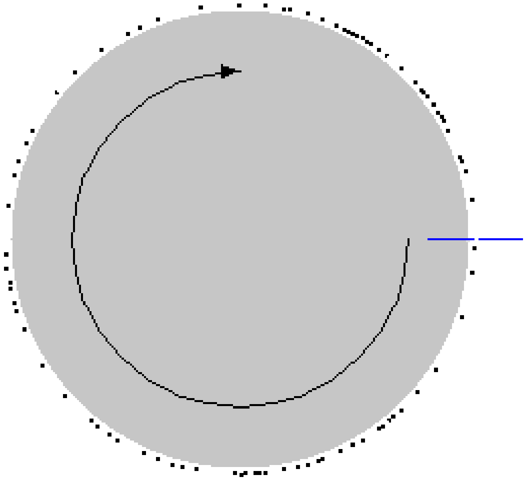


Image and description from: http://www.thp.uni-koeln.de/~as/Mypage/traffic.html

Car Following Microsimulation

http://www.thp.uni-koeln.de/~as/Mypage/simulation.html





- Continuous traffic flow example:
  - Upper plot is space/time diagram
  - Lower plot is actual traffic

High fidelity online simulation available at http://www.traffic-simulation.de/

Microsimulation Software

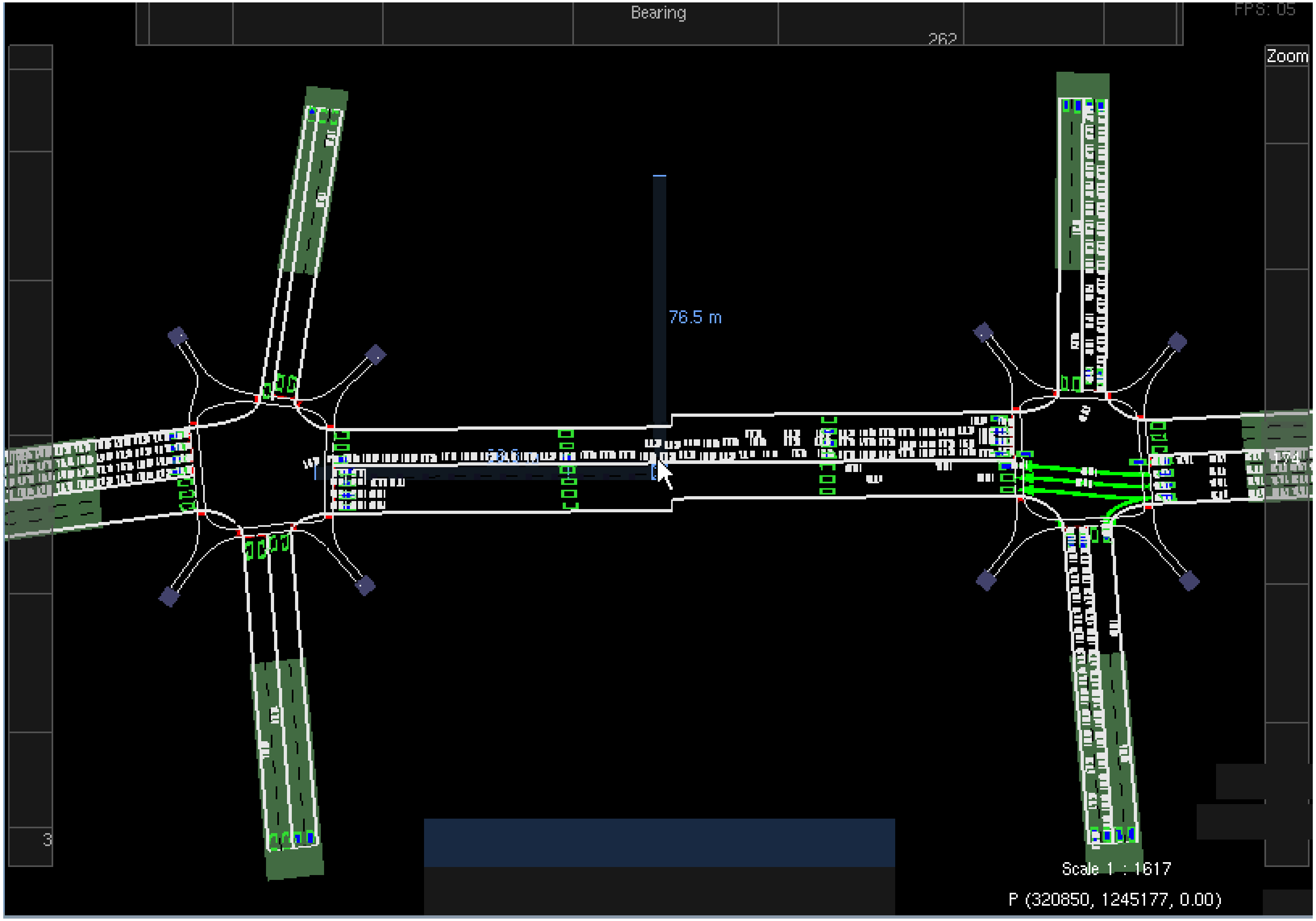
- Quadstone Paramics
  - Largest market share, $$$
  - Industrial strength, fast simulator

You most likely won't be able to test your traffic control tools in the real world, so microsimulation is the only way to test.

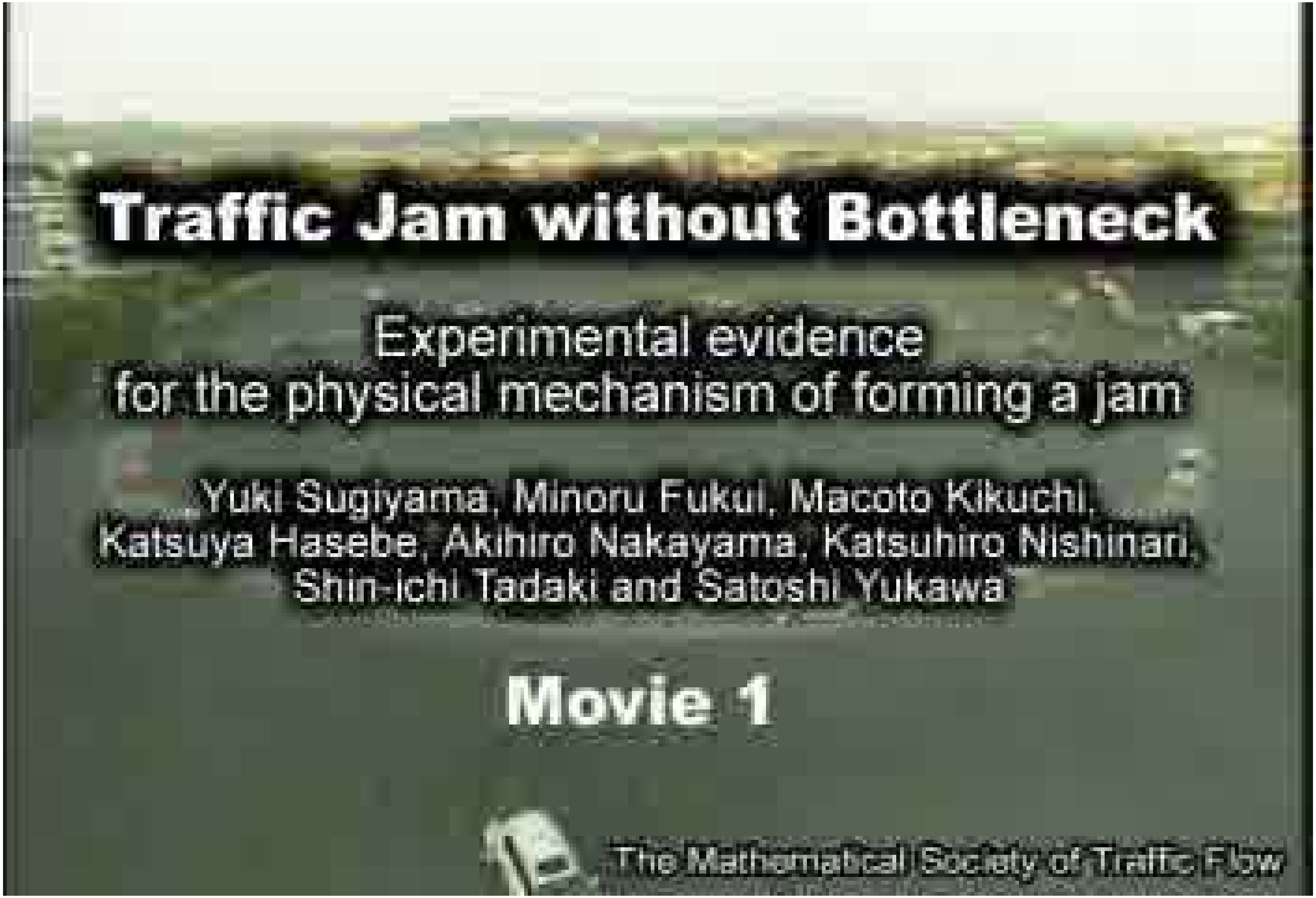
- Vissim
  - Highly used, $$$
  - Can model a variety of path-based user behavior

- SUMO
  - Free
  - Can download maps directly from OpenStreetMap

Microsimulator Example



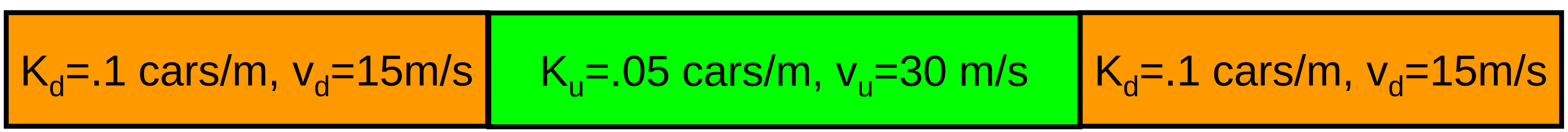
An Even Better Microsimulator



https://www.youtube.com/watch?v=Suugn-p5C1M

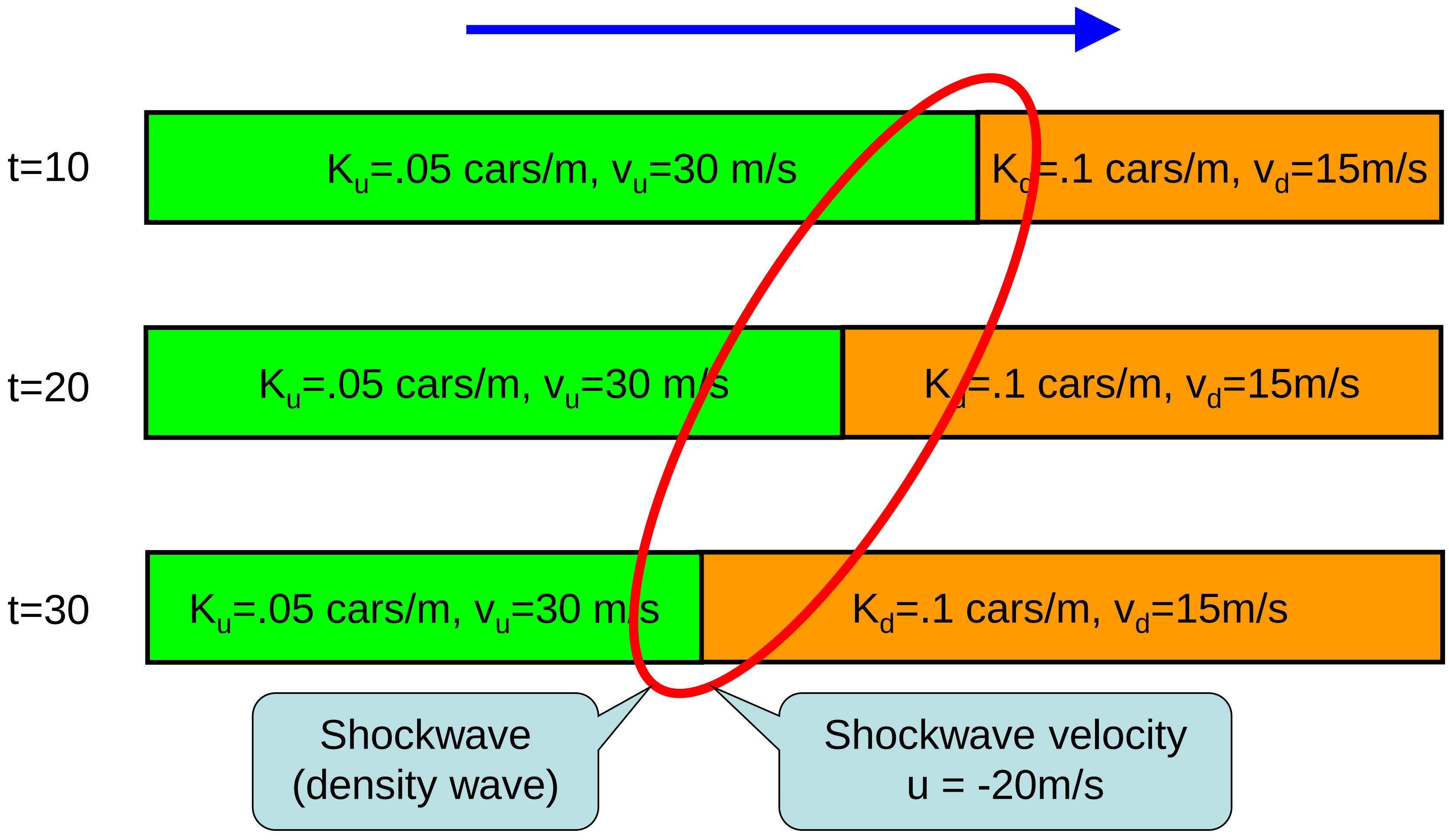
But microsimulation is difficult for real-time control

Ideally would like some form of closed-form **macro-model**



Shockwaves in Macro Models

- Low density traffic meets high density traffic…



Calculation of Shockwave Speed

- Law of conservation of cars:
  - "Cars can neither be created nor destroyed"
- Traffic flows in/out of shockwave at rate:

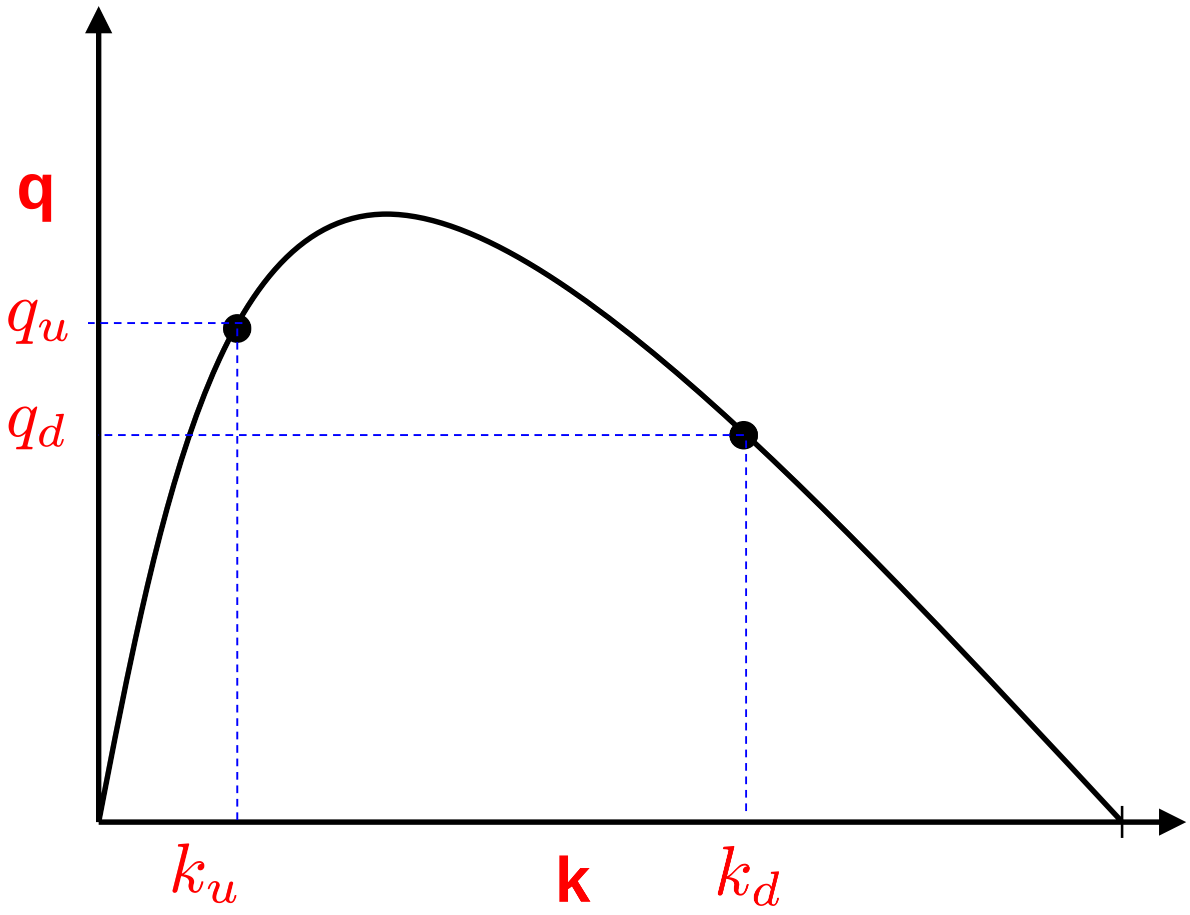
$$q_{enter} = k_u(v_u - u)$$

$$q_{exit} = k_d(v_d - u)$$

$$q_{exit} = q_{exit} \Rightarrow u = \frac{k_d v_d - k_u v_u}{k_d - k_u} = \frac{q_d - q_u}{k_d - k_u} = \frac{\Delta q}{\Delta k}$$

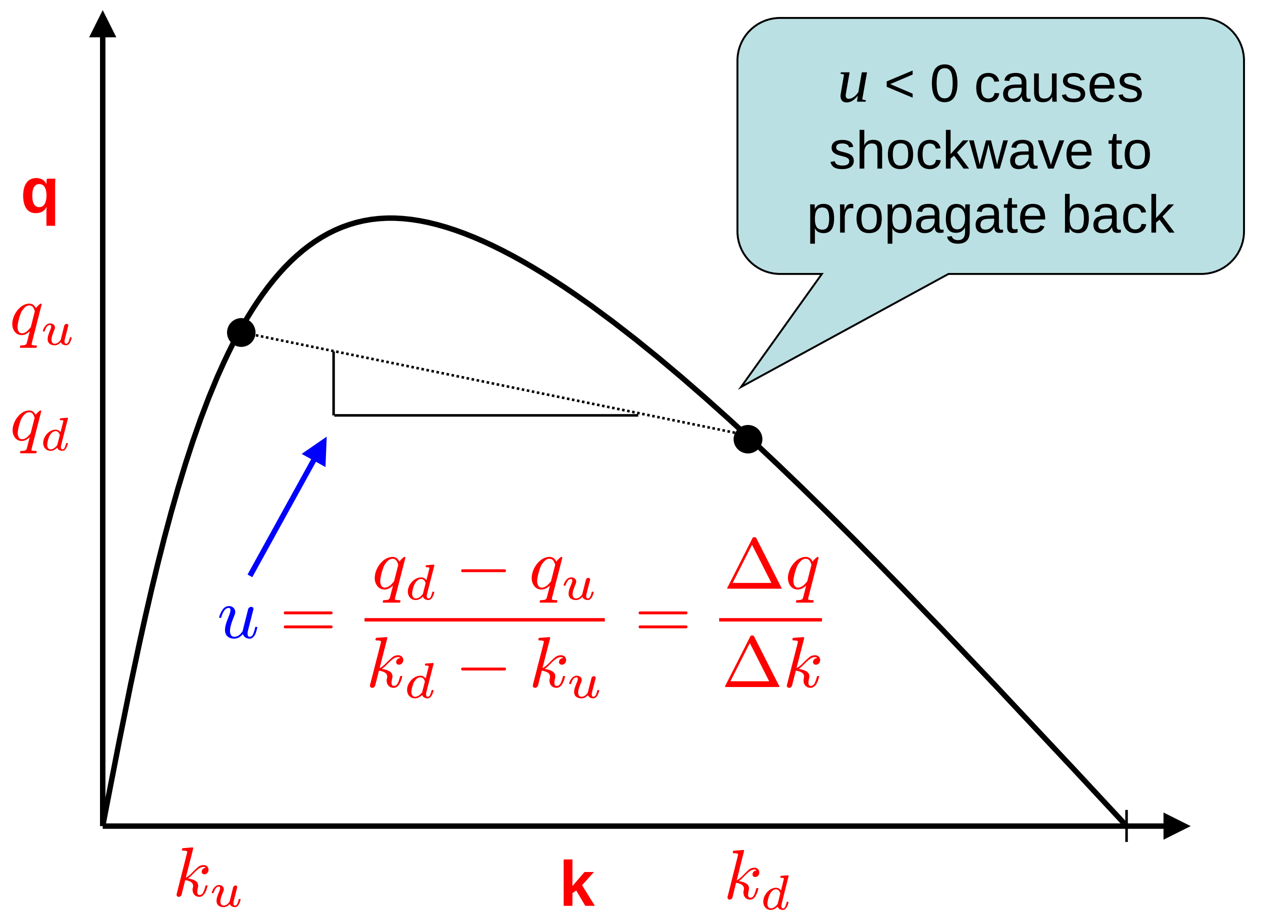
Theory of Shockwaves

Determine shockwave speed u from diagram:



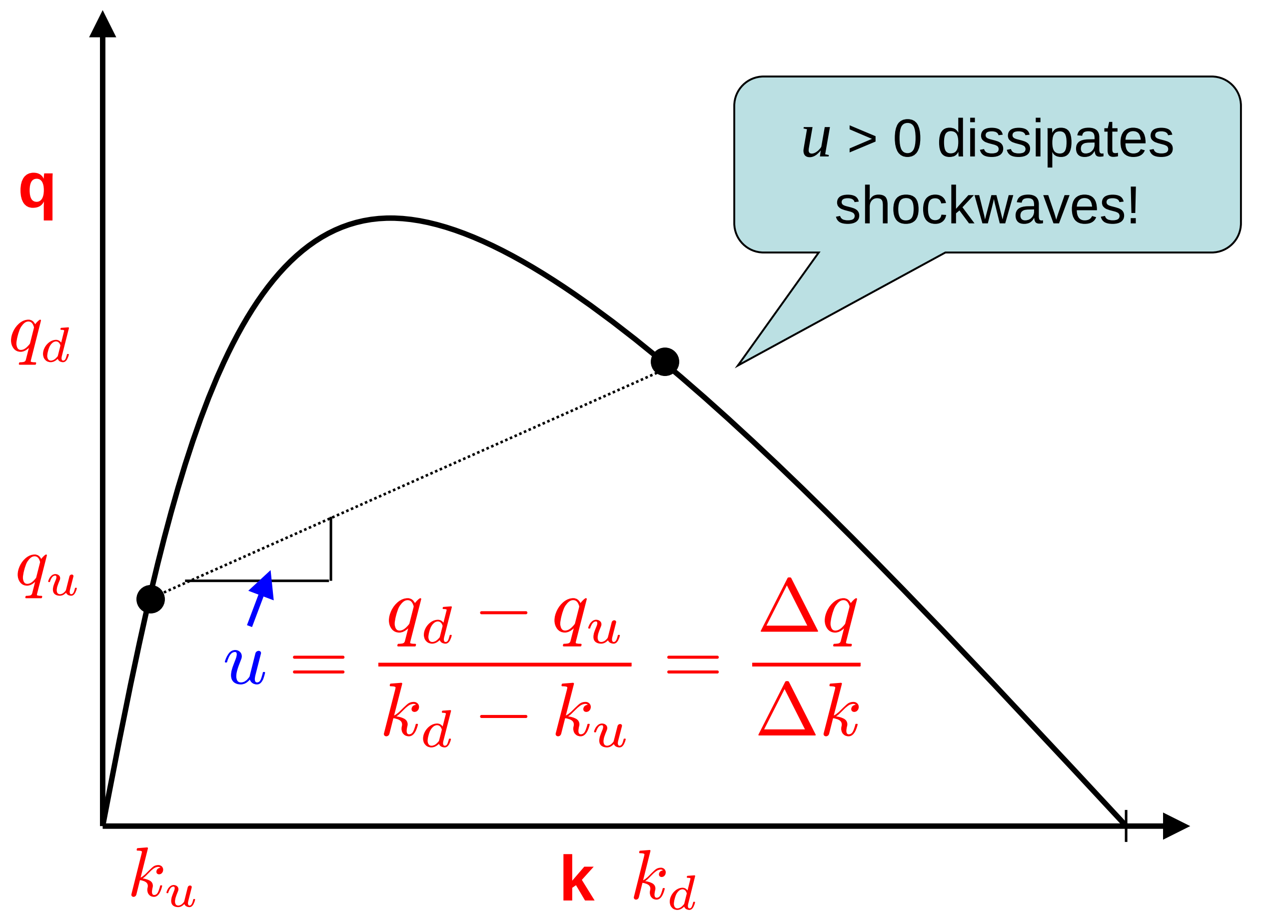
Theory of Shockwaves

Determine shockwave speed u from diagram:

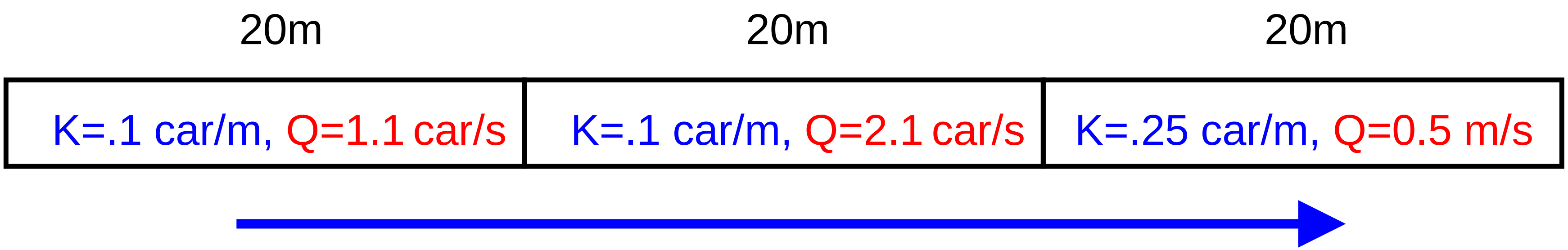


Theory of Shockwaves

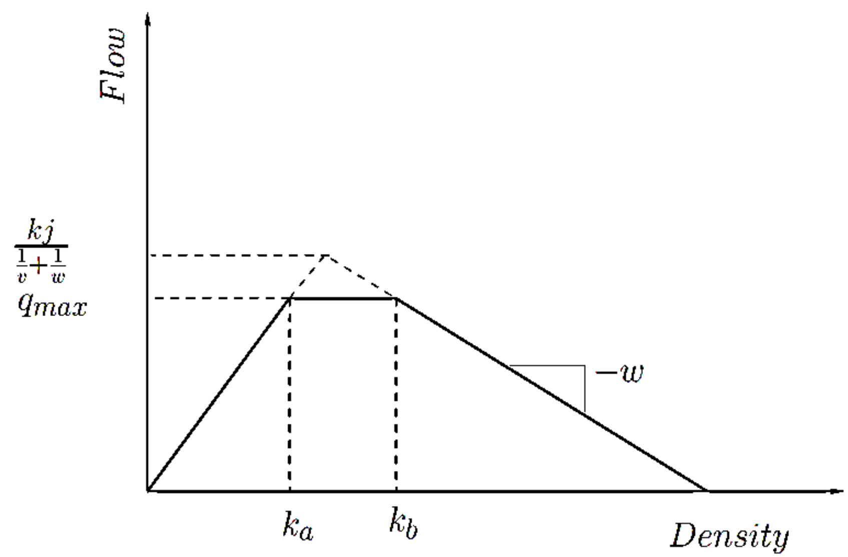
Determine shockwave speed u from diagram:



Cell Transmission Model (CTM)



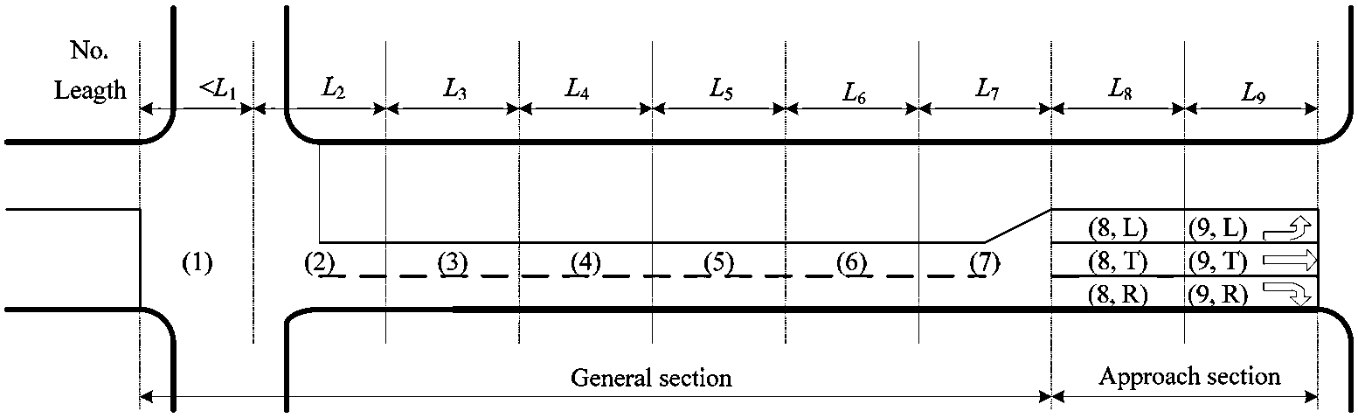
- CTM setup:
  - **Variables:** flow rate, density
  - **Constants:** max capacity, peak and jam densities
  - Piecewise linear difference equation transition model
  - **Recreates shockwave phenomena at macro-level!**



Carlos F. Daganzo, 1994. “The Cell Transmission Model: Network Traffic
http://www.path.berkeley.edu/path/publications/pdf/PWP/94/PWP-94-12.pdf

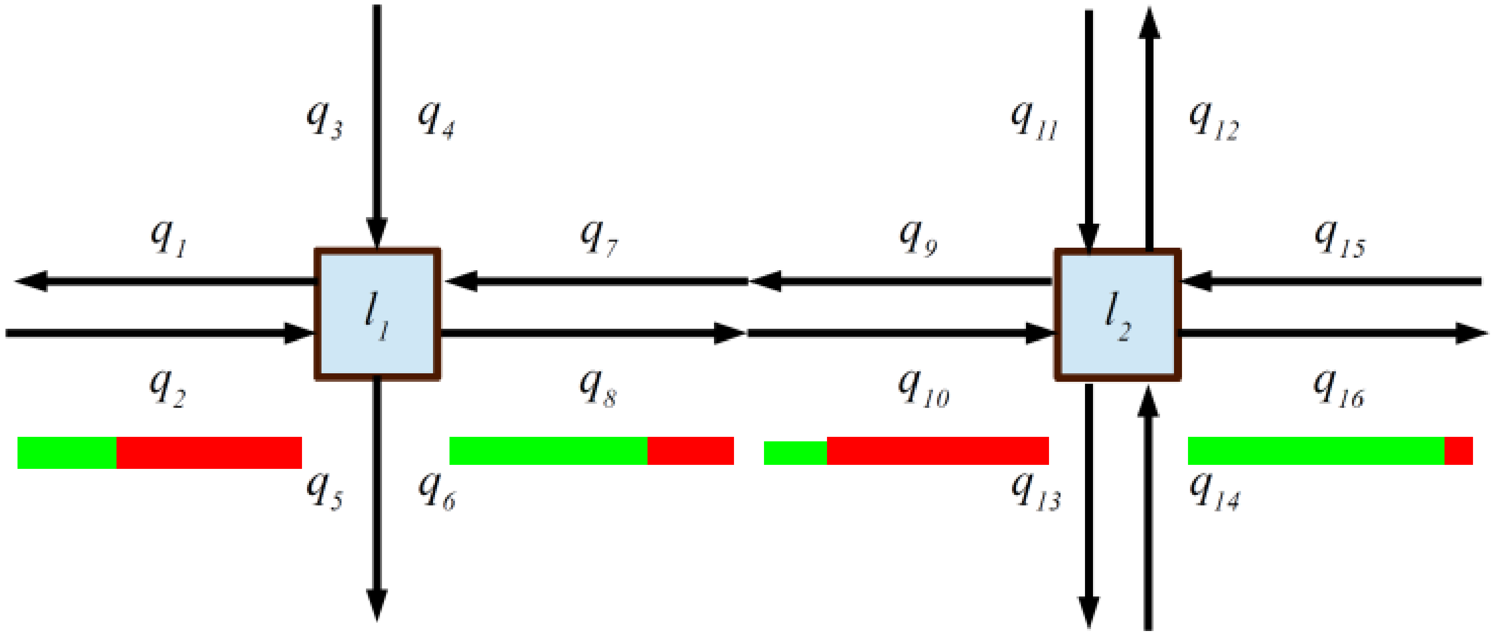
CTM in RDDL

CTM requires a lot of cells…



Is there a more high-level macrosimulation model?

Link-based Alternatives to CTM

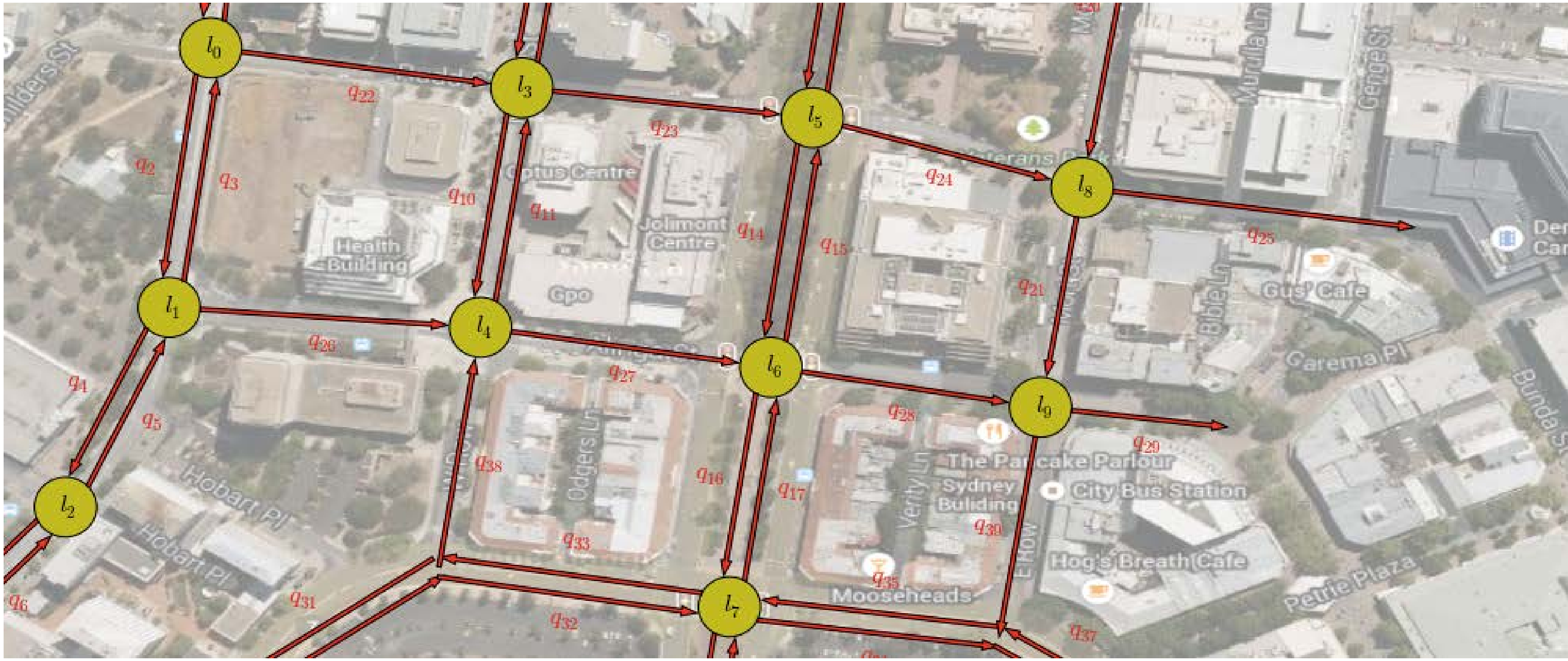


- Link is a traffic queue vertically stacked at stopline
- Limitations [Gartner'02, Han et al'12]
  - Some versions poorly model delay
  - Single traffic boundary (single platoon)

QTM: A Non-homogeneous Time Mixed Integer LP Formulation for Traffic Signal Control

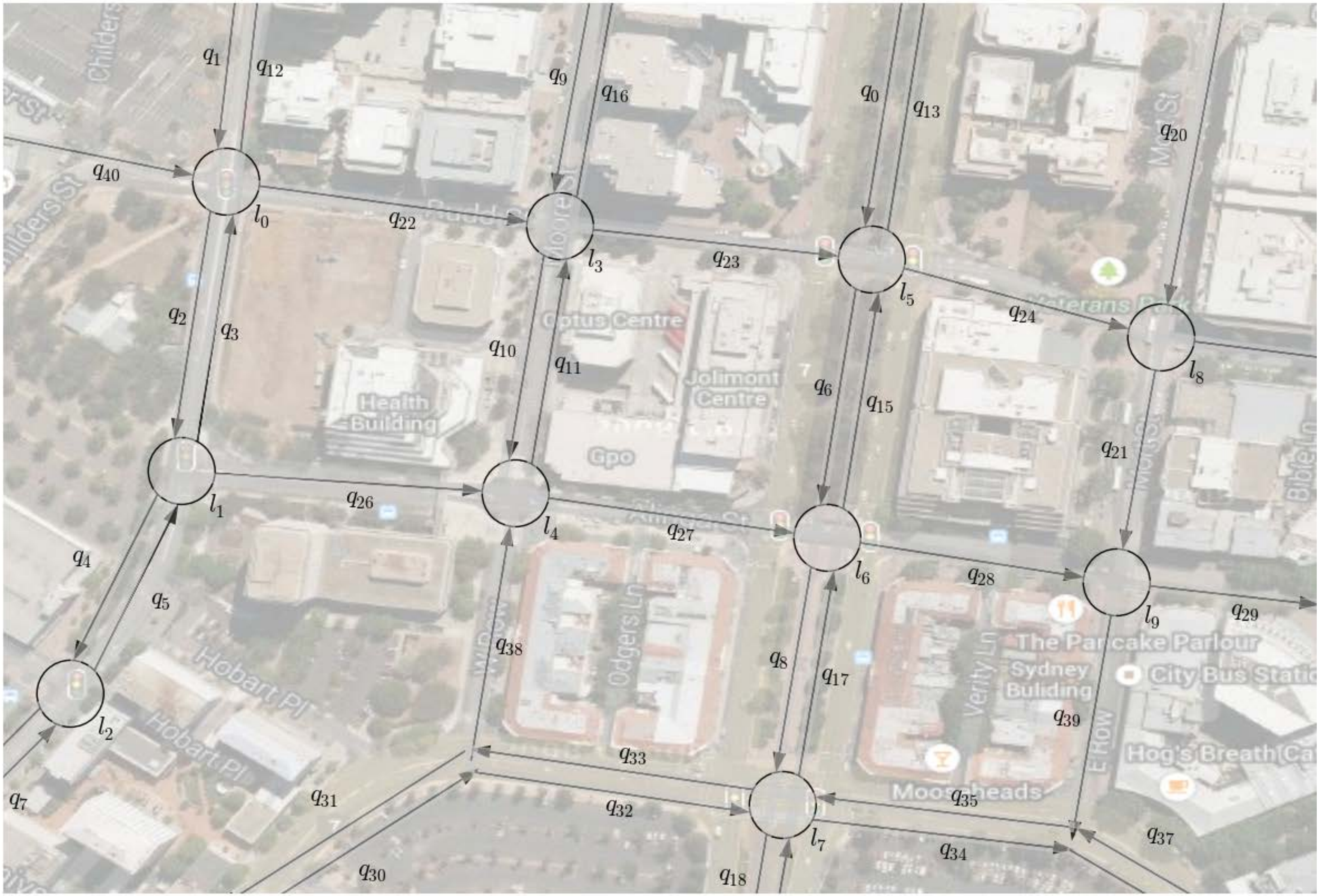
Iain Guilliard, Scott Sanner,
Felipe Trevizan, Brian Williams

A New Queue-based Model (QTM)



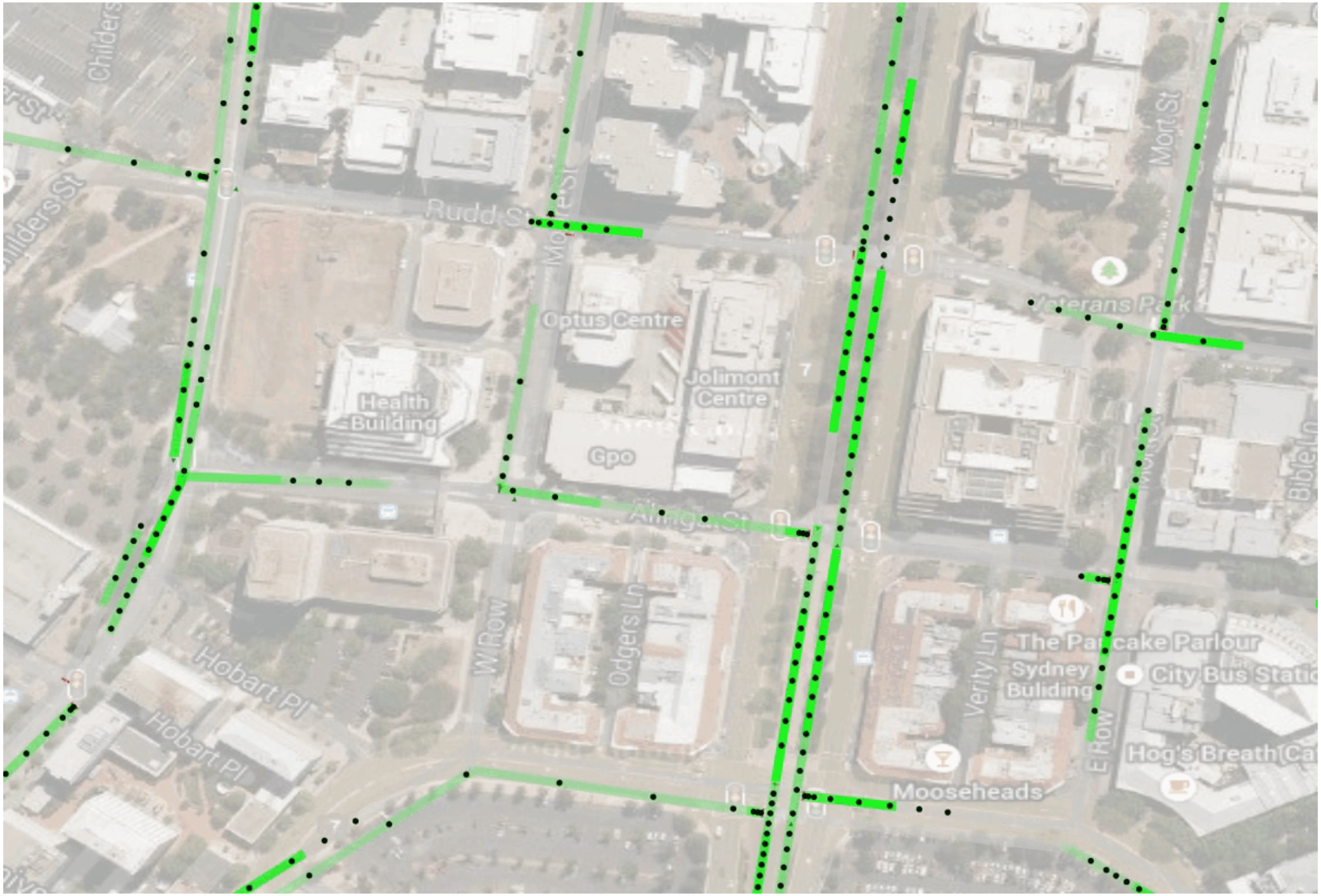
- Each link is a FIFO queue of traffic
- If traffic signals known, flow is an LP!
- If make traffic signals binary decisions ➔ MILP!

QTM Example

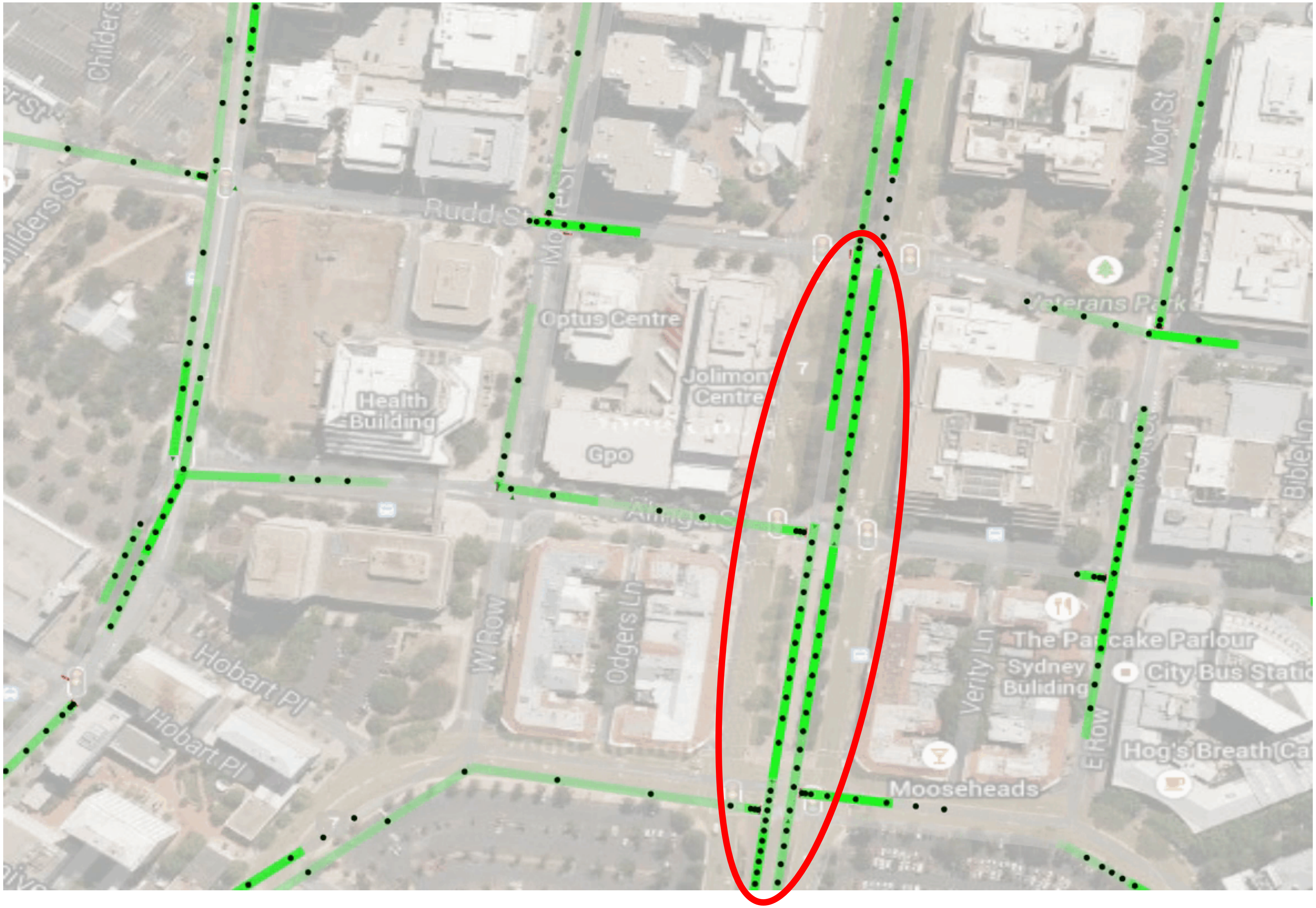


q_1
q_{12}
q_9
q_{16}
q_0
q_{13}
q_{20}
q_{40}
l_0
q_{22}
l_3
q_{23}
l_5
q_{24}
l_8
q_2
q_3
q_{10}
q_{11}
q_6
q_{15}
q_{21}
l_1
q_{26}
l_4
q_{27}
l_6
q_{28}
l_9
q_{29}
q_4
q_5
q_{38}
q_8
q_{17}
q_{39}
l_2
q_{33}
q_7
q_{31}
q_{32}
l_7
q_{35}
q_{34}
q_{37}
q_{30}
q_{18}
Optus Centre
Health Building
Jolimont Centre
Gpo
Hobart Pl
Odgers Ln
Verity Ln
The Pancake Parlour
Sydney Building
City Bus Station
Hog's Breath Ca
E Row

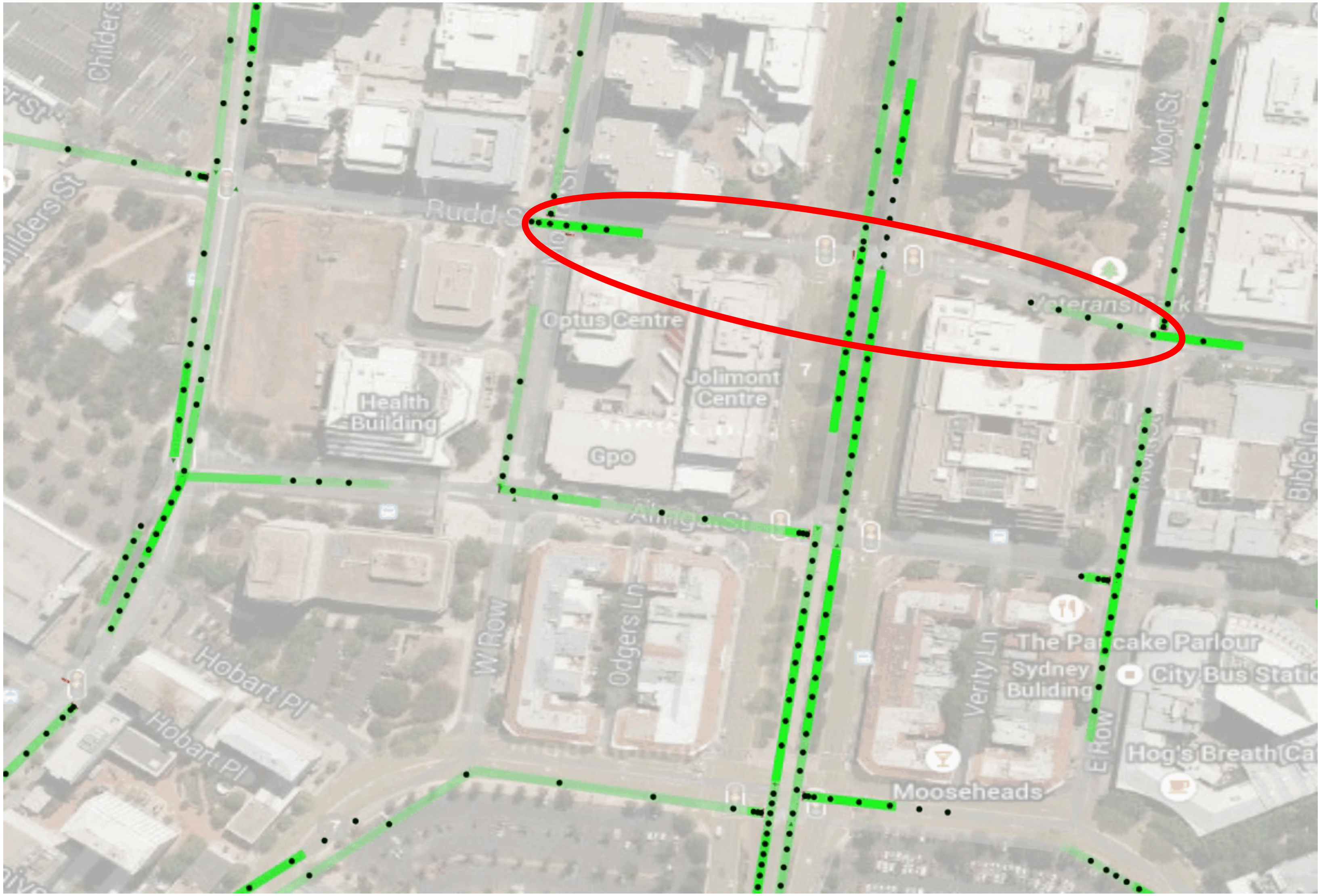
QTM Example – Flow with fixed control



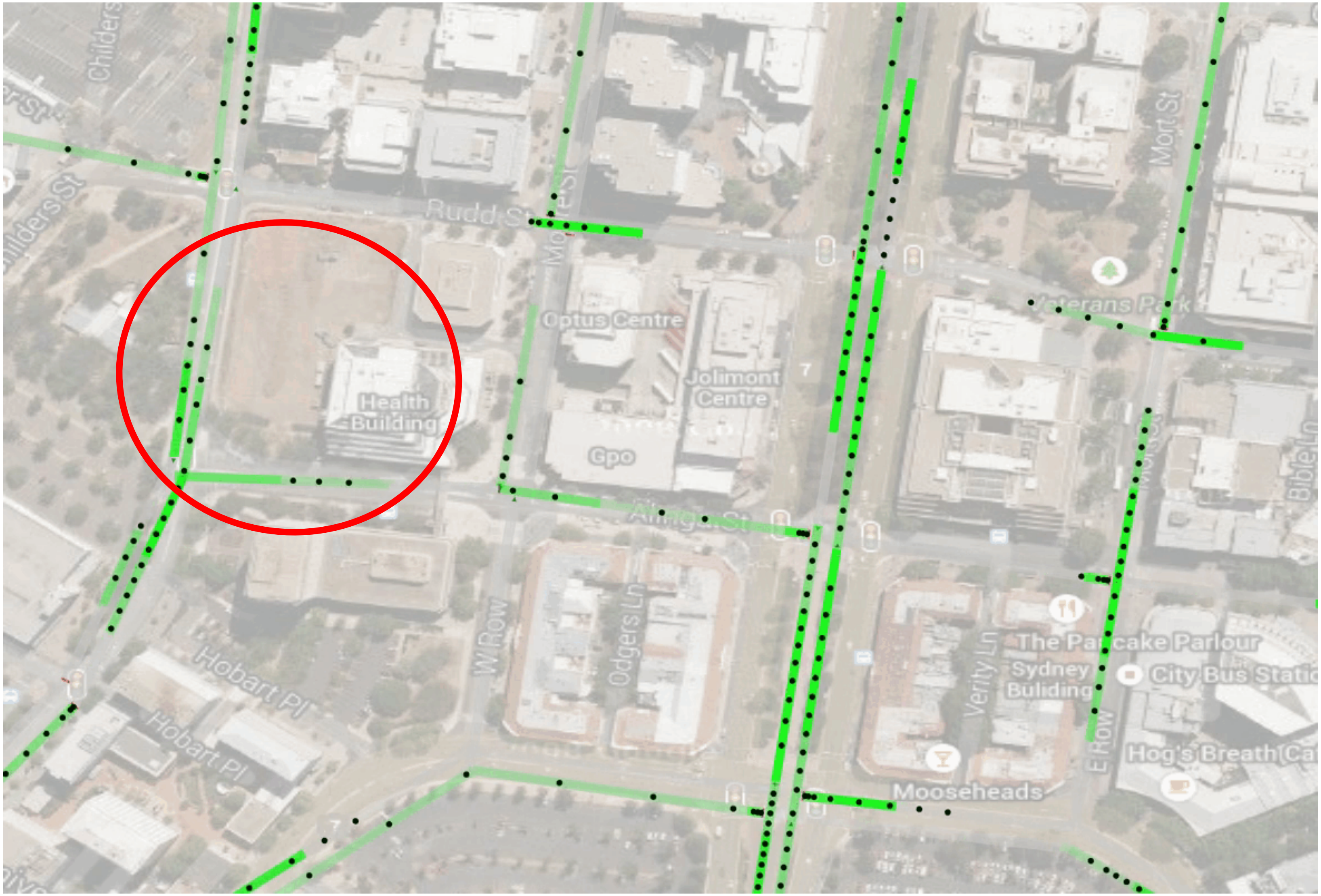
QTM Example – Queuing Behavior



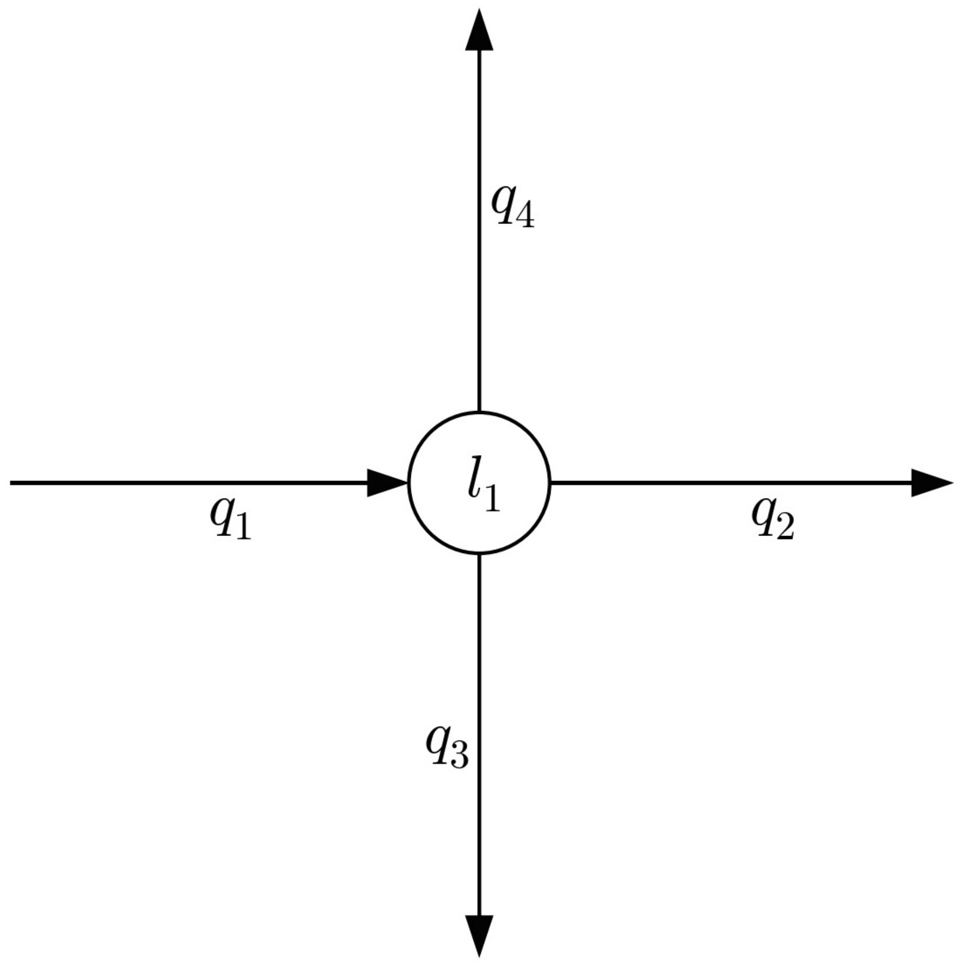
QTM Example – Platoons



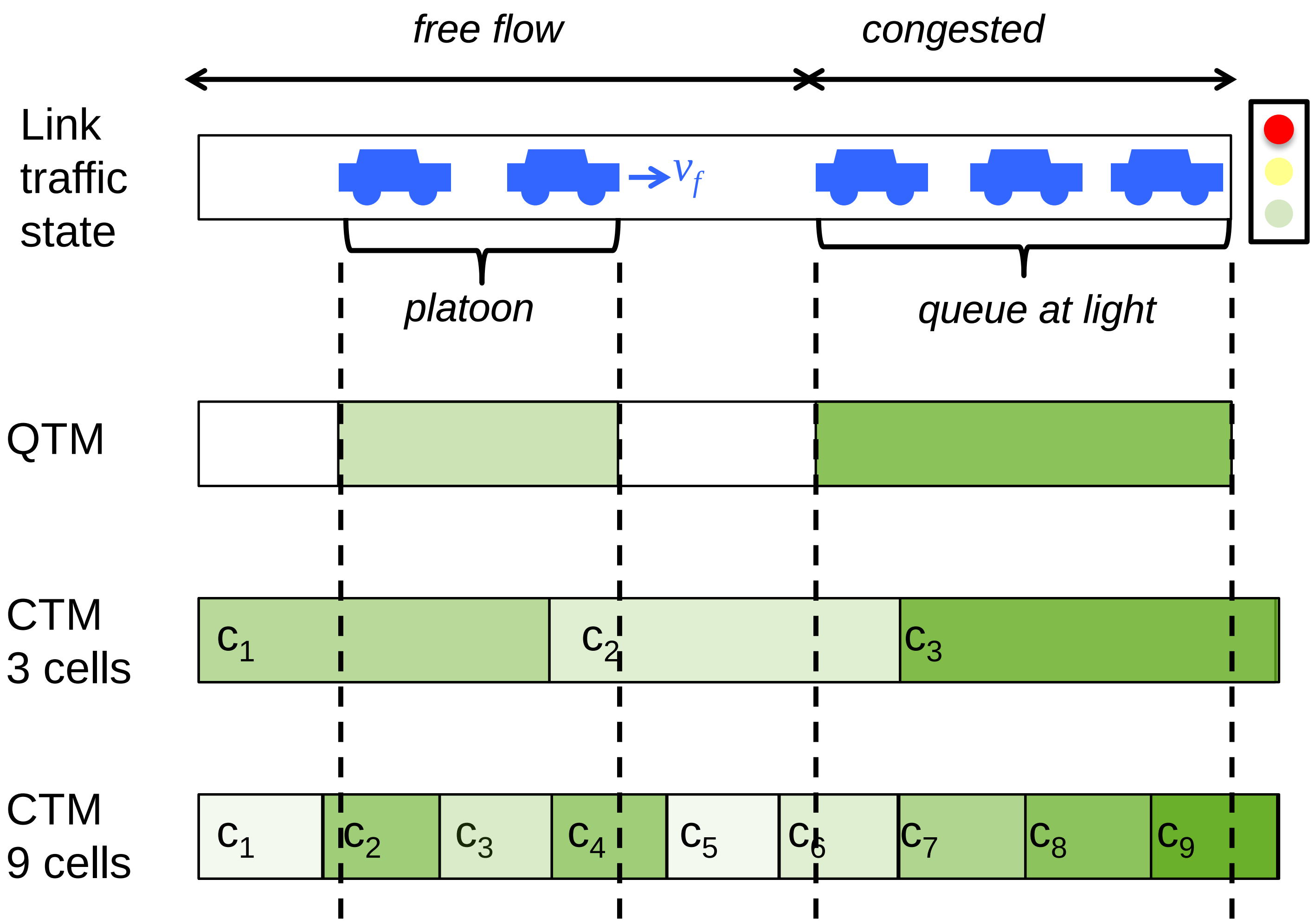
QTM Example – Turn Probabilities



QTM – Variables and Parameters



QTM - Dynamics



Non-Homogenous Link Flow LP

- Constraints

$$in_j^n \leq Q_j^{IN}$$

$$q_{j,in}^n = in_j^n \Delta t^n + \sum_{i=1}^{Q} f_{i,j}^n \Delta t^n$$

$$out_j^n \leq Q_j^{OUT}$$

Non-homogenous Δt!

$$q_{j,out}^n = out_j^n \Delta t^n + \sum_{i=1}^{Q} f_{j,i}^n \Delta t^n$$

$$f_{j,i}^n = F_{j,i}^{PROB} \sum_{k=1}^{Q} f_{j,k}^n$$

$$q_{j,out}^n \leq q_j^{n-1}$$

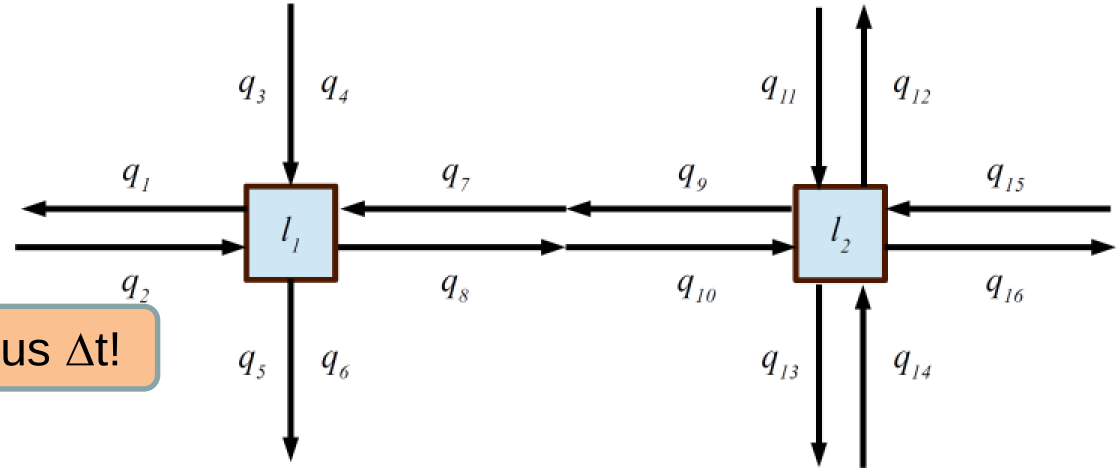
$$q_j^n \leq Q_j^{MAX}$$

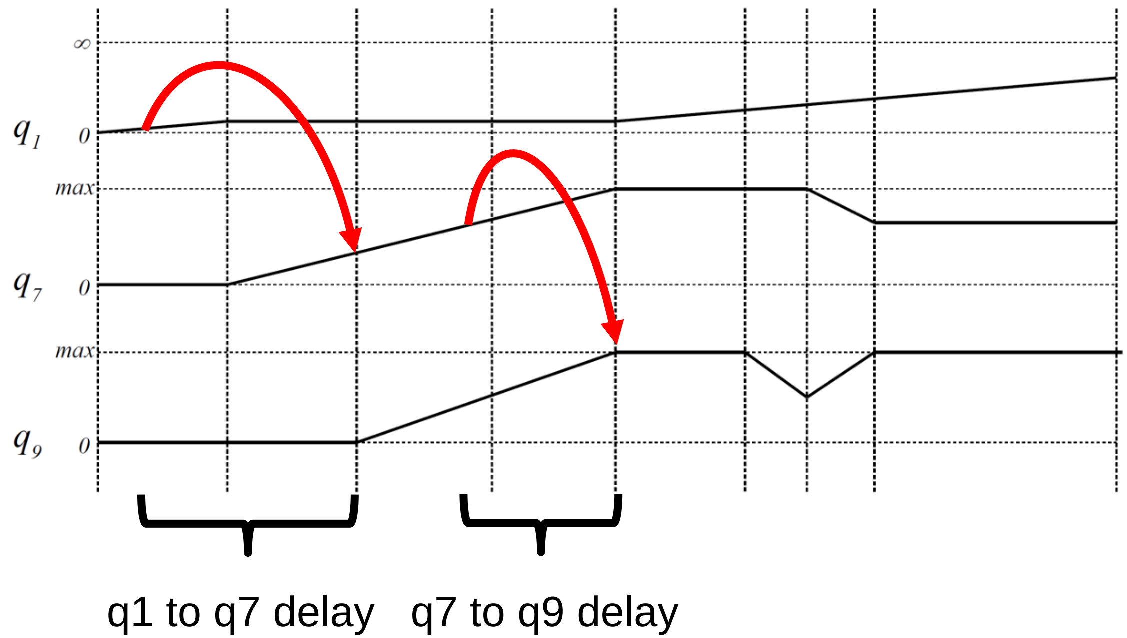
$$q_j^n = q_j^{n-1} - q_{j,out}^{n-1} + q_{j,in}^{n-delay}$$

$$q_{j,in}^{n-delay} + \sum_{k=n-\delta+2}^{n-1} q_{k,in}^k \leq Q_j^{MAX} - q_j^{n-1}$$

$$q_{j,in}^{n-delay} = (1-\alpha) q_{j,in}^{n-\delta} + \alpha q_{j,in}^{n-\delta+1}$$

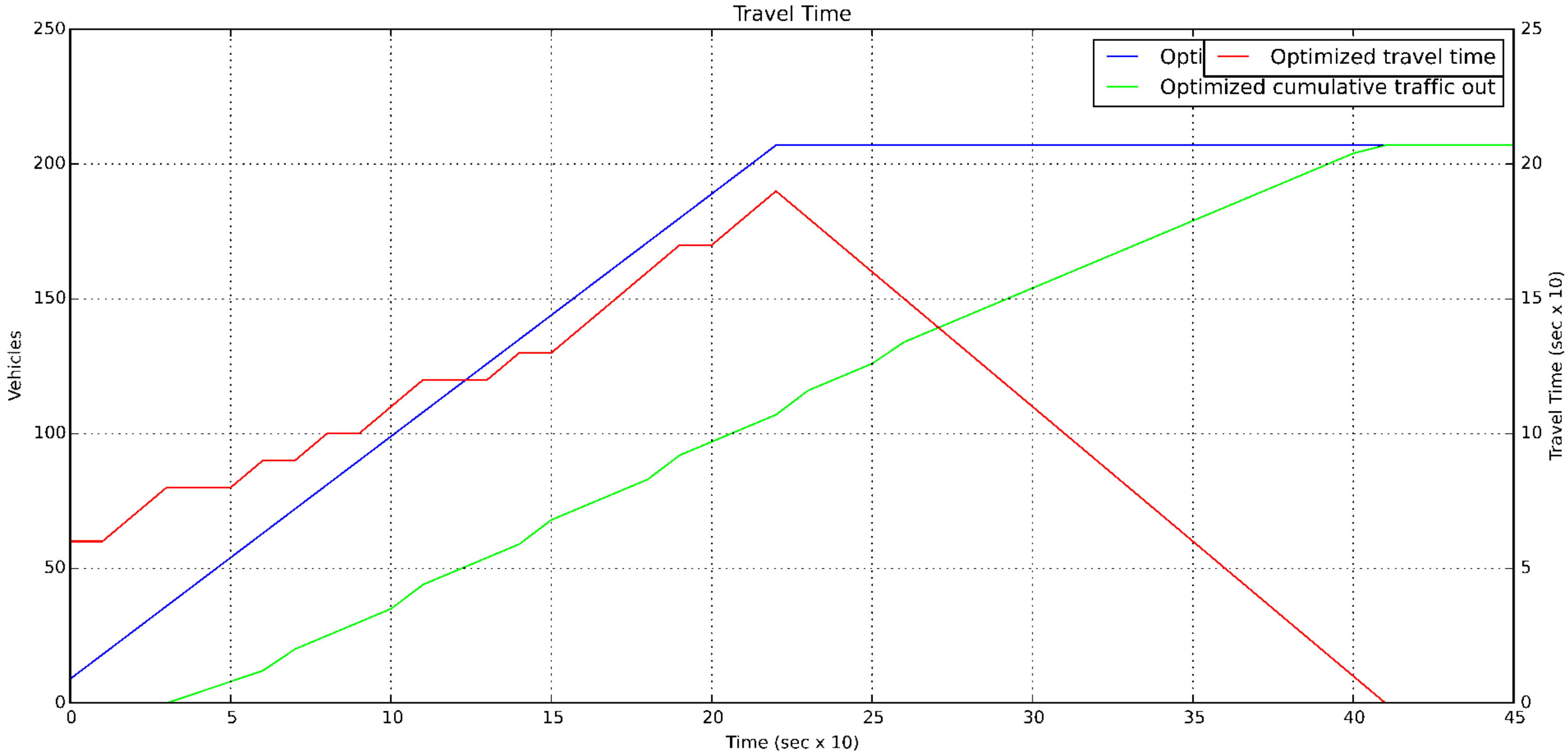
- Maximize ∑ outflows





What to Optimize?

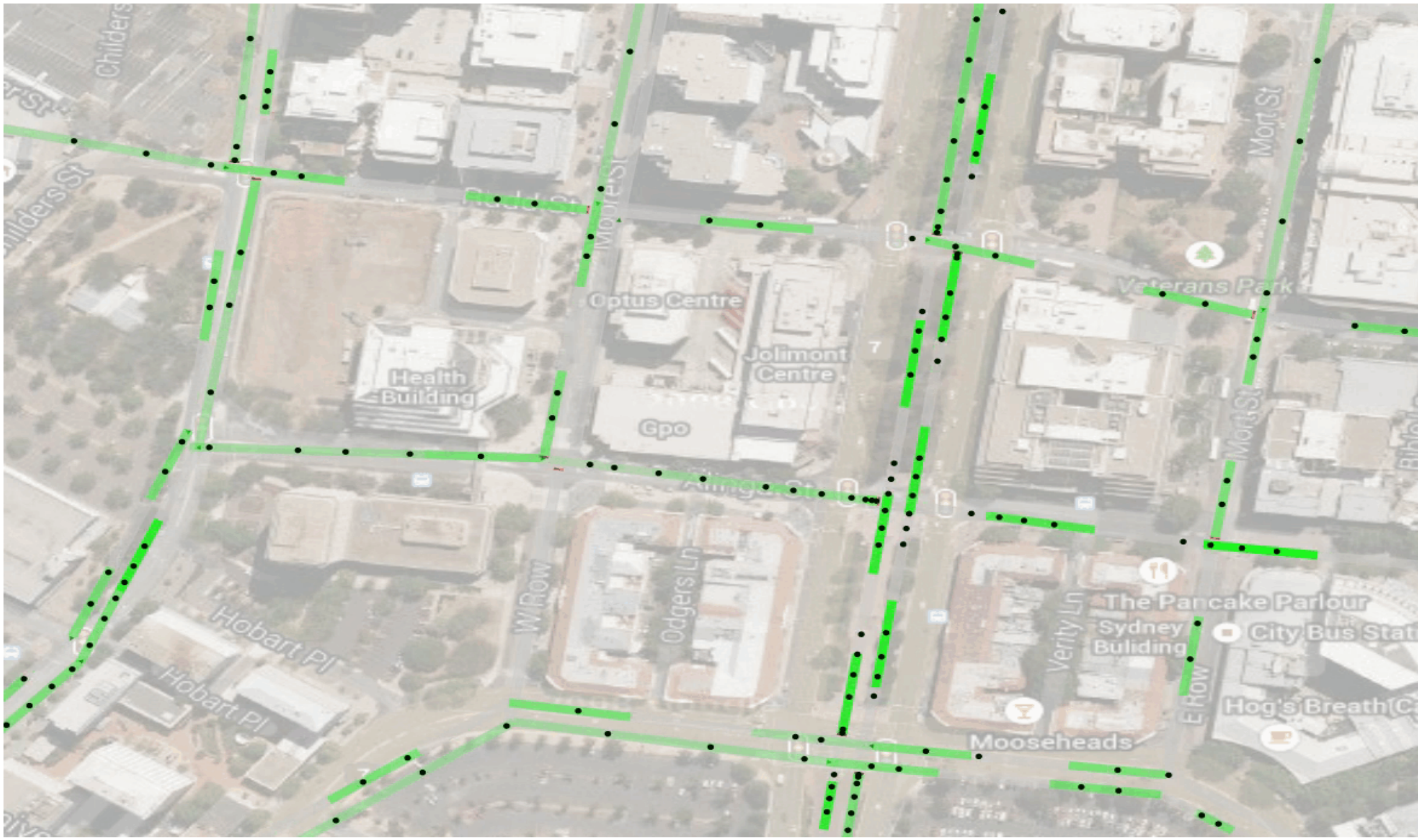
- Minimize delay, but how to define?



- Formally:

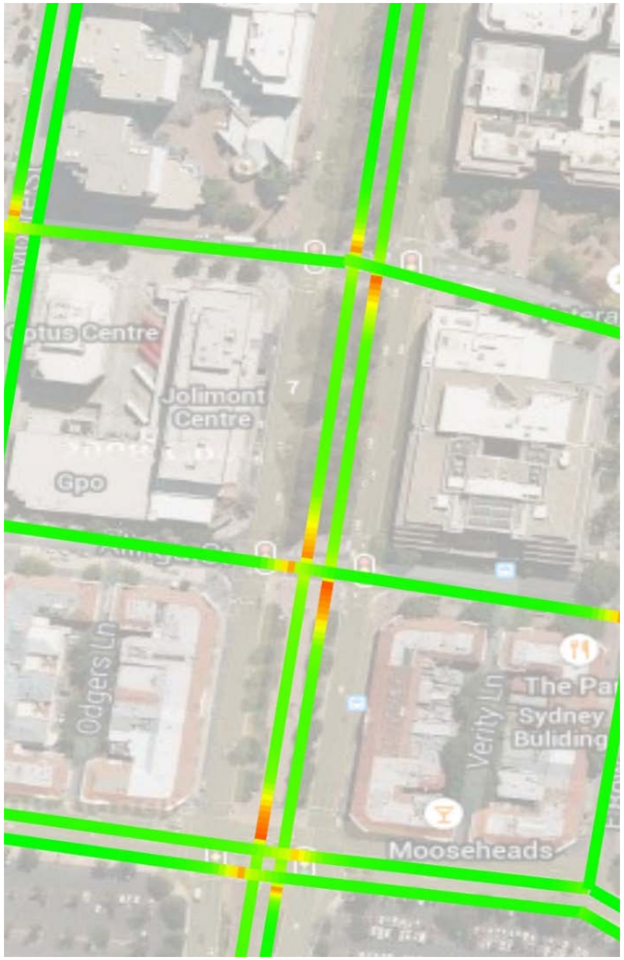
$$\left(\sum_{n=1}^{N}\sum_{j=1}^{Q}(T^{MAX}-t^{n}+1)q_{j,out}^{n}+\sum_{n=1}^{N}\sum_{j=1}^{Q}(T^{MAX}-t^{n}+1)in_{j}^{n}\right)$$

QTM with optimized control



Example: Delay Map, fixed vs optimized

Fixed



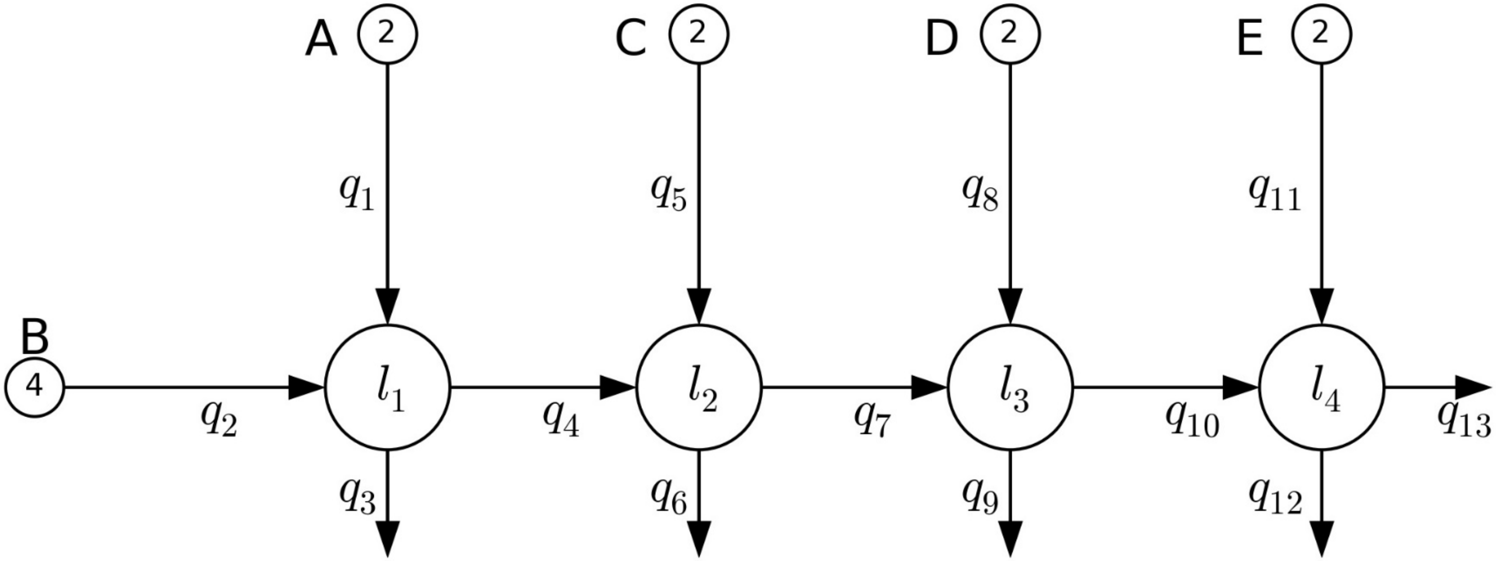
QTM Optimized

Extensions

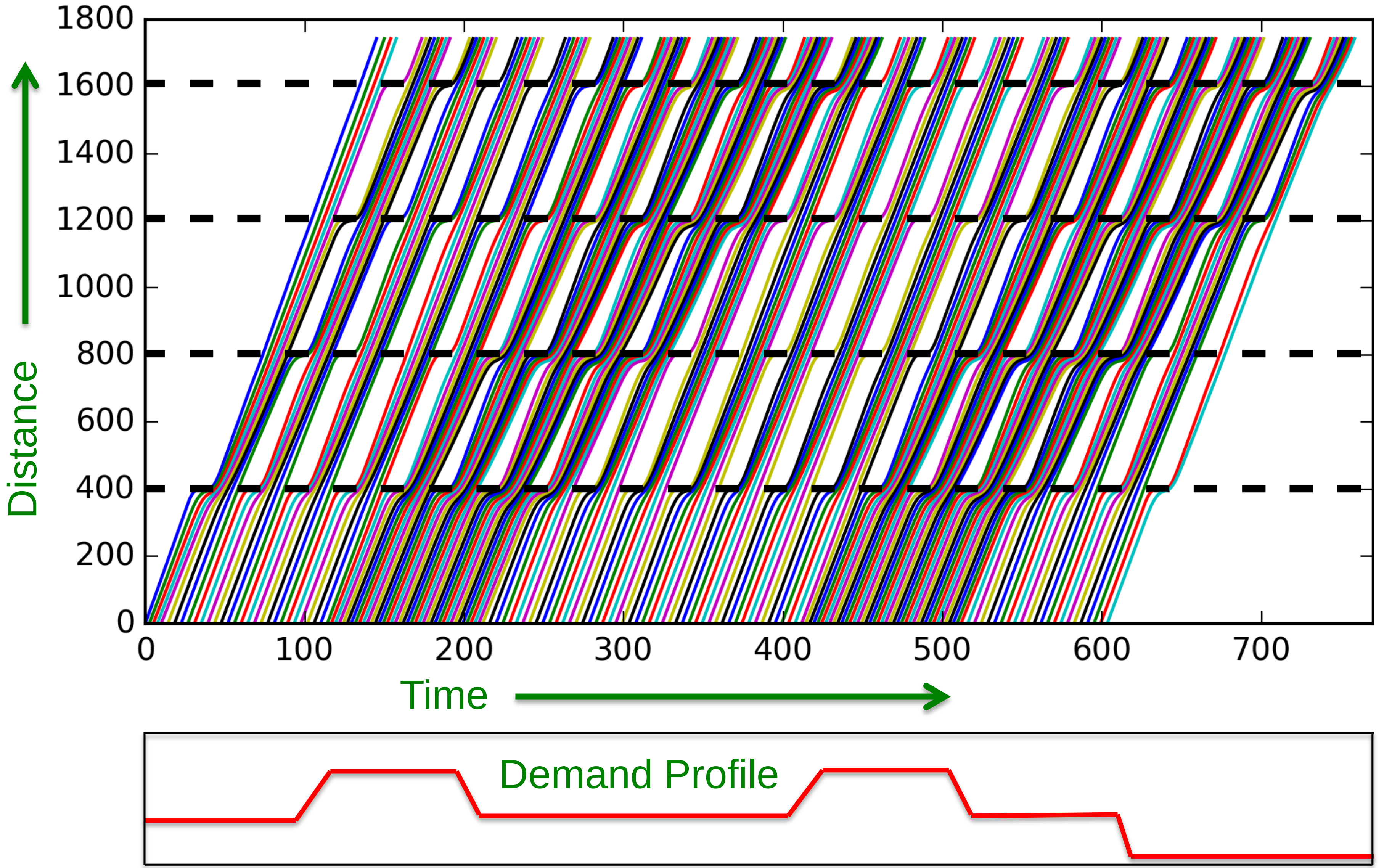
- Globally Optimal Fixed-time Control
  - Simulate fixed adaptive controllers (e.g. SCATS)
  - Pre-compute optimal schedules for fixed controllers
- Light Rail Schedules
  - nullify the impact of introducing light rail
- Uncontrolled intersections
  - Optimize via neighboring intersection signals

Globally Optimize Fixed-time Controllers

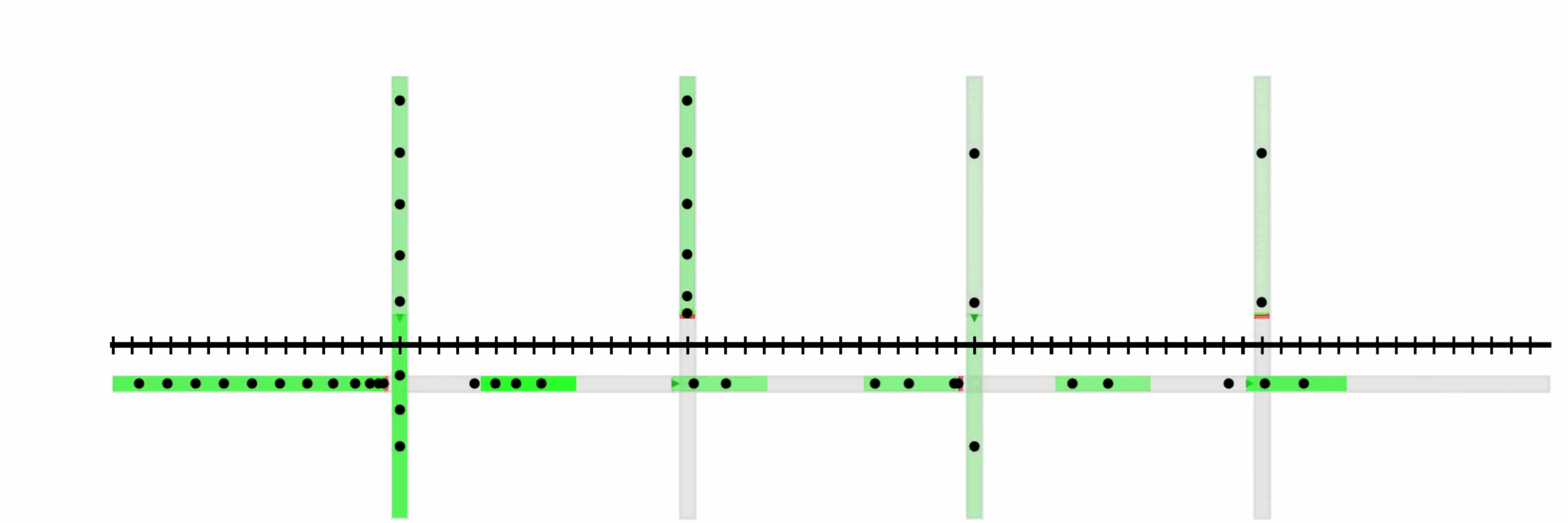
Constrain phase times to be same over all cycles – leads to best fixed-time controller!



Fixed Time Control – micro-simulation

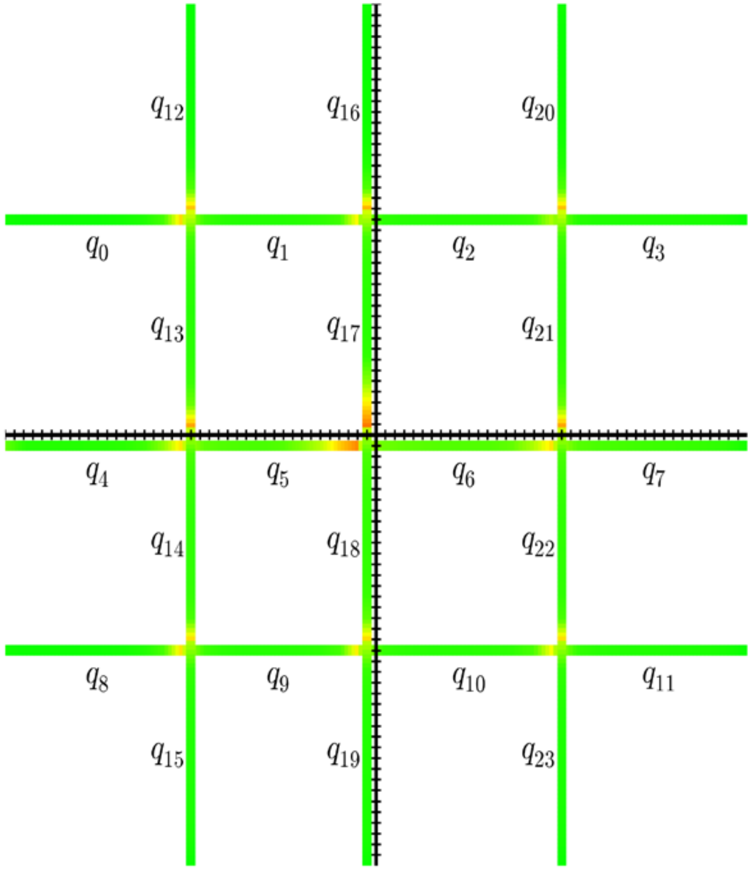


Light Rail – Network 1

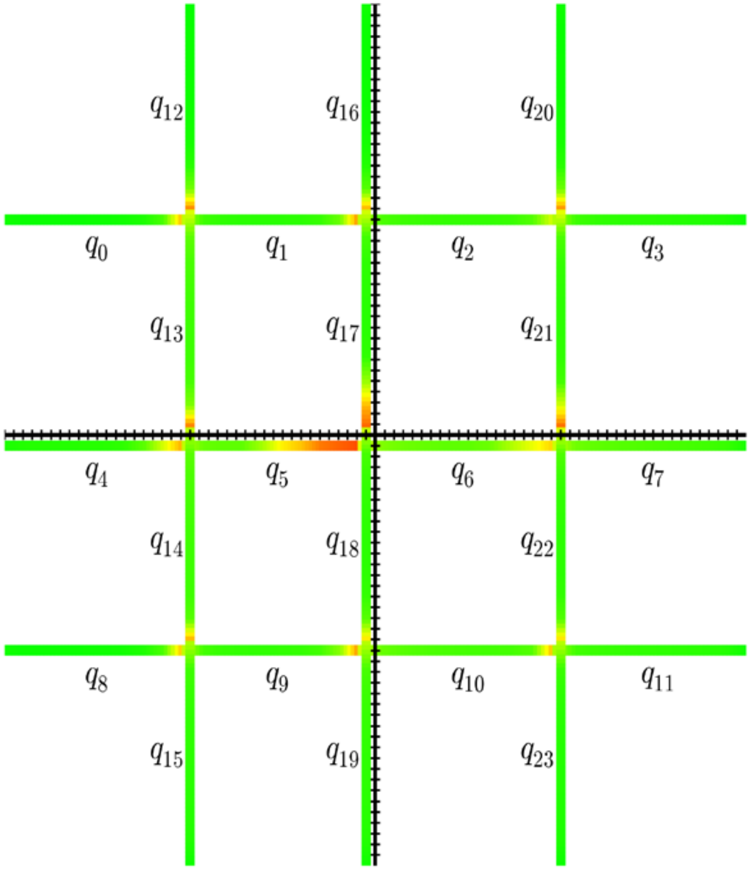


Light Rail – Delay Heat Map 3400 vph

Optimized

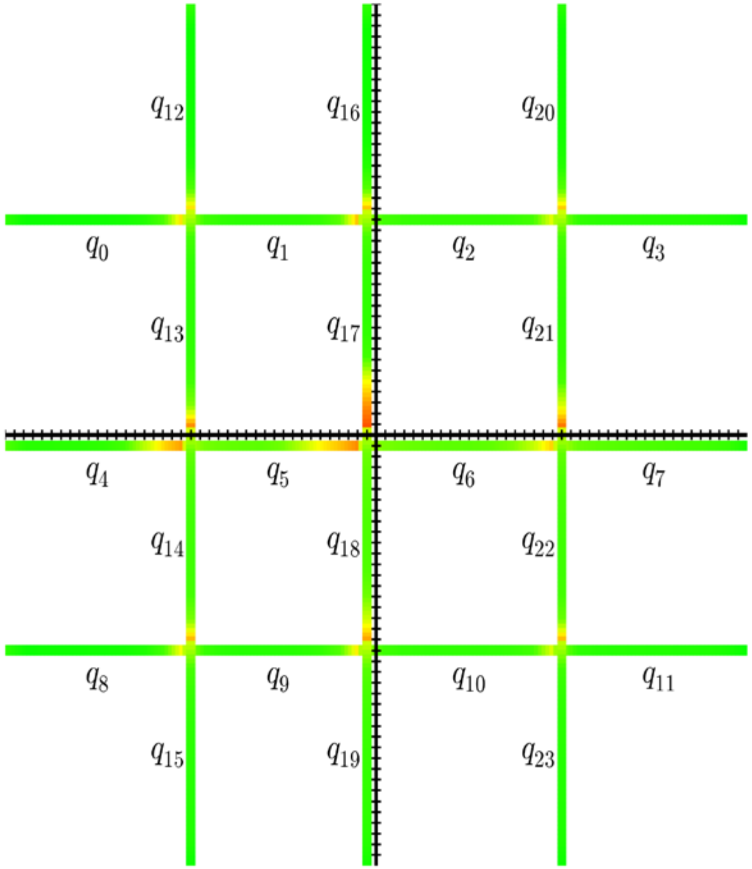


Fixed

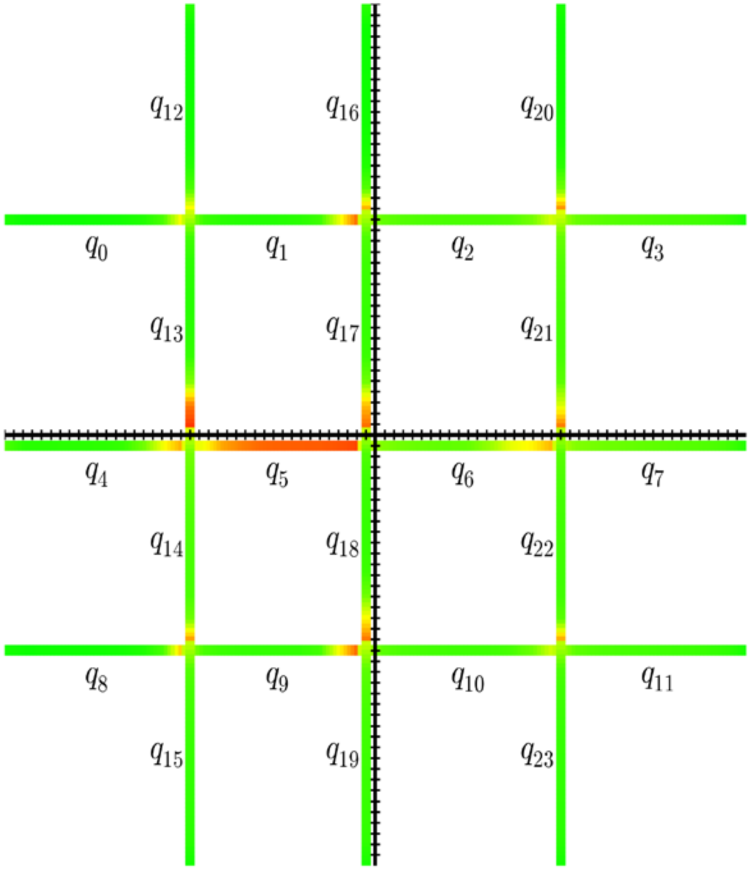


Light Rail – Delay Heat Map 4300 vph

Optimized



Fixed



Uncontrolled Intersections

Future Work

- Close the loop
  - Use high fidelity microsimulator
  - Learn QTM parameters from data

- Compare QTM:
  - with CTM and LTM MILPS

Code on Github:

github.com/iainguilliard/QTM_Traffic_Model

Lecture Midpoint Goals Recap

1) To understand fundamentals of traffic signal control in theory and practice

2) To understand QTM approach for optimizing traffic signals using MILPs

Done

Next

3) To understand the Surtrac job-shop scheduling approach to traffic signal control

4) To understand frontiers of traffic signal control: connected and autonomous vehicles