RHODES: a real-time traffic adaptive signal control system

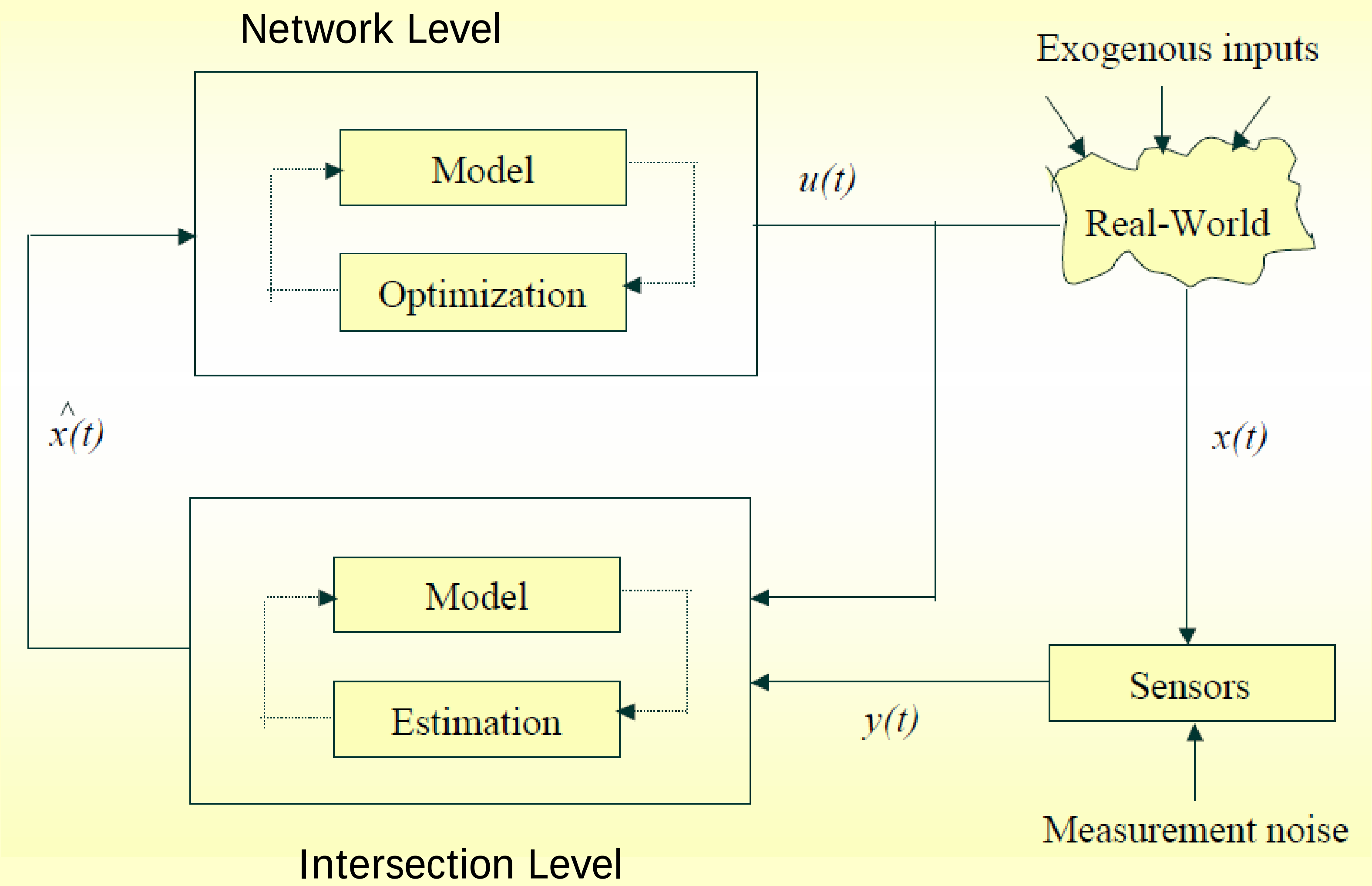
Contents

- Introduction of RHODES
- RHODES Architecture
- The prediction methods
- Control Algorithms
- Integrated Transit Priority and Rail/Emergency Preemption
- The RHODES prototype

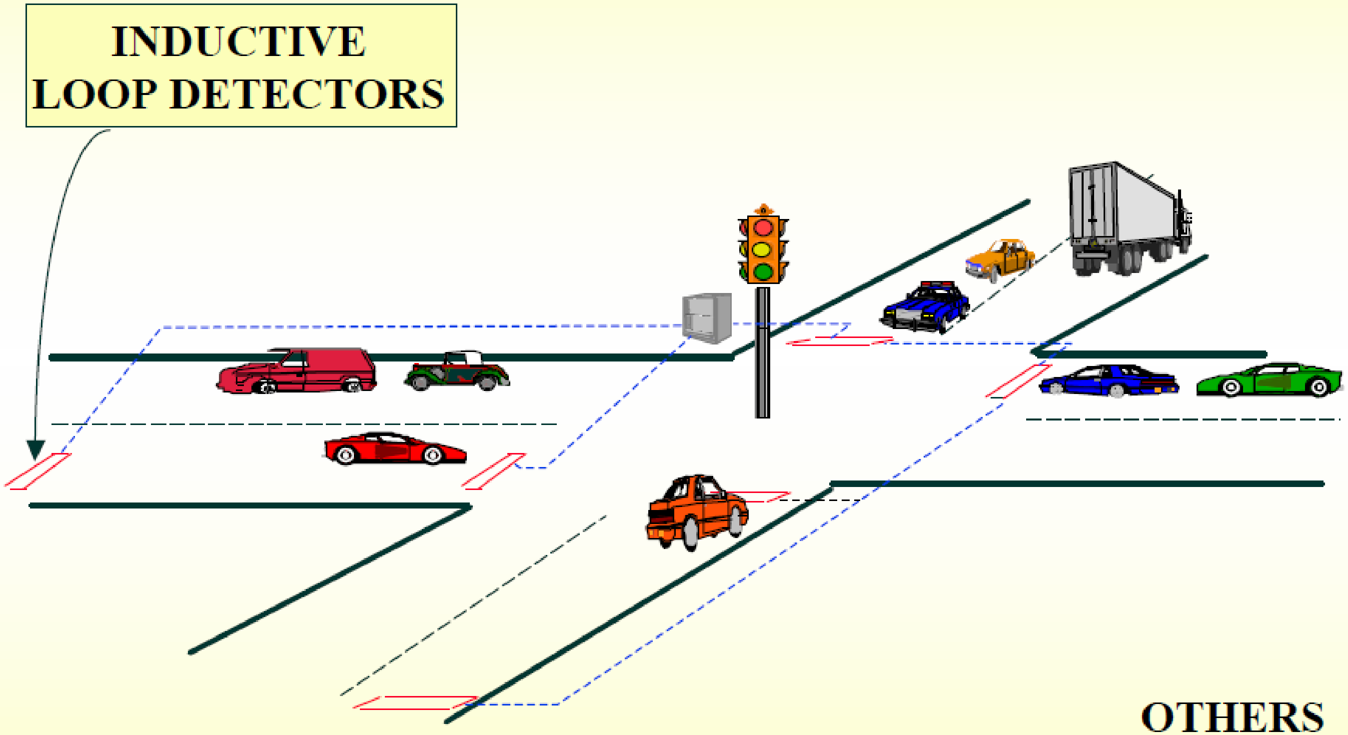
What is RHODES?

- ROHODES = Real-time Hierarchical Optimizing Distributed Effective System
  - Developed by a research team at the University of Arizona
  - Arizona DOT and FHWA have provided research funding to assist in the exploration of RHODES concepts.

RHODES General Idea



Sensors



<u>OTHERS</u>

- VIDEO DETECTORS
- SONAR DETECTORS
- RADAR DETECTORS

RHODES Architecture

- RHODES uses three-level hierarchy for characterizing and managing traffic.
- RHODES explicitly predicts traffic at these levels utilizing detector and other sensor information.
- RHODES requires

  (a) lane traffic data (e.g., through detectorization)

  (b) real-time communication to/from processors, and

  (c) PC-level computational capability.

RHODES: Logical Architecture

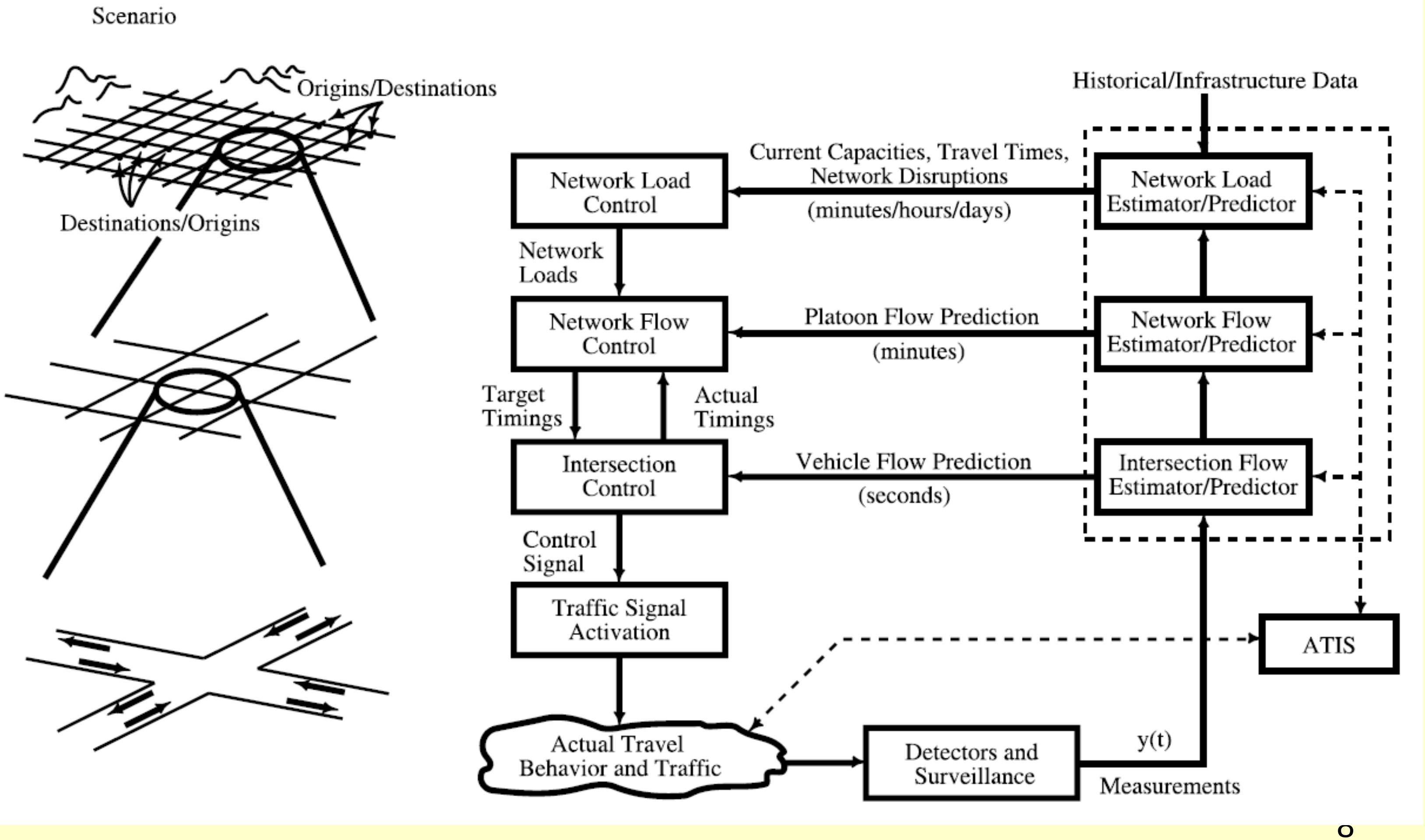
- The highest level is a "dynamic network loading" model that captures the slow-varying characteristics, which pertain to the network geometry and the typical route selection of travelers.

- Based on the traffic load on each particular link, RHODES allocate green time for each demand pattern and each phase. These decisions are made at the middle level of the hierarchy, referred as "Network flow control."

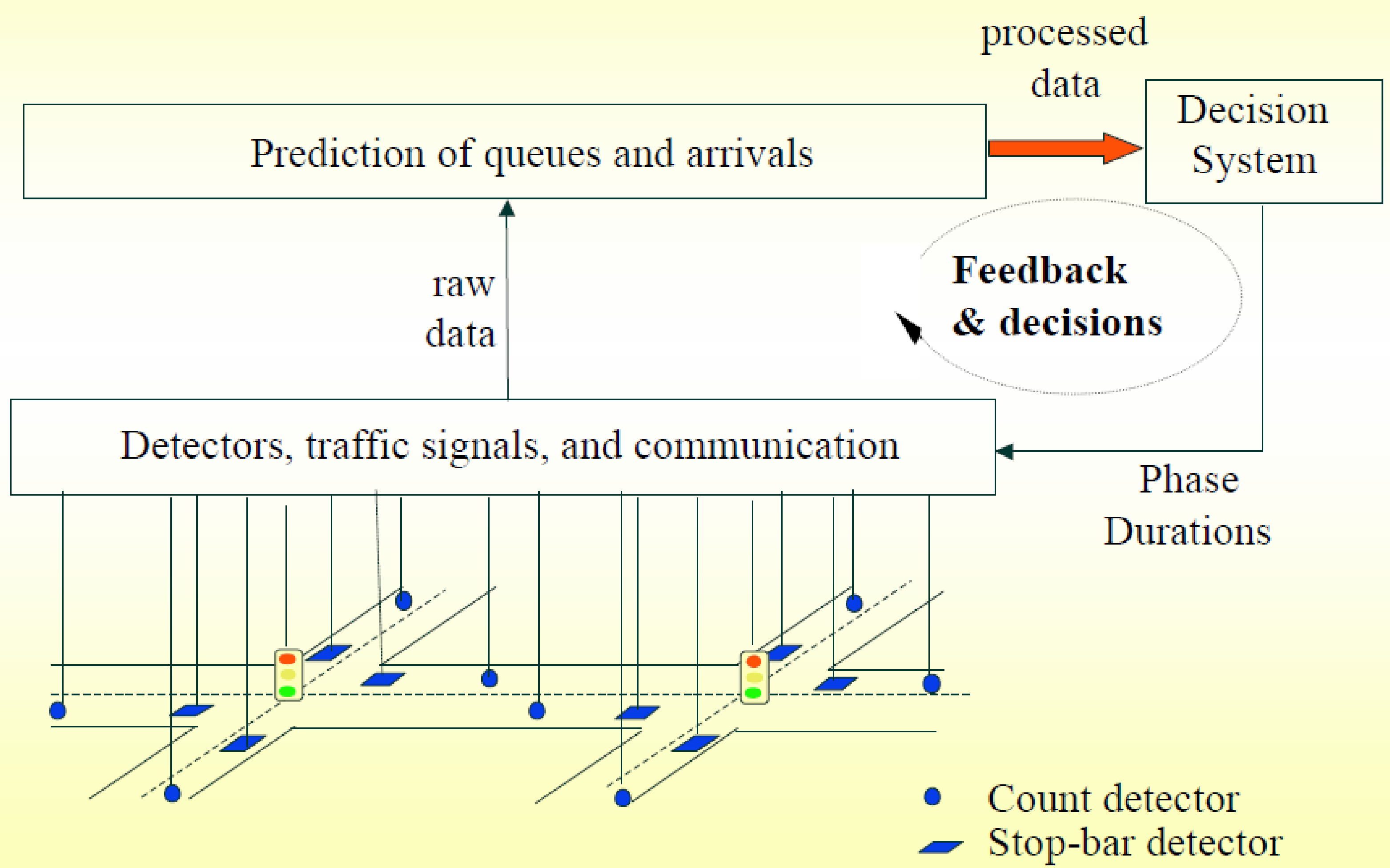
- Given the approximate green times, the "intersection control" at the third level selects the appropriate phase change epochs based on observed and predicted arrivals of individual vehicles at each intersection.

RHODES: Logical Architecture

- At each level, there is an estimation/prediction component and a control component.



Simplified Architecture



Prediction of Intersection Arrivals

- The PREDICT algorithm (Head,1995) used the output of the detectors on the approach of each upstream intersection.

- The prediction assumes that the arrival process can be divided into two parts: a predictable part and an unpredictable part:

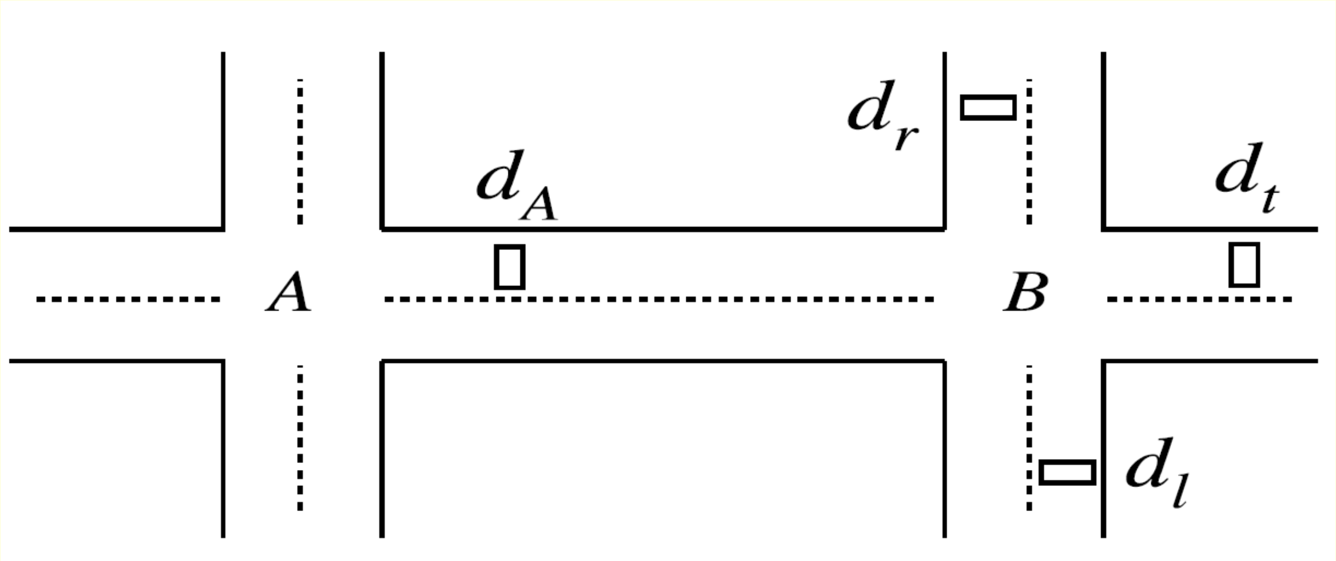
$$n(t) = n_p(t) + n_u(t)$$

The contribution of unpredictable part will influence the control strategy at intersection.

For example, if the process is highly predictable, the control strategy could be to allow platoon progression; if the process is highly unpredictable, then the control strategy could be to gather arrivals into platoons that can be accommodated at downstream intersection.

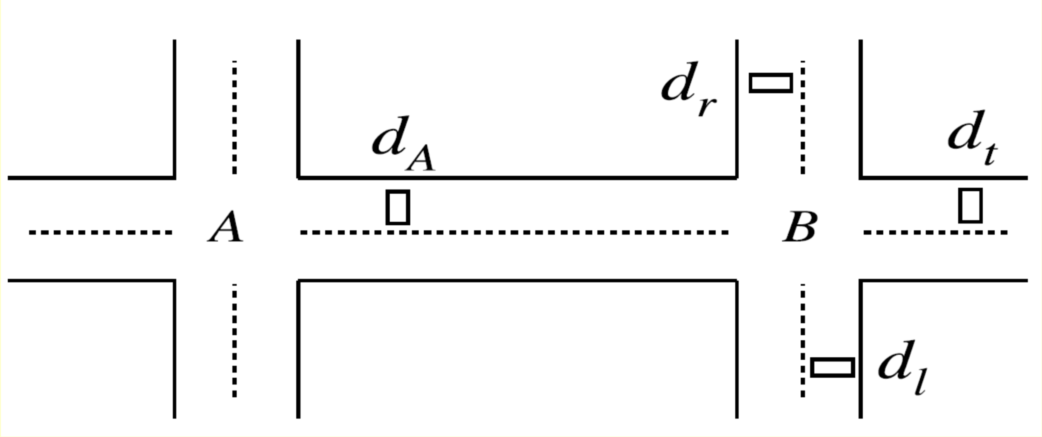
Prediction of Intersection Arrivals

For example: to predict the flow approaching intersection A at detector d_A.



Making this prediction is important because it is a point on Link AB where the actual flow can be measured, hence the quality of the prediction can be assessed in real-time.

Prediction of Intersection Arrivals

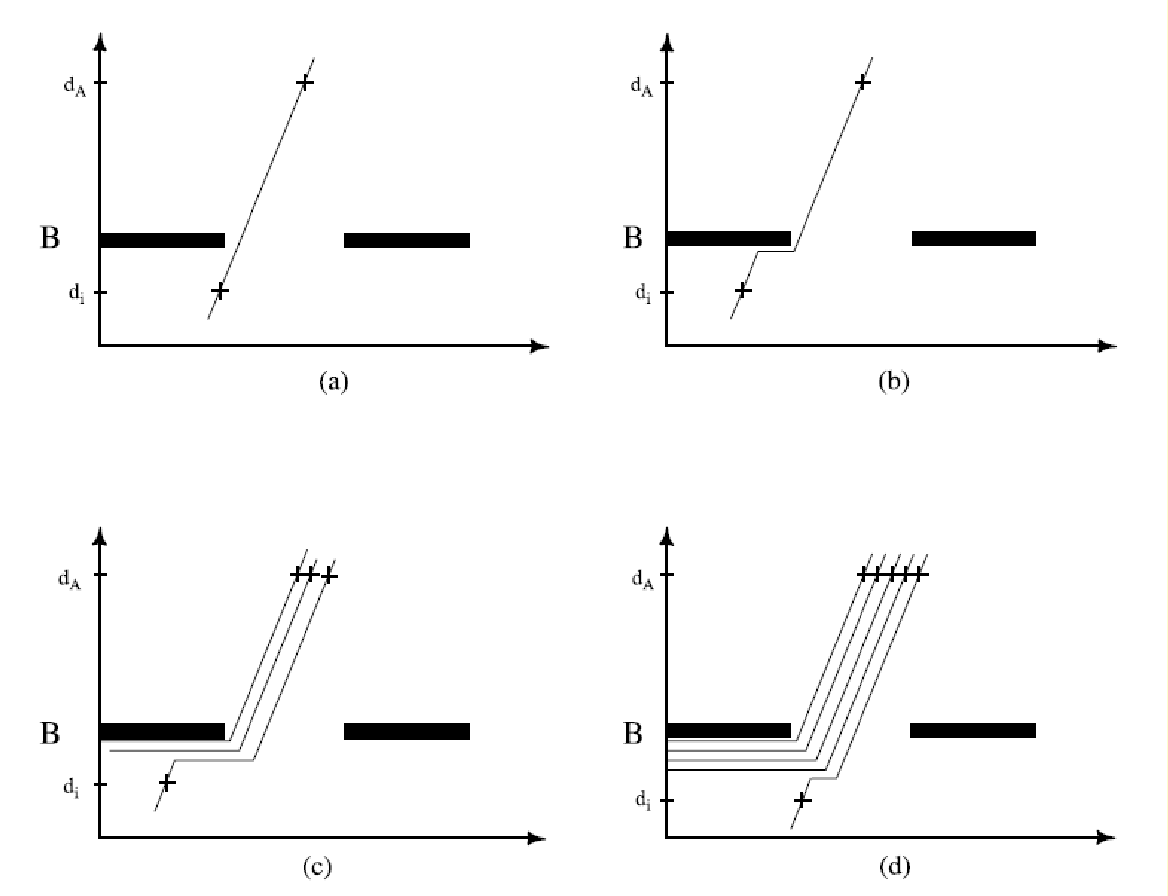


Several factors affect when and if the vehicle will arrive at d_A:

- Travel time from d_i (i=l,t,r) to the stop bar at the intersection B;
- Delay due to an existing queue at B;
- Delay due to the traffic signal at B;
- Travel time between B and d_A and
- Probability that the vehicle will travel along a route that includes location d_A.

Prediction of Intersection Arrivals

Associated with the first four factors, the following figures depict the delay of different scenarios:



Prediction of Intersection Arrivals

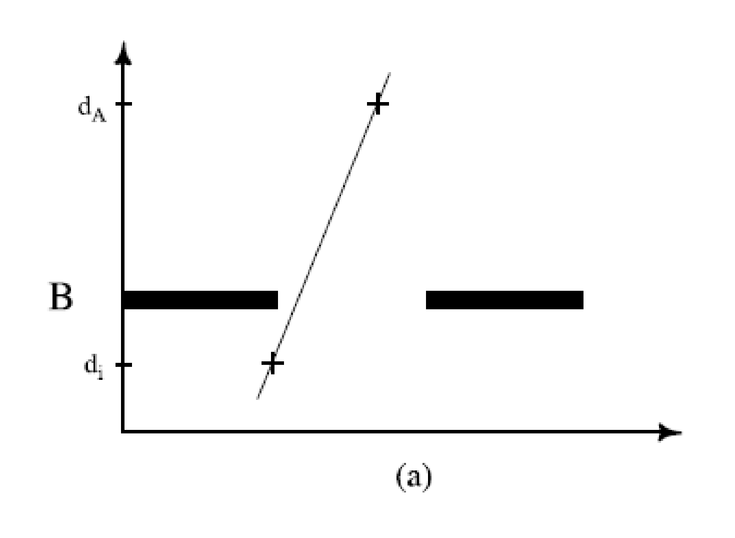


Figure (a): the vehicle arrives at detector di and passes freely to detector dA.

The arrival time is given by:

$$t_a = t_{d_i} + T_{d_i, S_B} + T_{S_B, d_A}$$

Arrival time at A = arrival time at di
+ travel time (di to Stop line of Intersection B)
+ travel time (stop line of Intersection B to dA)

Prediction of Intersection Arrivals

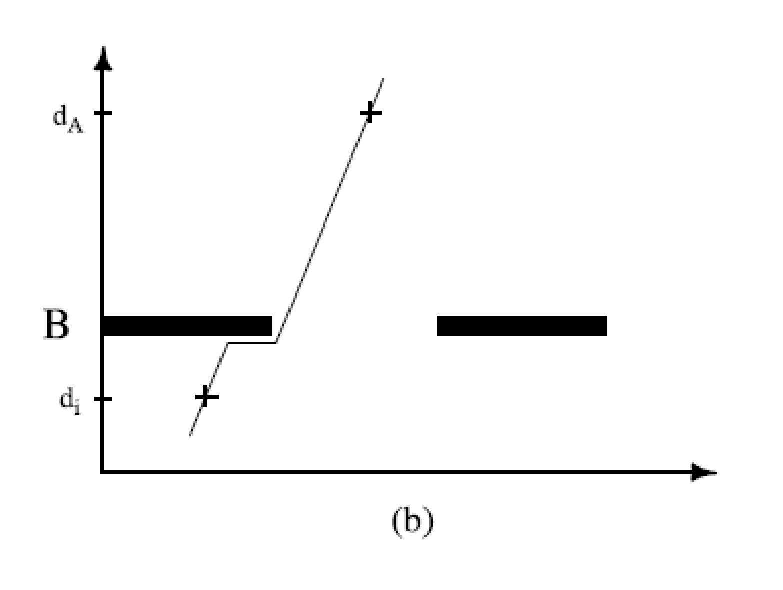


Figure (b): the vehicle arrives at detector di and is delayed by the signal at intersection B.

The arrival time is given by:

$$t_a = t_{d_i} + \max\{T_{d_i,S_B}, T_{u_B}\} + T_{S_B,d_A}$$

Arrival time at A = arrival time at di
+ max of {travel time of di to B, signal delay}
+ travel time (stop line of Intersection B to dA)

Prediction of Intersection Arrivals

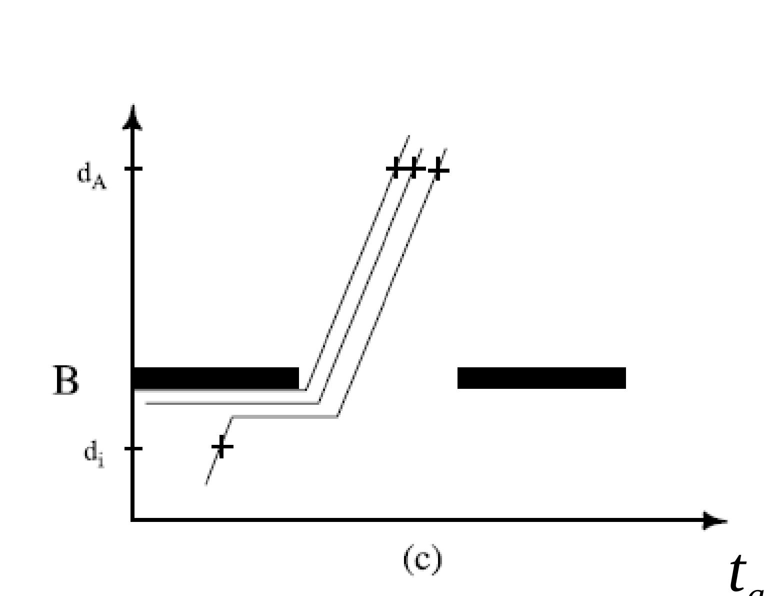


Figure (c): the vehicle arrives at detector di and encounters delay for the signal as ell as a standing queue.

The arrival time is given by:

$$t_a = t_{d_i} + \max\{T_{d_i,S_B}, T_{u_B} + T_{q_i}\} + T_{S_B,d_A}$$

Arrival time at A = arrival time at di
+ max of {travel time of di to B, signal delay + queue delay}
+ travel time (stop line of Intersection B to dA)

and $$T_{q_i} = a_0 + a_1 N_{q_i}$$

Where a0 and a1 are parameters that can be selected on each intersection and Nq is the number of vehicle in queue.

Prediction of Intersection Arrivals

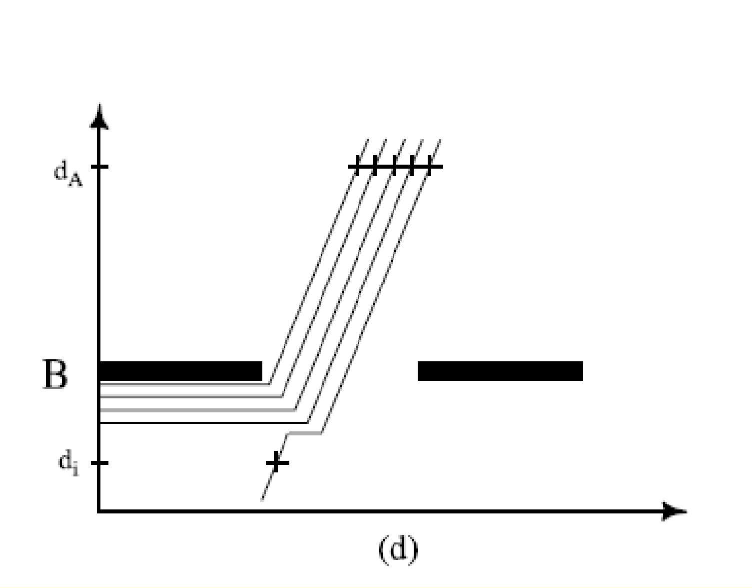


Figure (d): the vehicle arrives at detector di occurs after the signal has begun serving the desired phase, but a standing queue is present.

The arrival time is given by:

$$t_a = t_{d_i} + \max\{T_{d_i,S_B}, T_{q_i}\} + T_{S_B,d_A}$$

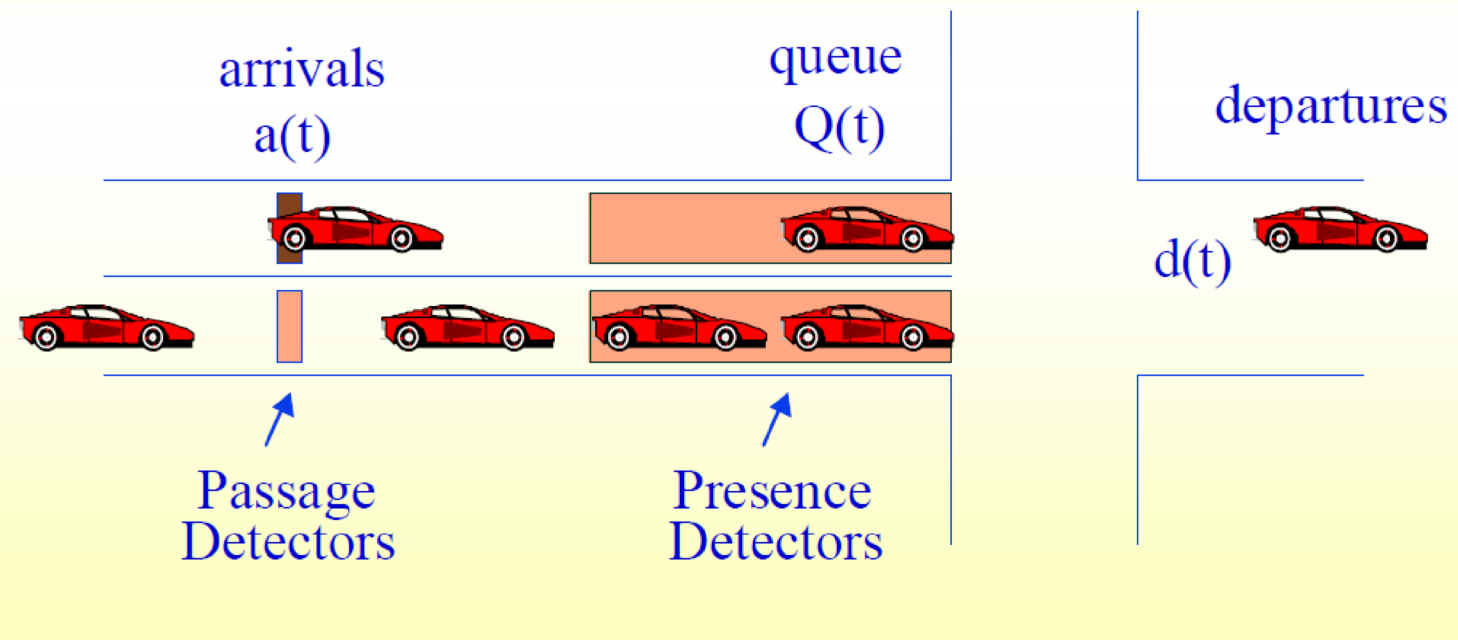
Arrival time at A = arrival time at di
+ + max of {travel time of di to B, queue delay}
+ travel time (stop line of Intersection B to dA)

Note: To estimate Queues

- Q(t+1)=Q(t)+a(t)-d(t)

  a(t) – count from a passage detector

  d(t) – depends on signal & departure rates



Prediction of Intersection Arrivals

- Given the estimate of the predicted arrival time, an arrival event at detector A can be anticipated with probability p_i^{BA}.

- The probability reflects the uncertainty that vehicle crossing the upstream detector i will actually travel on a route that will cross the detector at A.

- Then the expected number of arrivals at A can be predicted to be:

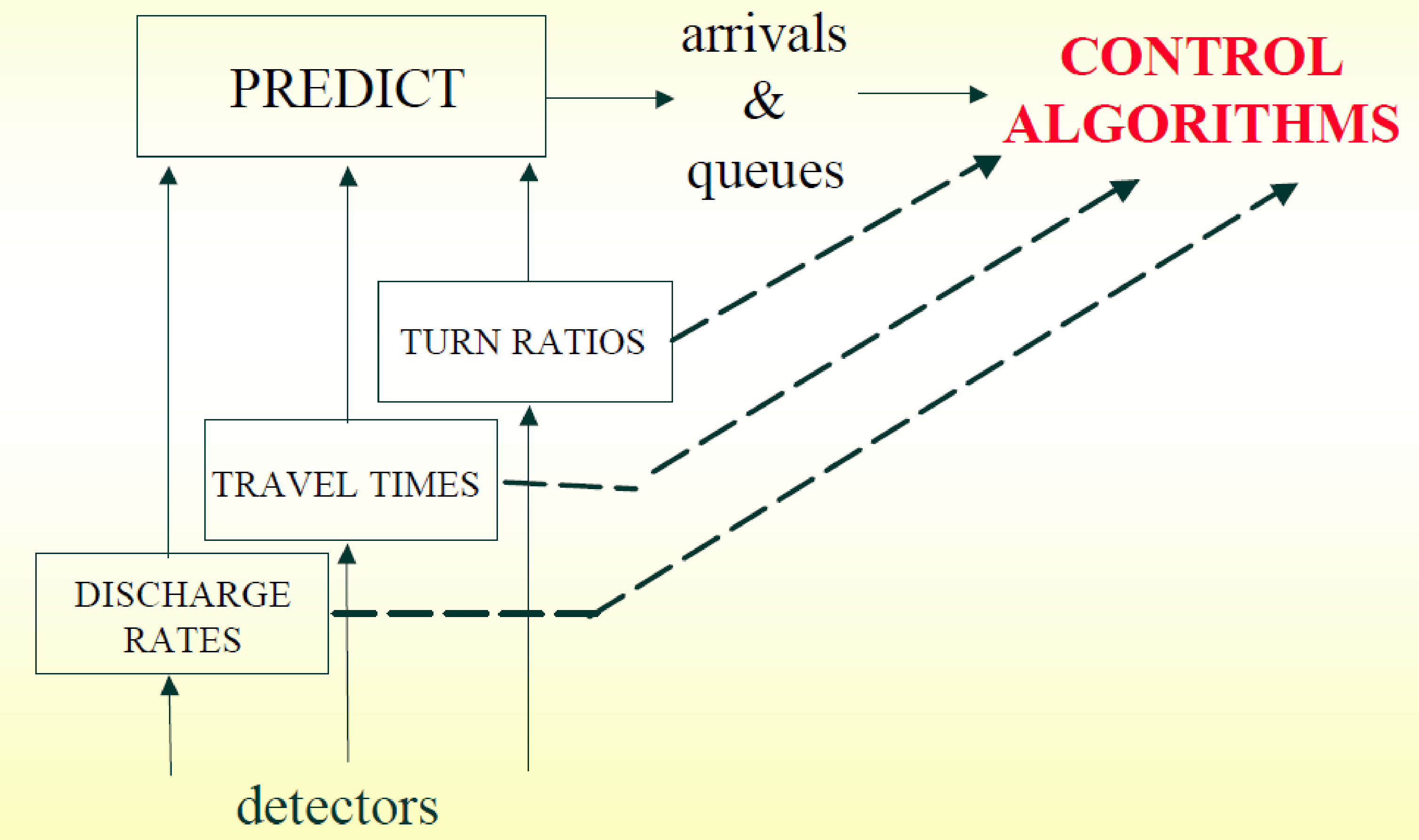
$$n_A(t_{d_A}) = \sum_{\forall i} \sum_{\forall t_p} p_i^{BA} n_i(t_p)$$

Estimation of parameters

Observe that to use the PREDICT model, three parameters need to be provided:

- (1) travel times on links (detector to detector) which depends on the link free-flow speed and current traffic volume
- (2) queue discharge rate which also depends on volumes
- (3) turning probabilities.

Estimation of parameters

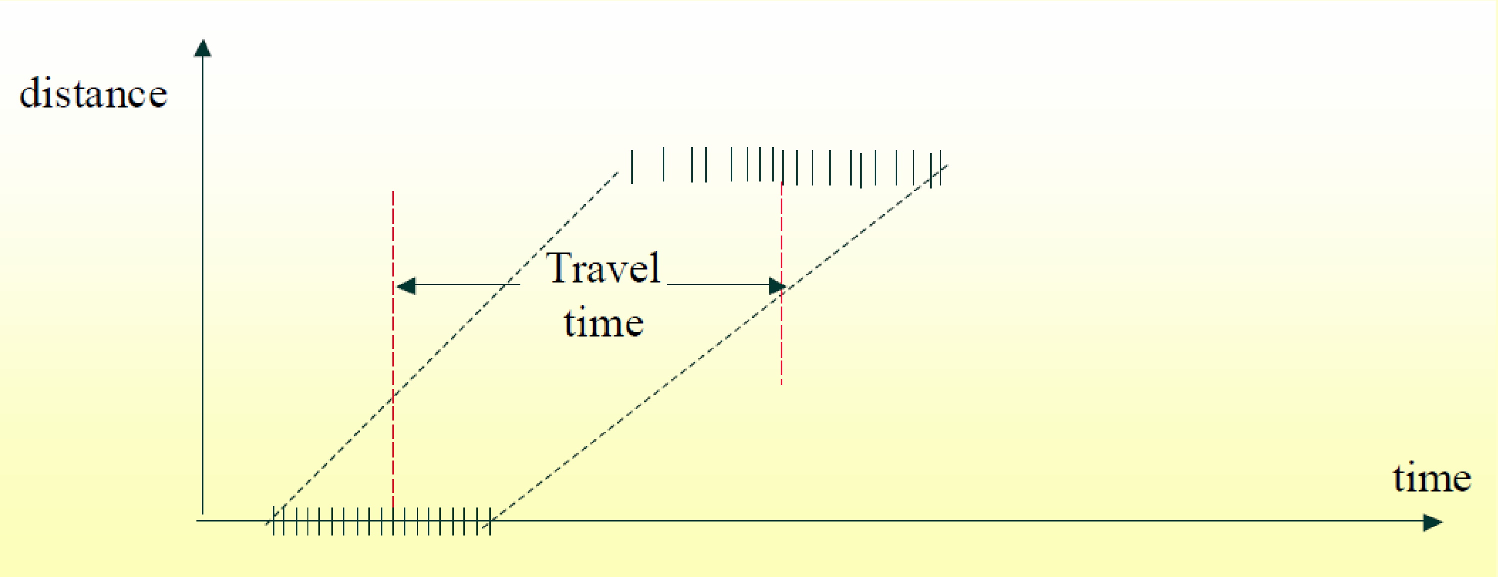


Turning Ratio Estimation

- 1. Based on Dynamic OD Estimation
- 2. Short Time (5 minutes)
- 3. Traffic Counts by Phases
- 4. Lanes and Geometry Dependent
- 5. Uses a Generalized Least Square

Travel Time Estimation

- 1. Based on Platoon Identification
- 2. By measuring Platoon Departure and Arrival Times



Discharge Rate Estimation

- Through-traffic queue discharge rates are effected by downstream through-traffic volumes, which can be easily measured.
- Left-turn queue discharge rates depend on opposing traffic volumes
- Right turn queue discharge rates depend on cross-traffic in that direction.
- These three discharge rates are initially given from calculated, but are then adjusted based on how well they predict remaining queues at the stop-bar presence detectors.

Intersection Control Logic

- RHODES uses a dynamic-programming (DP) based algorithm

- Effectively, the algorithm determines:

  - for a given phase order: A,B,C,D,A,B,C,D ...

  - what time durations should be given to Phase A, Phase B, ..., etc.

  -Considers a given decision time horizon T, with time increments of 1 sec.

Introduction of DP

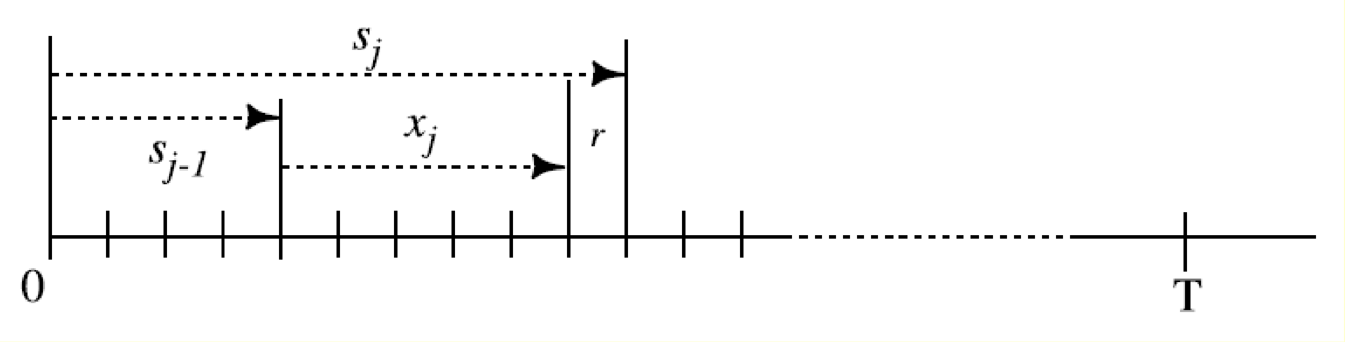
- Dynamic Programming is a method for solving complex problems by breaking them down into simpler subproblems.
- The key idea is to solve different parts of subproblems and then combine the solutions to reach an overall solution.
- The DP approach seeks to solve each subproblem only once, thus reducing the number of computations.

Introduction of DP

- DP usually refers to simplifying a decision by breaking it down into a sequence of decision steps over time. This is done by defining a sequence of value function V1, V2,...,Vn with an argument y representing the state of system.
- $V_n(y)$ is the value obtained in state y at the last time n.
- The value of Vi at earlier times i = n-1,n-1,...,1 can be found by working backwards, using a recursive relationship.
- The more detail of DP could be found in the reading materials.

The DP Algorithm in RHODES

- Decision variable: X_j (phase duration of stage j)
- Optimization state variable: S_j (time horizon with stage j)
- Incremental Value of Objective Function: $f(S_j, X_j)$
- Cumulative Value of Objective Function: $V_{j-1}(S_{j-1})$



$$V_j(s_j) = \min\{ f(s_j, x_j) + V_{j-1}(s_{j+1}) \mid x_j \in X_j \}$$

The DP Algorithm in RHODES

- Allow various objectives: stops, delays, queues
- Implementable in real-time
- Easily accommodates operational constraints:

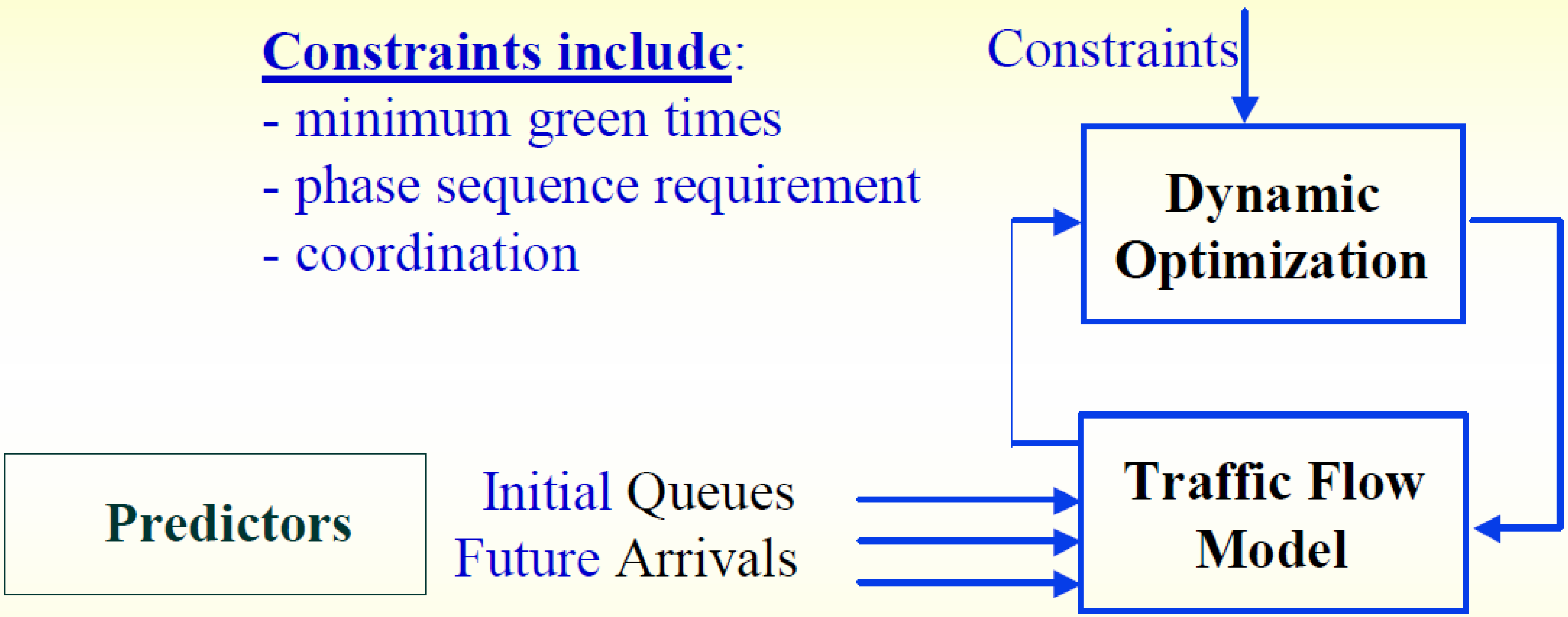
  - minimum green times

  - fixed/variable phase sequence

Note: If it is variable phase sequence, then the DP algorithm allows to skip phase by allocating zero time for the corresponding stage.

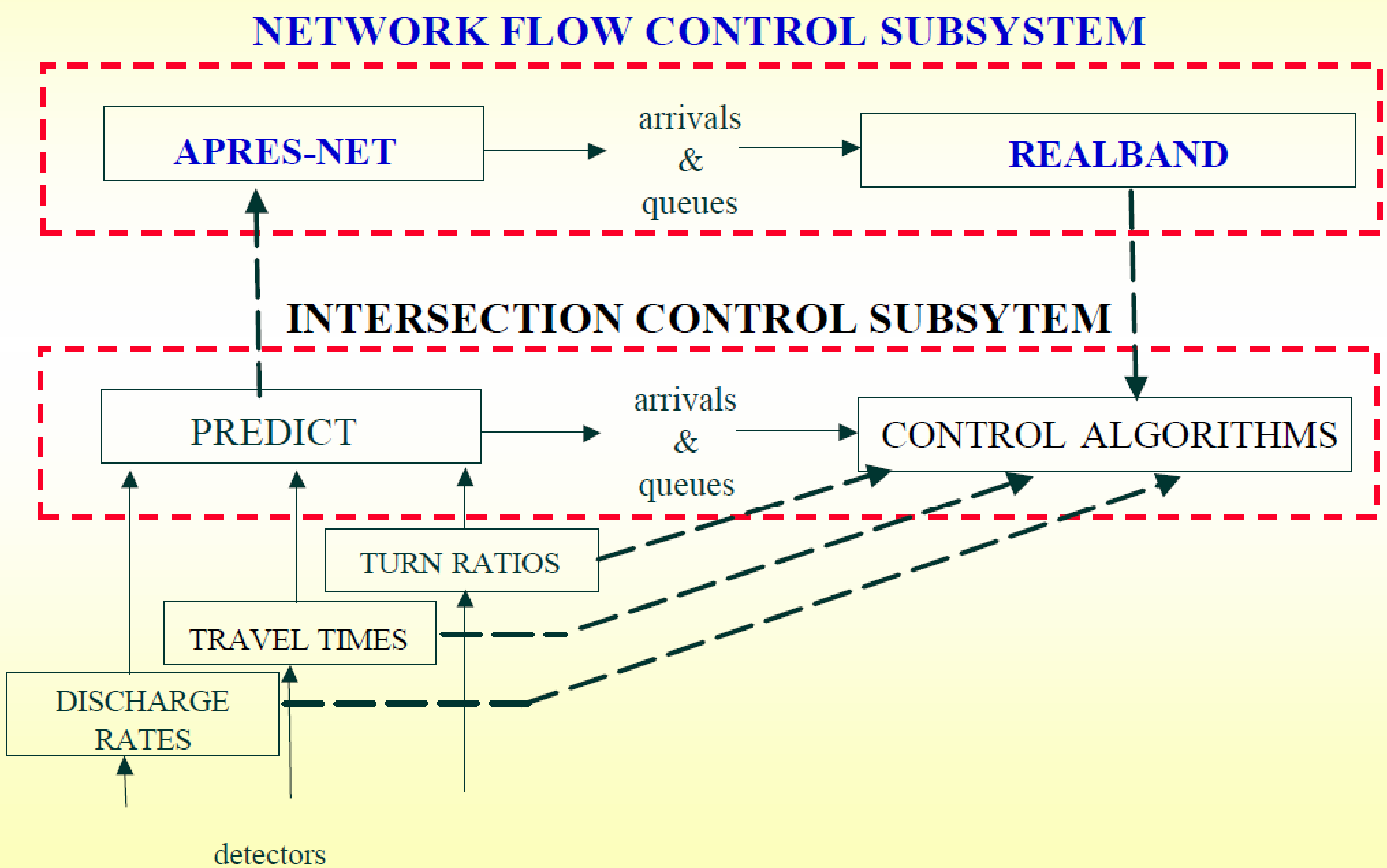
Summarizing
Intersection Control Decision-Making

Constraints include:
- minimum green times
- phase sequence requirement
- coordination



Traffic Flow Parameters
- start-up loss time
- queue discharge rates
- right-on-red rate
- permitted left turn rate
- turning proportions

Network Flow Control & Intersection Control



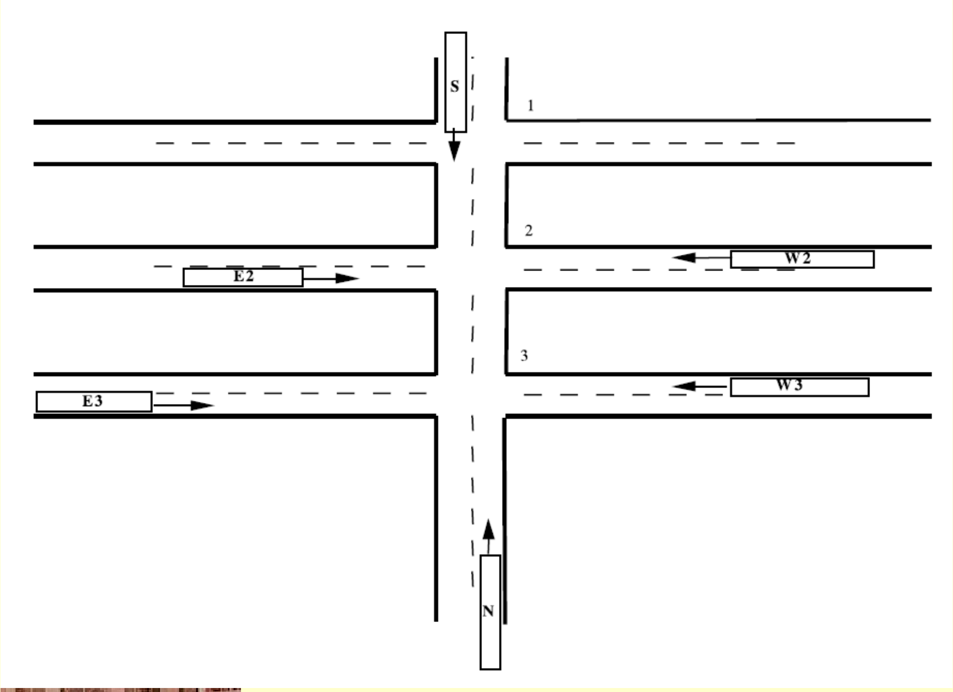
Network Control

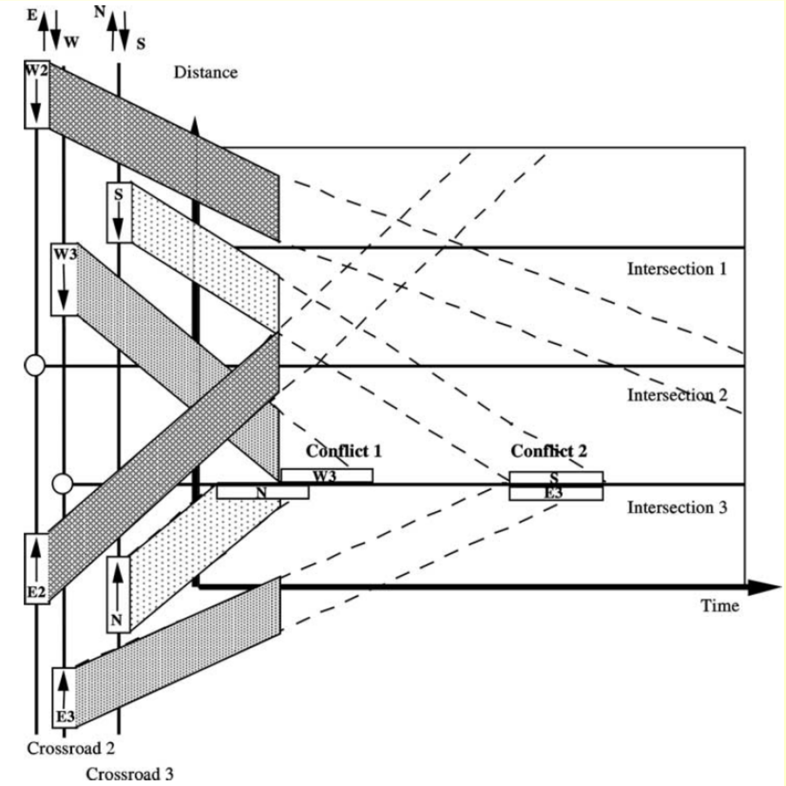
- APRES-NET: model for predicting the flow of platoons and evaluating the performance of each platoon conflict.

- REALBAND: Builds a binary decision tree for platoon conflicts.

Network flow prediction

- Propagate platoons by APRES-NET



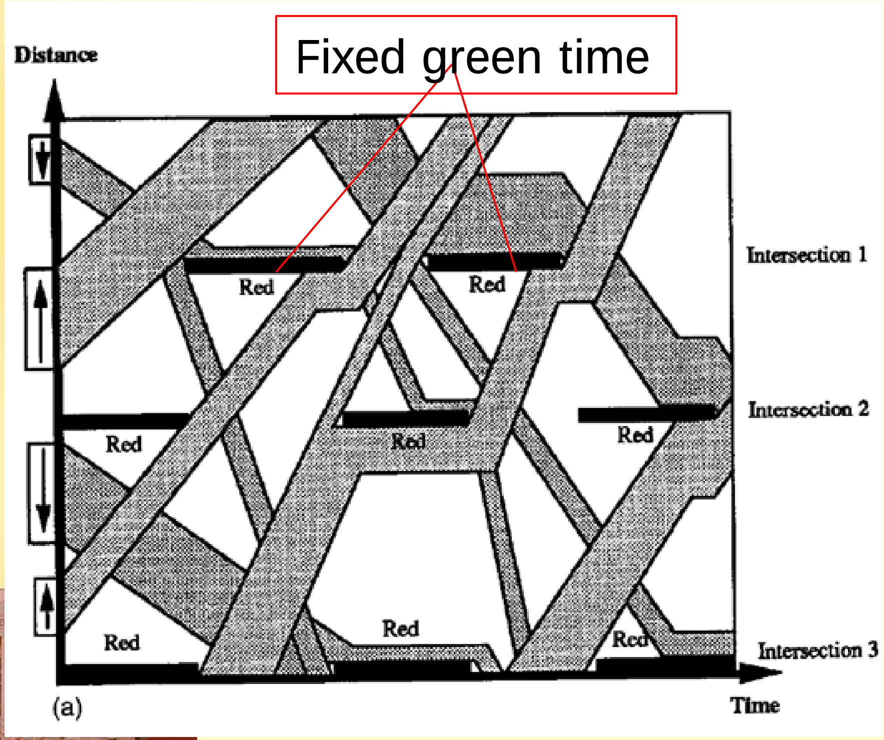


Note: this example would be used for further discussion.

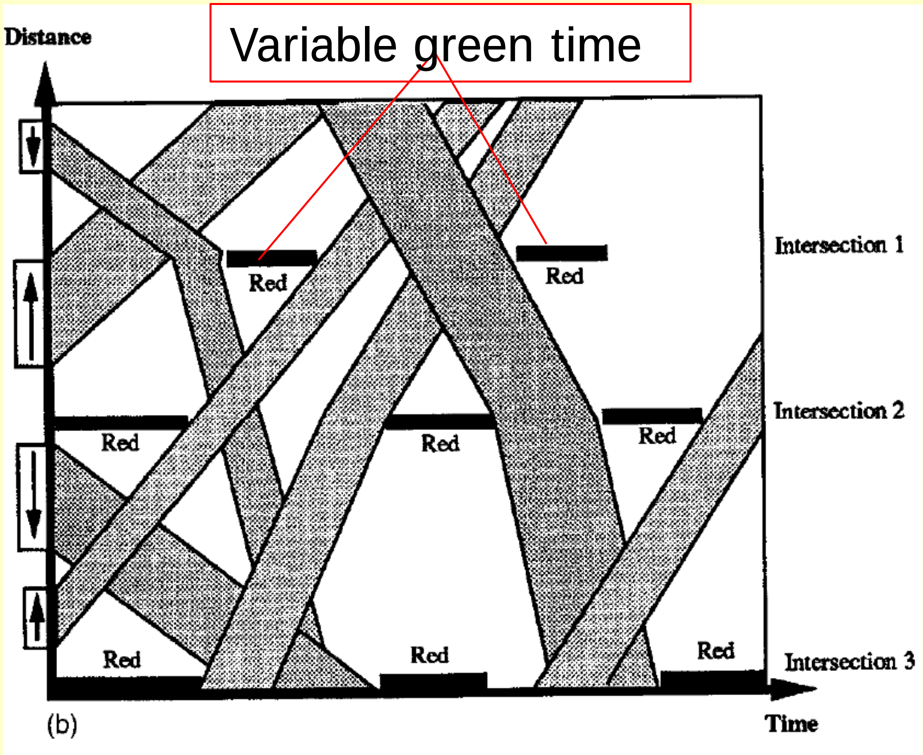
Figure 1

REALBAND (Network control)

- REALBAND identifies platoons and predicts their movement in the network by fusing and filtering the traffic data obtained in the last few minutes.



MAXBAND Concept



REALBAND Concept

REALBAND principle

- The signals are set so that the predicted platoons are provided appropriate green times to optimize a given performance criterion.
- If two platoons demanding conflicting movements arrive at an intersection at the same time, then either one or the other will be given priority for green time, or one of them is split to maximize the given measure of performance.

REALBAND principle

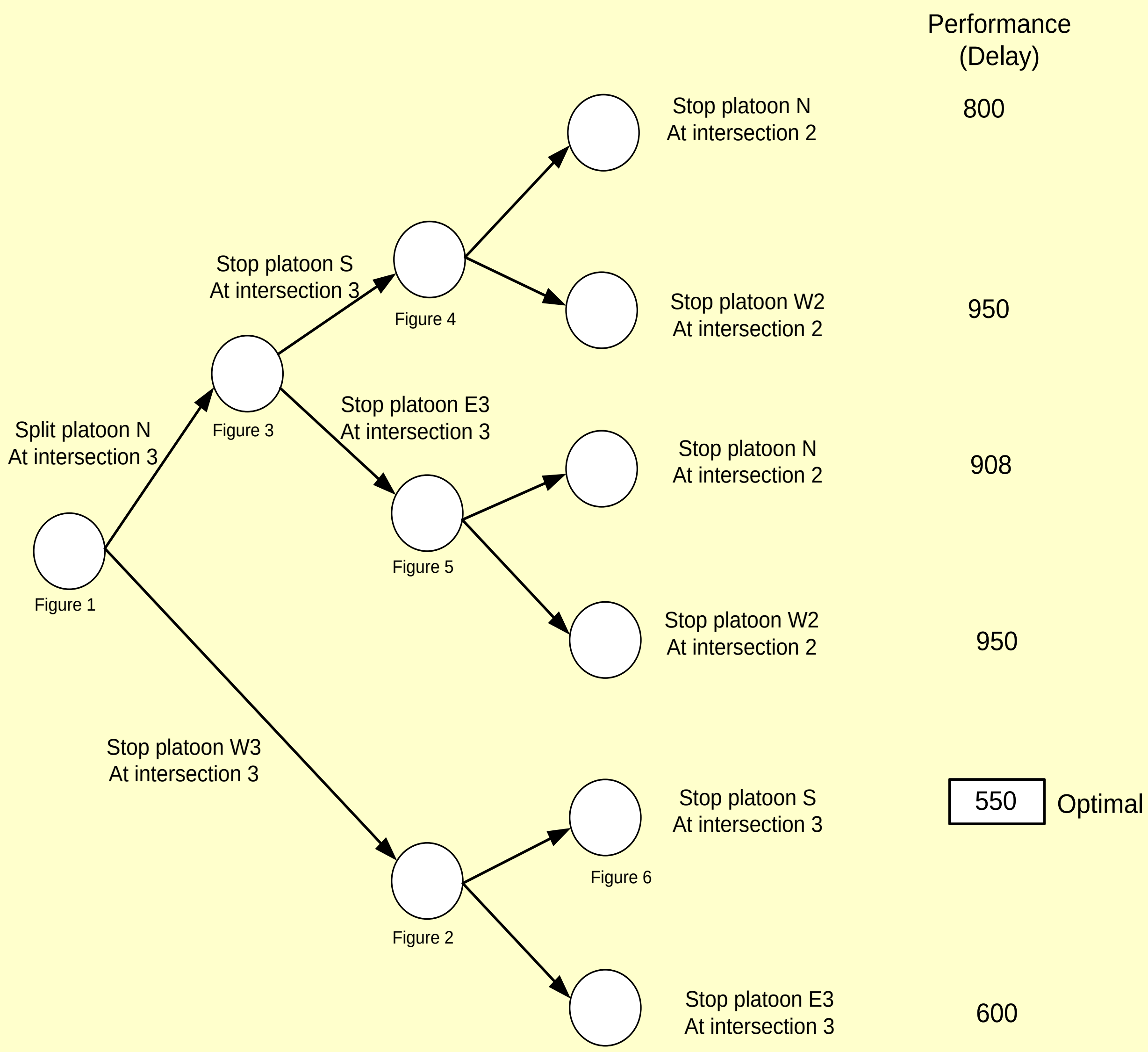
- REALBAND makes a forward pass in time.
- When a conflict arises a decision node in a tree is formed, the types of decisions at this node include:

  (a) give green time to Platoon A

  (b) give green time to Platoon B

  (c) split Platoon A (or B) .

Decision tree for the example



Illustrates of the Decision Tree Node

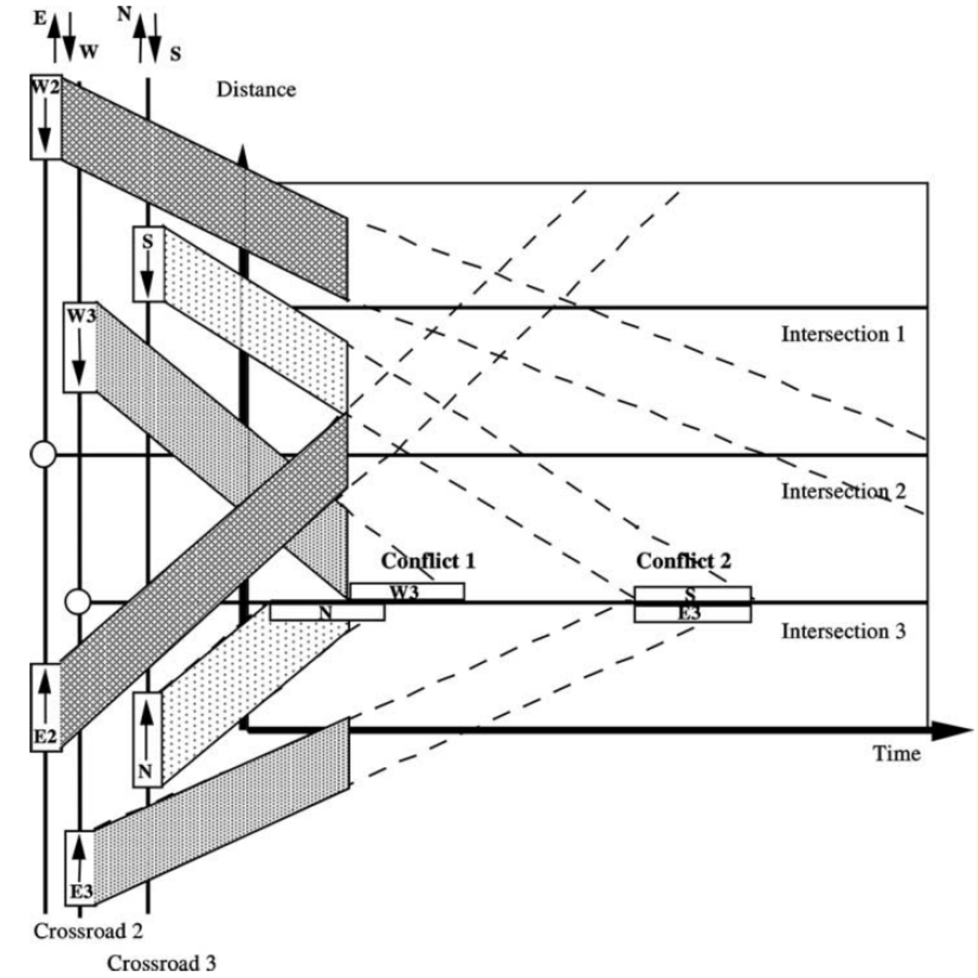


Figure 1: Propagate platoons by APRES-NET

Illustrates of the Decision Tree Node

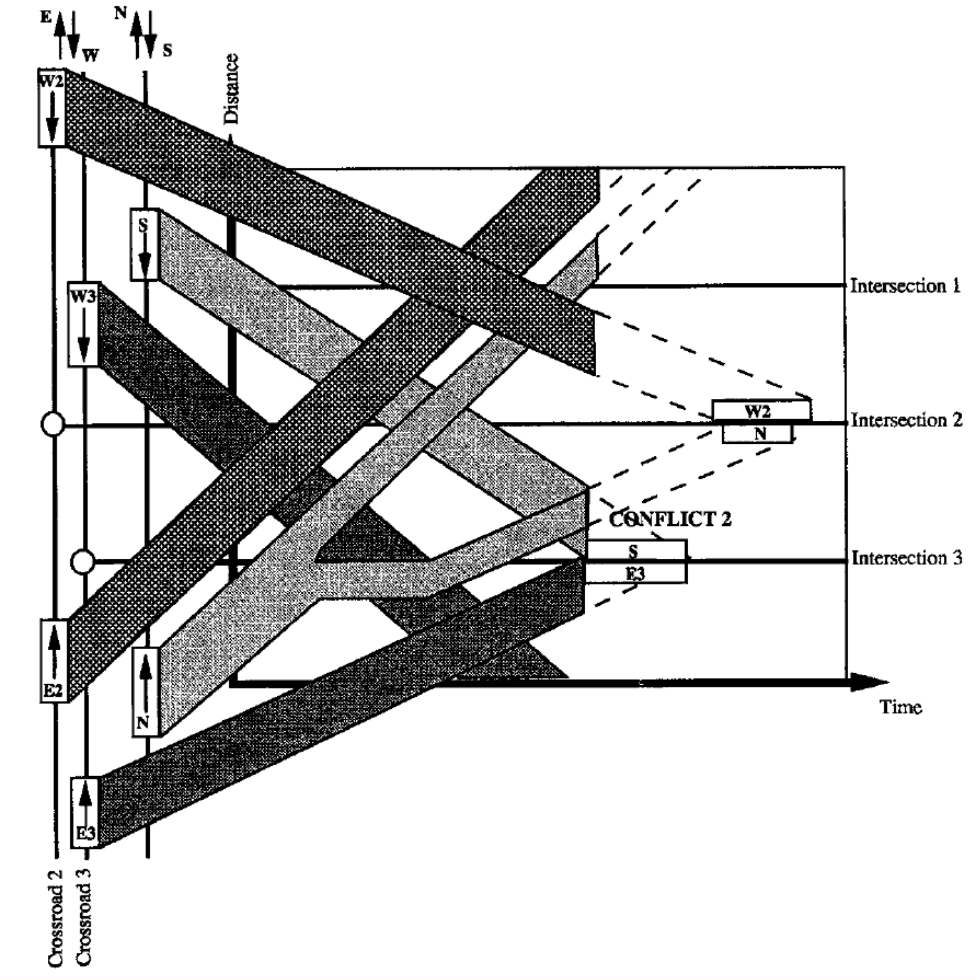


Figure 2:Stop platoon W3 at intersection 3

Illustrates of the Decision Tree Node

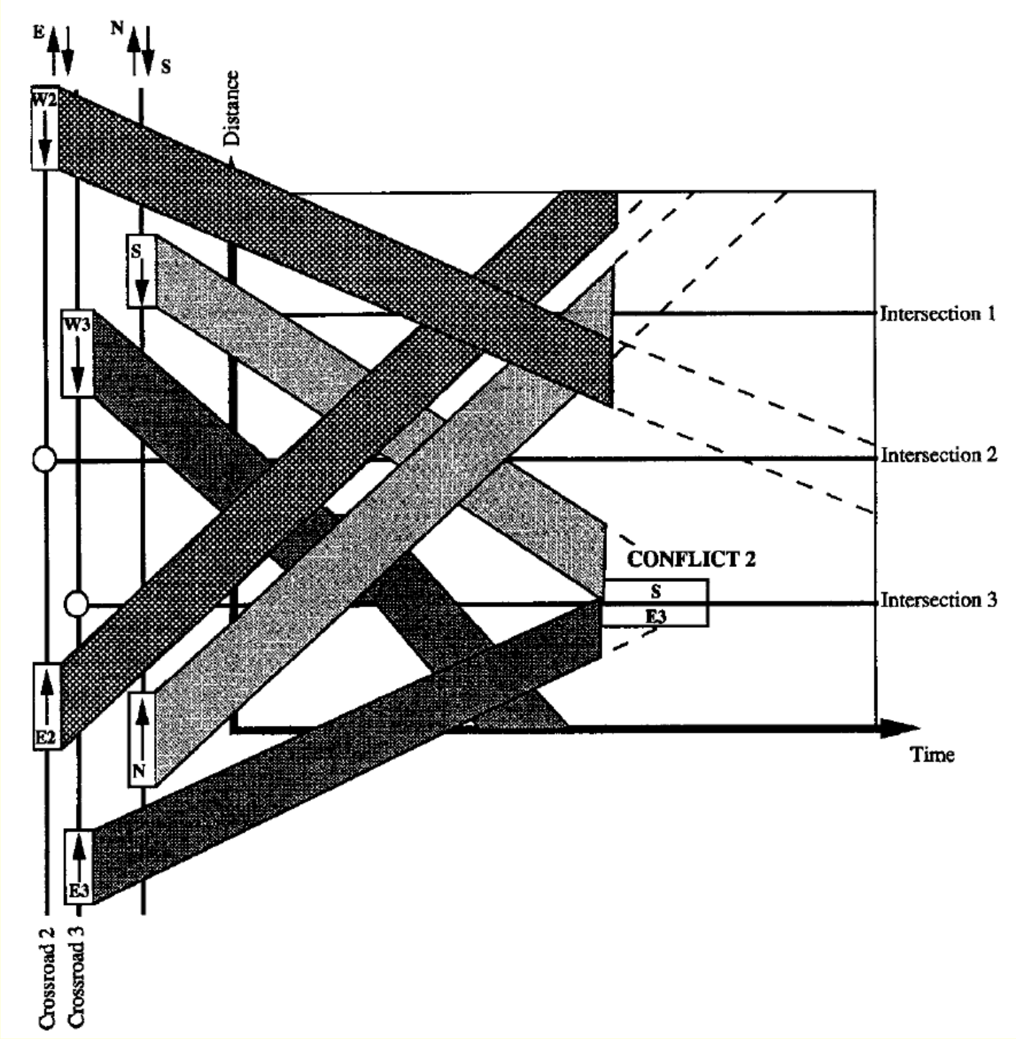


Figure 3: Split platoon N at intersection 3

Illustrates of the Decision Tree Node

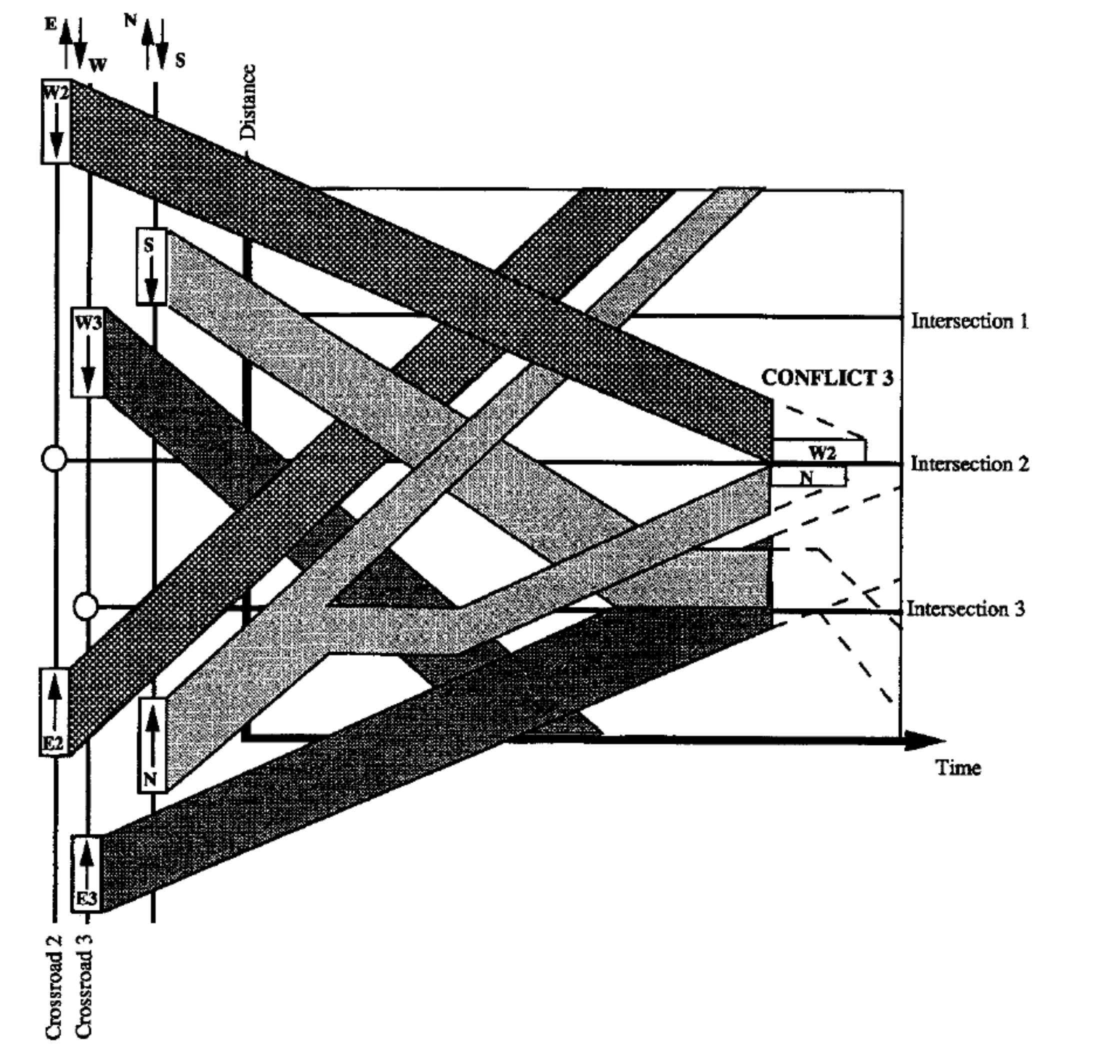


Figure 4: Stop platoon S at intersection 3

Illustrates of the Decision Tree Node

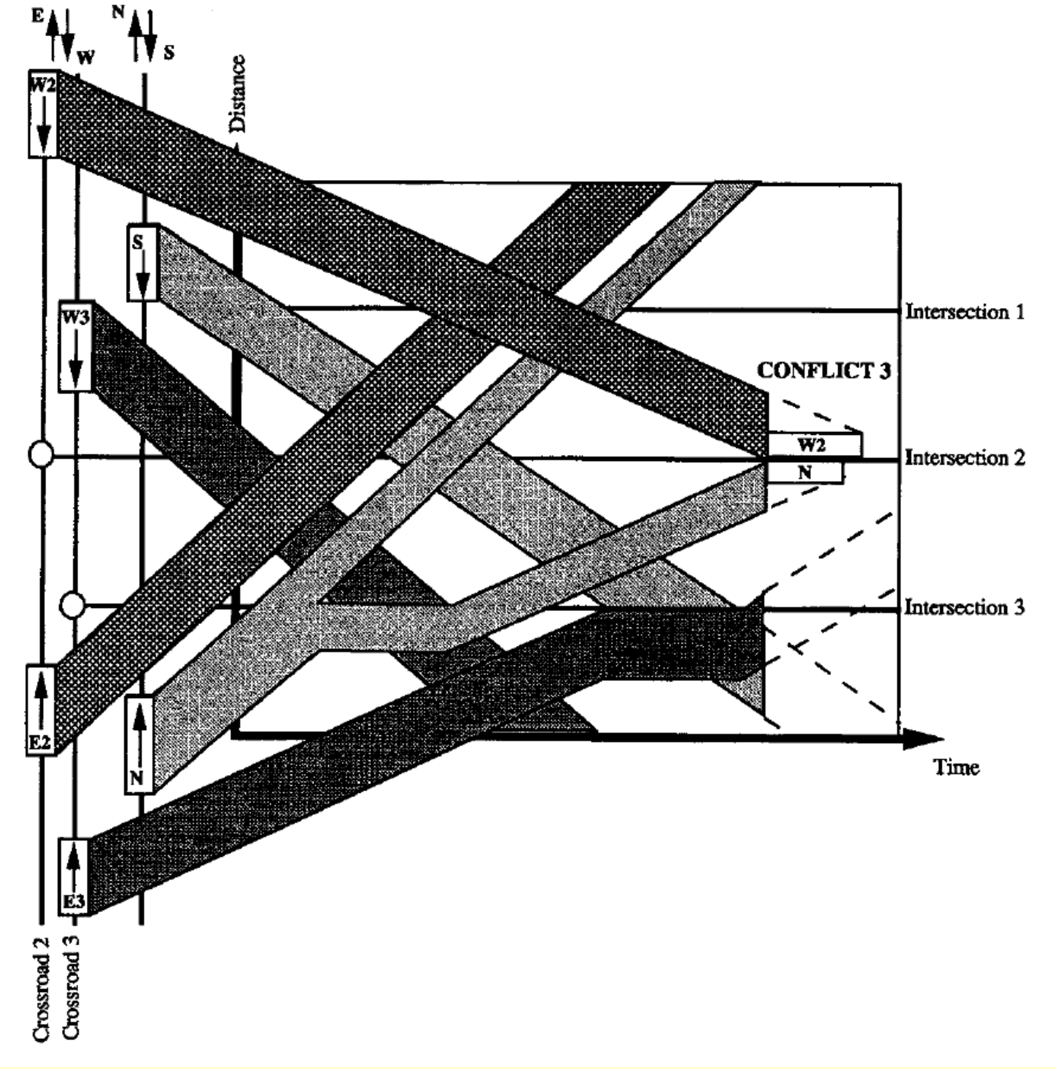


Figure 5: Stop platoon E3 at intersection 3

Illustrates of the Decision Tree Node

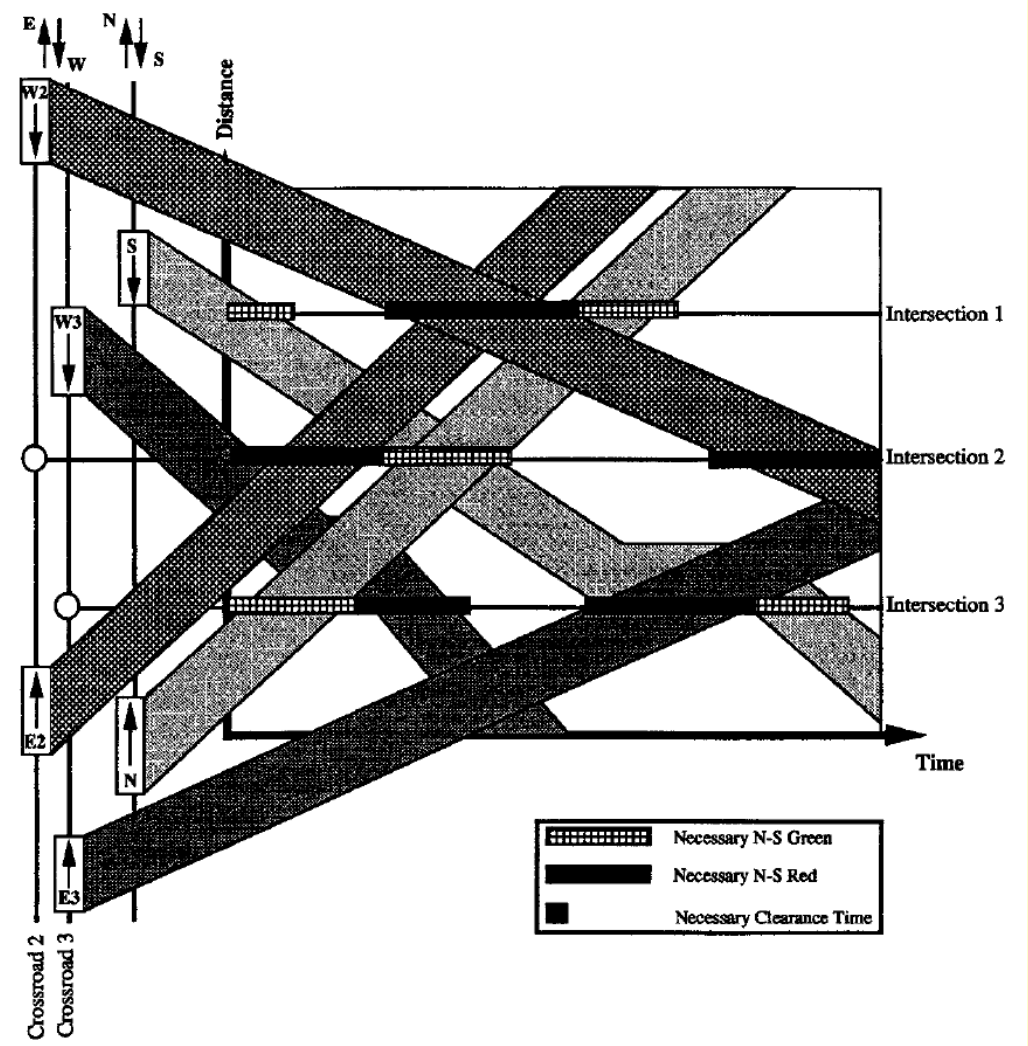
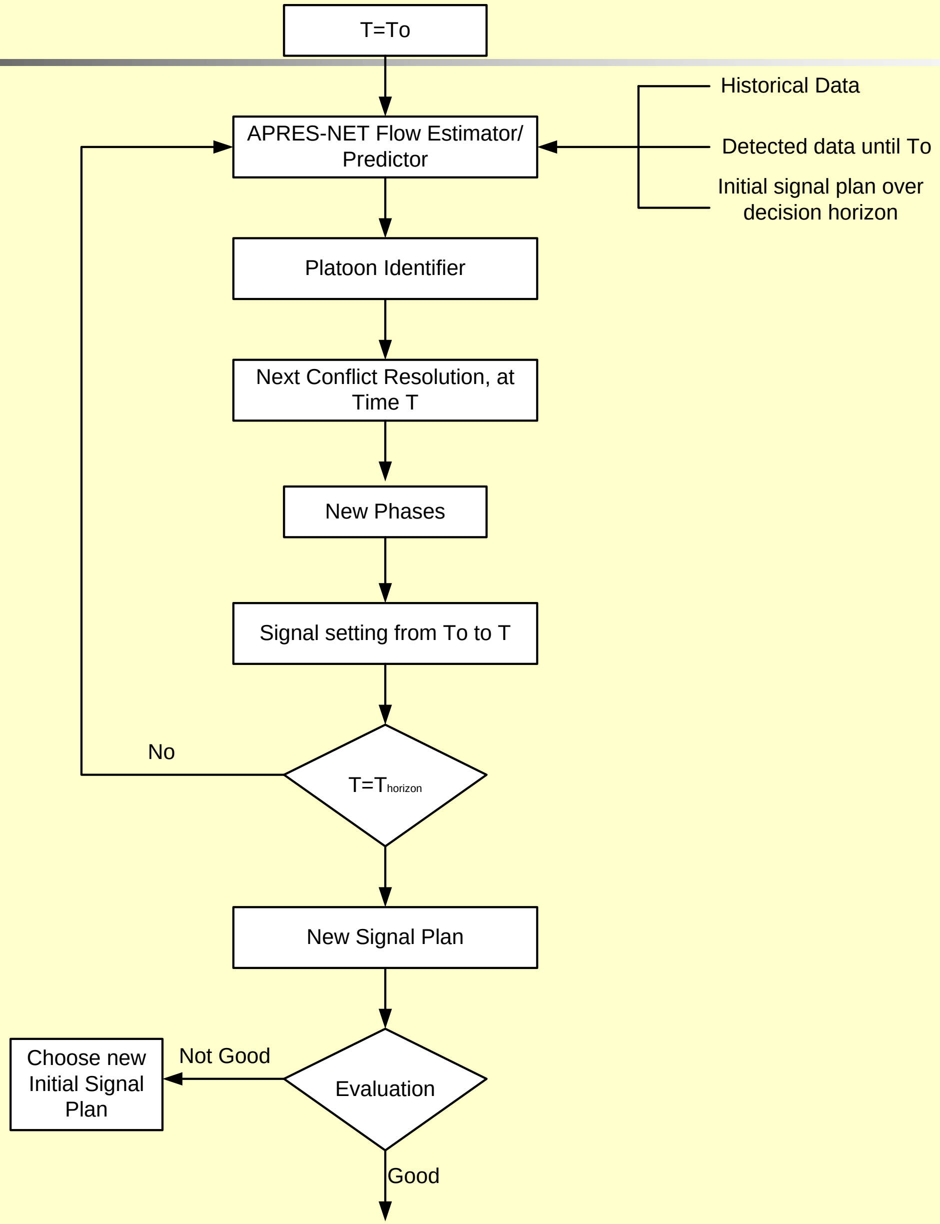


Figure 6: Stop platoon S at intersection 3

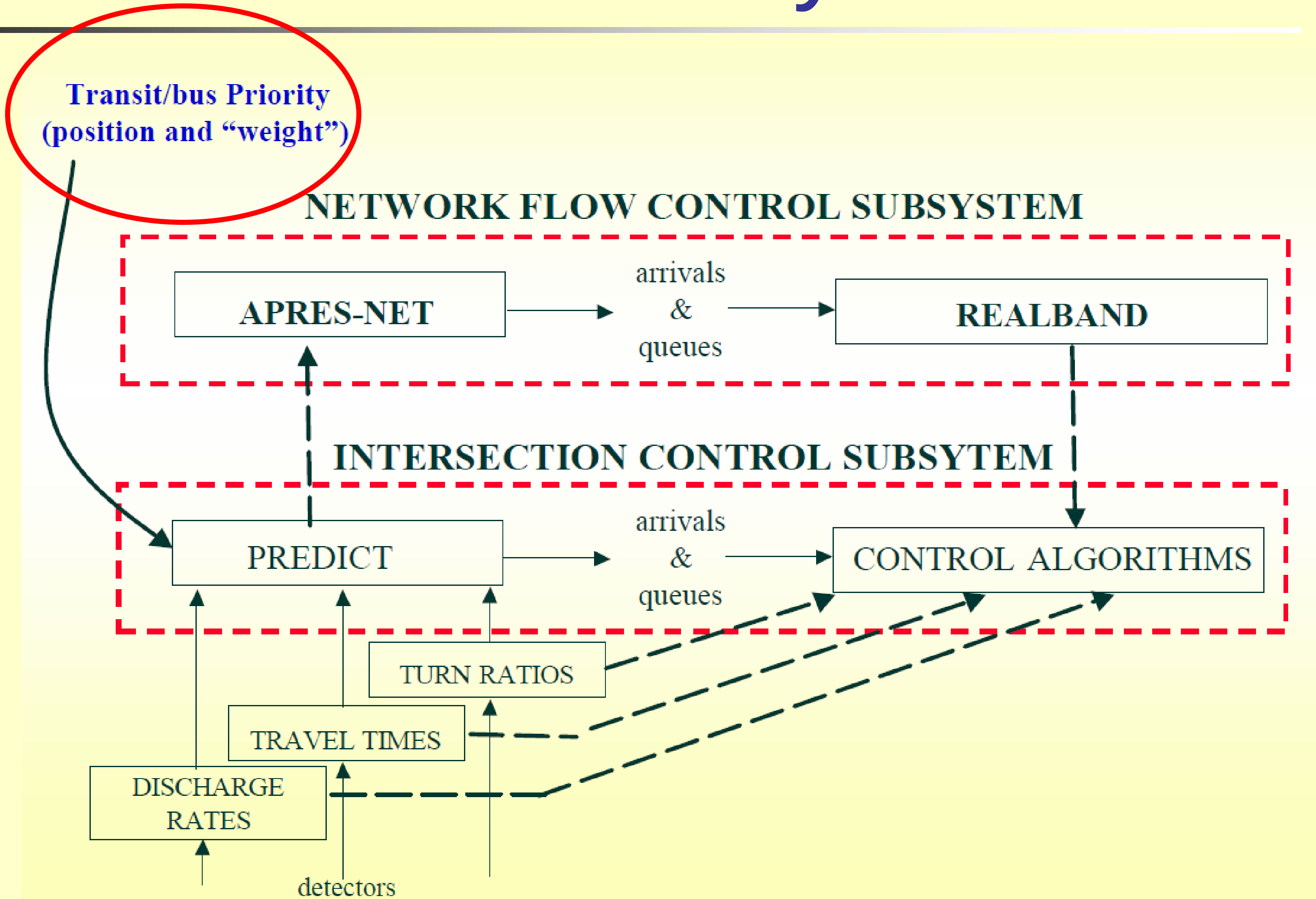
Measure of Performance

- The measure of performance is completed by APRES-NET model.
- APRES-NET model is a simplified traffic simulation model based on the same principles as the PREDICT model
- Compared with PREDICT model, APRES-NET propagates platoons of vehicles through a subnetwork instead of a signal vehicle.

Flow Chart of REALBAND



Transit/bus Priority



Transit/bus Priority

- In the intersection control strategy, each vehicle is treated alike by the dynamic program, and that is, they all have a identical "weight" of unity.
- For the bus priority control, RHODES gives each bus a variable weight that depends on the number of passengers and on how late is the bus.

Transit/bus Priority

- Then weight w_i for bus i given to RHODES was defined by the function:

$$w_i = n_i(1 + f_i)$$

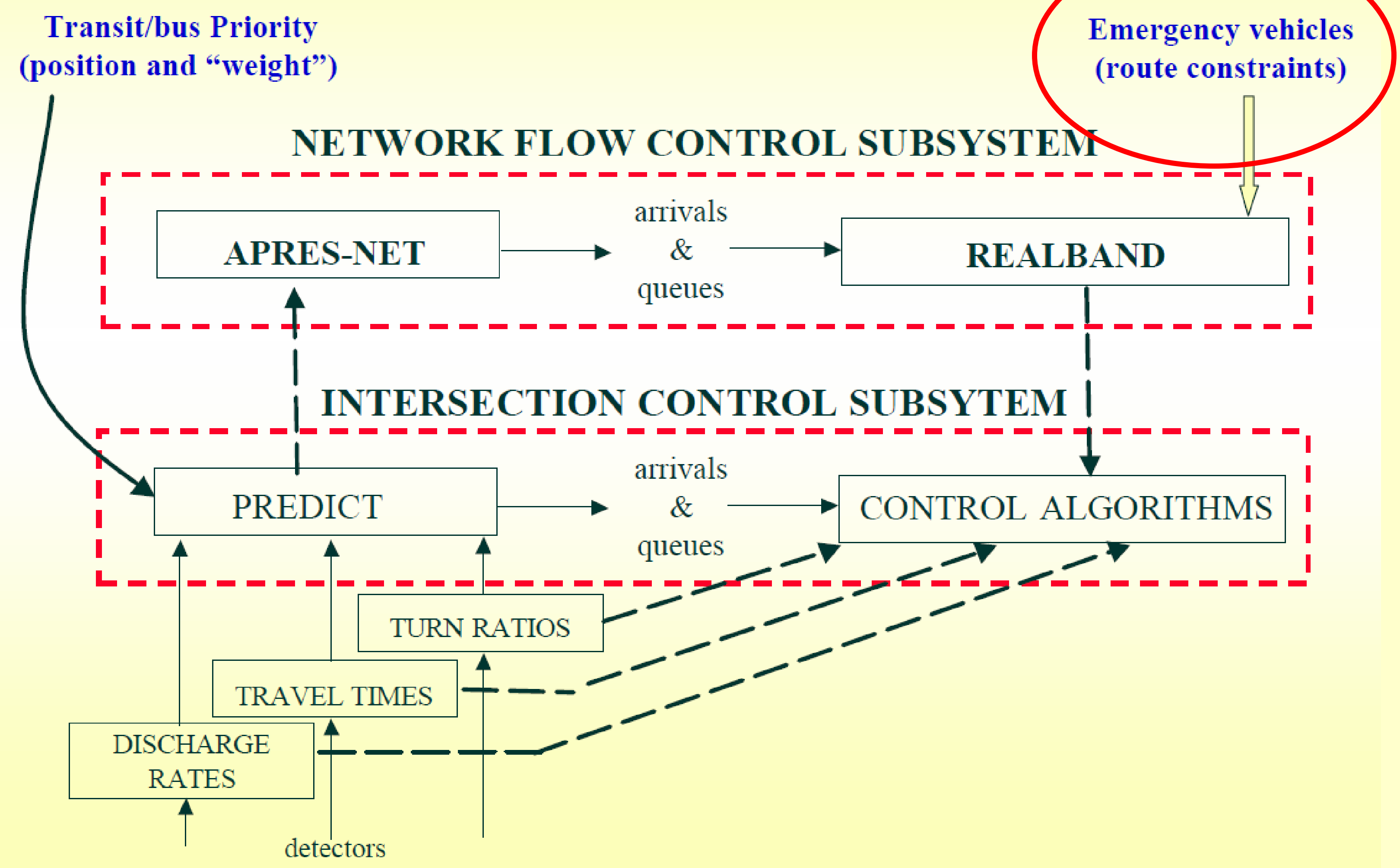
where delay factor

$$f_i = \begin{cases} 0 & \text{if lateness } d_i \le 0 \\ Kd_i & \text{if lateness } d_i > 0, \text{ where K is some constant} \end{cases}$$

n_i is the number of passengers on bus i.

With the inclusion of the bus passengers and the bus lateness in the computation of the objective function value, RHODES will tend to give higher priority for late buses with many passengers.

Emergency Vehicles



Emergency Vehicles

- Sometimes, emergency vehicles, such as ambulances, fir trucks are equipped with transponders that allow them to preempt signals in their route.
- In this case, the signals get preempted when there is a line of sight from the response unit to the signal heads and, once preempted, the signals transition into the required phase as quickly as possible.
- Emergency preemptions therefore result in considerable disruption of traffic patterns and add delays to other vehicles.

Emergency Vehicles

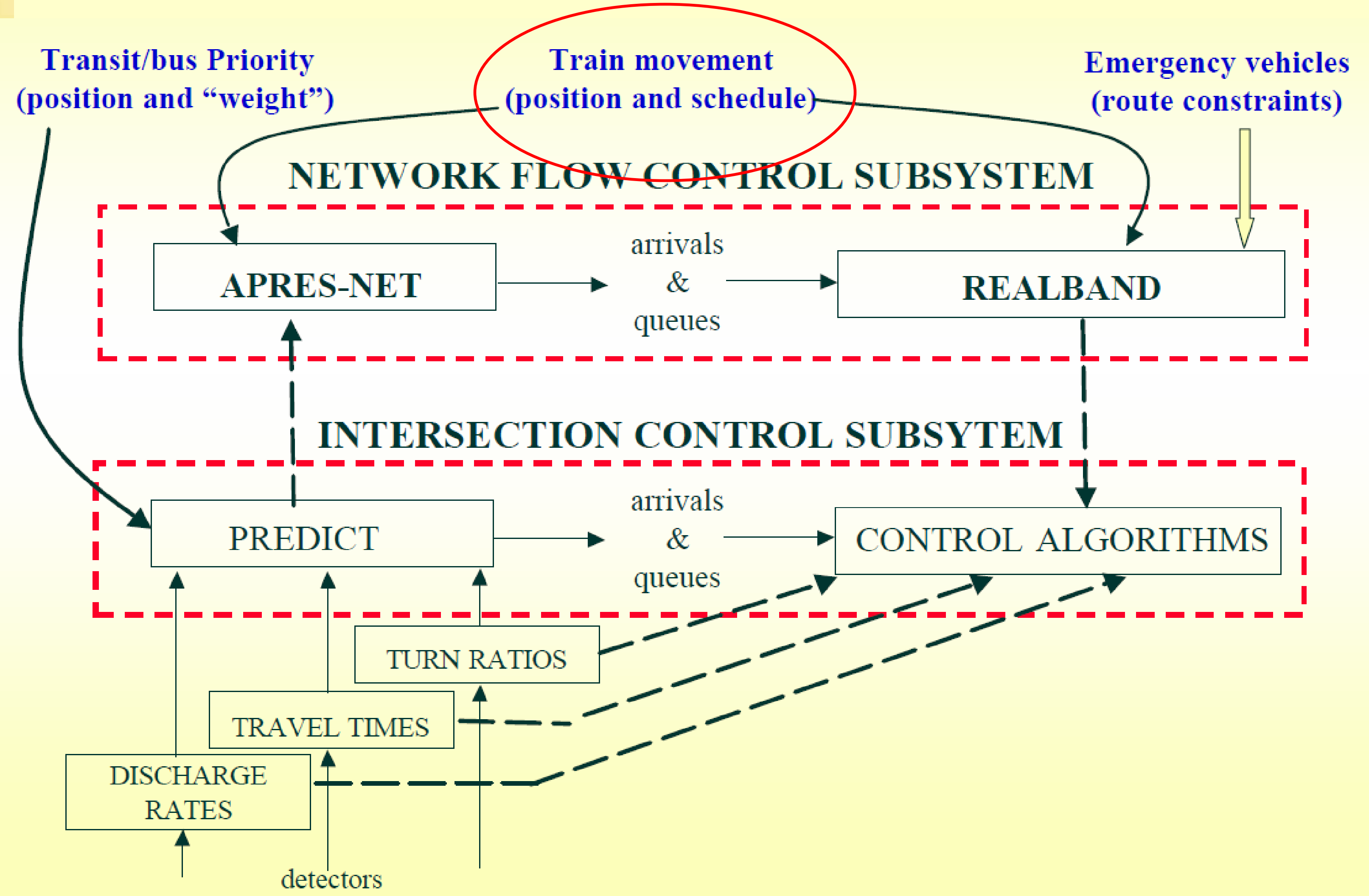
RHODES/CAPRI system allows to:

- (1) compute a real-time dynamic shortest path from emergency vehicles' origin to destination;
- (2) advise the responding unit of the path; and
- (3) schedule the signal phasing so that an emergency pathway is provided to the response unit while the resulting delays of other vehicles are minimized.

- Also, the RHODES/CAPRI system will know when the emergency vehicle will be at an intersection, will constrain the phase to be green for the responding unit, and use DP to schedule the remaining phases to minimize delay for the predicted vehicle arrivals.

- Effectively, the emergency vehicle is treated as a "platoon" on a specified route, and CAPRI computes the needed phase timings.

Rail/Train Preemption



Rail/Train Preemption

- At the highway-rail intersections, trains always have right of way; that is trains preempt the signals at grade crossings so that trains get a green phase while vehicles get a red phase.

- Research dealing with traffic management for at-grade railway crossing has had essentially two objective:

  (1) to reduce the risk of incidents at highway-rail intersections and

  (2) to minimize vehicular travel times across these intersections and prevent excessive wait times or bottlenecks.

Rail/Train Preemption

- Given the predicted preemption of the signals at the grade crossing, PHODES/CAPRI could schedule the phases in the neighborhood of the rail-highway intersection so that predicted and detected vehicles can go more efficiently.

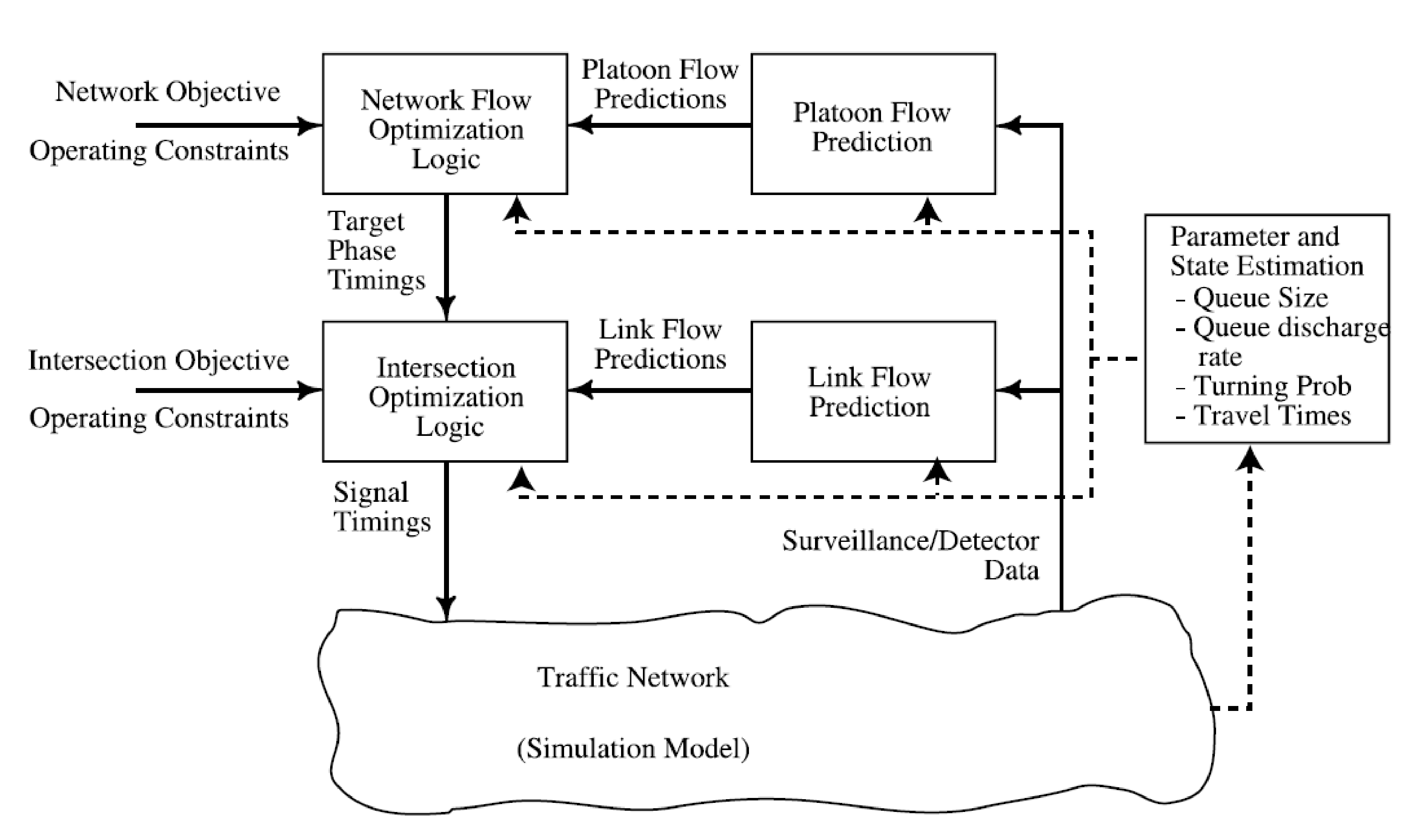
- PHODES treats the train as a predicted platoon where the signals at the intersections have a constrained red phase for arriving and waiting vehicles during the predicted train crossing.

RHODES prototype

To summery, the PHODES prototype logic consist of five modules:

- (1) Intersection Optimization Logic
- (2) Link Flow Prediction Logic
- (3) Network Flow Optimization Logic
- (4) Platoon Flow Prediction Logic
- (5) Parameter and State Estimation Logic

RHODES prototype



References

- Introductions and Reports

- Head, K.L., P. B. Mirchandani, and D. Shepherd. 1992. A hierarchical framework for real-time traffic control. Transportation Research Record 1360. National Research Council,Washington, D.C. p.82-88.
- K. Larry Head, Pitu B. Mirchandani, and Steve Shelby. “The RHODES prototype: a description and some results”. Presented as 77th Annual Meeting of the Transportation Research Board. 1998.
- Pitu Mirchandani, Larry Head. “RHODES: A Real-time Traffic Signal Control System: Architecture, Algorithms and Analaysis”. 1998.
- Pitu Michandani, Fei Yue Wang. “Rhodes to Intelligent Transportation System.” Intelligent Systems, IEEE, 2005.
- Larry Head, Pitu Mirchandani. “RHODES-ITMS”. Report: Sponsored by Arizona DOT. AZ-SP-9701.
- Head Larry, Pitu Mirchandani. “Final Report – RHODES Projects: Phase II (A)”. Report FHWA-AZ94-383. Arizona DOT.

References

- Algorithms

- Mirchandani, P., and L. Head. A Real-Time Traffic Signal Control System: Architecture, Algorithms, and Analysis. *Transportation Research Part C*, Vol. 9, No. 6, 2001, pp. 415-432.
- Head, K.L. 1995. An event-based short-term traffic flow prediction model. Transportation Research Record 1510. National Research Council, Washington, D.C. p.45-52.
- Mirchandani, P.B., K. L. Head, A. Knyazyan, and W. Wu. 2000. An approach towards the integration of bus priority, traffic adaptive signal control and bus information/scheduling system. Paper presented at the 8th International Conference on Computer-Aided Scheduling of Public Transportation at Berlin, Germany.
- Pitu B. Mirchandani and David E. Lucas. “Integrated Transit Priority and Rail/Emergency Preepmption in Real-Time Traffic Adaptive Signal Control”. Intelligent Transportation System, 8:101-115, 2004.
- Paolo Dell Olmo and P B Mirchandani. “Realband: An Approach for Real Time Coordination of Traffic Flows on Networks”. Transportation Research Record 1494. 1995.