Adaptive control in the Peek ATC1000 Controller

By

Donald M. Maas, Jr.

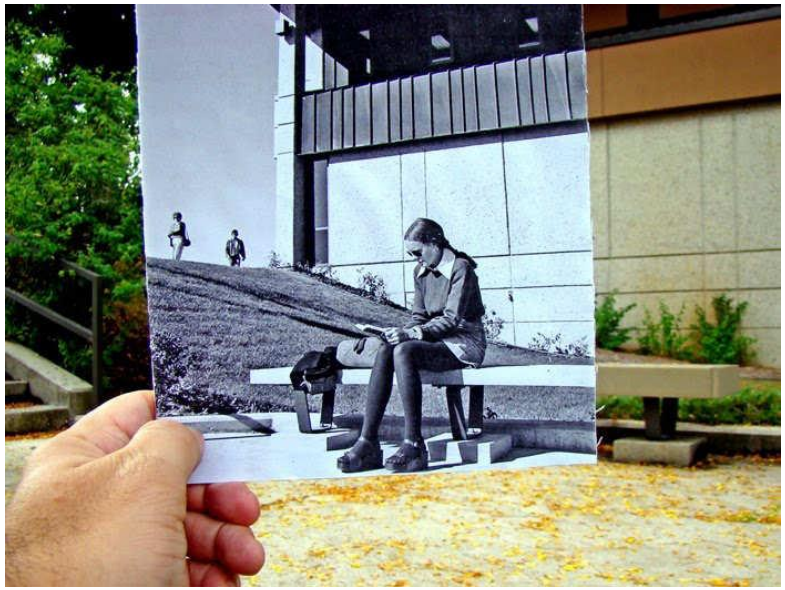
Senior Project Engineer

Peek Traffic

August 2012

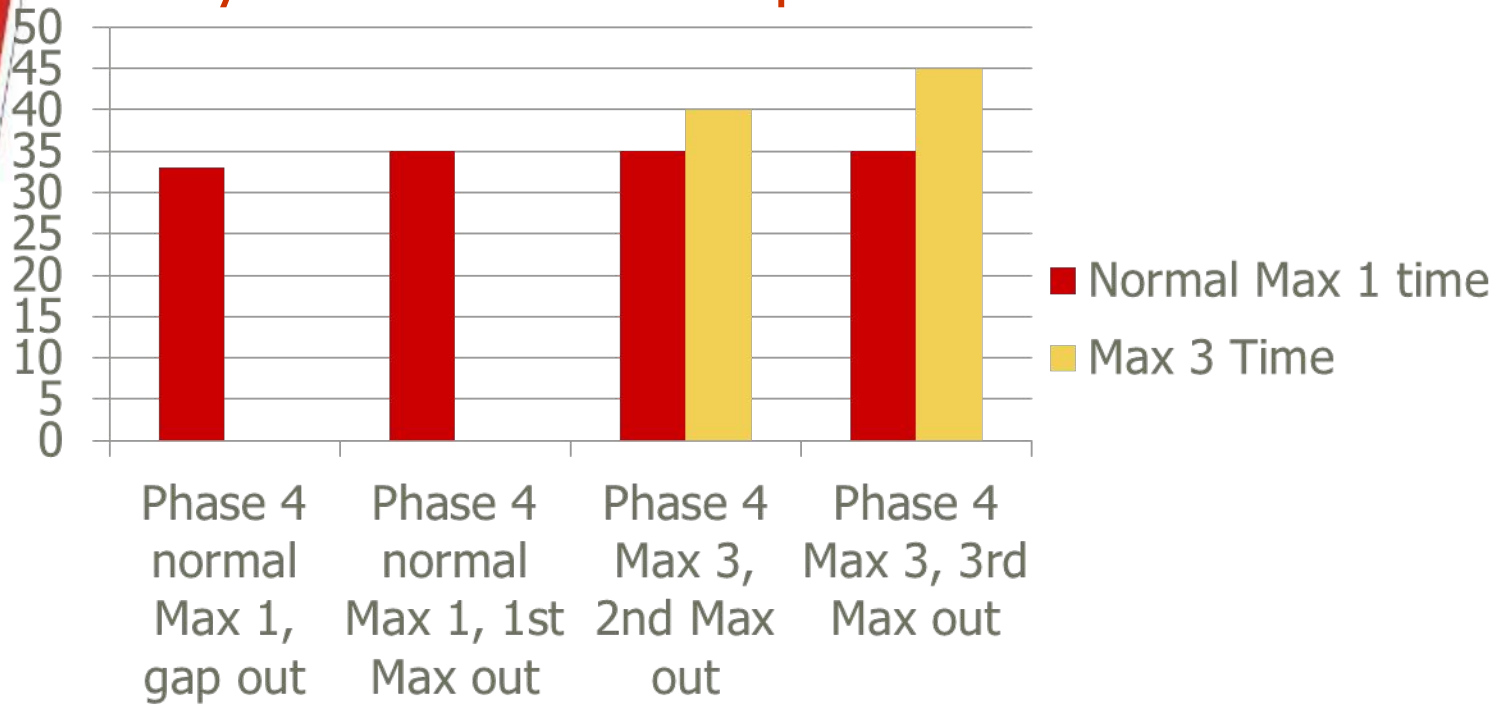
Looking Back…

- Back in the TS1 standard (circa 1989), the first 'Adaptive' algorithm appeared in traffic controllers
- It was called 'Dynamic Max 3' – the concept was simple, based on Max outs on a phase the controller would 'add' time – based on Gap outs, it would 'reduce' time



Dynamic Max 3 example on phase 4

- Dynamic Max 3 example…



Dynamic Max in modern controllers

- It is still supported in today's NTCIP machines
- Supported by the following objects…
  - phaseDynamicMaxLimit
  - "<Definition> This object shall determine either the upper or lower limit of the running max in seconds (0-255 sec) during dynamic max operation.

  - phaseDynamicMaxStep
  - "<Definition> This object shall determine the automatic adjustment to the running max in tenth seconds (0-25.5)

  Note: Fixed at 2 Max outs to increase and 2 Gap outs to decrease

Now, let's review Adaptive control in the ATC1000…

Introduction to the Peek ATC family

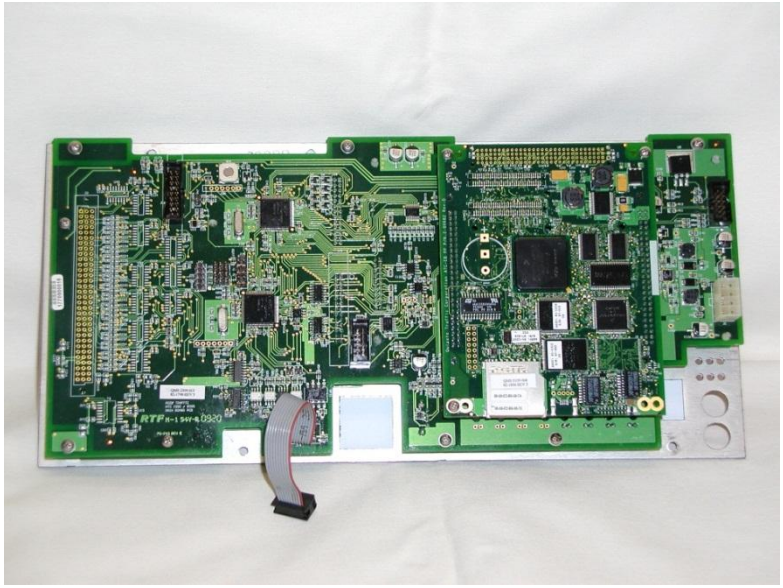
- Built entirely around the ATC 5.2b standard
- Offered in 2 models, one for the NYC CBD cabinet and the other in the NEMA TS2-1/TS2-2 form

The ‘goals’ of the ATC standard

- The goal of this standard is to provide an open architecture design for the next generation of transportation controller applications.
- First, the design specified in this standard is based on the concentration of computing power in a single component (the engine board) that is interchangeable with engine boards designed by other manufacturers.
- Second, the standard provides for required and optional features, all of which are based on open standard, common protocol communication standards.
- Third, the standard is responsive to the functional requirements

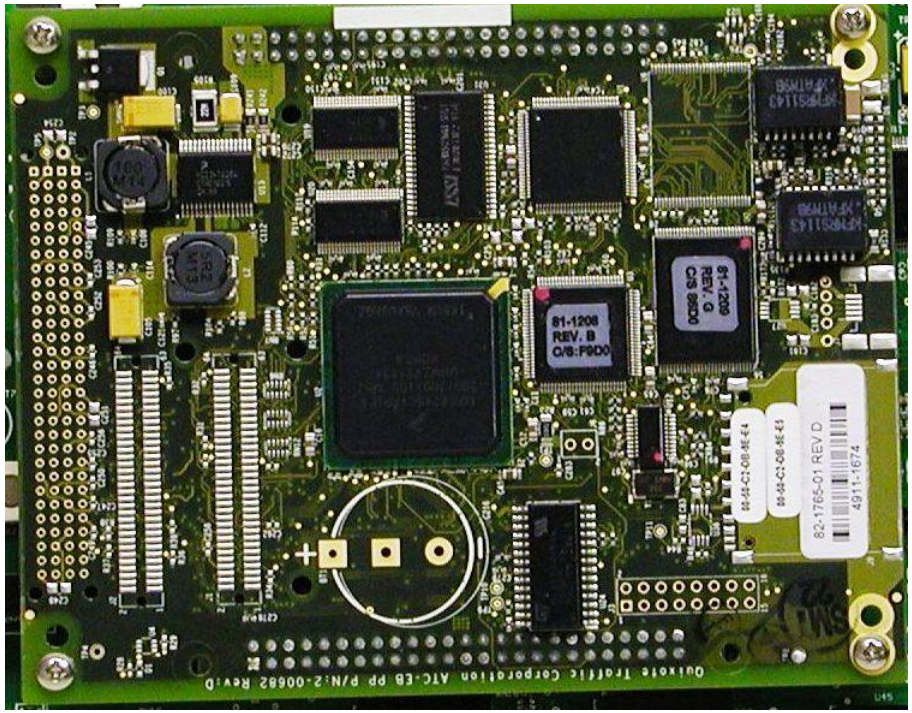
Form/Fit/Function

- The ATC provides for easy hardware upgrades to adapt to newer processors, operating systems and increased memory size and speed. It does this by requiring that the engine board (CPU module) conform to a specific physical form and pin-out interface
  - Shown mounted on host board



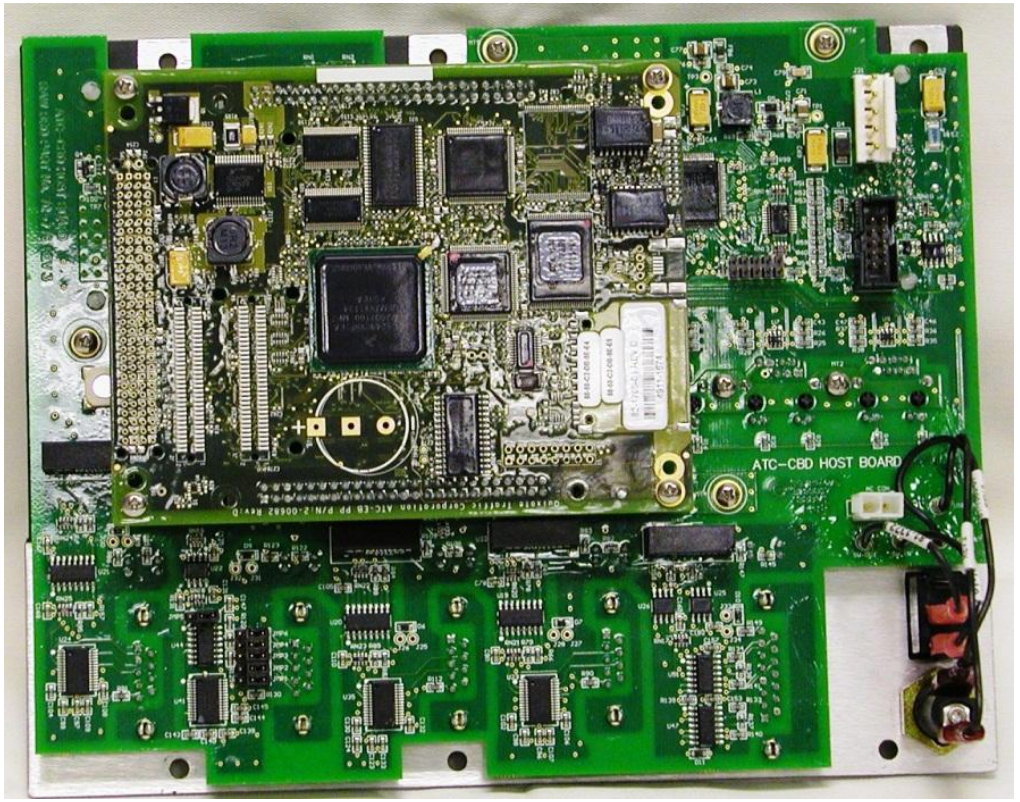
Engine Board (1)

- All computational functions are concentrated on an engine board within the ATC that meets a set of defined minimum requirements…



Engine Board Minimum Requirements

- CPU and RAM memory
- FLASH memory storage
- Serial ports
- Ethernet interface
- Clock/calendar maintenance
- Board support package



NTCIP operation (1201/1202)

- Since the ATC is a hardware standard, Peek coupled the powerful NTCIP standard software on the platform to create a ‘standards based’ solution

What exactly does adaptive mean?

- According to Dictionary.com
  - to adjust (someone or something, esp oneself) to different conditions, a new environment, etc…
  - In the context of traffic control, it is the ability of the controller unit or the central system to adjust (Adapt) the intersection timings based on current conditions

How does adaptive work?

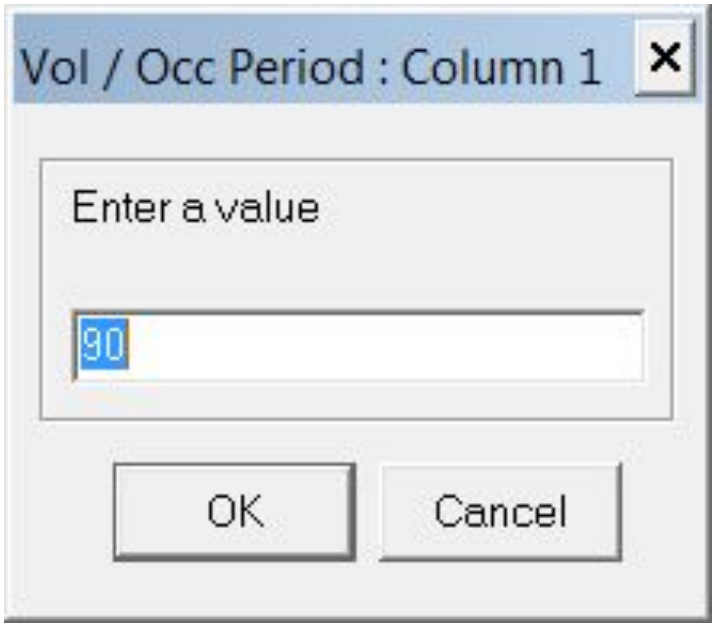
- How does it do this?
  - Typically, It does the adjustments by sampling the Volume (counts) and Occupancy (presence) of field detectors to determine Queue length and density

Data Objects for Adaptive control

- In the NTCIP 1202 standard, there are 'Objects' that provide this data to the controller and/or central system's algorithms to run adaptive control…

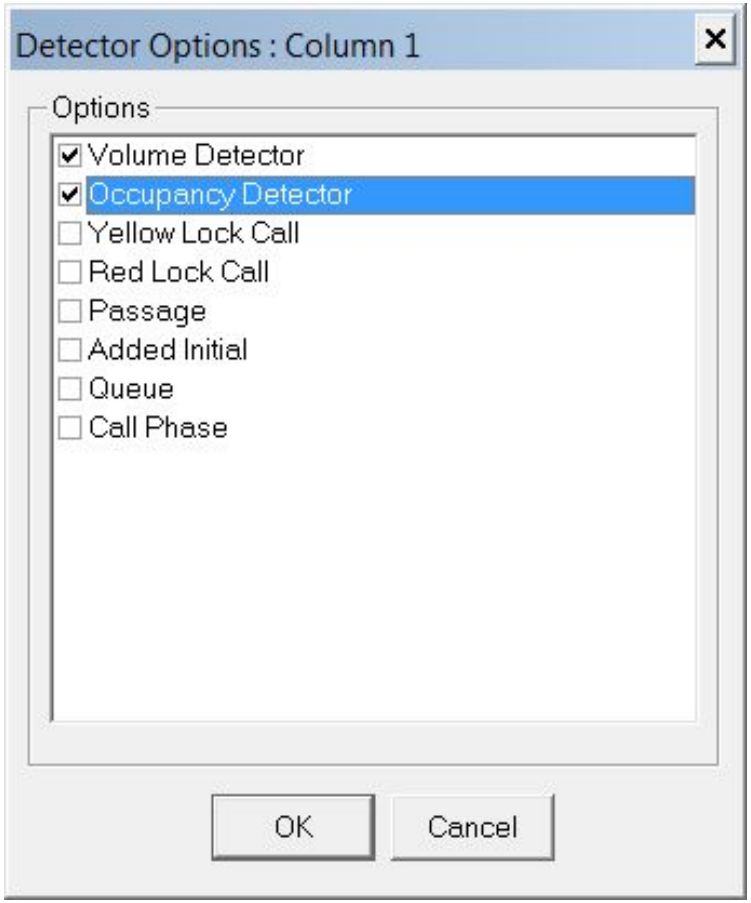
Data Objects for Adaptive control

- The Volume/Occupancy collection period is typically set to the Cycle length (in seconds)
- This method provides a way of 'framing' the traffic data on a cycle-by-cycle basis



Data Objects for Adaptive control

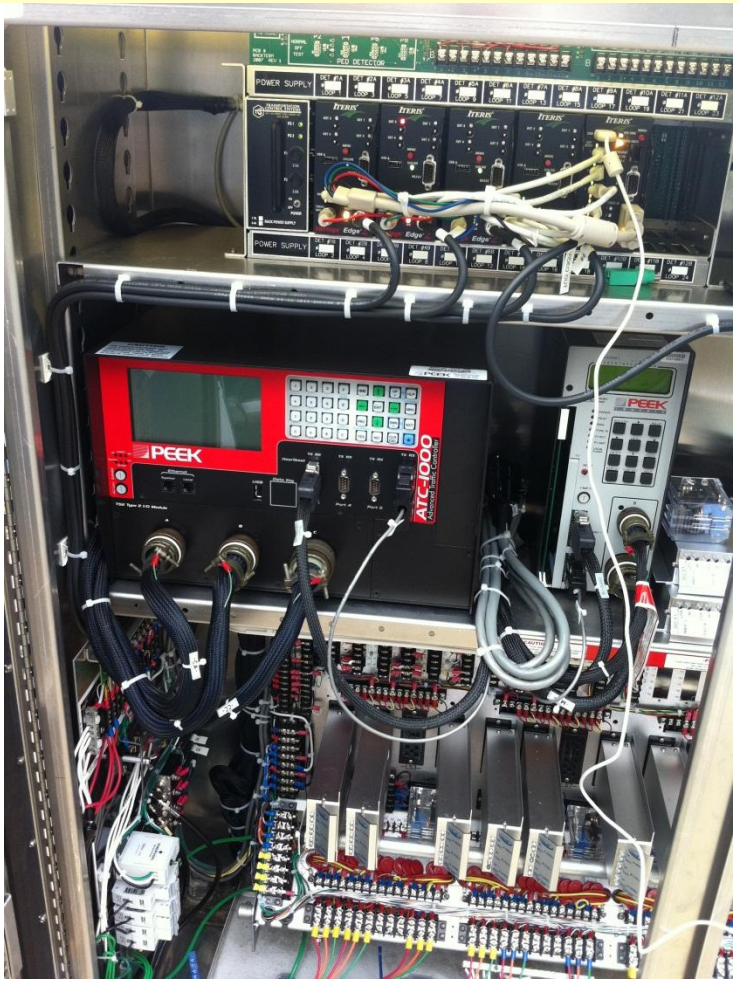
- In the detector options, there are flags for Volume and Occupancy
- This allows various adaptive algorithms to operate on Volume and/or Occupancy



How often are the changes made?

- How often should adjustments be made?
  - Highly debated, the general consensus is at a Minimum, once per cycle

Peek ATC controller installed @ intersection of PR 137/PR 155



Examples of Adaptive Control in the Peek controllers

- **ACSLite**: Supported by Peek, Econolite, Siemens and McCain – concept was created and the pilot project was run by the **FHWA**
- **OPAC**: Supported by Peek, Econolite, Siemens and D4 (2070 software company using Peek's 1C module) – developed by **Telvent**
- **CIC**: Supported by Peek – developed by **KLD Associates**

ASCLite (Adaptive Control Software)

- Utilizes an 'on-street' Master to adapt the offsets and splits in a conventional Closed Loop System, cycle times are still managed via TOD control
- Modifications to the offsets and splits take place every 5-15 minutes
- The ACSLite Master polls the intersections for status and detector information every minute
- Typical split adjustments are small (2-5 seconds), this allows the transitions to be made and accommodated quickly and efficiently

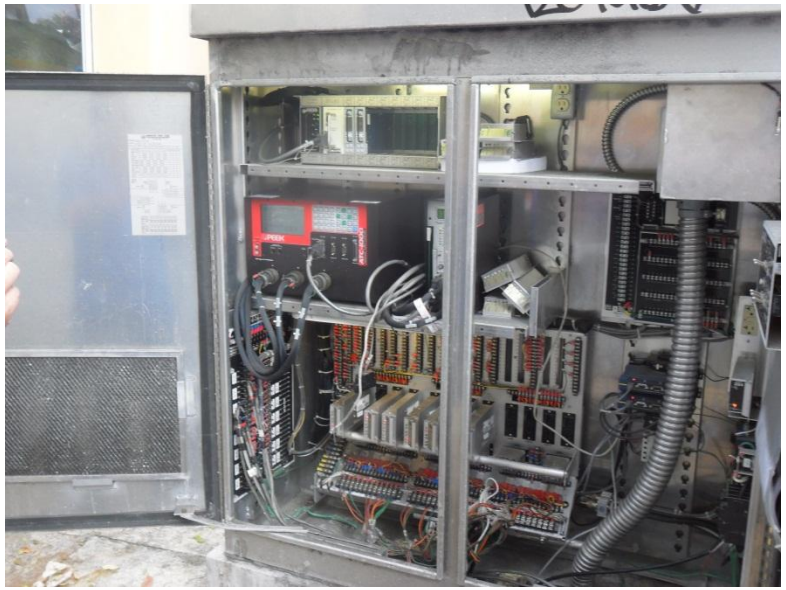
OPAC - (Optimized Policies for Adaptive Control)

- OPAC operates by sending the controller to Free, and then issues second by second force off and holds via NTCIP objects over an IP connection. Serial comm works as well, but IP is preferred
- It optimizes cycles, splits, and offsets. Cycles are recalculated every 5 minutes. Splits are adjusted every 4-5 seconds. Offsets are adjusted every cycle
- Upstream detectors (Using NTCIP/custom objects) provide count and Estimated time of arrival information

CIC (Critical Intersection Control)

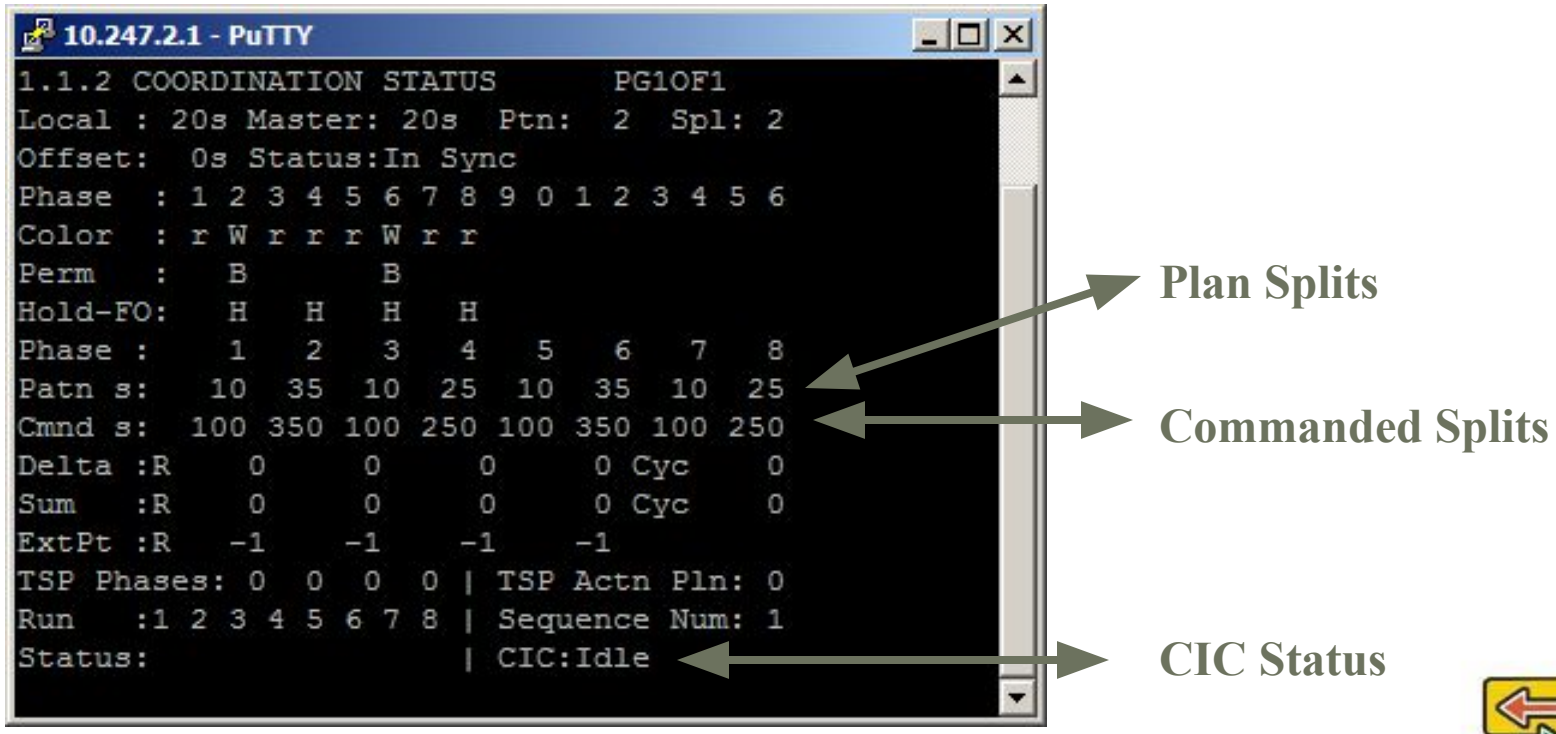
- CIC – works by allowing a central process access to the controller's Pattern (Cycle Length, Offset, Split values) – Standard NTCIP objects allow for collection of Volume/Occupancy
- CIC operates by setting the desired values (C/O/S) of the current running pattern and then setting the CIC Status object to 'Pending'
- At the next Local 0, the plan is evaluated and implemented - the controller sets the CIC Status object to 'Success'
- Deployed in New York City in Midtown Manhattan and in Staten Island

Typical operation of CIC in the ATC1000



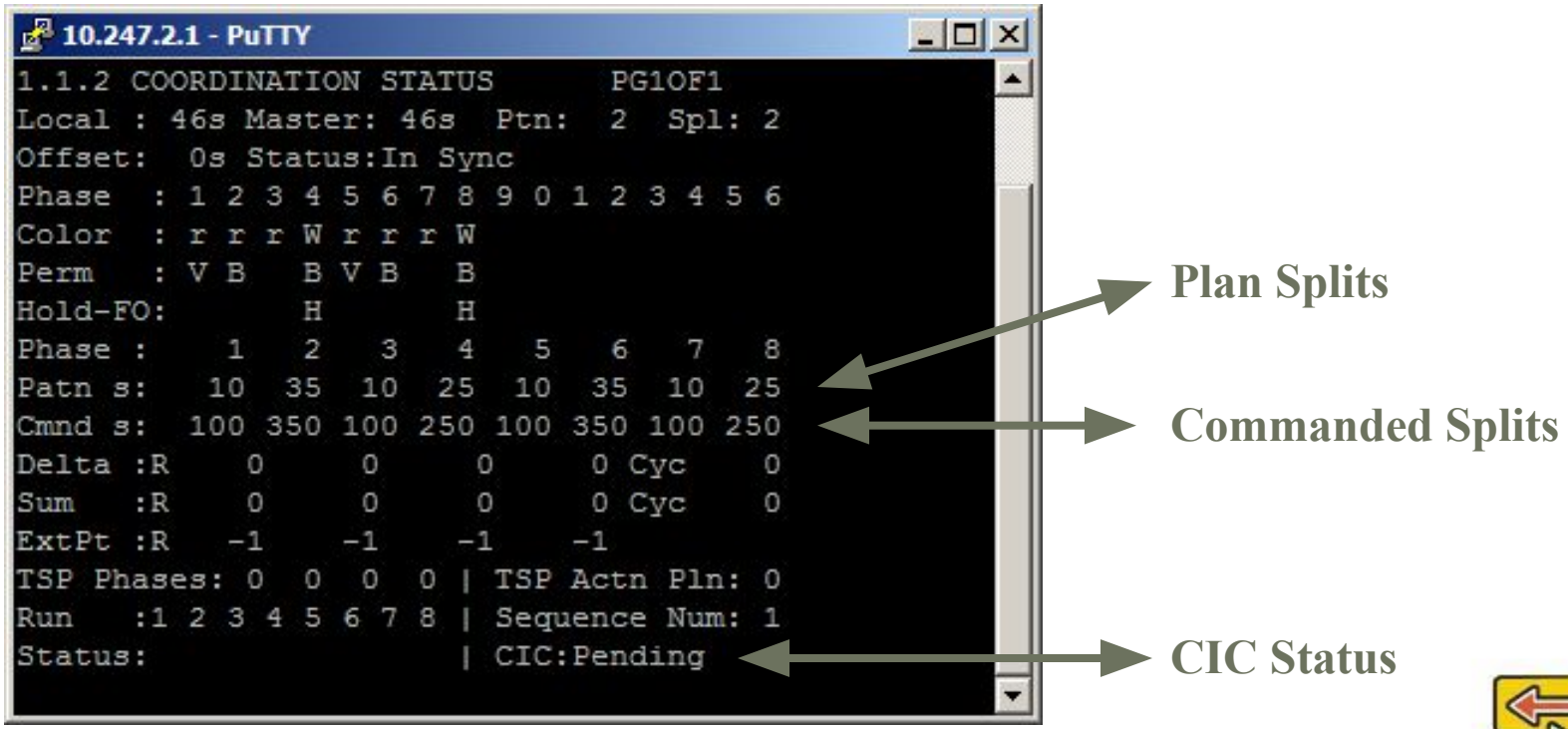
Data Collection, status ‘Idle’

- Step one; the intersection collects Volume/Occupancy data from selected detectors
- Controller is running normal plan splits…



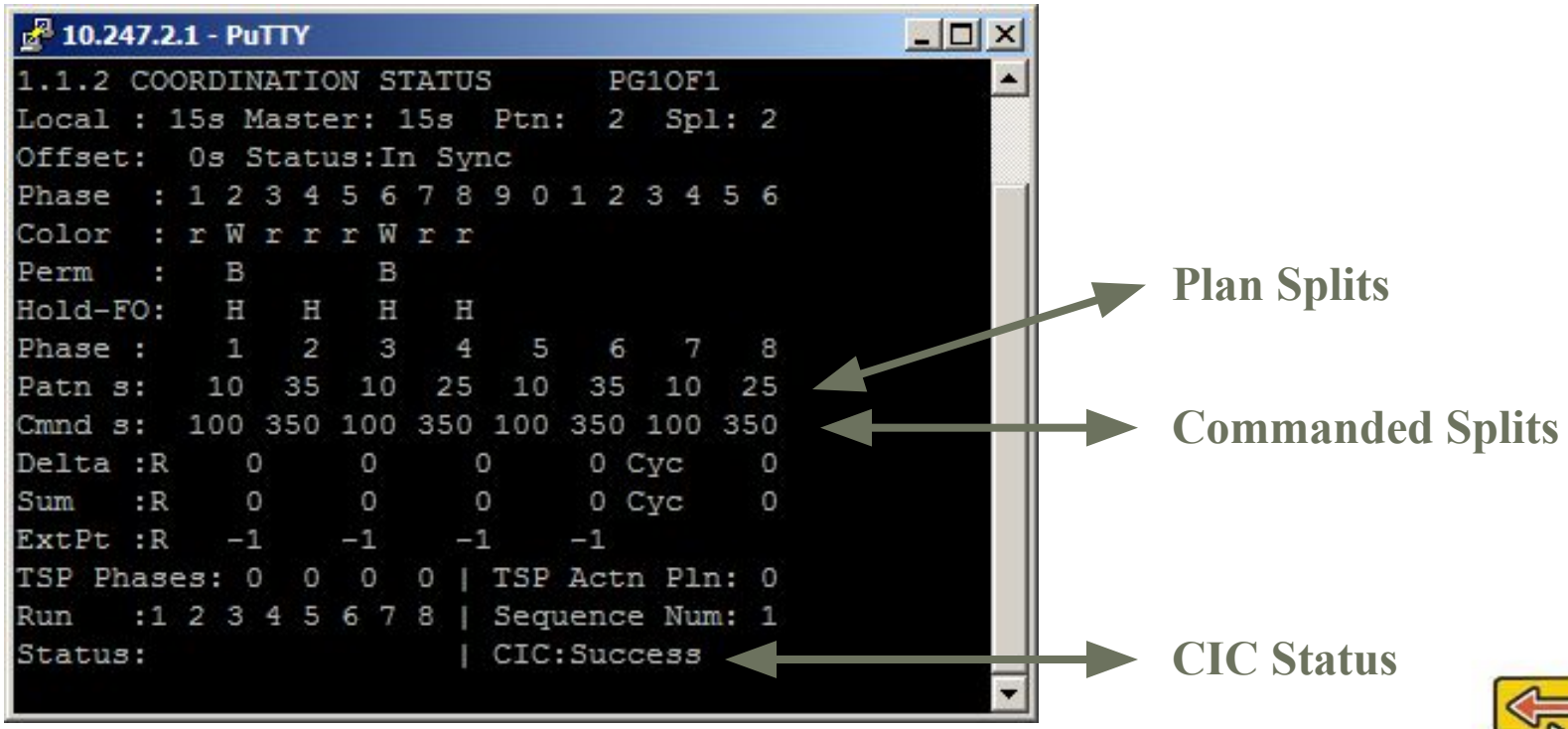
CIC object sets, status ‘Pending’

- Step two; the CIC engine sends the objects for the optimized C/O/S in the current pattern and sets the CIC status to ‘Pending’



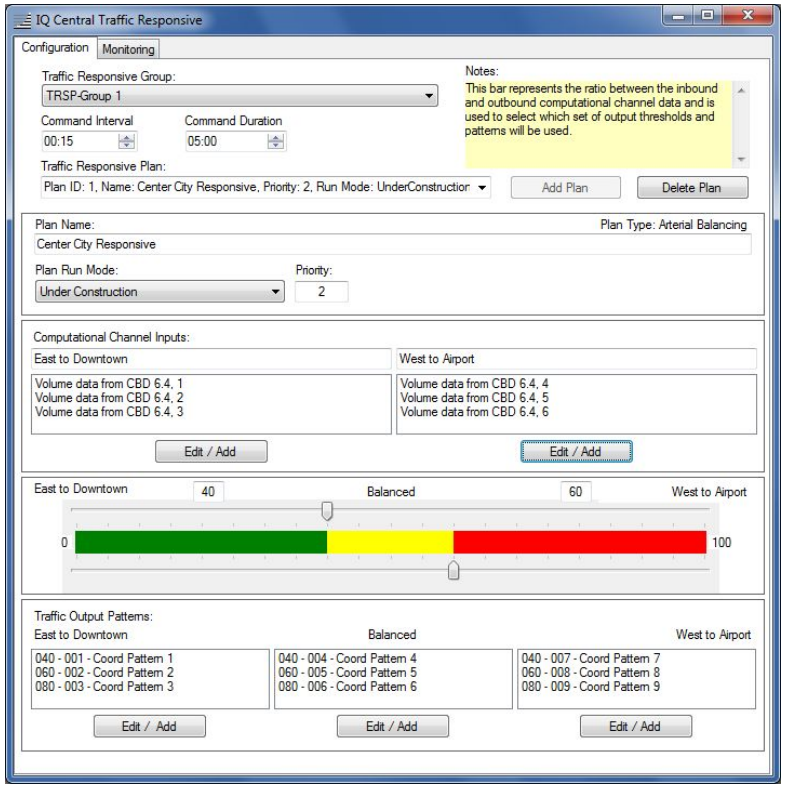
CIC implemented, status ‘Success’

- Step three; At the top of the next cycle, the CIC pattern data is validated and implemented. The controller sets the CIC status to ‘Success’



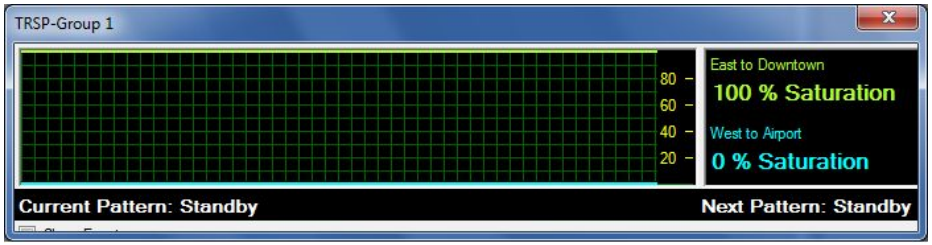
IQCentral Pattern control

- Operates on a group of intersections
- Choose the sensors to be sampled
- Determine the period between pattern updates
- Provides predictable responses to unpredictable conditions



IQCentral Pattern control

- Designed so the user decides when to go live
- 3 Modes of operation
  - ‘Under Construction’
  - ‘Testing Mode’
  - ‘Live’



Pattern control – Under Construction

- This mode is the default. The pattern control server will not load or run any channels in this mode. This is intended for times when you are first programming and setting up channels, or making changes to an existing channel.

Pattern control –Testing

- In this mode, the pattern control server will load and run the channel, but the pattern commands are not sent to the street. This tool allows the user to evaluate the effectiveness of the operation before going live.

Pattern control – Live

- This is the full run time mode. All pattern decisions will be sent to the traffic controllers in the street.

Questions/Comments?

donald.maas@peektraffic.com

- Thank you for your attention!