Distributed Schemes for Integrated Arrival Departure and Surface Scheduling

Second Year Final Report

Authors:

Husni Idris (Engility Corporation)

Ni Shen (Engility Corporation)

Aditya Saraf (ATAC Corporation)

Jason Bertino (ATAC Corporation)

Natasha Luch (ATAC Corporation)

NASA Technical Monitor:

Shannon Zelinski

October 31, 2016

Prepared for:

NASA Ames Research Center
Moffett Field, CA 94035-1000

Prepared By:

ENGILITY

900 Technology Park Drive, Suite 201
Billerica, MA 01821

ATAC

2770 De La Cruz Boulevard
Santa Clara, CA 94050

This page intentionally left blank

Preface

This document was prepared by Engility Corporation, 900 Technology Park Drive, Suite 201, Billerica, MA, under NASA Research Announcement (NRA) Contract Number NNA14AB46C. The report represents the deliverable "Cumulative Second Year Report, Prototype, and Analysis of Concepts Applied to at All Three Real World Problems with One High Fidelity Fast Time Simulation and Analysis" for the NRA titled "Distributed Schemes for Integrating Arrival Departure and Surface Scheduling".

This page intentionally left blank

Table of Contents

List of Figures

1 Introduction

Delays originate mostly at major airports, and particularly at ones that constitute complex metroplexes of multiple interacting airports. For example, in a prior queuing model-based analysis, the highest average delay per flight was measured at LaGuardia due to excessive departure queuing while the highest total delays were recorded at Atlanta Hartsfield (ATL) due to the high volume of traffic [1]. Major causes of delay at these metroplex systems are constrained resources on the airport surface particularly the runways and in the surrounding terminal airspace. These complex systems are characterized by high levels of interaction between flows of multiple adjacent airports sharing arrival and departure fixes and competing for gaps in the overhead traffic streams.

Solutions to mitigate these choke points have consisted mostly of isolated concepts and capabilities that are applied to components of the system. For example, concepts focused on either arrival flow management for arrival metering, sequencing and spacing [2, 3] or on departure flow management such as departure metering at gates [4-6], precision departure release control [7, 8], or runway scheduling [9, 10]. Integrated solutions are needed in order to reap the benefits of the isolated capabilities [11]. Research has been conducted to demonstrate integrating arrival operations from the en route phase of flight to the landing [12]. Initial research into integrating runway arrival and departure operations has also been conducted generating preliminary concepts and algorithms [13-15].

In this report, we describe a research activity that supports a NASA Advanced Technology Demonstration (ATD-2) project aimed at integrating departure and arrival operations in a metroplex, with an emphasis on integrating departure operations within arrival constraints [16-17]. The operations are integrated from the release from the gate on the airport surface, to the takeoff from the runway, to the crossing of departure fixes and the merging into the overhead traffic stream. Direction and from NASA was obtained to guide the focus and scope of these activities. This direction is summarized as follows:

1. In terms of airport/metroplex sites, focus on operations in Charlotte International Airport (CLT) and the interaction of its flows with the Atlanta International Airport (ATL) overhead streams.
2. For scheduling algorithm development:
    a. Investigate queue management strategies to increase the robustness of the scheduling to uncertainties while absorbing delays at the gate to reduce fuel consumption and environmental impacts.
    b. Investigate interaction between strategic (long term queue management) scheduling and tactical (near term runway, taxiway, and gate release) scheduling.
    c. Incorporate arrivals schedules into the integrated departure scheduling as constraints at the runway and the gates.
    d. Adopt heuristic rather than optimization approach to the derivation of schedules.
3. Availability of the use of NASA's Surface Operations Scheduler and Simulator (SOSS) as the high-fidelity simulation environment along with scenarios developed by NASA for CLT airport.
4. Build and validate a SOSS model for ATL.

Based on this direction from NASA, the following four activities were performed:

1. A MATLAB simulation-based analysis of the departure metering process was performed to investigate strategic queue management strategies and their robustness to uncertainty. Three metering strategies that are being considered for the ATD-2 strategic scheduling applied at CLT were compared. They respectively attempt to control the number of flights that (1) left the gate but did not take off, (2) left the ramp but did not take off, and (3) spent their unimpeded transit time to the runway but did not take off. Queue/delay buffer sizes were identified using historical data analysis that identified the buffer sizes that resulted in throughput saturation historically. Statistical distributions derived from historical data were used to model simulation parameters such as the unimpeded transit times between airport resources and the separation between runway operations. Arrivals schedules were modeled as constraints at the runway and gates. We drew insights on the performance of these strategies in terms of conforming to the runway schedule while transferring delay to the gates and ramp area. We assessed the performance of dynamic metering under different scenarios of uncertainty about the demand, the flight transit time and the runway schedule. It was observed that under deterministic and demand uncertainty conditions, the first strategy performed better than the other two strategies in terms of maintaining the runway throughput while transferring a significant average delay of two minutes to the gate. On the other hand, under uncertainties of flight transit time and runway service rate, all the strategies struggled to delay flights at the gate without a significant impact on the runway throughput.

2. A high-fidelity simulation-based analysis of the strategic departure metering process was performed using SOSS. In particular, we assessed the impact of delaying departures at their gates on blocking the arrivals destined for the same gates. We applied metering strategies that avoid this blocking effect by scheduling the release of a metered departure at certain time buffer prior to the next estimated arrival time at the gate. We used the same runway and metering algorithm developed in MATLAB under the first activity and interfaced the scheduling algorithm with the SOSS platform for simulation. The metering strategy used was the one that performed the best in the first activity, namely controlling the number of flights that pushed back but did not take off. We drew insights on the performance of metering with different gate blocking limit strategies in terms of conforming to the runway schedule and minimizing the blocking of arrivals while transferring delay to the gates. We determined the time buffers needed to eliminate blocking delays on arrivals as a result of departure metering and assessed the amount of departure metering that is feasible under such conditions. We observed that with a queue buffer size of fifteen flights for runway 18C and twelve flights for runway 18L, and with running the scheduler every five minutes over a one hour horizon, scheduling the release of departures three minutes prior to the estimated arrival at the same gate is sufficient to eliminate any additional delay on arrivals due to gate blocking.

3. Integrated departure scheduling algorithms were developed in the MATLAB prototype environment, including: Integration of departure operations between the gate, runways, departure fixes and en route merge points; integration of departures from major and secondary airports sharing departure fixes, and integration of the impacts of downstream restrictions such as Miles in Trail (MIT) and approval requests for insertion into the overhead stream (APREQ). In current day operations, ARTCC Traffic Management Unit (TMU) usually applies excess MIT restrictions on individual airport departure flows in order to provide the TRACON or ARTCC controllers with some leeway for merging these flights when they get closer to the departure fix or to the enroute merge fix. In future operations with ATD-2 departure metering, departure flows will become more predictable, departure sequences will become more optimal (occurrences of consecutive departures going to the same departure fix will be reduced) and takeoff times will be better coordinated with departure-fix merge time-slots as well as overhead enroute traffic stream gaps. As a result, the requirement for

additional MIT spacing at runway takeoff may be relaxed. This analysis aimed to evaluate whether MIT constraints can be relaxed when ATD-2 is in operation without impairing the safety of operations and to quantify the additional benefit that can be provided by relaxing MIT constraints in this situation. We analyzed historical CLT, CLT-satellite and ATL departure tracks to determine the major departure flows and the major constraint points where the CLT departures merge with other traffic during their transit from runway takeoff to enroute stream merge. The simulation included modeling of (i) departure and arrival operations on the CLT airport surface, (ii) departure operations in the CLT TRACON (including departures taking off from smaller satellite airports within the TRACON), (iii) CLT/CLT satellite departures merging into overhead enroute traffic streams in the Atlanta ARTCC (ZTL) airspace, and (iv) ATL airport departures taking off from ATL and merging with CLT/CLT satellite airport departures in ARTCC airspace. The simulation focused primarily on departure traffic headed to destination airports in the Northeast US. Simulation-based analysis demonstrated that maintaining MIT restrictions at current-day levels while ATD-2 is in operation, may impede the full realization of the benefits from ATD-2. Our results also demonstrated that relaxing MITs when ATD-2 scheduling is active would save around 1-3% total departure delay (gate + taxi + airborne delay) while retaining a high level of taxi and airborne delay savings over current-day operations, as well as maintaining a level of safety commensurate with current-day operations.

4. Models of ATL were added to NASA's SOSS simulation environment. Calibration of the ATL model with PDARS/ASDE-X data utilized as truth data was also performed. The activities included: (i) Data analysis for selecting the airport runway configuration(s), (ii) generating the required airport adaptation data, (iii) generating traffic scenarios, and (iv) running the simulation and validating simulation results against real operational data.

This report is organized as follow: First the ATD-2 concept is described briefly. Then four successive chapters describe the four activities described above. Each chapter is written as a standalone report which can be read without the need to refer to the previous chapters. Hence, some of the background needed for the chapter may be repeated from other chapters. Each chapter also draws its own conclusions and insights from the analysis.

2 ATD-2 Concept

The ATD-2 project aims to improve predictability and operational efficiency of air traffic in metroplex environments by enhancing existing and developing new arrival, departure and surface prediction, scheduling and collaborative decision making systems and integrating them in a single, state-of-the-art traffic management system. The eventual objective is to demonstrate this state-of-the-art traffic management system via human-in-the-loop (HITL) simulations and/or field evaluations, and transfer the component technologies to the FAA.

The operational environment for ATD-2 (also known as Integrated Arrival/Departure/Surface Operations, i.e. IADS) metroplex traffic management concept includes a primary TRACON, consisting of a major, well-equipped airport and multiple satellite airports that are less-equipped [10, 16, 17]. A well-equipped airport will typically have sophisticated automation aids such as surface traffic surveillance in the FAA towers as well as in ramp towers, and would be subject to heavy traffic demand including flights from multiple major airlines. The less-equipped airports will typically not have surface surveillance and are subject to smaller demands with smaller percentage of commercial air traffic from the major air carriers. See Figure 1.

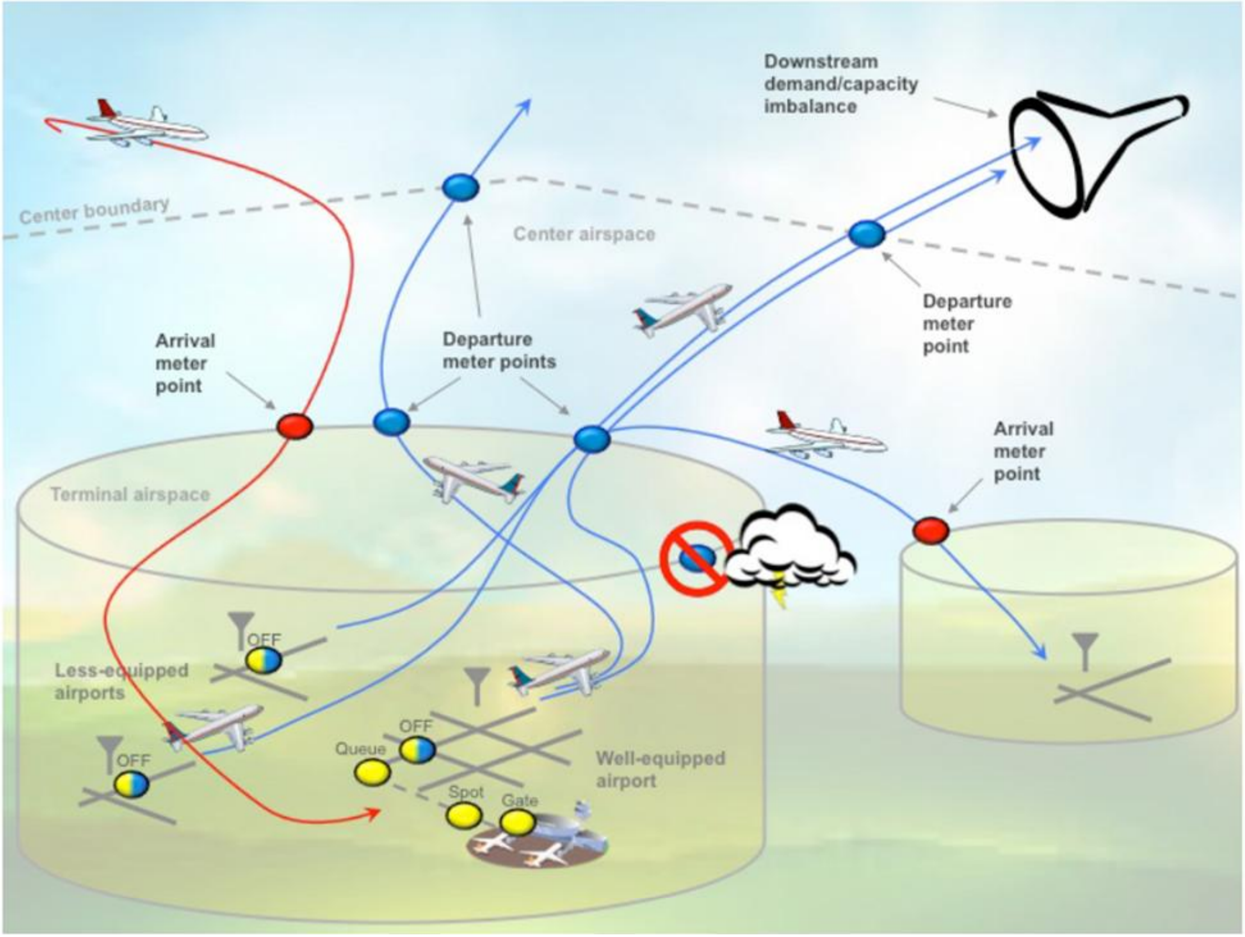


Figure 2-1. ATD-2 Concept (*Figure taken from NASA ATD-2 team presentation)*

Within this operational environment, ATD-2 traffic management tools' focus is on improving the coordination between metroplex-wide departures to enable efficient merging and metering of departure flows at the key exit-points of the TRACON (departure-fixes) and at merge points into overhead en route traffic streams. In addition, the tools will also enhance adherence to metered departure times from the primary TRACON airports, where the metered departure times are provided by a time-based metering system such as Traffic Management Advisor (TMA) at destination airports outside the TRACON (maybe even multiple centers away from the TRACON). The control points for ATD-2 traffic management tools may include gate pushback (by providing Target Off Block Times, TOBTs, to airline ramp controllers), movement area entry (by providing Target Movement Area Entry Times, TMATs, to the Ground Controller(s)) and runway takeoff (by providing Target Takeoff Times, TTOT, to the Local Controller) at the well-equipped airport and runway takeoff at the less-equipped airports (by providing TTOTs to the Local Controller).

In addition to NASA's research into new IADS traffic management tools, the FAA has also developed a Surface Collaborative Decision Making (CDM) concept [18] that will enable U.S. airports to make optimal use of available airport capacity. NASA plans to develop, test and deploy the ATD-2 tools while adhering to the concept of operations outlined by the FAA. This concept addresses the need for timely sharing of relevant operational data among Surface CDM Stakeholders to improve situational awareness and predictability through a common understanding of "real" airport demand and continuous predictions of demand/capacity imbalances. At the core of this concept is a set of well-defined capabilities and procedures which facilitate the proactive management of airport surface traffic flows and runway departure queues to equitably optimize local airport capacity and shared NAS resources.

One of the key capabilities included in the FAA's Surface CDM Conops is the efficient strategic management of departure queues and flows on the airport surface. This capability leverages improved situational awareness via data exchange (which is another key capability included in the Surface CDM concept) for accurate prediction of demand and capacity imbalances, notification of predicted imbalances to stakeholders, and implementation of Departure Metering Procedures or Programs (DMPs) to equitably allocate constrained NAS resources among stakeholders. DMPs include a specific set of functions, such as assignment of Target Movement Area entry Times (TMATs) and all associated processes and procedures. Conceptually, a DMP is very similar to a Ground Delay Program (GDP), which is currently implemented by the FAA's System Command Control Center (ATCSCC) in order to manage arrival traffic flows into constrained airports. The objective of a GDP is to absorb as much delay as possible on the surface at the origin airport rather than in the air. Similarly, the objective of the DMP is to absorb as much delay as possible at the gates (or at a holding location in the ramp or movement area) rather than in a departure taxi queue because it is more fuel efficient (since the engines are off) and convenient (because passengers can wait in the airport terminal area rather than inside an aircraft). NASA envisions that the ATD-2 decision support tool will be utilized to implement a DMP.

3 Comparison of Different Control Schemes for Strategic Departure Metering

Airports and their terminal airspaces are key choke points in the air transportation system causing major delays and adding to pollution. A solution aimed at mitigating these chokepoints integrates the scheduling of runway operations, flight release from the gates and ramp into the airport movement area, and merging with other traffic competing for downstream airspace points. Within this integrated concept, we present a simulation-based analysis of the departure metering process, which delays the release of flights into the airport movement area while balancing two competing objectives: (1) maintaining large enough queues at the airport resources to maximize throughput and (2) absorbing excess delays at the gates or in ramp areas to save on fuel consumption, emissions, noise, and passenger discomfort. Three metering strategies are compared which respectively attempt to control the number of flights that (1) left the gate but did not take off, (2) left the ramp but did not take off, and (3) spent their unimpeded transit time to the runway but did not take off. It was observed that under deterministic and demand uncertainty conditions, the first strategy performed better than the other two strategies in terms of maintaining the runway throughput while transferring a significant average delay of two minutes to the gate. On the other hand, under uncertainties of flight transit time and runway service rate, all the strategies struggled to delay flights at the gate without a significant impact on the runway throughput.

3.1 Introduction

Delays originate mostly at major airports, and particularly at ones that constitute complex metroplexes of multiple interacting airports [1]. Major causes of delay at these metroplex systems are constrained resources on the airport surface, particularly the runways and in the surrounding terminal airspace. These complex systems are characterized by high levels of interaction between flows of multiple adjacent airports sharing arrival and departure fixes and competing for gaps in the overhead traffic streams. Solutions to mitigate these choke points have consisted mostly of isolated concepts and capabilities that are applied to components of the system. For example, concepts have focused on arrival flow management for arrival metering, sequencing and spacing [2, 3], on departure flow management such as departure metering at gates [4-6] and precision departure release control [7, 8], or on runway scheduling [9, 10]. Integrated solutions are needed in order to reap the benefits of the isolated capabilities [11]. Research has been conducted to demonstrate integrating arrival operations from the en route phase of flight to the landing [12]. Initial research into integrating runway arrival and departure operations has also been conducted, generating preliminary concepts and algorithms [13-15]. In this chapter, we describe a research activity that supports a NASA Advanced Technology Demonstration (ATD-2) project aimed at integrating departure and arrival operations in a metroplex, with an emphasis on integrating departure operations within arrival constraints [16-17].

The integration of airspace and surface operations in a metroplex system aims to maintain high throughput of the system, efficient operations through expedited, uninterrupted movement, and minimum fuel burn and emission. This integration attempts to achieve these objectives through in part the generation of a coordinated schedule of departures at key resources or control points. These control points include primarily the release from the gates, the exit from the ramp area into the airport movement area, the takeoff from the runways, and the crossing of departure fixes, as depicted in Figure 3-1. It may be possible to simultaneously achieve these objectives under deterministic conditions. However, uncertainty in the operations and the environment brings about tradeoffs among them. For example:

1. In order to maintain high throughput under uncertainty, delay or queue buffers are needed to keep pressure on the choked resources of the system and take advantage of any service opportunities that may arise. Parallel queues provide controllers the ability to sequence departures optimally and the controllability needed for conformance to prescribed constraints such as flow management restrictions. If the schedule assumed the fastest transit between resources with zero delay or queue buffers it would be violated due to any disturbance that results in longer transit time. This requires absorbing some delay near downstream resources, such as in the airspace near departure fixes and on the airport surface near the runways, as shown in Figure 3-1.

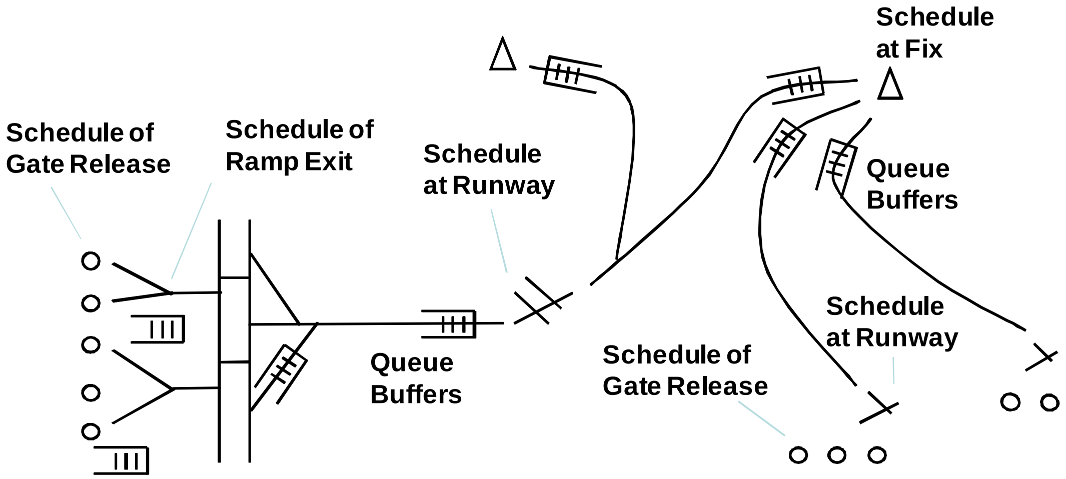


Figure 3-1. ATD Integrated scheduling with target queue buffers

2. On the other hand, delay is absorbed more efficiently and cleanly on the airport surface (movement area) rather than in the airspace, and in turn, at the gate while the engines are off rather than on the airport surface. Absorbing delay at the gates or in the ramp area saves on fuel consumption, emissions and noise in addition to allowing passengers to absorb some of the delay more comfortably while off the plane.

Therefore, the integrated scheduler needs to decide on the distribution and allocation of delay between the interconnected resources. NASA's ATD-2 concept includes a strategic surface scheduler which schedules the release of flights from the gates and/or the ramp when the demand for the runways is predicted to exceed their capacity in order to balance the tradeoff between throughput and efficiency. Key parameters used in this decision are the desired queue or delay buffer sizes that the scheduler should target to achieve an appropriate balance between these objectives.

We present a simulation-based analysis of the departure metering process. Three metering strategies that are being considered for the ATD-2 strategic scheduling applied at Charlotte International Airport (CLT) are compared. They respectively attempt to control the number of flights that (1) left the gate but did not take off, (2) left the ramp but did not take off, and (3) spent their unimpeded transit time to the runway but did not take off. We draw insights on the performance of these strategies in terms of conforming to the runway schedule while transferring delay to the gates and ramp area. We assess the performance of dynamic metering under different scenarios of uncertainty about the demand, the flight transit time and the runway schedule.

In the following sections we describe the ATD-2 operational concept and assumptions made for this analysis. Then, we describe the fast-time simulation model that was developed and used for this analysis, along with the underlying algorithms and statistical models. The analysis results are then reported with insights on the performance of the different metering strategies under different uncertainty scenarios, ending with concluding remarks and future extensions.

3.2 ATD-2 Concept and Assumptions

The ATD-2 concept includes several attributes that achieve the integration between the scheduling of operations at different resources. Two of these attributes are:

1. Integration between upstream and downstream schedules. Namely, the schedule of operations at the runway considers constraints stemming from scheduling flows at downstream airspace shared points such as TRACON departure gates and gaps in the overhead en route stream. In turn, the schedule of the release from the gates or ramp area into the airport movement area takes into consideration the restrictions stemming from the schedules at the runways and at downstream airspace merge points.

2. The concept integrates a strategic metering scheduler with a tactical control scheduler. The strategic scheduler implements the metering process; it runs at low frequency and attempts to control the congestion at reasonable levels when demand exceeds capacity. The tactical scheduler runs more frequently and controls the conformance of the flights to the integrated schedules at the different resources such as the runway and the release from the ramp and gates into the airport movement area.

The interactions between the upstream and downstream schedulers and between the strategic and tactical schedulers are a topic of continued research and design. For this analysis, we made some assumptions about these interactions, as described in the following paragraphs of this section.

The strategic scheduler implements a metering process by generating a desired schedule of releases from the gates and ramp area such that congestion is reduced and a desired level of queuing buffers is maintained. The queuing may be measured and controlled using different parameters, three of which are compared in this analysis: (1) the number of flights that left the gate but did not take off (2) the number of flights that left the ramp area but did not take off and (3) the number of flights that have spent their unimpeded transit time since they left the gate or the ramp to the runway but did not take off. The three queue buffer parameters are shown in Figure 3-2. These three queuing parameters correspond to the number of flights in three queuing systems of different sizes. The exit from all three systems is the takeoff event. The entry to the first system is the pushback, to the second system is the exit from the ramp into the movement area, and to the third system is when the flight’s unimpeded transit time since its pushback or its ramp exit has elapsed. The first two systems are physical while the third system is virtual because the elapsed unimpeded transit time does not correspond to a physical resource or location in the airport. The first system is the largest containing more flights, followed by the second system. Finally the third system is the smallest and attempts to approximate the number of flights that are closer to the runway end.

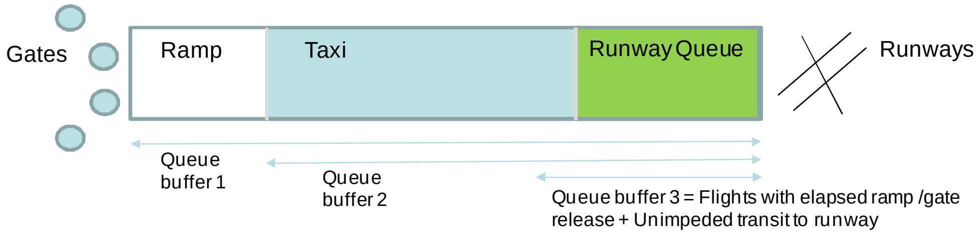

Gates
Ramp
Taxi
Runway Queue
Runways
Queue buffer 1
Queue buffer 2
Queue buffer 3 = Flights with elapsed ramp /gate release + Unimpeded transit to runway

Figure 3-2. Queue buffer definitions

The strategic scheduler requires a runway schedule in order to estimate the level of congestion and queuing based on the demand. Rather than generating its own runway schedule, it is assumed in this analysis that the strategic scheduler takes the most recent runway schedule generated by the tactical scheduler as an input. Therefore, the tactical scheduler was given the same time horizon as the strategic scheduler in this analysis; otherwise the strategic scheduler would need to extend the tactical schedule to a longer horizon. The tactical schedule ensures the separation requirements between successive runway operations and integrates restrictions stemming from downstream schedules at departure fixes and en route overhead merge points, and possibly at arrival metering points and destination airports. In this analysis, the model used for the runway scheduler (described in the next section) applies the separation requirements; however, it does not explicitly consider the other restrictions. A prior paper described an algorithm for integration of downstream schedules into the runway schedule [28].

The strategic scheduler generates desired release times from the gates to control the queuing parameter while ensuring that the input runway schedule remains feasible. The flight release time from the gate ensures that the flight can transit unimpeded to the runway in time to make its assigned runway time, even if the queuing parameter exceeded the desired threshold. In this case the queuing parameter may exceed the desired level in favor of enabling the runway schedule to be feasible. Alternative schemes that are not analyzed in this report include the strategic scheduler giving priority to maintaining the desired queuing buffers over conformance to the runway schedule. In this case the strategic scheduler may recommend gate release times that maintain the queuing parameter at or below the desired level but require changes to the input runway schedule. The strategic scheduler can suggest the changes in the runway schedule that correspond to the recommended gate release times. However, ultimately the tactical scheduler determines the final runway schedule and the corresponding gate release time reconciling the strategic scheduler recommendations with other constraints and objectives.

The tactical scheduler has the task of ensuring the conformance of flights to the runway schedule and to the releases from the gate and ramp into the movement area recommended by the strategic scheduler. It has to make a decision if the two schedules cannot be met simultaneously. In actual operations, this decision should be mostly resolved in favor of meeting the runway schedule to maintain the throughput of the runway bottleneck at the expense of exceeding the desired queuing buffers. However, in this analysis we assume that the tactical scheduler adheres to the desired release times recommended from the strategic scheduler in order to measure the performance of the strategic scheduler in terms of enabling the conformance to the runway schedule. Otherwise the tactical scheduler would override the release times computed by the strategic scheduler.

The strategic scheduler takes into account capacity constraints at the gates by ensuring that the departures are not delayed excessively such that arriving flights requiring the gate incur delay. The strategic scheduler may not have access to full information about the airline gate capacities and procedures hence this constraint needs to be approximated by the strategic scheduler. In this analysis we assumed that the available gates are pooled into one set of gates that are used by all users. The model is capable of limiting the gate capacity by users and aircraft types, but this capability was not exercised in this analysis.

The strategic scheduler takes as input the demand in terms of expected pushback times over the scheduling horizon. We assume that the strategic scheduler knows the actual push back times and the actual ramp exit times for the flights that have already pushed back or exited the ramp, respectively. For flights that have not pushed back yet, it assumes the published pushback time in the flight plan. For flights that have not exited the ramp yet, it assumes a scheduled ramp exit time estimated by adding an unimpeded transit time from the gate to the ramp to the flight plan pushback time. As a baseline for comparison, we assume that the pushback and exit times are known by the scheduler by using the actual pushback and ramp exit times instead of the pushback time published in the flight plan and its corresponding estimated ramp exit time.

3.3 Models and Algorithms

A fast time simulation was used for the analysis of the ATD-2 concept elements. The fast time simulation implements models for the strategic and tactical schedulers and the interactions between them according to the assumptions described in the previous section. The algorithms and underlying statistical models used for the three main components of this simulation are described in the following subsections.

3.3.1 Strategic Metering Scheduler

The strategic scheduler takes a runway schedule as an input and computes gate and ramp release times that maintain a queuing parameter at or below a desired value. Three queuing parameters are compared in this analysis: (1) the number of flights that left the gate but did not take off (2) the number of flights that left the ramp area but did not takeoff, and (3) the number of flights that have spent their unimpeded transit time since they left the gate or the ramp to the runway but did not takeoff. As described above, the three parameters correspond to three queuing systems, with different entry events and the same exit event (the takeoff). The entry to the first system is the pushback, to the second system is the ramp exit, and to the third system when the flight's unimpeded transit time to the runway elapses. Henceforth we call the time associated with the entry to any of the systems the entry time. We describe the flights that entered the third system (have spent their unimpeded transit time to the runway since they left the gate or the ramp) as ready for takeoff. This is an approximation as these flights may not be physically ready for takeoff if some of their unimpeded transit time was spent absorbing delay rather than transiting towards the runway. We also call the corresponding entry time to the third system (equal to the unimpeded transit time after leaving the gate or the ramp) the takeoff ready time. First the metering algorithm is described followed by the statistical model used to determine the desired queuing parameter threshold. The input runway schedule generated by the tactical scheduler is described in the next section.

3.3.1.1 Metering algorithm

Based on the input runway schedule, the metering algorithm computes three numbers for each flight within its scheduling horizon: the gate release time, the ramp exit time (also called taxi spot release time) and the takeoff ready time. In the first case, controlling the number of flights that pushed back but did not take off, the algorithm first computes the desired push back time and then derives the spot release and takeoff ready times by adding the unimpeded transit times of the flight. In the second case, controlling the number of flights that left the spot but did not takeoff, the algorithm first computes the desired spot release time and then derives the gate pushback time and the takeoff ready time by subtracting and adding (respectively) the unimpeded transit times of the flight. Finally, in the third case, controlling the number of flights that are ready for takeoff but did not takeoff, the algorithm first computes the takeoff ready time and then derives the other two times by subtracting the unimpeded transit times of the flight. The algorithm is the same for the three cases except for considering the entry time to the queuing system as the estimated pushback time, the estimated ramp exit time, or the estimated takeoff ready time, respectively. This time is referred to in the following description as the entry time. The metering algorithm first ranks the flights by the estimated pushback time as a representation of the first come first serve (FCFS) order and then performs the following steps for each flight in this FCFS order:

1. Estimate the queuing parameter value at the estimated entry time of the flight.

2. Determine the release time that satisfies the desired queuing parameter threshold: If the value estimated in step 1 is larger than the target parameter threshold (described in the following subsection), perform the following two steps:

    a. Identify the flights that have already entered the system (their entry time is earlier than or equal to the entry time of the flight being scheduled) but have not exited the system (their takeoff time is later than the entry time of the flight being scheduled).

    b. Set the desired entry time at the takeoff time of the flight with the earliest takeoff time that reduces the parameter to the target value.

3. Apply the runway schedule conformance constraint if turned on: If the algorithm is set to conform to the runway schedule, set the desired entry time to the minimum of two numbers: (1) the computed value in step 2 or, (2) the flight's schedule takeoff time minus its unimpeded transit time from its entry to the queuing system to the runway. This ensures that the flight is able to transit unimpeded to the runway in time to meet its takeoff schedule. If not set to conform to the runway schedule, the desired entry time from step 2 is unaltered. Note that for the third control strategy that uses the takeoff ready time as the entry time, the runway schedule conformance constraint is always on because the value (1) is never larger than the schedule takeoff time and the value (2) is exactly the schedule takeoff time. In this analysis, the runway schedule conformance constraint was always turned on to enable fair comparison between the strategies.

4. Apply the gate constraint if it is enforced: Find the arrivals that arrived at their gate before the flight's candidate gate release time computed in step 3. The gate arrival times used in the model are the actual parking times recorded in the ASPM database (IN times). Estimate the gate demand at each of these times as the sum of arrivals and departures that are, or would still be, at their gate. Set the desired entry time to the earliest value at which the gate demand drops below the threshold gate capacity value. The gate capacity was estimated using historical data analysis as the maximum number of gates that were occupied at the same time, which for CLT was determined to be 97 gates.

5. Apply a maximum position shift if enforced: a constraint is applied on the change between the sequence of the flight at the entry and its sequence in the runway schedule in terms of number of flights that the flight needs to pass between its entry and its takeoff. This constraint reflects

the limited ability to re-sequence flights between the entry point (gate, ramp, or runway end) and the takeoff. If the constraint is enforced, first the scheduled takeoff times of the flights that are in the system ahead of the flight are identified (as defined in step 2.a). Then the flight's entry time is set at the earliest of these takeoff times that keeps the required position shift below a threshold. In this analysis this constraint was relaxed by setting the maximum position shift threshold to infinity.

6. Once the desired entry time that satisfies all the active constraints is computed, the other two output times are computed by adding or subtracting the unimpeded transit times of the flight. The statistical models of the unimpeded transit times are described later in this section.

3.3.1.2 Saturation model

The strategic scheduler requires as a key input the target value of the queuing parameter for each of the three control strategies analyzed. This value was estimated using a throughput saturation analysis of historical data and varied in the analysis to determine the sensitivity of the scheduler performance to it. The throughput saturation models were described in [27]. The historically reported throughput is plotted against the historically reported number of flights in the queuing system as shown in Figure 3-3 for one CLT runway (18L) using one year of data (10-1-2011 to 9-30-2012). In this plot, the number of flights plotted on the horizontal axis is the first queuing parameter (the number of flights that left the gate but did not take off), which is used as an example to demonstrate the model. The model is applied to all the other queuing parameters in an identical manner but the corresponding plots are not shown. The runway throughput is plotted on the vertical axis and the number of flights in the queuing system, N(t), is plotted on the horizontal axis. The number of flights N plotted on the horizontal axis was measured at every minute t as the number of flights that were in the queuing system at that minute – this is the number of departures that have entered the system (their pushback time is less than or equal to t in this case) but have not exited from the system (their takeoff time is larger than t) by that minute. The entry time was set differently for each of the definitions of the queuing systems.

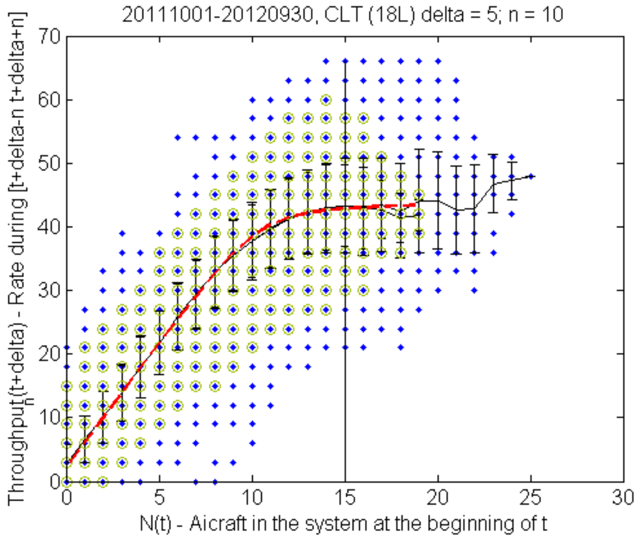


Figure 3-3. Throughput saturation analysis (CLT, Runway 18L departures that pushed back but did not take off)

The throughput plotted on the vertical axis was measured at every minute t as the number of flights that took off in a time window symmetrically centered on t. The time window was set to twenty minutes in this analysis (the variable n in the plot). The throughput measurement was plotted at a time offset (the variable delta in the plot) from the measurement of N. The offset was selected for each plot as zero, five, or ten minutes. The offset that resulted in the best correlation between N and the throughput was selected. In the example in Figure 3-3 the offset (delta) is shown in the title of the plot as five minutes. The throughput is reported per hour.

The average throughput is computed at each N value and connected with a solid line in the plot. Error bars show the variation in throughput at each N value. As shown in Figure 3-3, the average throughput increases and then saturates as N increases. In order to identify and quantify the saturation of the throughput at high N values, a hyperbolic curve was fitted to the average throughput versus N data (Dashed red line in Figure 3-3). The fitted hyperbolic curve has two asymptotes: one horizontal asymptote that tends towards a constant throughput value as N tends to infinity and a second asymptote that passes through the initial point at zero N. The resulting horizontal asymptote approximates the average throughput capacity of the runway. Before performing the curve fit, the least frequent N-throughput pairs were eliminated as outliers representing rare and off-nominal conditions. The removed pairs are the blue dots that are not encircled in Figure 3-3 and constituted in this case one percent of all pairs. The remaining pairs (green circles in the figure) were the only ones used in the curve fitting. As seen in the plot, the filtering excluded pairs of low throughput at high queue values. These pairs represent off-nominal conditions, such as airport closure or bad weather events, where high queues accumulated due to lack of throughput. The filtering also excluded pairs of high throughput at all N values. These pairs represent rare reports of high runway utilization.

In this analysis, the N value that is sufficient for throughput saturation was considered at the point where the slope of the hyperbolic fit curve is 0.005. In Figure 3-3 a vertical line is drawn at this point to indicate the value N on the horizontal axis that corresponds to throughput saturation. In this case fifteen flights between the gate and the runway were considered sufficient to saturate the runway throughput. To study the strategic scheduler sensitivity to this saturation threshold, 75% and 50% of the saturation value were analyzed as well. Figure 3-4 shows the corresponding throughput saturation levels for the number of flights that exited the ramp but did not take off (left) and the number of flights that spent their unimpeded transit to the runway but did not take off (right) for runway 18C.

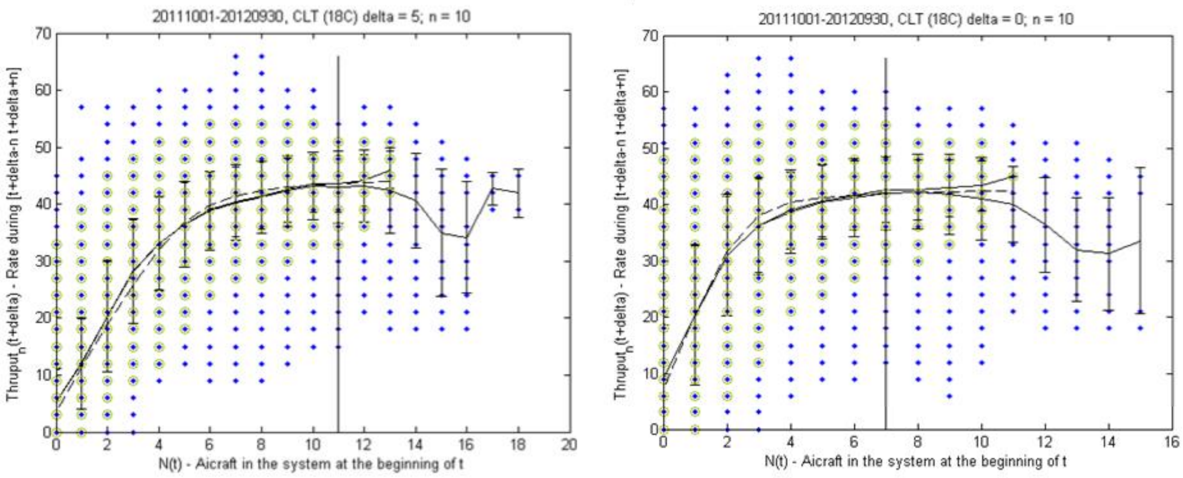


Figure 3-4. Throughput saturation analysis (CLT, Runway 18L departures that exited ramp but did not take off on left and that spent unimpeded transit to runway but did not take off on right)

3.3.2 Runway Scheduler

The tactical scheduler generates a runway operations schedule, which is used to simulate the runway operations. This schedule is assumed to be shared with the strategic scheduler described above, which uses the schedule as an input in the metering process. The tactical scheduler inserts departures and runway crossings in a FCFS order between a given arrival schedule. For example, departures and runway crossings on runway 18C at CLT are inserted in gaps between the arrivals on runway 18C and in gaps between the arrivals on runway 23, which converges on runway 18C. The scheduler does not change the given arrival schedule where arrivals are assumed to land at their actual landing times reported in the PDARS/ASDE-X data base. In addition to the landing times, the scheduler takes as input the takeoff and runway crossing ready times for each departure and crossing (respectively) in the schedule horizon, and the required time separation between successive operations. The algorithm is described first followed by the statistical models of the separation times and the unimpeded transit times used to generate the ready times.

3.3.2.1 Scheduling algorithm

The tactical scheduler algorithm first ranks the departures and runway crossings in the scheduling horizon according to their ready time, to takeoff or to cross the runway respectively. Then it applies the following steps to each flight according to this FCFS order:

1. Find the gaps between successive arrivals within which the ready time (to takeoff or to cross) lies. For example, for runway 18C, there may be two gaps, one between two successive arrivals on the same runway 18C and one between two successive arrivals on the dependent runway 23.

2. Find the separation required behind the leading arrival of each of the gaps. Identify the maximum of the leading arrival times plus the required separations as an upper limit on the desired schedule.

3. Find the departures and runway crossings that were already scheduled within the arrival gaps, if any. Identify the required separation behind the last departure operation and behind the last crossing. Identify the maximum of their scheduled times plus their corresponding required separations as another upper limit on the desired schedule.

4. Compare the limiting times computed in steps 2 and 3 and set the candidate schedule to the maximum of the two.

5. Find the required separation before the trailing arrival of each arrival gap and identify the minimum of the trailing arrival times minus the required separations as a lower limit on the desired schedule.

6. If the candidate time computed in step 4 violates the lower limit computed in step 5, identify the following arrival gaps and repeat steps 1 through 6 until a gap is found where the lower limit is not violated. Once found set the operation schedule at that time.

Note 1: The algorithm described is not guaranteed to produce an optimal schedule in terms of maximum throughput or minimum delay because of two reasons: (1) The algorithm schedules flights in a FCFS sequence and a flight is not visited more than once. (2) The algorithm ignores the triangular inequality which is characteristic of the separation requirements. If three flights A, B, and C are operated successively, the triangular inequality implies that the separation required between A and C is larger than the sum of the separations required between A and B and between B and C. In this case, ensuring that C is separated from its immediately leading flight B is not sufficient to ensure that it is separated from the preceding flight A. The algorithm described above only tests the separation from the immediately leading flight and the immediately trailing flight.

Note 2: The algorithm represents nominal operations. It is possible to space the flights by more than the minimum requirements if the capacity is reduced below nominal, by imposing a rate limit within a sliding time window.

3.3.2.2 Runway service model

One of the inputs to the tactical scheduler is the required separation between successive runway operations. These separations were derived using historical data analysis of the landing, takeoff and runway crossing times reported in the ASDE-X database over one year. Statistical distributions were derived for all relevant pairings of successive operations: departure-departure, crossing-crossing, crossing-departure, arrival-departure, and crossing-arrival. Different models were generated for different consecutive aircraft weight classes of arrivals and departures (for example, small behind heavy or B757, small behind large and large behind large) to capture the effect of wake turbulence. Different models were also generated for the different runway pairs (for example, departure on 18C behind an arrival on 18C versus behind an arrival on 23). Different models were also generated for consecutive runway crossings from the same crossing point or from independent crossing points, because crossings from independent crossing points can occur simultaneously.

Figure 3-5 shows an example of a distribution of the separation time between two departures of the weight class large on the same runway. To reduce the effect of lulls and outliers, only operations under queuing were used to derive the required separation distributions. In addition, for each distribution the least frequent (five percent) separations were removed. The effects of these filters are the difference between the top and bottom plots in Figure 3-5.

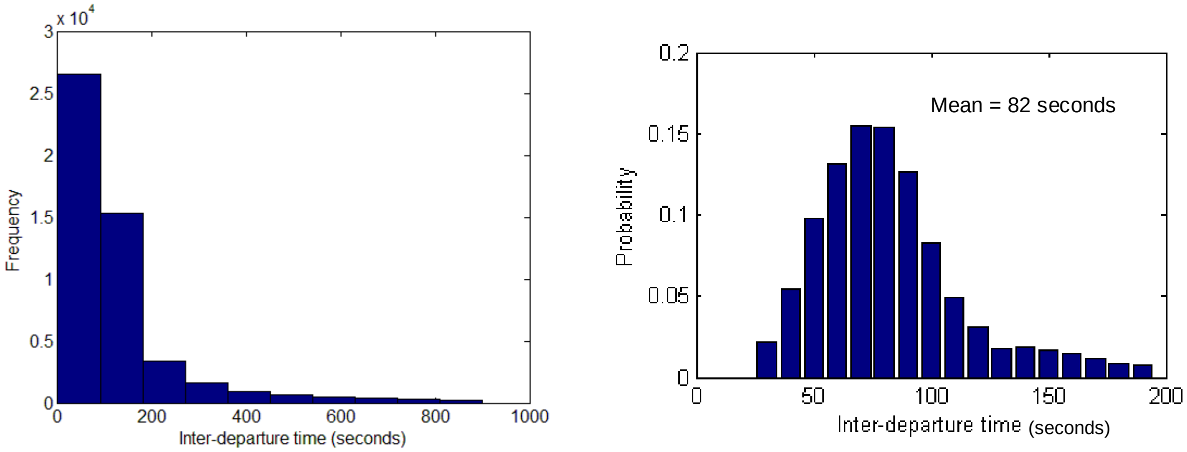


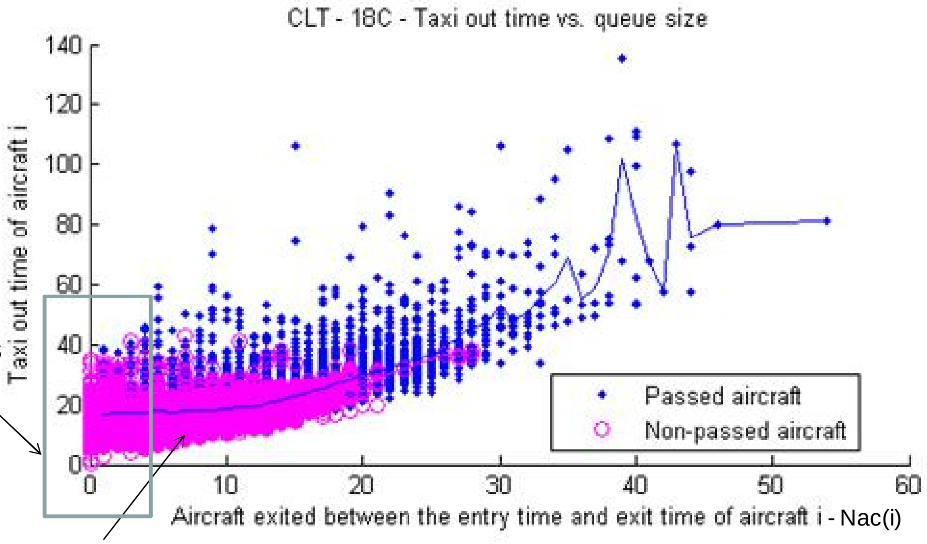
Figure 3-5. Distribution of separation between two successive departures of weight class large on CLT runway 18L (left: Histogram of times below 15 minutes; right: distribution after filtering)

The distributions were used in two cases in the simulation: (1) The tactical scheduler used the means of the distributions for generating a runway operations schedule within the scheduling horizon. (2) The traffic progress over a simulation time step used random sampling from the distributions to introduce an uncertain deviation from the expected schedule generated in (1). In the deterministic baseline, the simulation update used the means of the distributions to be consistent with the expected schedule generated in (1).

3.3.2.3 Unimpeded transit model

The unimpeded transit time of a flight from the gate to the runway and from the ramp spot to the runway is used for estimating the time a flight is ready for takeoff. The unimpeded transit time from landing to the runway crossing point is also used to estimate the time a flight is ready to cross the runway. These ready times are used by the tactical scheduler to schedule the runway operations and by the strategic scheduler to estimate the number of flights that are ready for takeoff but did not takeoff yet. Statistical models of the unimpeded transit time were generated from one year of historical data. ASDE-X provided the landing time, the runway crossing time, the ramp exit time, and the takeoff time of each flight, in addition to the runway that was used by the flight. ASPM provided the pushback time of each flight. Details of the derivation of these models were described in [27]. The process consists of identifying a subset of flights that were not impeded by holding or queuing delays. The process is shown in Figure 3-6. First the transit time of each flight is plotted on the vertical axis versus the queue experienced by the flight. Then to eliminate flights that experienced excessive non queuing delays, the flights that were passed by other flights are eliminated. Then to eliminate flights that experienced queuing delays, the flights that experience small queues up to the queue size where correlation becomes significant between the transit time and the queue are included. The remaining flights are included in the data set used to generate the unimpeded transit time distributions.

To reduce queuing impacts used low queue values before correlation starts



To reduce non-queuing restriction impacts removed passed flights:

for gate-to-runway: Nac(i) = { flights j with pushback(i) ≤ takeoff(j) < takeoff(i) and pushback(j) ≤ pushback(i) }
for gate-to-runway: Nac(i) = { flights j with rampexit(i) ≤ takeoff(j) < takeoff(i) and rampexit(j) ≤ rampexit(i) }
for land-to-cross: Nac(i) = { flights j with landing(i) ≤ crossing(j) < crossing(i) and landing(j) ≤ landing(i) }

Figure 3-6. Queuing analysis to generate unimpeded flights

Distributions were generated for the transit from the ramp or the gate to the runway for each pair of airline (to represent different ramp areas) and runway. Figure 3-7 shows an example of the unimpeded transit model from the ramp to the runway for one airline. Distributions were also generated for each pair of landing runway and runway crossing point. Figure 3-8 shows an example of the unimpeded transit model from the landing to the runway crossing.

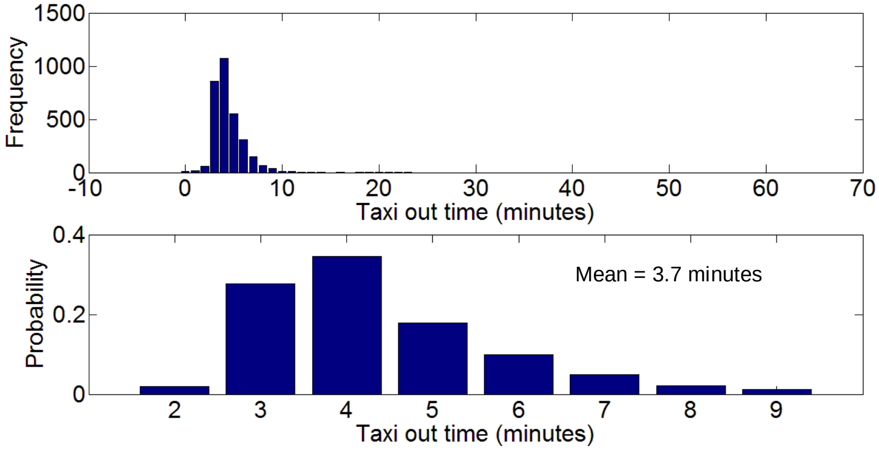


Figure 3-7. Distribution of unimpeded transit from ramp to runway 18L for one airline (top: histogram before filtering; bottom: distribution after filtering)

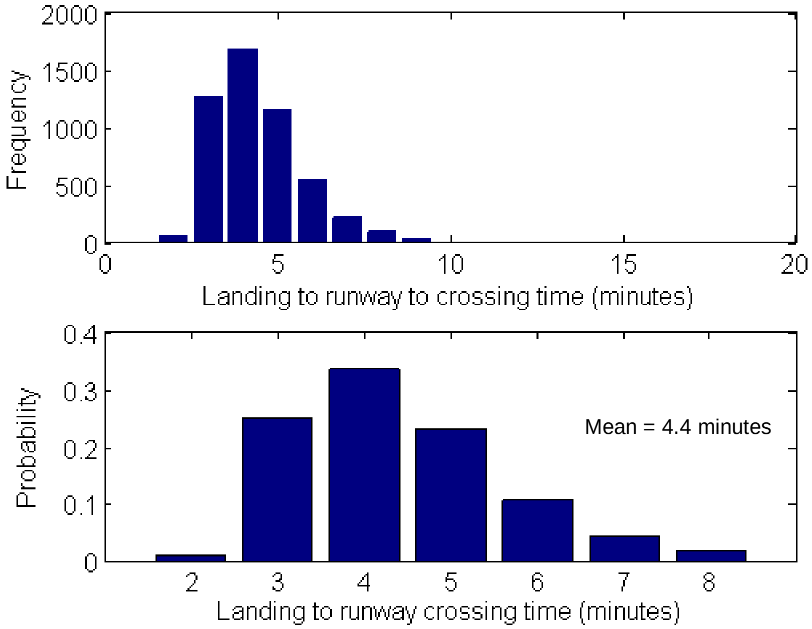


Figure 3-8. Distribution of unimpeded transit from landing runway 18L to crossing runway 18C for one crossing point (top: histogram before filtering; bottom: distribution after filtering)

To reduce the effect of lulls and outliers, only operations when queues were present were used to derive the transit time distributions [27]. In addition, for each distribution the least frequent five percent was removed. To show the effect of filtering the top of each figure shows the histogram of the data before filtering and the bottom shows the probability distribution after filtering.

The distributions were used in two cases in the simulation: (1) The tactical and strategic schedulers used the means of the distributions for scheduling runway operations and gate/ramp release times within the scheduling horizon, respectively. (2) The traffic progress over a simulation time step used random sampling from the distributions to introduce an uncertain deviation from the expected unimpeded transit used in (1). In the deterministic baseline, the simulation update used the means of the distributions to be consistent with the expected unimpeded transit times used to generate the schedule in (1).

3.3.3 Simulation Update Cycle

The strategic scheduler was called dynamically every fifteen minutes with a horizon that extended for the rest of the day. The tactical scheduler was also called every fifteen minutes because the analysis focused on the performance of the strategic scheduler and no effects were measured for running the tactical scheduler several times during every strategic period. The tactical scheduler horizon also extended for the rest of the day to be consistent with the strategic scheduler horizon. During a fifteen minute increment, the simulation performed the following tasks:

1. It pushed back all flights whose pushback time was due. The flight's actual pushback time (OUT time recorded in ASPM) was used to indicate the flight's readiness for pushback and hence a flight was never pushed back prior to this time. The following cases were considered: (a) If a flight was not metered (not delayed by the metering algorithm) or was metered and its pushback time assigned by the metering algorithm was less than its actual pushback time, then it was pushed back at its actual pushback time. (b) If a flight was metered and its pushback time assigned by the metering algorithm was larger than its actual pushback time, then it was pushed back at its metered pushback time. Note that if the flight plan pushback time was used to estimate demand, a metered flight may be delayed by metering relative to its flight plan pushback time but not delayed relative to its actual pushback time (which in this case is larger than the flight plan pushback time). In this case the flight was released at its actual pushback time, emulating a delay in the flight's readiness for pushback.

2. It released from the spot all flights whose ramp exit time was due. The logic is the same as described in step 1, using the actual ramp exit time recorded in the ASDE-X database instead of the actual pushback time and the ramp exit time assigned by metering instead of the pushback time assigned by metering. A flight plan ramp exit time was estimated by adding an unimpeded transit time from the gate to the taxi spot to the flight plan pushback time.

3. It added an unimpeded transit time for the flights that did not takeoff yet to estimate their ready for takeoff time. The transit time is from the ramp to the runway for flights that exited the ramp and from the gate to the runway for flights that did not exit the ramp. In deterministic scenarios, the simulation used the means of the unimpeded transit time distributions. In scenarios modeling transit time uncertainty, the simulation sampled transit times from the unimpeded transit time distributions.

4. It updated the takeoff times of the flights within the simulation increment by running the tactical scheduler given the takeoff ready times updated in step 3. This step simulates actual runway operations. In deterministic scenarios, the simulation used the means of the distributions of the separation times between successive runway operations. In scenarios modeling runway operations uncertainty, the simulation sampled separation times from the separation time distributions.

5. Finally, the simulation removed flights whose takeoff time (computed in step 4) was due.

3.4 Analysis and Results

The fast time simulation analysis was applied to one day, July 17, 2012 at CLT. The day was selected as a typical busy day with 759 departures, 751 arrivals and considerable queuing. On this day CLT operated in the south configuration landing runway 18L, 18C and 23 and departing runways 18C and 18L. Landings on runway 18R cross runway 18C on the way to the ramp. In the following subsections, we present a series of analyses using the scenarios outlined in Table 3-1.

Table 3-1. Analysis Scenarios

Scenario	1	2	3	4
Metering	Off	On	On	On
Demand Uncertainty	Off	Off	On	Off
Transit & Runway Separation Uncertainty	Off	Off	Off	On
Figures	6	7-12	13-14	13-14

First we present a validation of the simulation model by comparing its baseline performance to the actual operations on that day. The baseline used for the validation is the first scenario in Table 1, where metering and all uncertainties are turned off. Then we analyze the impact of metering using the three metering strategies by comparing the deterministic second scenario to the deterministic first scenario in Table 1. In the second scenario metering is turned on but the uncertainties are turned off as in the first scenario. Finally, we present two analyses that compare the performance of the three metering strategies under uncertainty scenarios: In the third scenario we investigate the effect of departure demand uncertainty alone and in the forth scenario the effect of transit and service time uncertainties, with demand uncertainty turned off. Three values of the queuing parameter thresholds were used for each analysis: the value that corresponded to the throughput saturation threshold (see Figure 3-3), 75% of this value and 50% of this value, for each runway.

The performance of the metering control strategies were measured using the following metrics highlighting the tradeoff between efficiency and throughput:

1. The reduction in the congestion level due to metering measures increased efficiency.
2. The delay that was absorbed at the gate due to metering, which corresponds to the reduction in congestion, also measures increased efficiency.
3. The change in the flight takeoff time due to metering measures the overall delay and the conformance to the given runway schedule computed by the tactical scheduler. Hence it also measures the impact on runway throughput.

As explained in each of the following subsections, the metrics are measured between the two scenarios compared. These scenarios are either two scenarios from Table 1 or a scenario from Table 1 compared to a corresponding baseline that differed from the scenario only by turning metering off (i.e., no strategic scheduler).

3.4.1 Baseline Model Validation

Figure 3-9 compares the performance of the model relative to the actual operations of July 17, 2012. It plots the actual number of flights that pushed back for runway 18L but did not take off at each five-minute increment. It also plots the corresponding number that was estimated by the baseline simulation of scenario 1 in Table 3-1.

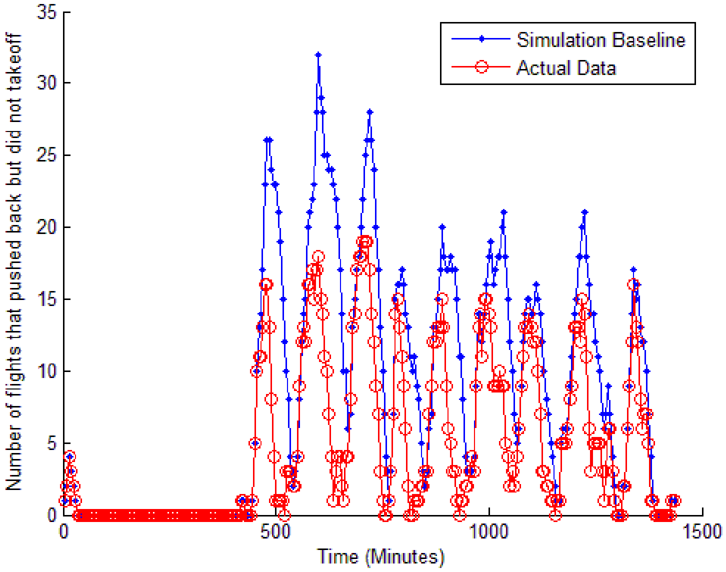


Figure 3-9. Actual and simulated baseline operations of one day

The baseline scenario 1 did not apply any metering; rather flights were released from the gate and ramp according to their actual pushback and ramp exit times reported in the historical data (ASPM and ASDE-X respectively). Scenario 1 is also deterministic as all the uncertainties were turned off. Turning demand uncertainty off meant that the demand was estimated using the actual pushback, ramp exit, and landing times. The means of the unimpeded transit time distributions were added to these times to estimate the takeoff ready time and the runway crossing ready time of each flight. Turning runway operations and transit time uncertainties off meant that no deviations were added between the runway separation times and unimpeded transit times that were assumed by the tactical and strategic schedulers and the times that materialized in the simulation update.

Figure 3-9 shows a reasonable match between the actual and simulated levels of congestion over the course of the day, with some overestimation by the model particularly during the earlier departure banks. The model performance can be improved by calibrating the separation and transit time distributions. For example, the mean of the inter-departure time distribution shown in Figure 3-5 may be reduced by removing more of the tail. This would increase the modeled runway service rate and hence reduce the resulting congestion in the model. However, since the analysis reported in this chapter only demonstrates mechanism effects (rather than reports validated benefits) such calibration was not conducted completely for this analysis and will be performed in future research for estimating validated benefits.

3.4.2 Impact of Metering on Congestion in Deterministic Scenario

In this section, the deterministic baseline described in the previous section (scenario 1 in Table 1) is compared to the corresponding deterministic metering scenario (scenario 2 in Table 1), which only differed from scenario 1 by applying metering. The uncertainties were all turned off. Figure 3-10, Figure 3-11, and Figure 3-12 show the impact of metering on queuing by comparing the queuing parameters under the baseline that does not apply metering and under the three control strategies. Figure 3-10 plots on the vertical axis the number of flights that pushed back from the gate for runway 18L but did not take off at each five-minute increment of the day. Figure 3-11 plots on the vertical axis the number of flights that left the ramp but did not takeoff at each five-minute increment. Finally, Figure 3-12 plots on the vertical axis the number of flights that spent their unimpeded transit time to the runway but did not take off at each five-minute increment. Each figure contains four plots: one for the baseline that did not apply metering and one for each of the three control strategies, applied at 75% of the throughput saturation threshold as an example.

The three figures show that the congestion and queuing has been reduced by all the control strategies relative to the baseline. However, the queuing parameter that is displayed on the vertical axis in each of the figures was maintained at a stable level only by the method that explicitly controlled it. On the other hand, the parameter was more variable under the other methods that implicitly controlled it. Namely, as shown in Figure 3-10, the number of flights that pushed back but did not take off was controlled at a stable desired value of eleven flights by the gate-to-runway control strategy that explicitly controlled this parameter. It was more variable under the other two strategies that controlled either the number of flights that exited the ramp but did not take off (spot-to-runway control) or the number of flights that spent their unimpeded transit time to the runway but did not take off (ready-to-runway control). Similarly, the number of flights that exited the ramp but did not take off in Figure 3-11 was controlled at a stable level of five flights only by the spot-to-runway control method. And finally, the number of flights that spent their unimpeded transit time to the runway but did not take off was controlled at a desired level of three flights only by the ready-to-runway control method.

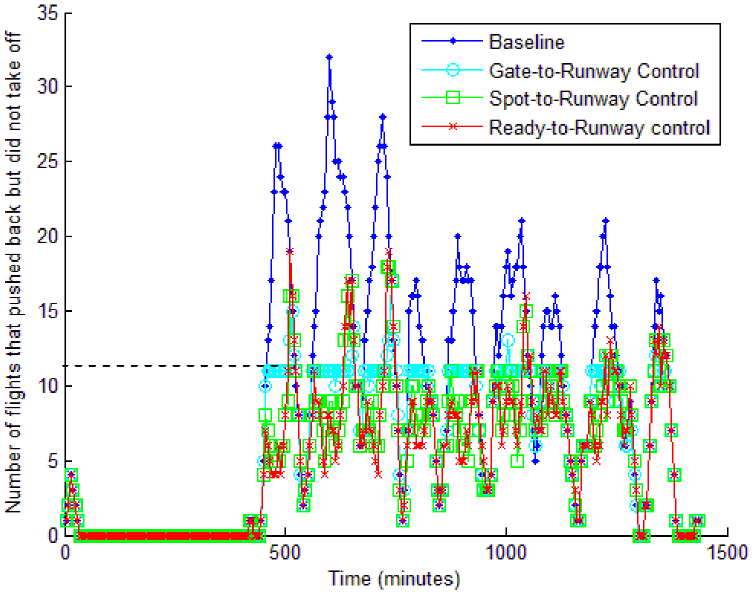


Figure 3-10. Metering impact on the number of flights that pushed back but did not takeoff

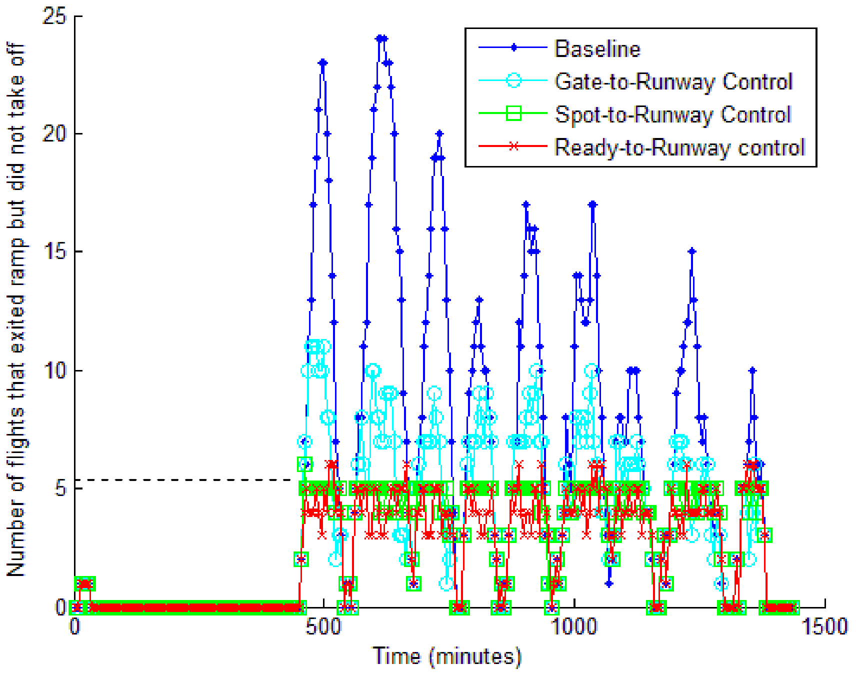


Figure 3-11. Metering impact on the number of flights that exited the ramp but did not takeoff

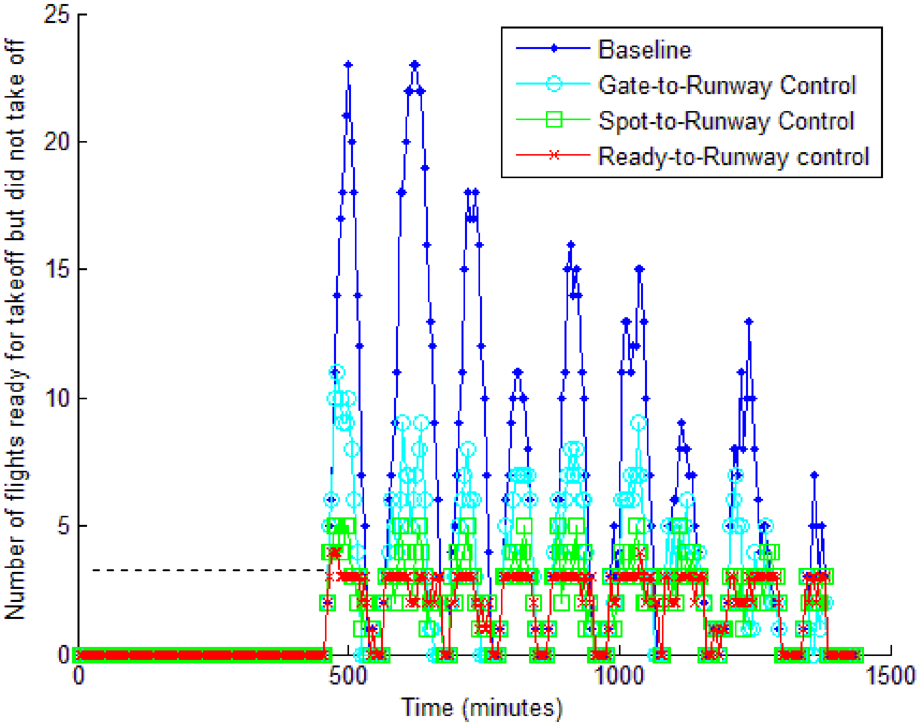


Figure 3-12. Metering impact on the number of flights that were ready to takeoff but did not takeoff

One can observe that occasionally, the parameter that was controlled at a stable level in each of the figures spiked above the stable level which was maintained most of the time. For example, the number of flights that pushed back but did not take off in Figure 3-10, increased to thirteen flights around the minute 1000. The reason for this violation of the desired queuing parameter threshold is the conformance to the runway schedule constraint. As was mentioned in the metering algorithm description (step 3), a flight is released from the gate such that it can transit unimpeded to the runway before its scheduled takeoff time, even if the desired queuing threshold is violated. The gate occupancy constraint (step 4 of the metering algorithm) can produce the same effect, where a flight is released before causing arrivals to wait for their gate, at the expense of violating the queuing threshold. However, in this analysis the gate capacity of 97 gates, shared by all airlines, was sufficiently high that the constraint was not invoked. Future research will investigate variations on the gate capacity assumptions made in this analysis.

Figure 3-13 compares the average values for each of the queuing parameters in Figures 3-7 through 3-9, over the busy period of the day (between 500 and 1400 minutes). The figure shows that all of the metering strategies reduced the average congestion relative to the baseline without metering. The mean values in Figure 3-13 are lower than the thresholds depicted in Figures 3-7 through 3-9, because the thresholds are used by the metering algorithm as maximum target values (except when runway conformance or gate capacity constraints necessitate otherwise) and the averaging period included high and low departure demand.

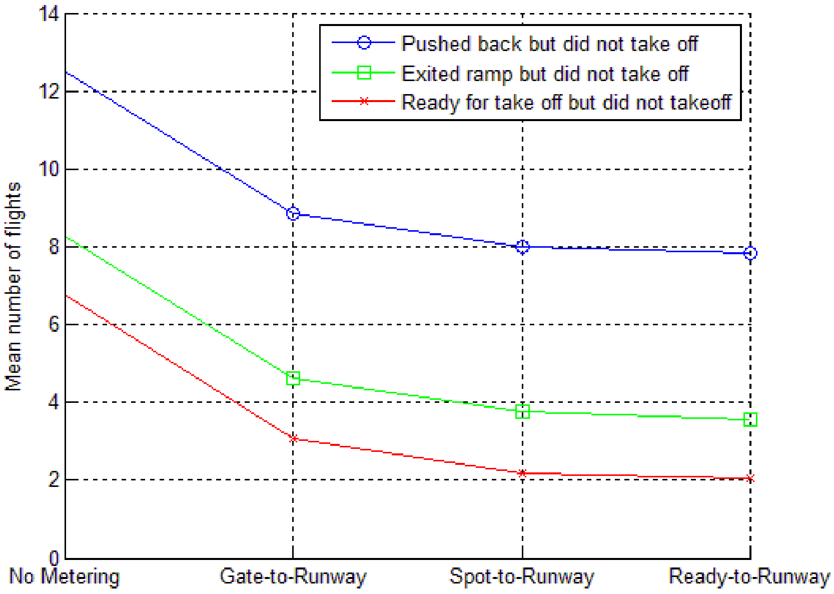


Figure 3-13. Average queuing under different metering strategies

Figure 3-13 also shows that the metering method that controlled the number of flights that pushed back but did not take off (gate-to-runway) resulted in higher mean values than the other two strategies for all of the three queuing parameters, in the deterministic scenario. This indicates a clear dependence of the congestion and queuing levels, measured by any of the three queuing parameters, on the size of the system being controlled.

3.4.3 Impact of Buffer Size in Deterministic Scenario

As mentioned in the introduction, the desire is to maximize the efficiency gains of transferring as much delay to the gate as possible while minimizing the impact on runway throughput. Figure 3-14 compares the performance of the three metering strategies in terms of the average delay absorbed at the gate because of the metering. It is followed by Figure 3-15, which shows the corresponding average increase in the takeoff time under each strategy. Each figure shows these averages for the three threshold values of the queuing parameters: the value that corresponded to historical throughput saturation, 75% of this value and 50% of this value.

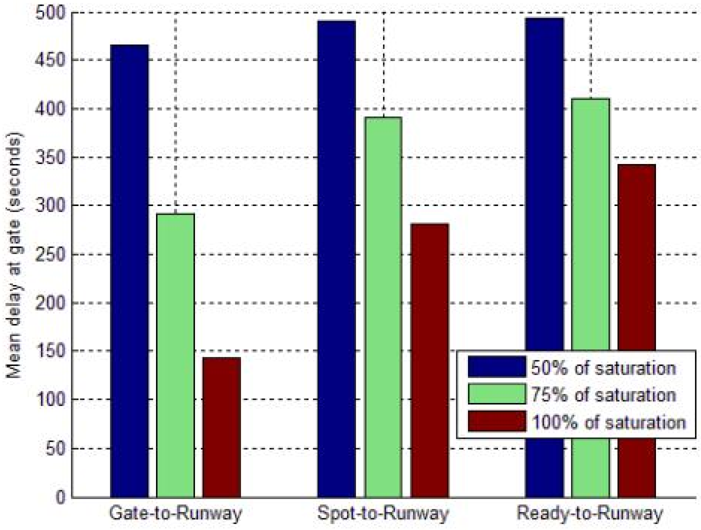


Figure 3-14. Gate delay using different control strategies under deterministic scenario

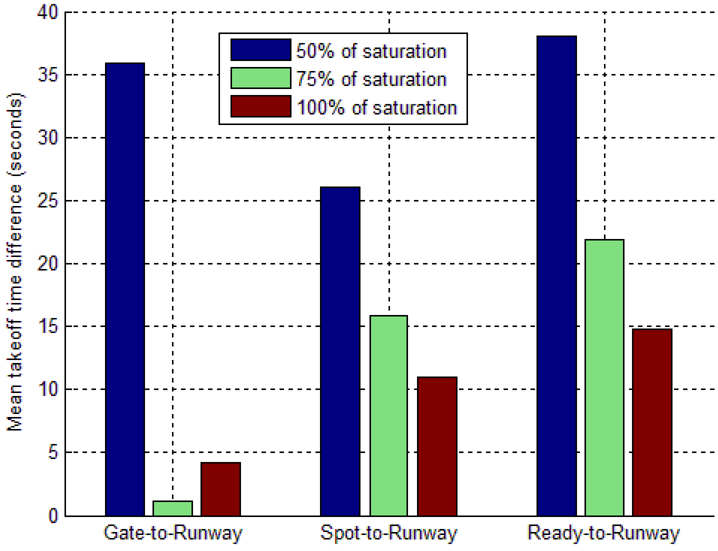


Figure 3-15. Takeoff time difference using different control strategies under deterministic scenario

Figure 3-14 shows that for each of the metering strategies, the average delay absorbed at the gate was larger using the smaller queuing parameter thresholds as expected because more delay at the gate is needed to maintain smaller queuing buffers. Figure 3-15 demonstrates that the larger amount of delay that is transferred to the gate occurs at the expense of delayed takeoff time. This tradeoff is clear for all the strategies. For example, using gate-to-runway control with the smallest queuing parameter threshold at 50% of the saturation level caused an average takeoff delay of 36 seconds. However, with the queuing parameter threshold set at the level needed for throughput saturation (100% of saturation), this takeoff time difference is negligible at less than five seconds. To achieve this low takeoff time delay however, a smaller average metering delay of 150 seconds (2.5 minutes) is possible.

The gate-to-runway control strategy performed better than the other two strategies in terms of maintaining small takeoff time difference as evident from Figure 3-15. As mentioned above, at the higher queuing parameter thresholds of 75% to 100% of the throughput saturation level, it was possible to maintain the takeoff time difference below five seconds with the gate-to-runway control strategy, with an average metering delay at the gate of 150 seconds. The spot-to-runway and ready-to-runway control strategies, respectively resulted in takeoff time differences of ten and fifteen seconds at the throughput saturation levels and metering gate delay of 275 and 350 seconds. This indicates that these strategies are less effective at maintaining throughput using the thresholds derived from the historical data analysis. Either larger queuing parameter thresholds should be used – ones that reduce the takeoff time difference to near zero, or they should be combined with gate-to-runway control elements to maintain sufficient flights in the system to ensure throughput saturation.

3.4.4 Impact of Uncertainty

The two non-deterministic scenarios in Table 3-1 are analyzed in this section: scenario 3 adds uncertainty in the demand and scenario 4 adds uncertainty in the transit time and runway separations (runway service rate) to the deterministic scenario described in the previous section. In scenario 3, the schedulers used the pushback times published in the flight plans instead of the actual pushback times, when estimating the demand (i.e., takeoff ready times) of flights that have not pushed back yet. To estimate the demand for the runway the modeled unimpeded transit time was added to the pushback time (for flights that did not exit the ramp yet) and to the ramp exit time (for flights that exited the ramp). No uncertainty was modeled in the arrival demand as the actual landing times reported in ASDE-X/PDARS were used for all cases. Otherwise the simulation is identical to the one used in the previous section. The schedulers and the simulation update used the means of the unimpeded transit time and separation time distributions to be consistent with each other. In scenario 4, the schedulers used the means of the unimpeded transit time and the separation time distributions when scheduling flights. On the other hand the simulation update during each time increment used random sampling from these distributions to introduce an uncertain deviation between the progress of a flight and its desired schedule. In this scenario the demand was known perfectly by the schedulers and the simulation update used the actual pushback and ramp exit times rather than the flight plan schedule.

The analysis of the previous section showed that the queuing parameter threshold that corresponds to the historical throughput saturation is needed to maintain runway throughput even in the deterministic case. Therefore, we use only this threshold in the analysis of the impact of uncertainty in this section. The threshold values that are smaller than this threshold resulted in similar trends but inferior performance. Figure 3-16 compares the mean gate delay that was imposed by the three metering strategies between the two uncertainty scenarios and the deterministic scenario from Figure 3-14. Figure 3-17 compares the corresponding takeoff time delays.

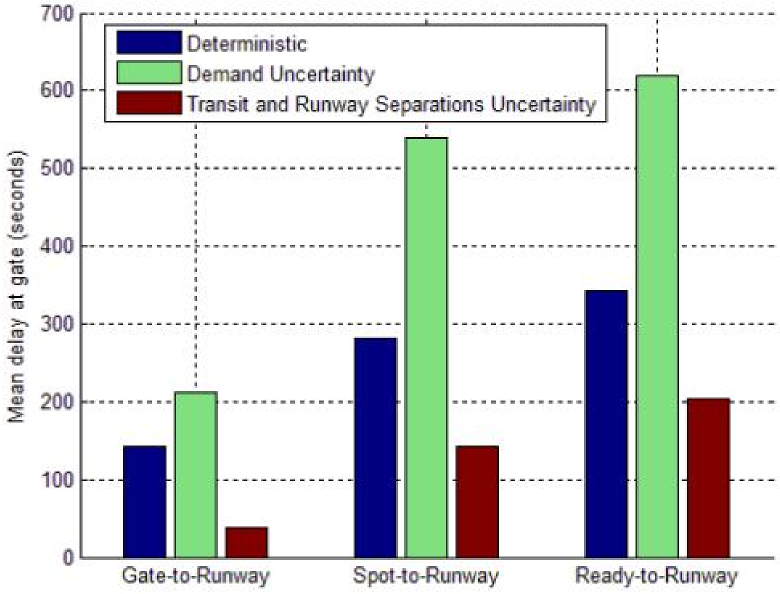


Figure 3-16. Gate delay using different control strategies under uncertainty

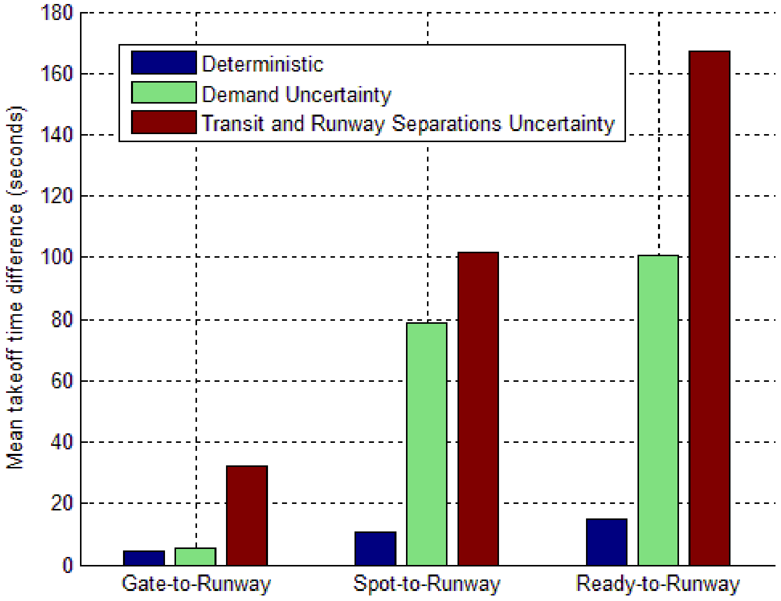


Figure 3-17. Takeoff time difference using different control strategies under uncertainty

3.4.4.1 Impact of Demand Uncertainty

Figure 3-17 shows that the uncertainty in the demand increased the takeoff time delay significantly (by almost ten times in the spot-to-runway and ready-to-runway control strategies) compared to the deterministic scenario. The gate-to-runway control strategy with the queuing parameter threshold set at

100% of the throughput saturation level is the only strategy that was able to reasonably handle the demand uncertainty; it kept the takeoff time delay small at about eight seconds. The corresponding gate delay was also significant at about 200 seconds for this case as shown in Figure 3-16. This gate delay value is significantly smaller than the gate delay that was applied by the other two control strategies. Since larger gate delay leads to larger takeoff time delay, this explains the smaller takeoff time delay that resulted under the gate-to-runway control strategy.

3.4.4.2 Impact of Transit Time and Runway Separation Uncertainties

Figure 3-17 shows that the uncertainty about the transit and runway service times increased the takeoff time delay in all of the metering strategies compared to the deterministic scenario. The increase was greater than the one due to the uncertainty in the demand, with a lowest takeoff time delay of about 35 seconds under the gate-to-runway control strategy. This may be due in part to the shorter time available to adjust to the uncertainties in the transit time and runway separations relative to the uncertainty in the pushback time (demand). An interesting observation is that the gate delay applied by all the metering strategies was smaller under the transit time and runway separations uncertainty compared to under demand uncertainty and the deterministic scenario, as evident from Figure 3-16. For example, the gate-to-runway metering strategy delayed flights at their gate only forty seconds on average leading to 35-second average takeoff time delay. Therefore, under these uncertainties, the metering strategies are much less effective at transferring delay to the gates without impacting the runway schedule or throughput. Hence, further research is needed to devise more robust metering approaches to handle such uncertainties.

3.5 Concluding Remarks

We presented a simulation-based analysis that compared the performance of different control strategies for departure metering, as part of an integrated departure scheduling concept. The performance was compared in terms of the ability of the different strategies to conform to a runway schedule and hence maintain runway throughput while transferring queuing delays from the airport movement area to the gate to save on fuel burn and emissions. The following observations are made:

A control strategy that applied metering by maintaining the number of flights that pushed back but did not takeoff was more effective than the strategies that controlled the number of flights that exited the ramp but did not takeoff or controlled the number of flights that spend their unimpeded transit time to the runway but did not takeoff. This was true both in a deterministic case and in cases with uncertainties about the demand, the transit time, and the runway service times. This observation indicates the need to maintain flights throughout the system between the gates and the runway rather than just at the runway end or between the ramp and the runway, to ensure continuity in traffic supply to the runway.

The best performing metering strategy that controlled the number of flights that pushed back but did not take off was able to handle the demand uncertainty modeled in this analysis, keeping the runway throughput high while delaying flights at their gate by an average of 2.5 minutes. However, under transit time and runway service uncertainties, a small gate delay impacted the runway throughput significantly even with the largest queuing parameter threshold.

It should be noted that these observations may not generalize to other scenarios of uncertainties that were not analyzed. Further research is being conducted to investigate the performance under additional scenarios of uncertainties and with alternative more robust metering strategies such as using a combination of the three strategies analyzed. Future research will also investigate different interactions between the strategic metering process and the tactical runway scheduling and schedule conformance processes. Finally, the application of these concepts will be studied at other airports and configurations.

4 Simulation Assessment of Departure Metering Impact on Arrival Gate Blocking

Airports and their terminal airspaces are key choke points in the air transportation system causing major delays and adding to pollution. A solution aimed at mitigating these chokepoints integrates the scheduling of runway operations, flight release from the gates and ramp into the airport movement area, and merging with other traffic competing for downstream airspace points. In addition, departure scheduling is integrated with the schedules of arrivals sharing resources such as runways and gates. Within this integrated concept, we present a high-fidelity simulation analysis of the departure metering process, which delays the release of flights into the airport movement area while balancing two competing objectives: (1) maintaining large enough queues at the airport resources to maximize throughput and (2) absorbing excess delays at the gates to save on fuel consumption, emissions, noise, and passenger discomfort. We assess the impact of delaying departures at their gates on blocking the arrivals destined for the same gates. We apply metering strategies that avoid this blocking effect by scheduling the release of a metered departure at certain time buffer prior to the next estimated arrival time at the gate. We determine the time buffers needed to eliminate delays on arrivals as a result of departure metering and assess the amount of departure metering that is feasible under such conditions.

4.1 Introduction

Delays originate mostly at major airports, and particularly at ones that constitute complex metroplexes of multiple interacting airports [1]. Major causes of delay at these metroplex systems are constrained resources on the airport surface, particularly the runways and in the surrounding terminal airspace. These complex systems are characterized by high levels of interaction between flows of multiple adjacent airports sharing arrival and departure fixes and competing for gaps in the overhead traffic streams. Solutions to mitigate these choke points have consisted mostly of isolated concepts and capabilities that are applied to components of the system. For example, concepts have focused on arrival flow management for arrival metering, sequencing and spacing [2, 3], on departure flow management such as departure metering at gates [4-6] and precision departure release control [7, 8], or on runway scheduling [9, 10]. Integrated solutions are needed in order to reap the benefits of the isolated capabilities [11]. Research has been conducted to demonstrate integrating arrival operations from the en route phase of flight to the landing [12]. Initial research into integrating runway arrival and departure operations has also been conducted, generating preliminary concepts and algorithms [13-15]. In this chapter, we describe a research activity that supports a NASA Advanced Technology Demonstration (ATD-2) project aimed at integrating departure and arrival operations in a metroplex, with an emphasis on integrating departure operations within arrival constraints [16-17].

The integration of airspace and surface operations in a metroplex system aims to maintain high throughput of the system, efficient operations through expedited, uninterrupted movement, and minimum fuel burn and emission. This integration attempts to achieve these objectives through in part the generation of a coordinated schedule of departures at key resources or control points. These control points include primarily the release from the gates, the exit from the ramp area into the airport movement area, the takeoff from the runways, and the crossing of departure fixes. It may be possible to simultaneously achieve these objectives under deterministic conditions. However, uncertainty in the operations and the environment brings about tradeoffs among them. For example:

1. In order to maintain high throughput under uncertainty, delay or queue buffers are needed to keep pressure on the choked resources of the system and take advantage of any service

opportunities that may arise. Parallel queues provide controllers the ability to sequence departures optimally and the controllability needed for conformance to prescribed constraints such as flow management restrictions. If the schedule assumed the fastest transit between resources with zero delay or queue buffers it would be violated due to any disturbance that results in longer transit time. This requires absorbing some delay near downstream resources, such as in the airspace near departure fixes and on the airport surface near the runways.

2. On the other hand, delay is absorbed more efficiently and cleanly on the airport surface (movement area) rather than in the airspace, and in turn, at the gate while the engines are off rather than on the airport surface. Absorbing delay at the gates or in the ramp area saves on fuel consumption, emissions and noise in addition to allowing passengers to absorb some of the delay more comfortably while off the plane.

Therefore, the integrated scheduler needs to decide on the distribution and allocation of delay between the interconnected resources. NASA's ATD-2 concept includes a strategic surface scheduler which schedules the release of flights from the gates and/or the ramp when the demand for the runways is predicted to exceed their capacity in order to balance the tradeoff between throughput and efficiency. Key parameters used in this decision are the desired queue or delay buffer sizes that the scheduler should target to achieve an appropriate balance between these objectives. In addition, departure scheduling is integrated with the schedules of arrivals sharing resources such as runways and gates.

We present a simulation-based analysis of the departure metering process. In particular, we assess the impact of delaying departures at their gates on blocking the arrivals destined for the same gates. We apply metering strategies that avoid this blocking effect by scheduling the release of a metered departure at certain time buffer prior to the next estimated arrival time at the gate. We draw insights on the performance of different metering with gate blocking limit strategies in terms of conforming to the runway schedule and minimizing the blocking of arrivals while transferring delay to the gates. We determine the time buffers needed to eliminate blocking delays on arrivals as a result of departure metering and assess the amount of departure metering that is feasible under such conditions. We assess the performance of the metering strategies under different scenarios of uncertainty about the demand, the flight transit time and the runway schedule.

In the following sections we describe the ATD-2 operational concept and assumptions made for this analysis. Then, we describe the fast-time simulation approach and the models that were developed and used for this analysis, along with the underlying algorithms. The analysis results are then reported with insights on the performance of the different metering strategies under different uncertainty scenarios, ending with concluding remarks and future extensions.

4.2 ATD-2 Concept and Assumptions

The ATD-2 concept includes several attributes that achieve the integration between the scheduling of operations at different resources. Three of these attributes are:

1. Integration between upstream and downstream schedules. Namely, the schedule of operations at the runway considers constraints stemming from scheduling flows at downstream airspace shared points such as TRACON departure gates and gaps in the overhead en route stream. In turn, the schedule of the release from the gates or ramp area into the airport movement area takes into consideration the restrictions stemming from the schedules at the runways and at downstream airspace merge points.

2. The concept integrates a strategic metering scheduler with a tactical control scheduler. The strategic scheduler implements the metering process; it runs at low frequency and attempts to control the congestion at reasonable levels when demand exceeds capacity. The tactical scheduler runs more frequently and controls the conformance of the flights to the integrated schedules at the different resources such as the runway and the release from the ramp and gates into the airport movement area.

3. The concept integrates the scheduling with departures and arrivals sharing resources such as runways, gates, and airspace volumes. In particular, the concept schedules departures whitin the prescribed arrival schedules at these resources. Minimal modification to the prescribed arrival schedules may be considered; however, priority is given to the arrivals.

The interactions between the schedulers are a topic of continued research and design. For this analysis, we made some assumptions about these interactions, as described in the following paragraphs of this section.

The strategic scheduler implements a metering process by generating a desired schedule of releases from the gates and ramp area such that congestion is reduced and a desired level of queuing buffers is maintained. The queuing may be measured and controlled using different parameters [29], the one used in this analysis is the number of flights that left the gate but did not take off.

The strategic scheduler requires a runway schedule in order to estimate the level of congestion and queuing based on the demand. Rather than generating its own runway schedule, it is assumed in this analysis that the strategic scheduler takes the most recent runway schedule generated by the tactical scheduler as an input. Therefore, the tactical scheduler was given the same time horizon as the strategic scheduler in this analysis; otherwise the strategic scheduler would need to extend the tactical schedule to a longer horizon. The tactical schedule ensures the separation requirements between successive runway operations and integrates restrictions stemming from downstream schedules at departure fixes and en route overhead merge points, and possibly at arrival metering points and destination airports. In this analysis, the model used for the tactical scheduler (described in the next section) applies the separation requirements; however, it does not explicitly take the other restrictions into consideration. A prior paper described an algorithm for this integration of downstream schedules into the runway schedule [28].

The strategic scheduler generates desired release times from the gates to control the queuing parameter while ensuring that the input runway schedule remains feasible. The flight release time from the gate ensures that the flight can transit unimpeded to the runway in time to make its assigned runway time, even if the queuing parameter exceeded the desired threshold. In this case the queuing parameter may exceed the desired level in favor of enabling the runway schedule to be feasible. Alternative schemes that are not analyzed include the strategic scheduler giving priority to maintaining the desired queuing buffers over conformance to the runway schedule. In this case the strategic scheduler may recommend gate release times that maintain the queuing parameter at or below the desired level but require changes to the input runway schedule. The strategic scheduler can suggest the changes in the runway schedule that correspond to the recommended gate release times. However, ultimately the tactical scheduler determines the final runway schedule and the corresponding gate release time reconciling the strategic scheduler recommendations with other constraints and objectives.

The tactical scheduler has the task of ensuring the conformance of flights to the runway schedule and to the releases from the gate and ramp into the movement area recommended by the strategic scheduler. It has to make a decision if the two schedules cannot be met simultaneously. In actual operations, this decision should be mostly resolved in favor of meeting the runway schedule to maintain the throughput of the runway bottleneck at the expense of exceeding the desired queuing buffers. However, in this analysis we assume that the tactical scheduler adheres to the desired release times recommended from the strategic scheduler in order to measure the performance of the strategic scheduler in terms of enabling the conformance to the runway schedule. Otherwise the tactical scheduler would override the release times computed by the strategic scheduler.

The strategic scheduler takes into account capacity constraints at the gates by ensuring that the departures are not delayed excessively such that arriving flights requiring the gate incur delay. The strategic scheduler may not have access to full information about the airline gate capacities and procedures hence this constraint needs to be approximated by the strategic scheduler. In this analysis we assumed that the gate assigned to a flight is known exactly by the scheduler. Future research may extend to more relaxed assumptions where the scheduler may not know the exact gate assignment but infer the gate capacity based on the number of gates available to an airline and assuming some flexibility in changing gate assignment to accommodate metering delays.

4.3 Approach

The analysis was performed using NASA's high-fidelity simulation environment SOSS (surface operations scheduling and simulation). SOSS is a trajectory-based simulation of the airport operations on a node-link representation of the airport surface. It applies default taxi routes and default sequencing strategies to conrol the progress of operations and their usage of the shared resources. It models behaviors such as plattoning and yielding between flights such that queuing effects are manifested. It also provides uncertainty modeling of certain parameters such as speed. It provides an interface to external schedulers that can dictate to the simulation the release of flights from different control points on the airport surface such as gates, taxi spot, taxi intersections, and runways.

In this analysis, SOSS was interfaced to an external scheduler written in MATLAB which computed and dictated the release of departures from the gates. The scheduler included scheduling algorithms for the runway operations and the release from the gates according to metering strategies as described in the next section. The scheduler used the information provided by SOSS every cycle about the state of the flights, their routes, and unimpeded transit along the routes. The scheduler used this information to estimate the demand for the runway resources based on which it scheduled the runway operations given separation requirements. The scheduler in this analysis used the same separation requirements and transit times used by the simulation to reduce the discrepancy between them. Then the scheduler computed the gate release times and sent them to SOSS every cycle. In this analysis a cycle of five minutes was used and a scheduling horizon of one hour. Future research may include variation of these parameters.

4.4 Models and Algorithms

The algorithms and underlying models used for the main components of the scheduler are described in the following subsections: the metering algorithm and the runway scheduling algorithm.

4.4.1 Metering Algorithm

The strategic scheduler takes a runway schedule as an input and computes gate release times that maintain a queuing parameter at or below a desired value. The queuing parameter used in this analysis is the number of flights that left the gate but did not take off. The exit from the system is the takeoff time and the entry to the system is the pushback, henceforth also called the entry time.

Based on the input runway schedule, the metering algorithm computes the gate release time for each flight within its scheduling horizon. The metering algorithm first ranks the flights by the estimated pushback time as a representation of the first come first serve (FCFS) order and then performs the following steps for each flight in this FCFS order:

1. Estimate the queuing parameter value at the estimated entry time of the flight.
2. Determine the release time that satisfies the desired queuing parameter threshold: If the value estimated in step 1 is larger than the target parameter threshold, perform the following two steps:
    a. Identify the flights that have already entered the system (their entry time is earlier than or equal to the entry time of the flight being scheduled) but have not exited the system (their takeoff time is later than the entry time of the flight being scheduled).
    b. Set the desired entry time at the takeoff time of the flight with the earliest takeoff time that reduces the parameter to the target value.
3. Apply the runway schedule conformance constraint if turned on: If the algorithm is set to conform to the runway schedule, set the desired entry time to the minimum of two numbers: (1) the computed value in step 2 or, (2) the flight's schedule takeoff time minus its unimpeded transit time from its entry to the queuing system to the runway. This ensures that the flight is able to transit unimpeded to the runway in time to meet its takeoff schedule. If not set to conform to the runway schedule, the desired entry time from step 2 is unaltered. In this analysis, the runway schedule conformance constraint was always turned on.
4. Apply the gate constraint if it is enforced: If an arrival is estimated to arrive at the gate of the departure before its candidate meering release time computed in Step 3, set the candidate metering gate release time to the maximum of (1) a time buffer prior to the estimate arrival time at the gate and (2) the time the departure is able to be ready for pushback. The time buffer was varied in this analysis to determine its effect of the amount of gate blocking and departure metering. The pushback ready time was provided by the SOSS simulation to the scheduler.

4.4.2 Runway Scheduling Algorithm

The runway scheduler generates a runway operations schedule, which is used to simulate the runway operations. This schedule is assumed to be shared with the strategic scheduler described above, which uses the schedule as an input in the metering process. The runway scheduler inserts departures and runway crossings in a FCFS order between a given arrival schedule. For example, departures and runway crossings on runway 18C at CLT are inserted in gaps between the arrivals on runway 18C and in gaps between the arrivals on runway 23, which converges on runway 18C. The scheduler does not change the given arrival schedule where arrivals are assumed to land at their estimated landing times provided by SOSS. In addition to the landing times, the scheduler takes as input the takeoff and runway crossing ready times for each departure and crossing (respectively) in the schedule horizon, and the required time separation between successive operations. The algorithm uses the separation times and the unimpeded transit times provided by SOSS to match the simulation.

The algorithm first ranks the departures and runway crossings in the scheduling horizon according to their ready time, to takeoff or to cross the runway respectively. Then it applies the following steps to each flight according to this FCFS order:

1. Find the gaps between successive arrivals within which the ready time (to takeoff or to cross) lies. For example, for runway 18C, there may be two gaps, one between two successive arrivals on the same runway 18C and one between two successive arrivals on the dependent runway 23.

2. Find the separation required behind the leading arrival of each of the gaps. Identify the maximum of the leading arrival times plus the required separations as an upper limit on the desired schedule.

3. Find the departures and runway crossings that were already scheduled within the arrival gaps, if any. Identify the required separation behind the last departure operation and behind the last crossing. Identify the maximum of their scheduled times plus their corresponding required separations as another upper limit on the desired schedule.

4. Compare the limiting times computed in steps 2 and 3 and set the candidate schedule to the maximum of the two.

5. Find the required separation before the trailing arrival of each arrival gap and identify the minimum of the trailing arrival times minus the required separations as a lower limit on the desired schedule.

6. If the candidate time computed in step 4 violates the lower limit computed in step 5, identify the following arrival gaps and repeat steps 1 through 6 until a gap is found where the lower limit is not violated. Once found set the operation schedule at that time.

Note 1: The algorithm described is not guaranteed to produce an optimal schedule in terms of maximum throughput or minimum delay because of two reasons: (1) The algorithm schedules flights in a FCFS sequence and a flight is not visited more than once. (2) The algorithm ignores the triangular inequality which is characteristic of the separation requirements. If three flights A, B, and C are operated successively, the triangular inequality implies that the separation required between A and C is larger than the sum of the separations required between A and B and between B and C. In this case, ensuring that C is separated from its immediately leading flight B is not sufficient to ensure that it is separated from the preceding flight A. The algorithm described above only tests the separation from the immediately leading flight and the immediately trailing flight.

Note 2: The algorithm represents nominal operations. It is possible to space the flights by more than the minimum requirements if the capacity is reduced below nominal, by imposing a rate limit within a sliding time window.

4.5 Analysis and Results

The fast time simulation analysis was applied to a four hour period on March 11, 2016 at CLT, a busy period with 199 departures, 175 arrivals and considerable queuing. CLT operated in the south configuration landing runway 18L, 18C and 23 and departing runways 18C and 18L. Landings on runway 18R cross runway 18C on the way to the ramp. In the following subsections, we present a series of analyses using the scenarios outlined in Table 4-1.

Table 4-1. Analysis Scenarios

Scenario	Run 1	Run 2	Run 3	Run 4
Metering	Off	On	On	On
Gate release buffer	Off	Off	2 min	3 min
Queue buffer (18C/18L)	Off	15/12	15/12	15/12
Frequency/horizon	Off	5/60 min	5/60 min	5/60 min

Run 1 is a baseline run of the SOSS simulation without the scheduler. Run 2 applied metering without a limit on gate blocking: the departures metering delay was not limited by a gate conflict with an arrival. Runs 3 and 4 applied metering with the gate blocking limit turned on with the departure that had a gate conflict with an arrival released two and three minutes, respectively, prior to the estimated time of arrival at the gate. All the metering cases were applied with a queue threshold of 15 flights for runway 18C and 12 flights for runway 18L. All runs with scheduling applied the scheduler with a frequency of five minutes and a horizon of one hour.

The performance of the metering control strategies were measured using the following metrics highlighting the tradeoff between efficiency and throughput:

1. The reduction in the congestion level due to metering measures increased efficiency.
2. The delay that was absorbed at the gate due to metering, which corresponds to the reduction in congestion, also measures increased efficiency.
3. The change in the flight takeoff time due to metering measures the overall delay and the conformance to the given runway schedule computed by the tactical scheduler. Hence it also measures the impact on runway throughput.
4. The change in the taxi in time representing the effect of gate blocking.

4.5.1 Scheduler Queue Prediction Error

Figure 4-1 shows the prediction error of the scheduler by comparing the takeoff time of each flight in the simulation (on the horizontal axis) with the takeoff time that was predicted by the scheduler (plotted on the vertical axis) over the one hour time horizon. The plot shows the prediction performance in the first scheduler cycle where no control was applied yet to the simulation. Hence the simulation takeoff times are taken from run 1 (with no metering) and the prediction times are taken from any one of the other runs with metering (In this case run 2).

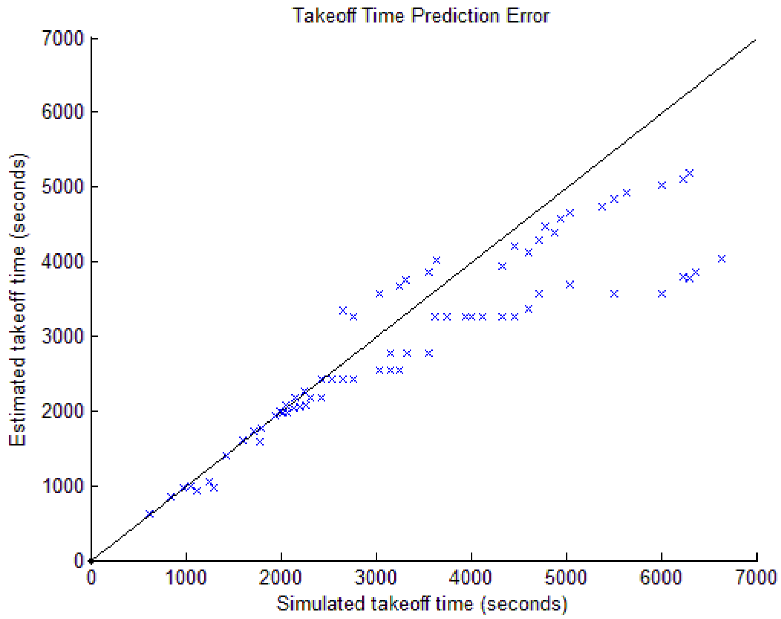


Figure 4-1. Simulated versus predicted takeoff times

Figure 4-1 shows that the prediction error is small initially (the simulation and prediction times are equal lying on the 45 degree line) then grows gradually (as the data points deviate from the 45 degree line). This is expected as the prediction error increases over time. It also shows that the predicted takeoff times tend to be earlier than the simulated time, indicating that the scheduler is using a higher runway service rate or smaller unimpeded transit times, or a combination of the two, relative to the simulation. While the scheduler is using the unimpeded transit times and the separation requirements that the SOSS simulation was using, the interactions between flights on the surface are not predicted by the scheduler. Since these interactions tend to introduce delay, it is reasonable to expect that the scheduler predicted takeoff time contain smaller delays than the simulated takeoff times.

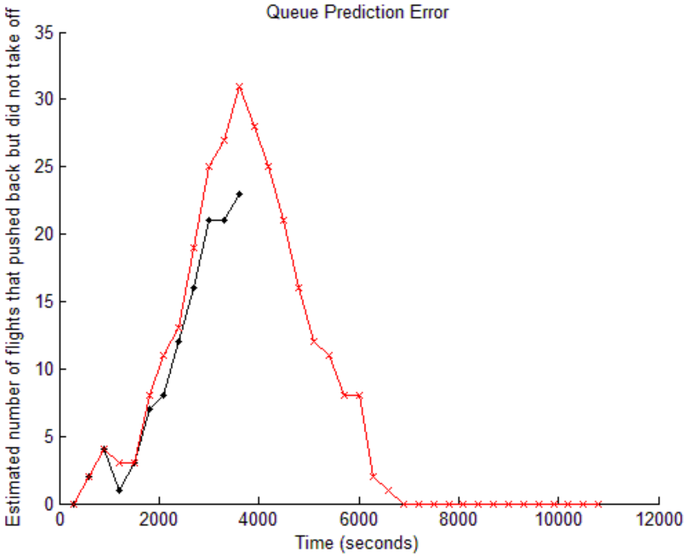


Figure 4-2. Predicted versus simulation queue over one hour prediction horizon

Figure 4-2 Shows the corresponding error in the prediction of the queue: the number of flights that pushed back but did not takeoff at each time. The figure shows again the prediction in the first cycle over one hour. As expected, the predicted queue is smaller than the simulated queue because the predicted takeoff times are earlier than the simulated takeoff times as was shown in Figure 4-1.

4.5.2 Impact of Metering on Congestion

Figure 4-3 shows the effect of the metering scheduler on controlling the queue: the number of flights that pushed back but did not takeoff. This number is plotted on the vertical axis over time on the horizontal axis for the full four hour duration for the baseline run 1 (without metering) and run 2 (metering without gate blocking limit).

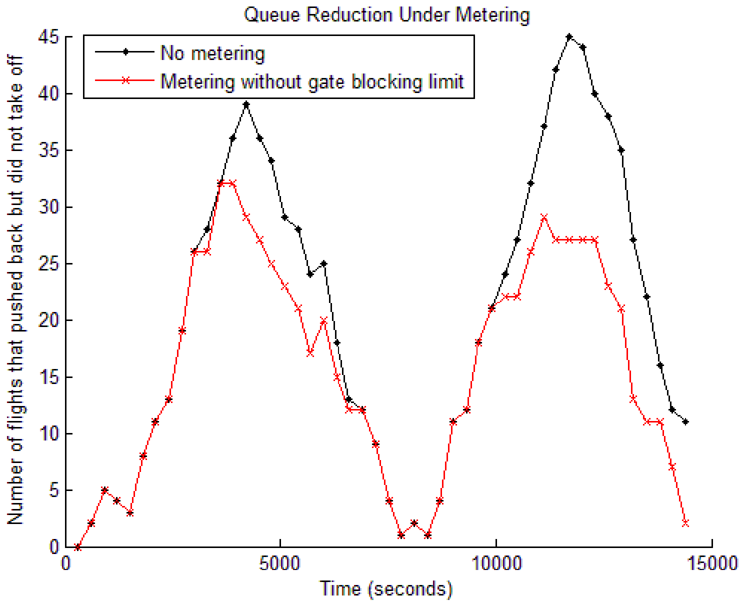


Figure 4-3. Metering impact on the number of flights that pushed back but did not takeoff

Figure 4-3 shows that the queue is effectively controlled to below the threshold value of 27 flights (15 for runway 18C plus 12 for runway 18L) most of the time. One limitation to the control is the frequency of the scheduler at 5 minutes. The control could be improved at higher frequencies (or with the addition of the tactical scheduler which implements the gate release times with a higher frequency). Future research may include additional variations on the frequency and horizon of the scheduler and different levels of uncertainty in the simulation.

As shown in Table 4-2, comparing the first two rows for runs 1 and 2, the takeoff delay is similar between the two runs at about 19 minutes per flight on average. This indicates that the queue thresholds used in run 2 were sufficient to impose enough pressure on the runway to maintain throughput.

4.5.3 Tradeoff between Gate Blocking and Metering

Figure 4-4 shown the impact of adding the gate blocking limit to the metering in terms of the resulting queue size. In addition to the two curves in Figure 4-3, the figure includes the queue size in run 4 where metering was applied with a gate release buffer of 3 minutes when a gate conflict exists between a metered departure and an arrival. The figure shows that the metering was reduced in this case (the queue was increased) relative the metering without gate blocking limit (run 2). This reduction in metering is because of the early release of some of the metered departures that blocked an arrival.

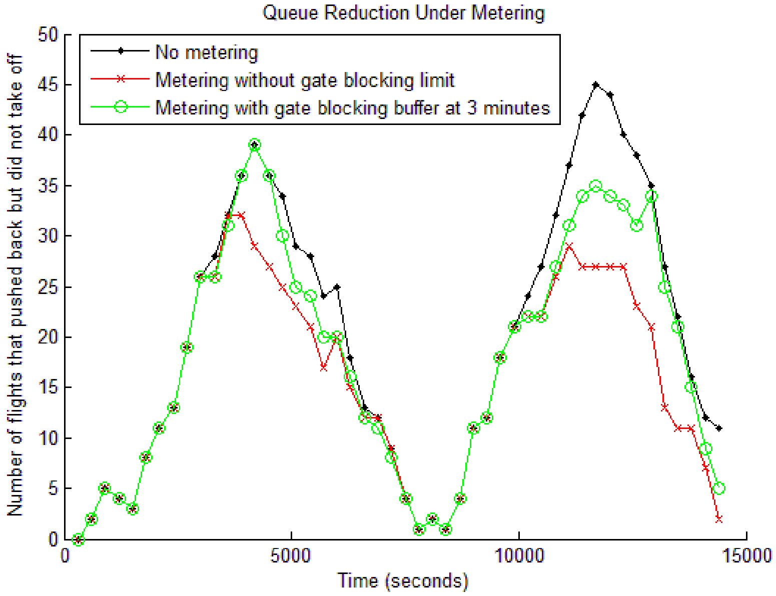


Figure 4-4. Impact of metering with gate blocking limit on congestion

Table 4-2 compares all the runs in terms of the average and total gate delay, takeoff delay and taxi in delay. It also shows the number of metered flights and the number of blocked arrivals in each run. Few observations are made from the table:

Table 4-2. Tradeoff of metering and gate blocking effects

	Average Gate Delay (Metered /All)	Total Gate Delay	Average Takeoff Delay	Total Takeoff Delay	Average Taxi-in Delay	Total Taxi-in Delay	Number Metered Flights	Number Blocked Flights
No metering	0	0	19.3	3835	1.7	293	0	0
Metering w/o gate blocking limit	8.4/3.25	588	18.9	3413	4.2	701	70	11
Metering with gate buffer 120	6.7/2.8	545	19.7	3826	2.2	391	86	1
Metering with gate buffer 180	5.9/2.5	475	19.6	3776	1.6	281	86	1

1. All of the runs maintain similar throughput as the overall delay of the takeoff time remains at about 19 minutes per flight. The deviation of the takeoff time delay is within 40 seconds of the baseline run without any metering.

2. The tradeoff between metering delay at the gate and the blocking effect in terms of taxi in delay is clear among the runs. The metering delay was reduced from 8.4 minutes per metered flight without

limiting the gate blocking to 6.7 minutes per flight with the gate blocking limit set at two minutes before an estimated arrival time, to 5.9 minutes per metered flights when departure are released three minutes prior to an estimated arrival time in gate conflict. Correspondingly, the taxi in time was reduced from 4.2 minutes per flight, to 2.2 minutes per flight, to 1.6 minutes per flight, respectively.

3. The average taxi in delay in the baseline without metering (run 1 in the first row) was 1.7 minutes per flight. This is similar to the average taxi in delay of metering with the gate blocking limit using the three minute gate release buffer between a departure and an estimated arrival at the blocked gate. Therefore, limiting gate blocking with a release buffer of three minutes between a departure and an estimated arrival at a blocked gate is sufficient to eliminate the gate blocking effect on arrivals. With this limit, a significant amount of metering at the gate, of 5.9 minutes per metered departure (2.5 minutes over all departures) can be achieved.

4.6 Conclusions

We presented a simulation-based analysis that compared the performance of different strategies for avoiding gate blocking in departure metering, as part of an integrated departure scheduling concept. The performance was compared in terms of the ability of the different strategies to conform to maintiain runway throughput and minimize the impact on arrival delay while transferring queuing delays from the airport movement area to the gate to save on fuel burn and emissions.

Future research will include additional extensions on the presented analyses such as:

1. Variations on the strategies of avoiding the gate blocking, for example, different assumptions about the scheduler knowledge about the gate assignment and flexibility in changing it.
2. The effects of uncertainties in the arrival times representing potential gaming schemes by the operators to avoid metering delays.
3. Variations on the scheduler frequency and horizon under different uncertainties in SOSS's transit and separation times.
4. Different queue threshold values.

5 Integrated Airspace-Surface Scheduling and its Effect on Miles-in-Trail Restrictions Relaxation

Current-day metroplex traffic management practices lead to multiple operational shortfalls causing unnecessary surface and airspace delays, underutilization of available metroplex capacity and lack of predictability. Under the ATM Technology Demonstration-2 (ATD-2) subproject, NASA plans to address these shortfalls by demonstrating Integrated Arrival, Departure, Surface (IADS) scheduling technologies and transitioning them for field-implementation. These technologies aim to increase the predictability, efficiency, and throughput of metroplex operations while meeting future air traffic demands. An important research question is how would these new technologies interact with current-day traffic flow and traffic merge management techniques such as the implementation of departure miles-in-trail (MIT) restrictions.

This Chapter studies the interaction of departure MIT restrictions with air traffic operations managed by an ATD-2 integrated airspace-surface scheduler, using fast time simulations. This Chapter also discusses a metroplex departure metering simulation platform developed for supporting this study. Simulation-based analysis demonstrated that maintaining MIT restrictions at current-day levels while ATD-2 is in operation, may impede the full realization of the benefits from ATD-2. Our results also demonstrated that relaxing MITs when ATD-2 scheduling is active would save around 1-3% total departure delay (gate + taxi + airborne delay) while retaining a high level of taxi and airborne delay savings over current-day operations, as well as maintaining a level of safety commensurate with current-day operations.

5.1 Introduction

The combined transit of departure flights from the airport surface, through the terminal airspace and merging into overhead enroute air traffic streams is a major source of delay in the National Airspace System (NAS) [11]. This is especially true in metroplex regions where departures and arrivals from/to major and secondary airports compete for limited resources (e.g., mixed-use runways, shared departure-fixes, gaps in busy overhead traffic streams). Current-day metroplex traffic management practices lead to multiple operational shortfalls in this flight domain: (i) Identical ticketed departure times, a pushback-when-ready operational paradigm, and reactive first-come-first-served (FCFS) management of clearances at spots, lead to inefficient departure sequences causing takeoff delays, taxi inefficiency (stop-and-go traffic) and throughput loss; (ii) Lack of predictability in the departure process leads to large variances in TRACON departure loads and forces traffic management coordinators to impose buffers (e.g., runway separation buffers) to ensure safety; and forces ARTCCs/TRACONs to impose inefficient departure restrictions (excess miles-in-trail (MIT), or approval requests, (APREQs)) on airports to make space for airborne merging; (iii) Lack of predictability in the gate pushback to runway takeoff process also causes airlines to set excess block times, which limits fleet utilization and increases personnel and fuel costs.

Under the ATM Technology Demonstration-2 (ATD-2) subproject, NASA plans to address these shortfalls by demonstrating Integrated Arrival, Departure, Surface (IADS) scheduling technologies and transitioning them for field-implementation. These technologies aim to increase the predictability, efficiency, and throughput of metroplex operations while meeting future air traffic demands [30]. The operational environment for the ATD-2 system consists of a local metroplex airspace overlying one or more well-equipped airports and multiple less-equipped airports. Departures from these airports may share departure fixes on the TRACON boundary and merge into busy enroute traffic streams in the ARTCC airspace. Departures are subject to multiple restrictions including MITs at enroute merge points and departure fixes; Expected Departure Clearance Times (EDCTs) from Ground Delay Programs (GDPs); Weather-related departure fix/gate closures; and Takeoff time restrictions due to arrival metering constraints (i.e., Time Based Flow Management, TBFM, allocated landing time-slots) at a destination airport. The ATD-2 system will compute time-based departure schedules for all airports in the local metroplex while accounting for all departure restrictions. An important research question is how ATD-2 scheduler-managed operations will interact with current-day departure-merge management strategies such as the imposition of MIT restrictions.

The research work documented in this report, including this Chapter, supports NASA's ATD-2 research. The overarching aim of our ongoing research is to examine different distributed scheduling schemes for improving the performance and robustness of IADS scheduling. Research presented in this Chapter supports the overarching aim by analyzing one key factor that may impede the full realization of benefits from ATD-2, and by suggesting procedural changes to mitigate the negative impacts of this key factor. The key factor we study here is the impact of departure MIT restrictions on the performance of the ATD-2 system. When MIT restrictions are imposed on departures, the impacted origin airport is required to depart the affected traffic with sufficient in-trail separation at runway takeoff, so that by the time the departures reach the impacted fix, they are in compliance with the MIT spacing requirement, without requiring significant delay maneuvering in the TRACON. Spacing increases away from the takeoff runway due to de-compression. So, in-trail spacing requirement at the runway is smaller than the MIT restriction size, however it is still more than the absolute minimum spacing requirement that the local controller would impose at runway takeoff if MITs were inactive.

In current day operations, ARTCC Traffic Management Unit (TMU) usually applies excess MIT restrictions on individual airport departure flows in order to provide the TRACON or ARTCC controllers with some leeway for merging these flights when they get closer to the departure fix or to the enroute merge fix. In future operations with ATD-2 departure metering, departure flows will become more predictable, departure sequences will become more optimal (occurrences of consecutive departures going to the same departure fix will be reduced) and takeoff times will be better coordinated with departure-fix merge time-slots as well as overhead enroute traffic stream gaps. As a result, the requirement for additional MIT spacing at runway takeoff may be relaxed. This relaxation may provide additional efficiency, throughput, and workload benefits, while still enabling sufficient spacing at the constrained departure fix or enroute merge fix for easy merging.

This Chapter aims to evaluate whether MIT constraints can be relaxed when ATD-2 is in operation without impairing the safety of operations and to quantify the additional benefit that can be provided by relaxing MIT constraints in this situation. We use fast-time simulations for our evaluations. The Chapter is structured as follows: Section 5.2 provides the details of the fast-time simulation based evaluation method we used in our work. This includes (i) details of the track data analysis we conducted to obtain a backbone departure route scheduling structure for modeling key locations where departure flows from the Charlotte Douglas International Airport (CLT) and CLT's satellite airports merge at the departure fixes, and where they merge with Atlanta Hartsfield International Airport (ATL) departures and into overhead enroute traffic streams, (ii) historical departure restrictions analysis to support modeling of these restrictions in a fast-time simulation, and (iii) description of the metroplex departure metering simulation platform used for fast-time simulations. Section 5.3 presents the results obtained from our simulations. Finally, Section 5.4 outlines the main conclusions drawn from our research work.

5.2 Fast-time Simulation based Method for Analysis of MIT Impact on Departure Metering Performance

Our research approach for assessing the impact of MIT restrictions on the performance of departure metering was to simulate operations in a busy metroplex environment under current-day operational procedures and separately under ATD-2 operational procedures, with varying levels of modeled MIT restrictions. Then, compare key operational performance metrics from the simulations to find out benefits and other impacts caused by relaxation of MIT restrictions. Departure restrictions models used in the simulations, including MIT restrictions and APREQs, were derived from data analysis of historical traffic flow restrictions.

Our research work addresses a known problematic departure management problem in the NAS: CLT airport departures, particularly CLT departures headed for destination airports in the Northeast U.S. (especially New York area airports and Washington D.C. area airports) face multiple capacity constraints in their transit. First, as shown in Figure 5-1, CLT airport surface consists of a large-size ramp area with constraining features such as a single-lane taxi-path near terminals D and E that handles bi-directional arrival and departure traffic. Second, CLT's most commonly-used runway configuration (South-flow, with departures on 18C and 18L, arrivals on 18R, 23, and 18C) involves a capacity-constrained, mixed-use runway 18C where departures have to be fit into gaps between arrivals landing on the same runway as well as arrivals crossing runway 18C after landing on runway 18R. Furthermore, takeoff clearances for 18C departures have to be coordinated with gaps in arrival traffic landing on a virtually intersecting runway (runway 23). After taking off, the CLT departures merge with satellite airport departures before exiting the CLT TRACON via specific departure fixes on the TRACON boundary. Finally, these departures merge into busy enroute traffic streams going to the New York area and Washington D.C. area airports, including departure streams originating from ATL airport.

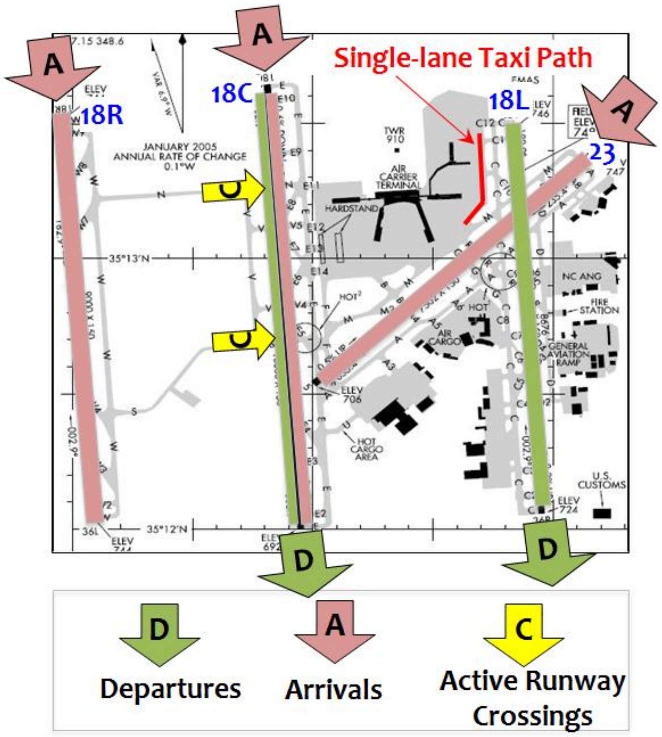


Figure 5-1. Main surface constraints at the CLT airport

Our first step, before developing a CLT and ATL-focused departure metering simulation, was to analyze historical CLT, CLT-satellite and ATL departure tracks to determine the major departure flows and the major constraint points where the CLT departures merge with other traffic during their transit from runway takeoff to enroute stream merge. Section 5.2-5.2.1 discusses the technical approach we used for this departure track analysis and outlines the major findings. Following the departure track analysis, we analyzed historical traffic flow restrictions imposed on CLT and ATL departures. The purpose of this analysis was two-fold: (i) to develop a model of departure restrictions to feed the CLT-ALT departure metering simulation model and (ii) to support selection of an appropriate historical day for generating simulation scenarios. Section 5.2-5.2.2 describes this historical analysis of traffic flow restrictions. Following the departure tracks and departure restrictions analysis we developed an air traffic simulation model for simulating departure metering operations at the CLT airport. The simulation included modeling of (i) departure and arrival operations on the CLT airport surface, (ii) departure operations in the CLT TRACON (including departures taking off from smaller satellite airports within the TRACON), and (iii) CLT/CLT satellite departures merging into overhead enroute traffic streams in the Atlanta ARTCC (ZTL) airspace. Our simulation also included a model of ATL airport departures taking off from ATL and merging with CLT/CLT satellite airport departures in ARTCC airspace. The simulation focused primarily on departure traffic headed to destination airports in the Northeast U.S. Section 5.2-5.2.3 describes the simulation model in detail. Sections 5.2-5.2.1, 5.2-5.2.2, and 5.2-5.2.3 together describe our simulation-based method for evaluating MIT impact on departure metering performance.

5.2.1 Identifying Major Departure Flows and Merge Locations

As mentioned above, our first step was to analyze historical departure track data from CLT, CLT-satellite airports and ATL—especially departures going to destination airports in the Northeast U.S. The purpose of this step was to identify the major departure flows and major merge/constraint points. The identified departure flows and merge-points would inform the simulation model development by providing a backbone scheduling network to base the departure restrictions and departure metering models on. For example, the identified merge-locations where CLT departures merge with ATL departures and overhead enroute traffic streams are used as key scheduling points when allocating overhead traffic stream time-slots to APREQ-impacted CLT and ATL departures, as we describe in Section 5.2-5.2.3.

We used surveillance track data from the Performance Data Analysis and Reporting System (PDARS) for three days in the summer of 2015 for the departure flow analysis task. (Summer is the time when departure restrictions are most prevalent.) PDARS is a fully integrated performance measurement tool designed to help the FAA improve NAS safety and efficiency. PDARS fuses air traffic control automation and radar data from nation-wide ARTCCs and regional TRACONs with other flight and environmental data into multi-dimensional flight tracks that cover NAS-wide IFR traffic in the U.S. from end-to-end. The task started with merging of track data from multiple FAA facilities involved. These included the CLT and ATL airports (ASDE-X surface surveillance track data), the A80 and CLT TRACONs, the ZTL ARTCC, as well as neighboring ARTCCs including ZDC, ZID, ZNY and ZOB.

As mentioned above, we focused on departure flows going to destination airports in the Northeast U.S. Figure 5-2 shows departure tracks from CLT and ATL going to major airports in the Northeast U.S. (New York area airports: JFK, LGA, and EWR; and Washington D.C. area airports: DCA, IAD and BWI). As seen from

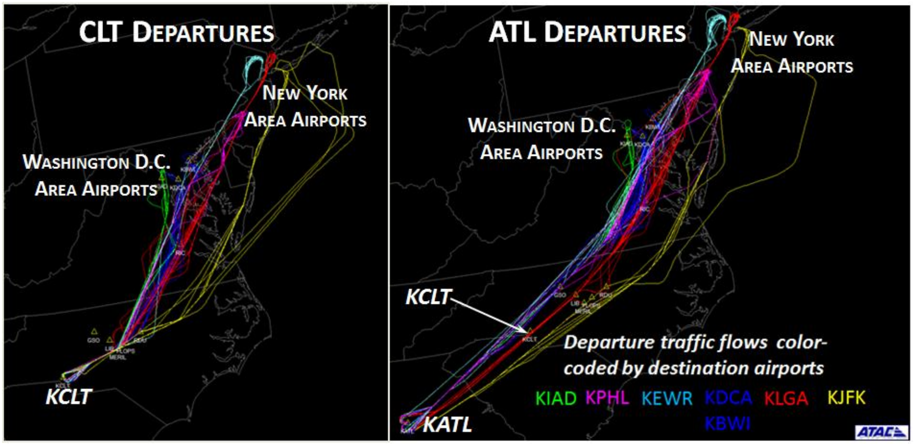

CLT DEPARTURES
NEW YORK AREA AIRPORTS
WASHINGTON D.C. AREA AIRPORTS
KCLT
ATL DEPARTURES
NEW YORK AREA AIRPORTS
WASHINGTON D.C. AREA AIRPORTS
KCLT
Departure traffic flows color-coded by destination airports
KIAD KPHL KEWR KDCA KLGA KJFK
KBWI
KATL
ATAC

Figure 5-2. PDARS tracks for CLT and ATL departures to Northeast airports.

Figure 5-2, the departure flows are more or less separated-out based on the destination airport. In fact, when we looked at the altutide profiles in conjunction with lateral paths, the departure tracks going to the Washington D.C area destination airports were found to be completely laterally and/or vertically separated from departure tracks going to the New York area destination airports. As a result, we analyzed these two sets of tracks separately to examine in more detail the exact location where CLT departures going to each of these two regions merge with ATL departures going to the respective regions.

Figure 5-3 shows CLT and ATL departure tracks going only to the New York area airports JFK, LGA and EWR. It can be seen from this Figure that the tracks going to the three airports are fairly separated from each other laterally. LGA bound departure flows from CLT and ATL appear to merge near the region surrounding the ZITTO waypoint; EWR bound departure flows merge in the vicinity of FAK VORTAC, and JFK bound departure flows merge near TYI. We further zoomed onto the identified region of potential merge location to assess the exact location of the merge. As shown in Figure 19 for EWR bound departures, we saw that the departures indeed merged exactly at the FAK VORTAC. We also looked at the altitude profiles of the departures to confirm that the merge indeed occurred at the identified waypoint.

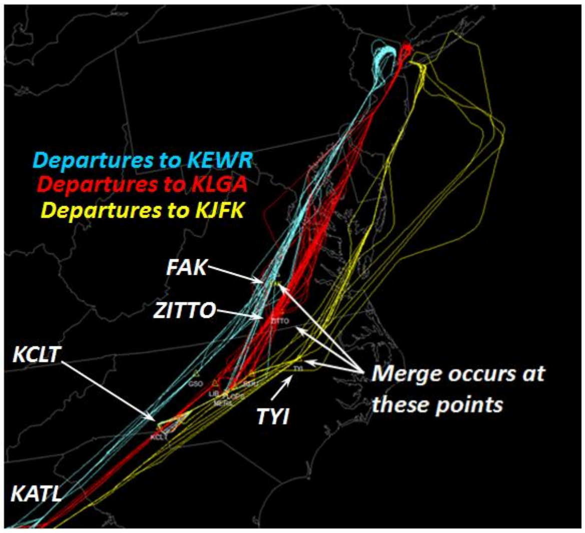
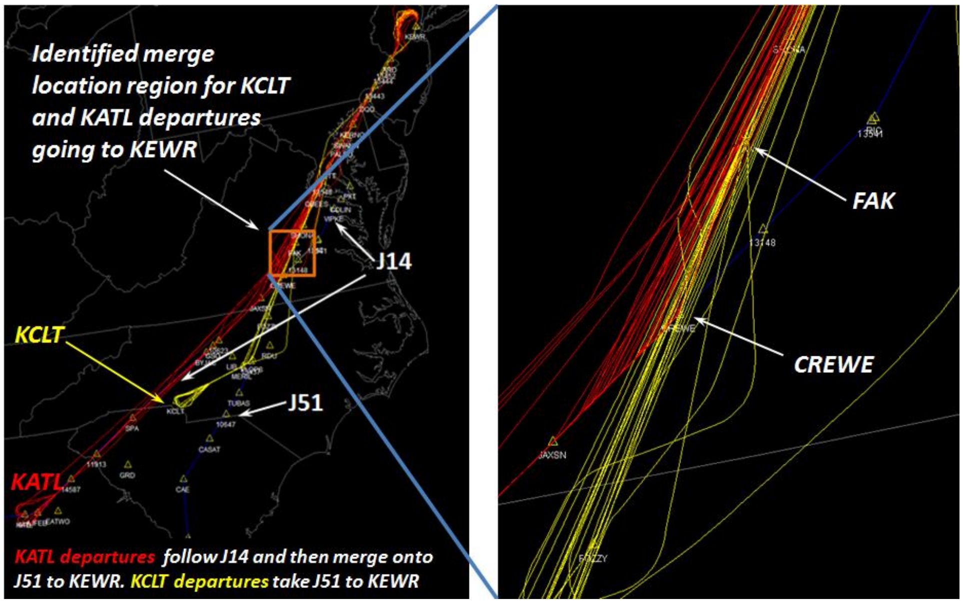


Figure 5-3. Merge locations for CLT and ATL departures going to New York area airports.

For departures going to Washington D.C. area destination airports, a clear merge in both lateral and vertical directions was not identied to occur until the flights are in the process of descending to their respective destination airports near their entry to the Potomac TRACON.

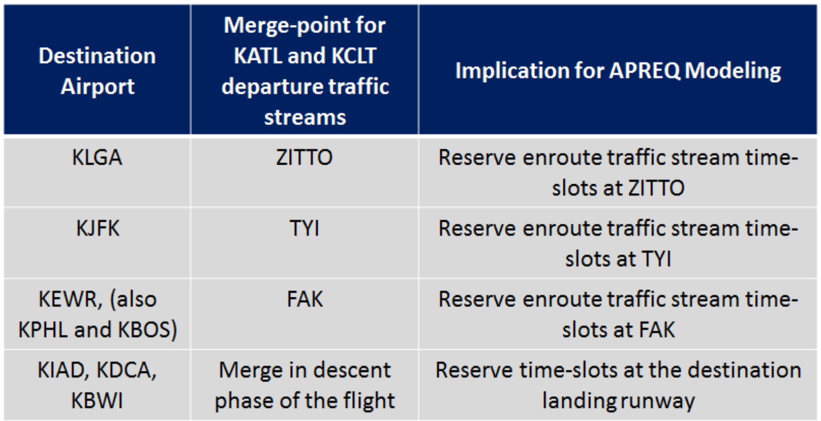

Identified merge location region for KCLT and KATL departures going to KEWR

KATL departures follow J14 and then merge onto J51 to KEWR. KCLT departures take J51 to KEWR

Figure 5-4. Identifying exact merge-locations for CLT and ATL departures

Table 5-1 shows the summary of findings from this track data analysis task and also outlines the implications of the identified merge-locations for departure restrictions (i.e., APREQ) modeling in our simulation model. In the case of LGA, EWR, and JFK-bound

Table 5-1. Identified merge-points for CLT and ATL departure streams



Destination Airport	Merge-point for KATL and KCLT departure traffic streams	Implication for APREQ Modeling
KLGA	ZITTO	Reserve enroute traffic stream time-slots at ZITTO
KJFK	TYI	Reserve enroute traffic stream time-slots at TYI
KEWR, (also KPHL and KBOS)	FAK	Reserve enroute traffic stream time-slots at FAK
KIAD, KDCA, KBWI	Merge in descent phase of the flight	Reserve time-slots at the destination landing runway

departure flows where we were able to identify a specific enroute waypoint as the location of merge, we apply APREQ modeling by reserving time-slots at the respective merge-locations. For example, (as we will explain in more detail in Section 5.2-5.2.3) LGA-bound departures from CLT and ATL, when impacted by APREQ restrictions, will receive runway takeoff time constraints based on their allocation to time-slots at the ZITTO VORTAC. In the case of DCA, IAD, and BWI-bound departure flows we identified a merge occurring in the descent phase of the flight. So, DCA, IAD, and BWI-bound departures when impacted by APREQs will receive runway takeoff time restrictions based on allocation of time-slots at the respective destination airport landing runway.

Next, we describe our historical departure flow restrictions analysis.

5.2.2 Historical Departure Restrictions Analysis

We conducted an analysis of historical departure restrictions, with two purposes in mind. One, was to support accurate and realistic modeling of these restrictions in the departure metering simulation platform. The second was to select a historical day with significant occurrence of departure restrictions, for generating simulation scenarios.

Our analysis looked at two types of departure restrictions—MITs and APREQs—imposed on CLT and ATL departures. We analysed one year worth of National Traffic Management Log (NTML) data for the year 2015 to support this analysis. The NTML data provides a single system for automated coordination, logging, and communication of traffic management initiatives (TMIs) throughout the NAS. NTML is a part of the Traffic Flow Management System (TFMS). ARTCC Traffic Management Units (TMUs) as well as ATC System Command Center (ATCSCC) traffic managers enter new TMIs and update existing TMIs via a graphical TFMS tool [31]. These entries are converted into database entries which are stored to aid TFM decision making and post operations analysis. More pertinent to the topic at hand, the NTML data contains a record of historically implemented MIT and APREQ departure restrictions (along with many other types of TMIs) including the times during which the restriction was active, the requesting and providing FAA facility, size of the restriction, and other relevant information such as which departure flows are impacted by the restriction. The data is stored as a series of TMI restriction records. Each record includes information about a new restriction imposed by a FAA facility or it could be an update to an existing restriction. Since one of our aims was to select a day with extended occurrence (e.g., number, duration and size of restrictions) and severe impact (e.g., departure delay impact) of MIT and APREQ restrictions on CLT and ATL departure flows going to the Northeast destination airports, we developed two normalized measures to help us with our analysis—a Normalized Departure Restriction Severity Score and a Normalized Departure Delay Score. Both these scores are computed for CLT and ATL per day.

Daily Normalized Departure Restriction Severity Score. This is measure of how severely an airport was impacted by MIT and APREQ departure restrictions on a particular day. The computation of this score starts by accessing NTML data records for one day at a time. The per day records are first filtered using multiple criteria to a smaller set of records containing features more relevant to the problem at hand. For our purpose the filtering criteria consisted of the following: (i) restriction is provided by the airport under consideration (CLT or ATL), (ii) the constrained NAS element causing the restriction is relevant to the problem we are focusing on, e.g., in this case, NAS elements of interest are constrained Northeast destination airports or enroute waypoints/sectors through which departure flows from ATL/CLT to the Northeast airports travel, and (iii) the restriction type is MIT or APREQ.

After the relevant NTML records are filtered, we next apply complex processing to assess whether each subsequent record represents a new restriction or if it represents an extension or update to an existing (already active) restriction. Identification of new or existing restriction is not straightforward because the restriction records originate from human controllers' input into a graphical tool and this human input process creates multiple discrepancies and peculiar features within the data records.

After this processing, records belonging to each individual restriction are merged and the real start and end times for each restriction are computed. For APREQ records the severity score is formed by simply summing the active durations of all APREQ records during the day. For MIT records the severity score is formed by multiplying the restriction duration with the respective MIT size, and summing this product over all the MIT restrictions active during the day. The departure restriction severity score for a particular day is computed by taking a weighted sum of the respective APREQ and MIT severity scores. After computing individual day scores for the entire year, all the scores are divided by the maximum score to obtain a normalized score (varying between 0 and 1).

Daily Normalized Departure Delay Score. This is a measure of the magnitude of departure delays experienced by an airport on a particular day. We used the FAA's Aviation System Performance Metrics (ASPM) database for computing this metric. In particular, we used the 'all flights' report provided by the ASPM access website. This report contains data-fields including the average gate delay, average taxi-out delay, number of scheduled departures, percent on-time gate departures and percent on-time takeoffs, for each 15-minute time-bin during a day. We combine all these metrics to compute the average departure delay (gate delay plus taxi out delay) over the entire day. The average departure delay is used as the departure delay score for each day. We compute the score for all days in the year and then divide all scores by the maximum departure delay to obtain a normalized departure delay score.

One purpose of this data analysis was to select a day with both, extensive presence of departure restrictions and occurrence of severe departure delays as a result of those restrictions. To aid with this selection, we plotted all the days in 2015 on a scatter plot with the X-axis depicting the daily Normalized Departure Restrictions Severity Score and the Y-axis depicting the daily Normalized Departure Delay Score. Figure 5-5 shows this scatter plot.

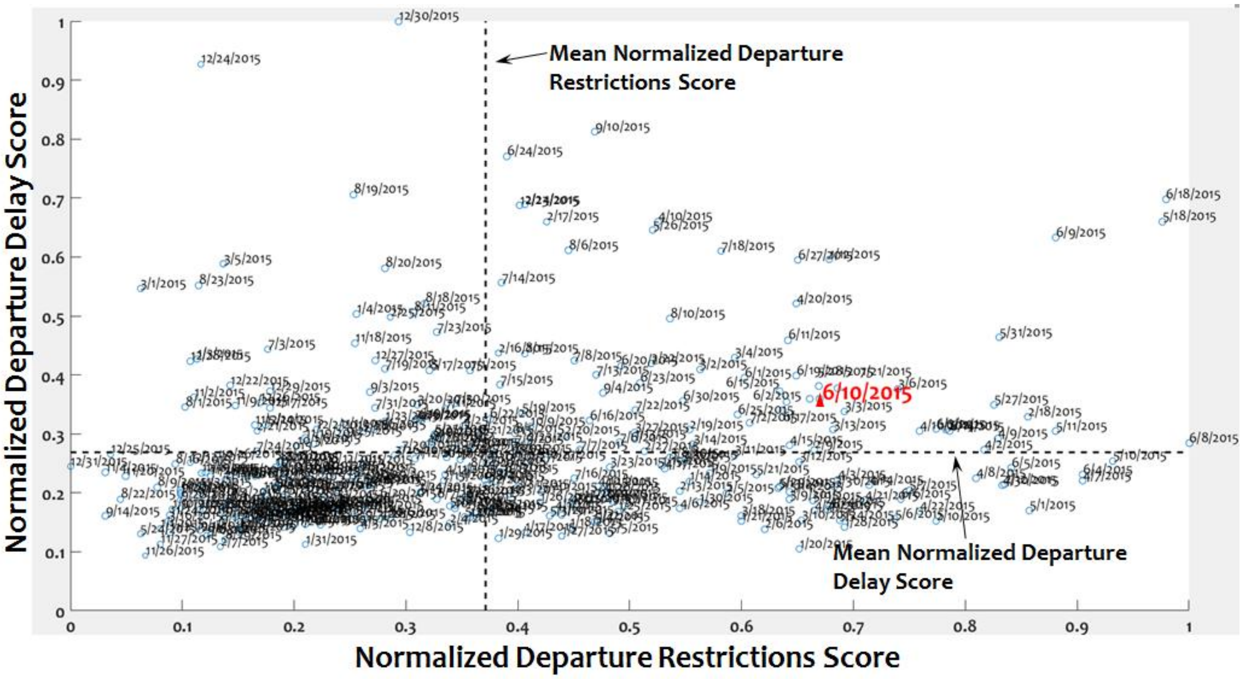


Figure 5-5. Categorization of days by severity of departure restrictions and departure delays

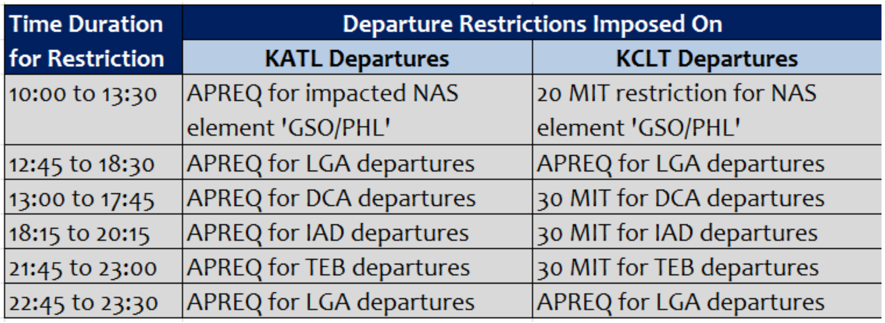
As expected, we see a roughly linear relationship between the daily departure restrictions score and the daily departure delay scores. But, we also see a lot of variance in the departure delay scores at similar levels of departure restriction scores. One reason for this could be delays caused by other factors besides departure restrictions. Moreover, we are only computing the departure restrictions score based on restrictions for departures going to Northeast airports, but the delay scores are computed by aggregating over ALL departures.

In Figure 5-5, days falling in the top right quadrant experienced higher than average severity of both, departure restrictions and departure delays. We further analyzed various features of departure restrictions (especially, what caused them) for the days that fell in the top right quadrant. Many of these days experienced departure restrictions caused by convective weather. In our research, we wanted to focus on days where all or a large majority of the departure restrictions were caused by volume related constraints rather than convective weather related constraints. This is because convective weather, we observed, tends to cause other types of TMIs such as Ground Delay Programs (GDPs) or Ground Stops (GSs) to occur in addition to MIT and APREQ departure restrictions. Delay-inducing effects of GDPs/GSs contaminate the analysis of the impact of MIT/APREQ departure restrictions. As a result, with a view to keep simulation results “clean”, we avoided days with convective weather constraints. By this criterion, 06/10/2015 was observed to be an ideal candidate for the target simulation day (shown in red in Figure 5-5).

Table 5-2 shows the departure restrictions that were active at ATL and CLT on 06/10/2015. All these restrictions were volume related. Separately, this restrictions data analysis helped us discover a major difference in the implementation of departure restrictions for ATL versus CLT. As seen from Table 5-2, ATL mostly received APREQ departure restrictions for specific destination airport-bound traffic, while CLT received MIT restrictions for departures headed to the same destination airports (with the exception of LGA). This trend was found to occur across multiple days and seems to be a standard procedure for handling enroute merge constraint involving departures from these two airports.

As we discuss in the next section, we modeled the departure restrictions outlined in Table 5-2 in the simulation platform. We describe the simulation model in the next section.

Table 5-2. CLT and ATL departure restrictions active on 6/10/2015



Time Duration	Departure Restrictions Imposed On	
for Restriction	KATL Departures	KCLT Departures
10:00 to 13:30	APREQ for impacted NAS element 'GSO/PHL'	20 MIT restriction for NAS element 'GSO/PHL'
12:45 to 18:30	APREQ for LGA departures	APREQ for LGA departures
13:00 to 17:45	APREQ for DCA departures	30 MIT for DCA departures
18:15 to 20:15	APREQ for IAD departures	30 MIT for IAD departures
21:45 to 23:00	APREQ for TEB departures	30 MIT for TEB departures
22:45 to 23:30	APREQ for LGA departures	APREQ for LGA departures

5.2.3 Metroplex Departure Metering Simulation Platform

The fast-time simulation platform we developed to support our research work, provides a capability for simulating airport surface, terminal airspace, and enroute air traffic over a prescribed time-horizon, for a metroplex-wide scope. Simulated traffic includes departures originating at the selected major airport or airports and departures from neighboring satellite airports that may interact with the major airport departures, as well as overhead enroute traffic that merges with the major metroplex departure flows in the ARTCC airspace. Simulation for departures starts at the terminal gates and ends at the enroute traffic stream merge point. Overhead enroute traffic is simulated by modeling reserved time-slots at the enroute merge-fix where metroplex airport departures merge with the overhead traffic stream. The simulation components are designed to evaluate the operational impacts and benefit mechanisms of ATD-2 decision support tools in comparison with current-day departure operations under varying levels of departure restrictions. Key components of the fast-time simulation were described in detail in our previous publication [26]. We have made some enhancements to the models described in that publication. Here, we discuss the important features, including new enhancements, for these fast-time simulation components in brief. The key simulation components are (i) link-node models that represent airport surface and airspace routes, (ii) transit time models for the links within the link-node network, and (iii) queue control methodology used for simulating the motion of aircraft over the link-node network.

Link-Node Models for Airport and Airspace Routes. We used links and nodes to model departure flows and key merge-locations obtained from the historical departure track data analysis described in Section 5.2-5.2.1. Nodes are located at the terminal gate areas (groups of gates in the same geographical region of the airport), departure runways, departure fixes (metering fixes at the TRACON boundary), and ARTCC airspace merge fixes (merge-points for the overhead en route traffic streams). The link-node models allow us to simulate traffic behavior accurately by allocating the correct ramp-area, taxi-path, runway-use, departure-fix use, en route stream merge-point use, and accurate inter-node transit times to the simulated flights.

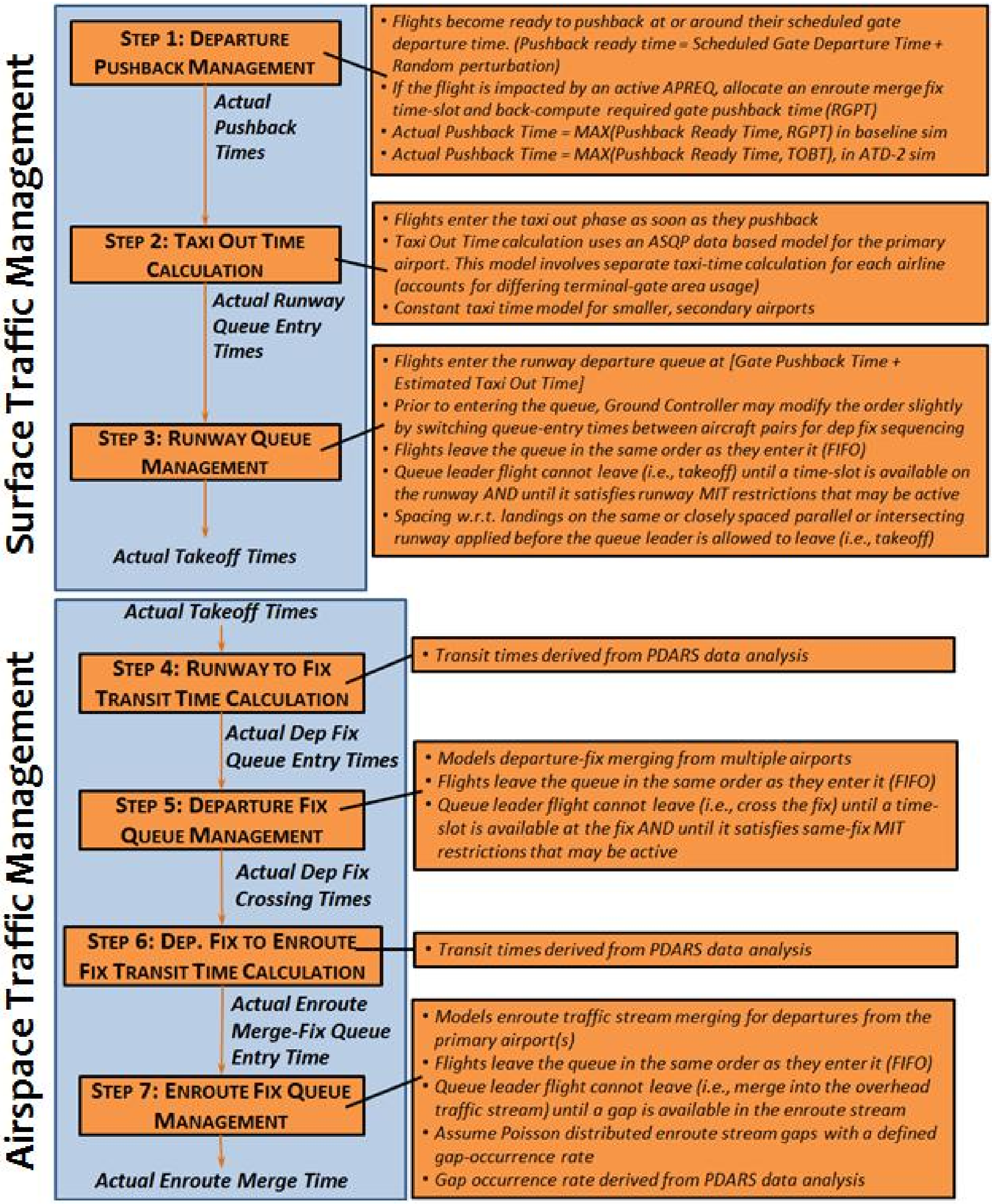
Transit Time Models. Links connecting the nodes are associated with characteristic unimpeded transit times. These transit times are derived from analysis of historical operational data. The typical link-node model for a departure flight’s route consists of one complex node representing a group of gates (where the flight originates), connected to a simple node representing the departure runway; then on to the departure-fix node; and finally from the departure-fix node to the enroute traffic stream merge node. So, the model contains three links—one link connecting the gate-group to the departure runway node, the second link connecting the departure runway node to the departure-fix, and the third connecting the departure fix node to the en route stream merge node.

The link joining each gate-group node to the departure runway node is characterized by a taxi-out time model in our simulation platform. We have developed two taxi-out transit time models of different fidelities. The simpler or lower fidelity model is primarily used for satellite airports that (i) do not have surface surveillance, and (ii) do not report taxi times to the Bureau of Transportation Statistics (BTS) for inclusion in the Airline Service Quality Performance System (ASQP) database. This simple model is a "Fixed" taxi-out time model. In this model, each departure flight from an airport is assigned the same fixed taxi-out time, assumed to be 10 minutes. The higher fidelity taxi-out transit time model is a "10-th Percentile" model. In this model, we first derive a distribution of taxi-out times for that airport, restricting to time-periods when the particular runway configuration that we are modeling was active. We use historical recorded taxi-out time data from the ASQP database to compute distributions of taxi-out times. Along with taxi-out time data, we also collect information on the airline to which each departure aircraft belongs—so that taxi-out distribution can be calculated for each airline separately. The underlying assumption is that since different airlines use gates/terminals in different geographical parts of the airport, their departures will accordingly take longer or shorter times to taxi to the departure runway. In other words, airline is used as a proxy for a specific gate-group. Once an airline-specific distribution is calculated, we pick the 10th percentile taxi-out time from the distribution for the respective gate-group and departure runway pair.

The second link connecting the departure runway to the departure-fix and the third link connecting the departure-fix to the enroute stream merge-point—are characterized by airborne transit time models. The airborne transit times are derived by analyzing PDARS track data, filtered by the appropriate runway, departure-fix and enroute-fix usage. An airborne transit time distribution for each combination of runway to departure-fix to enroute-fix is obtained by analyzing the PDARS data. The mean of this distribution is used to model the typical transit times for the two airborne links.

Queue Control at the Nodes. The algorithm for queue control at key nodes is the heart of the fast-time departure metering simulation platform. Figure 5-6 shows the main steps involved in the departure metering simulation. Each of the major control points in the link-node network (the gate-group, the departure runway, the departure-fix, and the en route stream merge-fix) is associated with a queue. The rest of this section describes how the simulation manages the entry and exit times for individual flights to/ from these queues. The fast-time simulation works in discrete time-steps of one minute. At each time-step, the simulation processes through the seven steps shown in Figure 5-6.

Step 1: Departure Pushback Management. In this step, the simulation first determines how many departure flights will be ready for pushing back at this time-step. When simulating current-day departure operations, the *Pushback Readiness Time* for a flight is computed by adding a random perturbation to the flight's *Scheduled Gate Departure Time*. This models pre-pushback delays. In current-day operations simulation, flights push back when they are ready (i.e., at their *Pushback Readiness Time*) with the execption of flights impacted by APREQ restrictions. Flights impacted by APREQ restrictions may receive gate delays. The simulation platform mimics the current-day APREQ implementation process where departure flights at CLT airport call the tower prior to pushback and wait until the tower obtains clearance from the ARTCC. In the simulation, when the CLT departure flight is ready to push back we predict its estimated enroute merge-fix crossing time using unimpeded transit time estimates for taxi and airborne links in its route. Merge-fix crossing is modeled as fitting flights into 8 nmi-wide merge-slots (see [26] for more detail), some of which are already occupied by overflights or departures from ATL. We allocate the CLT departure flight to the next available enroute stream gap nearest to its estimated enroute stream merge-fix crossing time, i.e., we assume that the flight will eventually enter the enroute stream at this gap. Then, we back-compute the required gate pushback time for "hitting" this enroute stream gap. Any delay that this may cause is absorbed at the gate.


Surface Traffic Management
STEP 1: DEPARTURE PUSHBACK MANAGEMENT
Actual Pushback Times
STEP 2: TAXI OUT TIME CALCULATION
Actual Runway Queue Entry Times
STEP 3: RUNWAY QUEUE MANAGEMENT
Actual Takeoff Times
Airspace Traffic Management
Actual Takeoff Times
STEP 4: RUNWAY TO FIX TRANSIT TIME CALCULATION
Actual Dep Fix Queue Entry Times
STEP 5: DEPARTURE FIX QUEUE MANAGEMENT
Actual Dep Fix Crossing Times
STEP 6: DEP. FIX TO ENROUTE FIX TRANSIT TIME CALCULATION
Actual Enroute Merge-Fix Queue Entry Time
STEP 7: ENROUTE FIX QUEUE MANAGEMENT
Actual Enroute Merge Time

- *Flights become ready to pushback at or around their scheduled gate departure time. (Pushback ready time = Scheduled Gate Departure Time + Random perturbation)*
- *If the flight is impacted by an active APREQ, allocate an enroute merge fix time-slot and back-compute required gate pushback time (RGPT)*
- *Actual Pushback Time = MAX(Pushback Ready Time, RGPT) in baseline sim*
- *Actual Pushback Time = MAX(Pushback Ready Time, TOBT), in ATD-2 sim*

- *Flights enter the taxi out phase as soon as they pushback*
- *Taxi Out Time calculation uses an ASQP data based model for the primary airport. This model involves separate taxi-time calculation for each airline (accounts for differing terminal-gate area usage)*
- *Constant taxi time model for smaller, secondary airports*

- *Flights enter the runway departure queue at [Gate Pushback Time + Estimated Taxi Out Time]*
- *Prior to entering the queue, Ground Controller may modify the order slightly by switching queue-entry times between aircraft pairs for dep fix sequencing*
- *Flights leave the queue in the same order as they enter it (FIFO)*
- *Queue leader flight cannot leave (i.e., takeoff) until a time-slot is available on the runway AND until it satisfies runway MIT restrictions that may be active*
- *Spacing w.r.t. landings on the same or closely spaced parallel or intersecting runway applied before the queue leader is allowed to leave (i.e., takeoff)*

- *Transit times derived from PDARS data analysis*

- *Models departure-fix merging from multiple airports*
- *Flights leave the queue in the same order as they enter it (FIFO)*
- *Queue leader flight cannot leave (i.e., cross the fix) until a time-slot is available at the fix AND until it satisfies same-fix MIT restrictions that may be active*

- *Transit times derived from PDARS data analysis*

- *Models enroute traffic stream merging for departures from the primary airport(s)*
- *Flights leave the queue in the same order as they enter it (FIFO)*
- *Queue leader flight cannot leave (i.e., merge into the overhead traffic stream) until a gap is available in the enroute stream*
- *Assume Poisson distributed enroute stream gaps with a defined gap-occurrence rate*
- *Gap occurrence rate derived from PDARS data analysis*

Figure 5-6. Departure Metering Simulation Steps

While simulating ATD-2 operations, the flight is assumed to be ready for pushback at its *Pushback Readiness Time,* but holds at the gate until the *Target Off Block Time* (*TOBT*). *TOBT* is the required gate pushback time computed by ATD-2 scheduling algorithm. Our prior publication [26] describes our model of the ATD-2 departure scheduling algorithm, which we use to compute *TOBTs*. Note that the scheduling algorithm performs all computations based on the knowledge of *Scheduled Gate Departure Times*, it does not know the exact time when the flight would be ready to push back (i.e., *Pushback Readiness Times*). The next section briefly describes the main features of this scheduling algorithm.

After identifying the flights that are ready for pushback at this time-step, the simulation pushes them back by updating their *actual gate pushback time*.

Step 2: Taxi-out Time Calculation. In this step, the simulation identifies aircraft that have pushed back at the current time-step and updates the *actual runway queue entry times* for these flights using a taxi-out time model. Different taxi-out time (transit time) models used by the simulation platform were discussed in the previous section. It is assumed that the flights will enter the runway departure queue in the first-come first-served (FCFS) order. The FCFS runway queue entry order may be modified if there is an instance of two successive new runway queue entrants with the same allocated departure fix. In such cases, we simulate the sequencing decisions made by Ground Controllers by allowing sequence switches to avoid two successive flights going to the same departure-fix. A sequence switch is allowed only if the runway queue entry times of the flights switching sequence with each other fall within an allowed range of time (we used a 2 minute allowed time range for our simulation platform).

Step 3: Runway Queue Management. In this step, the simulation determines whether the leader of each runway queue can leave the queue (i.e., takeoff from the runway) at this time-step. Two criteria are used to determine if the flight can takeoff or not—(i) Runway minimum separation requirement for safety is satisfied: This constraint is not applied by enforcing an exact required time-separation. Instead, we divide the runway time-line into time-slots, each time-slot being of sufficient length based on a called runway departure rate (obtained from ASPM data). Then, we allow only one departure to takeoff per runway time-slot; (ii) MIT separation requirement for consecutive flights going to the same departure fix is satisfied: To enforce this separation requirement (if a MIT restriction is active for the runway under consideration), the simulation keeps track of the last flight to depart from each runway to each departure-fix. The current leader of the runway queue is not allowed to depart until it satisfies the prescribed MIT separation with the last departure flight to the same fix to takeoff from that runway. MIT restrictions may be specific to certain departure flows, e.g., some restrictions are only applicable to departures going to certain destination airport(s). Our simulation platform enforces such special MIT restrictions also.

If the current leader of the runway queue is eligible for taking off during this time-step (i.e., if it satisfies the above two conditions), then the simulation updates its *actual runway takeoff time*. Then, the simulation evaluates whether the next flight in the runway queue (the new leader of the queue) is eligible for takeoff during this time-step. This process continues until a leader is found that cannot takeoff during this time-step.

Step 4: Runway to Departure Fix Transit Time Computation. At each time-step, the simulation identifies the flights that have taken off during this time-step. For these flights, Step 4 computes and updates their *actual departure-fix queue entry time* by adding the estimated runway-to-fix transit time to the flight's runway takeoff time.

Step 5: Departure Fix Queue Management. In this step, the simulation evaluates whether the leader of the departure-fix queue can leave the queue (i.e., cross the departure-fix) during this time-step. The criterion used to determine if the flight can cross the departure-fix is the minimum separation requirement with respect to the previous flight crossing the departure-fix. Rather than implementing this as a straight time-difference computation, the simulation (similar to its handling of runway minimum separations), we divide the departure-fix timeline into time-slots. The length of each time-slot is computed by assuming that the controllers will try to maintain on an average seven miles in-trail between consecutive departure-fix crossings (this is a number we found out from discussions with ex-controllers) and the flights will cross the departure-fix at 250 knots. If a MIT restriction is active at that departure-fix, then time-slots are calculated assuming the bigger MIT restriction. The flight can cross the departure-fix if the departure-fix time-slot associated with the current time-step is available. In this case, the simulation updates the flight's *actual departure fix crossing time*.

Step 6: Departure Fix to Enroute Merge Point Transit Time Calculation. In this step, the simulation computes and updates the *actual enroute merge-point queue entry time* for each new departure-fix crossing, by adding the estimated departure-fix to merge-fix transit time to the flight's departure-fix crossing time.

Step 7: Enroute Stream Merge Point Queue Management. In this step, the simulation determines whether the leader of the merge-fix queue can leave the queue (i.e., merge with the enroute traffic stream). The leader can merge into the enroute traffic stream only if a traffic gap concurrent with the current time-step is available. As discussed above, merge-fix crossing is modeled as fitting flights into 8 nmi-wide merge-slots (see [26] for more detail), some of which are already occupied by overflights or departures from ATL. Enroute stream gaps available to CLT and ATL departures are modeled as Poisson random events with a known occurrence rate. To compute the rate parameter for this Poisson process, we compute the actual rate at which CLT and ATL flights utilize enroute traffic stream time-slots for the day of simulation. We compute this rate by analyzing PDARS track data for ATL/CLT departures and overflight traffic merging with these departures before heading to destination airports in the Washington D.C. and New York areas. Once the rate is obtained, we apply Poisson random process with the defined rate to obtain a sequence of enroute merge-fix slots with slot-availability determined by the Poisson random process.

In Step 7, if an enroute traffic gap is available at the current time-step and a flight is waiting in the enroute merge-fix queue, then the simulation updates the flight's *actual enroute stream merge time*. The flight leaves the simulation at this time.

Models of Current-day and ATD-2 Departure Scheduling Algorithms. Since our aim is to assess how different levels of departure restrictions will impact ATD-2 operations as compared to current day operations, we developed scheduling algorithms that simulate sequencing and scheduling processes for departures under both, current-day operations as well as ATD-2 operations.

The following list summarizes the key events in a departure flight's transit in the current-day departure operations model:

- Departures pushback when ready. If the departure is impacted by an active APREQ restriction, then it may be required to absorb some delay at the gate (as explained above)
- Departures taxi according to taxi transit time models,
- Next, the departures enter the departure runway queue in a FCFS order. In some cases, the runway queue entry order may be slightly modified for departure-fix balancing (as explained above),
- Departures wait in runway queue until a time-slot becomes available and MIT restriction is satisfied if active,
- Departures takeoff and transit to the departure-fix queue according to the airborne transit time model,

- They wait in departure-fix queue until a departure-fix time-slot is available,
- Departures travel from the departure-fix to the enroute merge-fix queue according to the airborne transit time model,
- Departures wait in en route merge-fix queue until the next en route stream gap is available

For ATD-2 operations, it is assumed that a combination of airspace terminal departure scheduler and surface departure scheduler will generate *TOBTs* for each departure flight. These *TOBTs* take into consideration all the departure constraints in the model and will assign delays at the gate to optimize departure runway, departure-fix, and enroute stream gap utilization. The key events in a departure flight's transit under ATD-2 simulation are the same as current-day simulation events, except the departures push back at or near their *TOBTs* (they push back at the later of the following two times: *TOBT* and *Pushback Readiness Time*). *Pushback Readiness Time* for departures in both the current-day and ATD-2 simulations are exactly the same, i.e., we apply the same random perturbations to the respective *Scheduled Gate Departure Times* in both the cases. Note that the ATD-2 scheduling algorithm performs all its computations based on the knowledge of *Scheduled Gate Departure Times*, it does not know the exact time when the flight would be ready to push back (i.e., *Pushback Readiness Times*).

Furthermore, there is no gate holding for satisfying an APREQ constraint in ATD-2 operations. It is assumed that the ATD-2 scheduler will recommend a gate pushback time that maximizes the chances of hitting the allocated available enroute merge gap, and the current-day procedure of the tower holding the flight at the gate until a clearance from the ARTCC is obtained, will not be required.

An important parameter used by the ATD-2 scheduler is worth mentioning here. The parameter is the Target Departure Queue Length (TDQL). The ATD-2 scheduler holds flights at their gates in order to reduce airport surface congestion. If flights are released from their gates exactly at the *TOBTs* computed by ATD-2 scheduler, we would be releasing them just-in-time to make their assigned runway departure time-slot, departure-fix crossing time-slot and enroute merge-fix gap. If the flight experiences any delay in its transit then there is a risk of missing the assigned time-slot at one or more of these constrained NAS resources. Furthermore, there is a risk of creating additional delay for the flight (i.e., double penalty delay—delay at the gate plus delay on the airport surface or in the air to fit into a later available time-slot, not originally assigned to it). In order to minimize the risk of missing allocated time-slots, ATD-2 scheduling algorithms make provision for defining the TDQL, which represents a number of departure flights that are always desired to be waiting in the runway departure queue in order to maintain pressure on the runway and the downstream constrained NAS resources. As we will explain in the next section, we utilized this TDQL parameter as a control-knob for adjusting the departure metering strategy in order to obtain the maximum delay benefit for each MIT setting used in the simulations.

This section described the simulation platform used in our research. The next section describes the results obtained by conducting fast-time simulation experiments to assess the impact of MIT restriction relaxation on ATD-2 operations.

5.3 Results

Table 5-3. Departure fix usage by airport.

Departure Fix	CLT	EQY	HKY	JQF	SPA	SVH
MERIL	162	11	1	11		
NALEY	155	1	1	1	1	7
ZAVER	138	1	1	9		
BUCKL	92					
DEBIE	73					
ANDYS	31		1	2	1	
GANTS	29					
LILLS	26			1		
GIPPR	16					
Other	70					

Table 5-4. Enroute fix usage

Enroute Merge Fix	CLT	ATL	JQF
ZITTO	21	28	
FAK	39	59	2
TYI	8	8	
Potomac TRACON entry fix	33	53	

A preliminary fast-time simulation-based analysis was conducted to evaluate the impact of relaxing MIT restrictions on metroplex departure operations managed by ATD-2 scheduling algorithms. Here, we present results from this preliminary analysis. We simulated one full day of operations under three levels of MIT restrictions: (i) current-day MIT levels, (ii) current-day MITs reduced by 5 nmi each, and (iii) current-day MITs reduced by 10 nmi each. The day we simulated was June 10, 2015 (chosen for reasons explained in Section 5.2-5.2.2). This was a busy traffic day which featured a total of 792 departures from CLT and 1373 departures from ATL. Of these, 101 and 148 departures from the respective airports were impacted by the merge into the modeled overhead enroute traffic streams going to destination airports in the Northeast U.S. Table 5-3 summarizes the departure fix usage by CLT and CLT-satellite airport departures. Table 5-4 summarizes enroute merge-fix usage by CLT, CLT-satellite and ATL airports. Traffic demand sets for simulating departures at these airports (and arrivals for CLT) were derived from recorded PDARS data.

Our simulation also involved realistic models of departure restrictions. Table 5-2 in Section 5.2-5.2.2 showed the progression of departure restrictions impacting CLT and ATL on the simulated day. We simulated these MIT and APREQ departure restrictions using the approach described in Section 5.2-5.2.3. For MIT restrictions, in addition to applying the requisite spacing at the departure runway, we modeled a corresponding drop in departure-fix capacity and enroute traffic stream gap availability to coincide with the restriction active times. We utilized the respective MIT size to compute the magnitude of the departure-fix capacity drop. The nominal 15-minute capacity for a departure-fix was assumed to be 9 fix-crossings (roughly corresponding to a 7 nmi in-trail separation and 250 knot fix-crossing speed). By the same distance-time calculations, a 30 nmi MIT restriction, for example, meant that departure-fix capacity drops to roughly 3 crossings per 15 minutes during the MIT active time.

For computing a drop in enroute traffic stream gap availability during restriction active times, we conducted an analysis of CLT and ATL departure flows along with overhead enroute traffic flows that merge with them in ZDC airspace, before heading to the Northeast destination airports. In order to find out how many enroute traffic stream gaps were available to CLT and ATL departures to fit into, per 15 minute bin, we obtained the fix-crossing schedule (at the respective merge-fix) for all departures heading to major Northeast destination airports serviced by each enroute stream. Using this fix-crossing schedule (which was computed using PDARS data analysis) we calculated what percentage of the schedule slots were taken by CLT and ATL departures. As described in Section 5.2-5.2.1, we identified four separate enroute streams, each associated with one of the identified merge-locations: FAK for EWR, TEB, HPN, PHL, and BOS-bound departures, ZITTO for LGA-bound departures, TYI for JFK-bound departures, and Potomac TRACON entry-point for DCA, IAD, BWI, RIC-bound departures. The percentage of schedule slots used by ATL and CLT departures was used to derive a rate parameter for a Poisson process model. This Poisson process model simulated randomly occurring enroute stream gaps available to CLT and ATL departures. ATL and CLT flights were fitted into the available enroute stream gaps in Step 7 of the simulation process described in Section 5.2-5.2.3. For these enroute stream gap availability computations, we assumed that each enroute stream slot is 8 miles in length (as per observations from NASA for another enroute stream merge scenario [7]).

Using the traffic demand sets and realistic departure restrictions models described above, we performed multiple simulation-runs with models of current-day and ATD-2 departure scheduling processes. Size of MIT restrictions was varied between different simulation runs to assess the impact of relaxing MIT restrictions on the operations. For each current-day – ATD-2 pair of simulation runs with same MIT settings, we performed the following steps: (i) conduct current-day simulation (flights pushing back when ready), (ii) apply ATD-2 scheduling to compute *TOBTs*, and (iii) conduct ATD-2 simulation (flights pushing back at or near their *TOBTs*). The major simulation-runs performed for our preliminary evaluation are summarized below.

1. Run the current-day departure operations simulation model with the actual MIT settings that were active on the historical simulation day (6/10/2015). Compute metrics for current-day operations.

2. Apply the ATD-2 scheduling algorithm to compute *TOBTs* for traffic demand set from 6/10/2015.

3. Run the ATD-2 departure operations simulation model (i.e., flights push back at or near their respective *TOBTs*) with the actual MIT settings that were active on the historical simulation day (6/10/2015). Compute metrics for ATD-2 operations. Try steps 2 and 3 with different values of the TDQL parameter. Select the ATD-2 simulation run with TDQL parameter that provides the lowest overall total delay for the respective MIT setting.

4. Run the current-day departure operations simulation model with the MIT restrictions relaxed by 5 MIT each as compared to the active MIT restrictions on 6/10/2015. Compute metrics for current-day operations, with relaxed MITs.

5. Run the ATD-2 departure operations simulation model with the MIT restrictions relaxed by 5 MIT each as compared to the active MIT restrictions on 6/10/2015. Compute metrics for ATD-2 operations, with relaxed MITs.

6. Repeat current-day and ATD-2 simulations with more and further relaxed MITs, e.g., MITs relaxed by 10 nmi each, 15 nmi each, … as compared to the active MIT restrictions on 6/10/2015). When MITs were relaxed by more than 10 nmi, we observed that the airborne delays became untenably large. Hence, we stopped our analysis at the 10 nmi MIT relaxation stage.

The primary delay metrics of interest included (total and average) gate delay, taxi delay, TRACON delay, and en route delay for CLT departures. Gate delay is defined as the difference between the *actual pushback time* and the *pushback readiness time*. Taxi delay is defined as the *actual taxi out time* (pushback to takeoff) minus the unimpeded taxi time. TRACON delay is defined as the *actual TRACON transit time* (takeoff to departure-fix crossing) minus unimpeded TRACON transit time. Finally, enroute delay is defined as the a*ctual enroute transit time* (departure-fix to enroute stream merge) minus unimpeded transit time over the same enroute segment.

Figure 5-7 shows the total delays for CLT departures over the entire day of simulation (computed over 792 CLT departures) for four simulation scenarios: (i) current-day operations, (ii) ATD-2 operations with current levels of MIT restrictions, (iii) ATD-2 operations with all MITs relaxed by 5 nmi each, (iv) ATD-2 operations with all MITs relaxed by 10 nmi each. In Figure 5-7, delays are separated out by their location: gate, taxi, TRACON, and enroute. As seen from the Figure, very little delay was absorbed at the gate in the current-day simulation. This small gate delay was a result of the APREQ restriction impacting CLT departures going to LGA (see Table 5-2), which caused some departures to hold at their gates in order to absorb delay for hitting their APREQ runway takeoff time-slots. Besides the flights impacted by APREQs, all other flights left their gates when they were ready to pushback (i.e., at their *pushback readiness times*). As seen from the Figure, CLT departures experienced large taxi-out delays under simulated current-day operations. Airborne delays in the TRACON and enroute phases of flight were also extensive.

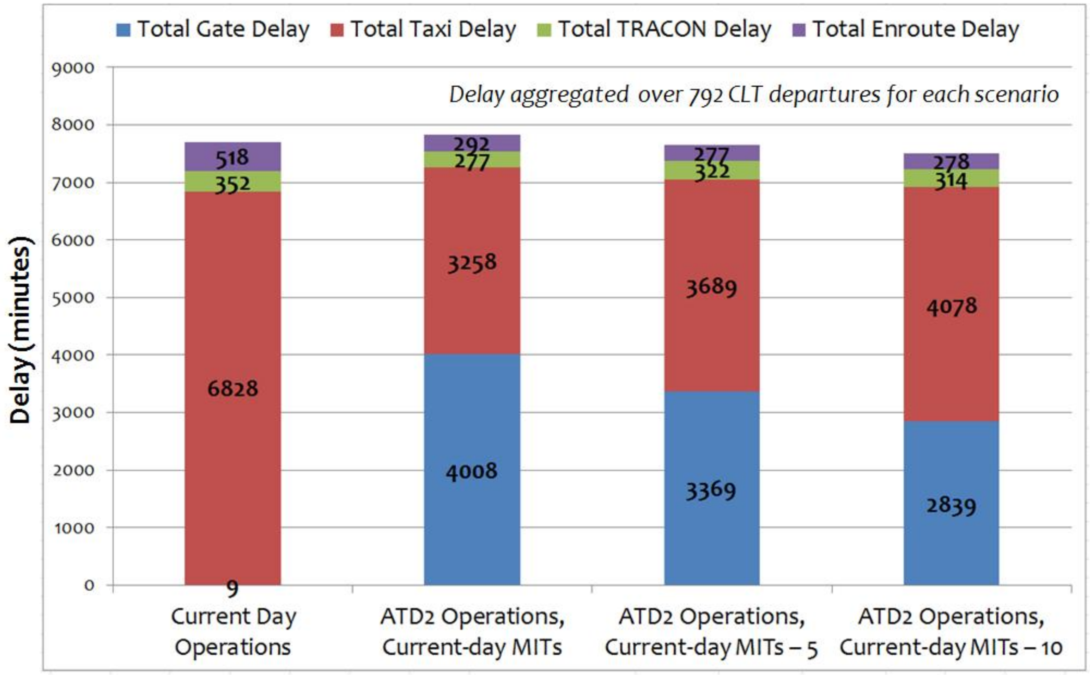


Figure 5-7. Delay distribution for different simulation settings

ATD-2 operations with current-day levels of MIT restrictions (second bar in the Figure) saw CLT departures experiencing 52% lesser taxi-delay, 21% lesser TRACON delay and 44% lesser enroute delay, as compared to current-day operations. Savings in taxi and airborne delays were more than compensated by an increase in gate delay. Thus, we see that the ATD-2 scheduler was successful in transferring the required delay from the inefficient delay absorption phases of taxi and airborne flight to the more efficient gate-hold delay absorption location. However, total delay aggregated over gate, taxi and airborne phases was bigger (by ~2%) in the ATD-2 operations as compared to current-day operations. Our hypothesis is that the main reason for this undesirable increase in overall delay was the following: application of restrictive MIT constraints at the runway had an adverse effect on the runway takeoff time coordination orchestrated by the ATD-2 scheduler via its recommendation of target gate pushback times. In other words, flights experienced double penalty delays as a result of MIT restrictions being implemented at runway takeoff on top of ATD-2 gate delays—many flights experienced gate delay as recommended by ATD-2 schedules and additional delay for applying runway MIT spacing constraints when they reached the runway departure queue.

Figure 5-8 shows the simulated delays experienced by CLT departures going to Washington D.C. area destination airports, in the CLT departure runway queue, for every 15-minute time-bin (red and green bars). The Figure also shows gate delays experienced by the same flights. Red bars show the runway queue delays in the current-day operations simulation. Green bars show runway queue delays in the ATD-2 simulation with the current-day MIT restrictions. Each bar shows the runway queue delays summed over all the flights that entered the runway queue during the respective time-bin. The green dotted line with diamond-shaped markers shows gate delays experienced by the same flights in the ATD-2 operations. (Note, in current-day operations Washington D.C. bound CLT departures did not experience any gate delays, because there was no corresponding APREQ active that day). It can be clearly seen from the Figure that especially during the time when KDCA and KIAD MIT restrictions were active (the blue and red shaded time-ranges), many flights experienced both, runway queue delays for MIT restriction implementation and gate delays imposed by the ATD-2 scheduler.

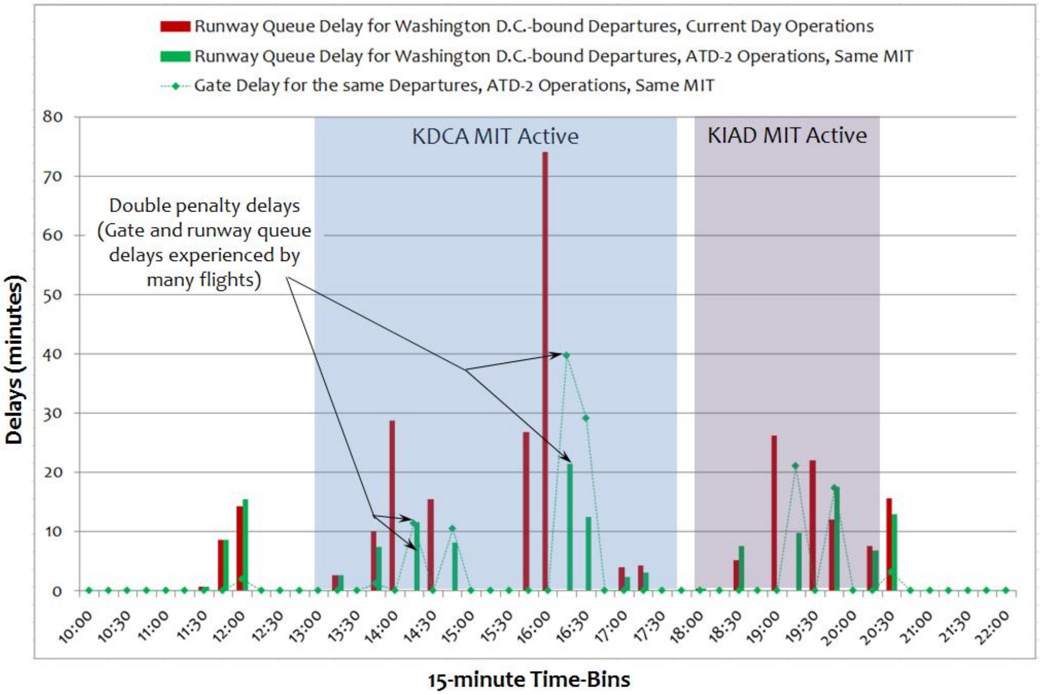
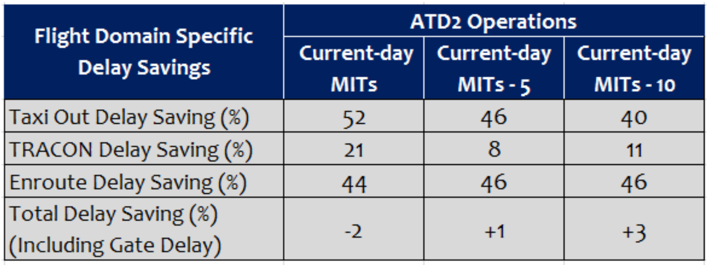


Figure 5-8. Runway queue and gate delays experienced by CLT departures to Washington D.C. area destination airports under different simulation scenarios

This double penalty delay effect can be avoided by relaxing the MIT restrictions when ATD-2 scheduler is in operation. Going back to Figure 5-7, we see that when MIT restrictions were relaxed by 5 and 10 nmi respectively, there was a drop in gate delays and a rise in taxi and airborne delays as compared to the ATD-2 operations with current-day MIT levels. More importantly, there was a drop in the total delay (gate + taxi + airborne). Furthermore, there were still significant savings in taxi and airborne delays as compared to current-day operations—specifically, 46% taxi delay saving, 8% TRACON delay saving, and 46% enroute delay saving over current-day operations for the ATD-2 operations with MITs reduced by 5 nmi; and 40% taxi delay saving, 11% TRACON delay saving, and 46% enroute delay saving over current-day operations for ATD-2 operations with MITs reduced by 10 nmi. Table 5-5 summarizes the percentage delay savings in different domains and also savings in the total delay (gate + taxi + airborne).

Table 5-5. Percent Delay Saving over Current-day Operations



Flight Domain Specific Delay Savings	ATD2 Operations		
	Current-day MITs	Current-day MITs - 5	Current-day MITs - 10
Taxi Out Delay Saving (%)	52	46	40
TRACON Delay Saving (%)	21	8	11
Enroute Delay Saving (%)	44	46	46
Total Delay Saving (%) (Including Gate Delay)	-2	+1	+3

Thus, our simulations show clear benefit of relaxing the MIT spacing requirements imposed at runway takeoff under ATD-2 operations. However, whether relaxed MITs will lead to unsafe operations is still an open question. The main concern here is that with relaxation of runway MIT spacing requirements, the TRACON may experience a faster inflow of departures headed for constrained departure-fixes such as MERIL, with lesser in trail separations between them. This may cause excess workload for the departure TRACON controllers because they may have to delay more departures in the TRACON for safe merging prior to exiting the TRACON via departure-fixes. Moreover, after crossing the departure-fix, the ARTCC controllers may also need to apply further delays to slow down departures for safe merging with overhead streams. We assessed the safety of operations under relaxed MIT restrictions by looking at airborne delay and TRACON departure demand metrics from our simulations. As seen from Figure 5-7, airborne delays for ATD-2 operations with relaxed MITs are well below the airborne delays with current-day MITs. So, this is a good initial indicator that relaxing the MITs when ATD-2 is in operation does not degrade the safety of operations. In contrast, when we conducted current-day operations simulation with relaxed MITs we saw that the airborne delays increased beyond the current-day operations with current-day levels of MITs, which is an indicator of worsening safety.

Further, we looked at TRACON departure demand for the constrained MERIL departure-fix, aggregated over departures from all airports within the CLT TRACON. The metric we focused on is the inter operation times between successive runway takeoffs from CLT TRACON airports for departures going to the MERIL fix. Figure 5-9 shows histograms of inter operation times for CLT TRACON departure takeoffs to MERIL fix. We have truncated the histograms at the ~15 minute level because inter operation times greater than this limit will have no implication for the safety of operations. As seen from the Figure, there is not much difference between the inter operation times for the ATD-2 operations with relaxed MITs as compared to the current-day operations or to the ATD-2 operations with current-day level of MITs. This is another indicator that relaxing MITs during active ATD-2 departure metering operations would not degrade the safety of operations and would not cause additional workload for TRACON departure controllers.

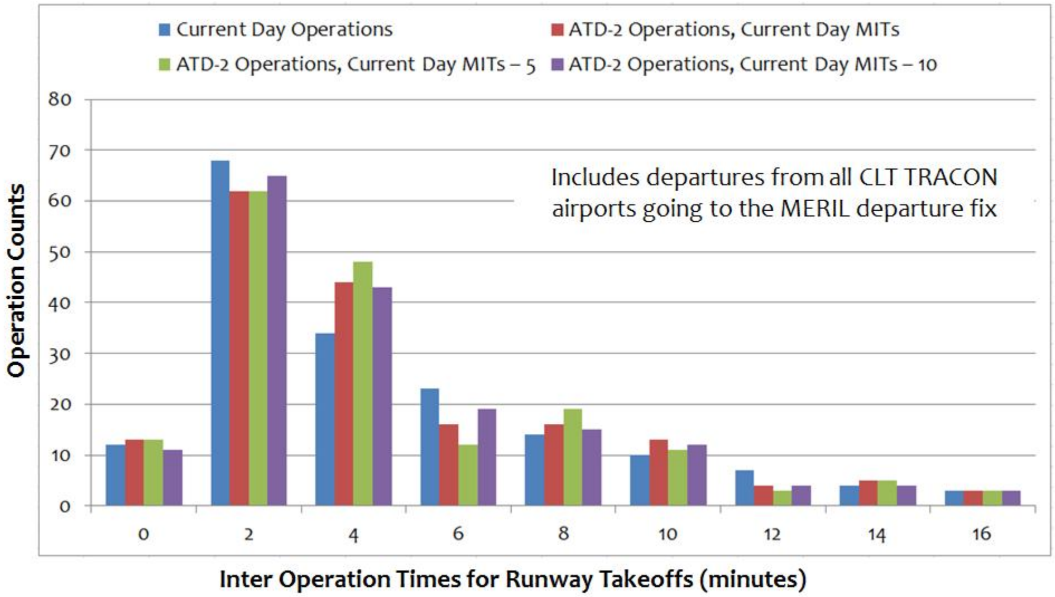


Figure 5-9. Inter operation times for runway takeoffs in CLT TRACON show that safety of operations does not degrade with relaxation of MITs under ATD-2 operations

5.4 Conclusions

This Chapter presented a simulation-based method for evaluating the impact of relaxing MIT restrictions imposed on departures from a metroplex airport, when under the control of ATD-2 departure metering. We developed a metroplex departure metering simulation platform for supporting this analysis. This simulation platform included a realistic link-node model for airport surface and airspace routes, developed based on a PDARS track data analysis based identification of major departure flows and key enroute merge-locations, ASPM operations data and PDARS track data analysis based quantification of taxi and airborne link transit times, as well as NTML departure restrictions data analysis based modeling of MIT and APREQ restrictions.

The objective of this analysis was two-fold: (i) to quantify how much efficiency benefit (i.e., delay reduction) can be gained by relaxing MIT restrictions when under ATD-2 scheduling and (ii) to evaluate whether relaxing MIT restrictions may lead to unsafe departure merging operations. We demonstrated that application of MIT restrictions along with ATD-2 gate holding causes double penalty delay effect, with flights receiving gate delays and then being delayed again in the departure runway queue. In our simulations, ATD-2 operations with current-day level of MIT restrictions demonstrated significant taxi and airborne delay savings over current-day operations, but the total delay (gate + taxi + airborne) was ~2% bigger in the ATD-2 operations owing to the double penalty delay effect. When MIT restrictions were relaxed by 5 and 10 nmi in our simulations, we saw that significant taxi and airborne delay savings with respect to current-day operations persisted. Furthermore, gate delays were reduced and overall, the total delay reduced below the total delay in current-day operations (with total delay savings of 1 and 3 % respectively).

We also evaluated the safety of operations under ATD-2 scheduling with relaxed MITs by comparing airborne delays and TRACON departure demand with the respective metrics from current-day operations. We found that the total airborne delays were lower in the ATD-2 operations and TRACON departure demands (inter operation times for runway takeoffs) were almost identical to current-day operations, despite the relaxed MIT restrictions. Thus, our preliminary conclusions are that relaxing MITs when ATD-2 scheduling is active would save around 1-3% total delays while retaining high level of taxi and airborne delay savings with respect to current-day operations, as well as maintaining a level of safety commensurate with current-day operations.

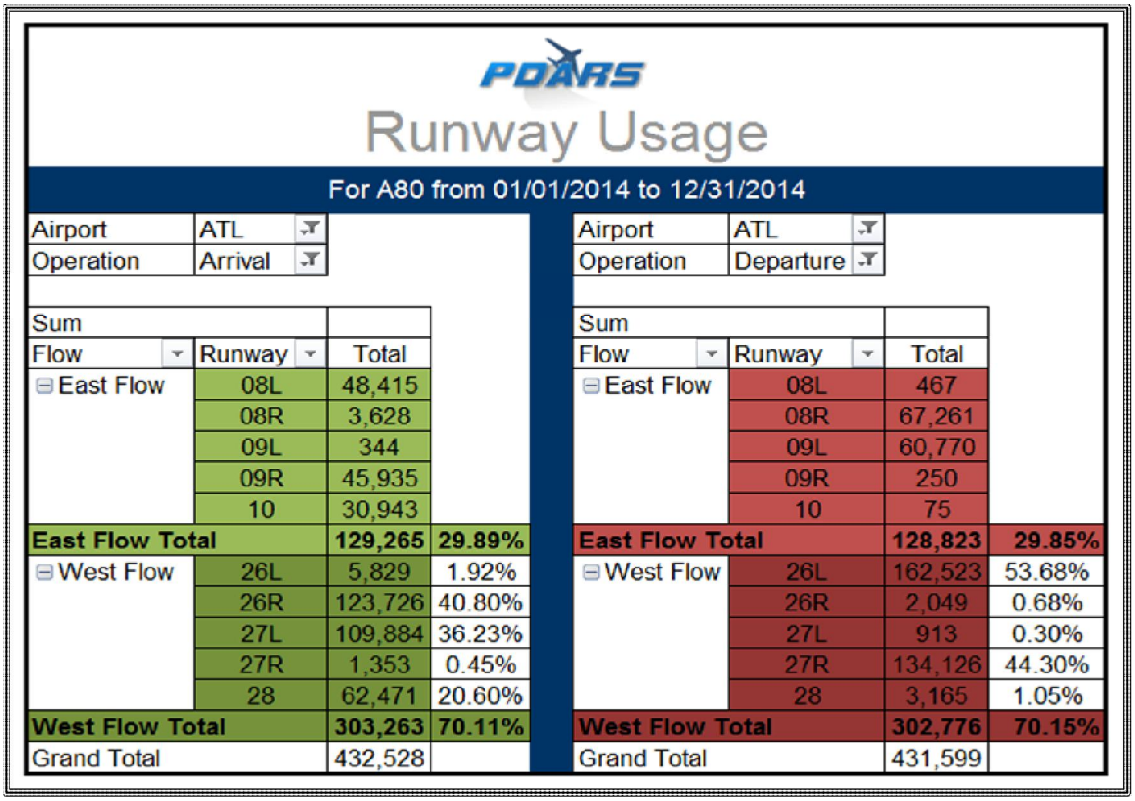
It is important to highlight that these are preliminary findings. The analyses documented in this Chapter can be further enhanced in multiple ways to provide a higher fidelity evaluation of the delay-reducing and safety-retaining effects of MIT relaxation. Some ways in which the analyses can be further enhanced are: (i) validating the departure flows, enroute merge locations, and enroute merge modeling approach adopted in the simulation platform via interviews with airport tower, TRACON and ARTCC controllers, (ii) conducting simulations over multiple days with different levels of historical MIT restrictions modeling, (iii) improving the accuracy of airport surface modeling by integrating NASA's high-fidelity surface simulator, the Surface Operations Simulator and Scheduler (SOSS) into the current metroplex departure metering simulation platform, and (iv) enhancing the current emulation model of ATD-2 scheduling algorithm via knowledge transfer between NASA developers of the actual ATD-2 scheduling algorithm and our research team.

6 Atlanta Hartsfield International Airport Model Development in SOSS

This Chapter describes the steps involved in generating an airport model for the Atlanta Hartsfield International Airport (KATL), for use with the SOSS simulation platform. The airport model generation process involves the following steps: (i) selecting the airport runway configuration(s), (ii) generating the required airport adaptation data, (iii) generating traffic scenarios, and (iv) running the simulation and validating simulation results against real operational data. The details of each task are described next.

6.1 Runway Configuration Selection

The team analyzed one year worth of runway usage data for KATL using PDARS data from January 1st, 2015 to December 31st, 2014. Customized reports generated by ATAC using PDARS data show runway usage by arrivals and departures in 15-minute time bins. The runway usage by arrivals and departures in each 15-minute time-bin is then used to identify the runway configuration active during that time-period. The most commonly used runway configurations for KATL in 2015 was West Flow, with runways 26R, 27L, and 28 used for arrivals and runways 26L and 27R used for departures. Figure shows the summary data from our analysis.



PDARS

Runway Usage

For A80 from 01/01/2014 to 12/31/2014

Airport	ATL
Operation	Arrival

Sum			
Flow	Runway	Total	
⊟ East Flow	08L	48,415	
	08R	3,628	
	09L	344	
	09R	45,935	
	10	30,943	
East Flow Total		**129,265**	**29.89%**
⊟ West Flow	26L	5,829	1.92%
	26R	123,726	40.80%
	27L	109,884	36.23%
	27R	1,353	0.45%
	28	62,471	20.60%
West Flow Total		**303,263**	**70.11%**
Grand Total		432,528	

Airport	ATL
Operation	Departure

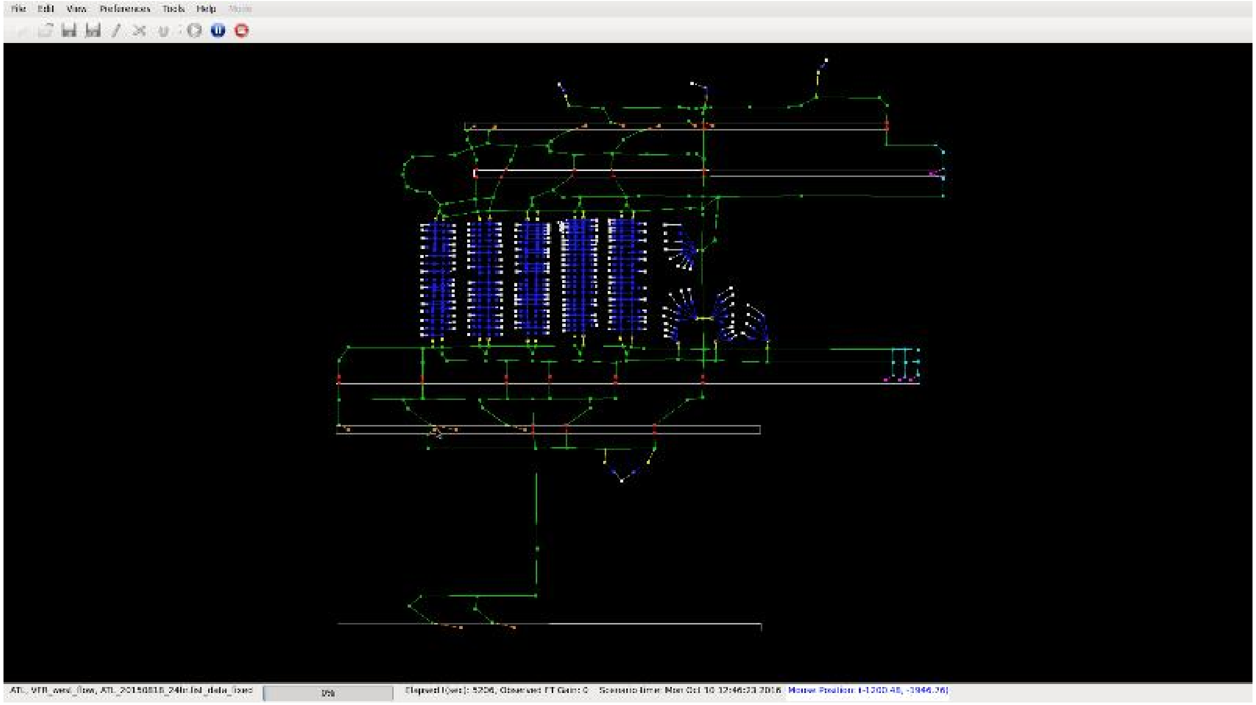
Sum			
Flow	Runway	Total	
⊟ East Flow	08L	467	
	08R	67,261	
	09L	60,770	
	09R	250	
	10	75	
East Flow Total		**128,823**	**29.85%**
⊟ West Flow	26L	162,523	53.68%
	26R	2,049	0.68%
	27L	913	0.30%
	27R	134,126	44.30%
	28	3,165	1.05%
West Flow Total		**302,776**	**70.15%**
Grand Total		431,599	

Figure 6-1. Runway Usage Analysis Using 2014 PDARS Data

The most used runway configuration, the West-flow, was used in our SOSS model of KATL.

6.2 Generating Airport Adaptation Data

ATAC developed in-house processes to convert high-fidelity airport adaptation data included in ATAC's SIMMOD PRO® software to SOSS node and link format. These processes generated a link-node model, which served as a two-dimensional representation of the airport surface. The KATL link-node model is shown below in Figure 6-2, and the specific characteristics of the model (i.e., the types and quantities of nodes and links) are defined in Table 6-1.



ATL_VFR_west_flow, ATL_20150818_24hr.txt_data_fixed

Elapsed t(sec): 5206, Observed FT Gain: 0 Scenario time: Mon Oct 10 12:46:23 2016 Mouse Position: (-1200.48, -3946.26)

Figure 6-2. KATL Link-Node Model

Table 6-1. KATL Link-node Model Characteristics

Node Type	Quantity	Link Type	Quantity
Arrival	13	Arrival	13
Departure	4	Departure	5
Queue	14	Queue	13
Runway Crossing	32	Runway Crossing	16
Taxiway	146	Taxiway	201
Spot	31	Spot	31
Ramp	441	Ramp	562
Gate	217	Gate	218

The team also generated a list of arrival and departure routes based on PDARS data analysis. Each route is defined by specifying a sequence of nodes between gates and runways. An arrival route starts with an arrival node at the runway and ends with a gate node. A departure route starts with a gate node and ends with a departure node at the runway.

6.3 Generating Traffic Demand Set

ATAC developed a process to take PDARS operational track data and convert it into SOSS runway and spot assignments for individual flights. For gate assignments, we relied on a round-robin, ad-hoc assignment method. The first step before PDARS data analysis was to select an appropriate simulation day to model. For this, ATAC analyzed 2015 PDARS runway usage data to first identify the peak mean month in terms of average daily operations. Then, within the peak month, ATAC identified days that had the West-flow runway configuration on for a majority of the day. Among the days that satisfied these criteria, the day with the maximum number of operations (arrivals and departures) was chosen as the simulation day. The selected day was August 18, 2015.

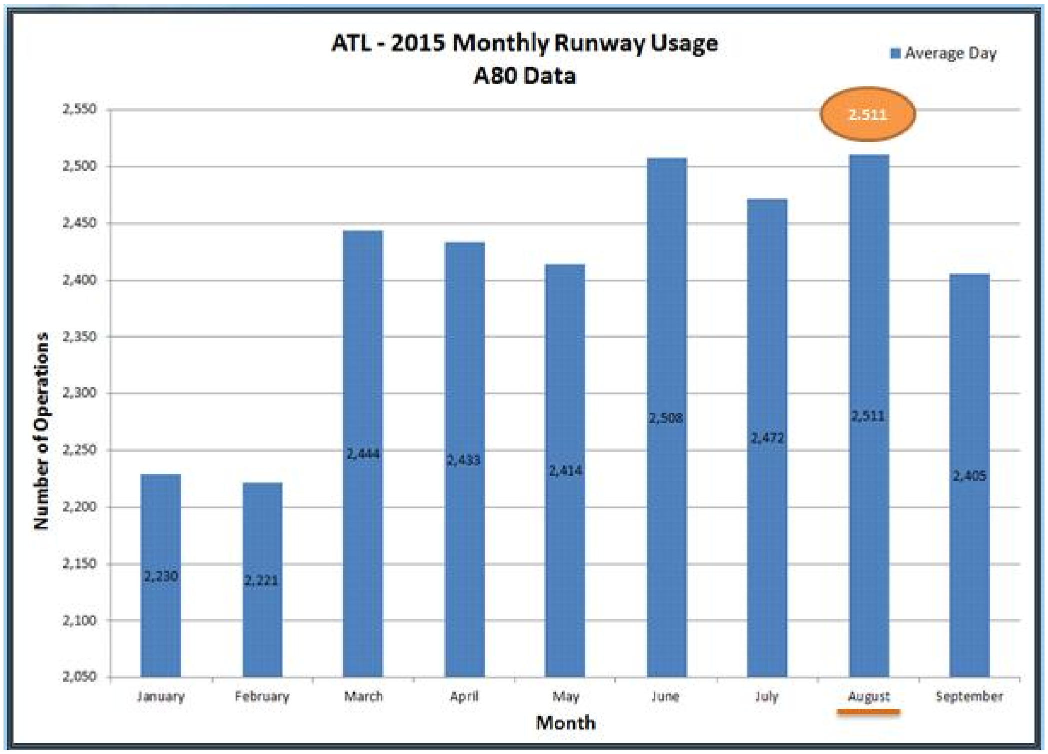


Figure 6-3. Peak 2015 Month By Number of Operations (Arrivals plus Departures) was August

After a day was selected, ATAC then processed the historical PDARS track data for the selected day and converted it into key SOSS traffic demand set attributes such as aircraft callsigns, aircraft types, assigned spots, assigned runways and assigned departure fixes. Since PDARS data does not cover ground flight tracks all the way back to/from their gates, we were not able to obtain gate assignment information from PDARS. Instead, we used an ad-hoc round-robin assignment of gates to flights to complete the development of the traffic demand set.

6.4 Validating the KATL SOSS model

In order to accurately capture the complexities of Atlanta's Hartsfield-Jackson International Airport (ATL), a computer simulation model based on the current conditions at ATL was created and calibration was attempted against existing conditions. Model validation is a critical step in the modeling process since an accurately calibrated model is the foundation for all alternative scenarios to be developed. To accurately model current and/or future operations, a model must first accurately model past operations. The remainder of this section discusses the calibration process and validation results.

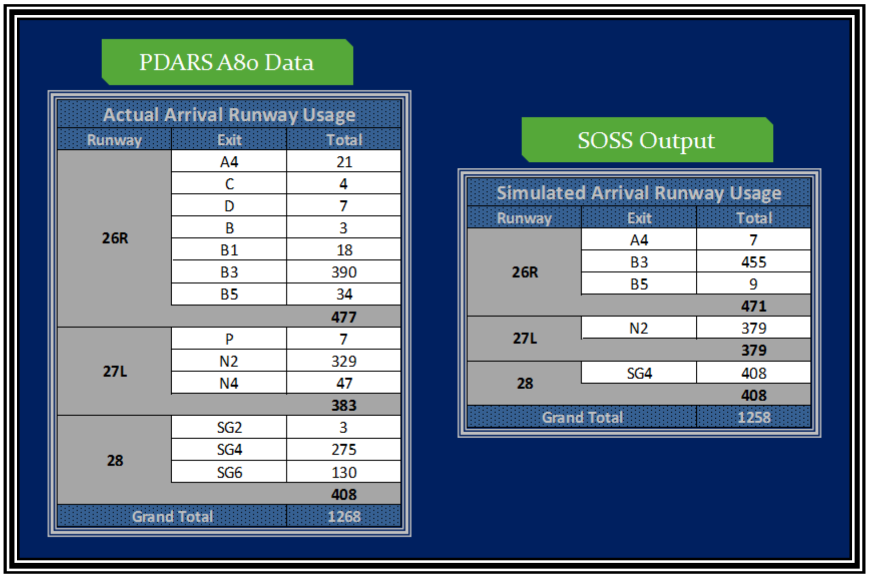
The airfield calibration analysis for ATL considered the typical taxi path routings used when operating during a west flow configuration which include arrivals to Runways 26R, 27L, and 28 and departures from Runways 26L and 27R. This configuration is predominant and is the preferred airfield configuration at ATL, accounting for approximately 70% of the operating time annually, as shown in Figure 6-1.

Arrival and departure flight track data was generated from PDARS for the sample day, August 18, 2015. Along with track data, the PDARS Atlanta TRACON (A80) Runway Usage Report and the ATL ASDE-X Taxi Time Report were used to validate runway usage and aircraft taxiing speeds for both arrivals and departures.

Calibrating simulated airfield activity is an iterative process. A link/node runway, taxiway, and ramp structure was created with the help of aerial imagery and CAD.

The model was then run and the output analyzed against the truth data reports. To validate runway usage, simulation output is compared to PDARS A80 Runway Usage Report.

Figure 6-4 displays the validation of the model's runway usage for arrivals. The number of simulated arrival operations is similar to truth data, with just a 10-operation difference in total arrival operations (1258 vs. 1268). When broken out into runway usage, simulated output shows 471 operations to Runway 26R vs. 477 in PDARS data, 379 operations to Runway 27L vs. 383 in PDARS data, and 408 operations to Runway 28, the same as shown in PDARS truth data.



PDARS A8o Data

Actual Arrival Runway Usage

Runway	Exit	Total
26R	A4	21
	C	4
	D	7
	B	3
	B1	18
	B3	390
	B5	34
		477
27L	P	7
	N2	329
	N4	47
		383
28	SG2	3
	SG4	275
	SG6	130
		408
Grand Total		1268

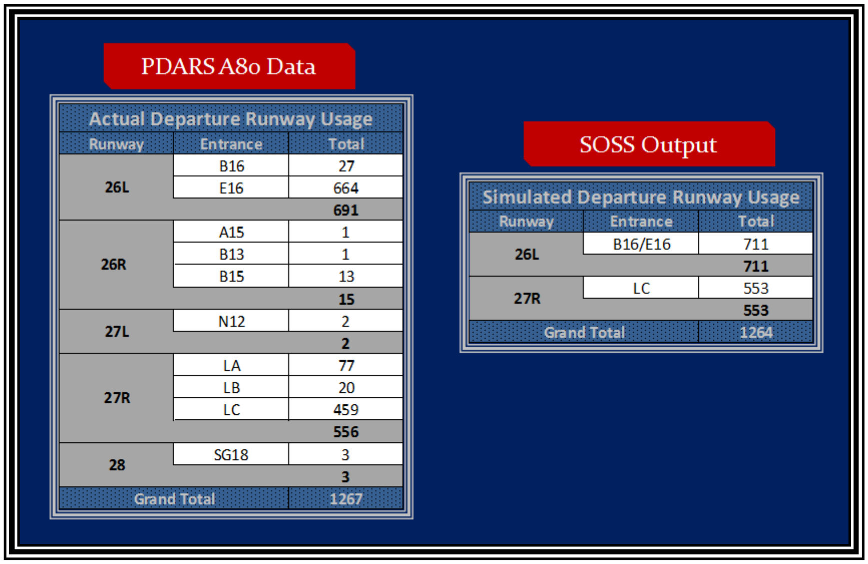
SOSS Output

Simulated Arrival Runway Usage

Runway	Exit	Total
26R	A4	7
	B3	455
	B5	9
		471
27L	N2	379
		379
28	SG4	408
		408
Grand Total		1258

Figure 6-4. ATL Runway Usage Calibration - Arrivals

Figure 6-5 displays the validation of the model's runway usage for departures. The number of simulated departure operations is similar to truth data, with just a 3-operation difference in total departure operations (1264 vs. 1267). When broken out into runway usage, simulated output shows 711 operations to Runway 26L vs. 706 north complex operations (Runways 26L and 26R combined) in PDARS data, and 553 operations to Runway 27R vs. 558 middle complex operations (Runways 27R and 27L combined) in PDARS data. Operations to the south complex (Runway 28) were not modeled for departures.



PDARS A80 Data

Actual Departure Runway Usage

Runway	Entrance	Total
26L	B16	27
	E16	664
		691
26R	A15	1
	B13	1
	B15	13
		15
27L	N12	2
		2
27R	LA	77
	LB	20
	LC	459
		556
28	SG18	3
		3
Grand Total		1267

SOSS Output

Simulated Departure Runway Usage

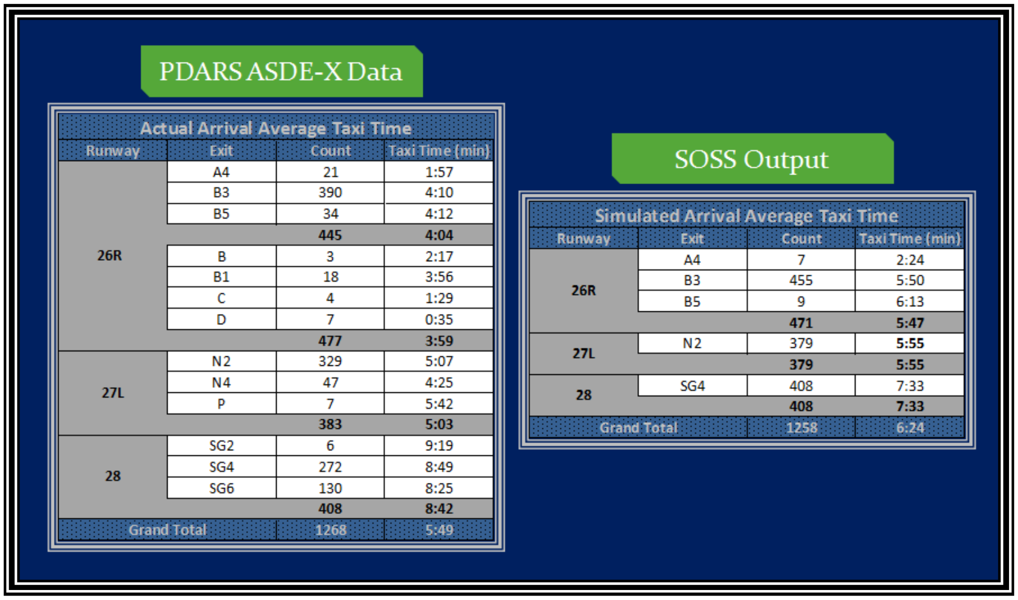
Runway	Entrance	Total
26L	B16/E16	711
		711
27R	LC	553
		553
Grand Total		1264

Figure 6-5. ATL Runway Usage Calibration - Departures

To validate taxi time, the simulation results are used to determine average taxi time and +/- 1 standard deviation (STD) from the simulation average. Standard deviation is a common statistic used as a measure of the dispersion, or variation, from the average in a distribution.

The calibration goal is to get the PDARS truth data to be within 1 STD of the simulated data. If the data is not within 1 STD, updates are made to the model, the model is exercised, and the data is rechecked. This process is generally repeated until the output falls within the proper distribution.

Figure 6-6 displays the results of the first iteration for ATL arrivals' taxi time calibration. The SOSS simulation model used the top 3 exits available from Runway 26R (in terms of utilization, taken from truth data), and the top exit from Runways 27L and 28.



PDARS ASDE-X Data

Actual Arrival Average Taxi Time

Runway	Exit	Count	Taxi Time (min)
26R	A4	21	1:57
	B3	390	4:10
	B5	34	4:12
		445	**4:04**
	B	3	2:17
	B1	18	3:56
	C	4	1:29
	D	7	0:35
		477	**3:59**
27L	N2	329	5:07
	N4	47	4:25
	P	7	5:42
		383	**5:03**
28	SG2	6	9:19
	SG4	272	8:49
	SG6	130	8:25
		408	**8:42**
Grand Total		1268	5:49

SOSS Output

Simulated Arrival Average Taxi Time

Runway	Exit	Count	Taxi Time (min)
26R	A4	7	2:24
	B3	455	5:50
	B5	9	6:13
		471	**5:47**
27L	N2	379	**5:55**
		379	**5:55**
28	SG4	408	7:33
		408	**7:33**
Grand Total		1258	6:24

Figure 6-6. ATL Taxi Time Calibration - Arrivals - Table Format

Figure 6-7 displays the results in graphical format. The red horizontal lines represent the simulation output's average taxi time for arrivals from exiting the runway to entering the ramp. The black vertical lines represent the dispersion of the simulation data from the average, or +/- 1 STD. The green horizontal lines depict where the ASDE-X truth data falls in relation to +/- 1 STD. The green lines should cross the vertical lines as close to the red line as possible

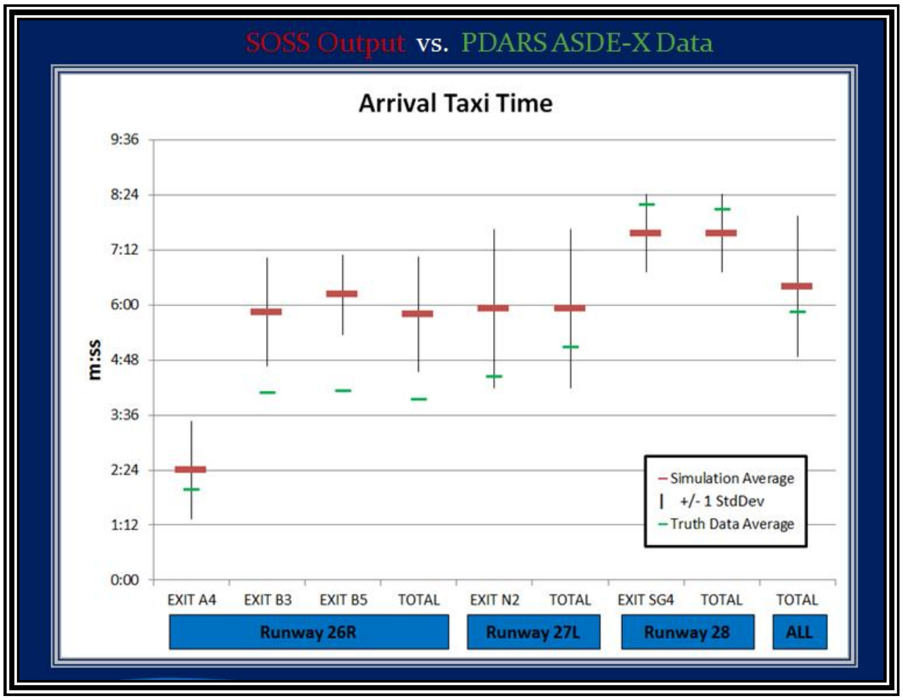
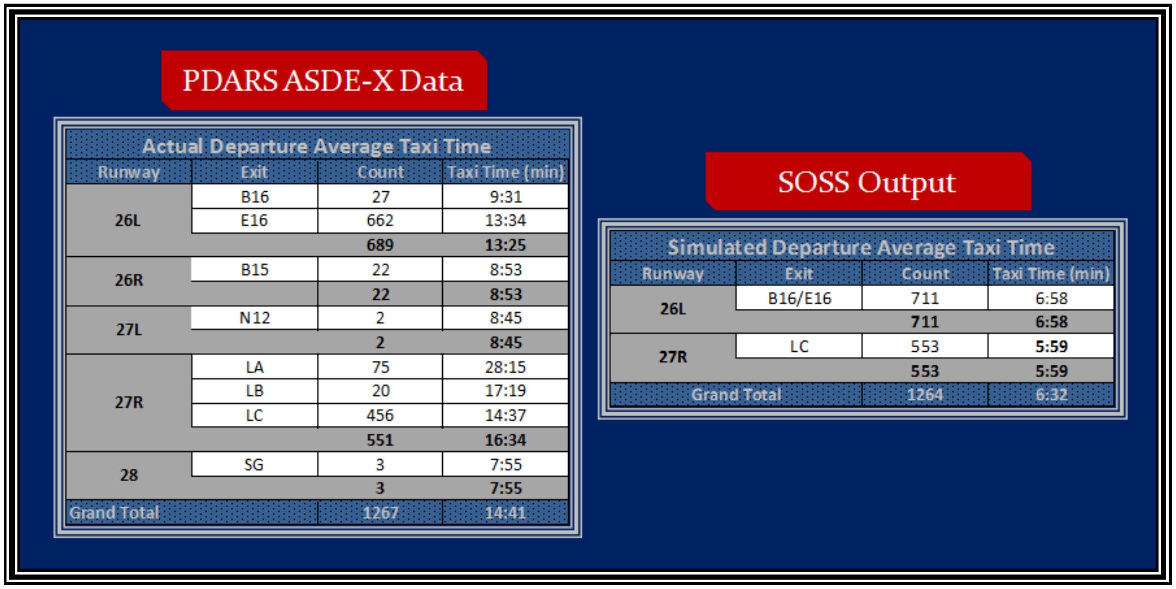


Figure 6-7. ATL Taxi Time Calibration - Arrivals - Graphical Format

As shown in Figure 6-7 Total Arrival Taxi Time aligns well between the simulation and truth data. In addition, Runways 27L and 28 both show simulation results being within the acceptable parameters. However, when looking at taxi time associated with simulation results for Runway 26R, the data does not align as well.

Arrivals utilizing Exit A4 from Runway 26R are calibrated, but simulated arrivals utilizing Exits B3 and B5 have a much greater (longer) taxi to ramp compared to truth data. The total taxi time from runway to gate also shows that the model needs slight adjustments to shorten the taxi time for arrivals, to allow for the truth data to fall within 1 STD of the simulation output.

Figure 6-8 displays the results of the first iteration for ATL departures' taxi time calibration. The SOSS simulation model used a combined southern departure queue (E16) and northern departure queue (B16) from Runway 26L, and the most utilized departure queue from Runway 27R (LC).



PDARS ASDE-X Data

Actual Departure Average Taxi Time

Runway	Exit	Count	Taxi Time (min)
26L	B16	27	9:31
	E16	662	13:34
		689	**13:25**
26R	B15	22	8:53
		22	**8:53**
27L	N12	2	8:45
		2	**8:45**
27R	LA	75	28:15
	LB	20	17:19
	LC	456	14:37
		551	**16:34**
28	SG	3	7:55
		3	**7:55**
Grand Total		1267	14:41

SOSS Output

Simulated Departure Average Taxi Time

Runway	Exit	Count	Taxi Time (min)
26L	B16/E16	711	6:58
		711	**6:58**
27R	LC	553	**5:59**
		553	**5:59**
Grand Total		1264	6:32

Figure 6-8. ATL Taxi Time Calibration - Departures - Table Format

Figure 6-9 displays the results in graphical format. Again, the red horizontal lines represent the simulation output's average taxi time for depatures from exiting the ramp to entering the runway. The black vertical lines represent the dispersion of the simulation data from the average, or +/- 1 STD. The green horizontal lines depict where the ASDE-X truth data falls in relation to +/- 1 STD. The green lines should cross the vertical lines as close to the red line as possible.

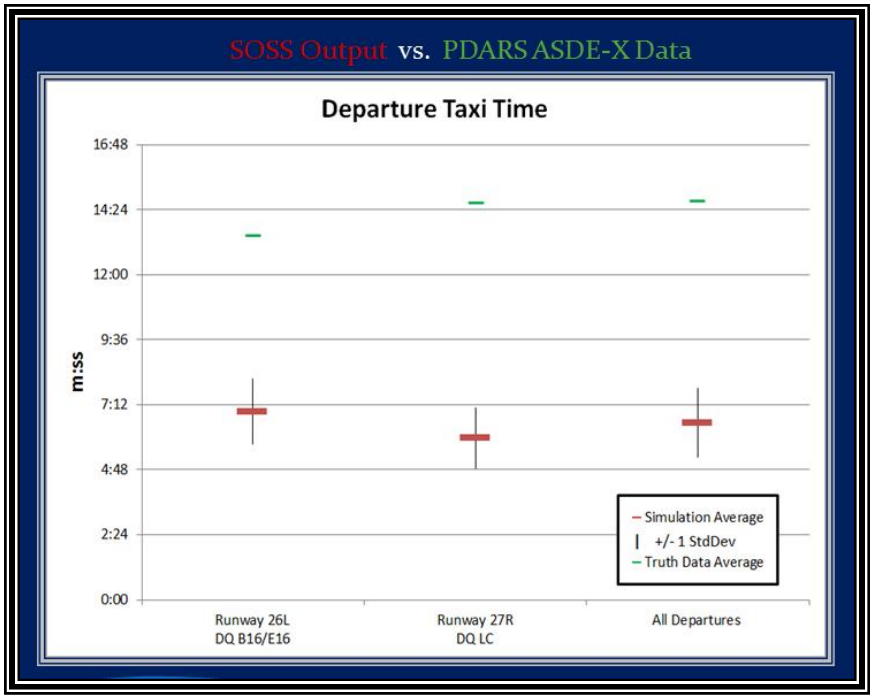


Figure 6-9. ATL Taxi Time Calibration - Departures - Graphical Format

The SOSS model output for departures portrays significantly faster taxi times in comparison to truth data, for both departure runways.

6.5 Calibration Conclusions

Runway, exit, and departure queue utilization are shown to be in good agreement with the SOSS ATL model. However, truth data taxi times were shown to be outside the acceptable 1 STD of the model's average requiring adjustments to the model in order to bring the taxi times within satisfactory tolerances. More work is needed to for the model to be considered fully calibrated.

When running the ATL SOSS model, the CD&R function was turned off. The model was unable to run completely with this function turned on. The CD&R function mimics ground and local air traffic control clearances which may explain why departure taxi time in the model is so much lower than the truth data. This and other adjustments to the model are required prior to confirming it is calibrated.

7 References

[1] Idris H., “Identification of Local and Propagated Queuing Effects at Major Airports,” Presented at the AIAA Aviation Forum, Aviation Technology, Integration and Operations (ATIO) conference, Dallas, TX, July 2015.

[2] Swenson H., Thipphavong J., Sadovsky A., Chen L., Sullivan C., and Martin L., “Design and Evaluation of the Terminal Area Precision Scheduling and Spacing System.” Ninth USA/Europe Air Traffic Management Research and Development Seminar (ATM2011), Berlin, Germany, 2011.

[3] Thipphavong J., Swenson H., Lin P., Seo A. and Bagasol A., “Efficiency Benefits Using the Terminal Area Precision Scheduling and Spacing System.” AIAA Aviation Technology Integration and Operations (ATIO), Virginia Beach, VA. 2011.

[4] Hayashi M., Hoang T., Jung Y., Gupta G., Malik W., and Dulchinos V., “Usability Evaluation of the Spot and Runway Departure Advisor (SARDA) Concept in a Dallas/Fort Worth Airport Tower Simulation.” Tenth USA/Europe Air Traffic Management Research and Development Seminar, Chicago, IL. 2013.

[5] Brinton C., Provan C., Lent S., Prevost T., and Passmore S., “Collaborative Departure Queue Management,” Ninth USA/Europe Air Traffic Management Research and Development Seminar (ATM2011), Berlin, Germany. 2011.

[6] Simaiakis I., Khadilkar H., Balakrishnan H., Reynolds T., Hansman R.J., Reilly B., and Urlass S., “Demonstration of Reduced Airport Congestion Through Pushback Rate Control,” Ninth USA/Europe Air Traffic Management Research and Development Seminar (ATM2011), Berlin, Germany. 2011.

[7] Engelland, S., Capps, A., Day, K., “Precision Departure Release Capability (PDRC) Concept of Operations,” NASA/TM–2013-216534, June 2013.

[8] Engelland S. and Capps A., “Trajectory-Based Takeoff Time Predictions Applied to Tactical Departure Scheduling: Concept Description, System Design, and Initial Observations,” 11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, 2011.

[9] Gupta G., Malik W., and Jung Y., “Incorporating Active Runway Crossing in Airport Departure Scheduling.” AIAA Guidance, Navigation, and Control Conference, Ontario, Canada, 2010.

[10] Capps A., Kistler M. and Engelland S., “Design Characteristics of a Terminal Departure Scheduler,” 14th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, 2014.

[11] Zelinski S., “A Framework for Integrating Arrival, Departure and Surface Operations Scheduling,” 33rd Digital Avionics Systems Conference (DASC), October 5-9, Colorado Springs, CO, 2014.

[12] Callantine T., Kupfer M., Martin L, Mercer J., and Prevot T, System-Level Performance Evaluation of ATD-1 Ground-Based Technologies. 14th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Atlanta, GA, 2014.

[13] Xue M. and Zelinski S., "Dynamic Stochastic Scheduler for Integrated Arrivals and Departures", 33rd Digital Avionics Systems Conference (DASC), October 5-9, Colorado Springs, CO, 2014.

[14] Bosson C., Xue M., and Zelinski S., "Optimizing Integrated Terminal Airspace Operations Under Uncertainty", 33rd Digital Avionics Systems Conference (DASC), October 5-9, Colorado Springs, CO, 2014.

[15] Bertsimas D., “Air Traffic Flow Management at Airports: A Unified Optimization Approach,” Tenth USA/Europe Air Traffic Management Research and Development

[16] Aponso B., Coppenbarger R., Jung Y., Quon L., Lohr G., O’Conner N., and Engelland S., “Identifying Key Issues and Potential Solutions for Integrated Arrival, Departure, Surface Operations by Surveying Stakeholder Preferences,” 15th AIAA Aviation Technology, Integration, and Operations (ATIO), Dallas, TX, 2015.

[17] Chevalley E., Parke B., Kraut J., Bienert N., Omar F., and Palmer E., “Scheduling and Delivering Aircraft to Departure Fixes in the NY Metroplex with Controller-Managed Spacing Tools,” 15th AIAA Aviation Technology, Integration, and Operations (ATIO), Dallas, TX, 2015.

[18] Shen N., H. Idris, and V. Orlando, 2012, Estimation Of Departure Metering Benefits At Major Airports Using Queuing Analysis, 31st Digital Avionics Systems Conference, Williamsburg, VA, Oct.

[19] Idris H. and N. Shen, 2014, Impact of Gate Availability on Departure Metering Benefit, 14th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Atlanta, Georgia.

[20] Nakahara, A., and T. G. Reynolds, An Approach for Estimating Current and Future Benefits of Airport Surface Congestion Management Techniques, 12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Indianapolis, Indiana, 2012.

[21] Simaiakis, I., and H. Balakrishnan, 2009, Queuing Models of Airport Departure Processes for Emissions Reduction Techniques, Proceedings of the AIAA Guidance, Navigation and Control Conference.

[22] Idris, H., A. Evans, S. Evans, and D. Kozarsky, 2004, Refined Benefits Assessment of Multi-Center Traffic Management Advisor for Philadelphia and New York, 4th AIAA Aviation Technology, Integration, and Operations (ATIO) conference, Chicago, IL.

[23] Evans A. and . Idris, Estimation of Arrival Capacity and Utilization at Major Airports, 6th USA/Europe Air Traffic Management Research and Development Seminar, Baltimore, Maryland, June 27 – 30.

[24] Pujet, N., 1999, Modeling and Control of the Departure Process of Congested Airports, PhD Thesis. Cambridge, Massachusetts Institute of Technology.

[25] Idris H., J-P Clarke, R. Bhuva, and L. Kang, 2002 Queuing Model for Taxi-Out Time Estimation, Air Traffic Control Quarterly Journal, Volume 10, Number 1.

[26] Saraf A., Timar S., Shen N., and Idris H., “Preliminary Queuing Analysis of Integrated departure operations in metroplex systems,” 34th Digital Avionics Systems Conference, Prague, Czech Republic, September 13-17, 2015.

[27] Idris H., Shen N., Saraf A., Bertino J., and Luch N., " Queue Buffer Sizing for Efficient and Robust Integrated Departure Scheduling," 16th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Washington, D.C., 2016.

[28] Saraf A., Bertino J., Luch N., Shen N., and Idris H., "Miles-in-Trail Restrictions Relaxation: A Key Benefit Mechanism of Integrated Arrival Departure Surface Traffic Management," 16th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference, Washington, D.C., 2016.

[29] Idris H, Shen N., Saraf A., Bertino J., and Zelinski S., "Comparison of Different Control Schemes for Strategic Departure Metering," 35th IEEE/AIAA Digital Avionics Systems Conference, Sacramanto, CA, September 2016.

[30] NASA, "A Concept for Integrated Arrival/Departure/Surface (IADS) Traffic Management for the Metroplex," Airspace Technology Demonstration 2 (ATD-2) ConOps Synopsis (DRAFT), 2015.

[31] Rios, Joseph. "Aggregate statistics of national traffic management initiatives." *10th Aviation Technology, Integration, and Operations (ATIO) Conference (AIAA)*. 2010.