Proceedings of the 2004 Winter Simulation Conference
R .G. Ingalls, M. D. Rossetti, J. S. Smith, and B. A. Peters, eds.

SIMULATING AIRSPACE REDESIGN FOR ARRIVALS TO DETROIT-WAYNE COUNTY AIRPORT (DTW)

Justin Boesel
David Bodoh

Center for Advanced Aviation System Development (CAASD)
The MITRE Corporation
7515 Colshire Dr.
McLean, Virginia 22102, U.S.A.

ABSTRACT

In 2001, Detroit Metropolitan Wayne County Airport (DTW) opened a new runway parallel to three existing runways. While this increases DTW's runway capacity, the airport is served by an airspace (routes, procedures, and controller assignments) that was designed only for a three-runway airport. To increase the airport's effective capacity, the Detroit-area Terminal Radar Approach Control facility (D21 TRACON) and nearby Air Route Traffic Control Centers (ARTCC) are redesigning their airspace. This paper describes the simulation modeling effort to estimate delay and cost benefits of the ARTCC redesign for arrival traffic. The model, written in the SLX simulation language, represents miles-in-trail (MIT) restrictions, as well as air traffic controllers' ability to direct flights to different paths dynamically, based on predicted demand downstream. The redesign work is part of the Federal Aviation Administration's Midwest Airspace Capacity Enhancement (MACE) project.

1 INTRODUCTION

Detroit Metropolitan Wayne County Airport (DTW) opened a new runway in December 2001. The new runway is parallel to three existing runways, as shown in Figure 1. (There are also two East-West runways at DTW, but they are rarely used because of prevailing winds, and therefore do not impact this study.) While the new runway increases DTW's potential capacity, the airport is served by an airspace (a set of routes, procedures, and controller assignments) that was designed for a three-runway airport. Consequently, DTW's four parallel runways could handle more traffic than the current airspace design allows.

To capitalize on this new runway capacity, the Detroit-area Terminal Radar Approach Control facility (D21 TRACON) and nearby Air Route Traffic Control Centers (ARTCC) are redesigning their airspace to accommodate more DTW arrival and departure traffic.

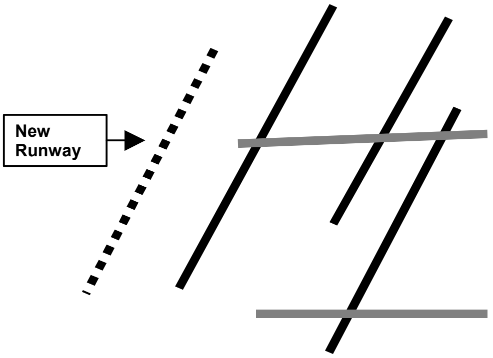

Figure 1: DTW Runways

This paper describes the simulation modeling effort to estimate delay, cost, and throughput benefits of the ARTCC redesign for DTW arrival traffic. The redesign work is part of the FAA's Midwest Airspace Capacity Enhancement (MACE) project, which is redesigning a large portion of the National Airspace System (NAS) in the Midwest.

2 THE SETTING

The D21 TRACON is responsible for arriving and departing flights within an approximately 40-nautical-mile radius around DTW. D21 typically handles flights within this radius when they are at altitudes below 12,000 feet. D21 currently runs a "four corner-post" operation for DTW arrivals; TRACON controllers receive flights from the Cleveland ARTCC at the four arrival fixes – CETUS, MIZAR, POLAR, and SPICA -- depicted in Figure 2. TRACON controllers deliver these arrivals to the final approaches, just before the arrival runways, where they are handed off to controllers in the airport tower.

Under the planned TRACON redesign, DTW will be served by five, rather than four, arrival fixes, also shown in Figure 2. The new fix, WEEDA will be located southeast of DTW; the old southeastern fix, CETUS, will be renamed GEMNI and moved several miles northeast.

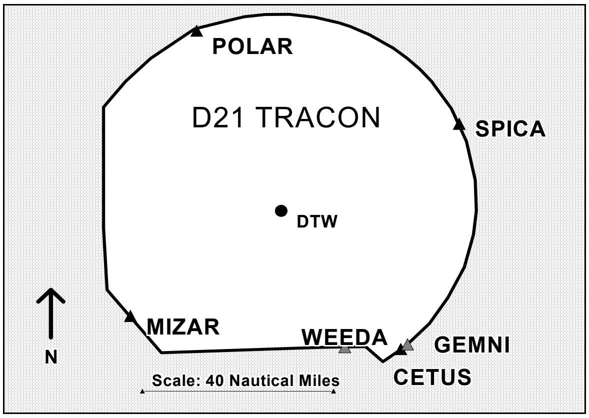

Figure 2: D21 TRACON with DTW Arrival Fixes

To feed this new five-arrival-fix TRACON, air traffic controllers from several Air Route Traffic Control Centers, (called "Centers"), redesigned their airspace. A Center, whose range of controlled airspace spans hundreds of miles, handles high-altitude, en-route traffic. Cleveland Center is the only Center that hands arrivals directly to D21 TRACON. However, the boundaries between the Indianapolis and Chicago Centers are fairly close to the D21 TRACON; therefore many DTW-bound flights spend relatively little time in Cleveland Center airspace before entering D21, as shown in Figure 3. Because of this proximity, the redesign effort required close cooperation between the three Centers.

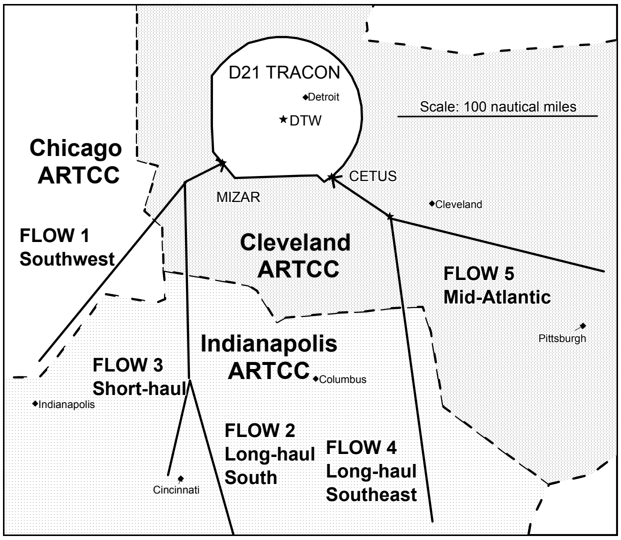

Figure 3: Current DTW Arrival Flows over Southern Fixes

2.1 Shifting Traffic Flows

Roughly speaking, a DTW arrival crosses the arrival fix closest to its origin city. For example, flights from Boston, which is northeast of Detroit, arrive over SPICA, the northeastern arrival fix, while flights from Dallas, which is southwest of Detroit, arrive over MIZAR, the southwestern arrival fix. With some minor exceptions, the traffic going over the two northern arrival fixes, POLAR and SPICA, will be unchanged in the redesign. The distribution of traffic arriving over the southern fixes, however will change substantially.

Broadly speaking, the DTW traffic currently arriving over the southern fixes can be divided into five flows (see Figure 3):

1. Traffic from the southwest that crosses through the Chicago Center before going over MIZAR.
2. Long-haul traffic from airports such as Atlanta, south of the Indianapolis Center, that crosses Indianapolis Center before going over MIZAR.
3. Shorter-haul traffic from airports such as Cincinnati, within Indianapolis Center, that go over MIZAR.
4. Traffic from the southeast (mainly Florida) that crosses Indianapolis Center before going over CETUS.
5. Traffic from the mid-Atlantic region, such as Washington and Baltimore, that travel a long distance in Cleveland Center before going over CETUS.

Under the future airspace design, flows 1 and 3 will still go to MIZAR and flow 5 will go to GEMNI (the replacement for CETUS), but flows 2 and 4 will go to the new fifth arrival fix, WEEDA. Figure 4 depicts the new routings.

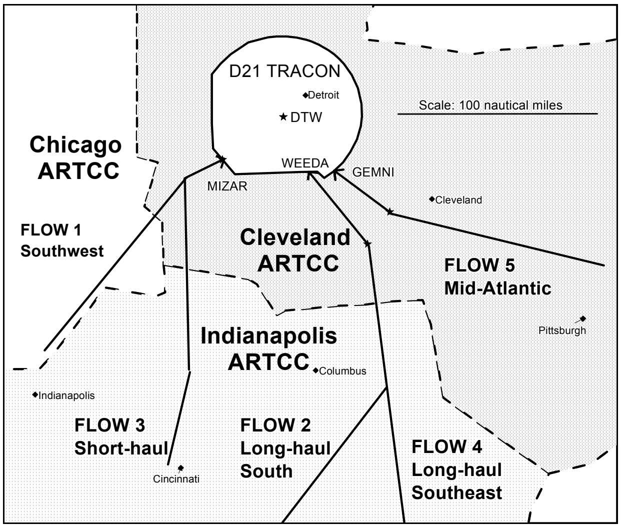

Figure 4: DTW Arrival Flows over Southern Fixes with MACE Redesign

3 RATIONALE FOR REDESIGN

Controllers merge flights by vectoring (turning) them to absorb delay and create the proper spacing between flights. Before handing jets off to D21 TRACON at the MIZAR waypoint, Cleveland Center controllers must merge jet flows 1 with flows 2 and 3 into a single flow at a single altitude

(Cleveland Center delivers propeller-driven aircraft to the TRACON at a lower altitude). During busy times of day, this merge presents problems to Cleveland Center controllers because there is not ample space in which to vector flights. As Figure 4 shows, flow 1 crosses from Chicago to Cleveland Center, and flows 2 and 3 cross from Indianapolis to Cleveland Center, fairly close to MIZAR. Cleveland Center controllers may impose Miles-In-Trail (MIT) restrictions on flights crossing these boundaries to ensure that the sector is not overwhelmed, which could lead to a potentially dangerous loss of separation. For instance, during busy times of day, Cleveland Center will require that flights from Chicago Center be spaced 20 miles apart. During quieter times of day, when no restriction is in place, flights at the same altitude may cross the boundary 7 to 10 miles apart, while flights at the different altitudes may cross simultaneously, separated only by altitude (in stacks).

Imposing MIT restrictions, however, places large delays on flights. Under the new airspace design, the rerouting of flow 2 from MIZAR means that fewer flights will have to be merged in the southwestern corner of Cleveland Center, so the difficulty of the merge will be reduced. This means that MIT restrictions will be imposed less frequently. For instance, 20-miles-in-trail restrictions may be imposed only twice per day (for 60 minutes each time), rather than five times per day. This reduction of MIT restrictions is the primary mechanism by which the MACE redesign reduces delay on DTW arrivals.

Under the new design, flows 2 and 4 will go primarily to WEEDA, and flow 5 will go primarily to GEMNI. If there is excess demand at one fix, however, controllers also have the option to offload flights from flows 2 and 4 to GEMNI, and flights from flow 5 to WEEDA. This offloading option, depicted in Figure 5, allows controllers to move excess traffic to the other, less busy, fix rather than pushing the delay back upstream.

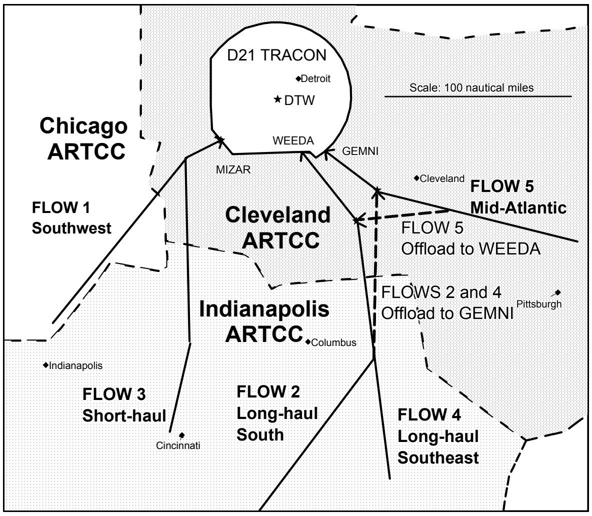

Figure 5: DTW Arrival Flows over Southern Fixes with MACE Redesign and Offload Options

4 THE STUDY

Because the goal of the study was to compare the ability of the Cleveland Center to deliver traffic to the D21 TRACON under the current airspace design and the MACE redesign, we made a number of simplifying assumptions.

- *Arrival fixes are constrained resources*
  We did not model the runways as constraints, and we assumed that once a flight crossed its arrival fix, it faced no constraints in the TRACON. We assume that with the fourth parallel runway, DTW and the D21 TRACON can handle the additional traffic it will get from the fifth arrival fix. The study assumes that the current throughput rates observed over MIZAR and CETUS can be sustained over MIZAR, WEEDA, and GEMNI, if the demand exists. Over each fix, jets and propeller flights cross independently of one another, because they cross at different altitudes.
- *Traffic over SPICA and POLAR will have little effect on our metrics*
  Because all of the flights that use the northern arrival fixes SPICA and POLAR in the baseline use the same routes under the redesign, we decided not to model them. Several flights that use MIZAR and CETUS in the baseline were rerouted to SPICA or POLAR in the MACE redesign case. The amount of delay these rerouted flights will cause, and face, at their new fixes should not be great, because most do not cross during peak times. However, we counted the distance flown by these flights in the redesign, even though they faced little delay on their new routes, because they had to fly a longer distance under the MACE redesign.
- *DTW departures and non-DTW traffic will have little effect on our metrics*
  This traffic was not expected to have a strong effect on the metrics in either the baseline or the redesign, and should not affect the two scenarios much differently. Therefore only arrival traffic for DTW was explicitly modeled.
- *Sector capacity is handled by boundary separations*
  The simulation adequately captures the main effects of sector capacity by modeling MIT restrictions on ARTCC boundaries, in-trail separation, and the separation requirements over the arrival fixes. Therefore we did not explicitly model sector capacity.

4.1 Traffic Selection, Current and Future

For our baseline traffic day (July 16, 2003), 360 flights crossed the MIZAR and CETUS arrival fixes. The distribution of the 360 flights included in the model, by engine type and fix, is given in Table 1. (As mentioned above, POLAR and SPICA are not modeled in the baseline.)

Table 1: Baseline Arrivals by Fix and Engine Type

Fix	Jets	Props
MIZAR	182	19
CETUS	141	18
POLAR	0	0
SPICA	0	0

Table 2 illustrates how these same 360 flights are distributed after the MACE redesign.

Table 2: MACE Arrivals by Fix and Engine Type

Fix	Jets	Props
MIZAR	149	19
GEMNI	96	3
WEEDA	67	11
POLAR	7	0
SPICA	4	4

Because DTW is a hub airport for Northwest Airlines, there are several peak periods of arrivals over the course of a day producing a saw-tooth pattern, as shown in Figure 6. To gauge the effects of the MACE redesign under future scenarios with more traffic, we added additional flights on the demand peaks, so that we had scenarios with 368, 376, 384, and 392 total flights. We did not create scenarios with additional flights during non-peak times, because the delay effects of such flights will be negligible. Figure 6 charts the number of simulated arrivals over the course of the day at DTW under the highest and lowest demand scenarios.

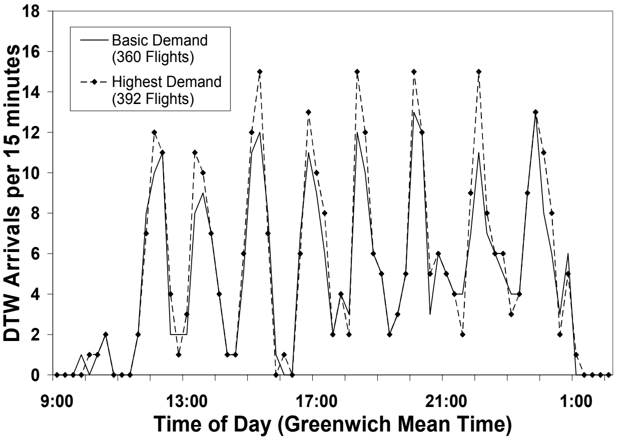

Figure 6: Number of DTW Arrivals from MIZAR and CETUS, per Fifteen Minutes, by Time of Day

4.2 Metric Selection

Because the average flight plan distance in the MACE redesign was slightly longer (in terms of planned mileage) than that in the baseline, comparing only delay between the baseline and the MACE redesign would be misleading and unfairly favor the MACE redesign. Instead, we compared total flight time (planned flight mileage and airborne delay). Although flight plans were slightly longer under the MACE redesign plan, the reduction in delay more than compensated for this increase.

4.3 Wind and Departure Time Adjustment

The study did not directly model the effects of wind. Wind can substantially change the transit time of a long flight, especially flights traveling with or against the jet stream. For instance, a flight from Baltimore to DTW will be slowed down by the wind, while a flight from Los Angeles to DTW will be speeded up by the wind. Because of this, and because the main modeling goal was to have flights demanding resources close to Detroit at the correct time, we did not use the actual observed departure times in the simulation. Instead, the departure times used in the simulation were adjusted so that the simulated time at which each flight crossed its D21 arrival fix matched the actual observed crossing time. The arrival fix crossing times were determined by analyzing radar data, called SAR data, from the ARTCC.

4.4 Calibration of Throughput over Arrival Fixes

We calibrated our model so that it matched the observed current delivery rate over the arrival fixes on the boundary between the ARTCC and TRACON. We used SAR data from July 16, 2003 to determine this delivery rate.

The model input parameters that we varied for calibration were the minimum allowed separation over the arrival fix and a stochastic noise factor -- generated randomly from a uniform distribution – that increased separation over the minimum.

Demand at a hub airport, and at the arrival fixes feeding them, is highly variable over the course of a day (see the saw-tooth pattern in Figure 6). Because demand over a fix is rarely sustained over a long period, we did not set our separation parameters by observing true hourly -- or even quarter-hourly-- rates. Calibrating by fixed-size time periods would understate the capacity of an arrival fix to handle spikes in traffic. On the other hand, we did not want to calibrate our model by looking only at the minimum observed separation between two flights, as this would overstate the sustainable capacity over the arrival fix.

Instead, we looked at longer series of flights as they crossed over the fixes. We determined the minimum separation (measured in time) among all pairs of jets (series of two) crossing the fix. We then measured the minimum separation (from first flight to last flight) of all series of three, four, and five jets crossing the fix. We adjusted the model's parameters until these minimum first-to-last separations in the simulation matched those in the radar data. We applied a similar methodology to propeller traffic.

We then applied these input parameter settings to the MACE redesign model, so that the maximum sustained rate over GEMNI, WEEDA, and MIZAR, (considered individually), were similar to the observed rates over CETUS and MIZAR.

4.5 Miles-In-Trail Restrictions in Baseline and MACE Redesign Models

In both the baseline and MACE redesign cases, the model represented MIT restrictions at the Cleveland Center boundary. Under the current airspace design, Cleveland Center imposes restrictions of 20 MIT on MIZAR-bound flights from Chicago and Indianapolis Centers five times per day, as shown in Figure 7. Cleveland Center also imposes restrictions of 20 MIT on CETUS-bound flights from Indianapolis, New York, and Washington Centers four times per day.

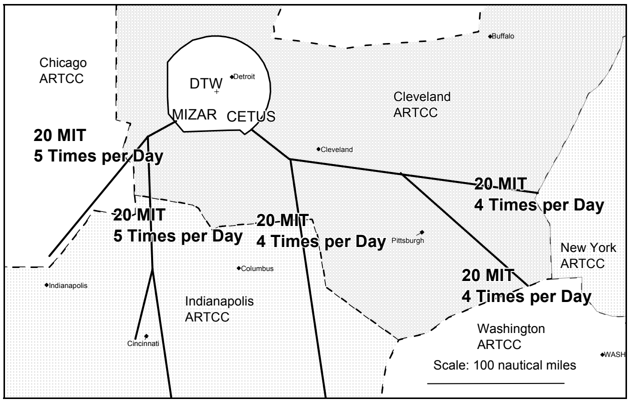

Figure 7: Current MIT Restrictions on DTW Arrivals Imposed by Cleveland ARTCC

Under the new MACE airspace design, Cleveland Center expects to impose restrictions of 20 MIT on MIZAR-bound flights from Chicago and Indianapolis Centers only twice per day, as shown in Figure 8. Cleveland Center plans no restrictions on flights to WEEDA, but will retain the restrictions on flights from Washington and New York Centers, which will go to GEMNI.

5 SIMULATION MODEL

The simulation model is based on a three-dimensional link-node network model of air traffic control (Boesel 2003). This model, which is written in the SLX simulation language (Henriksen 1998), explicitly accounts for the fact that airborne flights, unlike cars on the ground, can absorb only a very limited amount of delay. The model uses a look ahead time to anticipate contention for resources, such as busy merge points downstream, long before it occurs and spreads the required delay absorption out across the links. This prevents abrupt "hurry-up-and-wait" behavior, which is unrealistic in an air traffic scenario. Three object types define the model:

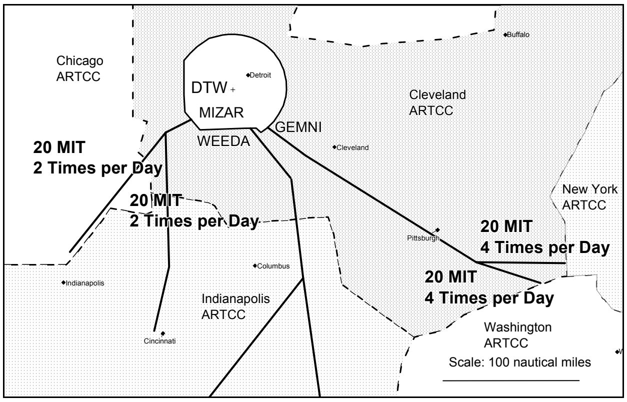

Figure 8: MIT Restrictions on DTW Arrivals Imposed by Cleveland ARTCC with MACE Redesign

- *Flight Object.* Each flight object represents a single flight. It has the flight's aircraft ID, aircraft type, desired departure time, and a flight plan, defined as a list of links.
- *Link Object.* Flights use links to get from one place to another. A link connects two points in three-dimensional space. Links are defined to be one-way only, and while a link can be shared by several flights, passing is not permitted on a link. Each link has pointers to all of the flights that will traverse it, and minimum required separations that define its capacity.
- *Flight-link Object.* A flight-link object represents a particular flight on a particular link. For example, if a flight has *n* links in its flight plan, then *n* flight-link objects will be created for that flight.

Each flight-link object has pointers to three other flight-link objects that define the object's relationship with the rest of the model. One pointer refers to the flight immediately ahead of it on the same link, and the other two pointers refer to the same flight on the next and previous links. Figures 9 and 10 illustrate these relationships.

Suppose flight B follows flight A across links h, j, and k, as shown in Figure 9, above. To represent this in the model, one would need flight objects for A and B, and link objects for h, j, and k. To represent the flights' movement over these links, one would need to create six flight-link objects, A_h, A_j, A_k, B_h, B_j, and B_k. Figure 10 illustrates the pointer relationships of flight-link B_j to its neighboring flight-link objects B_k (same flight, next link),

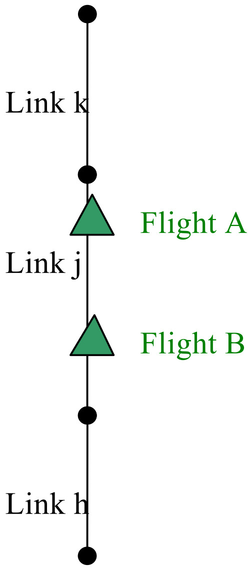

Figure 9: Flights A and B on Links h, j, and k

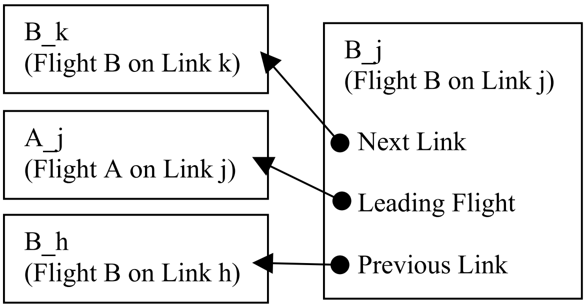

Figure 10: Flight-link Object B_j's Pointers to Neighboring Flight-link Objects

A_j (leading flight, same link), and B_h (same flight, previous link).

The model distributes computation to the flight-link objects. Every flight-link in the scenario is responsible for simulating its own state in the simulation. A number of distinct threads (called "pucks" in SLX) are established and activated, observing changes in simulation time and changes in the flight-links' neighbors. The state of a flight-link is contained in variables such as entry time (the time the aircraft starts flying on a link), exit time (the time the aircraft exits a link), and delay (amount of time currently absorbed on a flight-link). Each flight-link object also has two quantities -- minimum traversal time and maximum delay -- that determine the minimum and maximum amounts of time the flight can spend on the link.

A number of events can take place to cause a thread to react and change the current state of a flight-link. If there is a conflict at the merging of two flights' routes, one flight may be required to yield before reaching that merge point. There is a thread that is responsible for identifying this situation and will adjust the exit time for the flight-link appropriately. Delay on any given flight-link is automatically adjusted any time the exit time is changed. If the required delay for a single flight-link exceeds the maximum allowed, another thread will push back the entry time to this link, thereby causing the thread controlling the previous flight-link's exit time to adjust accordingly. This sequence of events propagates the effects of taking delay upstream in the system. Other threads are responsible for system integrity, such as ensuring that delay is not adjusted on any flight-links that the aircraft has already passed.

Although the schedule of flights on routes is provided at the start of the simulation, a look ahead is used to assign leading flights for each flight-link, well before the aircraft is scheduled to enter a link. This adjustable forecasting permits dynamic effects of delay to shift aircraft schedules and as a result, may assign a leading flight different than what was originally scheduled in the scenario.

The link-node model had to be adapted to assess the delay benefits of the MACE redesign. Specifically, the model was changed to allow it to represent miles-in-trail (MIT) restrictions and the ability of controllers to dynamically offload flights from one route to another.

5.1 Modeling Miles-In-Trail

The basic link-node model has two shortcomings that make it difficult to enforce MIT restrictions. First, the basic model could only separate aircraft whose routes placed them on the same links in the simulation; that is, one aircraft was leading the other, and separation could be maintained by virtue of their being located in the same place at different times. Miles-in-trail restrictions are applied to aircraft in different places (altitudes) at the same time. In the real world, if two flights, one at 25,000 feet (FL250) and one at 27,000 feet (FL270) want to cross an ARTCC boundary when no MIT restriction is in place, there is no need to separate the flights any further; altitude separation is enough, and one can fly directly above the other. The basic model, which would place these two flights on different links because of their different altitudes, was designed to represent this non-MIT situation. If, on the other hand, there is a 20-miles-in-trail restriction in place at the ARTCC boundary, the flights must be 20 miles apart when they cross the boundary, regardless of altitude separation. The basic model cannot represent this, because the links at the different altitudes cannot directly communicate with one another.

The second shortcoming of the basic model was that it had no mechanism for changing separation requirements as simulation time progressed. MIT restrictions are only in effect for relatively small time periods, for instance, 30 to 60 minutes, rather than for an entire day.

These issues related to MIT were addressed by adding a new object to the simulation that aggregated flight-links associated with each MIT time and location. With this added object, each flight-link now had the potential for two leading flights, a link-based leader (the earlier flight on the same link, as shown in Figure 10), and a MIT leader (earlier flight within the set of flight-links aggregated for a MIT restriction). Both these leaders had to be identified and assigned at the same time to prevent infinite looping. Often, both leaders for a flight-link were in fact the same flight. But if they were different flights, infinite looping was avoided because all the rules associated with taking delay from a leader were applied in a causally ordered manner. In other words, leader dependencies were resolved in strict upstream-to-downstream order for any given flight.

5.2 Modeling Dynamic Selection of Paths

To allow flights to choose different paths dynamically, based on downstream traffic congestion, we had to adapt the basic network model, which originally allowed flights to fly only one static path. We took advantage of the basic model's predictive infrastructure to build this capability. When a flight faces a choice between a default path and an alternate path (both diverging from point x and terminating at point y), the model predicts how long it would take to reach point y on each path. The flight chooses the default path unless the alternate path saves at least some user-specified amount of time (e.g. 90 seconds). The model incorporates this capability by replacing the next-link data item for each flight-link with a set of potential next links. A lookahead period is used to select the optimal choice from the set of next links before a leading flight is identified and assigned. Once a next-link is properly selected, any remaining unused paths are discarded, along with the threads belonging to the flight-links of those paths.

6 EXPERIMENTS AND RESULTS

With these new capabilities in hand, we created simulation models for the baseline and MACE designs, and replicated each experiment ten times. The main stochastic element in the model was flight departure time, which was allowed to vary ±2.5 minutes according to a uniform distribution, for each flight in each replication. This amount of variance was chosen to ensure that the model was not overly sensitive to a particular set of departure times.

Comparing the baseline to the MACE design on our sample traffic day with 360 flights, total flight time (including delay) decreased from a mean (over the 10 replications) of 27,632 minutes to 27,129 minutes. To estimate costs, the study assumed that all delay was taken in the air, rather than on the ground. Only direct costs, such as fuel and crew pay, were calculated; other costs, such as value of passenger time, were not included. Costs were calculated for each flight based on its aircraft type (Hoffer et al. 1998). For 360 flights the total mean daily cost savings were approximately $20,400. As the traffic was increased from 360 flights to 392 flights, mean daily cost savings increased to $25,200. Figure 11 shows the total mean flight times under the baseline and MACE designs for each of five traffic levels.

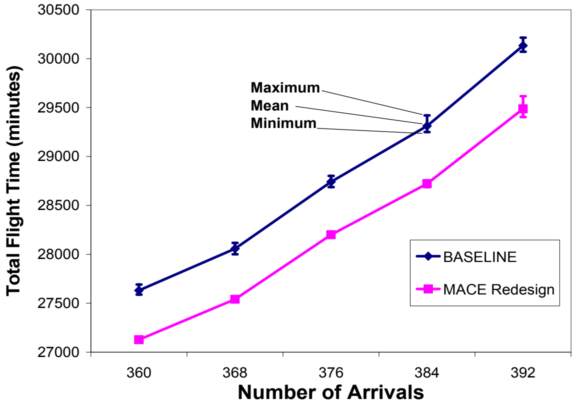

Figure 11: Total Flight Time, Including Delay, for DTW Arrivals, Baseline and MACE, Five Levels of Traffic

ACKNOWLEDGMENTS

The authors would like to thank: Rick Norris, John Harmon, and Mark Evans of the FAA (Great Lakes Region and Cleveland Center), for sharing their expertise on the MACE redesign; Thor Abrahamsen, Debra Pool, and Carla Gladstone of the MITRE Corporation for reviewing this article and for their guidance; and Ralf Mayer, of the MITRE Corporation, for developing the trajectory model used in this work.

This work was produced for the U.S. Government under Contract DTFA01-01-C-00001 and is subject to Federal Aviation Administration Acquisition Management System Clause 3.5-13, Rights in Data-General, Alt. III and Alt. IV (Oct., 1996).

The contents of this document reflect the views of the author and The MITRE Corporation and do not necessarily reflect the views of the FAA or the DOT. Neither the Federal Aviation Administration nor the Department of Transportation makes any warranty or guarantee, expressed or implied, concerning the content or accuracy of the views.

REFERENCES

Boesel, Justin. 2003. Simulating aircraft delay absorption. In *Proceedings of the 2003 Winter Simulation Conference,* ed. S. Chick, P. J. Sánchez, D. Ferrin, and D. J. Morrice, 1663-1669. Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Henriksen, James O. 1998. Stretching the boundaries of simulation software. In *Proceedings of the 1998 Winter Simulation Conference*, ed. Medeiros, D.J., E. Watson, M.S. Manivannan, and J. Carson, 227-234.

Piscataway, New Jersey: Institute of Electrical and Electronics Engineers.

Hoffer, Stefan, Frank Berardino, Jack Smith, and Stuart Rubin. 1998. *Economic Values for Evaluation of FAA Investment and Regulatory Decisions.* Technical Report FAA-APO-98-8. Office of Aviation Policy and Plans, Federal Aviation Administration, Washington, D.C.

AUTHOR BIOGRAPHIES

JUSTIN BOESEL is a Lead Simulation Modeling Engineer at The MITRE Corporation's Center for Advanced Aviation System Development (CAASD). He received his Ph.D. from Northwestern University and won the INFORMS George B. Dantzig Dissertation Award in 1999. He has been modeling air traffic control problems since 2000. His e-mail address is <`boesel@mitre.org`>.

DAVID BODOH is a Senior Systems Engineer at the MITRE Corporation's Center for Advanced Aviation System Development (CAASD). He has been developing distributed air traffic control simulation systems since 1991. His e-mail address is <`dbodoh@mitre.org`>.