

Modifications to the Design of the Multi-Center Traffic Management Advisor Distributed Scheduler

Steven Landry*

NASA Ames Research Center Moffett Field, CA 94035

Abstract

In a number of Air Route Traffic Control Centers (ARTCCs) around the United States, time-based metering of arrival aircraft is being used to control air traffic demand on busy airports. In attempting to introduce time-based metering for airports in the Northeast corridor (such as Philadelphia International Airport – PHL), several technical challenges were identified. Single-center Traffic Management Advisor (TMA), the most advanced decision support tool (DST) used for this purpose, is limited to managing traffic within one center, and to within about a 250 nautical mile (nm) radius. To overcome these restrictions, NASA engineers designed a distributed scheduling system, which uses a loosely coupled network of schedulers that can be extended across ARTCC boundaries and out beyond 400nm. The resulting design was incorporated into the Multi-center Traffic Management Advisor (McTMA), a traffic management DST whose implementation has undergone research and testing on PHL arrivals. This testing has yielded several key insights into the distributed scheduler, resulting in significant changes to the algorithm and underlying software. Three of these are discussed in this paper. First, the method used to account for and distribute unused capacity has been changed. Second, the resolution of the mechanism used to distribute restrictions has been increased. Lastly, an inconsistency in behavior between how close-in departures were being scheduled as compared to airborne aircraft was corrected.

I. Introduction

Time-based metering is a traffic management technique used to manage arrival demand during periods when it is expected to exceed the capacity of some National Airspace System (NAS) resource, such as a major airport. As such, metering is an alternative to strategies such as miles-in-trail (MIT) spacing or managed arrival reservoirs. Time-based metering uses automation to assign crossing times for aircraft at points along its route of flight. These “scheduled times of arrival” (STAs) are intended to delay aircraft, providing an efficient and orderly flow of traffic into the constrained resource, while also fully utilizing the available capacity of the resource.

The most widely accepted implementation of time-based metering is the Traffic Management Advisor – Single Center (TMA-SC). TMA-SC was designed and developed by researchers at NASA Ames Research Center in the mid-1990s as part of the Center-TRACON Automation System (CTAS). TMA-SC was handed off to the FAA and has since been deployed at eight Air Route Traffic Control Centers (ARTCCs): Los Angeles, Oakland, Denver, Minneapolis, Fort Worth, Atlanta, Miami, and Houston.

In considering how to expand time-based metering (and its benefits) to more congested airspace, researchers at NASA’s Ames Research Center (in collaboration with MITRE’s Center for Advanced Automation Systems Development) examined the feasibility of adapting TMA-SC to the Northeast corridor.

Several major problems were identified.¹ These problems required that a new architecture for time-based metering be developed for airspace such as the Northeast corridor. Researchers at NASA Ames, in concert with engineers at Computer Sciences Corporation, developed an innovative new architecture for time-based metering to overcome the limitations of the current technology for application to arrival metering within the congested Northeast corridor. This new architecture has been deployed as part of NASA’s Multi-center Traffic Management Advisor (McTMA). The focus of the McTMA research and evaluation effort is the Philadelphia International Airport (PHL), and includes the participation of four Air Route Traffic Control Centers (ARTCCs): New York (ZNY), Washington (ZDC), Cleveland (ZOB), and Boston (ZBW).

The approach makes use of a distributed network of loosely-coupled schedulers to distribute delay over a large area while mitigating errors in predicting trajectories over such distances. This distributed scheduling system is distinct from that used by TMA-SC in three key ways: (1) the system connects to any other McTMA systems, communicating over any network connection, (2) the system allows scheduling to any predefined point in the

*Aerospace Engineer, NASA Ames Research Center, Mailstop 210-6, Member AIAA

center’s airspace for any reason, and (3) the schedules at such points will also respect restrictions downstream of it, even if those restrictions originate in other centers.

The design of the distributed scheduling system was discussed in some detail in a previous paper.² Therefore, only a short description of the initial design will be provided here; rather, this paper will concentrate on the results of validation testing and changes to the initial design that resulted from that testing.

II. Distributed scheduling system design

The distributed scheduling system is a network of schedulers that pass back restrictions in the form of rates. A downstream scheduler creates a schedule, and then, for each upstream scheduling point, passes back the maximum number of aircraft that it can accept from those points within a period of time.

It is important to note that upstream schedulers do not need to assign aircraft to those rates in lockstep with the downstream scheduler; rather, each scheduler is free to choose which aircraft end up satisfying the rate. This allows sequence changes and schedule fluctuations to occur downstream without affecting upstream schedules. By allowing the downstream scheduler to sequence aircraft over its own (shorter) horizon, rather than over its flights’ entire duration, upstream schedules are somewhat isolated from errors in estimated times of arrival (ETAs) over long distances.

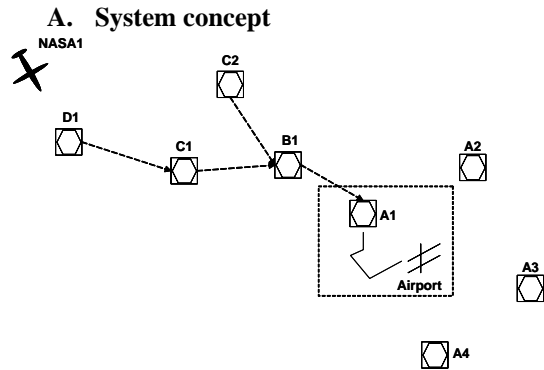


Figure 1. Generic metering topology.

To explain the design, a generic airport arrival route is shown in Figure 1, consisting of a runway and a number of meter points (D1, C1, C2, B1, and A1). A similar sequence of points would occur upstream of points A2 through A4, but, for the purposes of the example, only the A1 branch will be examined.

Consider an aircraft (NASA1) going from D1 to C1 to B1 to A1 to the runway. ETAs for NASA1 are calculated for the points along its route of flight, including D1, C1, B1, A1, and the runway. Similarly, ETAs are calculated for all the arrival aircraft flying in to these points.

The scheduler for the airport calculates restrictions with which A1, B1, C1, and D1 should comply in scheduling NASA1; A1 would calculate similar constraints for points upstream of it, and so on. The constraints are generic (i.e. not

attached to particular aircraft) so that particular sequences need not be adhered to by the controller in meeting these constraints.

B. Rate profiles

The constraints that are passed between schedulers are called “rate profiles”. Each rate profile contains a sequence of rates. These rates identify the number of aircraft that can be accepted from a particular stream in a given period of time. It is calculated by counting the number of scheduled aircraft that originated from a particular place upstream which have been scheduled at that point. An example is shown in Figure 2 for B1’s rate profile for point C1 and C2 (see Figure 1 for the geometry of these points).

Figure 2 shows a timeline from 15:00 to 15:15 for the point B1 that is fed by points C1 and C2. The timeline is read from the bottom, with the earliest time (15:00) at the bottom of the timeline, and the latest time (15:15) at the top of the timeline. The left side of the timeline indicates ETAs; the right side indicates STAs. The arrows on the right side of the timeline, for example, indicate the STAs for various aircraft, and the labels to the right of those arrows associate the source of the aircraft (either C1 or C2). Normally, this label would indicate

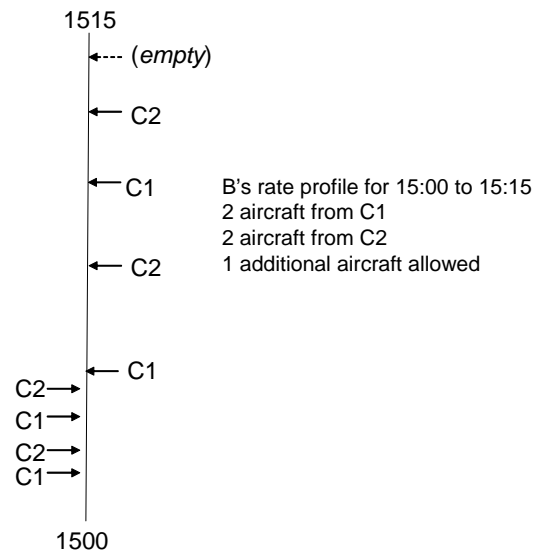


Figure 2. Example of rate profile.

the aircraft identification, but in this example only the source is relevant.

Since these aircraft have been successfully scheduled, the count shown in Figure 2 (2 aircraft from C1 and 2 from C2) represents the minimum number of aircraft from C1 and C2 respectively that can arrive at B1 within the given time period. Any unused capacity of this time period can also be determined and identified to all upstream points as being available. Such unused capacity, which may or may not be available for a given time period at a given point, is indicated in Figure 2 by the “empty” label. In this way the total capacity of a meter point is divided between streams entering the fix.

Such a time period is called a “bin,” which has a start time and duration. Bins can be labeled using the hour and an index, so that the bin from 15:00 to 15:15 would be called bin 15-01.

III. Testing distributed scheduling system design

The distributed scheduling system has been implemented as part of the McTMA fielded software. It has undergone testing in the lab by McTMA engineers as well as field testing by traffic management and sector controller cadre members.

A. Lab observation

The McTMA system can be observed in live operation (connected to live air traffic feeds) in the Center-TRACON Automation System Verification & Validation lab at NASA Ames Research Center. In the lab, the distributed scheduler has been evaluated by examining the accuracy of the amount of delay passed back to each tier. The amount of delay between any two tiers should be approximately equal to the “allowable maximum delay time” (AMDT) between those two tiers. AMDTs, typically between 2 and 5 minutes, are preset values which are read in by the software at start up. These values are determined by air traffic controllers familiar with the sectors in that tier and represents the amount of delay that the controllers in that sector can be realistically expected to impart to aircraft through vectoring and speed reductions.

The amount of delay assigned to aircraft can be varied by placing additional, manual restrictions on a particular meter point, or at the airport. Under normal operation, McTMA restricts traffic upstream from an airport based only upon the FAA mandated minimum separation standards. For operational reasons, such as to reduce congestion within certain sectors or at certain points, additional restrictions can be placed at any point. For example, the required miles-in-trail (MIT) between aircraft at some meter point can be increased from the FAA mandated minimum of 5 nautical miles (nm) for co-altitude aircraft. The scheduler would indicate how much delay a trailing aircraft needs in order to meet the new MIT separation at the meter point. Other restrictions that can be modified for scheduling purposes are: acceptance rates (at meter points and at the airport), blocked intervals at the airport (periods of time where no aircraft are allowed to arrive), the required spacing on final approach (normally equal to the minimum wake vortex separation), and the runway occupancy time.

B. Field observations

In addition to these lab evaluations, the McTMA system has undergone periodic field evaluation by air traffic control personnel in the four centers over the last year. These personnel (called cadre members) have undergone special training in the design and use of McTMA.

During these field evaluations, cadre members “shadowed” the system, which receives radar data from the FAA’s HOST computer system in the four ARTCCs (ZNY, ZDC, ZOB, and ZBW). The system’s displays reside in the Traffic Management Unit (TMU) of these four centers, which allowed the cadre to view the McTMA system while having access to their normal sources of information about traffic flow. The cadre evaluated McTMA predictions and schedules during traffic rushes with McTMA in operation but without using McTMA to make operational decisions. These field evaluations were extremely valuable in identifying and diagnosing system anomalies.

The cadre members involved in these shadowing exercises were connected via telecon to the other centers in order to coordinate system settings and to discuss interesting or unusual indications by the system during busy traffic periods. They also logged departure time performance and system discrepancies.

IV. Results

From the results of the above testing, several important findings about the system were made which affect the scheduler design. These findings have since been addressed by algorithmic changes to the distributed scheduler software.

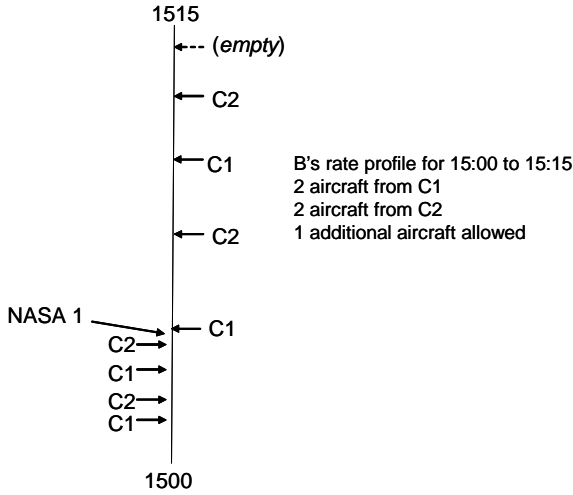


Figure 3. STA error example.

example, the delay required into C1 would be 1 minute. However, after passing C1, NASA1 will suddenly need an additional 14 minutes of delay. These 14 minutes represent STA error.

The error can be as large as the duration of a bin. The only algorithmic solutions to this problem would involve more tightly coupling upstream STAs to downstream STAs, which would increase the reliance on ETAs over long distances. This is undesirable. Therefore, the error can only be controlled by reducing the size of the bin. However, as the bin size is reduced, the messages which transmit the rate profile information across the network will increase in size. Although no bandwidth problems have been identified to date, it is desired to minimize the bandwidth used by the system. Therefore, a balance between message size and STA error needs to be maintained. Since controllers are only required to meet STAs to within 1 minute, it was decided that STA error of 1 minute was acceptable, and the bin size was changed to that value.

B. Bin hole accounting modifications

“Holes” represent additional capacity at a meter point. Between any two aircraft, such additional capacity may exist, depending on the spacing requirements and other restrictions. If the only restriction on the meter point is MIT, the number of holes in a gap between two aircraft can be determined by the following equation:

$$\frac{\text{Separation_in_miles}}{\text{MIT_restriction}} - 1 \quad (1)$$

So, for example, if 10 miles are required between aircraft, and there is a 40 mile gap between two aircraft, 3 additional aircraft can be placed in that gap. For the first iteration of the distributed scheduler, it was assumed that this capacity need only then be allocated to each bin within the gap. In order to determine the number of holes allowed in each bin within that gap, interpolation is required, using the formula shown in equation 2:

$$\text{bin_factor} = \frac{\min(\text{gap_end}, \text{bin_end}) - \max(\text{gap_start}, \text{bin_start})}{\text{gap_end} - \text{gap_start}} \quad (2)$$

The result of equation 2 was multiplied by the result of equation 1 for each bin, and giving the number of holes in each bin for that aircraft pair. The sum of the results of these multiplications for each pair of aircraft gave the total number of holes for each bin. This sum was truncated within each bin to give an integer number of holes.

A. STA error equal to bin size

When the system was run with a rate profile bin size of 15 minutes, significant errors were noted. It was noticed that the STAs at a given point could differ from that required to meet the downstream rate profiles. These STA errors, whose extent were up to one bin (in minutes), were going uncorrected by the system. As an example, consider an aircraft such as NASA1 in Figure 3, which needs to be delayed to be behind all the C aircraft on the right side of the timeline (into the “empty” slot). NASA1’s current ETA is 14:59 (in bin 14-4, which is from 14:45 to 15:00), but must be delayed 16 minutes so as to arrive at 15:15 (which is in bin 15-01). However, the rate profile system only requires that NASA1 be delayed so as to be *somewhere* in bin 15-01, and the upstream scheduler has no information about where in bin 15-01 NASA1 needs to be. It therefore delays it as little as possible, delaying it 1 minute so that it will arrive at the start of bin 15-01 (15:00). If NASA1 is currently flying to C1, for

However, one effect of this truncation is that fractional slots may be lost. This effect is heightened when the bin size is made small, as necessitated by the STA error being proportional to bin size as described above. For a given bin size, groundspeed, and MIT restriction, no holes will result from the above equations regardless of whether there are *any* aircraft if the following equation is true:

$$\frac{bin_size \times speed}{MIT_restriction} < 1 \quad (3)$$

So, for a 20 MIT restriction and a speed of 6 miles per minute, the bin size must be greater than 3:20 for any holes to appear, regardless of the spacing between aircraft.

1. *Errors in initial bin hole accounting*

An example of this is shown in Figure 4. For two aircraft spaced 48 miles apart in a stream that requires 20 MIT separation, it is apparent that one hole should exist in a bin somewhere between the two aircraft. Yet with one minute bins, this will not occur, as is seen by the calculations shown on Figure 4. Each of the bin hole counts rounds to zero, since the maximum value they can obtain is 1/8.

Current calculations:

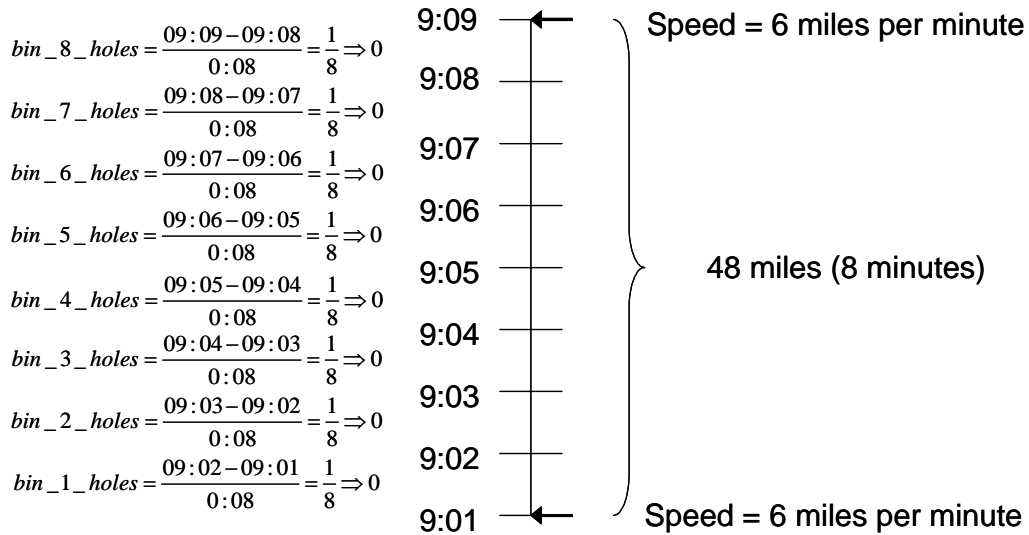


Figure 4. Current bin hole accounting example.

Despite the bin hole accounting, it is known that an aircraft can fit between these two aircraft and still have the 20 MIT separation. The current accounting scheme gives no indication where that hole should be. In contrast, Figure 5 shows where an aircraft can be and still have twenty miles in trail from the other aircraft on the timeline.

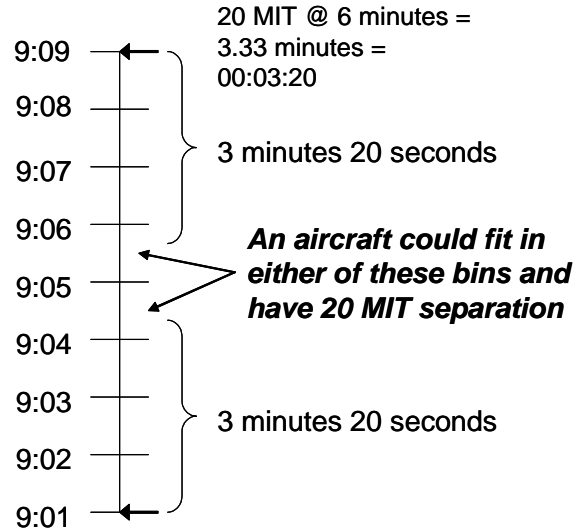


Figure 5. Actual hole locations example.

2. Initial solution to bin hole accounting errors

The example shown in Figure 5 led to a possible solution which was implemented initially, although this method was altered after a simplification was identified. This initial solution roughly used a technique of determining where there are open slots between two aircraft and then counting up those slots on a per-bin basis. Figure 6 gives an example of this approach.

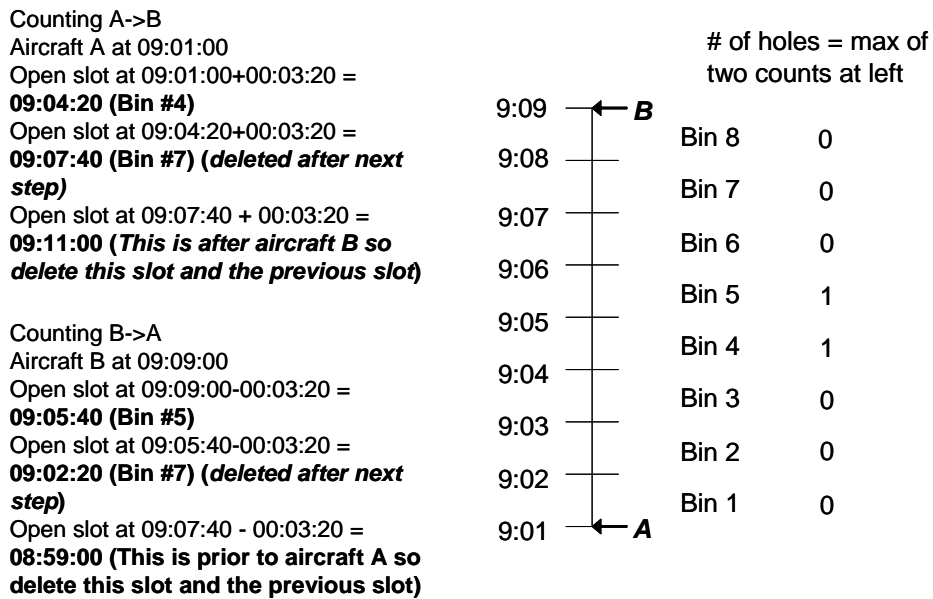


Figure 6. Second approach to counting holes example.

For each pair of aircraft, two counts are made. For the first, start at the earlier aircraft (A) and add the amount of time required between subsequent aircraft (3 minutes 20 seconds). This is repeated until the time that is being checked is equal to or after the next aircraft (B). If this time is after the next aircraft, then delete that slot and the previous one. If it is equal to the STA of the next aircraft, then just delete that one slot (not the previous one). This “deletion” is to make sure partial slots between two aircraft are not counted.

For the second count, start at the later aircraft (B) and subtract the amount of time required between subsequent aircraft. Similar to above, one would continue finding slots until the time that is being checked is equal to or prior to the previous aircraft. Again, if prior to the previous aircraft, then that slot and the previous one should be deleted; if equal, delete only the current slot.

The slots identified by the two counts are identified by time. They can therefore be attributed to bins. Note that the two counts may not be the same (as is the case in Figure 6). The count to be used is the **maximum** of the two counts, as reflected by the totals shown in the right column in Figure 6.

Note that this means that two holes appear, when only one aircraft can be accommodated. However, that one aircraft could appear in **either** bin. Once it is scheduled in one or the other, the other hole will disappear. Therefore, while two holes exist, only one aircraft will be allowed between these two aircraft.

3. Final bin hole accounting method

Upon examination of the algorithm, an additional simplification was added to remove the need to make two counts which then needed to be reconciled. Instead of essentially counting where holes can be, this method explicitly identifies them. First, the sizes of the holes in question are calculated using the following equation:

$$size_of_hole_sec = \left(\frac{miles_in_trail_nm}{groundspeed_knots} \right) \times 3600sec / hour \quad (4)$$

The number of holes in the gap in question can then be calculated using equation 5:

$$num_holes_in_gap = \left(\frac{size_of_gap_sec}{size_of_hole_sec} \right) \quad (5)$$

This calculation provides the number of holes that can be put in the gap. This count then needs to be allocated across each of the bins within the gap. The first bin in which a hole can exist is the STA which represents the front of the gap plus $size_of_hole_sec$. The last bin in which a hole can exist is the STA which represents the end of the gap minus $size_of_hole_sec$. This is shown in Figure 7.

Between this first and last bin, the number of holes in each bin must be determined. This is accomplished using equation 6. Note that equation 6 is again likely to be less than one. In this case, any non-zero result means there is potentially some capacity in that bin. Therefore, the result of equation 6 is rounded up rather than truncated.

$$num_holes_per_bin = \frac{num_holes_in_gap}{\left(\frac{size_of_gap_sec}{bin_size_sec} \right)} \quad (6)$$

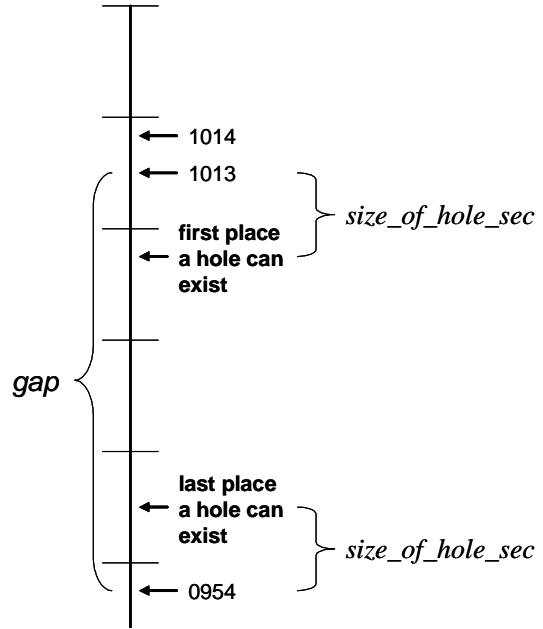


Figure 7. Identification of location of first and last holes.

C. Internal departure ETA calculation

McTMA also provides departure time advisories for aircraft bound for the airport being metered that depart from specified airports within the four McTMA ARTCCs. Such aircraft, called internal departures, have departure times that are controlled by the ARTCC in which the departure airport resides. The current method of controlling these departure times is called the “approval request” (APREQ) process. When APREQ is in effect, towers of the departure airports call when an aircraft destined for the affected airport is ready to take off. The ARTCC then identifies where in the overhead stream of aircraft that departure can fit and manually back-calculates a departure

time for that aircraft. The ARTCC advises the tower of that time, and the tower ensures that the aircraft meets that take off time.

McTMA can calculate APREQ departure times in the same manner as it calculates STAs for airborne aircraft, and indicate a departure time that would not require more delay from any sector in excess of the specified AMDTs. Both SC-TMA and McTMA made one distinction between internal departures and airborne aircraft, however. The delay for an airborne airplane at any particular point represents the amount of delay to be taken between that point and the aircraft's current point. For internal departures, this would include ground delay. Traffic managers, however, did not want to see ground delay being portrayed at a meter point. Moreover, it was desired that internal departures have no airborne delay to its first point, to prevent the aircraft from taking significant departure delay, only to begin taking more delay once airborne. These two factors caused developers to re-establish the ETA at the first meter point after an internal departure was scheduled. The previous ETA, based on the aircraft's flight planned departure time, would be changed to be equal to the scheduled departure time plus the calculated flight time to the first point. However, this new ETA included delay not already taken (the ground delay of the departure), unlike airborne aircraft, whose ETA assumes no delay which has not already been taken.

The effect of this was to create erroneous sequences and STA error. An example is shown in Figure 8, where ETAs are again on the left side of the timeline, and STAs are shown on the right. In Figure 8a, the aircraft labeled "D" is a departure, and is able to arrive between aircraft 1 and 2. According to the first-come, first-served rule, the proper sequence for the STAs is therefore aircraft 1, followed by aircraft D, followed by aircraft 2. The schedule assigns STAs based on that, as shown in Figure 8b. If the ETA is then updated to reflect its new arrival time (Figure 8c), however, the STAs are out of sequence with the ETAs, and the scheduler resequences aircraft 2 and aircraft D, as shown in Figure 8d. Recall that aircraft D is able to arrive prior to aircraft 2. By updating D's ETA, aircraft D has been given additional delay.

This occurs because the ETA for aircraft D shown in Figure 8c is a "delayed" ETA, whereas the ETAs for aircraft 1 and 2 in the same figure are "undelayed." This inconsistency has been corrected, and D's ETA is no longer updated until the aircraft is actually taking delay on the ground.

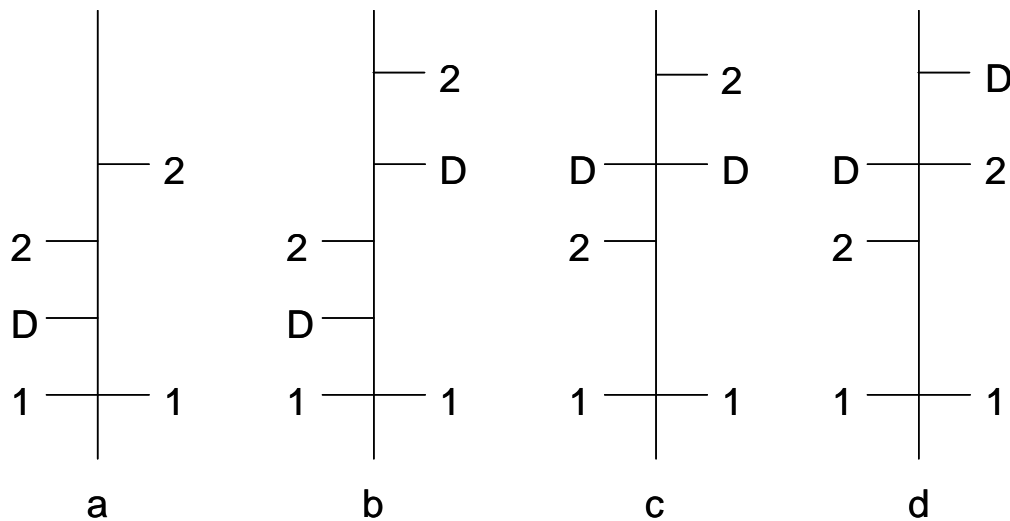


Figure 8. Illustration of departure bumping problem.

V. Future activities

An operational test of the departure scheduler and TMU operations is planned for October 2004, with an operational metering test to follow in late spring 2005. For the former test, traffic managers will utilize the McTMA departure advisories for PHL in place of their manual APREQ process, which is currently done without automation support. McTMA will provide a departure time that will not only provide proper separation within the flow of airborne aircraft, but also a departure time that will not overly burden downstream resources (such as the PHL airport).

Efforts to analyze recorded data are also underway. There is an extensive amount of data recorded by the McTMA system, including aircraft state data, flight plans, ETAs and STAs, and interprocess messages. Using this data, the McTMA team hopes to gather metrics of ETA accuracy and of the performance of the distributed scheduling system.

In addition to this analysis, and in cooperation with the Regional Metering initiative at NASA Ames, over the next year we will examine several key issues which bear on the ability to use McTMA as a general metering tool for the ARTCC. The end goal is to create a system which can meter generic streams of aircraft in an ARTCC for any reason, including to ARTCCs that do not have a McTMA system. This would mean, for example, that ZBW and ZOB could cooperate on metering their flow of traffic into Chicago-O'Hare airport. Another example would be to depart aircraft into an overhead stream (not dependent upon destination). A final example would be to generate restrictions based on sector congestion.

VI. Conclusions

McTMA is a time-based metering DST which was developed as improvements to Single-center TMA in order to expand time-based metering benefits to the congested Northeast corridor. In order to do so, a distributed scheduling architecture was developed which significantly enhanced the TMA system. This system allows multiple instances of TMA to share data and has the ability to schedule aircraft to metering points other than those associated with a TRACON, with those metering points respecting restrictions imposed on it by other connected TMA systems. The initial design of the system has been undergoing laboratory and field evaluations in order to validate the concept and architecture.

These evaluations discovered several design errors. Corrections to these errors have resulted in a number of fundamental changes to the system. Among these are the reduction of the rate profile bin size to constrain STA error, corrections to the way additional capacity is calculated, and a change to the way departure ETAs are calculated. These changes have undergone initial testing and appear to have significantly improved the system.

Additional analyses will occur after the upcoming tests, including an operational TMU test. The goal is to conduct an operational metering test involving the four McTMA ARTCCs cooperatively metering to PHL. This test is tentatively scheduled for late spring 2005.

¹ Farley, T., J. Foster, T. Hoang, and Lee, K. (2001). "A time-based approach to metering arrival traffic to Philadelphia," First AIAA Aircraft Technology, Integration, and Operations Forum, Los Angeles, CA.

² Landry, S. J., Farley, T., Foster, J., Green, S., Hoang, T., & Wong, G. (2003). Distributed Scheduling Architecture for Multi-Center Time-Based Metering. Proceedings of the AIAA's 3rd Annual Aviation Technology, Integration, and Operations (ATIO) Technical Forum.