Comparison of Methods for Passenger Flow Simulation of an Airport Terminal

GÁBOR KOVÁCS, ISTVÁN HARMATI  
BÁLINT KISS, GÁBOR VÁMOS  
Budapest University of Technology and Economics  
Department of Control Engineering and Informatics  
Magyar Tudósok krt. 2, H1117 Budapest  
HUNGARY  
{gkovacs,harmati,bkiss,vamos}@iit.bme.hu  

PÉTER MARÁCZY  
Budapest Airport Ltd.  
BUD International Airport  
H1185 Budapest  
HUNGARY  
peter.maraczy@bud.hu

Abstract: This paper presents a comparison of two methods for passenger flow modeling of an airport terminal. The two approaches, based on store-and-forward models and Petri nets, both have their own benefits and drawbacks. The paper presents the fundamentals of the two methods, their application to airport terminal passenger flow modeling, and then gives a comparison based on simulation results obtained on a small-scale terminal model.

Key Words: Simulation, Passenger flow modeling, Petri nets

1 Introduction

Air traffic has been rising significantly in the last decades, demanding the improvement of passenger handling at airport terminals. The most crowded airport, London Heathrow (UK) accommodates a daily average of 150,000 passengers. Such an increased passenger flow might lead to several problems: increase of security threat, delay of connecting flights, passengers missing their flights etc. Moreover, IATA classifies airports based on factors like passenger density and transfer times, so improvement of the passenger flow is of paramount importance during airport planning, development and reconstruction [1].

As the route of passengers through a terminal can hardly be changed, airport development needs suitable methods for passenger flow simulation during the phase of planning. Such simulation methods might help not just in finding optimal passenger paths, but also in discovering bottlenecks or designing adequate emergency evacuation routes [2],[3].

This paper presents two methods for passenger flow modeling adapted to airport terminals. The store-and-forward model, based on nonlinear difference equations, provides a macroscopic view on passenger flow. The Petri net based method is a microscopic one, which allows exact tracing of flow-related events and passenger distribution.

The remaining part of the paper is as follows. Section 2 presents the modeling of an airport passenger terminal based on functional cells. Section 3 introduces the store-and-forward model, while Section 4 discusses the Petri net-based approach. Section 5 presents simulation results on the same small-scale terminal model using the two methods and compares their properties. Section 6 concludes the paper.

2 Cell-based modeling of a passenger terminal

Investigating floor plans of different airport terminals shows that several regions of space open to passengers share similar properties and passenger behavior. Therefore, in order to provide a convenient way for building models of different airport terminals, fundamental building blocks of passenger terminals, referred to as cells in the sequel, should be constructed. Cell types include check-in counters and kiosks, security screening checkpoints, shops, bars and restaurants, lounges and corridors etc.

After constructing individual models for these cells, a directed graph with cell models as its nodes and arcs representing possible passenger flow can be constructed. A branching rate is assigned to each arc, representing that passengers choose the cell to proceed freely from reachable neighboring ones. It is straightforward that the sum of the branching rates of arcs leaving a cell must be exactly one.

Figure 1 presents a model of the passenger terminal of a small airport, used for the evaluation of the methods. Passengers arrive from the entry hall after check-in, and proceed to security screening. After the screening procedure, passengers arrive to the main hall, from where they might proceed directly to
the boarding gates, or visit one of the shops.

3 Store-and-forward model

3.1 Store-and-forward models

Store-and-forward models are based on difference equations and are widely used in urban and highway traffic modeling and control [4],[5]. These models are sampled ones, i.e. equations are evaluated at time instances \( kT \), where \( T \) is the sampling time.

For each cell, an input queue and a functional unit is defined. The functional unit represents actions taken in the current cell (e.g. screening of passengers, sell of goods at a store), while the input queue represents passengers waiting for the given action.

The general store-and-forward model of the \( i \)th cell is defined formally as follows:

\[
\begin{align*}
\tau_i(k) &= x_i(k-1) + T \sum_j t_{ji} u_j - \sum_k T t_{ik} u_i \\
x_i(k) &= \max(\min(Q_i, \tau_i(k)), 0)
\end{align*}
\]

(1)

where \( x_i(k) \) is the number of passengers in the cell at the time instance \( kT \), \( t_{ij} \) is the branching rate of passengers leaving cell \( i \) and entering cell \( j \) with \( \sum_i t_{ij} = \sum_j t_{ij} = 1 \) and \( t_{ij} = 0 \) if there is no direct path from cell \( i \) to cell \( j \). The parameter \( u_i \) represents the processing rate of the cell \( i \) in passenger per minute (PAX/min). The second equation assures that the number of passengers in the cell will be between zero and the capacity of the cell, denoted by \( Q_i \).

The processing speed of a cell might depend on several factors. Model-dependent parameters of the \( i \)th cell (e.g. processing time of a check-in kiosk) are denoted by \( p_{ij} \). Decision variables, which provide an interface for controlling the passenger flow (e.g. number of operating check-in kiosks) are given by \( d_{ij} \). Parameters and decision variables influence the passenger flow via the processing speed of the cell, i.e.

\[ u_i(k) = f(x_i(k), p_{i1}, \ldots, d_{i1}(k), \ldots) \]

(2)

Note that \( f \) can be any arbitrary positive semidefinite nonlinear function.

3.2 Cell models

3.2.1 Security screening

The processing speed of a screening checkpoint is denoted by \( p_{11} \) while the decision variable corresponding to the number of the operating checkpoints is \( d_{11} \). The processing speed of the cell is as follows:

\[
\begin{align*}
u_1 &= \begin{cases} 
\min(\frac{x_1(k)}{T}, p_{11} d_{11}) & \text{if } x_2(k) < Q_2 \\
0 & \text{if } x_2(k) \geq Q_2
\end{cases}
\end{align*}
\]

(3)

Then, by denoting the speed of incoming passenger flow by \( v_1 \), the number of passengers in the security screening cell is given by

\[
\begin{align*}
\tau_1(k) &= x_1(k-1) + T v_1 - T u_1 \\
x_1(k) &= \max(\min(Q_1, \tau_1(k)), 0)
\end{align*}
\]

(4)

3.2.2 Hall

The hall serves as an area for passengers to move between other cells, therefore its parameters correspond to the speed of passenger flow. Parameter \( p_{21} \) denotes the minimal speed of flow (in PAX/min) in case the hall is saturated, while \( p_{22} \) corresponds to the maximal free flow speed. Parameter \( p_{23} \) describes the limit of free flow, i.e. the number of passengers under which the flow speed is considered to be \( p_{21} \). The processing speed of the hall is as follows:

\[
\begin{align*}
u_2 &= p_{21} + \left[ 1 - \frac{\max(x_2(k) - p_{23}, 0)}{Q_2 - p_{23}} \right] (p_{22} - p_{21}) \\
\end{align*}
\]

(5)

Then the state equation of the cell reads

\[
\begin{align*}
\tau_2(k) &= x_2(k-1) + T u_1 - T (t_{23} + t_{24} + t_{25}) u_2 \\
x_2(k) &= \max(\min(Q_2, \tau_2(k)), 0)
\end{align*}
\]

(6)

3.2.3 Shop

Passengers entering the shop might become customers or just look around and leave without buying anything. The ratio of customers is defined by \( p_{33} \in [0, 1] \). Processing speed of a cashier is given by \( p_{31} \) (PAX/min) while average time spent in the shop in sampling times is defined by \( p_{32} \). The decision variable \( d_{31} \) denotes the number of cashier’s desks in operation. The processing speed of the shop is as follows.

\[
\begin{align*}
u_3 &= \min(p_{32} x_3(k), p_{31} d_{31}) + (1 - p_{33}) \frac{x_3(k - \text{floor}(p_{32}))}{T} \\
\end{align*}
\]

(7)

The state equation of the shop cell reads

\[
\begin{align*}
\tau_3(k) &= x_3(k-1) + T p_{23} u_2 - T u_3 \\
x_3(k) &= \max(\min(Q_3, \tau_3(k)), 0)
\end{align*}
\]

(8)

Equations for the second shop follow by substituting the adequate parameters.
4 Petri net model

4.1 Petri nets

Petri nets are bipartite place-transition graphs [6]. A place $p_i$ is an input place of a transition $t_j$ if there is an arc from $p_i$ to $t_j$ and an output place vice versa. Generalized Petri nets have weights (positive integers) associated to each arc. Two matrices, the input and output incidence matrices $I^-$ and $I^+$ with as many rows as places and as many columns as transitions can be defined such that the element $I^+_{i,j}$ contains the weight of the arc $p_i \rightarrow t_j$, while the element $I^-_{i,j}$ is the weight of the arc $t_j \rightarrow p_i$.

Each place of the net contains a discrete number of markings, referred to as tokens, which might represent resources or, in case of passenger flow modeling, passengers themselves. The marking of the place $p_i$ is denoted by $m_i$. The state of the net is given by the marking vector, containing markings of each state: $M = [m_1, \ldots, m_n]$. Dynamics of the net is described by the flow of tokens. A transition $t_i$ is enabled if each of its input places contain at least as many tokens as the weight of the arc leading from the given place to the transition, i.e. $m_j \geq w(p_j,t_i), \forall p_j \in P$. If a transition is enabled, it might fire. Firing of a transition $t_i$ removes $I^+_{i,j}$ tokens from its input places and adds $I^-_{i,j}$ tokens to its output places. A firing vector $s$ might be constructed with elements corresponding to the firing numbers transitions (a transition can fire multiple times). The marking of the net after the firing is given by $M' = M + (I^+ - I^-)s$, i.e. Petri net dynamics can be given by a system of difference equations. Note that these equations are evaluated asynchronously, at each time a transition is enabled to fire.

If two transitions $t_i$ and $t_j$ share at least one common input place and are both enabled, but their simultaneous firing is not enabled by the actual marking, then the two transitions are said to be in conflict. In order to choose a transition to fire, a random decision based on the branching rates $P(t_j) \in [0,1]$ will be made among the enabled transitions.

Time is included by assigning a sojourn time $\tau_i$ (possibly zero) to each place. If a token is added to a place $p_i$ at time $t$, it is not available for firing (i.e. it can not be counted when determining enabled transitions) before time $t + \tau_i$.

4.2 Cell models

4.2.1 Entry hall

The Entry hall cell serves as a generator of tokens (i.e. passengers) for the security screening cell. As shown by Figure 2, place $p_1$ is the generator, and its sojourn time $\tau_1$ defines the time between the appearance of two passengers. The subnet composed of places $p_2$, $p_3$ and $p_4$ serves as a guard condition for token generation. While there is a token at $p_4$, tokens in $p_1$ can pass to $p_5$ through transition $t_1$. When $t_3$ is fired, the token at $p_4$ is removed, so $t_1$ will be disabled until the firing of $t_2$. This structure is used to set the length of initial passenger flow during the simulation.

4.2.2 Security screening

Passengers willing to enter the terminal are represented by tokens at the place $p_5$, from where they can move only through transition $t_4$. Note that $t_4$ has an other input place, $p_{12}$. Marking of $p_{12}$ corresponds to the number of free places in the cell (and therefore its initial marking $m_{12,0}$ gives the capacity of the cell).

Place $p_6$ corresponds to the queue waiting for the screening procedure, represented by the place $p_{r7}$. The screening procedure requires a free checkpoint, i.e. a token at place $p_8$. Note that when the procedure is finished (transition $t_{6}$), a token is placed to $p_8$ denoting that a checkpoint has been freed.

Figure 2: Petri net model of the passenger terminal
Place \( p_0 \) represents the exit of the screening zone, where passengers put on their shoes and rearrange their bags after the procedure. This zone also has a limited capacity, represented by the initial marking of \( p_{10} \). From the Security screening cell, passengers move towards the hall (transition \( t_7 \)).

### 4.2.3 Hall

The hall itself is represented by two places, \( p_{12} \) and \( p_{13} \). The marking of \( p_{12} \) is the number of passengers at the hall, while marking of \( p_{13} \) is the difference of the hall capacity and the marking of \( p_{12} \). From the hall, passengers can move on towards the shops (transitions \( t_{10} \) and \( t_{16} \), latter not depicted on the figure) or towards the boarding gates (transition \( t_{22} \)).

The subnet containing places \( p_{14} \), \( p_{15} \) and \( p_{16} \) serve to ensure change of branching rates in case of boarding. The announcement of boarding is represented by transition \( t_9 \), which removes a token from \( p_{15} \), representing that there is no flight boarding, and \( p_{16} \), serving as a guard condition for \( t_{10} \) and \( t_{16} \). These guards ensure that passengers do not enter shops after the boarding has been announced. Length of the call for boarding is given by \( \tau_{15} \).

### 4.2.4 Shop

Passengers enter Shop \#1 through transition \( t_{10} \). Tokens at \( p_{18} \) represent passengers looking around and browsing amongst the goods. They shall consider to pass to the exit of the shop without buying anything (transition \( t_{11} \)), or proceed to the cashier’s desk (transition \( t_{13} \)), where the queue is represented by the place \( p_{20} \). The procedure of payment corresponds to place \( p_{21} \), which needs a free cashier, represented by a token at \( p_{22} \). From the exit of the shop (\( p_{19} \)), passengers return to the hall through transition \( t_{12} \).

In order to make Figure 2 more compact, Shop \#2 was omitted. However, its structure and connection to the hall is the same as in case of Shop \#1.

### 5 Simulation results

Simulations were carried out on the same test scenario for both models using the terminal layout of Figure 1.

The length of the simulation was chosen to be 150 minutes. Passengers arrive in the first 90 minutes, while the call for boarding is active from \( t = 120 \) min to \( t = 150 \) min. Branching rates are defined as \( t_{23} = P(t_{10}) = 0.6, t_{24} = P(t_{16}) = 0.35 \) and \( t_{25} = P(t_{22}) = 0.05 \). After the boarding call, branching rates of the store-and-forward model are changed to \( p_{23} = p_{24} = 0 \) and \( p_{25} = 1 \), representing that all passengers move towards the boarding gate.

<table>
<thead>
<tr>
<th>Screening cell capacity</th>
<th>( Q_1 = 50 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_{11,0} = 50 )</td>
<td>( m_{10,0} = 3 )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Screening speed/time</th>
<th>( p_{11} = 1 ) PAX/min</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau_{11} \in [30, 120] ) sec</td>
<td></td>
</tr>
<tr>
<td>( \tau_{10} \in [30, 90] ) sec</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Checkpoints in operation</th>
<th>( d_{11} = 3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_{8,0} = 3 )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hall capacity</th>
<th>( Q_2 = 100 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_{13,0} = 100 )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Hall speed/sojourn time</th>
<th>( p_{21} = 60 ) PAX/min</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau_{12} \in [60, 300] ) sec</td>
<td></td>
</tr>
<tr>
<td>( p_{22} = 100 ) PAX/min</td>
<td></td>
</tr>
<tr>
<td>( p_{23} = 70 ) PAX</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Checkpoints in operation</th>
<th>( d_{31} = 2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_{22,0} = 2 )</td>
<td></td>
</tr>
<tr>
<td>( m_{28,0} = 1 )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Shop capacities</th>
<th>( Q_3 = 80 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_{17,0} = 80 )</td>
<td></td>
</tr>
<tr>
<td>( Q_4 = 20 )</td>
<td></td>
</tr>
<tr>
<td>( m_{23,0} = 20 )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cashier speed/payment time</th>
<th>( p_{31} = 5 ) PAX/min</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_{21,0} \in [30, 90] )</td>
<td></td>
</tr>
<tr>
<td>( p_{41} = 1 ) PAX/min</td>
<td></td>
</tr>
<tr>
<td>( m_{21,0} \in [60, 140] )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Browsing times</th>
<th>( p_{32} = 1 ) T</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \tau_{18} \in [180, 420] ) sec</td>
<td></td>
</tr>
<tr>
<td>( p_{42} = 1 ) T</td>
<td></td>
</tr>
<tr>
<td>( \tau_{24} \in [30, 60] ) sec</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Customer rates</th>
<th>( p_{33} = 0.3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( P(t_{13}) = 0.3 )</td>
<td></td>
</tr>
<tr>
<td>( P(t_{13}) = 0.7 )</td>
<td></td>
</tr>
<tr>
<td>( p_{43} = 0.9 )</td>
<td></td>
</tr>
<tr>
<td>( P(t_{19}) = 0.9 )</td>
<td></td>
</tr>
<tr>
<td>( P(t_{17}) = 0.1 )</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Cashier’s desks in operation</th>
<th>( d_{31} = 2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m_{22,0} = 2 )</td>
<td></td>
</tr>
<tr>
<td>( d_{31} = 2 )</td>
<td></td>
</tr>
<tr>
<td>( m_{28,0} = 1 )</td>
<td></td>
</tr>
</tbody>
</table>

Table 1: Parameters of the models

Number of passengers in each cell is given by Figure 3 and 4, while processing speeds or flow rates (in passenger/minute) are presented by Figure 5 and 6 for the store-and-forward and the PN models, respectively. From a qualitative point of view, results seem similar, i.e. the shape of the plots are the same for the two methods, but interesting differences can be found in the details. The first difference is that the plots of the simulation using the store-and-forward model are much smoother, resulting from the fact that the model is a sampled one, so high frequency changes observable in case of PN-based simulation are filtered.
Figure 3: Number of passengers in cells using the store-and-forward model (min vs. PAX)

Figure 4: Number of passengers in cells using the Petri net model (min vs. PAX)

Figure 5: Processing speed of cells using the store-and-forward model (min vs. PAX/min)

Figure 6: Processing speed of cells using the Petri net model (min vs. PAX/min)
Comparing the plots of the Security screening cell, the more detailed PN model shows that the cell was working at the limit of its saturation, but has not been fully saturated as shown by Figure 3. Ramps of the passenger number curves are also different for the two models, due to the different processing speeds. Comparing curves of Figure 5 and Figure 6, it can be seen that stochastic behavior of the PN model causes oscillation, while the processing speed of the store-and-forward model is constant from 5 to 80 minutes. Oscillations after 80 minutes are related to the fact that the Hall has become crowded.

The largest difference between the two methods can be observed in case of the Hall. The number of passengers is about the double in case of the store-and-forward model compared to the PN model, however no significant differences can be found in the processing speeds. This phenomenon is related to the different parameters used in the models. Unlike in case of other cells, processing speeds and sojourn times can hardly be complied without additional information: the relation between the processing speed and the sojourn time depends largely on the size of the hall and other physical parameters. Note that the dashed curve of Figure 5 shows the theoretical processing speed \( \nu_2 \) of the cell. At \( t = 60 \) min the number of passengers reaches the limit of free flow, so the processing speed decreases.

Both methods show increasing number of passengers in Shop #1, up to the limit of saturation. Shop #2 becomes saturated at around 30 minutes according to both simulations, and remains fully crowded until the call for boarding. Note that the PN model shows slight oscillation, revealing that place of the leaving passengers are filled not immediately, but after a few moments. Both figures show that the passenger flow towards the boarding gates significantly increases after the announcement of boarding, i.e. \( t = 120 \) min.

Simulations were carried out in Matlab environment. The simulation time was 1 millisecond for the store-and-forward model and 2.54 sec for the PN model on a laptop computer with a Core 2 Dual processor, which shows a significant benefit of the former method. Also, while the computational time of the store-and-forward model depends on the number of cells and the sampling time, the computational time of the PN model depends largely on the number of tokens, i.e. the number of passengers.

### 6 Conclusion

Two methods for passenger flow modeling were evaluated on the example of a small-scale terminal model. Results obtained by simulations based on store-and-forward and Petri net based models are similar, however, some differences arise from the details of modeling principles. The Petri net model is more detailed, providing a way to include non-deterministic timing parameters, but its computational need rises with the number of passengers. Store-and-forward models need low computational resources, however their accuracy is limited due to their sampled behavior. Therefore a store-and-forward model based simulation might serve for a draft macroscopic evaluation of passenger flow, while final, accurate results should be deduced from the simulation of the corresponding Petri net model.

Future work includes evaluation of the methods on large-scale models, and improving their modeling capabilities by introducing variable step size for store-and-forward models and colored Petri nets.

**Acknowledgments:** This work was supported in part by the Hungarian National Scientific Research Foundation grant OTKA K71762.

Also, it is connected to the scientific program of the “Development of quality-oriented and harmonized R+D+I strategy and functional model at BME” project, supported by the New Hungary Development Plan (Project ID: TMOP-4.2.1/B-09/1/KMR-2010-0002).

Authors would like to thank the staff of Budapest Airport Ltd. the information provided on the operation of airport passenger terminals.

**References:**


