Proceedings of the 2005 Winter Simulation Conference
M. E. Kuhl, N. M. Steiger, F. B. Armstrong, and J. A. Joines, eds.

FLUID BASED SIMULATION MODEL FOR HIGH VOLUME DC CONVEYOR SYSTEMS

Ying Wang
Chen Zhou

Industrial & Systems Engineering Department
765 Ferst Drive, NW
Georgia Institute of Technology
Atlanta, GA 30332-0205,U.S.A.

ABSTRACT

In this paper, we present a fluid simulation methodology applying to high volume large conveyor networks operating in a slowly changing environment, often found in large distribution centers. Traditional discrete-event cell-based approach to simulate such networks becomes computationally challenging due to large number of events resulting from high WIP level, complex network and large conveyor footprint. The fluid simulation model is built on a Petri-Net based framework. We present the model and investigate the feasibility in modeling capability in terms of input and release control logics, performance evaluation and computational savings.

1 INTRODUCTION

To excel in today's fierce marketplace, companies make a constant effort to optimize their supply chain configuration. Large warehouses and distribution centers (DCs) emerge to meet the ever-growing customer base or to take advantage of the economy of scale. Automated conveyor system is an inevitable choice when a facility's throughput reaches certain level. The conveyor provides connectivity to large area and buffering to streamline the material flow between operations and processes. These conveyor systems are characterized by large footprint, complex configurations, high-throughput, high WIP and high cost. The design and control of these systems have significant impact on the system performance such as throughput, order response time and operating cost.

However, planning, dimensioning and configuration of these large conveyor networks is an challenging task. Analytical model is not available or requires unrealistic assumptions. Simulation model can fit the bill but a detailed model takes long time to develop and even longer time to run. The conveyor simulation constructs used in the state-of-art simulation languages, such as Witness, Arena, and Automod are cell-based discrete-event representation of conveyor systems. Conveyors are divided into cells and system tracks whether each cell is occupied or not. Each time a package moves from one cell to another, an event is triggered. In high volume large logistics hubs, hundreds or thousands of items move along tens of kilometers of complex conveyor system; the triggering events are very frequent. As a result, the simulation run time can become prohibitive.

Closer observation of such system reveals that different behaviors in the system are on different time scales. For example, the time for item interarrival is in seconds, while the time for rate change in loading or unloading is usually in minutes or tens of minutes. This type of system can be modeled as a network operating in a slowly changing environment which means the number of environment transitions is much fewer than the number of changes of the system state. We describe the longer time scale behavior by states of environment. The change in the state of environment triggers the change of interarrival rate or routing control parameters. In certain environment states, the system appears unstable in the sense that arriving rate exceeds the service rate. However, conveyors can provide the buffering and stability. The queueing network analysis of such system is touched by Choudhury and Mendelbarm et al. 1997 and Chang 2004. They show that the queueing systems in a random slowly changing environment can be approximated by stochastic fluid model. This motivates us to use fluid simulation to analyze high volume conveyor network in hope to reduce the computational burden of the simulation.

Fluid simulation has been developed in computer network paradigm to cope with the today's network growth in size and complexity (Kesidis et al.1996, Kumaran et al.1998). A fluid simulator models an Asynchronous Transfer Mode (ATM) network as some fluid sources followed by a set of fluid bandwidth schedulers linked with constant propagation delay. The fluid emitting rates from these sources are modeled as piece-wise constant functions. Thus in turn, at any time a set of input rates, a set of output rates and current buffer content can completely describe the status of a bandwidth switch. A fluid-based scheduling policy is used at each switch to determine the output rates of different sources from input rates. Some comparison between efficiencies of fluid simulation and packet-level simulation also has been done (Liu et al. 1999, Liu et al. 2001). Their primary conclusion is that the fluid simulation will generally outperform packet-level simulation for the simple network. As the network size and complexity grow, the fluid simulation suffers from the so-called "ripple effect" which makes the fluid simulation less efficient.

Applying fluid simulation on conveyor network is actually quite different from applying it on telecommunication network. The main reason is that the transportation along the conveyor can not be modeled as a constant delay due to accumulation and blocking phenomena, instead a time-space two dimensional formalism for describing the material flow evolution along the conveyor is needed to capture the requested operating detail of the system.

In a recent paper, we introduced a conveyor network fluid simulation implementation which uses an adaptation to Batches Petri Nets(BPN) called Stochastic Batches Petri Net (S-BPN) as an modeling framework (Wang and Zhou 2004). BPN, an extension of Hybrid Petri Nets, was first introduced by Demongodin and Prunet 1993 to model hybrid production system with continuous transfer elements. The fluid simulation treats discrete items in continuous batches characterized in length and density. These batches are assembled, disassembled, merged, diverged and directed in the system. The batch formalism of BPN which is a mathematical formalism of the parts flow circulation on a conveyor segment provides a good foundation for keeping track of batch evolution. Petri Net framework also provides natural transformation from model to a discrete event simulator. To make BPN a suitable fluid simulation framework for complex conveyor network, some features in BPN need to be modified and new features need to be added. These issues is addressed in S-BPN definition.

In this paper, we extends S-BPN further to make it more applicable to model real world system. We also describe the network model components that will be used as building blocks to construct complex network. The benefits for performing fluid simulation will also be illustrated. The paper is organized as follows. In section 2, the definition of S-BPN is given with some extensions. In section 3, we investigate the feasibility of utilizing S_BPN model as a fluid simulation framework from various perspectives. A concluding remark along with the further research area is given in section 4.

2 STOCHASTIC BATCHES PETRI NETS (S-BPN)

2.1 Summary of S-BPN Definition

In S-BPN, a conveyor segment is defined as a continuous element with input and output flow. It is characterized by length, driving speed and maximum density of parts. An *internal coherent Batch* (ICB) represents a set of items with the same density of repartition on the conveyance system. At a fixed instant *t*, ICB of batch *n* at time *t* is characterized by $[\ l^n, d^n, x^n\ ]$ to indicate the length, density and the beginning position of the batch. An *Output Internal Coherent Batch* (OICB) is an internal coherent batch whose beginning position is equal to the length of the batch place.

S-BPN consists of three disjoint sets of places: the discrete places P_d with a non-negative integer number of tokens, continuous places P_c with a real non-negative number that represents the fluid level in the buffer and batch places P_b that represents the conveyance segments. The marking of the batch place is an ordered set of ICBs. Transitions are also partitioned into a subset of discrete transitions and a subset of batch transitions. A discrete transition can be an immediate transition or a timed transition. We allow deterministic or stochastic timing structure associated to timed discrete transitions. Batch transitions have a maximum firing flow (MFF) associated to them.

The graphic symbolism of S-BPN is the following:

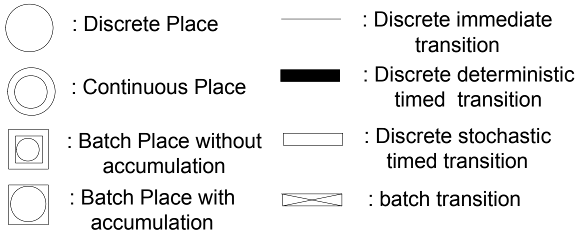

Figure 1. The Nodes of S-BPN

Generally speaking, the discrete part of the net which consists of discrete places and discrete transitions were used to model the external environment state and external control, while the continuous part of the net models the material flow. Interested readers are referred to Wang and Zhou 2004 for weight condition and enabling and firing rule definition of S-BPN.

S-BPN evolution can be described as a discrete event dynamic system that leads itself to simulation. The discrete event model evolves through a sequence of macro-states upon the occurrence of macro-events. Within a macro state, the fluid is drained from the input continuous places and batch places of a batch transition and pumped into the output continuous and batch places. The fluid content of a continuous place varies linearly with time, while the marking updating of a batch place is governed by a set of batch formalism which describes the six concepts of transformation along a conveyor segment: creation, motion, consolidation, accumulation ,collision and destruction.

Calculating instantaneous firing flow (IFF) vector in each macro state is the most complex step in S-BPN simulation. The complexity comes from the following three issues:

1. The IFF of a batch transition is the minimum of three values : its MFF value, the minimum speed constraints posted by pre-places and the minimum of the speed constraints posted by its post-places.
2. Specific merge or diverge release control logic is reflected as different way to allocate the merge/diverge capacity in IFF calculation.
3. An empty continuous place or a full batch place induces a cyclic dependence between its total upstream IFF and total downstream IFF. This dependence requires an iterative method to calculate IFFs.

Readers are referred to Wang and Zhou 2004 for the complete algorithm.

2.2 Extensions to S-BPN

In the following, we summarize some new features we introduce to enhance the S-BPN modeling capacity.

1. In BPN or S-BPN, all continuous places have infinite capacities. We extends this by associating a maximum buffer capacity to every continuous place. Accordingly, the characteristic function of continuous place is defined as $c(p_i) = b_i$, if $p_i \in P_c$. Finite capacity buffer can be modeled with this modification. In our initial S-BPN, we did not have explicit construct to handle the monitoring of accumulation levels. While automatic merge release logics are often based on the accumulation levels. To facilitate the implementation we added updates 2-4 .
2. The characteristics function of batch places is expanded to include accumulation type and sensor concepts. $c(p_i) =\{acc_i,l_i,v_i(t),d_i,Cap(i)\}$, if $p_i\in P_b$, where acc_i ='A' or 'N' indicating it is an accumulation segment or non-accumulation segment and l_i, $v_i(t)$, d_i are length , speed function and maximum density respectively. $Cap(i)= \{Cap_1(i),\ldots,Cap_n(i)\}$ is a finite set of sensors. $Cap_j(i) = \{t_j,x_j\}$is a sensor j associated with the batch place p_i with:
   -x_j is the position of the sensor ($x_j \le l_i$)
   -t_j is the type of the sensor and belongs to the set {a,d,e},that is {accumulation sensor, detection sensor, empty sensor}
   Sensor value function is defined for accumulation sensor as: $V(Cap_j(i)) = 1$ if the accumulation passed the sensor position and $V(Cap_j(i)) = 0$ otherwise.
   Sensor value function is defined for detective sensor as: $V(Cap_j(i)) = 1$ if there is an internal coherent batch at the sensor position and $V(Cap_j(i)) = 0$ otherwise.
   For empty sensor ,it is defined as: $V(Cap_j(i)) = 1$ if the batch place contains no internal coherent batches and $V(Cap_j(i)) = 0$ otherwise.
3. The weight of an arc linking a batch place to a discrete transition is a specific sensor of this batch place. We also call it a *sensor arc*. Sensor definition was included in an extension of BPN, called Controlled BPN (Audry and Prunet 1995). But there is no internal link between these sensor status to some control places. We define the sensor arc so that some automatic release logic that based on sensor status can be modeled.
4. The arc linking to a discrete transition can be regular arc or inhibit arc (with a small circle at the end). A necessary condition for a discrete transition to fire is the weight condition or sensor condition are not satisfied in all input places that link with it through inhibit arcs. Inhibit arc does not consume tokens when it fires.
5. The MFF of batch transition and driving speed of batch place are allowed to change over time. This change facilitates the integrating of fluid simulation with other external simulation units. For example, a loading capacity function over time pre-generated by a stochastic model can be fed into the fluid simulation for performance evaluation.
6. The structure condition that every batch place must have a continuous place which limits the capacity is relaxed. Instead, we define the full condition of a accumulation batch place as an event. This new feature allow S-BPN to model complex conveyor network in a more simplified and transparent manner than original representation.

With these new features added, the evolution rule needs to be changed accordingly. We includes the new enabling rule, firing rule and macro-event definitions in the Appendix.

3 FEASIBILITY AND BENEFITS OF FLUID SIMULATION

In this section we examine the feasibility of implementing fluid-simulation and illustrate some benefits of it when it is applicable. We address the problem from modeling capabilities in terms of input and control logic, performance measures and computation time.

3.1 Input Model

The process of inducting items into the conveyor system are abstracted into input models. The characteristics of input models and available performance measures determine the pros and cons between item-level simulation and fluid simulation. A slow changing environment is a key factor that justifies using fluid simulation. That means the time between rate changes should occur at least an order-of-magnitude longer than time between item interarrivals. Under this condition, the number of events is expected to be reduced drastically in S-BPN based fluid simulation thus offset the longer single event processing time in fluid simulation. In addition, the queueing effect between rate changes is more pronounced than the variability effect of individual arrivals. Therefore, the piece-wise constant sample-path approximation approach can maintain accurate.

S-BPN's flexibility allow it to model many arrival processes. Here, we provide two arrival model examples shown on Figure 2.

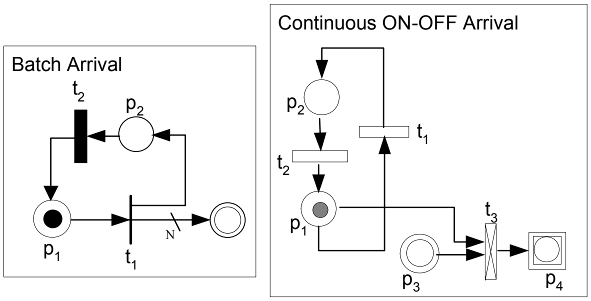

Figure 2. Sample Arrival Models

Batch Arrival models the item arrival in batches at a fixed time or random timed interval. For example, during the course of a day, the inbound trucks arrive according to a modeled schedule; once arrived, the packages are unloaded in a batch to a temporary buffer in a modeled rate. Continuous On-Off Arrival model can be used to model an operator's work schedule with or without some random nature. In Figure 2, transition t_3 can be considered as a loading operator, his working schedule is modeled by discrete part of the system which comprises discrete places p_1 , p_2 and discrete transitions t_1, t_2. If the arrival pattern is too complex to be modeled using discrete Petri Nets, we can use an external model to generate the sample path of the arrival pattern and feed it to the S-BPN model through time-varying MFF function.

3.2 Control Logic

There are many control logics at work in a conveyor system. In this paper, we only present some merge release control logics. We modeled three commonly used automatic merge release logics in distribution conveyor system into priority merge, processor share merge and time slice merge. We also show how to integrate these policies with accumulation control strategy utilizing the accumulation position information sent by sensors.

Priority Merge (PM) models the following: the items from the lower priority input conveyor can only merge when they do not impede the passing of items from the high priority input conveyors. An example application is the merge from a branch to the main. In S-BPN, this is implemented by setting the weights of arcs that link the pre batch transitions to the merge place as the priority numbers and apply PM merge capacity allocation algorithm in IFF calculation.

Processor Share Merge (SM) models the merge junction when each induction conveyor merges on an alternating or round robin basis, as long as there are objects available; an induction conveyor with no package just simply skipped. Suppose a merge place p is shared by J input sources and operates at a fixed rate r, the information needed to implement the SM merge policy is contained in the weight vector $\alpha = (\alpha_1,.., \alpha_J)$ which we assigned as the arc weights. When all sources have accumulation, source j is allotted a fraction α_j of the merge capacity. When some sources achieve no accumulation using less than their allotted capacity, the remaining merge capacity is split among the other sources in proportion to their α_j's. This SM allocation algorithm will be applied in IFF calculation.

Time Slice Merge models the situation where each input line is released for a pre-defined time period before switching to another line as long as it has positive flow. An induction conveyor with no package is just simply skipped. A high-speed saw-tooth merge often adopts this logic to increase the merge speed. Since at any time point, only one pre-transition of the merge place is enabled , it is not treated as a merging unit in S-BPN. Instead, the control logic is enforced through discrete part of the net.

Figure 3 shows the S-BPN of the Time Slice Merge release. The flows merge from two accumulation conveyor segment modeled by batch places p_1 and p_2 to a non-accumulation segment modeled as place p_3. Transition t_1 and t_2 becomes enabled alternatively according to the time delay set through timed transition t_3 and t_4. The inhibit arc linking from p_1 to t_4 will disable transition t_4 when p_1 becomes empty, so that the token will be kept at place p_5, thus flow from p_2 can transferred to p_3 uninterruptedly through transition t_2. Another sensor arc achieves the same result when p_2 becomes empty.

A pre-set release logic often changes according to the accumulation position of the input conveyors. For example, a common used strategy works as follows: The merge subsystem works at a predefined merge logic when all infeed lines accumulation are below 75%. If a line reaches the 75% level, the merge subsystem controller automatically prioritizes that line as the next line to release. The 75% full line runs until the condition no longer exists, and then it resumes its normal release sequence. If one or more in-feed lines also reach 75% level during this period time, these multiple lines will share the merge capacity according to predefined logic. Figure 4 shows the S-BPN model for this control strategy.

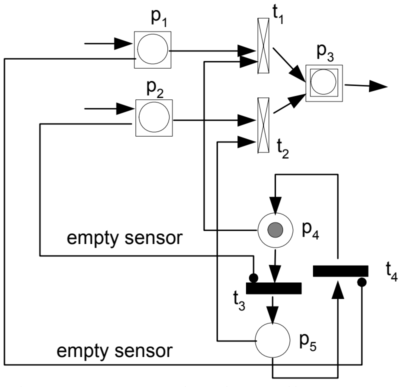

Figure 3. Example Time Slice Merge

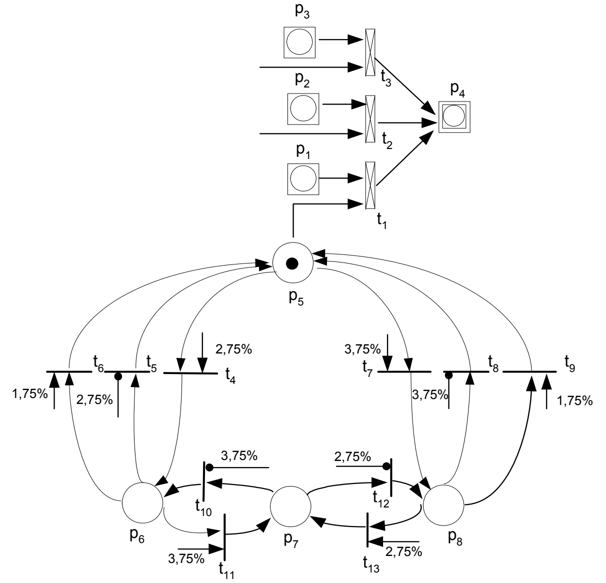

Figure 4. Example Accumulation Sensitive Merge Logic

In Figure 4, to save space, we only drew one discrete part to control the pre merge transition t_1. t_2 and t_3 are controlled similarly as t_1. Also we did not draw the sensor arc completely, instead we use a label to indicate the sensor position and associated batch place. For example, 2,75% means the 75% accumulation sensor of batch place 2. Initially, a token is place at place p_5. So transition t_1 is enabled and it will participate in the merge capacity allocation. At a time, if accumulation level of p_2 reaches 75%, the token will be transferred from p_5 to p_6. t_1 is disabled. Same logic is also applied to the discrete part that control t_3. Both t_1 and t_3 will be withdrew from the merge capacity allocation. The merge subsystem is dedicated to transition t_2. After the accumulation level of p_2 drop from 75% level, the token is transferred back to p_5 through transition t_5. The normal release logic is resumed.

3.3 Performance Measures

In simulation studies, we are interested in system throughput, operator and buffer utilization, conveyor segment density as well as item sojourn time. Fluid simulation generate three types of function of time: IFF function $\phi(t)$ for every batch transition, the buffer content function $b(t)$ for every continuous place and accumulation position function $a(t)$ for every batch place. Various node, segment and network performance measures of interest can be formulated in terms of these functions. For instance,

The utilization of a particular machine or operator modeled as a transition t_i is given by

$$\bar{u}_i = \frac{\int_0^T \phi_i(t)dt}{\int_0^T \Phi_i(t)dt}.$$

The utilization of a conveyor segment modeled as batch place p_i is given by

$$\bar{u}_i = \frac{\int_0^T \sum_{\{t_j \in T_b : t_j \in p_i^o\}} \phi_j(t)dt}{\int_0^T (v_i \cdot d_i)dt}.$$

Average workload of a buffer and average accumulation length of conveyor segment are given by

$$\overline{W}_i = \frac{1}{T}\int_0^T b_i(t)dt$$

$$\bar{A}L_i = \frac{1}{T}\int_0^T a_i(t)dt$$

respectively.

To get network level performance measure, we can add two continuous places with infinite buffer capacity p_k and p_l to denote a virtual source and a virtual sink. Let all flow originate from p_k and terminate at p_l. System throughput is given by the total amount of fluid into sink during [0,T] divided by T. Average system sojourn time is a little harder to derive since the individual identity is lost in fluid simulation. An approximate method using the input-output diagram approach can be done as follows. Draw a function of accumulative amount of fluid out of source and a function of accumulative amount of fluid into the sink as functions of time. The horizontal distance between these two curves along any horizontal straight line is the time spending at the system of a unit material in a particular position of arrival sequence if we assume this sequence is preserved during transportation. So the average sojourn time can be computed by the area enclosed by these two curves and any two horizontal lines divided by the vertical coordinator difference of these two horizontal line.

3.4 Computation Time Comparison

We did some preliminary computational comparison of cell based simulation approach and fluid simulation approach based on the example network in authors' paper 2004. The configuration is a hypothetic sortation facility with 3 miles total conveyor length. The multi-sources input model includes constant stream, deterministic on-off and exponential on-off input models. We use Arena to build a cell-based simulation model and developed S-BPN based fluid simulator using Java. We gradually increase the input intensity and keep the constant transportation capacity so that the WIP level will increase accordingly. Figure 5 shows the relationship between the simulation run time of 2 hours real time period and the WIP level of two approaches. We use log scale of run time in the vertical axis because otherwise the scale difference is too great.

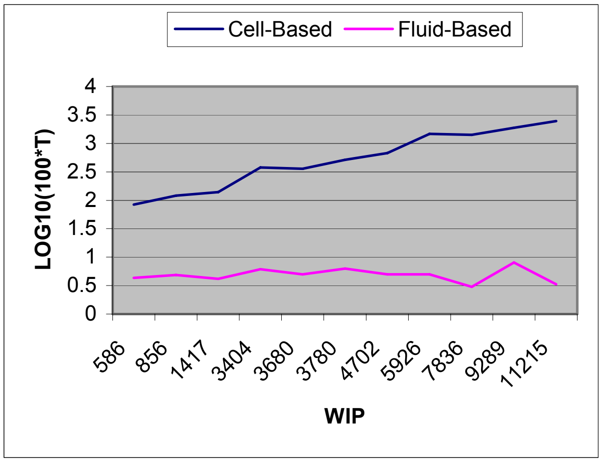

Figure 5. Simulation Runtime vs. WIP

We can see the cell-based simulation time increase exponentially as the WIP increase while the fluid based simulation is much faster and is not sensitive to the number of packages on the conveyor. This demonstrates the advantage of fluid simulation on high-volume system simulation.

Next we fixed the WIP level around 4500 and gradually decrease the average interval within which the arrival rate was constant of one input source. The relationship between runtime and this interval value is shown in Figure 6.

We use the ratio of mean rate change interval to mean arrival interval as the horizontal axis label. The ratio has little impact to cell-based simulation runtime, while fluid simulation runtime increased drastically as the ratio approaches one. This trend is expected since the number of events in fluid simulation will converge to the number of events in cell-based simulation as the batches becomes smaller and smaller. So the potential for computational sa-vings of fluid simulation is greatest when rates remain constant for significant periods of time. That's why we emphasize a slow changing environment is the key factor to justify using of fluid simulation.

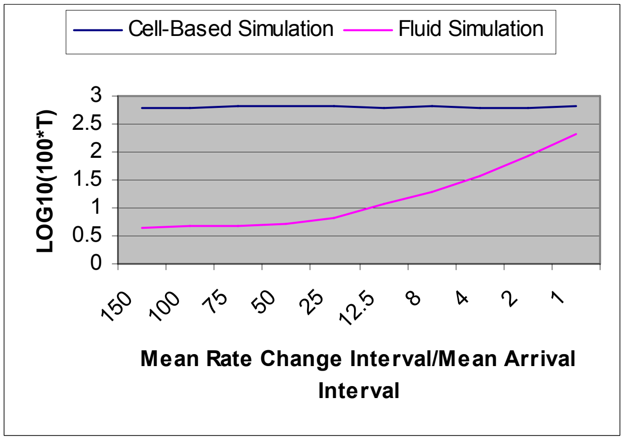

Figure 6. Simulation Runtime vs Release Interval

4 CONCLUSIONS

S-BPN is presented as a framework for fluid simulation of high volume large conveyor network. It utilizes the batch formalism proposed in Batches Petri Net model to describe the continuous circulation behavior of conveyance system. In initial S-BPN definition, we modified some BPN definitions, added new elements, trimmed many structure constraints and added new IFF calculation algorithms. In this paper , we further extends and updates the definition of S-BPN to facilitate modeling finite capacity buffer, accumulation-based control logic and integrating with other simulation units. The revised S-BPN is capable of modeling fairly sophisticated transportation oriented conveyor network in a simplified and transparent manner.

we address the feasibility of implementing fluid-simulation from modeling capabilities in terms of input and control logic, performance measures and computation time. Showed from preliminary computational experiments, S-BPN based fluid simulator generates much less discrete events compared to the item-level cell-based simulation in the current discrete event simulation system when simulating a high volume complex network operating on a slowly changing environment. Since it is a suitable high-level simulation approach, future research focuses on the investigation of the possibility of combining it with simulation optimization techniques to make it helpful in system design phase.

APPENDIX A: MACRO EVENTS LIST

Macro events are the events that can lead to possible state change. Macro events in S-BPN are defined as follows.

1. A discrete transition fires.
2. An internal coherent batch in a batch place becomes an output batch.
3. The output internal coherent batch in a batch place becomes null.
4. The value of a sensor in a sensor arc changed.
5. An accumulation batch place becomes full.
6. A continuous place becomes empty or full.
7. A marking of continuous place reaches the corresponding arc weight thus enabling some discrete transitions.
8. Speed characteristic of a batch place is modified
9. Maximum firing flow associated with a batch transition is modified

APPENDIX B: S-BPN ENABALING AND FIRING RULES

S-BPN enabling and firing rules are summarized in Table B-1 and Table B-2.

REFERENCES

Audry, N., and Prunet, F. 1995. Controlled Batches Petri Nets. In *Proceeding of IEEE International Conference on Systems, Man and Cybernetics* 2: 1849 – 1854.

Chang, J.X. 2004. Dynamic scheduling of open multiclass queueing networks in a slowly changing environment. Doctoral dissertation, Department of Industrial Engineering, Georgia Institute of Technology, Atlanta, Georgia.

Choudhury, G. L., Mandelbaum, A., Reiman, M. I., and Whitt, W. 1997. Fluid and diffusion limits for queues in slowly changing environments. *Stochastic Models* 13:121–146.

Demongodin, I., and Prunet, F. 1993. Batches Petri Nets. In *Proceeding of IEEE international Conference on Systems, Man and Cybernetics*, Le Touquet, France 607-617.

Kesidis, G., and Singh A. 1996. Feasibility of fluid-driven simulation for ATM network. In *Proceeding of IEEE GLOBECOM* 2013-2017.

Kumaran, K., and Mitra, D. 1998. Performance and fluid simulations of a novel shared buffer management system. In *Proceeding of IEEE INFOCOM 98.*

Liu, B., and Guo, Y. 1999. Fluid simulation of large scale networks: issues and tradeoff. *PDPTA 99* 2136-2142.

Liu, B., and Figueiredo, D. 2001. A study of networks simulation efficiency :fluid simulation vs.packet-level simulation. *INFOCOM'01*, April.

Table B-1. S-BPN Enabling Conditions

	$t_j \in T_d$		$t_j \in T_b$	
	Pre Condition	Post Condition	Pre Condition	Post Condition
$p_i \in P_d$	Regular arc:$m_i(t) \geq Pre(p_i,t_j)$ Inhibit arc: $m_i(t) < Pre(p_i,t_j)$		$m_i(t) \geq Pre(p_i,t_j)$	N/A
$p_i \in P_c$	Regular arc:$m_i(t) \geq Pre(p_i,t_j)$ Inhibit arc: $m_i(t) < Pre(p_i,t_j)$	$m_i(t) + Post(p_i,t_j) \leq b_i$	$m_i(t) > 0$ or $m_i(t) = 0$ and *fed*	$m_i(t) < b_i$ or $m_i(t) = b_i$ but *drain*
$p_i \in P_b$	Regular arc: $V(Cap_j(i))=1$ inhibit arc : $V(Cap_j(i))=0$	N/A	OICB≠null	*acc*(p_i)='A': not full or full but *drain* *acc*(p_i)='N': OICB=null or OICB≠null *but drain*

Table B-2. S-BPN Firing Rules

	$t_j \in T_d$		$t_j \in T_b$	
	Preplaces	PostPlaces	Preplaces	PostPlaces
$p_i \in P_d$	Regular arc: $m_i(t) = m_i(t) - Pre(p_i,t_j)$	$m_i(t) = m_i(t) + Post(p_i,t_j)$		N/A
$p_i \in P_c$	Regular arc: $m_i(t) = m_i(t) - Pre(p_i,t_j)$	$m_i(t) = m_i(t) + Post(p_i,t_j)$	$m_i(t+dt) = m_i(t) - \phi_j(t) * Pre(p_i,t_j)$	$m_i(t+dt) = m_i(t) + \phi_j(t) * Post(p_i,t_j)$
$p_i \in P_b$		N/A	Batch Formalism	Batch Formalism

Wang, Y., and Zhou, C. 2004. Fluid-based simulation approach for high volume conveyor transportation systems. *Journal of Systems Science and Systems Engineering* 13(3): 297-317.

AUTHOR BIOGRAPHIES

YING WANG currently is a Ph.D. student of School of Industrial and System Engineering at Georgia Institute of Technology. Her research interests include the simulation modeling methodology and architectures, and simulation based design in manufacturing and logistics areas. Her e-mail address is <mailto:ying@isye.gatech.edu>.

CHEN ZHOU is an Associate Professor in the School of Industrial and Systems Engineering at Georgia Institute of Technology. His research interests include manufacturing systems, warehousing systems, robotic applications, dimensional measurement, environmentally conscious design and manufacturing. His e-mail address is <chen.zhou@isye.gatech.edu>.